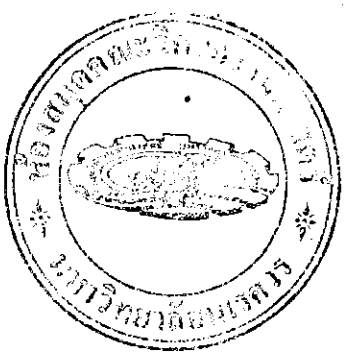
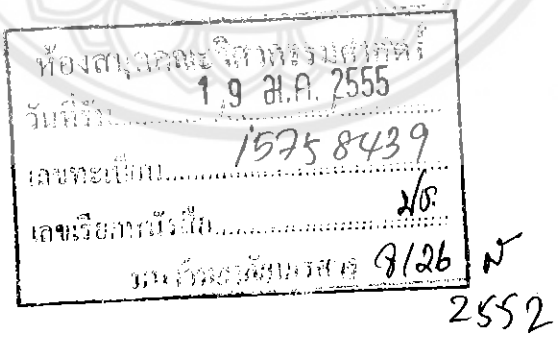


สร้างและทดลองวงจรขับและรับสัญญาณพัลส์ เพื่อติดต่อกับไมโครคอนโทรลเลอร์

BUILT AND TEST DRIVING AND RECEIVING PULSE SIGNAL CIRCUIT TO CONTACT WITH THE MICROCONTROLLER

นางสาวจงรัก สกุลนา รหัส 49360167

นางสาววราพร ไวเปีย รหัส 49361744



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2552

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	**สร้างและทดลองวงจรขับและรับสัญญาณพัลส์ เพื่อติดต่อกับไมโครคอนโทรลเลอร์**
ผู้ดำเนินโครงงาน	นางสาวจงรัก สกุลนา รหัส 49360167
	นางสาววราพร ไวเปีย รหัส 49361744
อาจารย์ที่ปรึกษา	ดร.อัครพันธ์ วงศ์กังแห
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2552

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ
(ดร.อัครพันธ์ วงศ์กังแห)

.....กรรมการ
(ดร.ชัยรัตน์ พินทอง)

.....กรรมการ
(อาจารย์เศรษฐา ตั้งค้าวานิช)

หัวข้อโครงงาน	สร้างและทดลองวงจรขับและรับสัญญาณพัลส์ เพื่อติดต่อกับไมโครคอนโทรลเลอร์
ผู้ดำเนินโครงงาน	นางสาวจงรัก สกุลนา รหัส 49360167
	นางสาววราพร ไวเปีย รหัส 49361744
อาจารย์ที่ปรึกษา	ดร.อัครพันธ์ วงศ์กังแห
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2552

บทคัดย่อ

โครงงานนี้จัดทำขึ้นเพื่อสร้างวงจรขับและรับสัญญาณพัลส์เพิ่มจำนวน 2 ชุด โดยนำมาเชื่อมต่อที่พอร์ต GPIO และ EXP Display ของบอร์ดวัดอุณหภูมิและความชื้น (AP-105) เพื่อพัฒนาบอร์ดฯ เนื่องจากบอร์ดเดิมสามารถเชื่อมต่อหัววัดฯ ได้เพียง 1 หัว แต่ปรับปรุงให้สามารถเชื่อมต่อกับหัววัดฯ ได้ 3 หัว ซึ่งจะนำไปประยุกต์ใช้กับเครื่องควบคุมอุณหภูมิและความชื้นที่จะนำหัววัดฯ ไปติดตั้งภายในอาคาร ภายนอกอาคาร และคอมเพรสเซอร์ ส่วนการทดลองนี้จัดทำเพื่อทดสอบประสิทธิภาพการทำงานของบอร์ด AP-105 แบบผ่านและไม่ผ่านวงจรขับและสัญญาณพัลส์ โดยจะเปลี่ยนระยะความยาวสายเป็น 0.2, 8, 15 และ 50 เมตร ตามลำดับ

จากผลการทดลองพบว่า วงจรที่สร้างขึ้นสามารถใช้งานได้จริง ซึ่งการวัดอุณหภูมิและความชื้นสัมพัทธ์แบบไม่ผ่านวงจรฯ ไม่สามารถวัดอุณหภูมิและความชื้นสัมพัทธ์ได้ ถ้าระยะสายสัญญาณยาวมากกว่า15 เมตร แต่การวัดผ่านวงจรฯ สามารถวัดค่าอุณหภูมิและความชื้นได้ในระยะ 1-50 เมตร โดยมีค่าความผิดพลาดเพียงเล็กน้อย

Project Title	Build and Test Driving and Receiving Pulse Signal Circuit To Contact the Microcontroller
Name	Miss Jongrak Sakunna ID. 49360167
	Miss Waraporn Waipia ID. 49361744
Project Advisor	Dr. Akaraphunt Vongkunghae
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic Year	2009

.....

ABSTRACT

This Project made to create driving and receiving pulse signal circuits 2 sets. Those circuits connected to GPIO and EXP Display Ports of Temperature and Humidity Board (AP-105) for development. Because the original board can connected only sensor, but improvements can be connected to 3 sensors. That will be applied to Temperature and Humidity Controller by installed the 3 sensors at inside, outside building, and compressor. By the testing was conducted for to test the performance of the AP-105 board in mode pass and not pass driving and receiving pulse signal circuits. And change cables are 0.2, 8, 15 and 50 meters respectively.

The test results showed that circuits can actually use. In mode non pass driving and receiving pulse signal circuits can not be measured temperature and humidity if cable more than 15 meters. But mode pass circuits can be measured temperature and humidity in the distance 1-50 meters with has little error.

กิตติกรรมประกาศ

โครงงานฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ด้วยความช่วยเหลือและแนะนำจากบุคคลหลายฝ่าย โดยเฉพาะอย่างยิ่ง

ขอขอบพระคุณ ดร.อัครพันธ์ วงศ์กังแห อาจารย์ที่ปรึกษาโครงงาน ดร.ชัยรัตน์ พินทอง และอาจารย์เศรษฐา ตั้งค้าวานิช กรรมการ ที่ได้ให้แนวคิด ความรู้ คำแนะนำ ตลอดจนเสียสละเวลา ในการตรวจ แก้ไขข้อบกพร่องต่างๆ

ขอขอบพระคุณ คุณเจริญ และคุณอำพร บัวเทศ ผู้จัดการห้างหุ้นส่วนจำกัด เซมดีไซน์ เป็นอย่างสูงที่มีส่วนร่วมในการให้ข้อมูล สนับสนุนอุปกรณ์การดำเนินงาน และช่วยเหลือในหลายๆ ด้านจนโครงงานนี้สำเร็จลุล่วงเป็นอย่างดี

ขอขอบคุณ เพื่อนๆ ที่ให้กำลังใจและความช่วยเหลือในการทำโครงงาน

ท้ายสุดนี้ ขอกราบขอบพระคุณบิดา มารดา คณาจารย์ และผู้มีพระคุณกับผู้จัดทำทุกท่านที่ได้อบรมสั่งสอน ให้ความรู้ ให้คำแนะนำ ให้คำปรึกษาที่มีคุณค่า แก่คณะผู้วิจัยด้วยดีเสมอมา

จงรัก สกุลนา
วราพร ไวเปีย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่ออังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ช

สารบัญรูป ..... ซ

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์โครงงานวิศวกรรม 1

1.3 ขอบเขตโครงงานวิศวกรรม 1

1.4 ขั้นตอนการดำเนินงาน 2

1.5 ผลที่คาดว่าจะได้รับ 2

1.6 งบประมาณ 3

บทที่ 2 ทฤษฎีและหลักการทำงาน

2.1 ไมโครคอนโทรลเลอร์ 4

2.1.1 โครงสร้างของพอร์ตไมโครคอนโทรลเลอร์ MCS-51 4

2.1.2 คุณลักษณะทั่วไปของไมโครคอนโทรลเลอร์เบอร์ P89V51RD2 5

2.2 บอร์ดวัดอุณหภูมิและความชื้น AP-105 8

2.2.1 คุณสมบัติทั่วไป 9

2.2.2 ภาพหน้าปัดของบอร์ดไมโครคอนโทรลเลอร์ 10

2.3 หัววัดอุณหภูมิและความชื้น (SHT 15) 10

2.3.1 หน้าที่ขาต่างๆ ของ SHT15 11

2.3.2 คุณลักษณะทางไฟฟ้า 12

2.3.3 การเชื่อมต่อกับหัววัดอุณหภูมิและความชื้น 13

2.3.4 การวัดอุณหภูมิและความชื้น 14

สารบัญ (ต่อ)

หน้า

2.3.5 ประสิทธิภาพในการวัดอุณหภูมิและความชื้นสัมพัทธ์ 16
2.4 การทำงานของรูปคลื่นสัญญาณในช่วงต่างๆ ที่วัดได้จากออสซิลโลสโคป 16
2.5 วงจรกรองสัญญาณความถี่ต่ำ (Low Pass Filter: LPF) 17

บทที่ 3 วิธีการดำเนินงาน

3.1 ภาพรวมของระบบ 20
3.2 วัสดุและอุปกรณ์ 21
3.2.1 อุปกรณ์สร้างแผ่นปริ้น 21
3.2.2 อุปกรณ์สร้างวงจรขับและรับสัญญาณพัลส์ 21
3.2.3 อุปกรณ์การทดลอง 21
3.3 ดำเนินการสร้าง 21
3.3.1 เขียนแบบและสร้างแผ่นปริ้น 21
3.4 อธิบายการรับ – ส่งข้อมูลของวงจรขับและรับสัญญาณพัลส์ 26
3.4.1 การส่งข้อมูล (จากไมโครคอนโทรลเลอร์ไปหัววัดฯ) 26
3.4.2 การรับข้อมูล (จากหัววัดฯ ไปไมโครคอนโทรลเลอร์) 27

บทที่ 4 การทดลองและวิเคราะห์ผลการทดลอง

4.1 การทดลองตอนที่ 1 29
4.1.1 วัตถุประสงค์การทดลองตอนที่ 1 29
4.1.2 ขั้นตอนการทดลองตอนที่ 1 29
4.1.3 ผลการทดลองตอนที่ 1 30
4.1.4 วิเคราะห์ผลการทดลองตอนที่ 1 35
4.2 การทดลองตอนที่ 2 36
4.2.1 วัตถุประสงค์การทดลองตอนที่ 2 36
4.2.2 ขั้นตอนการทดลองตอนที่ 2.1 36
4.2.3 ผลการทดลองตอนที่ 2.1 38
4.2.4 ขั้นตอนการทดลองตอนที่ 2.2 44
4.2.5 ผลการทดลองตอนที่ 2.2 45

สารบัญ (ต่อ)

หน้า

4.2.6 วิเคราะห์ผลการทดลองและอธิบายวงจรตอนที่ 2 57

บทที่ 5 สรุปการดำเนินงาน

5.1 สรุปการดำเนินงาน 58

5.2 ปัญหาที่เกิดจากการทดลอง 58

5.3 แนวทางการแก้ปัญหาและข้อเสนอแนะ 58

5.4 แนวทางในการพัฒนาเพิ่มเติม 58

เอกสารอ้างอิง 60

ภาคผนวก 61

ภาคผนวก ก 61

ภาคผนวก ข 68

ภาคผนวก ค 69

ภาคผนวก ง 73

ภาคผนวก จ 75

ประวัติผู้เขียนโครงงาน 76

สารบัญตาราง

ตารางที่		หน้า
2.1	รายละเอียดของขาใช้งานของไมโครคอนโทรลเลอร์ P89V51RD2	5
2.2	คุณลักษณะทางไฟฟ้า (DC) ของ SHT1x	12
2.3	คุณลักษณะสัญญาณอินพุตและเอาต์พุต ของ SHT1x	13
2.4	รายละเอียดคำสั่ง (List of Command)	14
4.1	ค่าอุณหภูมิและความชื้นสัมพัทธ์ ตอนที่ 1	34
4.2	ช่วงเวลาไต่ขึ้น – เวลาตกของสัญญาณ ตอนที่ 1	35
4.3	ค่าอุณหภูมิและความชื้นสัมพัทธ์ ตอนที่ 2	57
4.4	ช่วงเวลาไต่ขึ้น – เวลาตกของสัญญาณ ตอนที่ 2	57

สารบัญรูป

รูปที่		หน้า
2.1	โครงสร้างของ พอร์ตไมโครคอนโทรลเลอร์ MCS-51	4
2.2	(ก) ไมโครคอนโทรลเลอร์เบอร์ P89V51RD2	5
	(ข) ตำแหน่งขาไมโครคอนโทรลเลอร์เบอร์ P89V51RD2	5
2.3	หน้าปัดของบอร์ดวัดอุณหภูมิและความชื้น (AP-105V1.0)	9
2.4	ภาพจริงของหัววัดอุณหภูมิและความชื้น (SHT15)	10
2.5	วงจรของหัววัดอุณหภูมิและความชื้น (SHT15)	11
2.6	ลักษณะเงื่อนไขการทำงานในช่วงต่างๆของหัววัดอุณหภูมิและความชื้น (SHT15)	11
2.7	ช่วง Transmission Start	13
2.8	ลักษณะสัญญาณในการอ่านข้อมูลจากหัววัดฯ	15
2.9	ลักษณะสัญญาณในการอ่านข้อมูล T/H 2 ไบต์จากหัววัดฯ	15
2.10	(ก) กราฟแสดงค่าความถูกต้องของความชื้นสัมพัทธ์ของหัววัดแต่ละชนิด	16
	(ข) กราฟแสดงค่าความถูกต้องของอุณหภูมิของหัววัดแต่ละชนิด	16
2.11	ภาพรวมของกราฟสัญญาณ DATA I/O และ SCK	16
2.12	วงจรกรองสัญญาณความถี่ต่ำ (LPF)	17
2.13	กราฟแสดงอัตราขยายแรงดันเชิงความถี่	18
3.1	ขั้นตอนการดำเนินงาน	19
3.2	Flow Chart การทำงานของระบบเครื่องปรับอากาศที่นำไปใช้งานจริง	20
3.3	(ก) แบบเดิม และ (ข) แบบใหม่	20
3.4	วงจรขับและรับสัญญาณพัลส์ในบอร์ด AP-105 V1.0 ของบริษัทศิลารีเสิร์จ จำกัด	22
3.5	การ Save เป็น File.pcb	22
3.6	ลายวงจร	23
3.7	การปริ้นลายวงจรแบบ Mirror PCB Artwork	23
3.8	ลายวงจรที่ปริ้นแบบ Mirror PCB Artwork	24
3.9	เขียนลายวงจรลงแผ่นปริ้น	24
3.10	ลายวงจรที่เขียนเสร็จแล้ว	24
3.11	แช่ลายวงจรลงในกรดแก่	25
3.13	วงจรขับและรับสัญญาณพัลส์	26
4.1	การเชื่อมต่อสายสัญญาณของหัววัดฯ กับบอร์ด AP-105	28

สารบัญรูป(ต่อ)

รูปที่		หน้า
4.2	กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 20 ซม.	29
4.3	กราฟแสดงค่า t_r และ t_f ไม่ผ่านวงจรฯ ที่ความยาวสาย 20 ซม.	30
4.4	กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 10 เมตร.	31
4.5	กราฟแสดงค่า t_r และ t_f ไม่ผ่านวงจรฯ ที่ความยาวสาย 10 เมตร	32
4.6	กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 15 เมตร.	33
4.7	กราฟแสดงค่า t_r และ t_f ไม่ผ่านวงจรฯ ที่ความยาวสาย 15 เมตร	34
4.8	การเชื่อมต่อสายสัญญาณของหัววัดฯ กับบอร์ด AP-105	36
4.9	การเชื่อมต่อวงจรขับและรับสัญญาณพัลส์กับไมโครคอนโทรลเลอร์	37
4.10	กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 20 ซม.	38
4.11	กราฟแสดงค่า t_r และ t_f ผ่านวงจรในบอร์ด AP-105 V1.0 ที่ความยาวสาย 20 ซม.	39
4.12	กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 10 เมตร.	40
4.13	กราฟแสดงค่า t_r และ t_f ผ่านวงจรในบอร์ด AP-105 V1.0 ที่ความยาวสาย 10 เมตร	41
4.14	กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 50 เมตร.	42
4.15	กราฟแสดงค่า t_r และ t_f ผ่านวงจรในบอร์ด AP-105 V1.0 ที่ความยาวสาย 50 เมตร	43
4.16	ตำแหน่งในการวัดสัญญาณพัลส์	44
4.17	กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 20 ซม.	45
4.18	กราฟแสดงค่า t_r และ t_f ผ่านวงจรฯ ชุดที่ 1 (พอร์ต GPIO) ที่ความยาวสาย 20 ซม.	46
4.19	กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 10 เมตร.	47
4.20	กราฟแสดงค่า t_r และ t_f ผ่านวงจรฯ ชุดที่ 1 (พอร์ต GPIO) ที่ความยาวสาย 10 เมตร	48
4.21	กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 50 เมตร.	49
4.22	กราฟแสดงค่า t_r และ t_f ผ่านวงจรฯ ชุดที่ 1 (พอร์ต GPIO) ที่ความยาวสาย 50 เมตร	50
4.23	กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 20 ซม.	51
4.24	กราฟแสดงค่า t_r และ t_f ผ่านวงจรฯ ชุดที่ 2 (พอร์ต EXP Display) ที่ความยาวสาย 20 ซม.	52
4.25	กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 10 เมตร.	53
4.26	กราฟแสดงค่า t_r และ t_f ผ่านวงจรฯ ชุดที่ 2 (พอร์ต EXP Display) ที่ความยาวสาย 10 ม.	54
4.27	กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 50 เมตร.	55
4.28	กราฟแสดงค่า t_r และ t_f ผ่านวงจรฯ ชุดที่ 2 (พอร์ต EXP Display) ที่ความยาวสาย 50 ม.	56

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เนื่องจากคณะผู้จัดทำได้เข้าฝึกงานที่ห้างหุ้นส่วนจำกัดเซมดีไซน์ ซึ่งเป็นนิติบุคคลที่ดำเนินธุรกิจเกี่ยวกับการออกแบบ รับเหมาก่อสร้างและให้คำปรึกษา ด้านการอนุรักษ์พลังงานและสิ่งแวดล้อม ในช่วงที่ฝึกงานได้รับมอบหมายให้มีส่วนร่วมในการออกแบบและเขียนแบบงานระบบปรับอากาศอาคารศูนย์วิจัยพลังงานและสิ่งแวดล้อม ของมหาวิทยาลัยราชภัฏอุตรดิตถ์ ซึ่งระบบปรับอากาศนี้ต้องอาศัยเครื่องควบคุมอุณหภูมิและความชื้นชนิดที่ต้องใช้หัววัดอุณหภูมิและความชื้นจำนวน 3 หัว และมีระยะความยาวสายสัญญาณระหว่างหัววัดอุณหภูมิและความชื้นกับเครื่องควบคุมฯ ประมาณ 20 – 50 เมตร ซึ่งเครื่องควบคุม ฯ ที่ หจก. ฯ ใช้นั้นเป็นบอร์ดรุ่น AP-105 V1.0 ของบริษัทศิลารีเสิร์จ จำกัด สามารถเชื่อมต่อกับหัววัดได้เพียง 1 หัว ดังนั้นจึงต้องสร้างวงจรขับสัญญาณเพิ่มจำนวน 2 ชุด เพื่อเชื่อมต่อกับหัววัดฯ ที่เหลือ

ดังนั้นคณะผู้จัดทำจึงได้เสนอตัวขอมีส่วนร่วมในการสร้างและทดลองวงจรขับและรับสัญญาณพัลส์ ซึ่งจะใช้ร่วมกับบอร์ด AP-105 ที่ใช้วัดค่าอุณหภูมิและความชื้นจำนวน 3 หัว ซึ่งโครงงานนี้พัฒนาจากโครงงานเดิมเรื่อง เครื่องควบคุมอุณหภูมิและความชื้น ของนายจักรพันธ์ หวาจ้อย และนายปรัชญา จันทร์คามี ปีการศึกษา 2551 ซึ่งจากโครงงานเดิมใช้หัววัดอุณหภูมิและความชื้นจำนวน 1 หัวเท่านั้น

1.2 วัตถุประสงค์โครงงานวิศวกรรม

1.2.1 เพื่อสร้างวงจรขับและรับสัญญาณพัลส์เพิ่มจำนวน 2 ชุด

1.2.2 เพื่อต้องการวัดอุณหภูมิและความชื้นให้มีความเที่ยงตรง ที่ระยะความยาวสายสัญญาณระหว่าง 1-50 เมตร

1.2.3 เพื่อเปรียบเทียบประสิทธิภาพในการทำงานของหัววัดอุณหภูมิและความชื้น (SHT15) แบบผ่านวงจรฯ กับแบบไม่ผ่านวงจรฯ

1.3 ขอบเขตโครงงานวิศวกรรม

1.3.1 ศึกษาวงจรขับและรับสัญญาณในบอร์ด AP-105 V1.0 ของบริษัทศิลารีเสิร์จ จำกัด

1.3.2 ศึกษาคุณสมบัติของหัววัดอุณหภูมิและความชื้น SHT15

1.3.3 ศึกษาโปรแกรม PCB Wizard เพื่อสร้างลายแผ่นปริ้น

1.3.5 สร้างวงจรขับและรับสัญญาณพัลส์จำนวน 2 ชุด

1.3.6 ทดลองตอนที่ 1: ทดสอบประสิทธิภาพในการวัดอุณหภูมิและความชื้น โดยไม่ผ่านวงจรขับและรับสัญญาณพัลส์ ที่ระยะความยาวสายสัญญาณ 0.2, 8, 15 เมตร

1.3.7 ทดลองตอนที่ 2 : ทดสอบและเปรียบเทียบประสิทธิภาพในการวัดอุณหภูมิและความชื้น โดยผ่านวงจรฯ ในบอร์ด AP-105 และผ่านวงจรฯ ที่สร้างเพิ่ม 2 ชุด ที่ระยะความยาวสายสัญญาณ 0.2, 8, 50 เมตร

1.3.8 วิเคราะห์ผลการทดลอง

1.3.9 สรุปผลการดำเนินงาน

1.4 ขั้นตอนการดำเนินงาน

หัวข้องาน	พ.ศ.2552							พ.ศ. 2553		
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
ศึกษาวงจรขับและรับสัญญาณพัลส์	←	→								
ศึกษาคุณสมบัติของหัววัดฯ SHT15		←	→							
ศึกษาโปรแกรม PCB Wizard			←	→						
สร้างวงจรขับและรับสัญญาณพัลส์จำนวน 2 ชุด				←	→					
ทดลองตอนที่ 1 และ 2						←	→			
วิเคราะห์ผลการทดลอง								←	→	
สรุปผลการดำเนินงาน										← →

1.5 ผลที่คาดว่าจะได้รับ

1.5.1 มีชุดวงจรขับและรับสัญญาณพัลส์ต้นแบบ ที่ใช้วัดอุณหภูมิและความชื้นที่มีระยะความยาวสายสัญญาณ 1 – 50 เมตร

1.5.2 มีข้อมูลในการสร้างชุดวงจรขับและรับสัญญาณต้นแบบ

1.6 งบประมาณ

1. ค่าเอกสาร	500 บาท
2. ค่าวัสดุอุปกรณ์	
- อุปกรณ์อิเล็กทรอนิกส์	300 บาท
- หัววัดอุณหภูมิและความชื้น 3 หัว	3,000 บาท
3. ค่าจัดทำรูปเล่มรายงาน	1,000 บาท
4. อื่น ๆ	500 บาท
รวม	**5,300 บาท**

(ห้าพันสามร้อยบาทถ้วน)

หมายเหตุ ถัวเฉลี่ยทุกรายการ

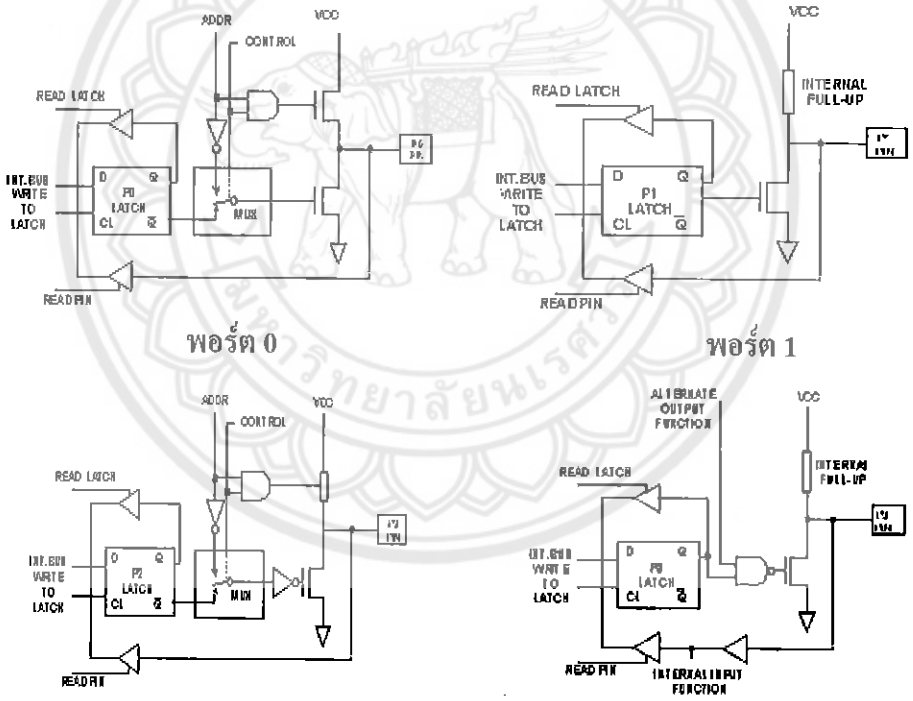
มหาวิทยาลัยนเรศวร

บทที่ 2
ทฤษฎีและหลักการทำงาน

2.1 ไมโครคอนโทรลเลอร์

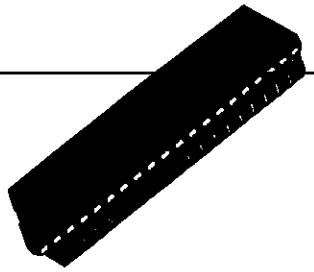
เป็นไมโครคอนโทรลเลอร์ตระกูล MCS 51 ของ Philips Semiconductor โครงสร้างภายในจะคล้ายกับ MCS-51 มาตรฐาน แต่จะมีส่วนเพิ่มเติมขึ้นมา เช่น มีหน่วยความจำภายในเป็นแบบแฟลช และมีการบรรจุโปรแกรมบูตรอม (Boot Rom) เข้าไปภายใน ทำให้สามารถเขียนหรือลบโปรแกรมได้โดยตรง โดยไม่ต้องถอดชิปออกจากวงจรที่เรียกว่า การรันโปรแกรมแบบ ISP (In System Programming)

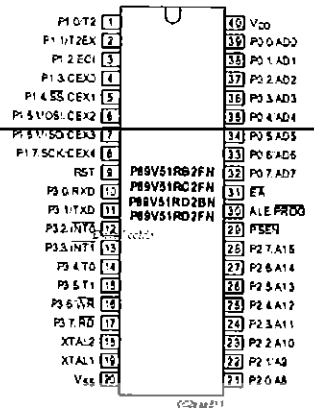
2.1.1 โครงสร้างของพอร์ตไมโครคอนโทรลเลอร์ MCS-51



รูปที่ 2.1 โครงสร้างของพอร์ตไมโครคอนโทรลเลอร์ MCS-51 [2]

2.1.2 คุณลักษณะทั่วไปของไมโครคอนโทรลเลอร์เบอร์ P89V51RD2





(ก) (ข)

รูปที่ 2.2 (ก) ไมโครคอนโทรลเลอร์เบอร์ P89V51RD2 [2]

(ข) ตำแหน่งขาไมโครคอนโทรลเลอร์เบอร์ P89V51RD2 [2]

- เป็นไมโครคอนโทรลเลอร์ 8 บิต ที่เข้ากันได้กับไมโครคอนโทรลเลอร์ MCS-51 พื้นฐาน
- หน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์เป็นแบบแฟลช ลบและเขียนใหม่ได้ถึงหนึ่งหมื่นครั้ง ขนาดของหน่วยความจำโปรแกรมสูงถึง 64 กิโลไบต์
- หน่วยความจำข้อมูลแรมภายในมีขนาด 1 กิโลไบต์
- ขาพอร์ต 8 บิต 4 พอร์ต แบบกึ่งสองทิศทาง เป็นได้ทั้งอินพุตและเอาต์พุต
- มีรีจิสเตอร์ตัวชี้ตำแหน่งข้อมูลหรือ DPTR 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 8 ประเภท
- กำหนดนัยสำคัญของการตอบสนองอินเตอร์รัปต์ได้ 4 ระดับ
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
- ไทเมอร์/เคาน์เตอร์ขนาด 16 บิต 3 ตัว (ไทเมอร์ 0, 1 และ 2)

ตารางที่ 2.1 รายละเอียดของขาใช้งานของไมโครคอนโทรลเลอร์ P89V51RD2

ชื่อขา	ขาที่	ชนิด	หน้าที่และการทำงาน
Vcc	40	อินพุต	ต่อไฟเลี้ยง +5V
GND	20	อินพุต	ต่อกราวด์
P0.0- P0.7	39 32	อินพุต/ เอาต์พุต	- ใช้งานเป็นขาพอร์ตอินพุตเอาต์พุต ถ้าต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุต สามารถทำได้โดยการเขียน

ตารางที่ 2.1(ต่อ) รายละเอียดของขาใช้งานของไมโครคอนโทรลเลอร์ P89V51RD2

ชื่อขา	ขาที่	ชนิด	หน้าที่และการทำงาน
			ข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ทำให้มีสถานะลอย (float) ค่าอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ - ใช้ในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้การมัลติเพล็กซ์เข้าช่วยเพื่อสลับการทำงานให้เป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูลในการติดต่อกับหน่วยความจำภายนอก
P1.0-P1.7	1-8	อินพุต/เอาต์พุต	- ใช้งานเป็นขาพอร์ตอินพุตเอาต์พุตสำหรับใช้งานทั่วไป เฉพาะขา P1.5-P1.7 สามารถขับกระแสได้สูง 16 mA ต่อขา - เป็นขาสัญญาณของไทเมอร์ 2 และขาสัญญาณของโมดูล PCA ดังมีรายละเอียดต่อไปนี้ T2 (P1.0 : ขา 1) เป็นขาอินพุตสำหรับนับค่าของไทเมอร์ 2 และขาเอาต์พุตสัญญาณนาฬิกาแบบได้ T2EX (P1.1 : ขา 2) เป็นขาอินพุตสำหรับควบคุมการทำงานของไทเมอร์/เคาน์เตอร์ 2 ECI (P1.2 : ขา 3) เป็นขาอินพุตสัญญาณนาฬิกาจากภายนอกสำหรับโมดูล PCA CEX0 (P1.3 : ขา 4) เป็นขาอินพุตเอาต์พุตภายนอกของวงจรตรวจจับและเปรียบเทียบสัญญาณสำหรับ PAC โมดูล 0 CEX1 (P1.4 : ขา 5) เป็นขาอินพุตเอาต์พุตภายนอกของวงจรตรวจจับและเปรียบเทียบสัญญาณสำหรับ PAC โมดูล 1
P2.0-P2.7	21.28	อินพุต/เอาต์พุต	- ใช้งานเป็นขาพอร์ตอินพุตเอาต์พุตสำหรับใช้งานทั่วไป - ใช้ต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15) เมื่อติดต่อด้วย
P3.0-P3.7	10-17	อินพุต/เอาต์พุต	- ใช้งานเป็นขาพอร์ตอินพุตสำหรับใช้งานทั่วไป - ใช้งานเป็นขาพอร์ตหน้าที่พิเศษ มีรายละเอียดดังต่อไปนี้ RxD (P3.0 : ขา 10) ใช้เป็นขาอินพุตสำหรับรับข้อมูลสื่อสารแบบอนุกรม TxD (P3.0 : ขา 11) ใช้เป็นขาอินพุตสำหรับรับข้อมูลสื่อสารแบบอนุกรม

ตารางที่ 2.1 (ต่อ) รายละเอียดของขาใช้งานของไมโครคอนโทรลเลอร์ P89V51RD2

ชื่อขา	ขาที่	ชนิด	หน้าที่และการทำงาน
			INT0 (P3.0 : ขา 12) ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากนอกช่อง 0 INT1 (P3.0 : ขา 13) ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากนอกช่อง 1 T0 (P3.0 : ขา 14) ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0 T1 (P3.0 : ขา 15) ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 WR (P3.0 : ขา 16) ใช้เป็นขาสัญญาณ WR ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก RD (P3.0 : ขา 17) ใช้เป็นขาสัญญาณ RD ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก
RESET	9	อินพุต	ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณรอจิก "1" อย่างน้อยเป็นเวลา 2 แมชีนไซเกิล โดยที่วงจรกำหนดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างปกติ
ALE	30	เอาต์พุต	Address Latch Enable ออกมาทุกๆ แมชชีนไซเกิล อย่างไรก็ตาม สามารถดิสเอเบิลสัญญาณพัลส์นี้ได้ โดยการเซตบิต 0 ของรีจิสเตอร์ AUXR
PSEN	29	เอาต์พุต	- Program Store Enable : ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้ง - นอกจากนี้ยังใช้ประกอบในการอ่าน-เขียนข้อมูลในหน่วยความจำโปรแกรมด้วยกระบวนการ ISP สำหรับเบอร์ P89V51RD+ ให้ต่อขานี้ลงกราวด์ แล้วป้อนไฟ+12V (±0.5V) เข้าที่ขา EA/Vpp สำหรับเบอร์ P89V51RD2 ให้ต่อขานี้ลงกราวด์, ป้อนลอจิก "1" เข้าที่ขา P2.7 และป้อนแรงดัน +5V เข้าที่ขา EA/Vpp

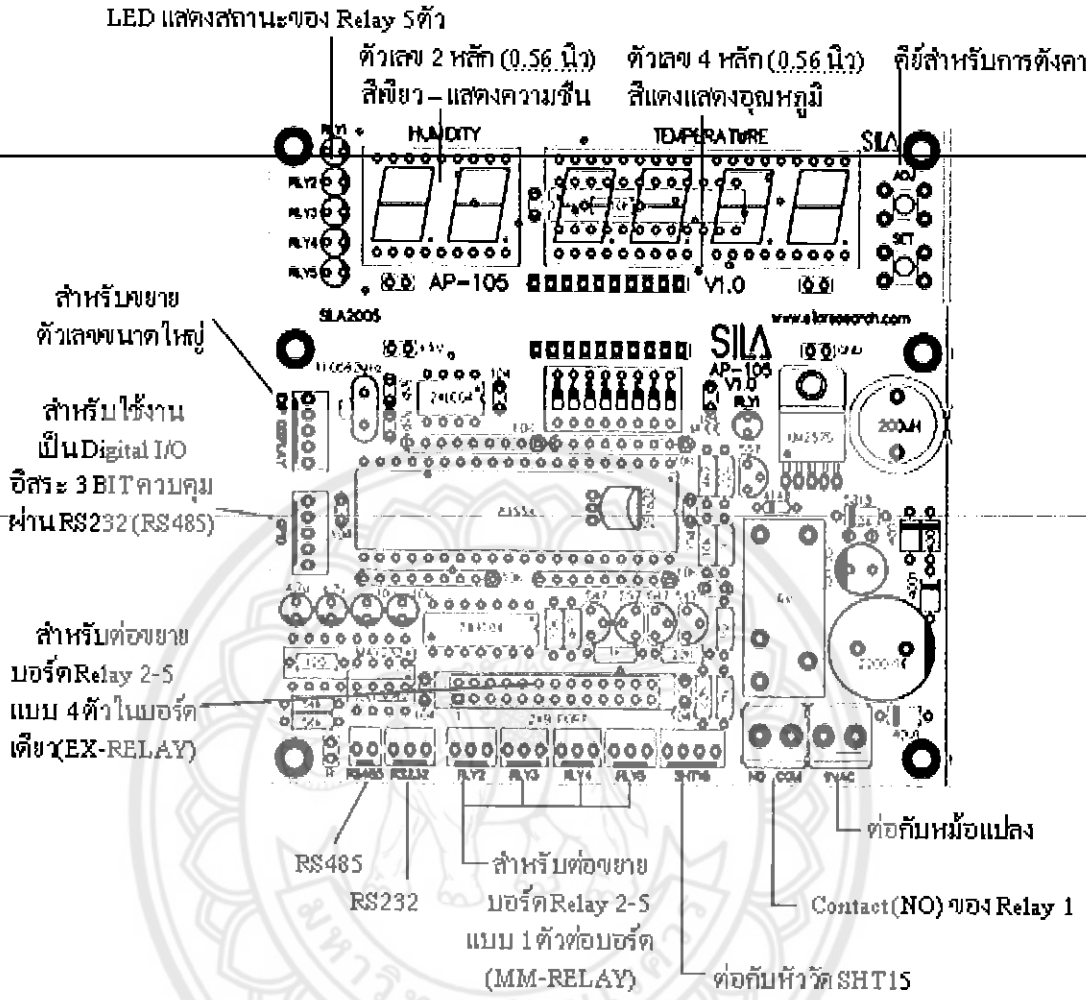
ตารางที่ 2.1 (ต่อ) รายละเอียดของขาใช้งานของไมโครคอนโทรลเลอร์ P89V51RD2 (ต่อ)

ชื่อขา	ขาที่	ชนิด	หน้าที่และการทำงาน
EA/Vpp	31	อินพุต	- External Access enable/Programming voltage input : ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ "0" เลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก "1" เลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายใน - นอกจากนี้ที่ขานี้ยังใช้เป็นอินพุตสำหรับรับแรงดัน สำหรับโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับเบอร์ P89V51RD+ ต้องการแรงดัน +12V (±0.5V) สำหรับเบอร์ P89V51RD2 ต้องการแรงดัน +5V
XTAL1	19	อินพุต	ขาอินพุตรับสัญญาณจากวงจรขยายออสซิลเลเตอร์ (ขา XTAL2) และจากภายนอก ในการใช้งานปกติ ขานี้และขา XTAL2 ต่อเข้ากับคริสตอลและตัวเก็บประจุชดเชยค่าน้อยๆ
XTAL2	18	อินพุต	ขาอินพุตของวงจรขยายออสซิลเลเตอร์ภายในไมโครคอนโทรลเลอร์ ในการใช้งานปกติ ขานี้และขา XTAL1 ต่อเข้ากับคริสตอลและตัวเก็บประจุชดเชยค่าน้อยๆ

2.2 บอร์ดวัดอุณหภูมิและความชื้น AP-105

AP-105 คือบอร์ดวัดอุณหภูมิและความชื้น หัววัดแบบดิจิตอลความละเอียดสูง ใช้วัดในบรรยากาศใช้วัดอุณหภูมิได้ตั้งแต่ -40 ถึง 120 องศาเซลเซียส ความละเอียด 0.1 องศา และวัดความชื้นได้ตั้งแต่ 10 ถึง 90% ความละเอียด 1% สามารถตั้งควบคุม Relay ได้ถึง 5 ตัว (บนบอร์ดมี Relay ให้ 1 ตัวจากนั้นต้องขยายเพิ่ม) โดย Relay แต่ละตัว สามารถเลือกควบคุมด้วยอุณหภูมิหรือความชื้นได้และเลือกการควบคุมเป็นแบบ คุมร้อน (Hot) หรือ คุมเย็น (Cool) ก็ได้ ตั้งค่า High และ Low เพื่อการควบคุมนอกจากนี้ยังมีพอร์ตการสื่อสาร RS232(RS485) เพื่อรับ,ส่งข้อมูลกับเครื่องคอมพิวเตอร์ PC และต่อเป็นระบบ Network ได้ ใช้งานง่ายด้วยปุ่มบนหน้าปัดเพียง 2 ปุ่ม ประยุกต์ใช้กับงานวัดอุณหภูมิและความชื้นได้หลากหลายรูปแบบ

2.2.1 ภาพหน้าปัดของบอร์ดไมโครคอนโทรลเลอร์



รูปที่ 2.3 หน้าปัดของบอร์ดวัดอุณหภูมิและความชื้น

2.2.1 คุณสมบัติทั่วไป

2.2.1.1 ทำงานด้วยไมโครคอนโทรลเลอร์เบอร์ 89C55WD ความถี่ 11.0592 Mhz

2.2.1.2 จอแสดงผล 7-Segment LED ความสูง 0.56 นิ้ว จำนวน 6 หลัก โดยแสดงค่าความชื้น 2 หลักด้วยตัวเลขสีเขียว และค่าอุณหภูมิ 4 หลัก (xxxx) ด้วยตัวเลขสีแดง

2.2.1.3 มี LED สีแดงแสดงสถานะของ Relay 5 ตัว

2.2.1.4 มีคีย์ 2 คีย์บนหน้าปัด สำหรับการตั้งค่าต่าง ๆ โดยจะเก็บค่าไว้ในหน่วยความจำ EEprom

2.2.1.5 หัววัดอุณหภูมิแบบดิจิตอลเบอร์ SHT15 วัดได้ตั้งแต่ -40 ถึง 120 องศาเซลเซียส ความละเอียด 0.1 องศา และวัดความชื้นได้ตั้งแต่ 10 ถึง 90% ความละเอียด 1% ใช้วัดในบรรยากาศเท่านั้นและมีวงจรทำให้ต่อสายยาวได้ถึง 100 เมตร (สายที่มากับสินค้ายาว 5 ฟุต)

2.2.1.6 มี Relay Output 1 ตัว ต่อใช้งานแบบ Normal Open 250 VAC 10 A และสามารถขยาย Relay ได้อีก 4 ตัว ผ่านทางขั้วต่อ 3 Pin จำนวน 4 จุด เพื่อเพิ่มบอร์ด Relay 1 ตัวต่อจุด หรือผ่านทางขั้วต่อ 24 Pin เพื่อเพิ่มบอร์ด Relay 4 ตัว ในบอร์ดเดียว

2.2.1.7 มีพอร์ตสื่อสาร RS232 หรือ RS485 (ตัวชิพ MAX232, MAX3082 เป็น Option)

2.2.1.8 มีพอร์ต 3B สำหรับการขยายตัวเลขขนาดใหญ่ ด้วยบอร์ดตัวเลขของศิลา

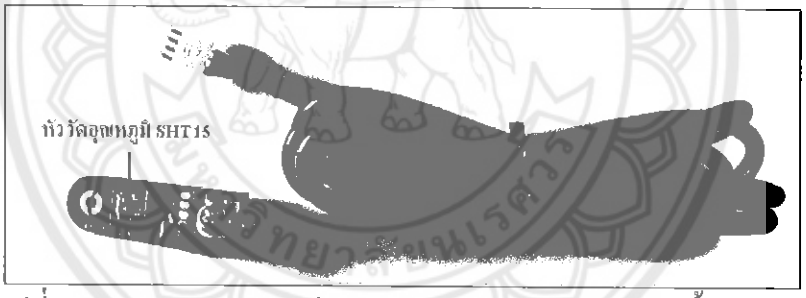
2.2.1.9 มีพอร์ต 3B (GPIO) สำหรับประยุกต์ใช้งานต่าง ๆ สั่งงานด้วยคำสั่งจาก RS232 (RS485) ได้

2.2.1.10 มี Dip-Switch ขนาด 8 จุด สำหรับตั้ง Address สำหรับการสื่อสารแบบ Network และตั้งความสว่างของตัวเลข รวมทั้งตั้งลักษณะการแสดงผลของตัวเลขที่ขยายใหญ่

2.2.1.11 มีหน่วยความจำแบบ EEprom 24LC04 สำหรับเก็บค่า Setup ที่ตั้งไว้

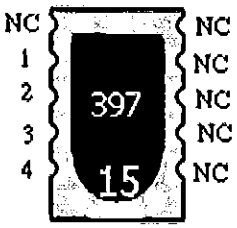
2.2.1.12 ไฟเลี้ยง 220 VAC ผ่านหม้อแปลงเป็น 9 VAC จ่ายไปยังบอร์ด มีวงจรจ่ายไฟแบบ Switching ทนทาน, ไม่ร้อน

2.3 หัววัดอุณหภูมิ SHT15



รูปที่ 2.4 ภาพจริงของการใช้งานของหัววัดอุณหภูมิและความชื้น SHT15

2.3.1 หน้าที่ขาต่างๆของ SHT15



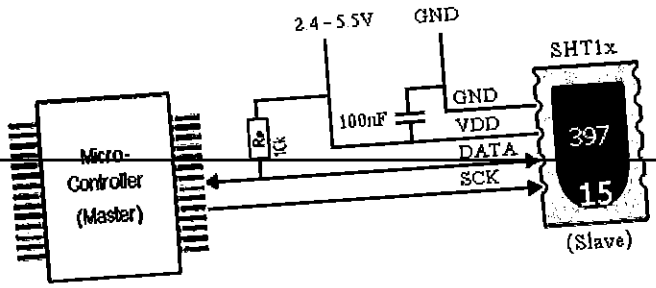
ขา 1 : Ground (GND) สายกราวด์

ขา 2 : Source Voltage (VDD) แหล่งจ่ายแรงดัน (2.4-5.5 V)

ขา 3 : Serial Data (DATA) ส่งข้อมูลได้ 2 ทาง

ขา 4 : Serial Clock Input (SCK) รับข้อมูลเข้าอย่างเดียว

2.3.1.1 Power Pins (VDD, GND) แหล่งจ่ายแรงดันของ SHT 1x จะอยู่ในช่วง 2.4-5.5 V แต่แนะนำให้ใช้แรงดัน 3.3 V ซึ่งขาของแหล่งจ่ายไฟ แหล่งจ่ายแรงดัน (VDD) และกราวนด์ (GND) จะต้องมีคาปาซิเตอร์ 100 nF ต่อลงกราวนด์อยู่ด้วย ดังรูปที่ 2.5

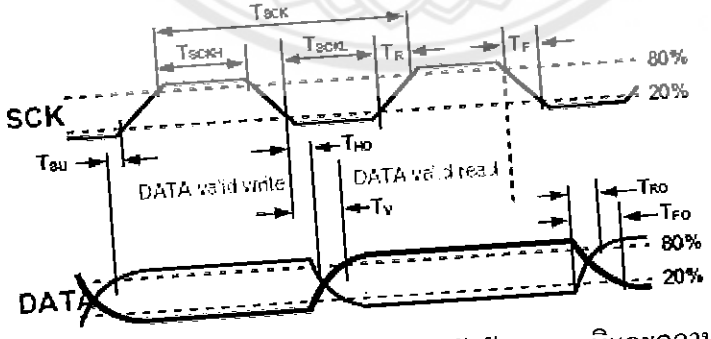


รูปที่ 2.5 วงจรของหัววัดอุณหภูมิและความชื้น SHT15

2.3.1.2 Serial Clock Input (SCK) ใช้ในการติดต่อประสานงานระหว่างไมโครคอน-โทรลเลอร์กับหัววัดฯ SHT 1x

2.3.1.3 Serial Data (DATA) ใช้ในการส่งข้อมูลเข้าและออกจากหัววัดฯ สำหรับการส่งคำสั่งไปยังหัววัดฯ ข้อมูลที่ถูกต้องจะอยู่ที่ขอบขาขึ้นของสัญญาณนาฬิกาที่ต่อเนื่องกัน (SCK) แต่หลังจากสัญญาณ SCK เป็นช่วงขอบขาลง ค่าจะเปลี่ยนแปลง เพื่อความปลอดภัยในการสื่อสารข้อมูลที่ถูกต้อง จะต้องขยาย T_{SU} และ T_{HO} ก่อนที่ขอบสัญญาณของ SCK จะเพิ่มขึ้น และหลังที่ขอบสัญญาณของ SCK จะลดลงตามลำดับ ดังรูปที่ 2.6 สำหรับการอ่านข้อมูลจากหัววัดฯข้อมูลที่ถูกต้อง ค่าของ T_v หลังจากสัญญาณ SCK จะต้องต่ำลง และยังคงเป็นเช่นนั้น จนกระทั่งถึงขอบขาลงของ SCK ถัดไป

เพื่อหลีกเลี่ยงความสับสนของสัญญาณ ไมโครคอนโทรลเลอร์จะต้องขับ Data เป็น 0 (Low) เท่านั้น การต่อ Pull-up resistor (ในตัวอย่างเป็น 10 kΩ) เพื่อต้องการดึงสัญญาณให้สูงขึ้น



รูปที่ 2.6 ลักษณะเงื่อนไขการทำงานในช่วงต่างๆของหัววัดอุณหภูมิและความชื้น SHT15

โดยDATA เส้นหนาถูกควบคุมโดยเซ็นเซอร์ ส่วน DATA เส้นปกติถูกควบคุมโดยไมโครคอนโทรลเลอร์

หมายเหตุ Pull-up resistors รวมอยู่ในอินพุตและเอาท์พุตของวงจรไมโครคอนโทรลเลอร์

2.3.2 คุณลักษณะทางไฟฟ้า

ลักษณะทางไฟฟ้า เช่น การใช้ไฟ ระดับ ต่ำหรือสูง แรงดันอินพุตและเอาท์พุตขึ้นอยู่กับแหล่งจ่ายแรงดัน ดังตารางที่ 2.2 ให้คุณลักษณะทางไฟฟ้าของ SHT 1x ถ้าไม่ได้ระบุเป็นค่าอื่น สมมติให้ใช้แรงดันไฟฟ้า 5 V สำหรับการสื่อสารที่เหมาะสมกับหัววัดฯ เป็นสิ่งที่สำคัญ เพื่อให้แน่ใจว่าสัญญาณที่ออกแบบเป็นไปอยู่ภายในขอบเขตที่จำกัด ในตาราง ที่ 2.3 และรูปที่ 2.6

ตารางที่ 2.2 คุณลักษณะทางไฟฟ้า (DC) ของ SHT1x

Parameter	**Condition**	**min**	**typ**	**max**	**Units**
Power supply DC		2.4	3.3	5.5	V
Supply current	measuring		0.55	1	mA
	average	2	28		µA
	sleep		0.3	1.5	µA
Low level output voltage	I_{OL} < 4 mA	0		250	mV
High level output voltage	R_P < 25 kΩ	90%		100%	VDD
Low level input voltage	Negative going	0%		20%	VDD
High level input voltage	Positive going	80%		100%	VDD
Input current on pads				1	µA
Output current	on			4	mA
	Tri-stated (off)		10	20	µA

ตารางที่ 2.3 คุณลักษณะสัญญาณอินพุตและเอาท์พุต ของ SHT1x

	Parameter	**Condition**	**min**	**typ**	**max**	**Units**
F_{SCK}	SCK Frequency	VDD > 4.5V	0	0.1	5	MHz
		VDD < 4.5V	0	0.1	1	MHz
T_{SCKx}	SCK hi/low time		100			ns
T_R/T_F	SCK rise/fall time		1	200	*	ns
T_{FO}	DATA fall time	OL (Output Load) = 5pF	3.5	10	20	ns
		OL (Output Load) = 10pF	30	40	200	ns
T_{RO}	DATA rise time		**	**	**	ns
T_V	DATA valid time		200	250	***	ns
T_{SU}	DATA setup time		100	150	***	ns
T_{HO}	DATA hold time		10	15	****	ns

* $T_{R_max} + T_{F_max} = (F_{SCK})^{-1} - T_{SCKH} - T_{SCKL}$

** T_{R0} is determined by the R_P*Cbus time-constant at DATA line

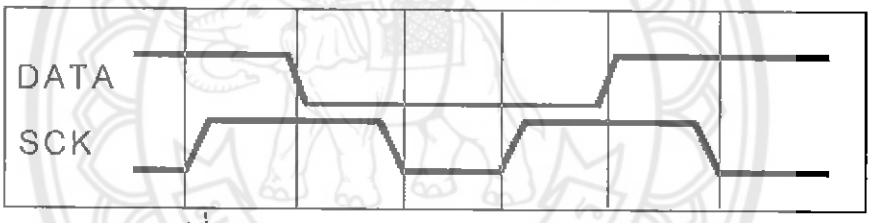
*** T_{V_max} and T_{SU_max} depend on external pull-up resistor (R_P) and total bus line capacitance (C_{bus}) at DATA line

**** $T_{H0_max} < T_{V-max}(T_{R0}, T_{F0})$

2.3.3 การเชื่อมต่อกับหัววัดอุณหภูมิและความชื้น

2.3.2.1 เริ่มต้นใช้งานหัววัด ในขั้นแรกเซ็นเซอร์จะเลือกแหล่งจ่ายแรงดัน(VDD) ซึ่งอัตราการเปลี่ยนแปลงระหว่างพลังงานไฟฟ้าจะตกไม่เกิน 1V/ms หลังจากที่กำลังไฟฟ้าเพิ่มขึ้น หัววัดต้องการเวลา 11 ms เพื่อที่จะเข้าสู่ช่วง Sleep Mode โดยจะต้องไม่มีคำสั่งใดส่งไปก่อนเวลานั้น

2.3.2.2 การส่งคำสั่ง เริ่มต้นด้วยการส่ง Transmission Start ออกมาอย่างเป็นลำดับ ซึ่งลักษณะเงื่อนไขของสัญญาณ มีลักษณะดังรูปที่ 2.7



รูปที่ 2.7 ลักษณะสัญญาณช่วง Transmission Start

- DATA จะเปลี่ยนจาก 1 เป็น 0 ขณะที่ SCK ลูกที่ 1 เป็น 1
- DATA จะเปลี่ยนจาก 0 เป็น 1 ขณะที่ SCK ลูกที่ 2 เป็น 1

หลังจากคำสั่ง Transmission Start คำสั่งที่ตามมาประกอบด้วย address มีขนาด 3 บิต (เป็น "000" เท่านั้น) และตามด้วย Command Bit อีก 5 บิต โดย SHT 1x จะแสดงการรับคำสั่งด้วยการดึงขา DATA ลงเป็น 0 (บิต ACK) หลังจากขอบขาลงลูกที่ 8 ของ SCK หลังจากนั้นสาย DATA ก็จะถูกปล่อยเป็น 1 หลังจากขอบขาลงของ SCK ลูกที่ 9

ตารางที่ 2.4 รายละเอียดคำสั่ง (List of Command)

Command	Code
สำรอง (Reserved)	0000x
วัดอุณหภูมิ (Measure Temperature)	00011
วัดความชื้น (Measure Humidity)	00101
แสดงสถานะการอ่าน (Status Register Read)	00111
แสดงสถานะการเขียน(Status Register Write)	00110
สำรอง (Reserved)	0101x-1110x
รีเซ็ท (Soft Reset)	11110

2.3.4 การวัดอุณหภูมิและความชื้นสัมพัทธ์

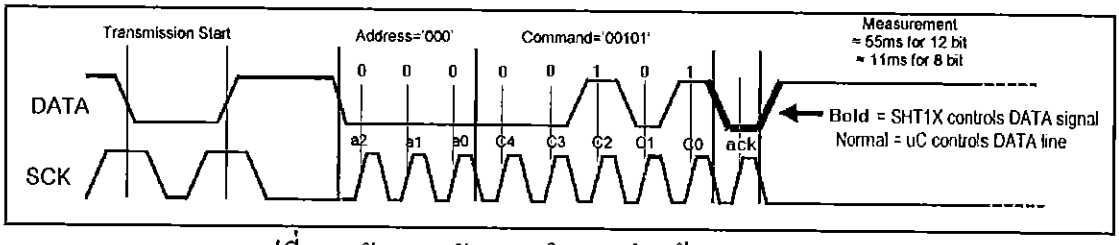
หลังจากออกคำสั่งการวัดความชื้นสัมพัทธ์และอุณหภูมิ "00000101" (สำหรับการวัดอุณหภูมิใช้ "00000011") โดยตัวควบคุมจะรอจนกว่าการวัดเสร็จสมบูรณ์ ซึ่งการวัดจะใช้เวลาสูงสุด 20/80/320 ms สำหรับการวัด 8/12/14 บิต ตามลำดับ ซึ่งเวลาที่ใช้ในการวัดจะแตกต่างกันขึ้นอยู่กับความเร็วภายในของออสซิลเลเตอร์ และสามารถลดลงได้ถึง 30%

โดย SHT1x จะดึงขา DATA เป็น 0 และเข้าสู่ช่วง Sleep Mode เพื่อรอให้สัญญาณการวัดเสร็จสิ้น ซึ่งตัวควบคุมจะต้องรอให้สัญญาณนี้พร้อมก่อนที่จะเริ่มต้น SCK ใหม่ เพื่อแสดงผลข้อมูลการวัด ซึ่งข้อมูลการวัดจะถูกเก็บไว้จนกว่าจะแสดงผลทางหน้าจอ เพื่อที่ตัวควบคุมจะสามารถทำงานอื่นต่อไปได้ และสะดวกในการแสดงผลข้อมูล

ต่อจากนั้นก็จะส่ง SCK ต่อไปอีก 2 ไบต์ เพื่อรับข้อมูลการวัดค่าความชื้นสัมพัทธ์ และ 1 ไบต์ของการตรวจสอบข้อมูล CRC Checksum (ถ้ามี) จะถูกส่งต่อไป โดยที่ไมโครคอนโทรลเลอร์จะรับทราบถึงการส่งข้อมูลแต่ละไบต์ด้วยการดึงขา DATA เป็น 0 (Low)

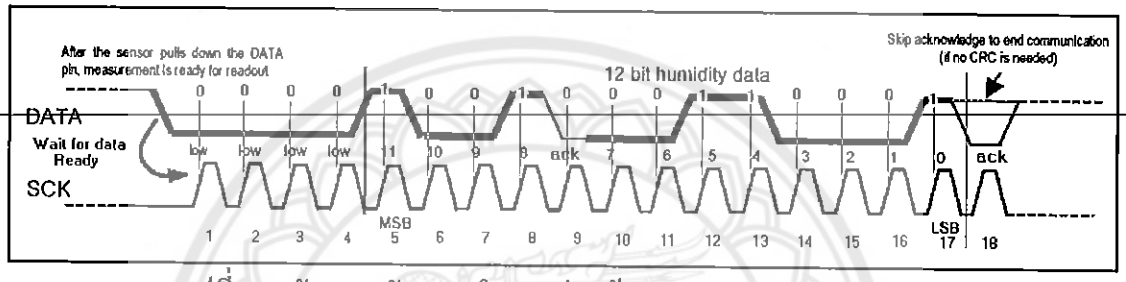
การสื่อสารจะสิ้นสุดลงหลังจากรับทราบบิตข้อมูลของ CRC แต่ถ้าไม่ๆด้ใช้ CRC-8 checksum ก็จะยุติการสื่อสารหลังจากข้อมูลการวัดของ LSB โดยทำให้ ACK เป็น 1 หลังจากนั้นอุปกรณ์จะกลับเข้าไปสู่ช่วง Sleep Mode

ตัวอย่าง เมื่อไมโครคอนโทรลเลอร์ ต้องการอ่านค่าความชื้นสัมพัทธ์จากหัววัดฯ ซึ่งมี Address เป็น 000 และ Command Bit เป็น 00101 จะมีชุดคำสั่งดังนี้



รูปที่ 2.8 ลักษณะสัญญาณในการอ่านข้อมูลจากหัววัดฯ

เมื่อหัววัดฯ รับทราบคำสั่งแล้วจะตอบรับคำสั่งนั้นโดย ACK ด้วยการดึงขา DATA ลงเป็น 0 (เส้นทึบ)



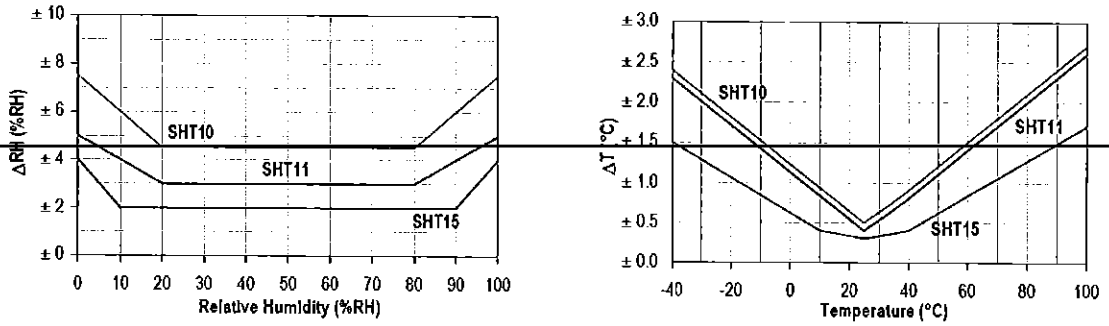
รูปที่ 2.9 ลักษณะสัญญาณในการอ่านข้อมูล T/H 2 Byte จากหัววัดฯ

ข้อมูลของความชื้นมีขนาด 12 บิต เมื่อตอบ ACK แล้วให้ไมโครฯ รอประมาณ 210 ms เพื่อให้หัววัดฯ พร้อม แล้วจึงส่งสัญญาณ SCK ต่อไปอีก 2 ไบต์ สำหรับรับข้อมูล และ 1 ไบต์ สำหรับตรวจสอบความผิดพลาดของข้อมูล (CRC)

จากตัวอย่างจะสามารถอ่านข้อมูล 12 บิต ของความชื้น (4 บิตแรกเป็น 0 เสมอ) จะได้ 0000 1001 0011 0001 = 2353 (dec) = 75.79%RH

เมื่อได้ข้อมูลครบแล้ว หากต้องการ CRC Checksum ให้ตอบรับทราบด้วยการดึงขา DATA ลงเป็น 0 (Low) หากไม่ต้องการก็ข้ามขั้นตอนนี้ไป

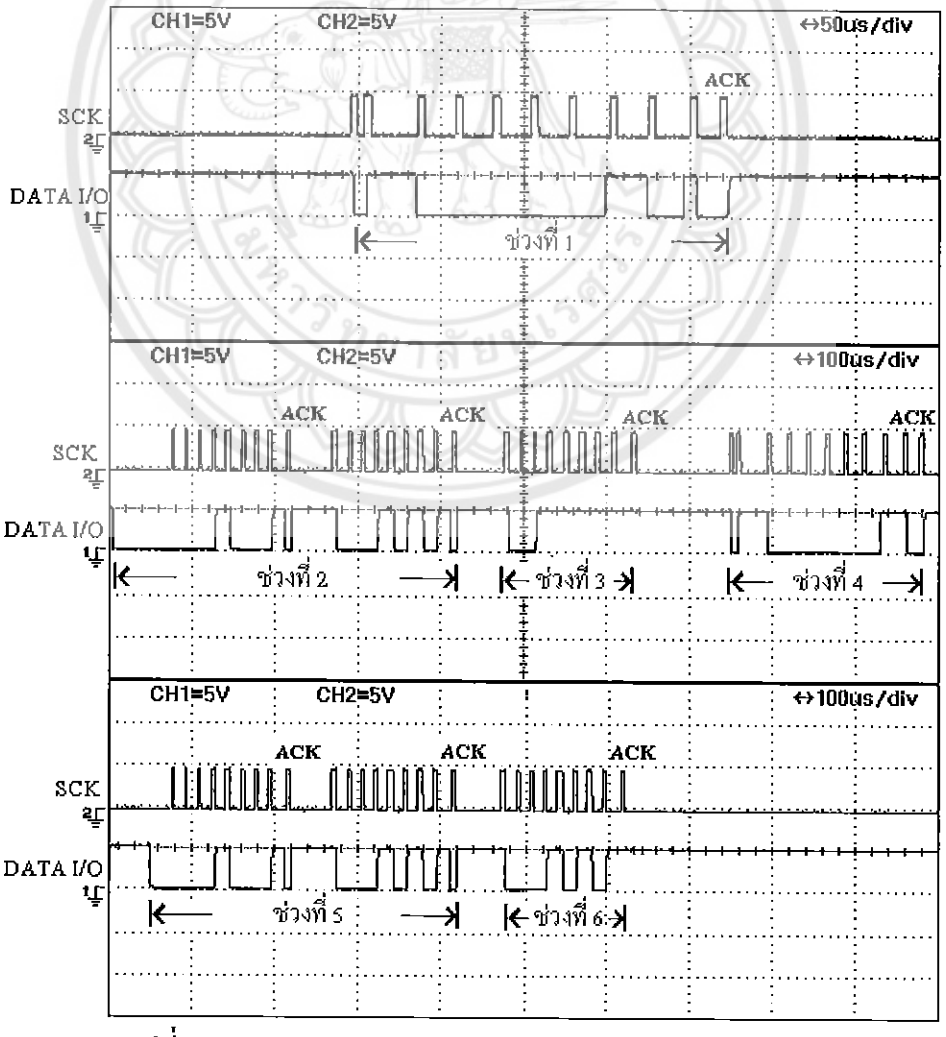
2.3.5 ประสิทธิภาพในการวัดอุณหภูมิและความชื้นสัมพัทธ์



(ก) (ข)

รูปที่ 2.10 (ก) กราฟแสดงค่าความถูกต้องของความชื้นสัมพัทธ์ของหัววัดแต่ละชนิด
(ข) กราฟแสดงค่าความถูกต้องของอุณหภูมิของหัววัดแต่ละชนิด

2.4 การทำงานของรูปคลื่นสัญญาณในช่วงต่างๆ ที่วัดได้จากออสซิลโลสโคป



รูปที่ 2.11 ภาพรวมของกราฟสัญญาณ DATA I/O และ SCLK

ช่วงที่ 1 : เป็นช่วงการส่งคำสั่งจากไมโครคอนโทรลเลอร์ให้อ่านค่าความชื้นสัมพัทธ์ (Relative Humidity) ขนาด 12 บิต เริ่มจากการส่ง Transmission Start และส่งค่า Address=000, Command=00101 ไปให้ Sensor และเมื่อ Sensor รับทราบคำสั่งแล้ว จะส่ง ACK ตอบด้วยการดึงขา DATA ลงเป็น 0 เพื่อให้อ่านค่าข้อมูลต่อ โดยให้ไมโครคอนโทรลเลอร์รอสักครู่ จน Sensor ทำการวัดเสร็จ

ช่วงที่ 2 : เป็นช่วงส่งสัญญาณต่อไปอีก 2 ไบต์ เพื่อรับค่าความชื้นสัมพัทธ์ที่อ่านได้จาก Sensor

ช่วงที่ 3 : เป็นช่วงการส่งสัญญาณ SCK อีก 1 ไบต์ เพื่อตรวจสอบความผิดพลาดของข้อมูลที่ส่งมาจาก Sensor หลังจากนั้นจะปล่อยให้ DATA เป็น 1 ไว้ เมื่อมีสัญญาณ SCK ครบ 9 ลูก ก็จะเข้าทำงานต่อในช่วงที่ 4

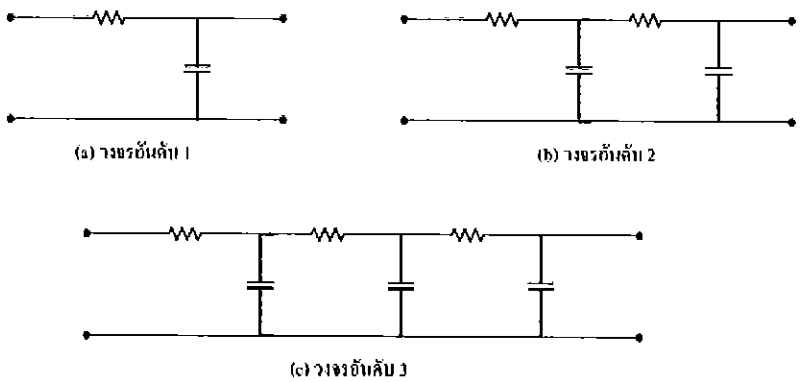
ช่วงที่4 : เป็นช่วงของการส่งคำสั่งจากไมโครคอนโทรลเลอร์ให้อ่านค่าอุณหภูมิ (Temperature) ขนาด 14 บิต จาก Sensor โดยส่งคำสั่ง Address=000 และCommand=00011 ไปให้ Sensor และเมื่อ Sensor รับทราบคำสั่งแล้ว จะส่ง ACK ตอบด้วยการดึงขา DATA ลงเป็น 0 เพื่อให้อ่านค่าข้อมูลต่อ โดยให้ไมโครคอนโทรลเลอร์รอสักครู่ จน Sensor ทำการวัดเสร็จ

ช่วงที่ 5 : เป็นช่วงการส่งสัญญาณ SCK ต่อไปอีก 2 ไบต์ เพื่อรับค่าอุณหภูมิที่อ่านได้จาก Sensor

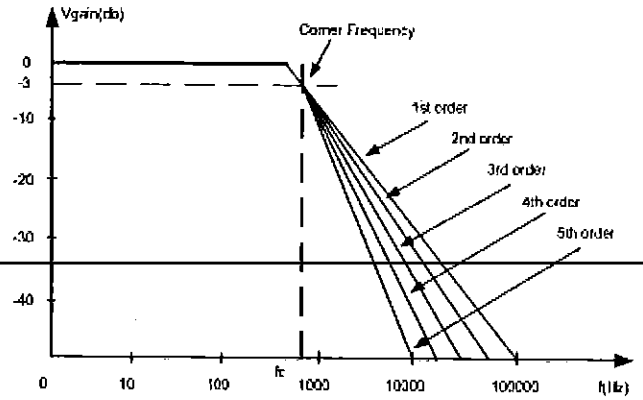
ช่วงที่ 6 : เป็นช่วงการส่งสัญญาณ SCK อีก 1 ไบต์ เพื่อตรวจสอบความผิดพลาดของข้อมูลที่ส่งมาจาก Sensor หลังจากนั้นจะปล่อยให้ DATA เป็น 1 ไว้ เมื่อมีสัญญาณ SCK ครบ 9 ลูก ก็จะเข้าสู่ช่วง Sleep Mode

2.5 วงจรกรองสัญญาณความถี่ต่ำ (Low Pass Filter: LPF)

หมายถึง วงจรที่จะยอมให้สัญญาณความถี่ตั้งแต่ 0 Hz ถึงความถี่ที่กำหนดผ่านไปได้ ส่วนความถี่ตั้งแต่ที่กำหนดสูงขึ้นไปเรื่อย ๆ จะลดทอนไปตามลำดับ ลักษณะของวงจรมีตั้งแต่อันดับหนึ่งขึ้นไป



รูปที่ 2.12 วงจรกรองสัญญาณความถี่ต่ำ (LPF) [1]



รูปที่ 2.13 กราฟแสดงอัตราขยายแรงดันเชิงความถี่ [1]

จากรูปวงจรเมื่อทำการป้อนความถี่ต่ำเข้าวงจร ที่คาปาซิเตอร์ จะมีค่า XC สูง ทำให้ความถี่ต่ำไหลผ่านความต้านทานที่มีค่าน้อยกว่า XC ได้สะดวก ระดับสัญญาณเอาท์พุตจึงผ่านได้มาก แต่เมื่อความถี่สูงกว่าจุดที่กำหนด ค่า XC จะลดลง ทำให้ความถี่ ผ่านขดลวดได้ลดลง บางส่วนที่ผ่านไปได้ก็จะถูกคาปาซิเตอร์ ดึงลงกราวด์ ระดับสัญญาณเอาท์พุตจึงผ่านได้น้อยมาก

ในทางทฤษฎีวงจรกรองความถี่ต่ำจะยอมให้สัญญาณใดๆ ที่มีความถี่ตั้งแต่ 0 Hz จนถึงความถี่ที่กำหนดซึ่งเรียกว่า ความถี่คัตออฟ (Cutoff Frequency) หรือ ความถี่ขอบเขต ผ่านไปได้โดยไม่มีการลดทอนของสัญญาณ และถ้าความถี่ของสัญญาณเข้ามีค่าเกินที่กำหนดไว้ สัญญาณออกควรจะมีค่าเป็นศูนย์ แต่ในทางปฏิบัติไม่สามารถจะทำเช่นนั้นได้ เนื่องจากการตอบสนองสัญญาณที่ความถี่ต่างๆ ของอุปกรณ์ประเภทพาสซีฟ จะเป็นแบบค่อยเป็นค่อยไปไม่เปลี่ยนแปลงทันทีทันใด ดังนั้นผลที่ได้จึงเป็นดังรูปที่ 2.13 คือเมื่อสัญญาณมีความถี่สูงขึ้นวงจรจะลดสัญญาณลงไปเรื่อยๆ จนกระทั่งจะลดลงในอัตราคงที่ค่าหนึ่ง

ความถี่คัตออฟ (ω_c) การตอบสนองความถี่ของวงจรขยายนั้นจะตอบสนองได้ดี มีความสัมพันธ์กับค่าเวลาไต่ขึ้น (t_r) และเวลาตก (t_f) ดังสมการดังนี้

$$t_r = t_f = RC \tag{2.1}$$

$$\omega_c = 2\pi f_c = \frac{1}{RC} \tag{2.2}$$

โดยที่ ω_c มีหน่วยเป็น เรเดียนต่อวินาที (rad/s)

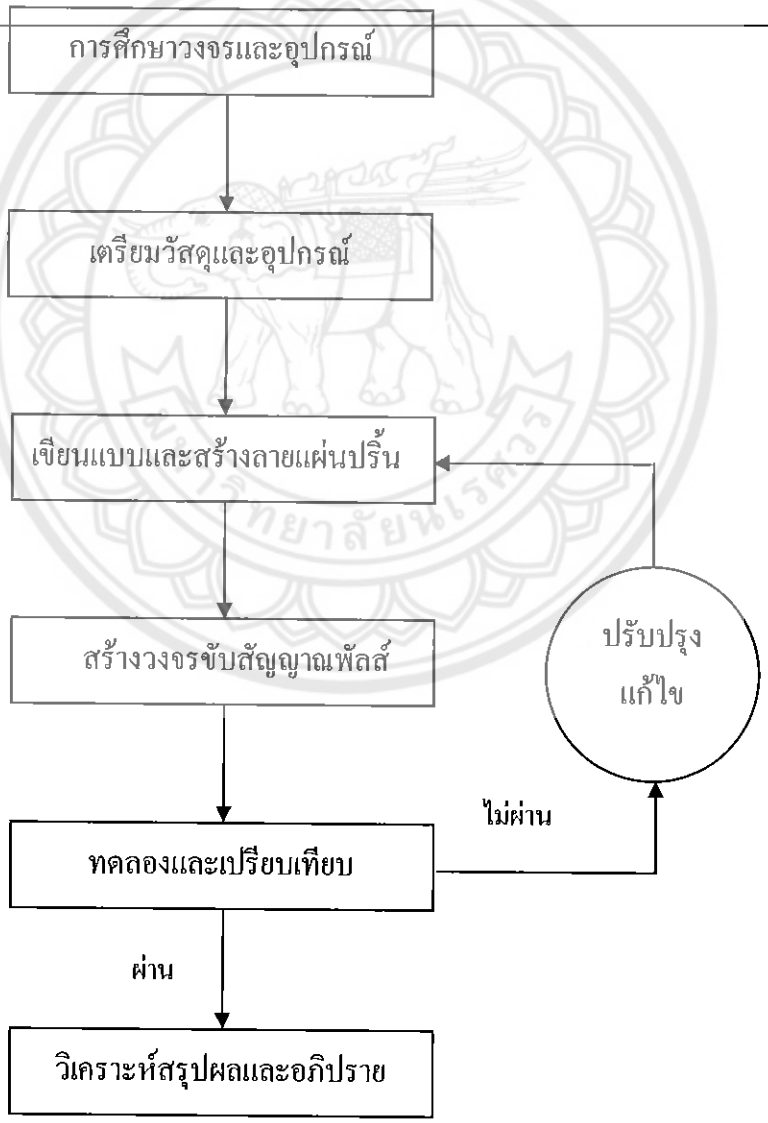
f_c มีหน่วยเป็น เฮิรตซ์ (Hz)

R มีหน่วยเป็น โอห์ม (Ω)

C มีหน่วยเป็น ฟารัด (F)

บทที่ 3
วิธีการดำเนินงาน

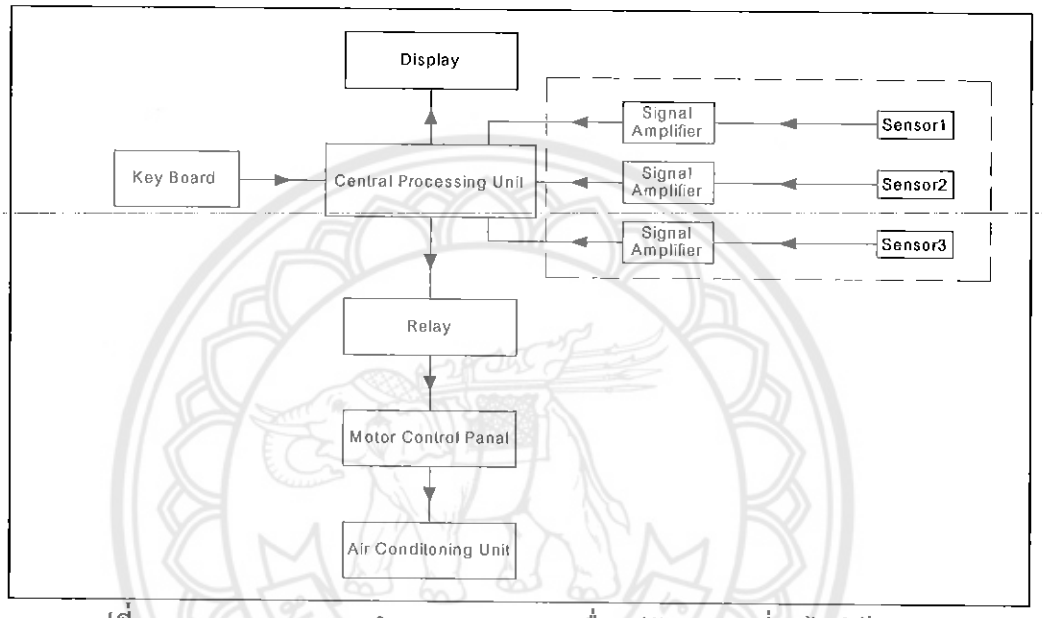
การดำเนินงานในการแก้ไขปัญหาเครื่องควบคุมอุณหภูมิและความชื้นของนายจักรพันธ์ หวาจ้อย และนายปรัชญา จันทร์คามิ ปีการศึกษา 2551 จึงจำเป็นต้องศึกษาหลักการทำงานของอุปกรณ์อิเล็กทรอนิกส์ ที่เกี่ยวข้องกับวงจรขับและรับสัญญาณพัลส์ในบอร์ด AP-105 V1.0 ของบริษัทศิลารีเสิร์จ จำกัด เพื่อสร้างวงขับและรับสัญญาณพัลส์เพิ่มขึ้นอีกจำนวน 2 ชุด โดยการดำเนินงานสามารถเขียนเป็นผังงาน (Flow Chart) ได้ดังต่อไปนี้



รูปที่ 3.1 ขั้นตอนการดำเนินงาน

3.1 ภาพรวมของระบบ

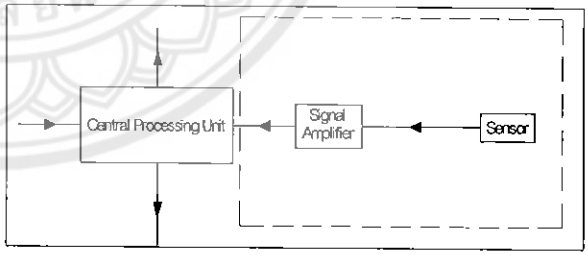
โครงงานนี้เริ่มจาการการศึกษาข้อมูลการทำงานของวงจรขับและรับสัญญาณพัลส์ โดยข้อมูลที่เราได้ศึกษานั้นจะนำไปใช้ประโยชน์ในการส่งสัญญาณพัลส์เพื่อวัดอุณหภูมิและความชื้นสัมพัทธ์ เนื่องจากหัววัดอุณหภูมิและความชื้นกับไมโครคอนโทรลเลอร์อยู่ห่างกันมาก จึงต้องมีการสร้างวงจรขับและรับสัญญาณพัลส์ เพื่อให้สามารถส่งข้อมูลได้ในระยะไกล โดยไม่ทำให้การวัดฯ มีความผิดพลาด



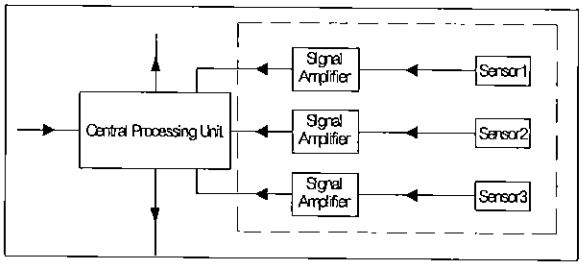
รูปที่ 3.2 Flow Chart การทำงานของระบบเครื่องปรับอากาศที่นำไปใช้งานจริง

แบบเดิม - ระยะความยาวสายสัญญาณระหว่างหัววัดฯ กับไมโครคอนโทรลเลอร์อยู่ใกล้กัน และมีหัววัดภายในเพียง 1 หัวเท่านั้น

แบบใหม่ - เนื่องจากอาคารมีลักษณะกว้าง ทำให้ระยะความยาวสายสัญญาณระหว่างหัววัดฯ กับไมโครคอนโทรลเลอร์อยู่ไกลกันและมีหัววัด 3 หัว คือ ภายในอาคาร ภายนอกอาคาร และคอมเพรสเซอร์ จึงต้องสร้างวงจรขับและรับสัญญาณพัลส์เพิ่มอีก 2 ชุด เพื่อให้สามารถรับและส่งข้อมูลได้ในระยะไกล



ก



ข

รูปที่ 3.3 (ก) แบบเดิม
(ข) แบบใหม่

3.2 เตรียมวัสดุและอุปกรณ์

3.2.1 อุปกรณ์สร้างแผ่นปริ้น

3.2.1.1 แผ่นปริ้น

3.2.1.2 ลายวงจร

3.2.1.3 ไขควงเล็กๆ หรือมีดคัตเตอร์

3.2.1.4 ปากกาเขียนซีดี เบอร์ F

3.2.1.5 กรดกัดปริ้น สำเร็จรูป

3.2.1.6 จานพลาสติก ใส่น้ำกรด และอุปกรณ์ ใช้จับหรือเขี่ยแทนมือ (ระวังอย่าใช้มือหรือผิวหนังสัมผัสน้ำกรดโดยตรง)

3.2.1.7 น้ำเปล่า เอาไว้ล้างแผ่น

3.2.1.8 น้ำยาล้างเล็บ / ทินเนอร์

3.2.1.9 สำลี / กระดาษทิชชู่

3.2.1.10 สว่าน + ดอกสว่าน ขนาดพอดีกับขาอุปกรณ์

3.2.2 อุปกรณ์สร้างวงจรขับและรับสัญญาณพัลส์

3.2.2.1 ทรานซิสเตอร์ เบอร์ BC547	จำนวน 3 ตัว/1 ชุด
3.2.2.2 ทรานซิสเตอร์ เบอร์ BC557	จำนวน 1 ตัว/1 ชุด
3.2.2.3 ตัวต้านทาน (1/2W) 1 kΩ, 10 kΩ, 20 kΩ	อย่างละ 2 ตัว/1 ชุด
3.2.2.4 ตัวต้านทาน (1/2W) 47 kΩ	จำนวน 1 ตัว/1 ชุด
3.2.2.5 ตัวเก็บประจุ เบอร์ 104, 33p	อย่างละ 1 ตัว/1 ชุด
3.2.2.6 แผ่นปริ้นที่ทำลายวงจรแล้ว	จำนวน 1 แผ่น/1 ชุด
3.2.2.7 อุปกรณ์บัดกรี	จำนวน 1 ชุด

3.2.3 อุปกรณ์การทดลอง

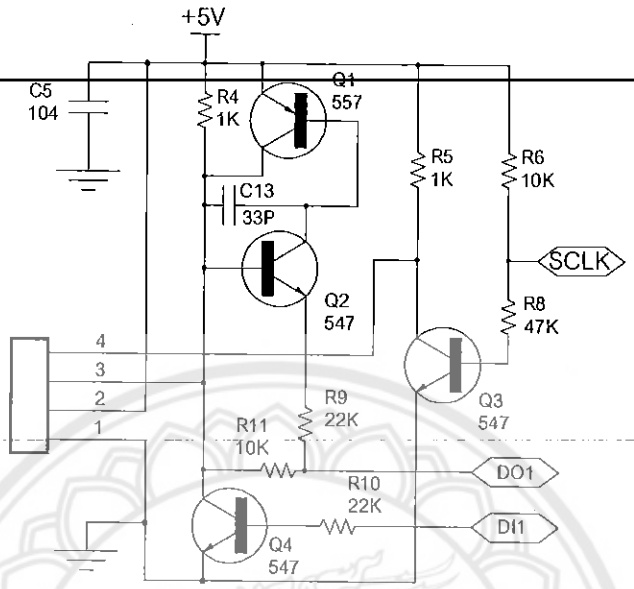
3.2.3.1 บอร์ดวัดอุณหภูมิและความชื้น AP-105 V1.0	จำนวน 1 เครื่อง
3.2.3.2 ออสซิลโลสโคป	จำนวน 1 เครื่อง
3.2.3.3 ชุดวงจรขับและรับสัญญาณพัลส์	จำนวน 2 ชุด
3.2.3.4 สายสัญญาณและหัววัดอุณหภูมิและความชื้น SHT15	จำนวน 3 หัว

3.3 ดำเนินการสร้าง

3.3.1 เขียนแบบและสร้างแผ่นปริ้น

3.3.1.1 เข้าโปรแกรม PCB Wizard เปิดหน้าต่าง Circuit Symbol Gallery เพื่อเลือกสัญลักษณ์ของอุปกรณ์ต่างๆ นำออกมาวางให้ครบก่อน โดยการคลิกเลือกแล้วลากมาวาง แล้วกำหนดเบอร์และค่าอุปกรณ์ต่างๆ ตามที่ต้องการ

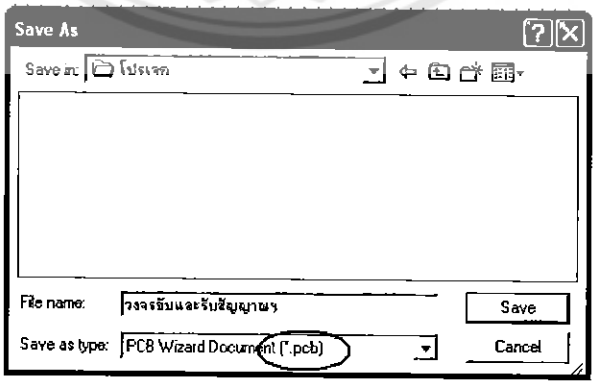
3.3.1.2 หลังจากนั้นแล้วค่อยๆ จัดวางอุปกรณ์ต่างๆที่เรียงมาตามรูปที่ 3.4 หากต้องการหมุนตัวอุปกรณ์ให้ใช้ Ctrl+L สำหรับหมุนซ้ายทีละ 90 องศา



รูปที่ 3.4 วงจรขับและรับสัญญาณพัลส์ในบอร์ด AP-105 V1.0 ของบริษัทศิลารีเสิร์จ จำกัด

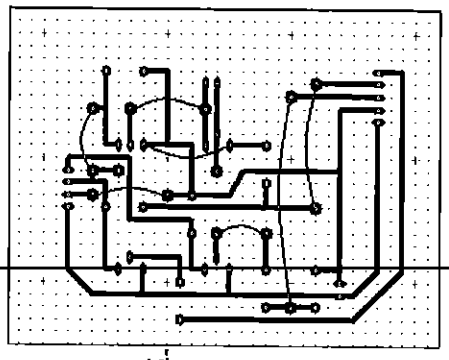
3.3.1.3 ทำการลากเส้นเชื่อมต่อขาอุปกรณ์ โดยเลือกที่ขาต้นทางพร้อมทั้งคลิกเมาท์ค้างไว้แล้วลากเป็นเส้นไปจนถึงขาอุปกรณ์ปลายทางซึ่งจะมีจุด(●) แล้ววางเมาท์ โปรแกรมจะเชื่อมต่อกันเอง ที่สำคัญเส้นวงจรที่เขียนขึ้น จะลากผ่านตัวอุปกรณ์ไม่ได้

3.3.1.4 เมื่อวาดวงจรเสร็จเรียบร้อยแล้วจะได้ดังรูปที่ 3.4 ให้บันทึกเก็บเป็นไฟล์ไว้รูปแบบ File.pcb ดังรูปที่ 3.5



รูปที่ 3.5 การ Save เป็น File.pcb

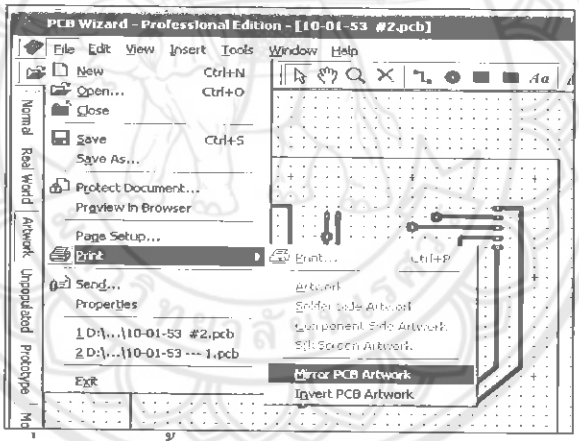
3.3.1.5 ทำการ Convert ลายวงจร โดยที่โปรแกรมจะทำการ Convert เอง จะได้ลายวงจรตามรูปที่ 3.6



รูปที่ 3.6 ลายวงจร

3.3.1.6 หลังจาก Convert เรียบร้อยแล้ว ผลที่ออกมาต้องได้ 100% ถ้าไม่ได้ก็ต้องต่ออุปกรณ์ใหม่เพื่อให้ได้ 100% ให้ไปที่ Tools > Auto Route > Route All Net โปรแกรมก็จะทำการ Convert ให้อีก 1 รอบ

3.3.1.7 พอได้ลายทองแดงก็ปริ้นใส่แผ่นใสออกมา โดยเลือก File > Print > Mirror PCB Artwork ตามรูปที่ 3.7



รูปที่ 3.7 การปริ้นลายวงจรแบบ Mirror PCB Artwork

3.3.1.8 นำแผ่นรูปวงจรที่ปริ้นได้เท่าของจริงมาตรวจสอบ

รูปที่ 3.8 ลายวงจรที่ปริ้นแบบ Mirror PCB Artwork

3.3.1.9 นำลายวงจร(ลายวงจรที่ปริ้นแบบ mirror) มาวางทาบลงบนแผ่นทองแดง (แผ่นทองแดงอยู่ด้านล่างของกระดาษ) เพราะถ้าวางรูปวงจรหงายจะทำให้เมื่อเสียบอุปกรณ์ จะกลับด้านกัน ข้อนี้โปรดระวังให้มากๆ เพราะหากวางผิดด้านวงจรจะไม่ทำงาน และอุปกรณ์อาจเสียหายได้

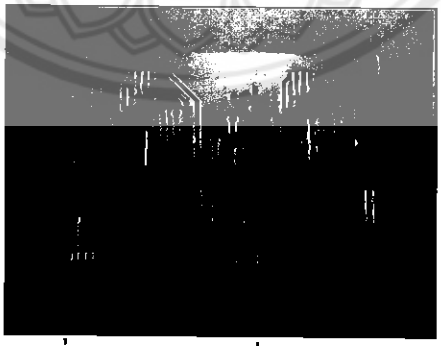
3.3.1.10 นำไขควงเล็กๆ หรือมีดคัตเตอร์กดลงไปตามจุดที่เห็นขาอุปกรณ์ เพื่อให้เกิดรอยบนแผ่นปริ้นด้านล่าง แล้วเอาวงจรออก จะเห็นว่ามีรอยเป็นจุดตามที่กด บนแผ่นทองแดง

3.3.1.12 นำปากกาเขียนซีดี เบอร์ F จุดลงบนแผ่นตามรอยที่ปรากฏ ทำเป็นวงกลมขนาดเล็กก่อน เพราะอาจมีการเขียนลายเฉียดวงกลม จะได้มีพื้นที่ลากผ่านไปได้

3.3.1.13 เขียนลายวงจรช้าๆ เพื่อให้น้ำหมึกลงมากๆ อย่าลากเร็วเพราะเวลาเอาน้ำกรดกัดอาจทะลุ ดังรูปที่ 3.9

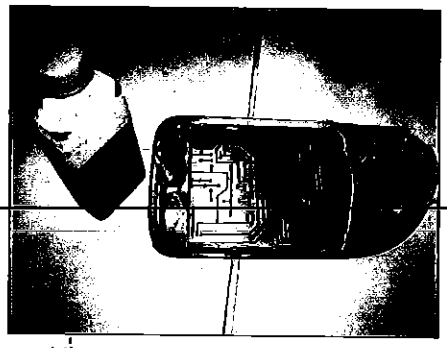
รูปที่ 3.9 เขียนลายวงจรแผ่นปริ้น

3.3.1.14 ขยายจุดตรงขาอุปกรณ์ให้ใหญ่ขึ้น จะเห็นว่าตรงไหนลายวงจรชิดมากๆ ก็สามารถขยายวงกลม ไปทางอื่นได้โดยสะดวก



รูปที่ 3.10 ลายวงจรที่เขียนเสร็จแล้ว

3.3.1.15 เอาลงกัดในกรดแก่หรือเฟอริกคลอไรด์ อย่าผสมเข้มข้นมากไป ต้องใจเย็นๆ และที่สำคัญ อย่าคนน้ำกรดเพื่อเร็งความเร็วเพราะจะทำให้หมึกหลุดได้ ดังรูปที่ 3.12 เวลาในขั้นตอนนี้ประมาณ 1 ชั่วโมง



รูปที่ 3.11 แช่ลายวงจรลงในกรดแก่

ข้อควรระวัง : อย่าให้หมึกสัมผัสผิวหนังเด็ดขาด

3.3.1.16 เมื่อทองแดงโดนกัดจนหมดแล้ว ให้นำมาล้างด้วยน้ำเปล่า (ระวังอย่าให้โดนกรด) แล้วใช้สำลีหรือทิชชูชุบน้ำยาล้างเล็บหรือทินเนอร์แล้ว เช็ดหมึกออก

3.3.1.17 ตรวจสอบลายวงจร อาจมีการขาดบ้าง แล้วใช้มิเตอร์วัดว่าแต่ละจุดเชื่อมกันดีหรือไม่ ถ้าวงจรขาด อาจใช้สายไฟเชื่อมภายหลัง แต่ถ้าช๊อทให้ใช้คัตเตอร์กรีดวงจรออก

3.3.1.18 เจาะรูที่แผงวงจร

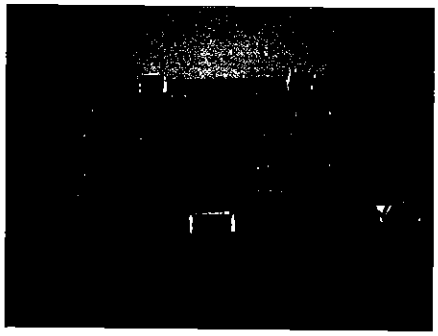
3.3.2 สร้างวงจรขับและรับสัญญาณพัลส์

3.3.2.1 เตรียมอุปกรณ์ และแผ่นปริ้นที่เราสร้างขึ้นตามหัวข้อ 3.3.1

3.3.2.2 บัดกรีอุปกรณ์ต่างๆ ลงในแผ่นปริ้นที่เขียนลายวงจรไว้

3.3.2.3 เริ่มบัดกรี โดยใช้หัวแร้งจี้เข้าไปที่แผ่นทองแดงก่อนแล้วจึงค่อยๆ นำตะกั่วใส่เข้าไป ใช้เวลาประมาณ 3-5 วินาที เพื่อไม่ให้แผ่นทองแดงเสียหาย

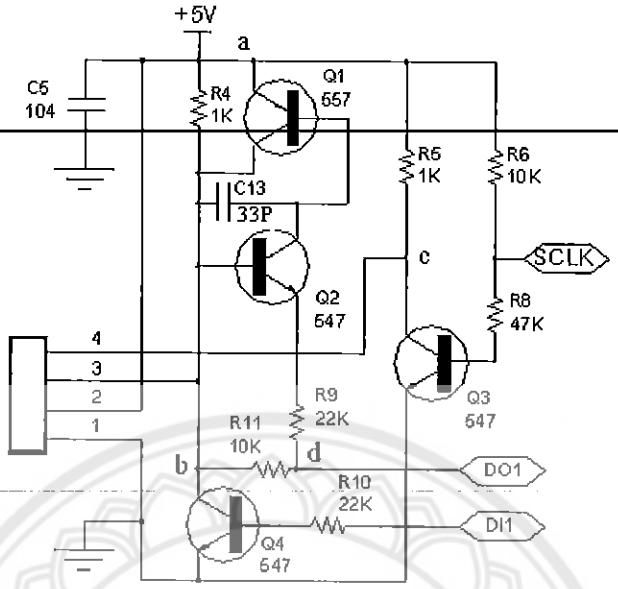
3.3.2.4 เมื่อบัดกรีเสร็จแล้วก็ตัดขาอุปกรณ์ออก และตรวจสอบว่าวงจรช๊อตกันหรือไม่ ถ้าช๊อตเนื่องจากการบัดกรีก็ใช้หัวแร้งจี้ตรงที่ติดกันแล้วนำที่ดูดตะกั่วมาดูดออก แล้วบัดกรีใหม่



รูปที่ 3.12 วงจรขับและรับสัญญาณพัลส์ที่สร้างขึ้น

1575 8439
2/ร.
ว126 ส
2552

3.4 อธิบายการรับ – ส่งข้อมูลของวงจรขับและรับสัญญาณพัลส์



รูปที่ 3.13 วงจรขับและรับสัญญาณพัลส์.

3.4.1 การส่งข้อมูล (จากไมโครคอนโทรลเลอร์ไปหัววัดฯ)

3.4.1.1 การส่ง DATA บิต 0 เลือกป้อน DO1 เป็นบิต 0 (แรงดัน 0 โวลต์) ทำให้ Q2 และ Q1 ยังคงสถานะ OFF เช่นเดิม เพราะไม่มีกระแสไหลเข้าขาเบส ทำให้ค่าความต้านทานรวมระหว่างจุด a ถึงจุด b เพิ่มสูงขึ้น ส่งผลให้แรงดันที่ตกคร่อมช่วงนั้นมีค่ามาก และเลือกป้อน DI1 เป็นบิต 1 (แรงดัน 5 โวลต์) จะทำให้ Q4 เปลี่ยนสถานะจาก OFF เป็น ON กระแสจะไหลจากขาเบสไปยังขาอีมิตเตอร์ลงกราวนด์ จึงส่งผลให้กระแสที่ไหลจากแหล่งจ่ายไฟฟ้ากระแสตรง 5V ไหลผ่าน Q4 ลงกราวนด์อย่างรวดเร็ว สาย DATA (ขา 3) ที่หัววัดฯ ก็จะรับข้อมูลเป็นบิต 0

3.4.1.2 การส่ง DATA บิต 1 เลือกป้อน DO1 เป็นบิต 1 (แรงดัน 5 โวลต์) ที่จุด a-b ก็จะมีแรงดันตกคร่อมอยู่ (0<V<5) และเลือกป้อน DI1 เป็นบิต 0 Q4 ก็ยังคงสถานะเป็น OFF เช่นเดิม ไม่มีกระแสไหลลงกราวนด์ สาย DATA (ขา 3) ที่หัววัดฯ ก็จะรับข้อมูลเป็นบิต 1

3.4.1.3 การส่ง SCLK บิต 0 ป้อน SCLK เป็นบิต 1 (5 โวลต์) Q3 ก็จะเปลี่ยนสถานะจาก OFF เป็น ON แรงดันที่ไหลเข้าขาเบสและแรงดันที่ไหลมาจากแหล่งจ่าย ก็จะไหลผ่านQ3 ลงสู่กราวนด์ สาย SCLK (ขา4) ที่หัววัดฯ จะรับข้อมูลเป็นบิต 0

3.4.1.4 การส่ง SCLK บิต 1 ป้อน SCLK เป็นบิต 0 (0 โวลต์) Q3 ก็ยังคงสถานะ OFF ไว้ เนื่องจากไม่มีกระแสไหลเข้าขาเบส และกระแสที่ไหลมาจากแหล่งจ่ายก็จะไหลมายังจุด c ทำให้มีแรงดัน (0<V<5) สาย SCLK (ขา4) ที่หัววัดฯ จะรับข้อมูลเป็นบิต 1

3.4.2 การรับข้อมูล (จากหัววัดฯ ไปไมโครคอนโทรลเลอร์)

3.4.2.1 การรับ DATA บิต 0 จากสาย DATA ของ Sensor รับข้อมูลเป็นบิต 0 แรงดันที่จุด b เป็น 0 โวลต์ ทำให้ Q4 คงสถานะเป็น OFF ไม่มีแรงดันที่ขาเบสของ Q2 ทำให้ Q2 และ Q1 มีสถานะเป็น OFF ที่จุด b มีแรงดันเป็น 0 สาย DATA (ขา DI1) จะรับข้อมูลเป็นบิต 0

3.4.2.2 การรับ DATA บิต 1 จากสาย DATA ของ Sensor รับข้อมูลเป็นบิต 1 แรงดันที่จุด b เป็น 5 โวลต์ เพราะ Q4 ยังคงสถานะเป็น OFF อยู่ จึงมีกระแสไหลเข้าขาเบสของ Q2 ทำให้ Q2 เปลี่ยนสถานะจาก OFF เป็น ON ทำให้กระแสที่ขาคอลเลคเตอร์ไหลเข้าขาเบสของ Q1 ทำให้ Q1 เปลี่ยนสถานะเป็น ON ค่าความต้านทานช่วง a-b มีค่าต่ำ แรงดันตกคร่อมจึงมีน้อย ทำให้แรงดันที่จุด b ถูกดึงขึ้นเป็น 5V ด้วยความรวดเร็ว สาย DATA (ขา DI1) จะรับข้อมูลเป็นบิต 1

บทที่ 4
การทดลองและวิเคราะห์ผลการทดลอง

4.1 การทดลองตอนที่ 1 : แบบไม่ผ่านวงจรขับและรับสัญญาณพัลส์

4.1.1 วัตถุประสงค์การทดลองตอนที่ 1

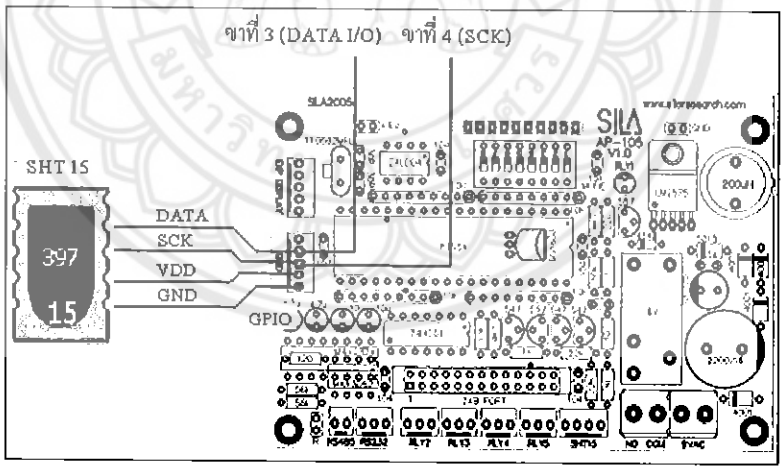
4.1.1.1 เพื่อศึกษาและเปรียบเทียบรูปคลื่นสัญญาณพัลส์ที่ระยะความยาวสายสัญญาณในแต่ละช่วง

4.1.1.2 เพื่อหาปัจจัยที่มีผลต่อรูปคลื่นสัญญาณพัลส์ที่เกิดขึ้น

4.1.2 ขั้นตอนการทดลอง

4.1.2.1 นำโปรแกรมวัดอุณหภูมิและความชื้นโดยตรงที่เขียนไว้ โหลดลงไมโครคอนโทรลเลอร์ ของบอร์ด AP-105 V1.0 (โดยกำหนดให้ DATA เป็นพอร์ต P2.1 และ SCK เป็นพอร์ต P2.0)

4.1.2.2 นำหัววัดอุณหภูมิและความชื้น SHT15 ที่เชื่อมต่อกับสายสัญญาณยาว 20 เซนติเมตร ต่อเข้าที่พอร์ต GPIO ของบอร์ด AP-105 ตามรูปที่ 4.1



รูปที่ 4.1 การเชื่อมต่อสายสัญญาณของหัววัดฯ กับบอร์ด AP-105

4.1.2.3 นำสายโพรบของออสซิลโลสโคปวัดสัญญาณพัลส์ โดยที่ CH1 จับที่ขา 3 (DATA) และCH2 จับที่ขา 4 (SCK) ของพอร์ต GPIO ตามรูปที่ 4.1

4.1.2.4 จ่ายไฟให้กับบอร์ด AP-105

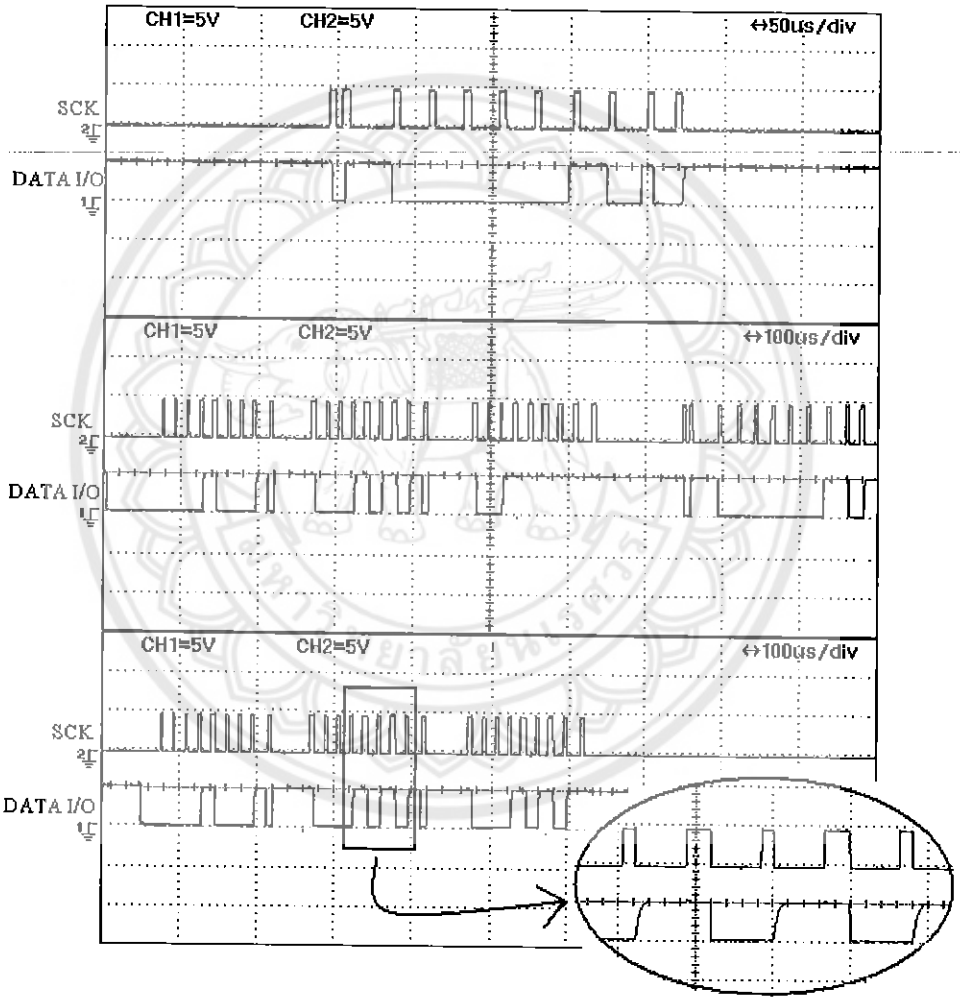
4.1.2.5 บันทึกรูปคลื่นสัญญาณพัลส์ที่ได้ลงใน Floppy disk

4.1.2.6 ทำซ้ำข้อ 4.1.2.2 - 4.1.2.5 โดยเปลี่ยนความยาวของสายสัญญาณพร้อมหัววัดอุณหภูมิและความชื้น SHT15 เป็น 8 เมตรและ 15 เมตรตามลำดับ

หมายเหตุ การแสดงผลการวัดค่าอุณหภูมิและค่าความชื้นสัมพัทธ์จะแสดงออกทางหน้าจอคอมพิวเตอร์ โดยใช้ Program HyperTerminal

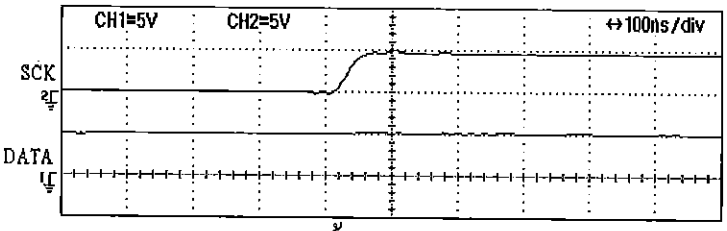
4.1.3 ผลการทดลองตอนที่ 1

4.1.3.1 สายสัญญาณยาว 20 เซนติเมตร วัดค่าอุณหภูมิได้ 25.1 องศาเซลเซียส วัดค่าความชื้นสัมพัทธ์ได้ 78 เปอร์เซ็นต์

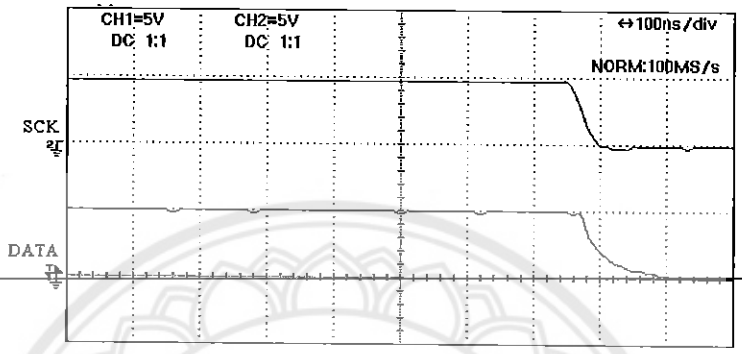


รูปที่ 4.2 กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 20 เซนติเมตร

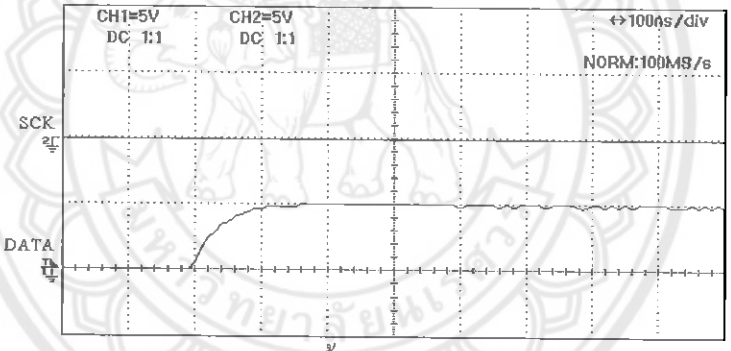
อธิบายรูปที่ 4.2 ภาพรวมของสัญญาณ โดยรวมแล้วสัญญาณ มีลักษณะเป็นแท่งสี่เหลี่ยม ส่วนภาพขยายด้านล่าง จะเห็นว่า สัญญาณ SCK มีค่า 5V ซึ่งเป็นไปตามความต้องการของ Chip SHT15 ส่วนสัญญาณ DATA ใช้เวลาไต่ขึ้น – ลงระหว่าง 0 – 5V น้อย จึงทำให้สามารถส่งข้อมูลระหว่าง SHT15 กับไมโครคอนโทรลเลอร์ ได้ถูกต้อง



(ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK



(ข) ช่วงเวลาตกของสัญญาณ SCK และ DATA



(ค) ช่วงเวลาไต่ขึ้นของสัญญาณ DATA

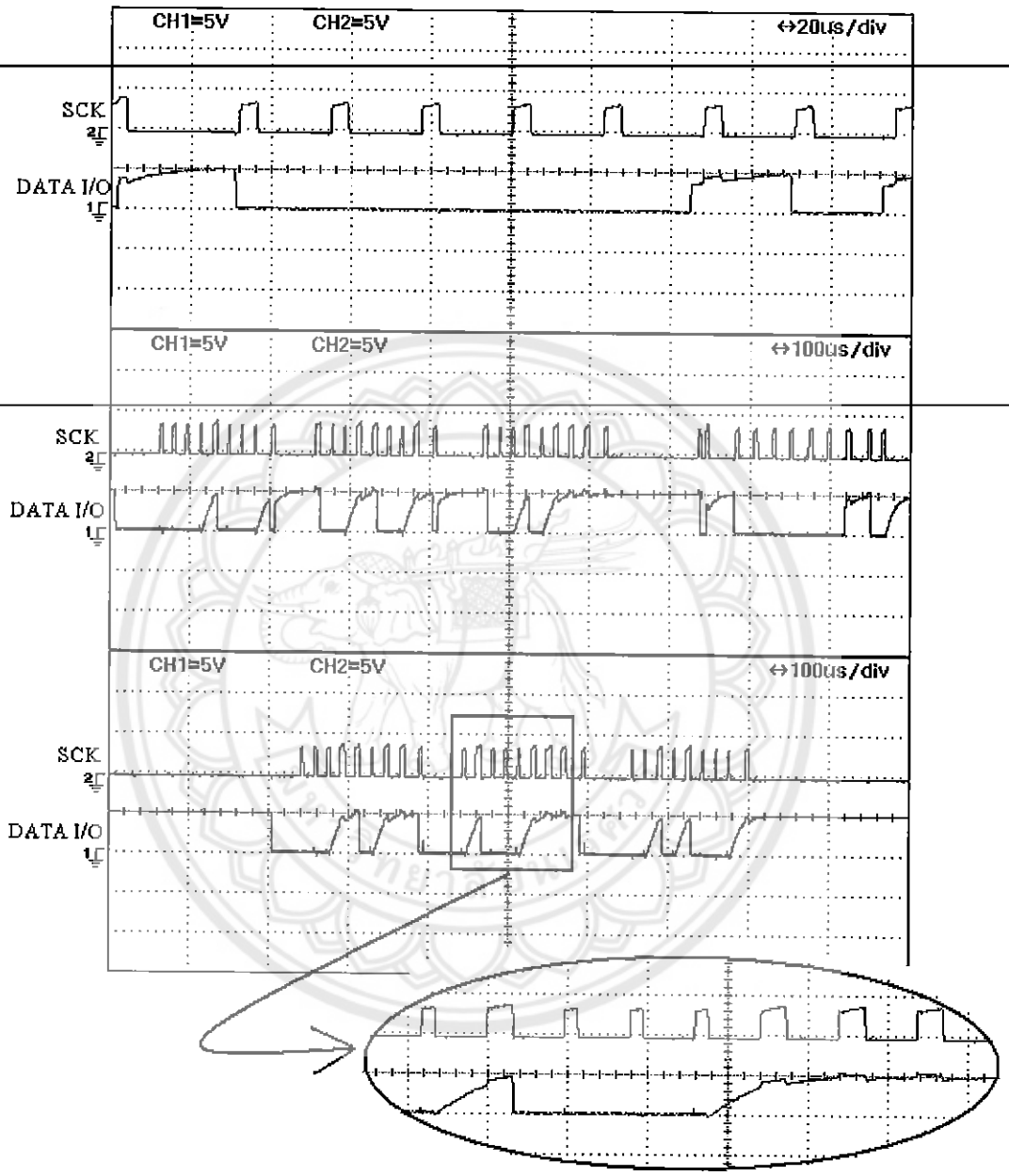
รูปที่ 4.3 กราฟแสดงค่า t_r และ t_f ไม่ผ่านวงจรฯ ที่ความยาวสาย 20 เซนติเมตร

อธิบายรูปที่ 4.3 (ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK จาก 0 V ขึ้นไป 4.8V โดยใช้เวลาไต่ขึ้นประมาณ 100ns ส่วนขา DATA มีแรงดัน 5V ลักษณะของสัญญาณมี noise เล็กน้อย

รูปที่ 4.3 (ข) ช่วงเวลาตกของสัญญาณ SCK จาก 4.8V ลงมา 0V โดยใช้เวลาตกของสัญญาณประมาณ 120ns ส่วนขา DATA มีช่วงเวลาตกของสัญญาณจาก 5V ลงมา 0V ใช้เวลาประมาณ 160ns ลักษณะของสัญญาณมี noise เล็กน้อย

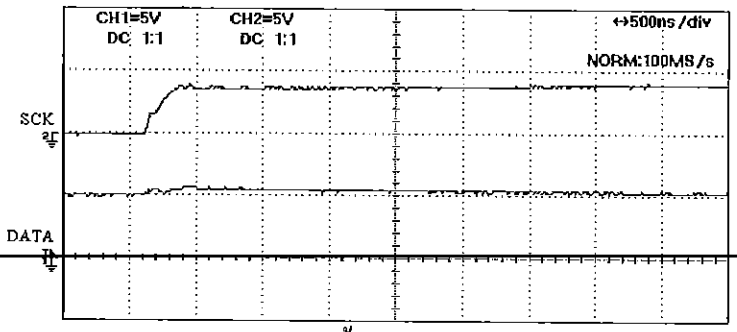
รูปที่ 4.3 (ค) จากรูปขา SCK มีแรงดัน 0V ส่วนขา DATA เป็นช่วงเวลาไต่ขึ้นของสัญญาณจาก 0 V ขึ้นไป 5V โดยใช้เวลาไต่ขึ้นประมาณ 180ns ลักษณะของสัญญาณขา SCK เรียบแต่ขา DATA มี noise เล็กน้อย

4.1.3.2 สายสัญญาณยาว 8 เมตร วัดค่าอุณหภูมิได้ 32 องศาเซลเซียส วัดค่าความชื้นสัมพัทธ์ได้ 63 เปอร์เซ็นต์

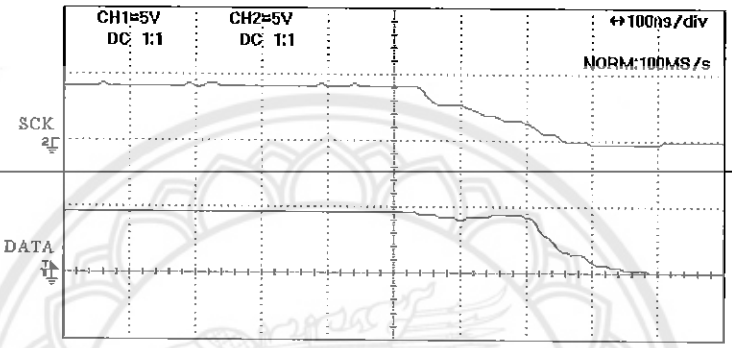


รูปที่ 4.4 กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 10 เมตร

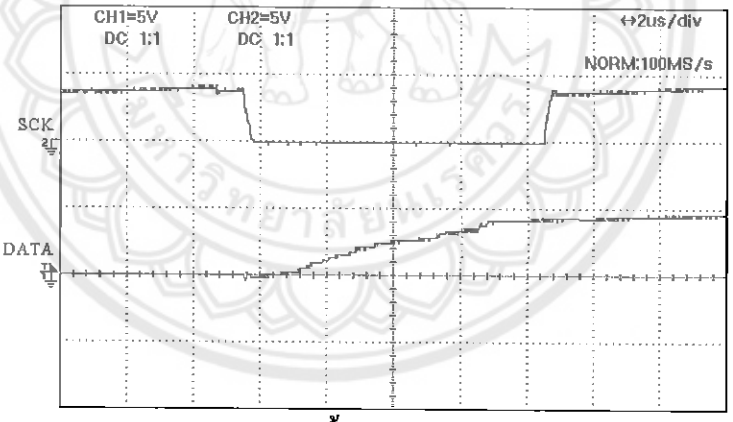
อธิบายรูปที่ 4.4 ภาพรวมของสัญญาณ โดยรวมแล้วสัญญาณ มีลักษณะเป็นสไปค์ขึ้น ส่วนภาพบนสุดขยายเพื่อให้เห็นความเพี้ยนของสัญญาณ SCK ที่มีค่าน้อยกว่า 80% ของแรงดันแหล่งจ่าย ซึ่งไม่เป็นไปตามความต้องการของ Chip SHT15 ส่วนสัญญาณภาพขยายด้านล่างจะพบว่า สัญญาณ DATA ขึ้นลงระหว่าง 0 – 5V แต่ใช้เวลานาน จึงทำให้การส่งข้อมูลระหว่าง SHT15 กับไมโครคอนโทรลเลอร์ คลาดเคลื่อนไปจากเดิมที่วัดที่ความยาวสายสัญญาณ 20 เซนติเมตร



(ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK



(ข) ช่วงเวลาตกของสัญญาณ SCK และ DATA



(ค) ช่วงเวลาไต่ขึ้นของสัญญาณ DATA

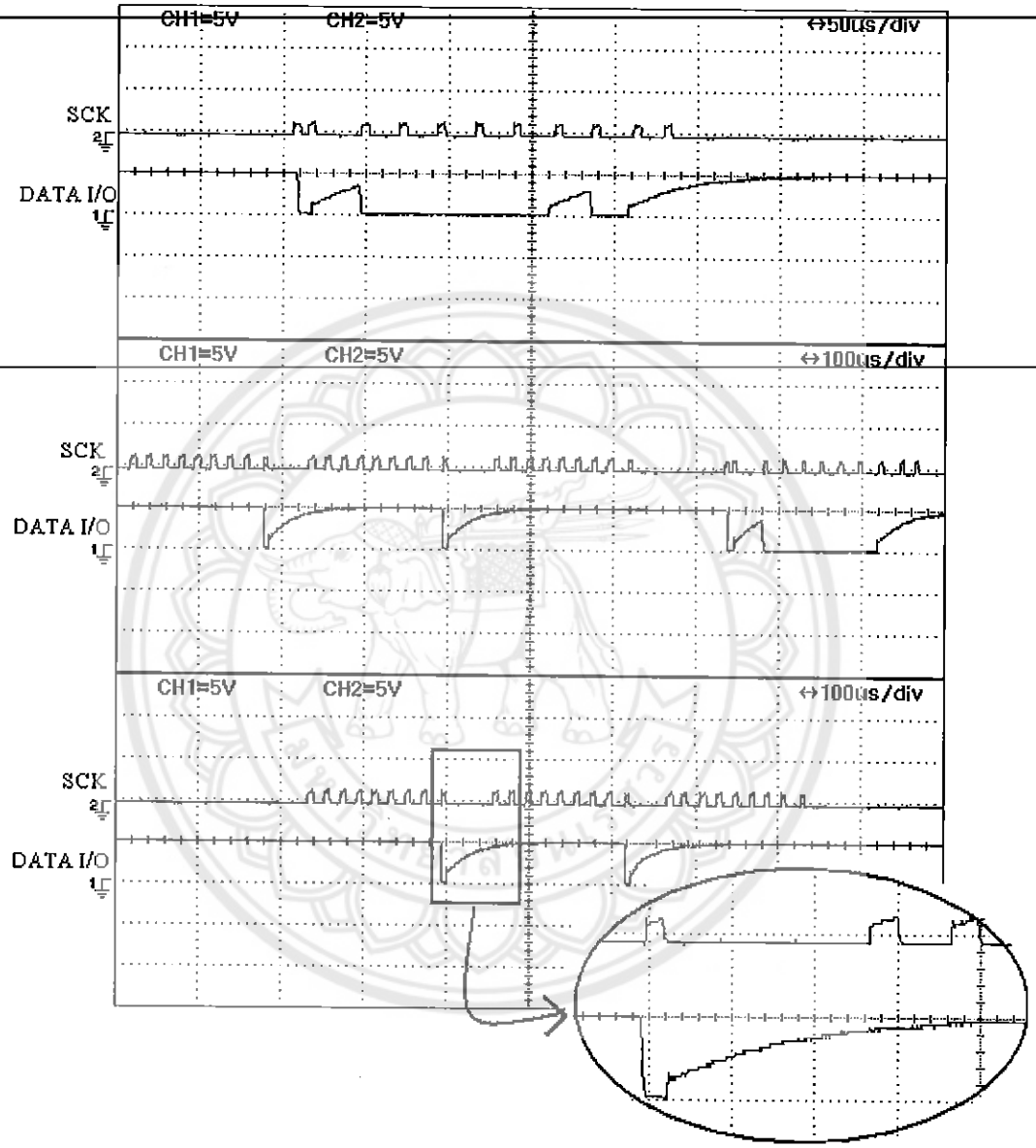
รูปที่ 4.5 กราฟแสดงค่า t_f และ t_r ไม่ผ่านวงจรฯ ที่ความยาวสาย 8 เมตร

อธิบายรูปที่ 4.5 (ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK จาก 0 V ขึ้นไป 4V โดยใช้เวลาไต่ขึ้นประมาณ 400ns ส่วนขา DATA มีแรงดันประมาณ 5V ลักษณะของสัญญาณมี noise ค่อนข้างมาก

รูปที่ 4.5 (ข) ช่วงเวลาตกของสัญญาณ SCK จาก 4V ลงมา 0V โดยใช้เวลาตกของสัญญาณประมาณ 400ns ส่วนขา DATA มีช่วงเวลาตกของสัญญาณจาก 4.8V ลงมา 0V ใช้เวลาประมาณ 380ns ลักษณะสัญญาณมี noise ค่อนข้างมาก

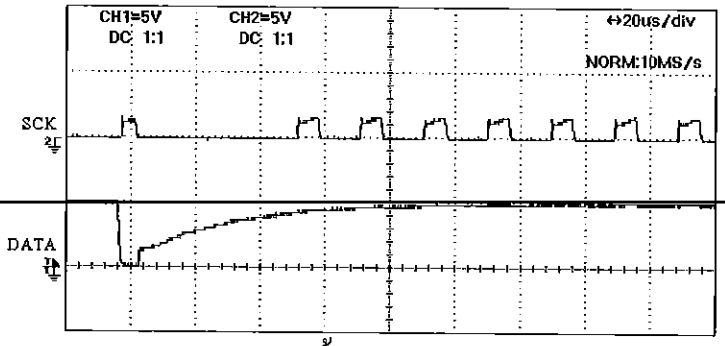
รูปที่ 4.5 (ค) ขา DATA มีช่วงเวลาไต่ขึ้นของสัญญาณจาก 0V ขึ้นไป 5V ใช้เวลาประมาณ 6,000 ns ลักษณะสัญญาณมี noise ค่อนข้างมาก

4.1.3.3 สายสัญญาณยาว 15 เมตร ไม่สามารถวัดค่าความชื้นสัมพัทธ์และค่าอุณหภูมิได้

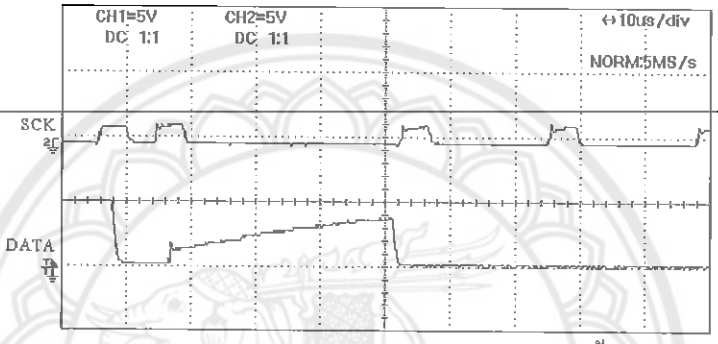


รูปที่ 4.6 กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 15 เมตร

อธิบายรูปที่ 4.6 ภาพรวมของสัญญาณ จากภาพขยายด้านล่างจะเห็นว่า ช่วงเวลาไต่ขึ้นของสัญญาณ DATA ต้องใช้เวลานานมาก ส่วนสัญญาณ SCK โดยรวมแล้ว มีค่าเพียง 1V เท่านั้น ไม่เป็นไปตามความต้องการของ Chip SHT15 จึงทำให้ไมโครคอนโทรลเลอร์ ไม่สามารถส่งสัญญาณ SCK ให้กับ SHT15 ได้ เนื่องจากสภาพสายสัญญาณที่มีค่า R และ C สูง ส่งผลให้ไม่สามารถวัดอุณหภูมิและความชื้นได้



(ก) ช่วงเวลาไต่ขึ้นของสัญญาณ DATA



ข) ช่วงเวลาตกของสัญญาณ SCK และ DATA และช่วงเวลาไต่ขึ้นของสัญญาณ SCK

รูปที่ 4.7 กราฟแสดงค่า t_r และ t_f ไม่ผ่านวงจรฯ ที่ความยาวสาย 15 เมตร

อธิบายรูปที่ 4.7 (ก) ช่วงสัญญาณ SCK ระหว่าง 0 ถึง 1.5V ส่วนขา DATA มีช่วงเวลาไต่ขึ้นของสัญญาณจาก 0V ขึ้นไป 5V ใช้เวลาประมาณ 180,000ns ลักษณะสัญญาณมี noise ค่อนข้างมาก

รูปที่ 4.7 (ข) ช่วงสัญญาณ SCK ระหว่าง -0.5 ถึง 1V ส่วนขา DATA มีช่วงเวลาไต่ขึ้นของสัญญาณจาก 0V ขึ้นไป 3.8V ใช้เวลาประมาณ 40μs แต่ช่วงเวลาตกของสัญญาณจาก 3.8V ลงมา 0V ใช้เวลาประมาณ 4μs ลักษณะสัญญาณมี noise ค่อนข้างมาก

ตารางที่ 4.1 ค่าอุณหภูมิและความชื้นสัมพัทธ์ ตอนที่ 1

ระยะความยาวสายสัญญาณ (m.)	อุณหภูมิ (°C)	ความชื้นสัมพัทธ์ (%)
0.20	25.1	78.0
8	24.5	76.8
15	ไม่สามารถวัดได้	ไม่สามารถวัดได้

ตารางที่ 4.2 ช่วงเวลาไต่ขึ้น – เวลาตกของสัญญาณ ตอนที่ 1

ค่าช่วงเวลา	ย่านที่ต้องการ (ns)	ค่าช่วงเวลาต่างๆ ที่ความยาวสายสัญญาณฯ (ns)		
		20 cm.	**8 m.**	**15 m.**
T_{R_SCK}	$T_R > 1$	45	3400	-
T_{F_SCK}	$T_F > 1$	30	220	-
T_{FO_DATA}	$1 < T_{FO} < 200$	74	160	500
T_{RO_DATA}	$T_{RO} < 435.3$	70	-	-
	$T_{RO} < 17,333$		6,000	-
	$T_{RO} < 32,500$			40,000

หมายเหตุ
- T_{R_SCK} คือ ช่วงเวลาไต่ขึ้นของสัญญาณ SCK
- T_{F_SCK} คือ ช่วงเวลาตกของสัญญาณ SCK
- T_{FO_DATA} คือ ช่วงเวลาตกของสัญญาณ DATA
- T_{RO_DATA} คือ ช่วงเวลาไต่ขึ้นของสัญญาณ DATA
- t_r คือ ช่วงเวลาไต่ขึ้น
- t_f คือ ช่วงเวลาตก

4.1.4 วิเคราะห์ผลการทดลองตอนที่ 1

จากผลการทดลองตอนที่ 1 พบว่า ที่ความยาวสายสัญญาณ 0.2, 8 เมตร สามารถอ่านค่าความชื้นสัมพัทธ์และค่าอุณหภูมิได้ใกล้เคียงกัน ส่วนสายสัญญาณยาว 15 เมตร ไม่สามารถอ่านค่าความชื้นสัมพัทธ์และค่าอุณหภูมิได้เลย และเมื่อวิเคราะห์จากตารางที่ 4.2 พบว่าที่ความยาวสายสัญญาณ 0.2, 8 เมตร T_{RO} และ T_{FO} อยู่ในย่านที่ต้องการ ส่วนที่สายสัญญาณยาว 15 เมตร T_{RO} และ T_{FO} มีค่ามากเกินย่านที่ต้องการของหัววัด SHT15 เนื่องจากลักษณะของสายสัญญาณ ซึ่งทำตัวเป็นสายส่งสัญญาณ ภายในสายสัญญาณจะมีค่า RC ทำตัวเป็น ตัวกรองความถี่ต่ำ (Low pass filter) เมื่อสายสัญญาณยาวเกินไปจะทำให้ความถี่สูงผ่านสายสัญญาณวัดไปได้น้อย

จากสมการที่ 2.1 และ 2.2 แสดงว่า ความถี่คัตออฟ (ω_c) ขึ้นอยู่กับ t_r และ t_f ดังนั้น ถ้า t_r มาก ω_c ก็ยิ่งน้อย จึงเกิดความคลาดเคลื่อนของความถี่ Output มากขึ้น ซึ่งธรรมดาแล้วความถี่ Output กับ Input ต้องเท่าเดิม แสดงว่า ถ้าเราเพิ่มความยาวสายส่งสัญญาณก็เท่ากับเพิ่มค่า RC จะทำให้ t_r และ t_f เพิ่มขึ้น

4.2 การทดลองตอนที่ 2 : แบบผ่านวงจรขับและรับสัญญาณพัลส์

4.2.1 วัตถุประสงค์การทดลองตอนที่ 2

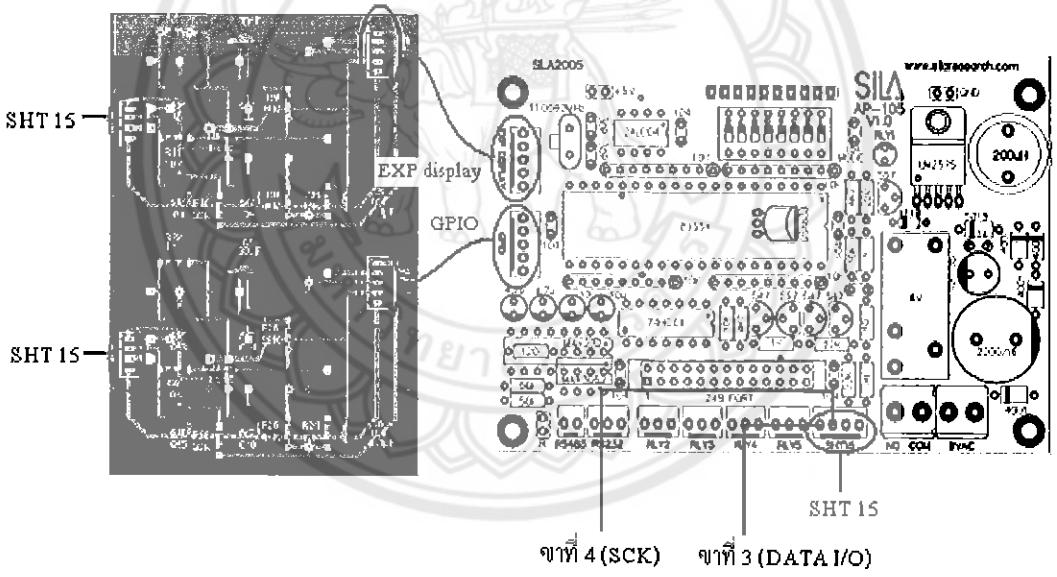
4.2.1.1 เพื่อศึกษาและเปรียบเทียบรูปคลื่นสัญญาณพัลส์ที่ระยะความยาวสายสัญญาณในแต่ละช่วง

4.2.1.2 เพื่อเปรียบเทียบคุณภาพรูปคลื่นสัญญาณพัลส์ของวงจรฯ ในบอร์ด AP-105 กับวงจรที่สร้างขึ้นมาใหม่อีก 2 ชุด

4.2.2 ขั้นตอนการทดลองตอนที่ 2.1 : ผ่านวงจรฯ เดิมของบอร์ด AP-105

4.2.2.1 นำโปรแกรมที่วัดอุณหภูมิและความชื้นสัมพัทธ์ โดยผ่านวงจรขับและรับสัญญาณพัลส์ที่เขียนไว้ โหลดลงบอร์ด AP-105 (โดยกำหนดให้ DO1 เป็นพอร์ต P0.1, DI1 เป็นพอร์ต P0.2 และ SCK เป็นพอร์ต P2.3)

4.2.2.2 นำหัววัดอุณหภูมิและความชื้น SHT15 ที่เชื่อมต่อกับสายสัญญาณยาว 20 เซนติเมตร ต่อเข้ากับพอร์ต SHT15



รูปที่ 4.8 การเชื่อมต่อสายสัญญาณของหัววัดฯ กับบอร์ด AP-105

4.2.2.3 นำสายโพรบของออสซิลโลสโคป วัดสัญญาณพัลส์ โดยที่ CH1 จับที่ขา 3 (DATA) และCH2 จับที่ขา 4 (SCK) ของพอร์ต SHT15 ตามรูปที่ 4.8

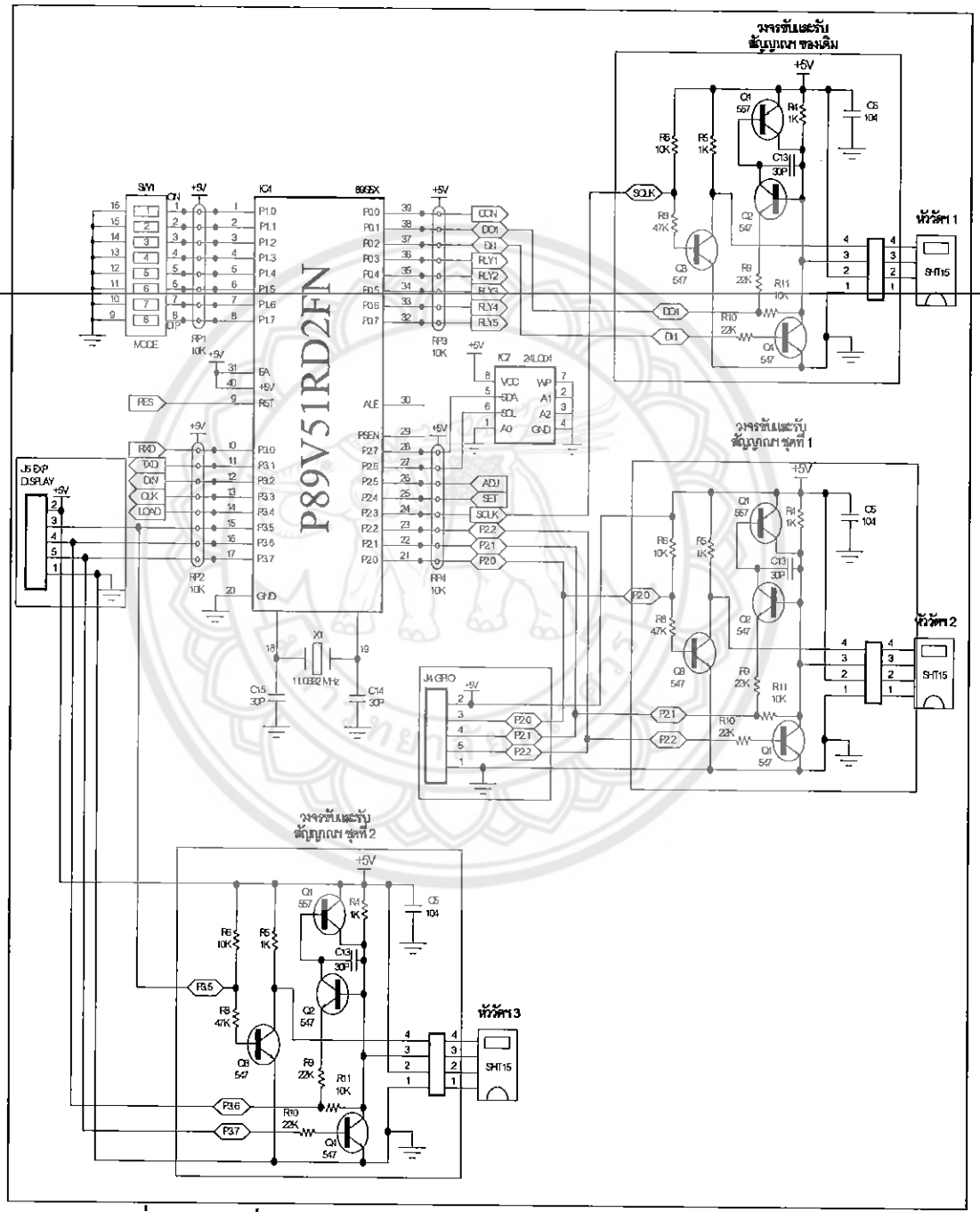
4.2.2.4 จ่ายไฟให้กับบอร์ด AP-105

4.2.2.5 บันทึกรูปคลื่นสัญญาณพัลส์ที่ได้ลงใน Floppy disk

4.2.2.6 ทำซ้ำข้อ 4.2.2.2 - 4.2.2.5 โดยเปลี่ยนความยาวของสายสัญญาณพร้อมหัววัดอุณหภูมิและความชื้น SHT15 เป็น 8 เมตรและ 50 เมตรตามลำดับ

หมายเหตุ - การแสดงผลการวัดค่าอุณหภูมิและค่าความชื้นสัมพัทธ์จะแสดงออกมาทางจอแสดงผลของบอร์ดไมโครคอนโทรลเลอร์

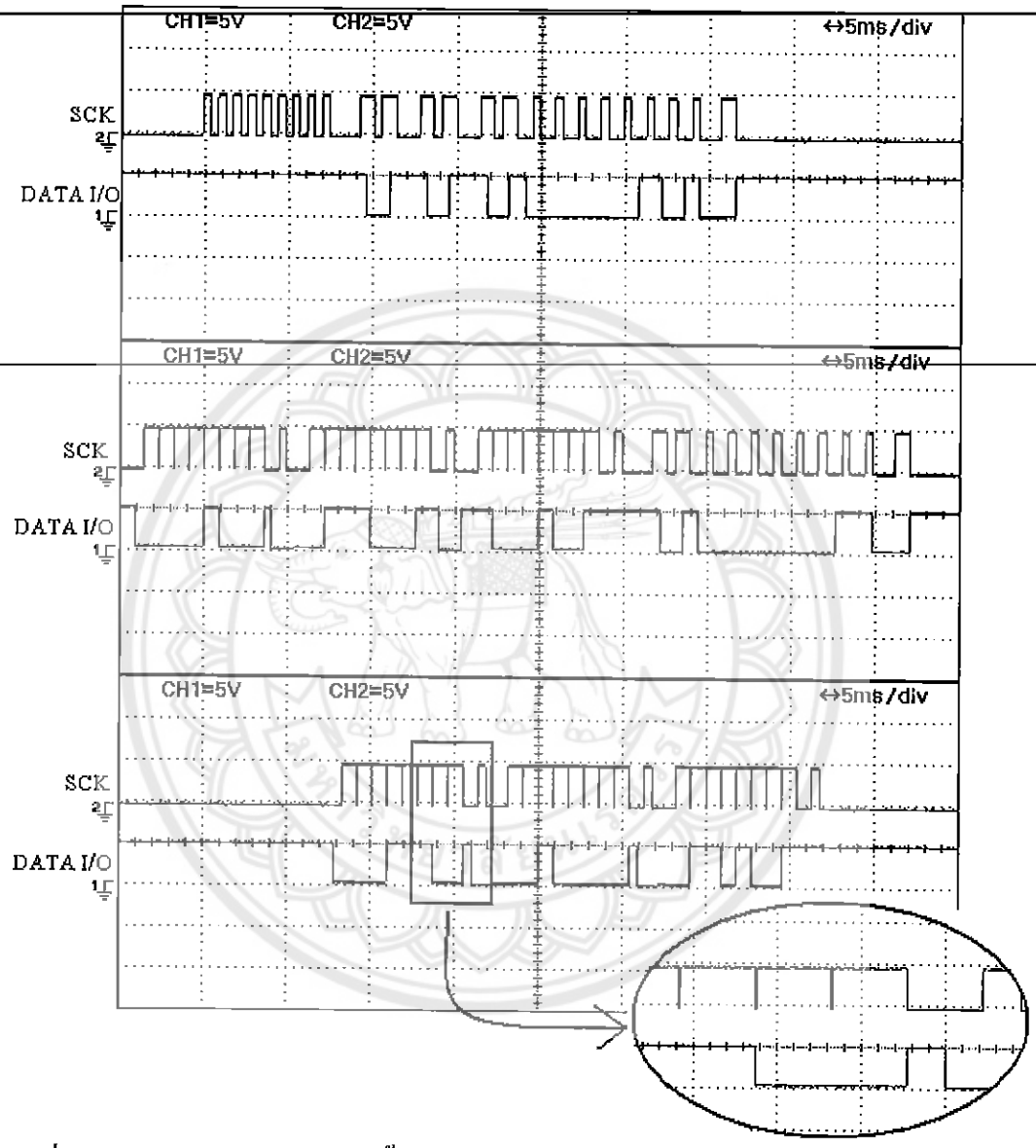
- โปรแกรมที่วัดอุณหภูมิและความชื้นสัมพัทธ์ ผ่านวงจรขับและรับสัญญาณพัลส์ โดยในแต่ละลักษณะการทดลองต้องเปลี่ยนพอร์ตในโปรแกรมให้สอดคล้องกับรูปที่ 4.9



รูปที่ 4.9 การเชื่อมต่อวงจรขับและรับสัญญาณพัลส์กับไมโครคอนโทรลเลอร์

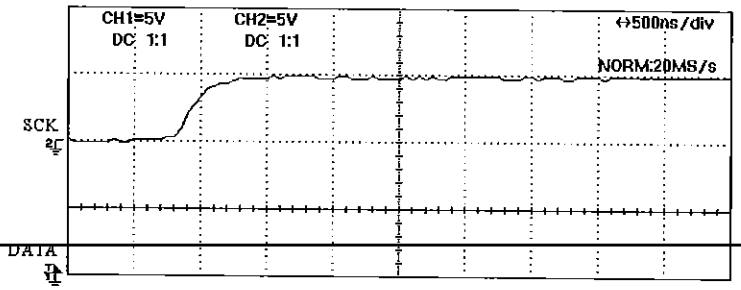
4.2.3 ผลการทดลองตอนที่ 2.1 : ผ่านวงจรฯ เดิมของบอร์ด AP-105

4.2.3.1 สายสัญญาณยาว 20 เซนติเมตร วัดค่าอุณหภูมิได้ 26.0 องศาเซลเซียส วัดค่าความชื้นสัมพัทธ์ได้ 80 เปอร์เซ็นต์

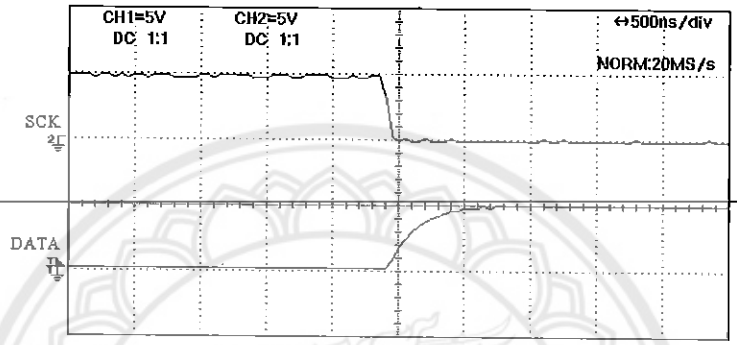


รูปที่ 4.10 กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 20 เซนติเมตร

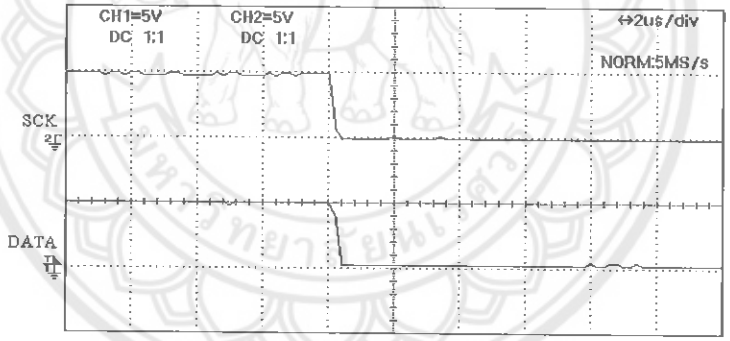
อธิบายรูปที่ 4.10 ภาพรวมของสัญญาณ โดยรวมแล้วสัญญาณ SCK และ DATA มีลักษณะเป็นแท่งสี่เหลี่ยมสมบูรณ์ ส่วนภาพด้านล่างขยายเพื่อให้เห็นความชันของขาสัญญาณ จะเห็นว่าสัญญาณ SCK มีค่า 5V ซึ่งเป็นไปตามความต้องการของ Chip SHT15 ส่วนสัญญาณ DATA ใช้เวลาไต่ขึ้น – ลงระหว่าง 0 – 5V น้อยมากๆ จึงทำให้กราฟมีความชันมาก ส่งผลให้สามารถส่งข้อมูลระหว่าง SHT15 กับไมโครคอนโทรลเลอร์ ได้ถูกต้อง



(ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK



(ข) ช่วงเวลาตกของสัญญาณ SCK และ ช่วงเวลาไต่ขึ้นของสัญญาณ DATA



(ค) ช่วงเวลาตกของสัญญาณ DATA และ SCK

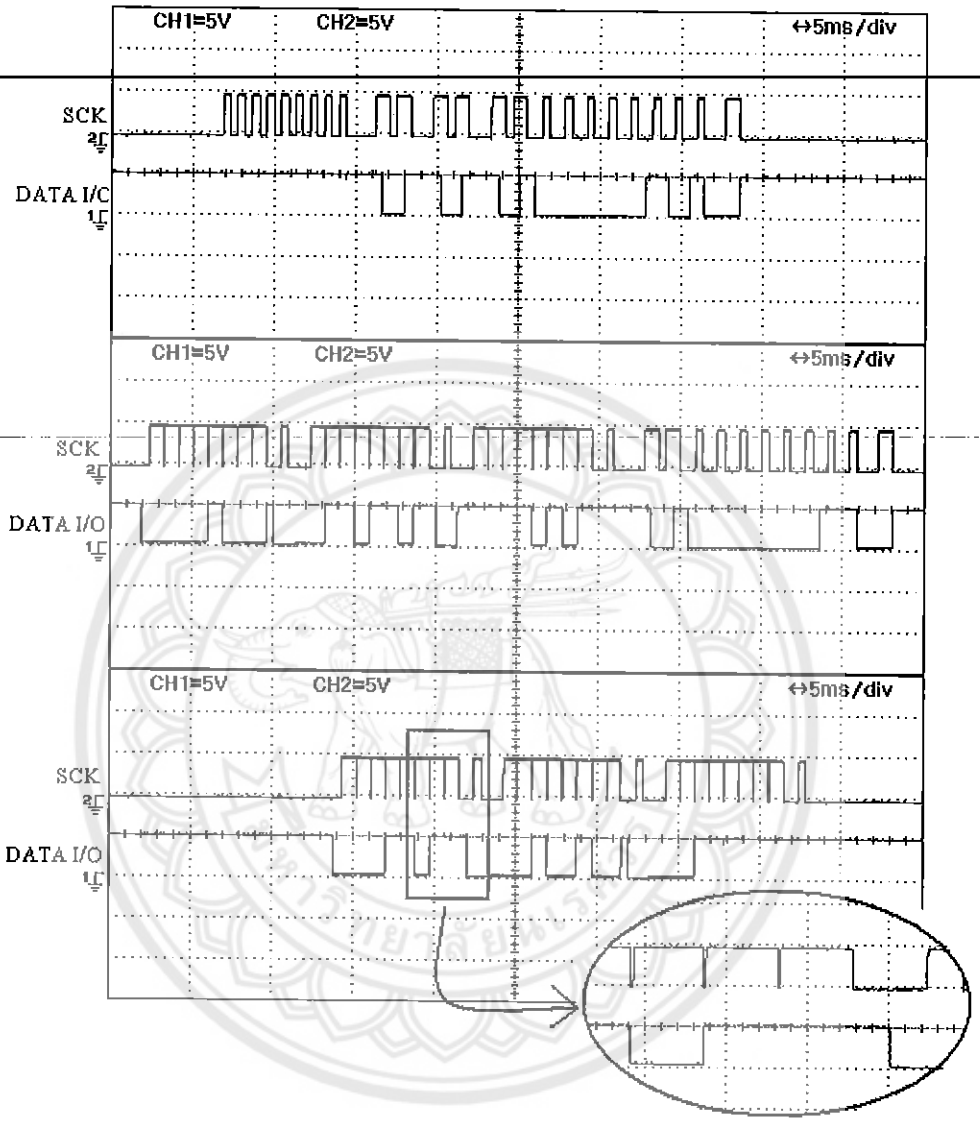
รูปที่ 4.11 กราฟแสดงค่า t_r และ t_f ผ่านวงจรในบอร์ด AP-105 V1.0 ที่ความยาวสาย 20 เซนติเมตร

อธิบายรูปที่ 4.11 (ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK จาก 0V ขึ้นไป 5V ใช้เวลาประมาณ 1,200μs ส่วนขา DATA มีแรงดันประมาณ 5V ลักษณะสัญญาณของ SCK มี noise ค่อนข้างมากแต่สัญญาณ DATA เรียบ

รูปที่ 4.11 (ข) ช่วงเวลาตกของสัญญาณ SCK จาก 5V ลงมา 0V ใช้เวลาประมาณ 200ns ส่วนขา DATA เป็นช่วงเวลาไต่ขึ้นของสัญญาณจาก 0V ลงมา 5V ใช้เวลาประมาณ 900ns ลักษณะสัญญาณของ SCK มี noise ค่อนข้างมากแต่สัญญาณ DATA เรียบ

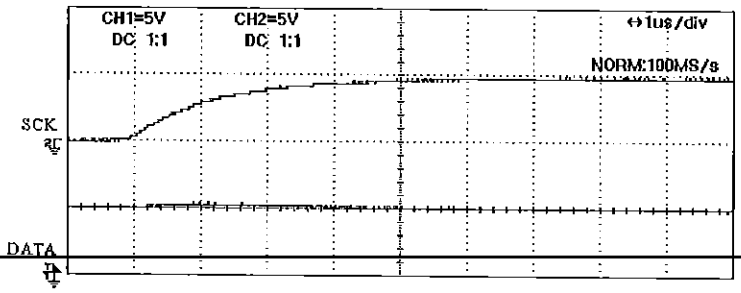
รูปที่ 4.11(ค) ช่วงเวลาตกของสัญญาณ SCK จาก 5V ลงมา 0V ใช้เวลาประมาณ 0.4μs และขา DATA จาก 5V ลงมา 0.2V ใช้เวลาประมาณ 0.4μs ลักษณะสัญญาณมี noise เล็กน้อย

4.2.3.2 สายสัญญาณยาว 8 เมตร วัดค่าอุณหภูมิได้ 25.3 องศาเซลเซียส วัดค่าความชื้นสัมพัทธ์ได้ 78 เปอร์เซ็นต์

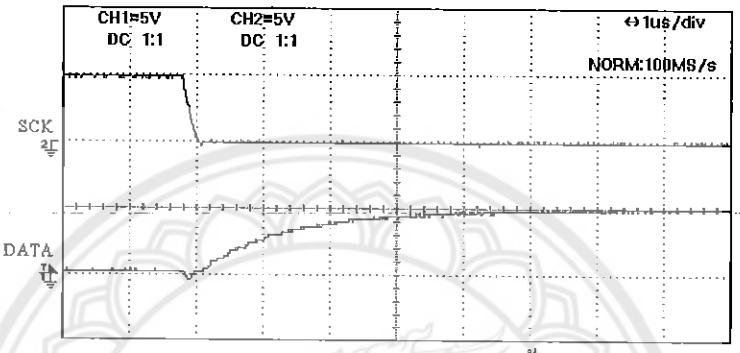


รูปที่ 4.12 กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 8 เมตร

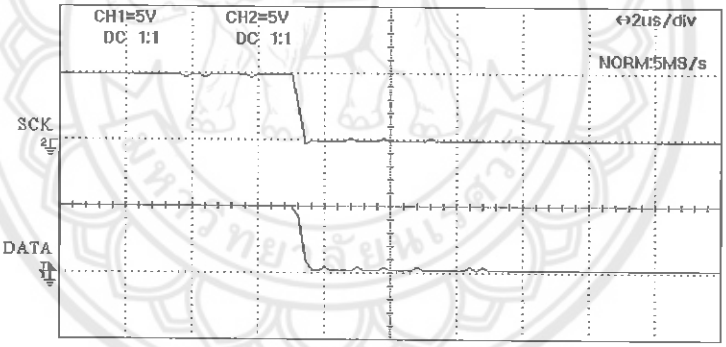
อธิบายรูปที่ 4.12 ภาพรวมของสัญญาณ โดยรวมแล้วสัญญาณ SCK และ DATA มีลักษณะเป็นแท่งสี่เหลี่ยมสมบูรณ์ ส่วนภาพด้านล่างขยายเพื่อให้เห็นความชันของขาสัญญาณ จะเห็นว่าสัญญาณ SCK มีค่า 5V ซึ่งเป็นไปตามความต้องการของ Chip SHT15 แต่ขาสัญญาณมีลักษณะบิ่นไปเล็กน้อยยังไม่เกินค่าที่ต้องการของ Chip SHT15 ส่วนสัญญาณ DATA ใช้เวลาไต่ขึ้น – ลงระหว่าง 0 – 5V น้อยมากๆ จึงทำให้กราฟมีความชันมาก ส่งผลให้สามารถส่งข้อมูลระหว่าง SHT15 กับไมโครคอนโทรลเลอร์ ได้ถูกต้อง



(ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK



(ข) ช่วงเวลาตกของสัญญาณ SCK และ ช่วงเวลาไต่ขึ้นของสัญญาณ DATA



(ค) ช่วงเวลาตกของสัญญาณ SCK และ DATA

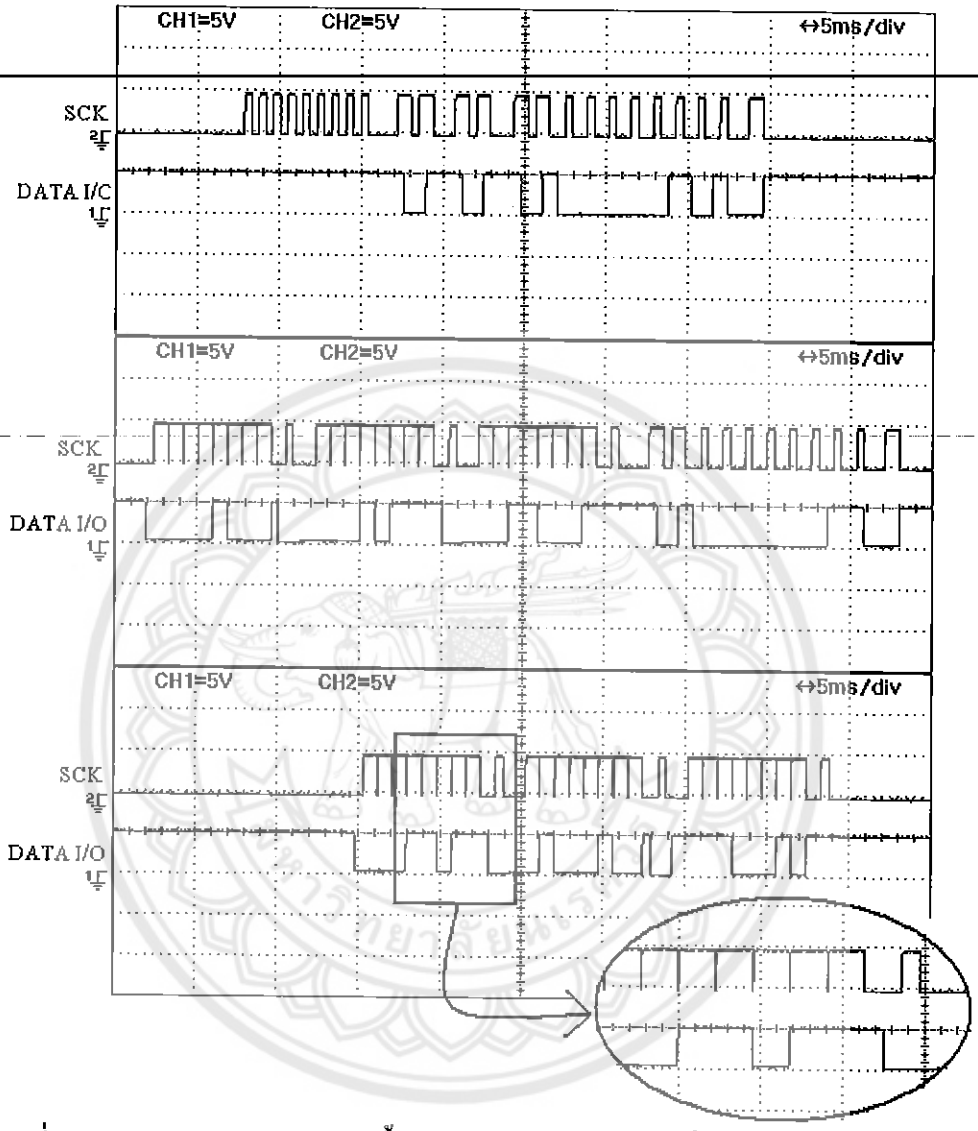
รูปที่ 4.13 กราฟแสดงค่า t_r และ t_f ผ่านวงจรในบอร์ด AP-105 V1.0 ที่ความยาวสาย 8 เมตร

อธิบายรูปที่ 4.13 (ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK จาก 0V ขึ้นไป 4.8V ใช้เวลาประมาณ 9.2μs ส่วนขา DATA มีแรงดันประมาณ 5V ลักษณะสัญญาณมี noise เล็กน้อย

รูปที่ 4.13 (ข) ช่วงเวลาตกของสัญญาณ SCK จาก 5V ลงมา 0V ใช้เวลาประมาณ 0.2μs ส่วนขา DATA เป็นช่วงเวลาไต่ขึ้นของสัญญาณจาก 0V ขึ้นไป 5V ใช้เวลาประมาณ 4.2μs ลักษณะสัญญาณมี noise เล็กน้อย

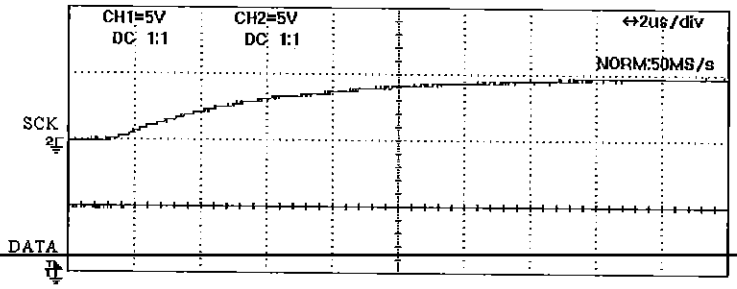
รูปที่ 4.13 (ค) ช่วงเวลาตกของสัญญาณ SCK จาก 5V ลงมา 0V ใช้เวลาประมาณ 0.8μs ส่วนขา DATA มีช่วงเวลาตกของสัญญาณจาก 5V ลงมา 0V ใช้เวลาประมาณ 0.8μs ลักษณะสัญญาณมี noise เล็กน้อย

4.2.3.3 สายสัญญาณยาว 50 เมตร วัดค่าอุณหภูมิได้ 25.5 องศาเซลเซียส วัดค่าความชื้นสัมพัทธ์ได้ 79 เปอร์เซ็นต์

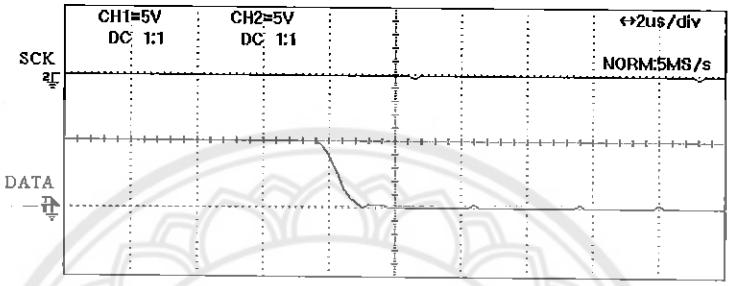


รูปที่ 4.14 กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 50 เมตร

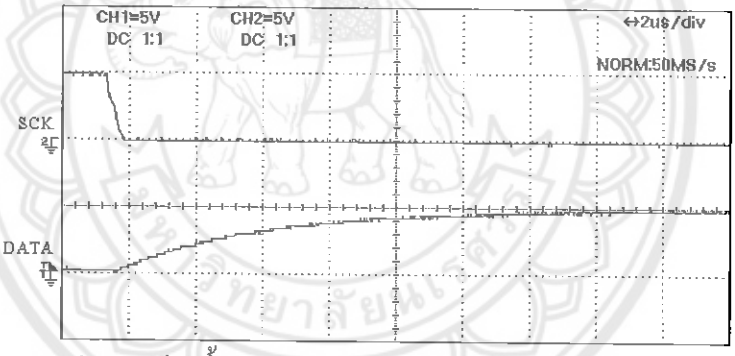
อธิบายรูปที่ 4.14 ภาพรวมของสัญญาณ โดยรวมแล้วสัญญาณ SCK และ DATA มีลักษณะเป็นแท่งสี่เหลี่ยม ส่วนภาพด้านล่างขยายเพื่อให้เห็นความชันของขาสัญญาณ จะเห็นว่า สัญญาณ SCK มีค่า 5V ซึ่งเป็นไปตามความต้องการของ Chip SHT15 ส่วนสัญญาณ DATA ขาขึ้นของสัญญาณบิ้นไปเล็กน้อย มีผลมาจากค่า RC ในสายเพิ่มขึ้น ส่วนขาลงของสัญญาณมีค่าน้อยมาก แต่ค่าช่วงเวลาไต่ขึ้นและเวลาตกยังคงอยู่ในช่วงความต้องการของ Chip SHT15 อยู่ จึงทำให้สามารถส่งข้อมูลระหว่าง SHT15 กับไมโครคอนโทรลเลอร์ ได้ถูกต้อง



(ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK



(ข) ช่วงเวลาตกลงของสัญญาณ DATA



(ค) ช่วงเวลาไต่ขึ้นของสัญญาณ DATA และช่วงเวลาตกของ SCK

รูปที่ 4.15 กราฟแสดงค่า t_r และ t_f ผ่านวงจรในบอร์ด AP-105 V1.0 ที่ความยาวสาย 50 เมตร

อธิบายรูปที่ 4.15 (ก) ช่วงเวลาตกของสัญญาณ SCK จาก 0V ขึ้นไป 5V ใช้เวลาประมาณ 18.8μs ส่วนขา DATA มีแรงดันประมาณ 5V ลักษณะสัญญาณ SCK มี noise เล็กน้อย แต่สัญญาณ DATA เรียบ

รูปที่ 4.15 (ข) จากรูป SCK มีแรงดันประมาณ 0V ส่วนขา DATA เป็นช่วงเวลาตกของสัญญาณจาก 5V ลงมา 0V ใช้เวลาประมาณ 2.4μs ลักษณะสัญญาณมี noise เล็กน้อย

รูปที่ 4.15 (ค) ช่วงเวลาตกของสัญญาณ SCK จาก 5V ลงมา 0V ใช้เวลาประมาณ 0.8μs ส่วนขา DATA เป็นช่วงเวลาไต่ขึ้นของสัญญาณจาก 0V ขึ้นไป 5V ใช้เวลาประมาณ 18.4μs ลักษณะสัญญาณมี noise เล็กน้อย

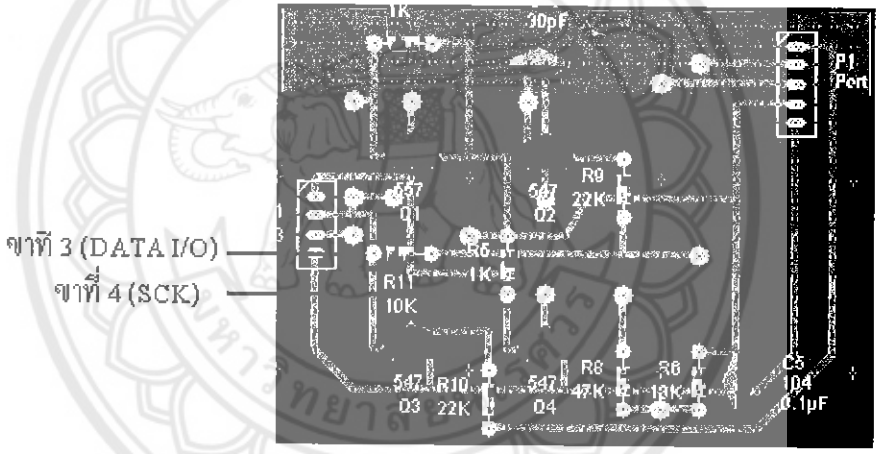
4.2.4 ขั้นตอนการทดลองตอนที่ 2.2 : ผ่านวงจรฯ ที่สร้างขึ้นใหม่

4.2.4.1 นำโปรแกรมที่วัดอุณหภูมิและความชื้นสัมพัทธ์ โดยผ่านวงจรขับและรับสัญญาณพัลส์ที่เขียนไว้ โหลดลงไมโครคอนโทรลเลอร์ของบอร์ด AP-105 (โดยวงจรที่ 1 กำหนดให้ DO2 เป็นพอร์ต P2.1, DI2 เป็นพอร์ต P2.2 และ SCK2 เป็นพอร์ต P2.0 ส่วนวงจรที่ 2 กำหนดให้ DO3 เป็นพอร์ต P3.6, DI3 เป็นพอร์ต P3.7 และ SCK3 เป็นพอร์ต P3.5)

4.2.4.2 นำวงจรขับและรับสัญญาณต่อเข้าที่พอร์ต GPIO (วงจรที่ 1) และพอร์ต EXP Display (วงจรที่ 2) ตามรูปที่ 4.8

4.2.4.3 นำหัววัดอุณหภูมิและความชื้น SHT15 ที่เชื่อมต่อกับสายสัญญาณยาว 20 เซนติเมตร ต่อเข้า กับวงจรขับและรับสัญญาณพัลส์ ชุดที่ 1 ตามรูปที่ 4.8

4.2.4.4 นำสายโพรบของออสซิลโลสโคป วัดสัญญาณพัลส์ โดยที่ CH1 จับที่ขา 3 (DATA) และCH2 จับที่ขา 4 (SCK) ตามรูปที่ 4.16



รูปที่ 4.16 ตำแหน่งในการวัดสัญญาณพัลส์

4.2.4.5 จ่ายไฟให้กับบอร์ด AP-105

4.2.4.6 บันทึกรูปคลื่นสัญญาณพัลส์ที่ได้ลงใน floppy disk

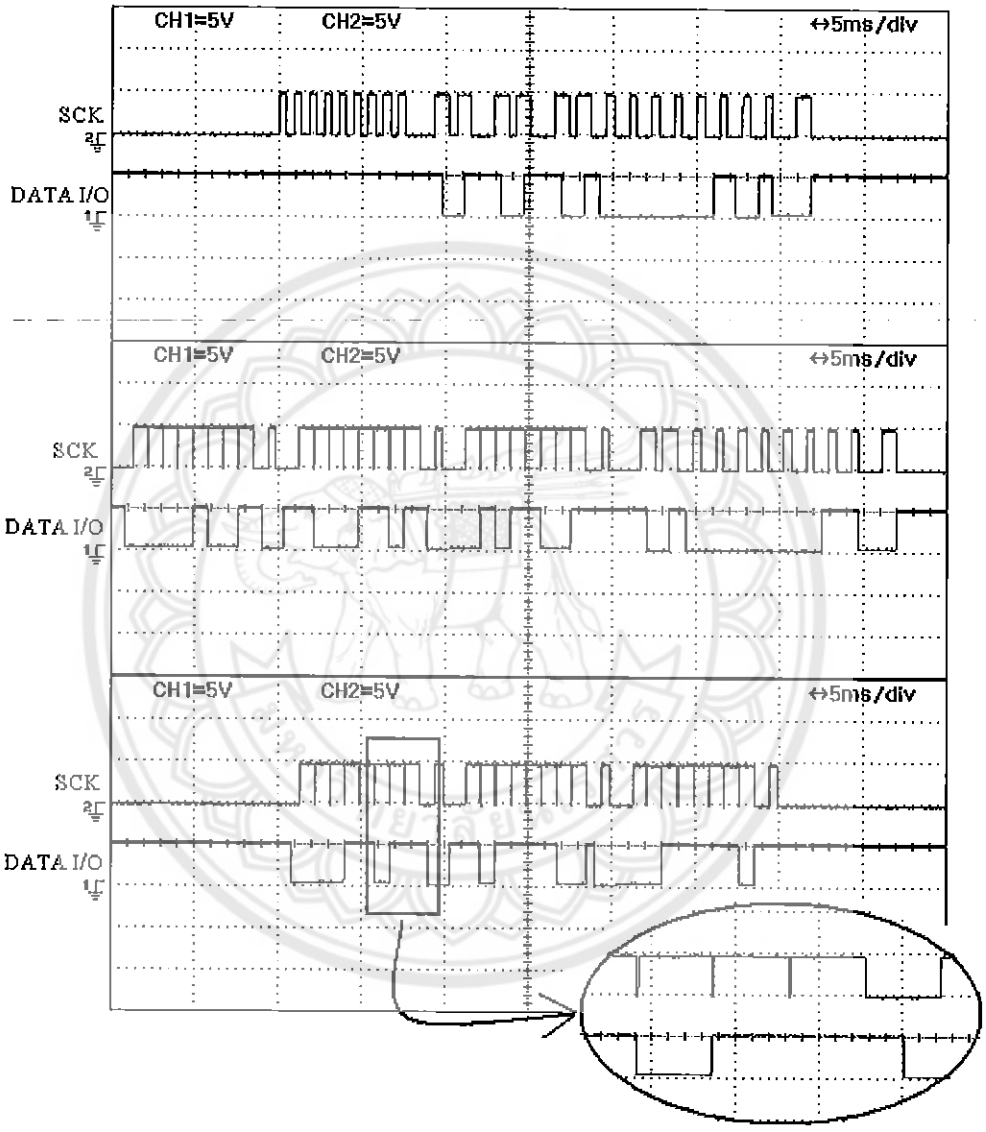
4.2.4.7 ทำซ้ำข้อ 4.2.4.2 - 4.2.4.6 โดยเปลี่ยนความยาวของสายสัญญาณพร้อมหัววัดอุณหภูมิและความชื้น SHT15 เป็น 8 เมตรและ 50 เมตรตามลำดับ

4.2.4.8 ทำซ้ำข้อ 4.2.4.3 - 4.2.4.7 โดยเปลี่ยนเป็นวงจรชุดที่ 2

4.2.5 ผลการทดลองตอนที่ 2.2 : ผ่านวงจรฯ ที่สร้างขึ้นใหม่

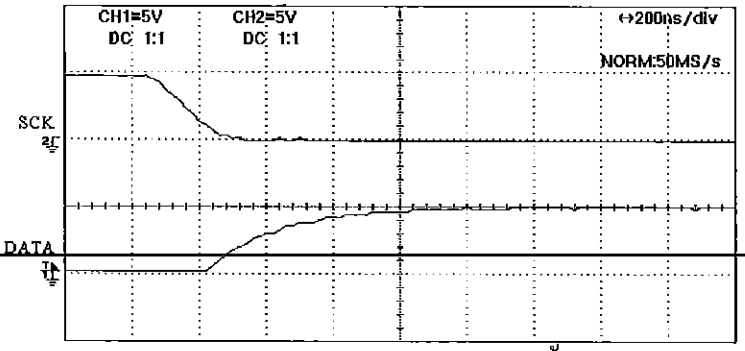
4.2.5.1 ผ่านวงจรฯ ชุดที่ 1 (พอร์ต GPIO)

- สายสัญญาณยาว 20 เซนติเมตร วัดค่าอุณหภูมิได้ 26.1 องศาเซลเซียส วัดค่าความชื้นสัมพัทธ์ได้ 76 เปอร์เซ็นต์

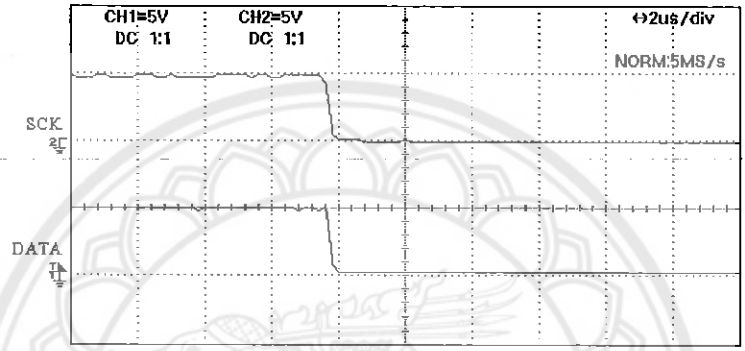


รูปที่ 4.17 กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 20 เซนติเมตร

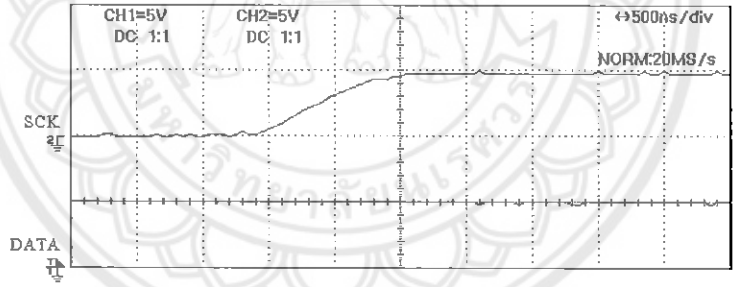
อธิบายรูปที่ 4.17 ภาพรวมของสัญญาณ โดยรวมแล้วสัญญาณ SCK และ DATA มีลักษณะเป็นแท่งสี่เหลี่ยมสมบูรณ์ ส่วนภาพด้านล่างขยายเพื่อให้เห็นความชันของขาสัญญาณ จะเห็นว่าสัญญาณ SCK มีค่า 5V ซึ่งเป็นไปตามความต้องการของ Chip SHT15 ส่วนสัญญาณ DATA ใช้เวลาไต่ขึ้น – ลงระหว่าง 0 – 5V น้อยมากๆ จึงทำให้กราฟมีความชันมาก ส่งผลให้สามารถส่งข้อมูลระหว่าง SHT15 กับไมโครคอนโทรลเลอร์ ได้ถูกต้อง



(ก) ช่วงเวลาตกลงของสัญญาณ SCK และช่วงเวลาไต่ขึ้นของสัญญาณ DATA



(ข) ช่วงเวลาตกลงของสัญญาณ SCK และ DATA



(ค) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK

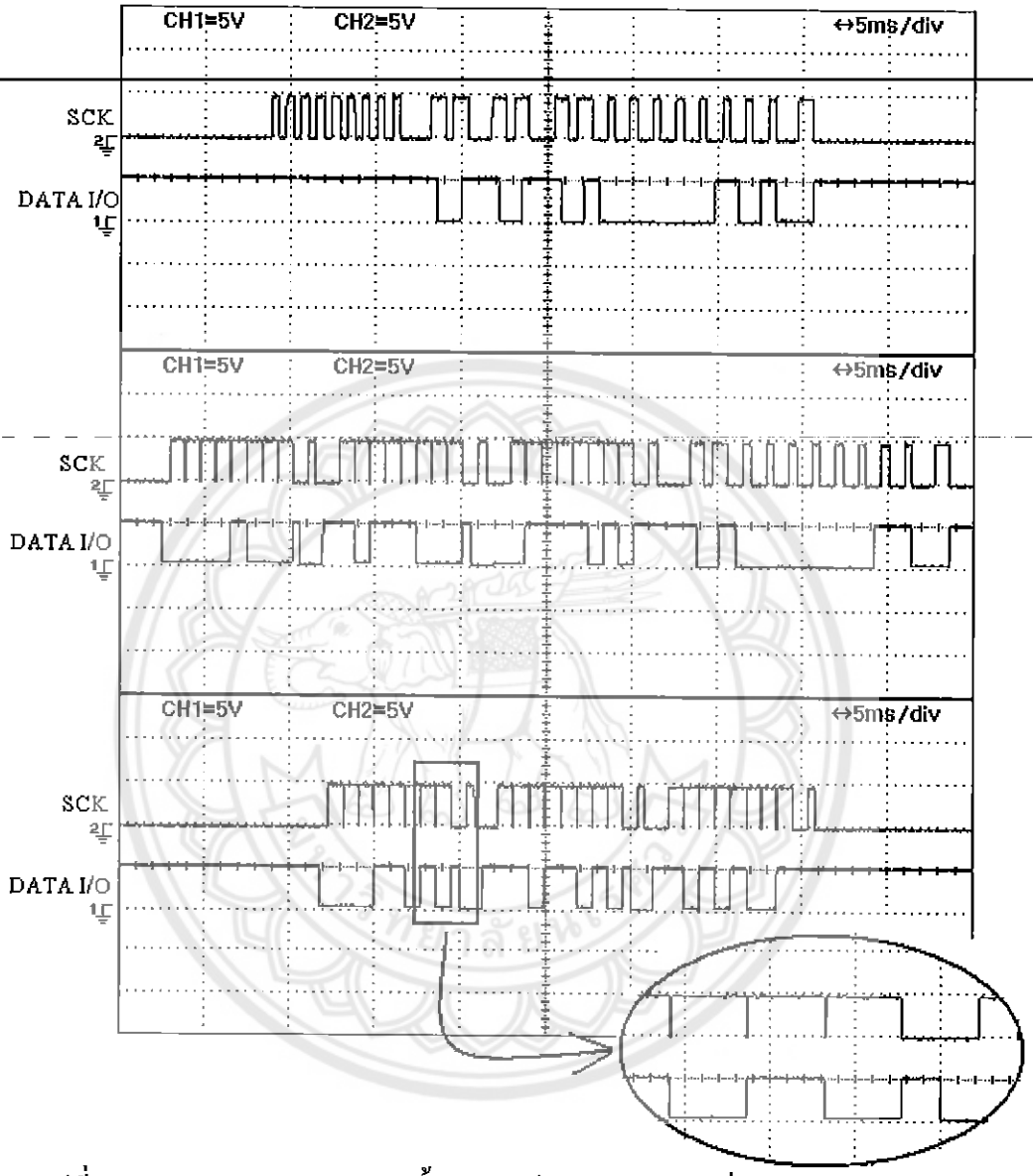
รูปที่ 4.18 กราฟแสดงค่า t_r และ t_f ผ่านวงจรชุดที่ 1 (พอร์ต GPIO) ที่ความยาวสาย 20 เซนติเมตร

อธิบายรูปที่ 4.18 (ก) ช่วงเวลาตกของสัญญาณ SCK จาก 4.8V ลงมา 0V ใช้เวลาประมาณ 360ns ส่วนขา DATA เป็นช่วงเวลาไต่ขึ้นของสัญญาณจาก 0V ขึ้นไป 5V ใช้เวลาประมาณ 1,000µs ลักษณะสัญญาณมี noise เล็กน้อย

รูปที่ 4.18 (ข) ช่วงเวลาตกของสัญญาณ SCK จาก 5V ลงมา 0V ใช้เวลาประมาณ 0.8µs ส่วนขา DATA ช่วงเวลาตกของสัญญาณจาก 5V ลงมา 0V ใช้เวลาประมาณ 0.4µs ลักษณะสัญญาณมี noise เล็กน้อย

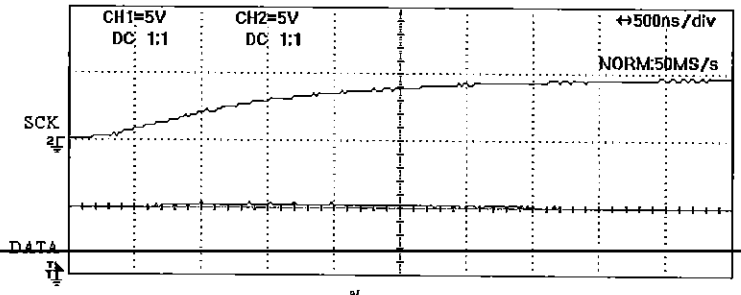
รูปที่ 4.18 (ค) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK จาก 0V ขึ้นไป 4.8V ใช้เวลาประมาณ 2,500ns ส่วนขา DATA มีแรงดันประมาณ 5V ลักษณะสัญญาณมี noise เล็กน้อย

- สายสัญญาณยาว 8 เมตร วัดค่าอุณหภูมิได้ 25.6 องศาเซลเซียส วัดค่าความชื้นสัมพัทธ์ได้ 79 เปอร์เซ็นต์

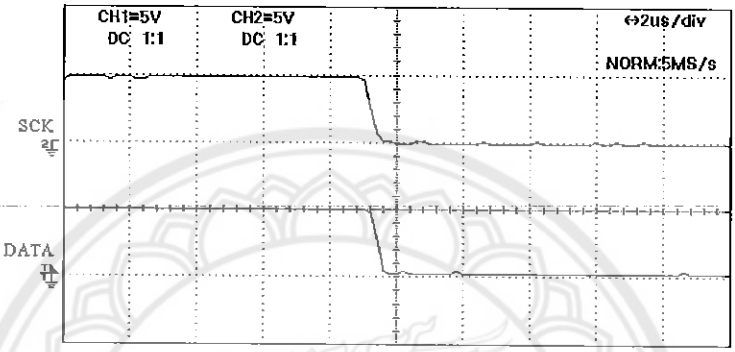


รูปที่ 4.19 กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 8 เมตร

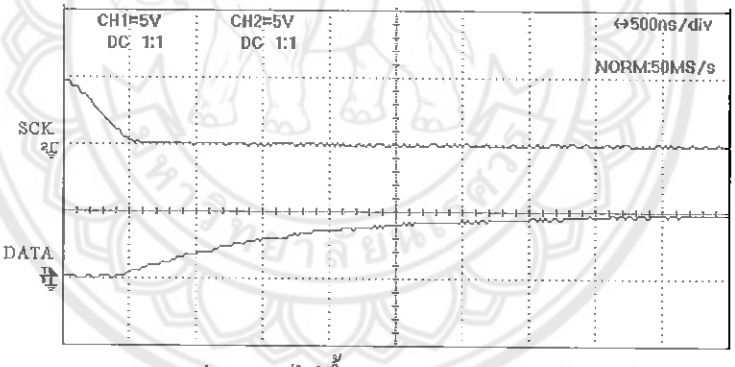
อธิบายรูปที่ 4.19 ภาพรวมของสัญญาณ โดยรวมแล้วสัญญาณ SCK และ DATA มีลักษณะเป็นแท่งสี่เหลี่ยมสมบูรณ์ ส่วนภาพด้านล่างขยายเพื่อให้เห็นความชันของขาสัญญาณ จะเห็นว่าสัญญาณ SCK มีค่า 5V ซึ่งเป็นไปตามความต้องการของ Chip SHT15 ส่วนสัญญาณ DATA ใช้เวลาไต่ขึ้น – ลงระหว่าง 0 – 5V น้อยมากๆ จึงทำให้กราฟมีความชันมาก ส่งผลให้สามารถส่งข้อมูลระหว่าง SHT15 กับไมโครคอนโทรลเลอร์ ได้ถูกต้อง



(ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK



(ข) ช่วงเวลาตกลงของสัญญาณ SCK และ DATA



(ค) ช่วงเวลาไต่ขึ้นของสัญญาณ DATA

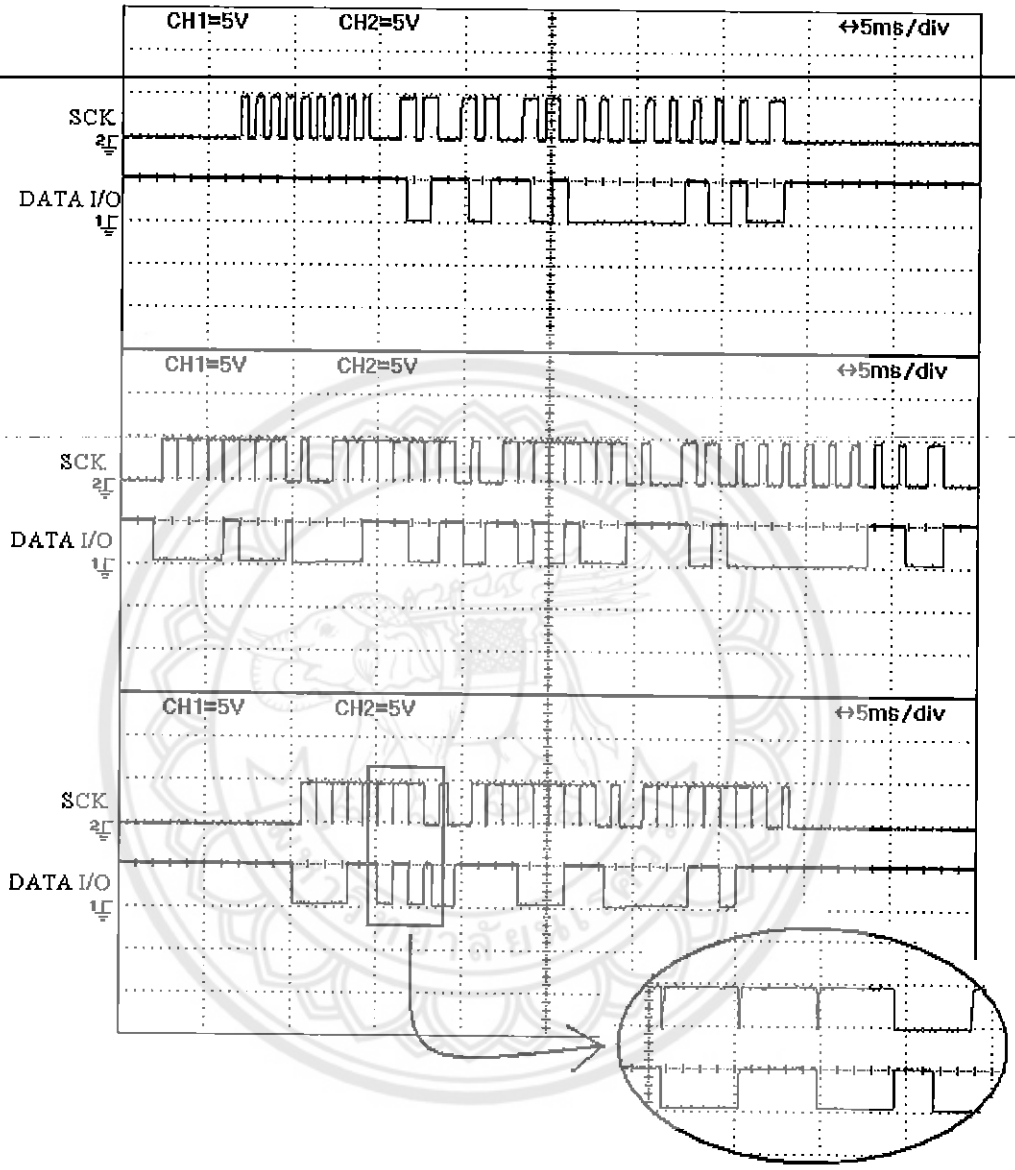
รูปที่ 4.20 กราฟแสดงค่า t_r และ t_f ผ่านวงจรชุดที่ 1 (พอร์ต GPIO) ที่ความยาวสาย 8 เมตร

อธิบายรูปที่ 4.20 (ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK จาก 0V ขึ้นไป 4.8V ใช้เวลาประมาณ 5,000ns ส่วนขา DATA มีแรงดันประมาณ 5V ลักษณะสัญญาณมี noise เล็กน้อย

รูปที่ 4.20 (ข) ช่วงเวลาตกของสัญญาณ SCK จาก 5V ลงมา 0V ใช้เวลาประมาณ 1.2μs ส่วนขา DATA ช่วงเวลาตกของสัญญาณจาก 5V ลงมา 0.2V ใช้เวลาประมาณ 1.2μs ลักษณะสัญญาณมี noise เล็กน้อย

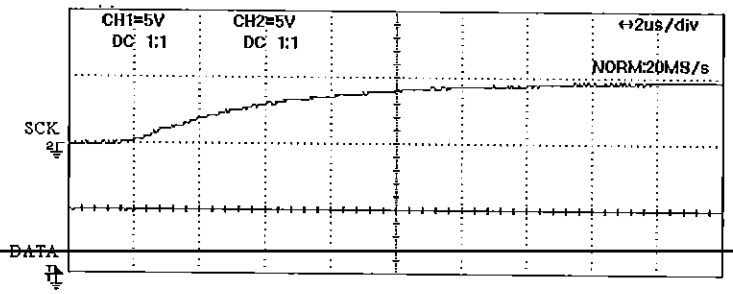
รูปที่ 4.20 (ค) จากรูปสัญญาณ SCK มีแรงดัน 0V ส่วนขา DATA เป็นช่วงไต่ขึ้นของสัญญาณจาก 0V ขึ้นไป 5V ใช้เวลาประมาณ 4,500ns ลักษณะสัญญาณมี noise ค่อนข้างมาก

- สายสัญญาณยาว 50 เมตร วัดค่าอุณหภูมิได้ 25.6 องศาเซลเซียส วัดค่าความชื้นสัมพัทธ์ได้ 78 เปอร์เซ็นต์

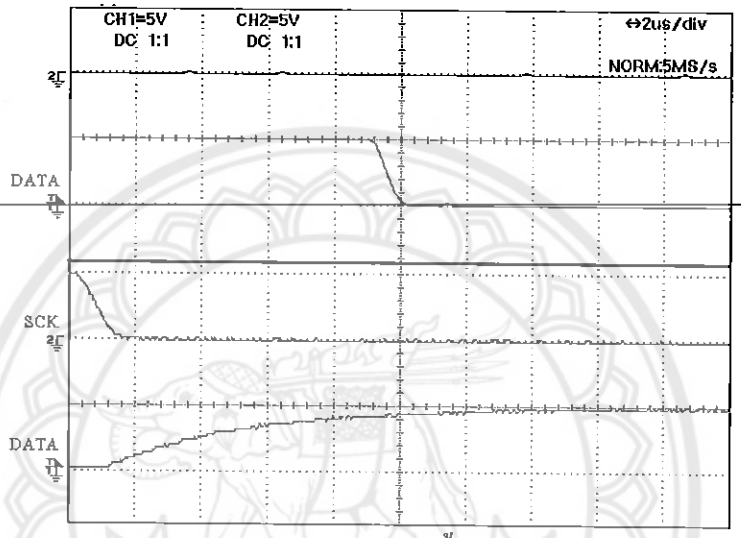


รูปที่ 4.21 กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 50 เมตร

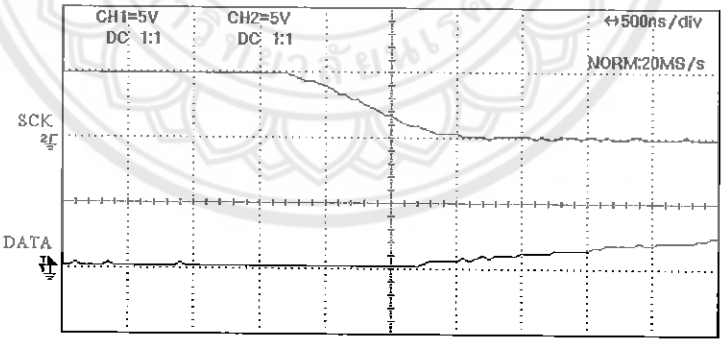
อธิบายรูปที่ 4.21 ภาพรวมของสัญญาณ โดยรวมแล้วสัญญาณ SCK และ DATA มีลักษณะเป็นแท่งสี่เหลี่ยม ส่วนภาพด้านล่างขยายเพื่อให้เห็นความชันของขาสัญญาณ จะเห็นว่า สัญญาณ SCK มีค่า 5V ซึ่งเป็นไปตามความต้องการของ Chip SHT15 ส่วนสัญญาณ DATA ขาขึ้นของสัญญาณบิ่นไปเล็กน้อย มีผลมาจากค่า RC ในสายเพิ่มขึ้น ส่วนขาลงของสัญญาณมีค่าน้อยมาก แต่ค่าช่วงเวลาไต่ขึ้นและเวลาตกยังคงอยู่ในช่วงความต้องการของ Chip SHT15 อยู่ จึงทำให้สามารถส่งข้อมูลระหว่าง SHT15 กับไมโครคอนโทรลเลอร์ ได้ถูกต้อง



(ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK



(ข) ช่วงเวลาตกลงและช่วงเวลาไต่ขึ้นของสัญญาณ DATA



(ค) ช่วงเวลาตกลงของสัญญาณ SCK

รูปที่ 4.22 กราฟแสดงค่า t_r และ t_f ผ่านวงจรชุดที่ 1 (พอร์ต GPIO) ที่ความยาวสาย 50 เมตร

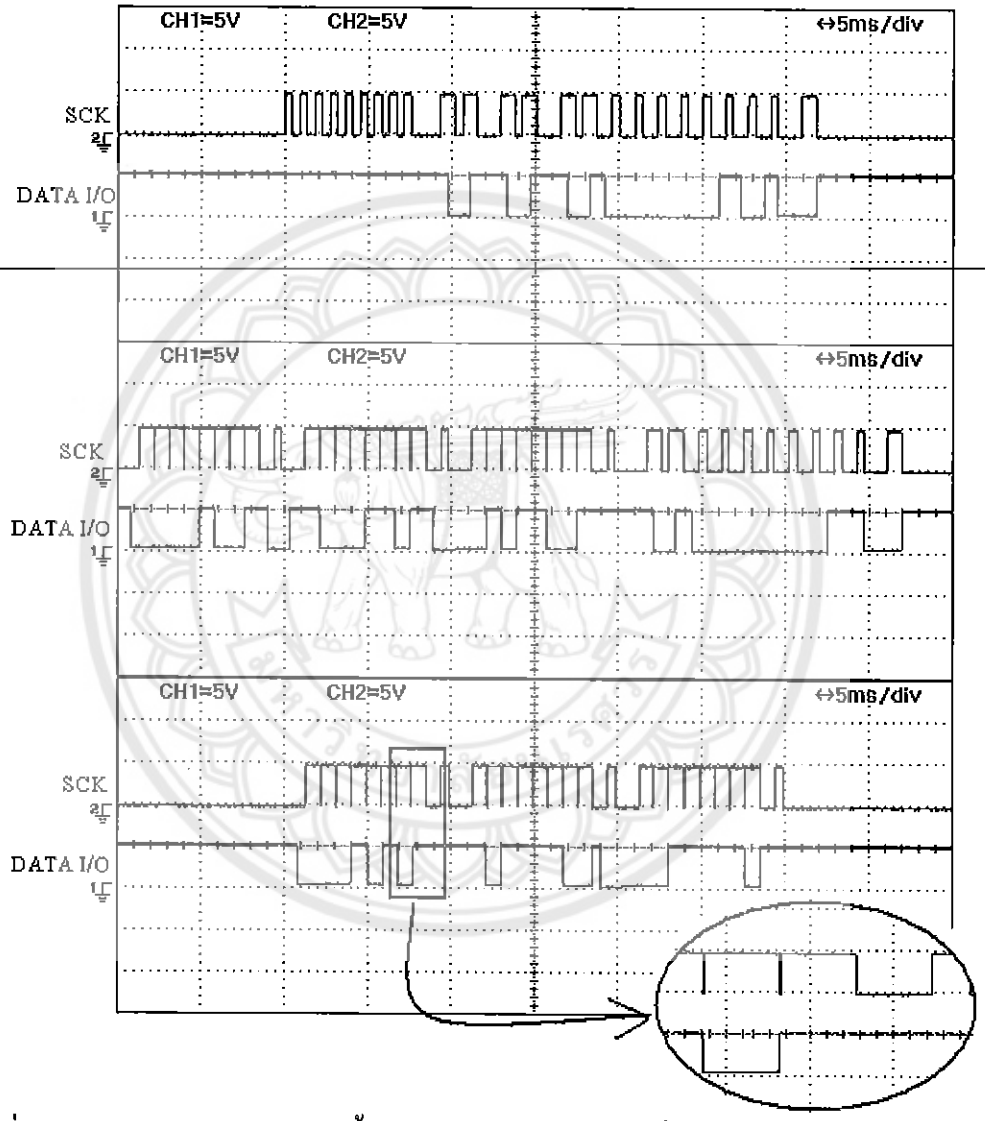
อธิบายรูปที่ 4.22 (ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK จาก 0V ขึ้นไป 4.8V ใช้เวลาประมาณ 18μs ส่วนขา DATA มีแรงดันประมาณ 5V ลักษณะสัญญาณมี noise เล็กน้อย

รูปที่ 4.22 (ข) ช่วงเวลาตกของสัญญาณ SCK จาก 5V ลงมา 0V ใช้เวลาประมาณ 2μs ส่วนช่วงเวลาไต่ขึ้นของสัญญาณ DATA จาก 0V ขึ้นไป 5V ใช้เวลาประมาณ 9.8μs ลักษณะสัญญาณมี noise เล็กน้อย

รูปที่ 4.22 (ค) ช่วงเวลาตกของสัญญาณ SCK จาก 5V ลงมา 0V ใช้เวลาประมาณ 1,500ns ลักษณะสัญญาณมี noise เล็กน้อย

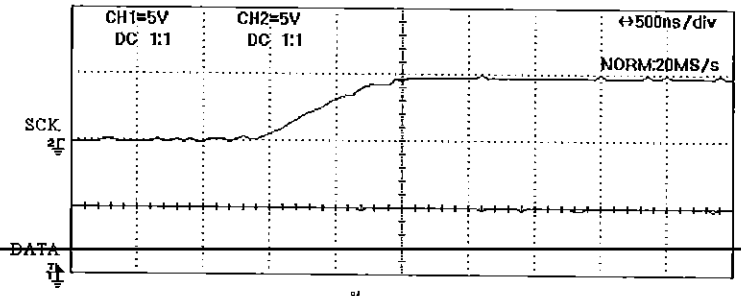
4.2.5.2 ผ่านวงจรฯ ชุดที่ 2 (พอร์ต EXP Display)

- สายสัญญาณยาว 20 เซนติเมตร วัดค่าอุณหภูมิได้ 25.9 องศาเซลเซียส วัดค่าความชื้นสัมพัทธ์ได้ 80 เปอร์เซ็นต์

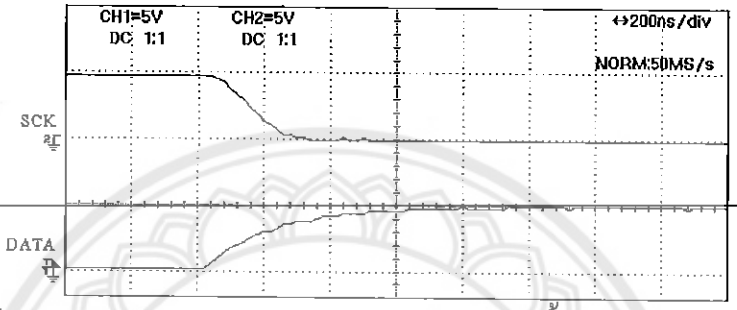


รูปที่ 4.23 กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 20 เซนติเมตร

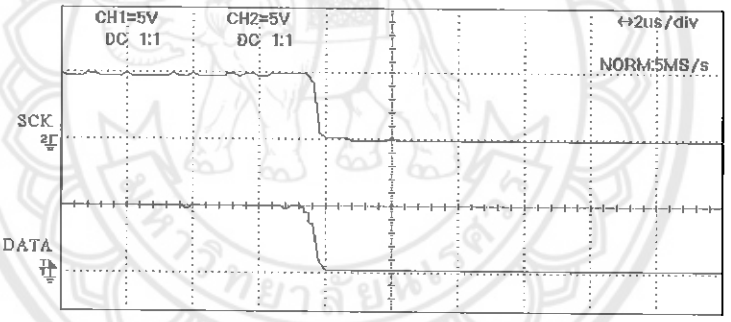
อธิบายรูปที่ 4.23 ภาพรวมของสัญญาณ โดยรวมแล้วสัญญาณ SCK และ DATA มีลักษณะเป็นแท่งสี่เหลี่ยมสมบูรณ์ ส่วนภาพด้านล่างขยายเพื่อให้เห็นความชันของขาสัญญาณ จะเห็นว่าสัญญาณ SCK มีค่า 5V ซึ่งเป็นไปตามความต้องการของ Chip SHT15 ส่วนสัญญาณ DATA ใช้เวลาไต่ขึ้น – ลงระหว่าง 0 – 5V น้อยมากๆ จึงทำให้กราฟมีความชันมาก ส่งผลให้สามารถส่งข้อมูลระหว่าง SHT15 กับไมโครคอนโทรลเลอร์ ได้ถูกต้อง



(ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK



(ข) ช่วงเวลาตกลงของสัญญาณ SCK และช่วงเวลาไต่ขึ้นของสัญญาณ DATA



(ค) ช่วงเวลาตกลงของสัญญาณ SCK และ DATA

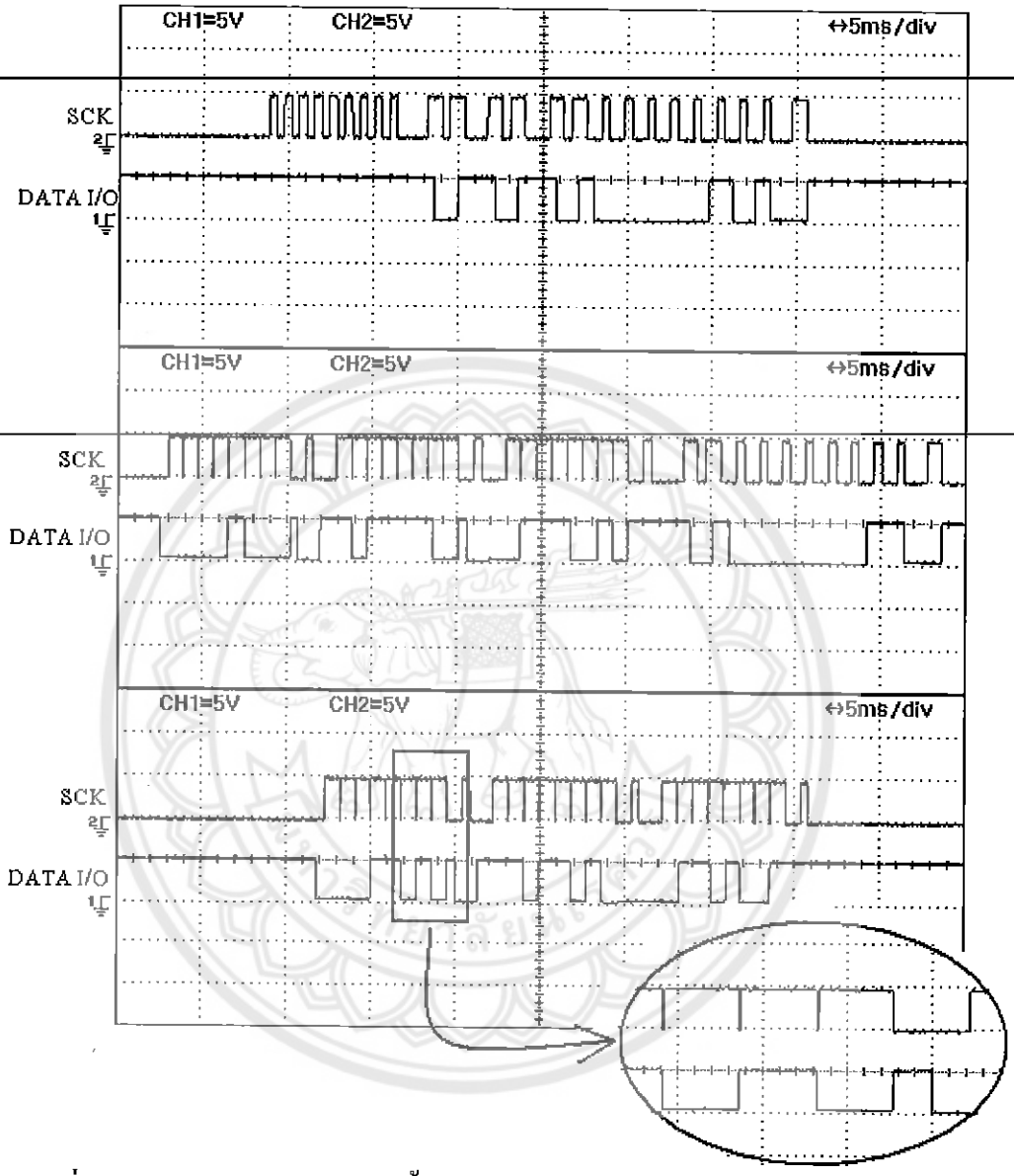
รูปที่ 4.24 กราฟแสดงค่า t_r และ t_f ผ่านวงจรฯ ชุดที่ 2 (พอร์ต EXP Display) ที่ความยาวสาย 20 เซนติเมตร

อธิบายรูปที่ 4.24 (ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK จาก 0V ขึ้นไป 4.8V ใช้เวลาประมาณ 1,600ns ส่วนขา DATA มีแรงดันประมาณ 5V ลักษณะสัญญาณมี noise เล็กน้อย

รูปที่ 4.24 (ข) ช่วงเวลาตกของสัญญาณ SCK จาก 4.8V ลงมา 0V ใช้เวลาประมาณ 360ns ส่วนขา DATA เป็นช่วงเวลาไต่ขึ้นของสัญญาณจาก 0.2V ขึ้นไป 5V ใช้เวลาประมาณ 1,000ns ลักษณะสัญญาณมี noise เล็กน้อย

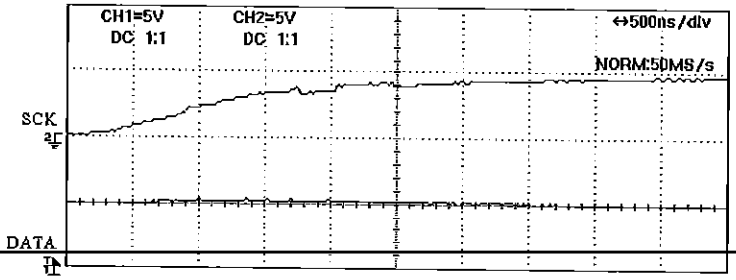
รูปที่ 4.24 (ค) ช่วงเวลาตกของสัญญาณ SCK จาก 4.8V ลงมา 0V ใช้เวลาประมาณ 0.8µs ส่วนขา DATA ช่วงเวลาตกของสัญญาณจาก 5V ลงมา 0.2V ใช้เวลาประมาณ 0.8µs ลักษณะสัญญาณมี noise เล็กน้อย

- สายสัญญาณยาว 8 เมตร วัดค่าอุณหภูมิได้ 25.4 องศาเซลเซียส วัดค่าความชื้นสัมพัทธ์ได้ 79 เปอร์เซ็นต์

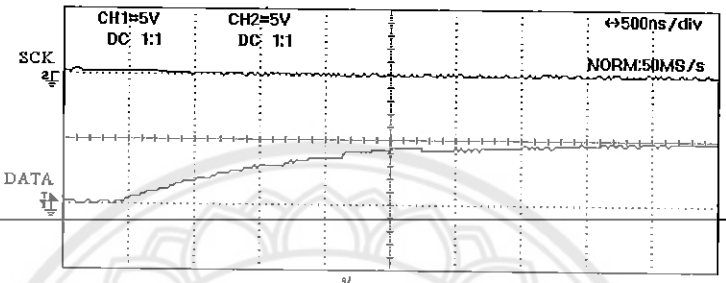


รูปที่ 4.25 กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 8 เมตร

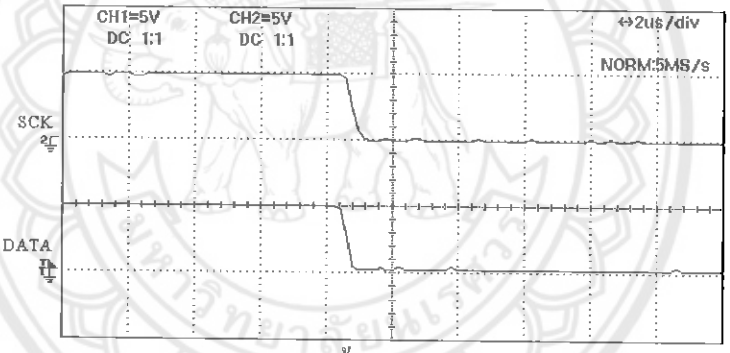
อธิบายรูปที่ 4.25 ภาพรวมของสัญญาณ โดยรวมแล้วสัญญาณ SCK และ DATA มีลักษณะเป็นแท่งสี่เหลี่ยม ส่วนภาพด้านล่างขยายเพื่อให้เห็นความชันของขาสัญญาณ จะเห็นว่า สัญญาณ SCK มีค่า 5V ซึ่งเป็นไปตามความต้องการของ Chip SHT15 ส่วนสัญญาณ DATA ใช้เวลาไต่ขึ้น – ลงระหว่าง 0 – 5V น้อยมากๆ จึงทำให้กราฟมีความชันมาก ส่งผลให้สามารถส่งข้อมูลระหว่าง SHT15 กับไมโครคอนโทรลเลอร์ ได้ถูกต้อง



(ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK



(ข) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK



(ค) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK

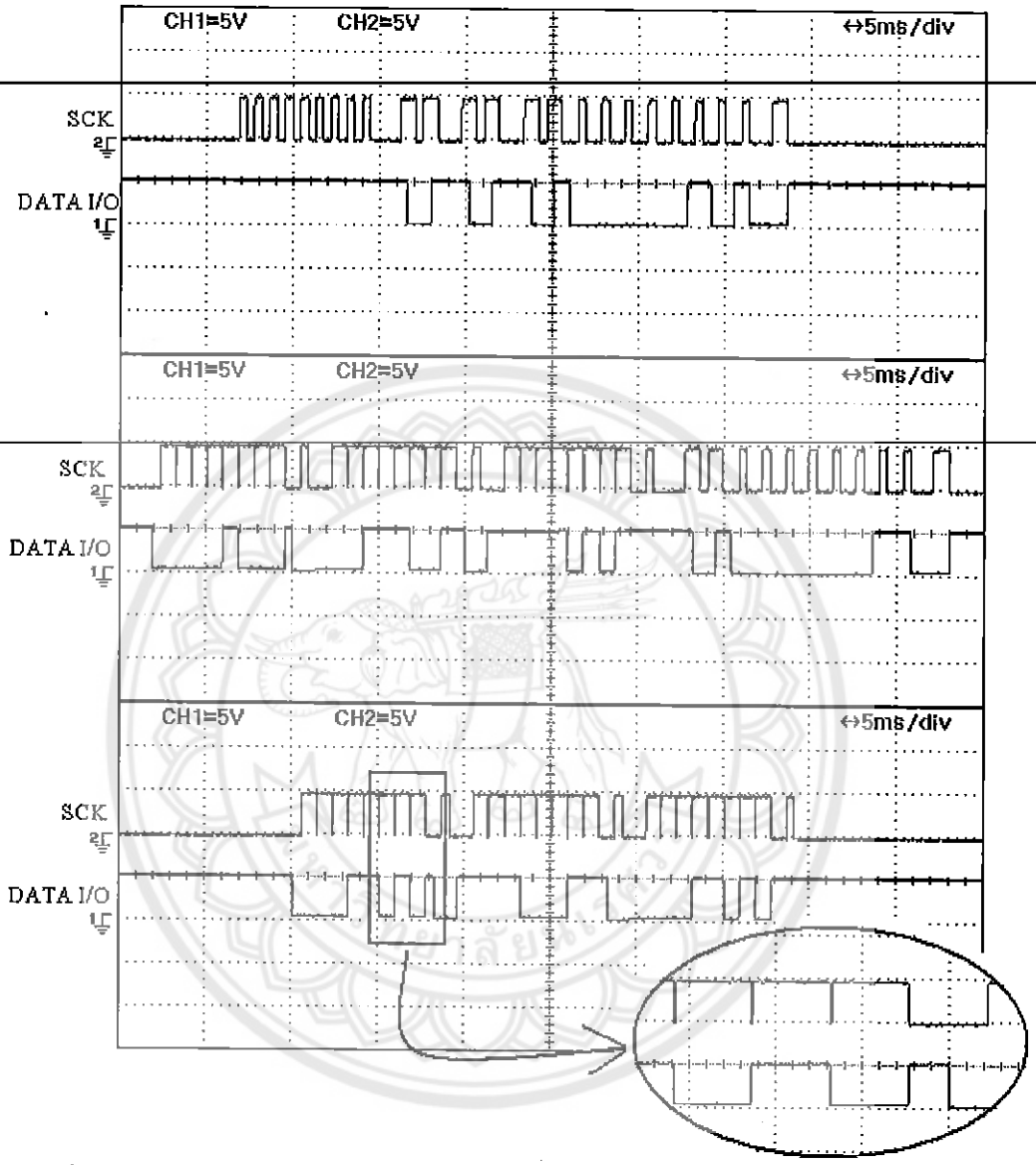
รูปที่ 4.26 กราฟแสดงค่า t_r และ t_f ผ่านวงจรฯ ชุดที่ 2 (พอร์ต EXP Display) ที่ความยาวสาย 8 เมตร

อธิบายรูปที่ 4.26 (ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK จาก 0V ขึ้นไป 4.8V ใช้เวลาประมาณ 5,000ns ส่วนขา DATA มีแรงดันประมาณ 5V ลักษณะสัญญาณมี noise เล็กน้อย

รูปที่ 4.26 (ข) จากรูปสัญญาณ SCK มีแรงดันประมาณ 5V ส่วนสัญญาณ DATA เป็นช่วงเวลาไต่ขึ้นของสัญญาณจาก 0V ขึ้นไป 4.8V ใช้เวลาประมาณ 5,000ns ลักษณะสัญญาณมี noise เล็กน้อย

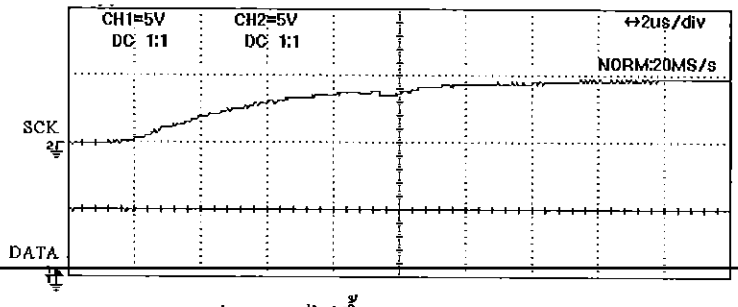
รูปที่ 4.26 (ค) ช่วงเวลาตกของสัญญาณ SCK จาก 5V ลงมา 0V ใช้เวลาประมาณ 0.8μs ส่วนขา DATA ช่วงเวลาตกของสัญญาณจาก 5V ลงมา 0.2V ใช้เวลาประมาณ 0.8μs ลักษณะสัญญาณมี noise เล็กน้อย

- สายสัญญาณยาว 50 เมตร วัดค่าอุณหภูมิได้ 25.5 องศาเซลเซียส วัดค่าความชื้นสัมพัทธ์ได้ 79 เปอร์เซ็นต์

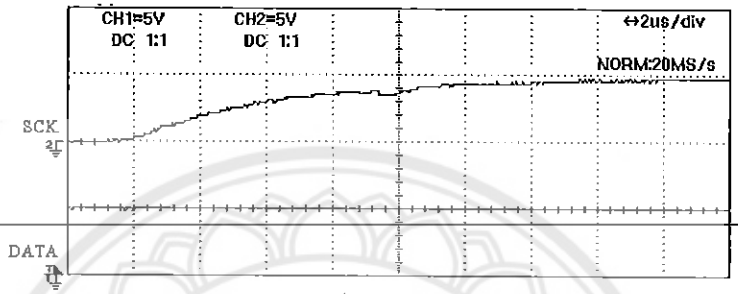


รูปที่ 4.27 กราฟแสดงผลค่าความชื้นสัมพัทธ์และค่าอุณหภูมิ ที่สายสัญญาณยาว 50 เมตร

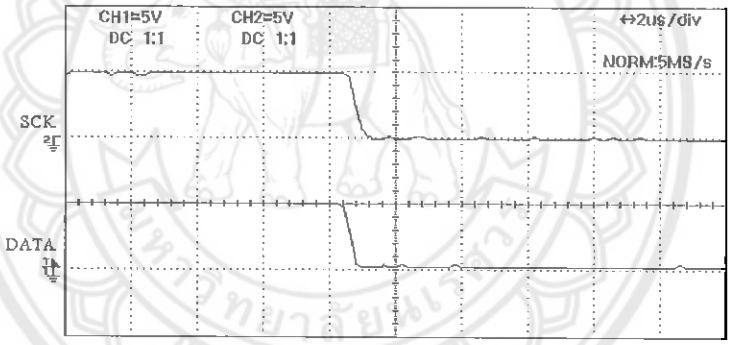
อธิบายรูปที่ 4.27 ภาพรวมของสัญญาณ โดยรวมแล้วสัญญาณ SCK และ DATA มีลักษณะเป็นแท่งสี่เหลี่ยม ส่วนภาพด้านล่างขยายเพื่อให้เห็นความชันของขาสัญญาณ จะเห็นว่า สัญญาณ SCK มีค่า 5V ซึ่งเป็นไปตามความต้องการของ Chip SHT15 ส่วนสัญญาณ DATA ใช้เวลาไต่ขึ้น – ลงระหว่าง 0 – 5V น้อยมากๆ จึงทำให้กราฟมีความชันมาก ส่งผลให้สามารถส่งข้อมูลระหว่าง SHT15 กับไมโครคอนโทรลเลอร์ ได้ถูกต้อง



(ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK



(ข) ช่วงเวลาไต่ขึ้นของสัญญาณ DATA



(ค) ช่วงเวลาตกลงของสัญญาณ SCK และ DATA

รูปที่ 4.28 กราฟแสดงค่า t_r และ t_f ผ่านวงจรฯ ชุดที่ 2 (พอร์ต EXP Display) ที่ความยาวสาย 50 เมตร

อธิบายรูปที่ 4.28 (ก) ช่วงเวลาไต่ขึ้นของสัญญาณ SCK จาก 0V ขึ้นไป 4.8V ใช้เวลาประมาณ 9.8μs ส่วนขา DATA มีแรงดันประมาณ 5V ลักษณะสัญญาณมี noise เล็กน้อย

รูปที่ 4.28 (ข) จากรูปสัญญาณ SCK มีแรงดันประมาณ 0V ส่วนสัญญาณ DATA เป็นช่วงเวลาไต่ขึ้นของสัญญาณจาก 0V ขึ้นไป 5V ใช้เวลาประมาณ 10.2μs ลักษณะสัญญาณมี noise เล็กน้อย

รูปที่ 4.28 (ค) ช่วงเวลาตกของสัญญาณ SCK จาก 5V ลงมา 0V ใช้เวลาประมาณ 1.2μs ส่วนขา DATA ช่วงเวลาตกของสัญญาณจาก 5V ลงมา 0.2V ใช้เวลาประมาณ 1.2μs ลักษณะสัญญาณมี noise เล็กน้อย

ตารางที่ 4.3 ค่าอุณหภูมิและความชื้นสัมพัทธ์ ตอนที่ 2

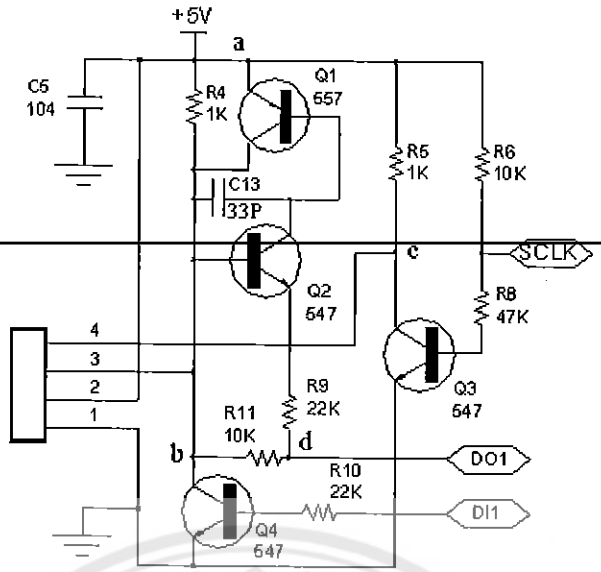
หัววัด SHT15 รับ – ส่งสัญญาณผ่าน	ระยะความยาวสายสัญญาณ (m.)	อุณหภูมิ (°C)	ความชื้นสัมพัทธ์ (%)
วงจรขับและรับฯ ในบอร์ด AP-105	0.2	26.0	80
	8	25.3	78
	50	25.5	79
วงจรขับและรับฯ ชุดที่ 1	0.2	26.1	76
	8	25.6	79
	50	25.6	78
วงจรขับและรับฯ ชุดที่ 2	0.2	25.9	80
	8	25.4	79
	50	25.5	79

ตารางที่ 4.4 ช่วงเวลาไต่ขึ้น – เวลาตกของสัญญาณ ตอนที่ 2

ค่าช่วงเวลา	ย่านที่ต้องการ (ns)	ค่าช่วงเวลาต่างๆ ที่ความยาวสายสัญญาณฯ (ns)								
		ผ่านวงจรฯ เดิม			ผ่านวงจรฯ ชุดที่ 1			ผ่านวงจรฯชุดที่ 2		
		0.2m.	8m.	50m.	0.2m.	8m.	50m.	0.2m.	8m.	50m.
T_{R_SCK}	$T_R > 1$	350	2,050	8,800	800	2,350	8,000	800	1,900	8,400
T_{F_SCK}	$T_F > 1$	100	160	360	180	480	900	170	380	600
T_{FO_DATA}	$1 < T_{FO} < 200$	400	900	900	200	500	600	400	400	500
T_{RO_DATA}	$T_{RO} < 435.5$	350			320			400		
	$T_{RO} < 17{,}333$		2,200			2,100			1,800	
	$T_{RO} < 108{,}335$			6,800			6,200			7,200

4.2.6 วิเคราะห์ผลการทดลองและอธิบายวงจรตอนที่ 2

จากผลการทดลองตอนที่ 2 พบว่า ที่ความยาวสายสัญญาณ 0.20, 8, 50 เมตร สามารถอ่านค่าความชื้นสัมพัทธ์และค่าอุณหภูมิได้ใกล้เคียงกันมาก และเมื่อวิเคราะห์จากตารางผลการทดลองตอนที่ 2 พบว่าที่ความยาวสายสัญญาณ 0.20, 8, 50 เมตร T_{RO} อยู่ในย่านที่ต้องการ



รูปที่ 4.29 วงจรขับและรับสัญญาณพัลส์.

จากการทำงานของวงจรฯ แสดงว่า ถ้าค่า RC ในสายส่งสัญญาณมากส่งผลให้แรงดันในสายส่งตก ทำให้ค่าแรงดันที่รับจากเซ็นเซอร์ มาที่จุด b มีค่าน้อยมาก แต่มีแรงดันที่แหล่งจ่ายไฟ 5 โวลต์จากวงจรฯ ทำให้จุด b มีแรงดันเพิ่มขึ้นเกือบ 5 โวลต์ Q4 ยังคงสถานะเป็น OFF โดยไมโครคอนโทรลเลอร์ จึงมีกระแสไหลเข้าขาเบสของ Q2 ทำให้ Q2 เปลี่ยนสถานะจาก OFF เป็น ON และทำหน้าที่ไบแอส Q1 ให้นำกระแส ส่งผลให้ความต้านทานช่วง จุด a – จุด b มีค่าต่ำมาก แรงดันตกคร่อมจึงมีน้อย ทำให้แรงดันที่จุด d ถูกดึงขึ้นเป็น 5V ด้วยความรวดเร็ว ผ่าน Q1 และ R11 ส่งผลให้ค่า t_r มีค่าน้อยมาก

แต่ถ้า DATA รับข้อมูลจากเซ็นเซอร์ เป็นบิต 0 แรงดันที่จุด b เป็น 0 โวลต์ ทำให้ Q4 คงสถานะเป็น OFF โดยไมโครคอนโทรลเลอร์ เมื่อไม่มีแรงดันที่ขาเบสของ Q2 ทำให้ Q2 และ Q1 มีสถานะเป็น OFF ค่าความต้านทานช่วง จุด a – จุด b เพิ่มสูงขึ้น ทำให้แรงดันที่ตกคร่อมช่วงนั้นมีค่ามาก ทำให้กระแสที่จุด d ถูกดึงลงสู่ 0 อย่างรวดเร็ว ส่งผลให้ค่า t_r มีค่าน้อยมาก

บทที่ 5
สรุปการดำเนินงาน

5.1 สรุปการดำเนินงาน

จากผลการทดลองพบว่า การทำงานของหัววัด SHT15 แบบผ่านวงจรขับและรับสัญญาณพัลส์มีประสิทธิภาพดีกว่าแบบไม่ผ่านวงจรฯ โดยการทำงานของบอร์ดวัดอุณหภูมิและความชื้น แบบที่ไม่มีวงจรฯ ไม่สามารถวัดค่าอุณหภูมิและความชื้นสัมพัทธ์ได้ ถ้าระยะความยาวสายสัญญาณมากกว่า 15 เมตร ส่วนแบบที่ผ่านวงจรฯ ทั้ง 3 ชุด สามารถวัดอุณหภูมิและความชื้นสัมพัทธ์ได้ใกล้เคียงกันมากที่ระยะความยาวสายสัญญาณ 1 – 50 เมตร ดังนั้นสรุปได้ว่าวงจรที่สร้างเพิ่มจำนวน 2 ชุด สามารถนำไปใช้งานได้จริง โดยมีค่าอุณหภูมิและความชื้นใกล้เคียงกับค่าที่วัดจากวงจรเดิมในบอร์ดฯ

5.2 ปัญหาที่เกิดจากการทดลอง

5.2.1 ขณะทำแผ่นวงจร ตัวเก็บประจุ 30 พิโคฟารัด หาซื้อไม่ได้ จึงใช้ ตัวเก็บประจุขนาด 33 พิโคฟารัดแทน ทำให้ช่วงเวลาตกของขา DATA (t_{fo}) กว้างขึ้น ไม่เป็นตามตามเงื่อนไขของหัววัดฯ SHT 15 แต่ยังคงสามารถวัดค่าอุณหภูมิและความชื้นได้

5.2.2 การโหลด Code ลงบอร์ดใช้เวลานาน และต้องเปลี่ยนพอร์ตไปตามการทดลอง ทำให้การวัดอุณภูมิและความชื้นแต่ละครั้งอาจคลาดเคลื่อน เนื่องจากไม่มีสถานที่ที่สามารถควบคุมอุณหภูมิหรือทำให้อุณหภูมิคงที่ได้

5.3 แนวทางการแก้ไขปัญหาและข้อเสนอแนะ

5.3.1 ควรใช้ตัวเก็บประจุขนาด 30 พิโคฟารัด หรือน้อยกว่าเล็กน้อย มาใส่ในวงจรแทนตัวเก็บประจุ 33 พิโคฟารัด เพราะจะทำให้ช่วงเวลาตกของ DATA (t_{fo}) แคบลงและเป็นไปตามเงื่อนไขของหัววัดฯ SHT 15 ซึ่งจะทำให้สัญญาณพัลส์ที่ได้นั้นสมบูรณ์มากขึ้น

5.3.2 หาสถานที่ที่สามารถควบคุมหรือมีอุณหภูมิคงที่ เพื่อให้ผลการวัดอุณหภูมิและความชื้นที่ได้มีความแม่นยำขึ้น

5.4 แนวทางในการพัฒนาเพิ่ม

5.4.1 ออกแบบลายวงจรให้มีขนาดแผ่นทองแดงเล็กลง

5.4.2 พัฒนาโปรแกรมให้รวมการวัดอุณหภูมิและความชื้นทั้ง 3 หัว ในโปรแกรมเดียว เพื่อความสะดวกในการวัด

เอกสารอ้างอิง

[1] นภัทร วัจนเทพินทร์. ทฤษฎีและการออกแบบวงจรพัลซ์. พิมพ์ครั้งที่ 1. กรุงเทพฯ : บริษัทสยามสปอร์ต ซินดิเคท จำกัด. 2538

[2] นคร ภัคดีชาติ, ชัยวัฒน์ ลิ้มพรจิตรวิไล. ทดลองและใช้งานไมโครคอนโทรลเลอร์ MCS-51 ด้วยโปรแกรมภาษา C ฉบับ P89V51RD2. พิมพ์ครั้งที่ 1. กรุงเทพฯ : บริษัทอินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด.2521

[3] ประจิน พลังสันติกุล, ชัยวัฒน์ ลิ้มพรจิตรวิไล. ปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 กับ Keil C51. กรุงเทพฯ : บริษัทอินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด. 2521

[4] วรพจน์ กรแก้ววัฒนกุล. สนุกกับไมโครคอนโทรลเลอร์ ฉบับ MCS-51. กรุงเทพฯ : บริษัทอินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด.2521

ภาคผนวก

ก. โปรแกรมวัดอุณหภูมิและความชื้นสัมพัทธ์โดยตรง

```
#include <AT89s53.h> //Microcontroller specific library, e.g. port definitions
#include <intrins.h> //Keil library (is used for _nop()_ operation)
#include <math.h>    //Keil library
#include <stdio.h>   //Keil library

typedef union
{ unsigned int i;
  float f;
} value;
enum {TEMP,HUMI};

#define     DATA          P2_1 ; Port GPIO
#define     SCK           P2_0 ; Port GPIO

#define noACK 0
#define ACK   1
                                  //adr  command  r/w
#define STATUS_REG_W 0x06         //000   0011    0
#define STATUS_REG_R 0x07         //000   0011    1
#define MEASURE_TEMP 0x03         //000   0001    1
#define MEASURE_HUMI 0x05         //000   0010    1
#define RESET        0x1e         //000   1111    0
//----------------------------------------------------
char s_write_byte(unsigned char value)
//----------------------------------------------------
// writes a byte on the Sensibus and checks the acknowledge
{
 unsigned char i,error=0;
```

```
 for (i=0x80;i>0;i/=2)             //shift bit for masking
 { if (i & value) DATA=1;          //masking value with i , write to SENSI-BUS
  else DATA=0;
  SCK=1;                           //clk for SENSI-BUS
  _nop_();_nop_();_nop_();         //pulswith approx. 5 us
  SCK=0;
 }
 DATA=1;                           //release DATA-line
 SCK=1;                            //clk #9 for ack
 error=DATA;                       //check ack (DATA will be pulled down by SHT11)
 SCK=0;
 return error;                     //error=1 in case of no acknowledge
}
char s_read_byte(unsigned char ack)
// reads a byte form the Sensibus and gives an acknowledge in case of "ack=1"
{
 unsigned char i,val=0;
 DATA=1;                           //release DATA-line
 for (i=0x80;i>0;i/=2)             //shift bit for masking
 { SCK=1;                          //clk for SENSI-BUS
  if (DATA) val=(val | i);         //read bit
  SCK=0;
 }
 DATA=!ack;                        //in case of "ack==1" pull down DATA-Line
 SCK=1;                            //clk #9 for ack
 _nop_();_nop_();_nop_();          //pulswith approx. 5 us
 SCK=0;
 DATA=1;                           //release DATA-line
 return val;
}
//--------------------------------------------------------------------------
void s_transstart(void)
```

```
// generates a transmission start
{
  DATA=1; SCK=0;                    //Initial state
  _nop_();
  SCK=1;
  _nop_();
  DATA=0;
  _nop_();
  SCK=0;
  _nop_();_nop_();_nop_();
  SCK=1;
  _nop_();
  DATA=1;
  _nop_();
  SCK=0;
}
//---------------------------------------------------------------------------
void s_connectionreset(void)
// communication reset: DATA-line=1 and at least 9 SCK cycles followed by transstart
{
 unsigned char i;
 DATA=1; SCK=0;                    //Initial state
 for(i=0;i<9;i++)                  //9 SCK cycles
 { SCK=1;
  SCK=0;
 }
 s_transstart();                   //transmission start
}
//---------------------------------------------------------------------------
char s_softreset(void)
// resets the sensor by a softreset
{
```

```
 unsigned char error=0;
 s_connectionreset();            //reset communication
 error+=s_write_byte(RESET);     //send RESET-command to sensor
 return error;                   //error=1 in case of no response form the sensor
}
//-----------------------------------------------------------------------------
char s_read_statusreg(unsigned char *p_value, unsigned char *p_checksum)
// reads the status register with checksum (8-bit)
{
 unsigned char error=0;
 s_transstart();                 //transmission start
 error=s_write_byte(STATUS_REG_R); //send command to sensor
 *p_value=s_read_byte(ACK);      //read status register (8-bit)
 *p_checksum=s_read_byte(noACK); //read checksum (8-bit)
 return error;                   //error=1 in case of no response form the sensor
}
//-----------------------------------------------------------------------------
char s_write_statusreg(unsigned char *p_value)
// writes the status register with checksum (8-bit)
{
 unsigned char error=0;
 s_transstart();                 //transmission start
 error+=s_write_byte(STATUS_REG_W);//send command to sensor
 error+=s_write_byte(*p_value);  //send value of status register
 return error;                   //error>=1 in case of no response form the sensor
}
//-----------------------------------------------------------------------------
char s_measure(unsigned char *p_value, unsigned char *p_checksum, unsigned char mode)
// makes a measurement (humidity/temperature) with checksum
{
 unsigned error=0;
 unsigned int i;
```

```
s_transstart();                 //transmission start
switch(mode){                   //send command to sensor
  case TEMP     : error+=s_write_byte(MEASURE_TEMP); break;
  case HUMI     : error+=s_write_byte(MEASURE_HUMI); break;
  default     : break;
}
for (i=0;i<65535;i++) if(DATA==0) break; //wait until sensor has finished the measurement
if(DATA) error+=1;                      // or timeout (~2 sec.) is reached
*(p_value) =s_read_byte(ACK);           //read the first byte (MSB)
*(p_value+1)=s_read_byte(ACK);          //read the second byte (LSB)
*p_checksum =s_read_byte(noACK);        //read checksum
return error;
}
//-----------------------------------------------------------------------
void init_uart()
//9600 bps @ 11.059 MHz
{SCON  = 0x52;
 TMOD  = 0x20;
 TCON  = 0x69;
 TH1   = 0xfd;
}
//-----------------------------------------------------------------------
void calc_sth11(float *p_humidity ,float *p_temperature)
// calculates temperature [℃] and humidity [%RH]
// input :  humi [Ticks] (12 bit)
//          temp [Ticks] (14 bit)
//          output:  humi [%RH]
//          temp [℃]
{ const float C1=-4.0;              // for 12 Bit
  const float C2=+0.0405;           // for 12 Bit
  const float C3=-0.0000028;        // for 12 Bit
  const float T1=+0.01;             // for 14 Bit @ 5V
```

```
const float T2=+0.00008;        // for 14 Bit @ 5V

float rh=*p_humidity;           // rh:     Humidity [Ticks] 12 Bit
float t=*p_temperature;         // t:      Temperature [Ticks] 14 Bit
float rh_lin;                   // rh_lin: Humidity linear
float rh_true;                  // rh_true: Temperature compensated humidity
float t_C;                      // t_C   : Temperature [℃]

t_C=t*0.01 - 40;                //calc. temperature from ticks to [℃]
rh_lin=C3*rh*rh + C2*rh + C1;   //calc. humidity from ticks to [%RH]
rh_true=(t_C-25)*(T1+T2*rh)+rh_lin;  //calc. temperature compensated humidity [%RH]
if(rh_true>100)rh_true=100;     //cut if the value is outside of
if(rh_true<0.1)rh_true=0.1;     //the physical possible range

*p_temperature=t_C;             //return temperature [℃]
*p_humidity=rh_true;            //return humidity[%RH]
}
//--------------------------------------------------------------------
float calc_dewpoint(float h,float t)
// calculates dew point
// input:   humidity [%RH], temperature [℃]
// output:  dew point [℃]
{ float logEx,dew_point;
 logEx=0.66077+7.5*t/(237.3+t)+(log10(h)-2);
 dew_point = (logEx - 0.66077)*237.3/(0.66077+7.5-logEx);
 return dew_point;
}
//--------------------------------------------------------------------------------
void main()
//--------------------------------------------------------------------------------
// sample program that shows how to use SHT11 functions
// 1. connection reset
```

```
// 2. measure humidity [ticks](12 bit) and temperature [ticks](14 bit)
// 3. calculate humidity [%RH] and temperature [ฐC]
// 4. calculate dew point [ฐC]
// 5. print temperature, humidity, dew point
-------------------------------------------------------------------------------
{ value humi_val,temp_val;
 float dew_point;
 unsigned char error,checksum;
 unsigned int i;
 init_uart();
 s_connectionreset();
 while(1)
 { error=0;
  error+=s_measure((unsigned char*) &humi_val.i,&checksum,HUMI);  //measure humidity
  error+=s_measure((unsigned char*) &temp_val.i,&checksum,TEMP);  //measure
temperature
  if(error!=0) s_connectionreset();              //in case of an error: connection reset
  else
  { humi_val.f=(float)humi_val.i;                //converts integer to float
   temp_val.f=(float)temp_val.i;                 //converts integer to float
   calc_sth11(&humi_val.f,&temp_val.f);          //calculate humidity, temperature
   dew_point=calc_dewpoint(humi_val.f,temp_val.f); //calculate dew point
   printf("temp:%5.1fC humi:%5.1f%% dew
point:%5.1fC\n",temp_val.f,humi_val.f,dew_point);
  }
  //----------wait approx. 0.8s to avoid heating up SHTxx----------------------------
  for (i=0;i<40000;i++);    //(be sure that the compiler doesn't eliminate this line!)
 }
}
```

ข. โปรแกรมวัดอุณหภูมิและความชื้นสัมพัทธ์ โดยผ่านวงจรขับและรับสัญญาณพัลส์

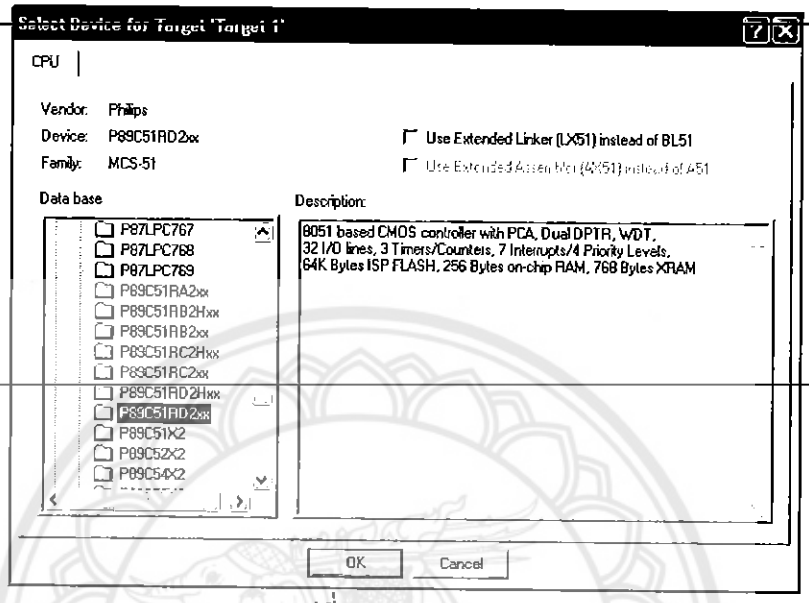
กรุณาติดต่อที่

คุณเจริญ บัวเทศ ผู้จัดการ ห้างหุ้นส่วนจำกัดเซมดีไซน์ หรือ

ดร.อัครพันธ์ วงศ์กังแห อาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
มหาวิทยาลัยนเรศวร

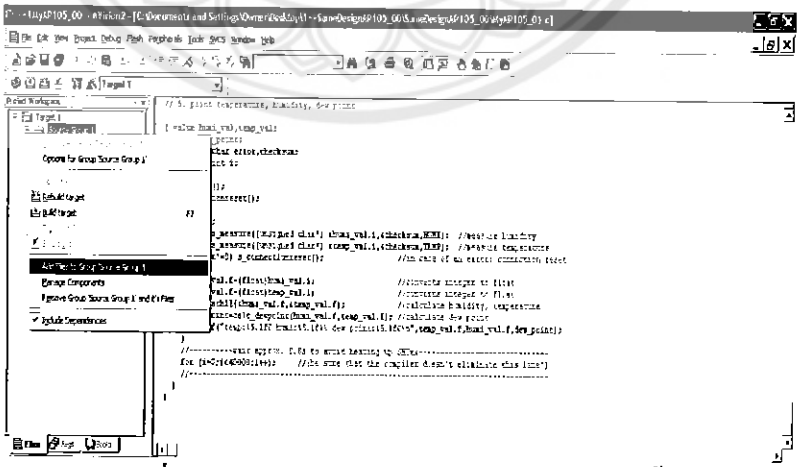
ค. วิธีการโหลดโปรแกรมลงบอร์ด AP-105 V1.0

1. เปิดโปรแกรม Keil uVision 2, สร้างไฟล์โปรเจคและเลือก CPU ของ Philips-P89C51RD2xx ที่จะใช้ จาก device database ดังรูปที่ 1



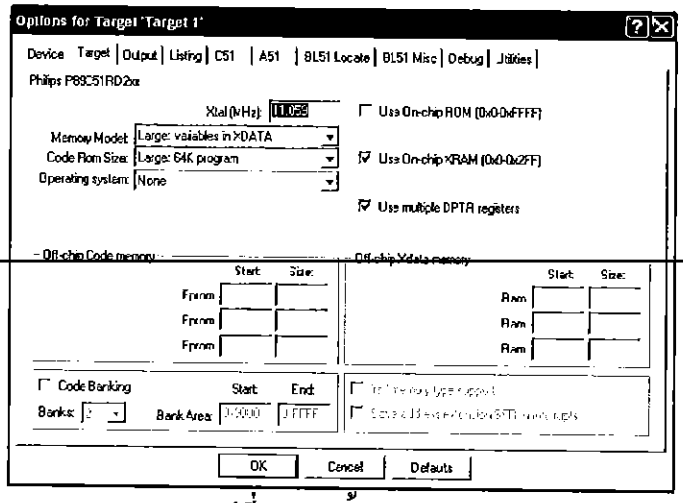
รูปที่ 1 Select device

2. เขียนโปรแกรมภาษาซี ลงหน้าต่างที่ว่างๆ เมื่อพิมพ์เสร็จแล้วเลือกบันทึกไฟล์โดยเลือกนามสกุลไฟล์เป็น ".C" เช่น AP105_01.C แล้ว Source file ใส่ลงไปในไฟล์ที่สร้างขึ้นในข้อที่ 1 โดยคลิกขวาที่ Source Group > Add File to Group "Source Group" เลือก AP105_01.C ดังรูปที่ 2



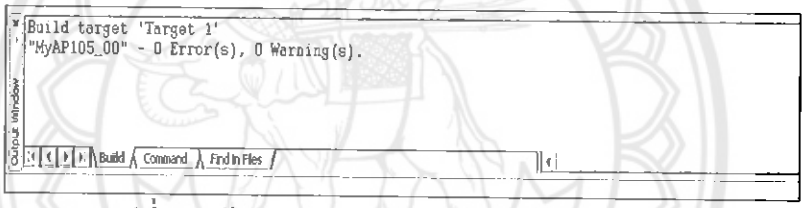
รูปที่ 2 การ Source file ใส่ลงไปในไฟล์ที่สร้างขึ้น

3. ตั้งค่า Target โดยคลิกที่ Project > Options for Target 'Target 1' > Target ดังรูปที่ 3



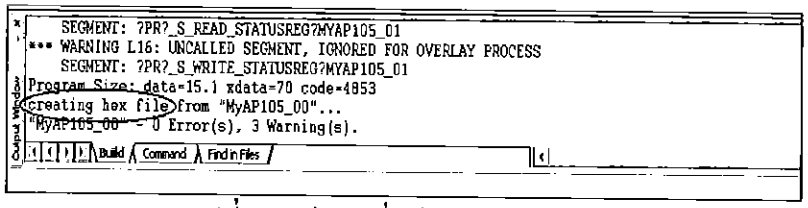
รูปที่ 3 การตั้งค่า Target

4. ตรวจสอบโปรแกรมว่าถูกต้องหรือเปล่า โดยคลิกที่ Build Target ถ้าไม่มีข้อผิดพลาดจะปรากฏหน้าต่างดังนี้



รูปที่ 4 หน้าต่างตรวจสอบข้อผิดพลาดของโปรแกรม

5. สร้าง HEX File คลิกที่ Project > Options for Target 'Target 1' > Output แล้วคลิกที่ Create HEX File เสร็จแล้วทำการ Build Target อีกครั้ง จะได้ HEX File ดังรูปที่ 5 สำหรับโหลดลงบอร์ด AP-105 V1.0



รูปที่ 5 หน้าต่างที่สร้าง HEX File แล้ว

6. ต่อสายสัญญาณ RS232 ระหว่างคอมพิวเตอร์กับบอร์ด AP-105v1.0 และจ่ายไฟให้กับบอร์ดเพื่อให้พร้อมรับคำสั่ง

7. เปิดโปรแกรม Flash Magic เพื่อโหลดโปรแกรมลงไมโครคอนโทรลเลอร์

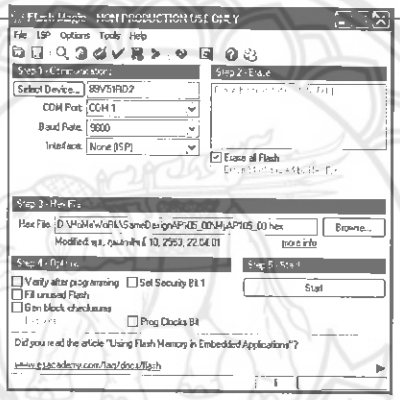
8. Step 1 ตั้งค่า : Communication ดังรูปที่ 6 ดังนี้

- เลือก Device ตามการใช้งานจริง คือเลือก P89V51RD2
- เลือก Comport ตามที่ต่อสายไว้จริง คือเลือก Com 1
- เลือก Baud rate เป็น 9600
- เลือก interface เป็น None (ISP)

9. Step 2 ตั้งค่า : Eraser โดยเลือกกำหนด รูปแบบการลบข้อมูล ซึ่งถ้าไม่แน่ใจว่า CPU ถูก Lock ไว้หรือไม่ ให้เลือก Erase All Flash ดังรูปที่ 6

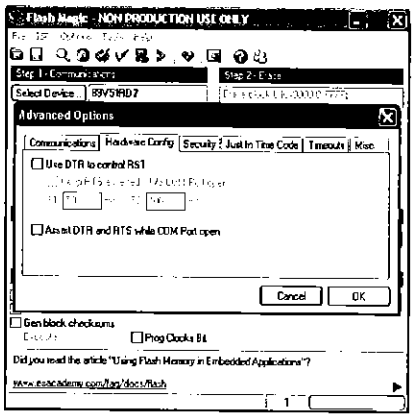
10. Step 2 คลิกที่ Browse... เลือก Hex File ที่ต้องการดาวน์โหลด ดังรูปที่ 6

11. Step 4 ตั้งค่า : Option เลือก Verify After Programming ดังรูปที่ 6



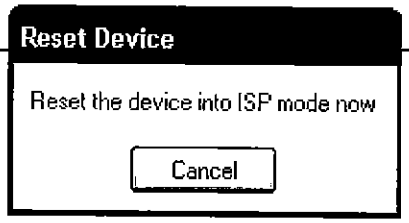
รูปที่ 6 การตั้งค่า Flash Magic

12. ตั้งค่า Option ของโปรแกรม โดยคลิกที่ Options > Advance Options > Hardware Config > คลิกเครื่องหมาย ✓ หน้า Use DTR to control RST



รูปที่ 7 การตั้งค่า Option

13. คลิกที่ Start เพื่อสั่ง load ข้อมูลให้กับ CPU ซึ่งจะปรากฏหน้าต่างบอกให้ RESET การทำงานของไมโครคอนโทรลเลอร์ ให้เริ่มต้นทำงานใน ISP Mode ดังรูปที่ 7



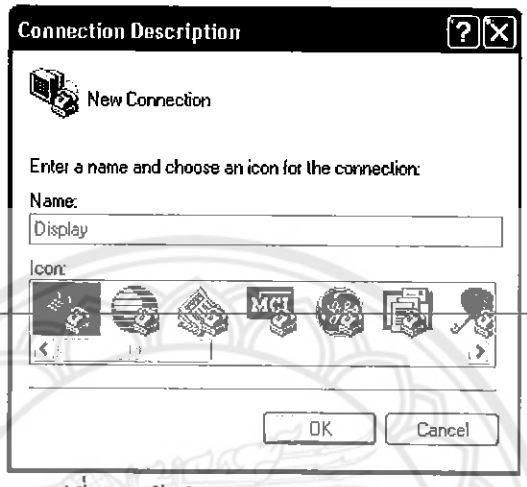
รูปที่ 8 หน้าต่างให้ RESET การทำงานของไมโครคอนโทรลเลอร์

14. ให้ทำการกดสวิตช์ RESET บนบอร์ด AP-105v1.0 หรือถอดปลั๊กแล้วเสียบปลั๊กเข้าไปใหม่ ซึ่งหน้าต่างข้างต้นจะหายไป และรอโปรแกรมโหลดลงบอร์ดจนเสร็จ

15. ให้ทำการกดสวิตช์ RESET บนบอร์ด AP-105v1.0 อีกครั้ง หรือถอดปลั๊กแล้วเสียบปลั๊กเข้าไปใหม่ เพื่อให้ไมโครคอนโทรลเลอร์ เริ่มต้นทำงานตามโปรแกรมที่โหลดไปแล้วทันที

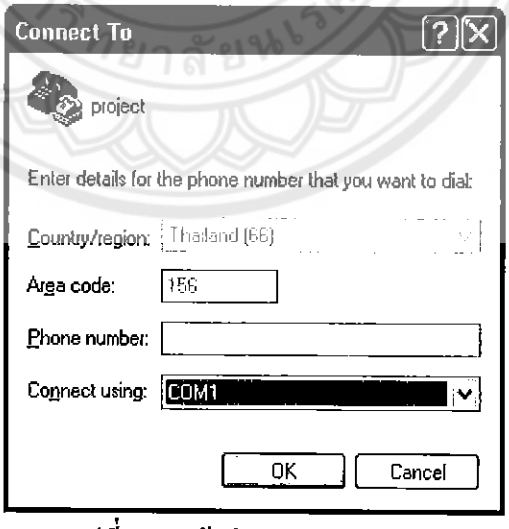
ง. การแสดงผลอุณหภูมิและความชื้นสัมพัทธ์ ที่ได้ออกทางหน้าจอคอมพิวเตอร์โดยใช้ Program HyperTerminal

1. ไปที่ Start > Program > Accessories > Communications เลือก HyperTerminal
2. ให้ตั้งชื่อ เช่น Display ในช่อง Name แล้วคลิก OK ดังรูปที่ 9



รูปที่ 9 หน้าต่าง Connection Description

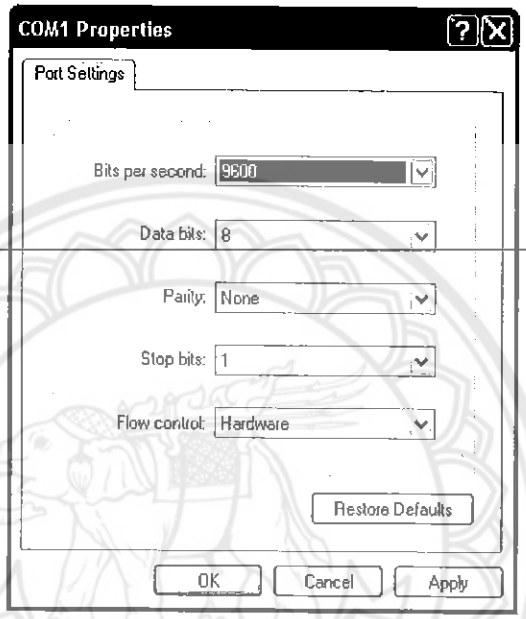
3. จะปรากฏหน้าต่าง Connection To เลือก COM1 ในช่อง Connect using ตามด้วย OK ดังรูปที่ 10



รูปที่ 10 หน้าต่าง Connection To

4. จะปรากฏหน้าต่าง Com1 Properties ขึ้นเพื่อกำหนดค่าต่างๆตามนี้
   - ในช่อง Bit per second เลือก 9600

- ในช่อง Data bits เลือก 8
- ในช่อง Parity เลือก None
- ในช่อง Stop bit เลือก 1
- ในช่อง Flow Control เลือก Hardware

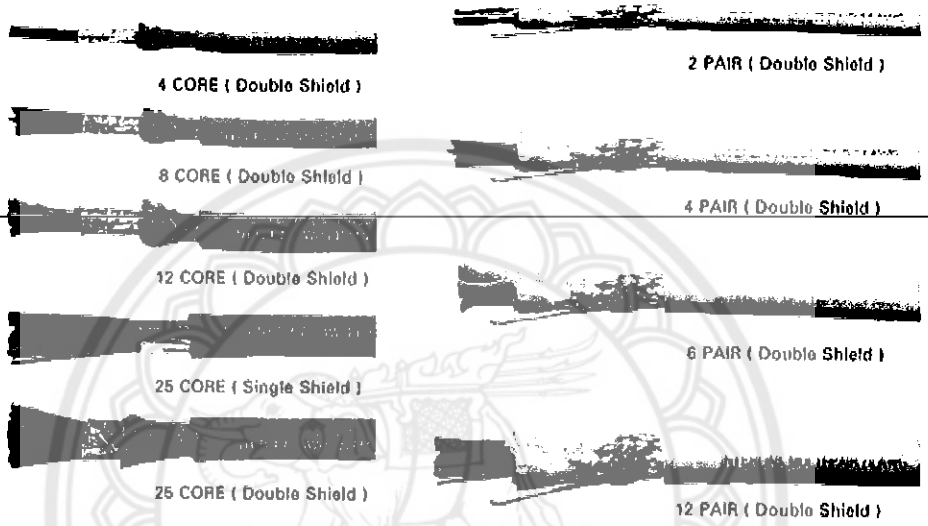


รูปที่ 11 การตั้งค่า COM1

5. ค่าอุณหภูมิและค่าความชื้นสัมพัทธ์ที่วัดได้ จะแสดงผลที่หน้าต่างของ Display - Hyper Terminal หากจะหยุดการเชื่อมต่อให้กด Disconnect

จ. ค่า Capacitor ในสายสัญญาณชนิด Multicore Cable

CCTV, TV & MULTICORE



UL-2464 MULTI - CORE CABLE

APPLICATIONS FOR RS-423, RS -232C HEAVY-DUTY

UL-2919 MULTI - PAIR CABLE

APPLICATIONS FOR RS-423, RS-422 or RS-485

UL (STYLE -2464) or (LSTYLE-2919, CSA , AWM PASSES THE VW-1 VERTICAL WIRE FLAME TEST
24 AWG (7/0.2 MM STRANDED TINNED ANNEALED COPPER CONDUCTOR) PVC INSULATED,
ALUMINIUM/POLYESTER OVERALL FOIL PLUS TINNED COPPER BRAID 65% SHIELD

UL Style & CSA Type	Conductor		No. of Core or Pair	Insulation Thickness	Braid Shield	Jacket Thickness	Overall Diameter	Nominal Capa.	
								A	B
	AWG	No./mm		mm.	No./mm	mm.	mm.	pF/ft	pF/ft
UL 2464 CSA LIIA (AWM) (Double Shield)	24	7/0.203	4C	0.25	16/5/0.127	0.80	5.1±0.15	35	65
			6C	0.25	16/8/0.127	0.80	6.2±0.15	30	55
			12C	0.25	24/8/0.127	0.85	7.4±0.20	30	55
			25C	0.25	24/8/0.127	1.00	9.1±0.25	30	55
(Single Shield)	24	7/0.203	25C	0.25	-	1.00	8.6±0.20	30	55
UL 2919 CSA LIIA (AWM) (Double Shield)	24	7/0.203	2P	0.25	16/5/0.127	0.80	7.2±0.15	16	52.5
			4P	0.25	16/8/0.127	1.00	8.2±0.15	16	52.5
			6P	0.25	24/8/0.127	1.10	9.2±0.20	16	52.5
			12P	0.25	24/8/0.127	1.30	11.5±0.25	16	52.5

ประวัติผู้เขียนโครงงาน

ชื่อ นางสาวจงรัก สกุลนา

ภูมิลำเนา 110/6 ม.4 ต.วังแดง อ.ตรอน จ.อุตรดิตถ์

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนตรอนตรีสินธุ์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : **som_jongrak@hotmail.com**

ชื่อ นางสาววราพร ไวเปีย

ภูมิลำเนา 54 ม.4 ต.หาดกรวด อ.เมือง จ.อุตรดิตถ์

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนอุตรดิตถ์ดรุณี
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : **aloas4_ee@hotmail.com**