สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องควบคุมอุปกรณ์ไฟฟ้าอัตโนมัติตามสภาพแวดล้อม

AUTOMATIC CONTROL ELECTRIC EQUIPMENT BY ENVIRONMENT

โดย

นายจักรกริช มูลดี

นายณัฐชัย มิ่งเมืองชัย

นายสุรเชษฐ์ สุจริต

วศ.
จ211ค
2548

เลขหมู่.....
เลขทะเบียน 62371
วัน,เดือน,ปี 16 ส.ค. 2549

b 11621283
i

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

ผ่านการตรวจชิ้นงานแล้ว

(ลงชื่อ).....ผู้ตรวจ

ผ่านการตรวจรูปเล่มแล้ว

(ลงชื่อ).....ผู้ตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมอุปกรณ์ไฟฟ้าอัตโนมัติตามสภาพแวดล้อม

AUTOMATIC CONTROL ELECTRIC EQUIPMENT BY ENVIRONMENT

โดย

นายจักรกริช มูลดี 46015045

นายณัฐชัย มิ่งเมืองชัย 46015050

นายสุรเชษฐ์ สุจริต 46015075

อาจารย์ที่ปรึกษา

อาจารย์ธเนศ พัฒนธาดาพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **เครื่องควบคุมอุปกรณ์ไฟฟ้าอัตโนมัติตามสภาพแวดล้อม**

AUTOMATIC CONTROL ELECTRIC EQUIPMENT BY ENVIRONMENT

ผู้จัดทำ

1. นายจักรกริช	**มูลดี**	**46015045**
2. นายณัฐชัย	**มิ่งเมืองชัย**	**46015050**
3. นายสุรเชษฐ์	**สุจริต**	**46015075**

..... อาจารย์ที่ปรึกษา

(อ. ธเนศ พัฒนาธาดาพงษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมอุปกรณ์ไฟฟ้าอัตโนมัติตามสภาพแวดล้อม

Automatic control electric equipment by environment

โดย นายจักรกริช มูลดี 46015045

นายณัฐชัย มิ่งเมืองชัย 46015050

นายสุรเชษฐ์ สุจริต 46015075

อาจารย์ที่ปรึกษา .....

(อ. ธเนศ พัฒนธาดาพงษ์)

บทคัดย่อ

โครงงานนี้เป็นการศึกษาและประดิษฐ์อุปกรณ์ที่ใช้ในการเก็บข้อมูลของสภาพแวดล้อมระยะไกล อันได้แก่ อุณหภูมิ ปริมาณน้ำฝน ความชื้น ความดัน ลม และแสงแดด ผ่านระบบสื่อสารไร้สาย โดยข้อมูลที่วัดได้จากการใช้เซนเซอร์ จะถูกส่งผ่านอุปกรณ์ส่งสัญญาณแบบไร้สายไปยังเครื่องรับโดยในส่วนของเครื่องรับ จะใช้จอแอลซีดีในการแสดงผลของข้อมูลต่างๆที่รับมาและผลที่ได้นั้นสามารถนำไปควบคุมอุปกรณ์ไฟฟ้าเช่นให้ทำงานได้เองโดยอัตโนมัติ หรือนำไปใช้ประยุกต์ใช้ได้กับการเกษตร โรงงานอุตสาหกรรม และที่พักอาศัย

Abstract

This project presents the study of electronic controlled equipment by environment. This equipment will collect the data from environment such as temperature, rainfall, humidity, pressure, windy and sunlight. Data from sensor will be sent through wireless communication equipment and show measurement data by LCD monitor at receiver. Finally, control electronic device. The result of this project will be used to control electronic equipment such as agricultural equipment or other device in industry, company and residence.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณ อ.ธเนศ พัฒนธาดาพงษ์ อาจารย์ที่ปรึกษา ที่ค่อยอำนวยความสะดวกในการทดลองพร้อมทั้งสนับสนุนเครื่องมือทุกอย่างเท่าที่อาจารย์จะเพิ่งมีและหาให้ได้ และความเป็นกันเองที่ทำให้นักศึกษาไม่ต้องกังวลว่าจะโดนดุว่ากล่าว เพราะบางอย่างเราก็ไม่ทราบด้วยความสัตย์จริง พวกเราในนามเด็กสร้างของอาจารย์รู้สึกดีทุกครั้งที่เราทำงานล่าช้าหรือว่าบกพร่องในบางจุดว่าเราจะไม่โดนอาจารย์ว่ากล่าวแบบเสียๆ หายๆ เพราะอาจารย์เป็นบุคคลที่พร้อมจะให้คำปรึกษาที่เราจะต้องประหลาดใจว่าทำไมอาจารย์ไม่ว่าเราเลย แต่กลับให้ทางออกกับเราเสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทที่ 1	บทนำ	1
บทที่ 2	ทฤษฎีและหลักการ	2
	2.1 อุปกรณ์ตรวจสอบสภาพแวดล้อม	2
	2.1.1 อุณหภูมิและความชื้น	3
	2.1.2 ความดัน	4
	2.1.3 ลม	5
	2.1.4 น้ำฝน	6
	2.1.5 แสงสว่าง	7
	2.2 ภาคประมวลผลความถี่ มัลติเพล็กซ์และดีมัลติเพล็กซ์	8
	2.2.1 Microcontroller	9
	2.2.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์	9
	2.2.1.2 โครงสร้างภายในของไมโครคอนโทรลเลอร์	10
	2.2.1.3 การจัดขาของไมโครคอนโทรลเลอร์	11
	2.2.2 การมัลติเพล็กซ์โดยใช้ไมโครคอนโทรลเลอร์	13
	2.3 ภาครับ-ส่งสัญญาณ RF	14
	2.3.1 การทำงานแบบ RF Receive Only	15
	2.3.2 การทำงานแบบ RF Transmit Only	15
	2.4 ส่วนแสดงผลผ่านจอแอลซีดี	16
	2.4.1 จังหวะการทำงานของแอลซีดีโมดูล	17
	2.5 ส่วนควบคุมอุปกรณ์ไฟฟ้าโดยอัตโนมัติ	17
	2.5.1 ออปโต้ไอโซเลเตอร์	18
บทที่ 3	การคำนวณและการสร้าง	19
	3.1 อุปกรณ์ตรวจสอบสัญญาณ	19
	3.1.1 อุณหภูมิและความชื้น	19
	3.1.2 ความดัน	21
	3.1.3 ลม	23
	3.1.4 น้ำฝน	24
	3.1.5 แสงสว่าง	26
	3.2 ส่วนนับและประมวลผลความถี่	26
	3.3 ภาคมัลติเพล็กซ์และดีมัลติเพล็กซ์	27
	3.3.1 การมัลติเพล็กซ์โดยใช้ไมโครคอนโทรลเลอร์	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.3.2 การดีมัลติเพล็กซ์โดยใช้ไมโครคอนโทรลเลอร์	28
3.4 เครื่องรับ-ส่งวิทยุ RF	29
3.5 ส่วนควบคุมอุปกรณ์ไฟฟ้า	32
3.6 ส่วนแสดงผลผ่านหน้าจอแอลซีดี	33
บทที่ 4 การทดลองและผลการทดลอง	36
4.1 เซนเซอร์ความดัน	36
4.2 เซนเซอร์ความเร็วลม	37
4.3 เซนเซอร์ปริมาณน้ำฝน	37
4.4 เซนเซอร์ความเข้มแสง	39
บทที่ 5 บทวิจารณ์และบทสรุป	42
ภาคผนวก	
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

บทที่ 2 ทฤษฎีและหลักการ

รูปที่ 2.1 บล็อกไดอะแกรมการทำงานโดยรวม 2

รูปที่ 2.2 โครงสร้างภายในของเซนเซอร์วัดความดัน 4

รูปที่ 2.3 กราฟเปรียบเทียบแรงดันไฟฟ้ากับความดันอากาศจากดาต้าชีท 4

รูปที่ 2.4 บล็อกไดอะแกรมของระบบเซนเซอร์ตรวจสอบความดันอากาศ 5

รูปที่ 2.5 บล็อกไดอะแกรมของระบบเซนเซอร์ตรวจวัดความเร็วลม 5

รูปที่ 2.6 แสดงเอาท์พุตและอินพุทของวงจร Comparator 6

รูปที่ 2.7 บล็อกไดอะแกรมของระบบเซ็นเซอร์ตรวจวัดปริมาณน้ำฝน 6

รูปที่ 2.8 วงจรอะสเตเบิล มัลติไวเบรเตอร์ 7

รูปที่ 2.9 สัญญาณทริกเกอร์แร้มป์ และสัญญาณเอาท์พุต 7

รูปที่ 2.10 บล็อกไดอะแกรมของระบบเซ็นเซอร์ตรวจวัดแสง 7

รูปที่ 2.11 โครงสร้างของ LDR 8

รูปที่ 2.12 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ 9

รูปที่ 2.13 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51 10

รูปที่ 2.14 ตำแหน่งขาของไมโครคอนโทรลเลอร์ตระกูล MCS-51 11

รูปที่ 2.15 แสดงการจัดเฟรมของข้อมูล 13

รูปที่ 2.16 แสดงสายสัญญาณ RS232 เพื่อใช้เครื่องแปลงสัญญาณ ในโหมด RF Receive Only และ RF Transmit Only 16

รูปที่ 2.17 วงจรพื้นฐานของ Opto isolator 18

บทที่ 3 การคำนวณและการสร้าง

รูปที่ 3.1 บล็อกไดอะแกรมของเซ็นเซอร์อุณหภูมิและความชื้น 19

รูปที่ 3.2 แสดงการต่อเซนเซอร์วัดอุณหภูมิกับความชื้น 19

รูปที่ 3.3 แสดงเซนเซอร์ความชื้นและอุณหภูมิ 20

รูปที่ 3.4 วงจรภายในไอซีตรวจสอบแรงดัน 21

รูปที่ 3.5 วงจร Op-Amp ขยายแรงดัน 100 เท่า 21

รูปที่ 3.6 แสดงวงจรโวลท์เตจคอนโทรออสซิลเลเตอร์ที่ใช้ในการสร้างพัลส์ 22

รูปที่ 3.7 วงจรตรวจสอบความเร็วลม 23

รูปที่ 3.8 แผ่นตัดแสง 23

รูปที่ 3.9 แสดงการต่อวงจรเซนเซอร์เข้ากับวงจรอะสเตเบิล 24

รูปที่ 3.10 แสดงแผงเซนเซอร์น้ำฝน 25

รูปที่ 3.11 แสดงช่วงเวลา T-on และ T-off 26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 3.12 การต่อ LDR เข้ากับ IC 555 เพื่อเป็น วงจรอะสเตเบิล มัลติไวเบรเตอร์ 26

รูปที่ 3.13 แสดงการต่อใช้งานในส่วนของการประมวลผลความถี่ 27

รูปที่ 3.14 แสดงการมัลติเพล็กซ์โดยใช้ไมโครคอนโทรเลอร์ 27

รูปที่ 3.15 แสดงการดีมัลติเพล็กซ์โดยใช้ไมโครคอนโทรลเลอร์ 28

รูปที่ 3.16 วงจรเครื่องรับ-ส่งวิทยุ 2.4GHz 29

รูปที่ 3.17 แสดงรูปแบบการตั้งค่าภายในโดยใช้โปรแกรมในการตั้งค่า 30

รูปที่ 3.18 แสดงการต่อวงจรรีเลย์สำหรับการควบคุมอุปกรณ์ไฟฟ้า 32

รูปที่ 3.19 การต่อ Display LCD เข้ากับชุดคอนโทรลบอร์ด 33

รูปที่ 3.20 แสดงโมดูลภาครับ ส่วนควบคุมอุปกรณ์ไฟฟ้าและแสดงผลผ่านหน้าจอแอลซีดี 34

รูปที่ 3.21 แสดงเซนเซอร์ลม น้ำฝน และ แสงแดด 35

รูปที่ 3.22 แสดงโมดูลภาคส่งและส่วนประมวลผลความถี่จากเซนเซอร์ 35

บทที่4 การทดลองและผลการทดลอง

รูปที่ 4.1 แสดงค่าของความถี่เมื่อปริมาณน้ำฝนน้อย 37

รูปที่ 4.2 แสดงค่าของความถี่เมื่อปริมาณน้ำฝนปานกลาง 38

รูปที่ 4.3 แสดงค่าของความถี่เมื่อปริมาณน้ำฝนมาก 38

รูปที่ 4.4 แสดงผลบนหน้าจอแอลซีดี 41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

บทที่ 3 การคำนวณและการสร้าง

ตารางที่ 3.1 Temperature conversion coefficient 20

ตารางที่ 3.2 Humidity conversion coefficient 20

ตารางที่ 3.3 แสดงลักษณะสัญญาณ RS RW และ E 34

บทที่ 4 การทดลองผลการทดลอง

ตารางที่ 4.1 แสดงผลการทดลองความดันอากาศที่อุณหภูมิ 25 องศาเซลเซียส เทียบกับความถี่ 36

ตารางที่ 4.2 แสดงความสัมพันธ์ระหว่างความเร็วลมและความถี่ของสัญญาณเอาท์พุตที่วัดได้ 37

ตารางที่ 4.3 ค่าความเข้มแสงเทียบกับความถี่ที่สร้างขึ้นของวงจรอะสเตเบิล โดย LDR 39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

นักวิจัยและพัฒนาเทคโนโลยีต่างพากันสร้างฝันไว้ว่า อีกไม่กี่ปีข้างหน้านับจากนี้ เทคโนโลยีไร้สายจะสามารถทำให้อุปกรณ์สื่อสารติดต่อสั่งงานกันเองได้โดยมนุษย์ไม่ต้องเข้าไปเกี่ยวข้อง อุปกรณ์อิเล็กทรอนิกส์ที่ว่านี้ จะเชื่อมต่อกันเองด้วยสัญญาณคลื่นวิทยุในความถี่ที่เป็นย่านความถี่สาธารณะที่อนุญาตให้ใช้งานได้อยู่แล้ว อุปกรณ์เหล่านี้อาจจะเป็นเครื่องคอมพิวเตอร์ต่อเชื่อมกับเซ็นเซอร์ที่จะทำงานผ่านระบบเครือข่าย เพื่อสั่งงานโดยอัตโนมัติ

นอกจากนี้ ยังอาจจะเป็นอุปกรณ์ไร้สายที่ติดตั้งอยู่ที่อุปกรณ์เครื่องไฟฟ้า เช่น ระบบเครื่องปรับอากาศ เครื่องทำความร้อน ที่จะเชื่อมต่อคำสั่งกับเครื่องคอมพิวเตอร์ของเจ้าของบ้านที่ทำงานอยู่ที่ทำงาน หรือเชื่อมต่อกับหน่วยรักษาความปลอดภัยของหมู่บ้านหากมีเหตุการณ์ผิดปกติเกิดขึ้นกับตัวบ้านหรืออุปกรณ์เครื่องไฟฟ้าต่างๆ เหล่านี้นำไปใช้ในทางด้านการเกษตรและ อุตสาหกรรมโดยสั่งงานโดยอัตโนมัติเมื่อสภาพแวดล้อม หรือภูมิอากาศ เปลี่ยนแปลง ทำให้ได้ผลผลิตที่เพิ่มมากขึ้น และมีคุณภาพพร้อมสำหรับการส่งออก หรือนำออกจำหน่ายภายในประเทศ

ด้วยแนวคิดเหล่านี้ทำให้เกิดแรงบันดาลใจในการทำโครงงานชิ้นนี้ขึ้นมาเพื่อศึกษา ทั้งระบบการสื่อสารไร้สาย ระบบควบคุมด้วยไมโครคอนโทรลเลอร์ และเซ็นเซอร์สำหรับการตรวจวัดสภาพแวดล้อมแบบต่างๆ และสามารถนำไปประยุกต์ใช้งานจริงได้ โครงงานเรื่องเครื่องควบคุมอุปกรณ์ไฟฟ้าอัตโนมัติตามสภาพแวดล้อมนี้ จึงออกแบบมาให้นำไปใช้ในงานด้านต่างๆ อย่างหลากหลาย ด้วยการนำอุปกรณ์ตรวจสอบสัญญาณหลายๆ ตัวมารวมกันให้อยู่ภายในอุปกรณ์โมดูลตัวเดียว ที่สามารถควบคุมการทำงานของอุปกรณ์ไฟฟ้าโดยอัตโนมัติ และยังเพิ่มความคล่องตัวและการนำไปประยุกต์ใช้งานที่หลากหลาก ด้วยการใช้การสื่อสารแบบไร้สายใช้สัญญาณคลื่นวิทยุชุมชนในการส่งสัญญาณจากที่หนึ่งไปยังอีกที่หนึ่ง โดยแสดงผลออกมาทางจอแอลซีดีที่ตัวโมดูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

โครงงานนี้จะมีส่วนประกอบแบ่งออกเป็นส่วนใหญ่ ๆ อยู่ 5 ส่วนด้วยกัน ดังนี้

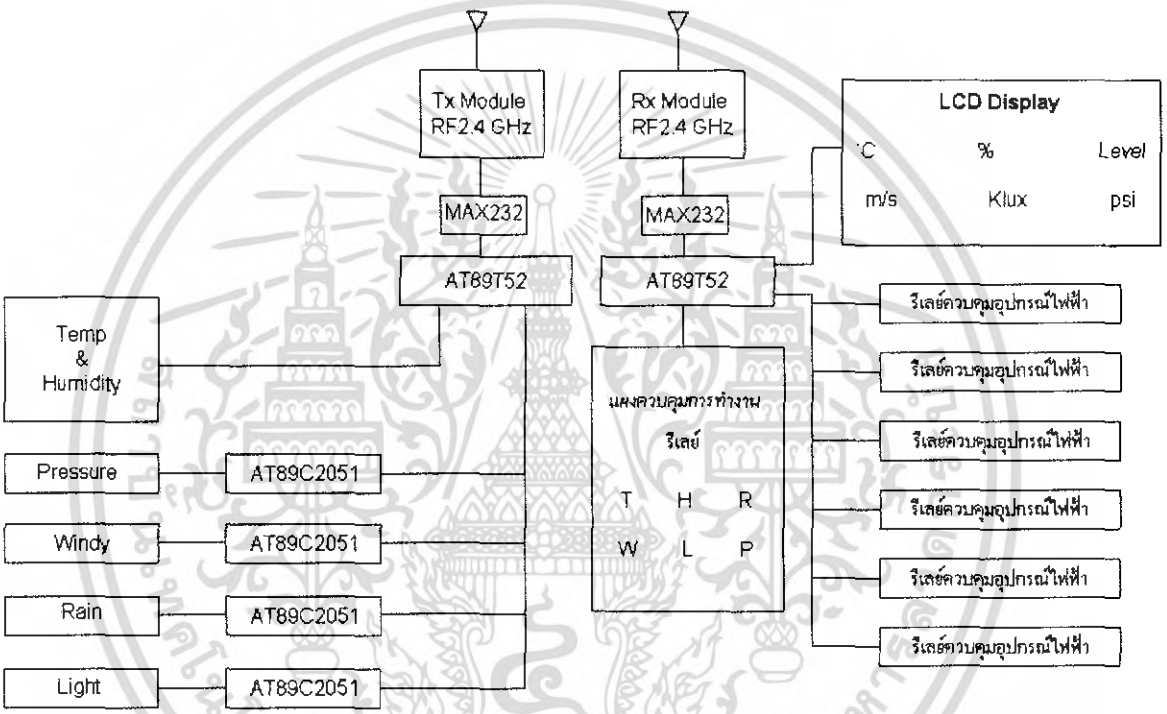
1.อุปกรณ์ตรวจสอบสภาพแวดล้อม

2.ภาคมัลติเพล็กซ์และดีมัลติเพล็ก

3.ภาครับ-ส่งสัญญาณ FM

4.แสดงผลผ่านจอ LCD

5.ส่วนควบคุมอุปกรณ์ไฟฟ้าโดยอัตโนมัติ



รูปที่ 2.1 บล็อกไดอะแกรมการทำงานโดยรวม

2.1 อุปกรณ์ตรวจสอบสภาพแวดล้อม

อุปกรณ์ตรวจสอบสัญญาณมีทั้งหมด 6 ชนิด ดังนี้

1.อุณหภูมิ

2.ความชื้น

3.ความดัน

4.ลม

5.ฝน

6.แสงแดด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 อุณหภูมิและความชื้น

เนื่องจากในวิทยานิพนธ์นี้ใช้เซนเซอร์ตรวจสอบค่าอุณหภูมิและความดันเป็นเซ็นเซอร์ที่อยู่ในตัวเดียวกันจึงขอรวมทฤษฎีของการวัดทั้งสองเข้าด้วยกันเนื่องจากค่าของการวัดทั้งสองมีการเกี่ยวพันซึ่งกันและกัน

อุณหภูมิของอากาศ (Air temperature)

คือ โดยปกติอุณหภูมิที่นิยมใช้วัดกันในประเทศไทยจะมีหน่วยเป็นองศาเซลเซียส (Celsius) หรือ C แต่ถ้าต้องการแปลงเป็นองศาโรเมอร์ (Romer) หรือ องศาฟาเรนไฮต์ (Farenhi) จะมีสมการดังนี้

$$\frac{C}{5} = \frac{R}{4} = \frac{F-32}{9} \tag{2.1}$$

เมื่อ C คือ ค่าที่วัดอุณหภูมิโดยใช้หน่วยองศาเซลเซียส

R คือ ค่าที่วัดอุณหภูมิโดยใช้หน่วยองศาโรเมอร์

F คือ ค่าที่วัดอุณหภูมิโดยใช้หน่วยองศาฟาเรนไฮต์

ความดันไอน้ำ (Vapor pressure)

คือ การที่ไอน้ำเข้าไปแทรกในโมเลกุลของอากาศได้อย่างสนิท โมเลกุลของไอน้ำเบา อากาศชื้นจึงเบากว่าอากาศแห้ง เมื่อปริมาตรและอุณหภูมิเดียวกัน

ความชื้นสมบูรณ์ (Absolute humidity)

คือ ปริมาณไอน้ำที่มีอยู่ได้เต็มที่ในสภาวะอากาศขณะนั้น มีสมการ

$$\text{Absolute humidity} = \frac{0.217\times10^{-6}e}{T} \tag{2.2}$$

เมื่อ Absolute humidity มีหน่วยเป็น gm.cm-3

T คือ absolute temperature = 273+C

e คือ actual vapor pressure

ความชื้นสัมพัทธ์ (Relative humidity)

คือ อัตราส่วนระหว่างความจุไอน้ำของอากาศที่มีอยู่จริงขณะนั้นกับความชื้นสมบูรณ์หรือความชื้นที่มีได้เต็มที่ ณ อุณหภูมิเดียวกัน

$$R.H. = \frac{e}{e_s}\times100 \tag{2.3}$$

เมื่อ R.H. คือ ค่าความชื้นสัมพัทธ์มีหน่วยเป็น %

e คือ ความชื้นที่มีอยู่จริง

es คือ ความชื้นที่มีอยู่ได้เต็มที่ในอากาศขณะนั้นหรือความชื้นสมบูรณ์

ความชื้นจำเพาะ (Specific humidity)

คือ จำนวนไอน้ำที่มีอยู่ในสัดส่วนของมวลอากาศต่อ 1 หน่วยเช่นต่อ 1 กิโลกรัมต่อ 1 millibar เป็นต้น

$$q = 0.622\frac{e}{p} \tag{2.4}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ q = ความชื้นจำเพาะ ไม่มีหน่วย

e = น้ำหนักของไอน้ำ หน่วยเป็น Millibar

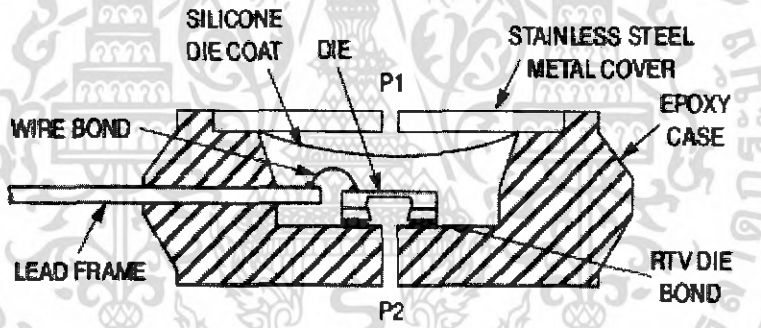
p = น้ำหนักของความชื้นอากาศ (moist air) หน่วยเป็น Millibar

จากสมการของความชื้นสัมบูรณ์ จะพบว่ามีความเกี่ยวข้องกับอุณหภูมิ โดยที่ถ้าเราทราบความชื้นก็จะสามารถคำนวณหาค่าของอุณหภูมิได้

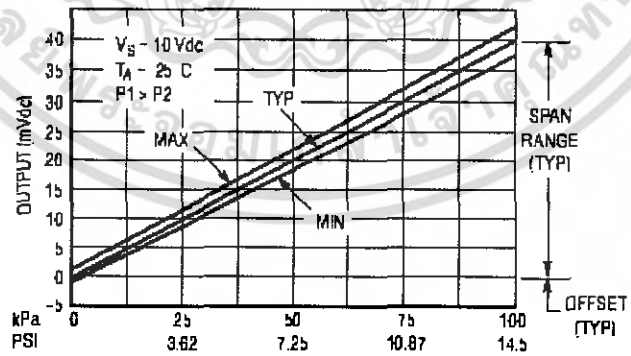
จากหลักการนี้จึงเกิดการสร้างเซ็นเซอร์ที่มีความสามารถตรวจจับค่าความชื้นและอุณหภูมิขึ้นมาซึ่งอยู่ในตัวเดียวกันโดยมีฟังก์ชั่นการคำนวณและการประมวลผลอยู่ภายในตัวเอง โดยเซ็นเซอร์ตัวนี้คือดิจิตอลเซ็นเซอร์ของ บริษัท Sensirion เบอร์ SHT 1X ซึ่งมีบล็อกไดอะแกรมดังรูปที่ 3.1 ในบทที่ 3

2.1.2 ความดัน

อุปกรณ์ตรวจสอบความดันอากาศ ในที่นี้เลือกใช้ไอซีของโมโตโรล่าร์ MPX10D โดยตัวโครงสร้างทำมาจาก Monolithic Silicon piezoresistor ซึ่งจะเกิดการเปลี่ยนแปลงแรงดันเอาท์พุตเมื่อความดันเกิดการเปลี่ยนแปลง โดยความดันที่สามารถวัดออกมานั้นจะเป็นแรงดันที่เปรียบเทียบจากขา P1 และ P2 ตามรูป โดยสามารถวัดได้ความดันสูงสุด 10 kPa



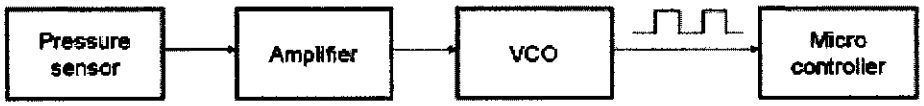
รูปที่ 2.2 โครงสร้างภายในของเซ็นเซอร์ตรวจจับความดัน



รูปที่ 2.3 กราฟเปรียบเทียบค่าแรงดันไฟฟ้ากับความดันอากาศจากคาต้าชีท

เอาท์พุทของเซ็นเซอร์วัดความดัน จะเป็นแรงดันไฟฟ้าที่เปลี่ยนแปลงตามค่าแรงดัน โดยแรงดันที่จ่ายออกมา จะนำไปควบคุมการสร้างสัญญาณนาฬิกา ดังบล็อกไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

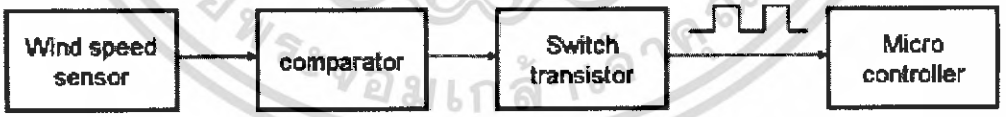


รูปที่ 2.4 บล็อกไดอะแกรมของระบบเซ็นเซอร์ตรวจสอบความดันอากาศ

การสร้างสัญญาณนาฬิกานั้น สร้างจากส่วนของวงจร โวลเตจคอนโทรลออสซิลเลเตอร์ เพื่อให้ความถี่ที่จ่ายออกมาสัมพันธ์กับแรงดัน คือเมื่อแรงดันที่จ่ายออกมาจากเซ็นเซอร์ตรวจสอบความดันมีค่ามาก จะทำให้การผลิตความถี่ของวงจรออสซิลเลเตอร์ มีค่าความถี่สูงตามไปด้วย โดยในส่วนของวงจรขยายจะใช้ ไอซีออปแอมป์ ทำหน้าที่เป็นส่วนขยายสัญญาณ และในส่วนของวงจรโวลเตจคอนโทรลออสซิลเลเตอร์ ก็จะใช้ไอซี VCO ในการสร้างความถี่เพื่อให้ไมโครคอนโทรลเลอร์ทำการประมวลผลต่อไป

2.1.3 ลม

วงจรวัดความเร็วลมนี้ใช้หลักการของการตรวจจับแสงจาก Infrared Emitting Diode ที่ผ่านเข้าตัว Photo Transistor เมื่อถูกตัดแสงจากแผ่นเพลทตัดแสง โดยแผ่นเพลทตัดแสงจะถูกติดตั้งไว้กับแกนหมุนที่หมุนเมื่อมีกระแสลมไหลผ่าน แผ่นเพลทที่หมุนตามแกนจะไปตัดลำแสงที่ถูกฉายจากไดโอดไปยังโฟโต้ทรานซิสเตอร์ ซึ่งโฟโต้ไดโอดจะอยู่ในช่วงนำกระแสและไม่นำกระแสสลับไปมา จากนั้นแรงดันที่สลับไปมาจะถูกส่งไปยังออปแอมป์เพื่อกำหนดเป็นแรงดันแล้วทำการขยายสัญญาณโดยใช้หลักการของ Comparator และเอาท์พุทของออปแอมป์จะทำหน้าที่เป็นตัวสวิทช์ให้กับทรานซิสเตอร์เพื่อกำเนิดสัญญาณพัลส์ แล้วส่งให้ไมโครคอนโทรลเลอร์ทำการนับความถี่ต่อไป

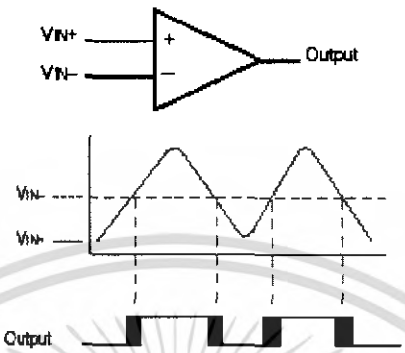


รูปที่ 2.5 บล็อกไดอะแกรมของระบบเซ็นเซอร์ตรวจวัดความเร็วลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร Comparator

Comparator Circuit คือวงจรที่เราใช้ในการตรวจสอบ voltage ของ input หนึ่งว่ามีขนาดสูงหรือต่ำกว่าอีก input หนึ่ง ดังรูปที่ 2.6 แสดงพื้นฐานของวงจร comparator โดยการใช้ op-amp

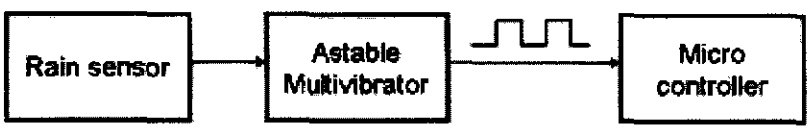


รูปที่ 2.6 แสดงอินพุทและเอาท์พุทของวงจร Comparator

ถ้าหากว่ามีความแตกต่างของ Voltage ของ V1 และ V2 ในขณะใดๆแล้ว จะทำให้ Op-amp นั้นเกิดการทำงานในสถานะ Saturation ซึ่ง Vout ของ Op-amp ก็จะมีเฟสตาม ความแตกต่างของ V1 และ V2 *เมื่อใดก็ตามที่สัญญาณ Vin ที่ขา non-inverting เข้ามามีแรงดันเป็นบวกมากกว่า Vin+ แรงดัน* output ของ Op-amp ก็จะมีแรงดัน +Vsat, ถ้าหากสัญญาณ Vin เข้ามาเป็นแรงดันด้านลบ Output ของ Op-amp ก็จะเป็น –Vsat โดยค่าของ Vsat จะมีค่ากำจัดโดยขึ้นกับค่า ± Vsupply ที่จ่ายให้กับ Op-amp

2.1.4 วงจรตรวจสอบฝน

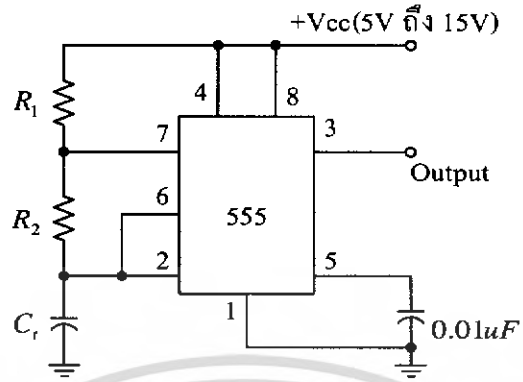
วงจรตรวจสอบฝนใช้หลักการการสร้างความถี่จากการเปลี่ยนแปลงของปริมาณน้ำฝนที่ตกกระทบกับตัวตรวจจับฝน โดยตัวตรวจจับฝนนั้นจะมีคุณสมบัติที่เปลี่ยนแปลงค่าความต้านทานตามปริมาณน้ำที่ตกกระทบ โดยเมื่อไม่มีน้ำมาตกกระทบกับตัวตรวจจับจะทำให้มีความต้านทานสูงมากและเมื่อมีน้ำมาตกกระทบความต้านทานจะลดลงโดยที่เราจะนำความต้านทานที่เปลี่ยนแปลงได้นี้ ไปต่อร่วมกับวงจรกำเนิดความถี่แบบอะสเตเบิลมัลติไวเบรเตอร์จะสามารถสร้างความถี่ออกมาหลายๆ ค่าตามความต้านทานที่เปลี่ยนแปลง



รูปที่ 2.7 บล็อกไดอะแกรมของระบบเซ็นเซอร์ตรวจวัดปริมาณฝน

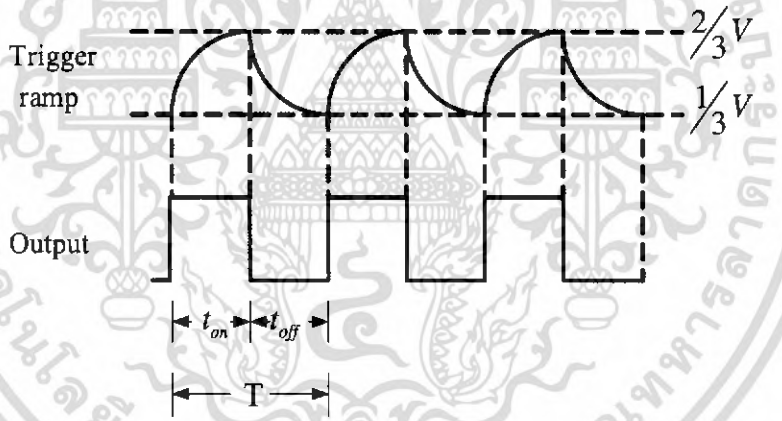
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรอะสเตเบิล มัลติไวเบรเตอร์ (Astable Multivibrator) เป็นวงจรกำเนิดสัญญาณพัลส์ โดยเมื่อจ่ายแรงดันให้กับวงจรจะสามารถผลิตสัญญาณพัลส์ ออกมาโดยสามารถกำหนด T_{on} และ T_{off} ได้



รูปที่ 2.8 วงจรอะสเตเบิล มัลติไวเบรเตอร์

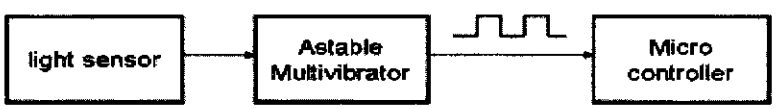
การกำหนดความถี่หรือช่วงเวลา T_{on} และ T_{off} ถูกกำหนดโดย ค่า R1, R2, Ct ซึ่ง R1 และ R2 จะเป็นตัวกำหนดกระแสไฟฟ้า ของวงจรเพื่อไปประจุให้กับ Ct โดยการ Charge และ Discharge ของ Ct จะเป็นตัวกระตุ้นให้ IC 555 สร้าง สัญญาณพัลส์



รูปที่ 2.9 สัญญาณทริกเกอร์แร้มป์ และสัญญาณเอาท์พุต

2.1.5 วงจรตรวจสอบแสง

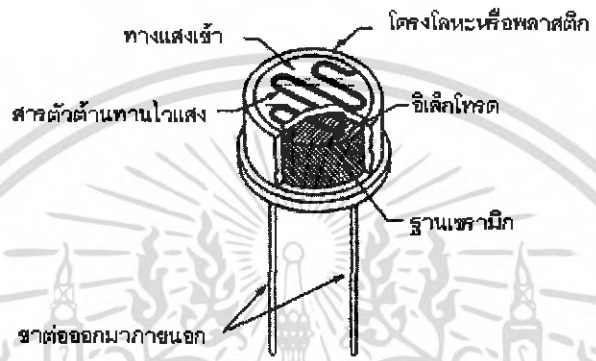
ใช้ LDRเป็นตัวตรวจจับความเข้มของแสง โดยอาศัยคุณสมบัติของ LDR ที่เปลี่ยนแปลงความต้านทานตามความเข้มของแสง โดยนำมาต่อร่วมกับวงจรอะสเตเบิล มัลติไวเบรเตอร์ เช่นเดียวกับวงจรตรวจสอบฝน



รูปที่ 2.10 บล็อกไดอะแกรมของระบบเซ็นเซอร์ตรวจวัดแสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Light Dependent Resistor (LDR)มีชื่อเรียกอีกอย่างหนึ่งว่าโฟโตรีซิสเตอร์ (Photoresistor) ซึ่งเป็นตัวต้านทานที่ทำงานโดยอาศัยแสงที่มาตกกระทบ นั่นคือ วัสดุที่ใช้ทำโฟโตริซิสเตอร์ เมื่อถูกแสงจะมีค่าความนำมากขึ้น หรือทำให้ค่าความต้านทานลดลงนั่นเอง โฟโตริซิสเตอร์สร้างจากวัสดุนำแสงที่มีลักษณะเป็นแผ่นบางๆ ซึ่งค่าความต้านทานของวัสดุนี้ลดลงเมื่อมีแสงมาตกกระทบ โดยพลังงานแสงจะถูกดูดซึมจากอะตอมที่มีอยู่มากมายในวัสดุนำแสงนี้และทำให้เกิดการปลดปล่อยอิเล็กตรอนที่อยู่วงนอกสุด (Valence Electron) ออกมา ด้วยเหตุผลของจำนวนอิเล็กตรอนอิสระมากขึ้น จึงทำให้กระแสไฟฟ้าสามารถไหลผ่านโฟโตริซิสเตอร์ได้มาก ดังนั้นจึงทำให้ความต้านทานมีค่าลดลงด้วย



รูปที่ 2.11 โครงสร้างของ LDR

ผลตอบสนองทางไฟฟ้าของ LDR นั้น โดยอัตราส่วนระหว่างความต้านทานของ LDR ในขณะที่ไม่มีแสง กับขณะที่มีแสง อาจจะเป็นได้ตั้งแต่ 100 เท่า 1,000 เท่า หรือ 10,000 เท่า แล้วแต่รุ่น แต่โดยทั่วไปแล้ว ค่าความต้านทานในขณะที่ไม่มีแสงจะอยู่ในช่วง ประมาณ 0.5 เมกะโอห์ม ขึ้นไป ในที่มืดสนิทอาจขึ้นไปได้มากกว่า 2 เมกะโอห์มและในขณะที่มีแสงจะเป็นประมาณ 10 – 20 กิโลโอห์มลงไปอาจจะเหลือเพียงไม่กี่โอห์ม หรือ ไม่ถึงโอห์ม ทนแรงดันสูงสุดได้ไม่ต่ำกว่า 100 V และ กำลังสูญเสีย อย่างต่ำประมาณ 50 mW

จากคุณสมบัติของ LDR ที่เปลี่ยนแปลงความต้านทานตามความเข้มของแสงเราจึงสามารถนำมาออกแบบวงจรกำเนิดความถี่ ที่มีการแปรค่าความถี่ตามความเข้มของแสง โดยนำ LDR มาต่อร่วมกับวงจร Astable Multivibrator เช่นเดียวกับวงจรตรวจจับฝน

2.2 ภาคประมวลผลความถี่ มัลติเพล็กซ์และดีมัลติเพล็กซ์

ในส่วนของการประมวลผลความถี่ที่ถูกสร้างขึ้นมาจากเซ็นเซอร์ เราจะใช้ไมโครคอนโทรลเลอร์ MCS 51 ชนิด 20 ขา ในการที่จะทำการตรวจนับความถี่ เพื่อส่งค่าที่ทำการนับได้ ไปทำการมัลติเพล็กซ์โดยใช้ไมโครคอนโทรลเลอร์อีกตัวหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของการมัลติเพล็กซ์และดีมัลติเพล็กซ์นั้นจะใช้ไมโครคอนโทรลเลอร์ชนิด 40 ขาในการที่จะทำการมัลติเพล็กซ์สัญญาณจากไมโครคอนโทรลเลอร์ย่อยที่ทำการนับความถี่จากเซ็นเซอร์ รวมถึงการดีมัลติเพล็กซ์ แสดงผล และควบคุมอุปกรณ์ไฟฟ้า

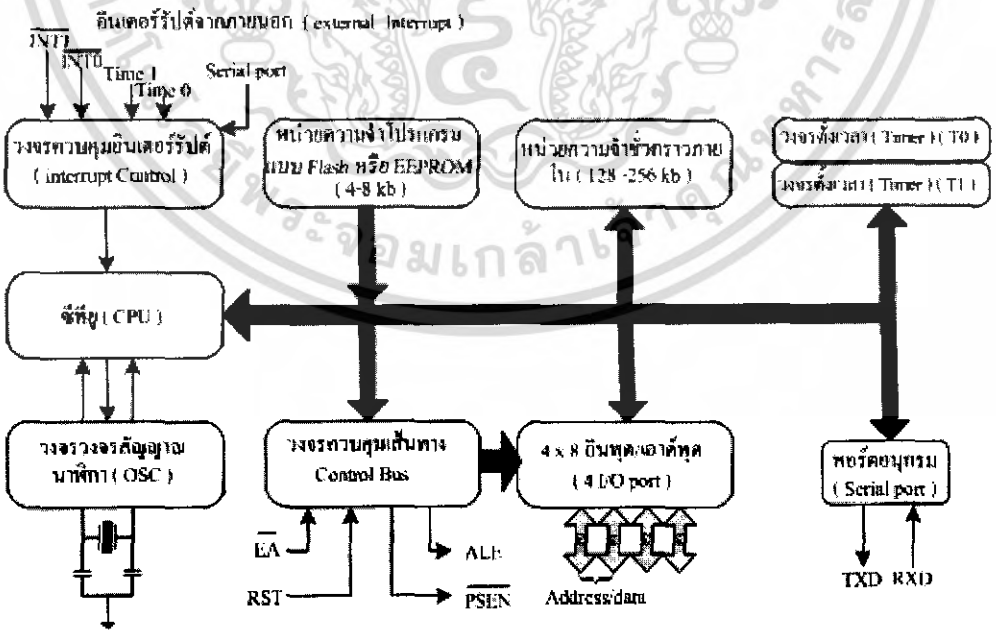
2.2.1 MICROCONTROLLER

2.2.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51

โครงงานนี้ใช้ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 ซึ่งมีหน่วยความจำภายในเป็นแบบแฟลต (Flash memory) ทำให้สามารถลบและเขียนใหม่ได้นับพันครั้ง จึงสามารถใช้งานในรูปแบบของไมโครคอนโทรลเลอร์ชิปเดี่ยวไม่ต้องใช้หน่วยความจำภายนอก ส่งผลให้สามารถใช้งานพอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ได้อย่างเต็มประสิทธิภาพ โดยคุณสมบัติคือ

-เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
-ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง
-หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม
-ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาต์พุต
-มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
-ไทเมอร์/เคาน์เตอร์ขนาด 16 บิตอย่างน้อย 2 ตัว
-สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
-สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
-มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป

โครงสร้างพื้นฐานของ MCS-51 มีส่วนประกอบต่างๆ ดังรูปที่ 2.12

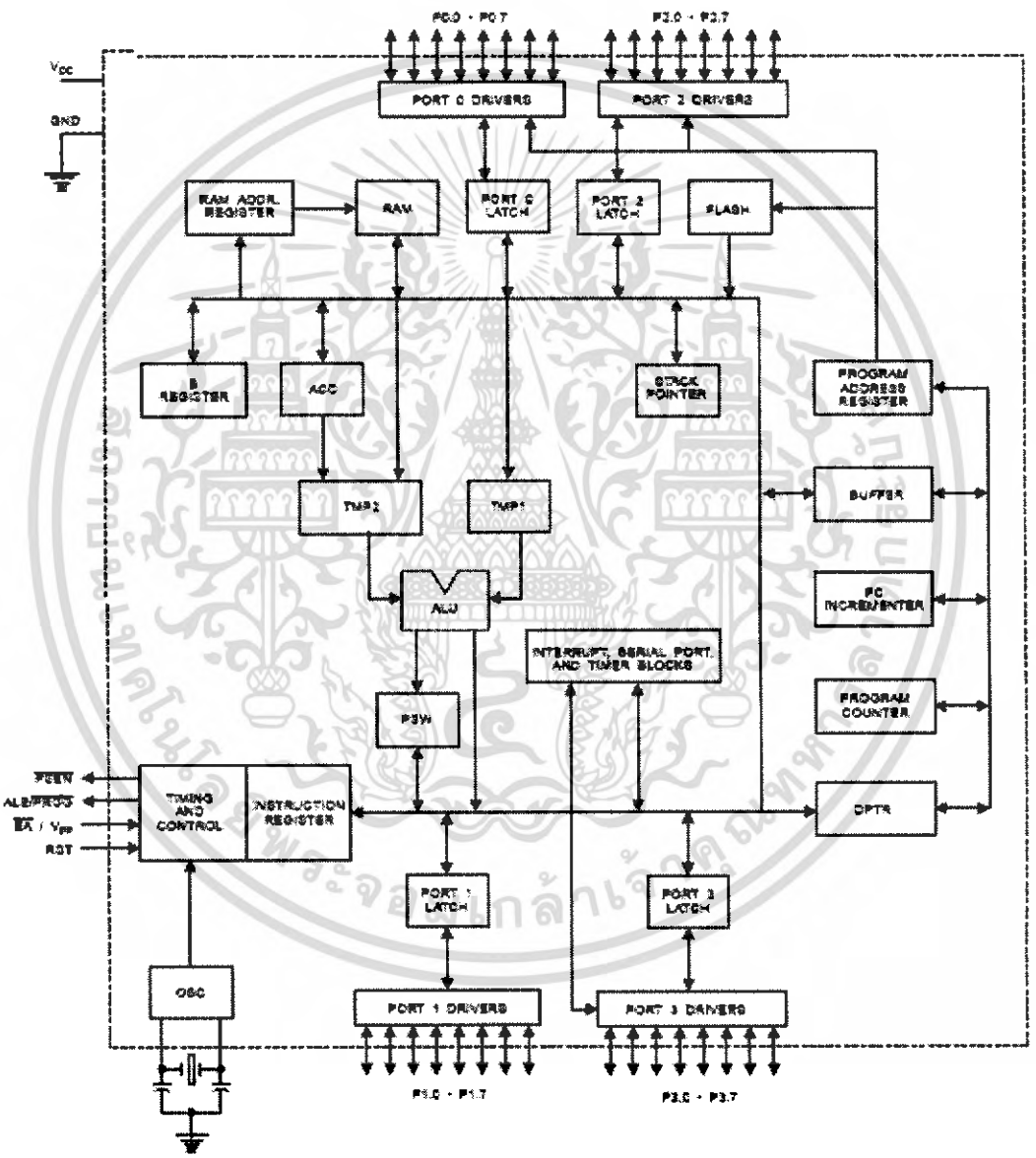


รูปที่ 2.12 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.2 โครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51

วงจรภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ประกอบด้วยวงจรอินพุตและเอาต์พุตทั้งหมด 4 พอร์ต แต่ละพอร์ตจะเป็น 8 บิต หน่วยความจำโปรแกรมภายใน (EPROM , EEPROM และ Flash) หน่วยความจำที่เป็นข้อมูลนั้น (RAM) ซึ่งอยู่จะในวงจรหลักของ ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ตลอดจนวงจรการคำนวณทางคณิตศาสตร์และลอจิก (ALU) วงจรรีจีสเตอร์ทั่วไป และรีจีสเตอร์ฟังก์ชั่นการใช้งานเฉพาะ แสดงดังรูปที่ 2.13

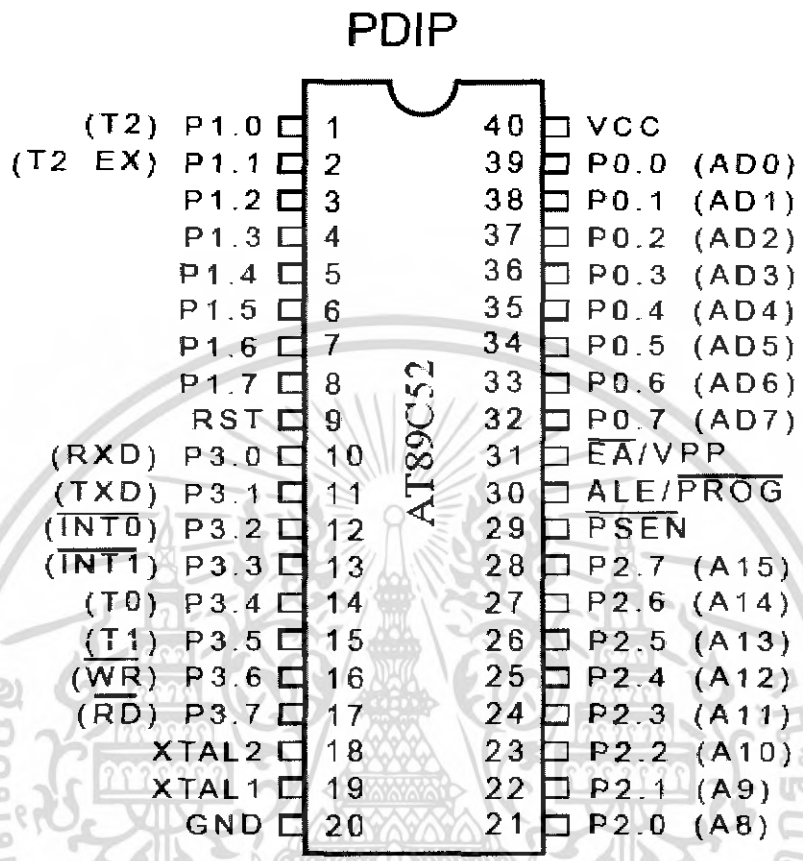


รูปที่ 2.13 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.3 การจัดขาของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ MCS-51 ทุกเบอร์จะมีสถาปัตยกรรมและขาใช้งานพื้นฐานเหมือนกัน ดังแสดงในรูปที่ 2.14 โดยมีรายละเอียดขั้นต้นดังนี้



รูปที่ 2.14 ตำแหน่งขาของไมโครคอนโทรลเลอร์ตระกูล MCS-51

ขา VCC ใช้สำหรับต่อไฟเลี้ยง +5V

ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 0 (P0.0-P0.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูงสมารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

ขาพอร์ต 1 (P1.0-P1.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย นอกจากนั้นจะใช้ขา P1.0 เป็นขาอินพุตสำหรับนับค่าของไทเมอร์ 2 และ P1.1 เป็นขาอินพุตทริกเกอร์ของไทเมอร์ 2 ในขณะที่ขา P1.4 ถึง P1.7 เป็นขาสำหรับเชื่อมต่อแบบ SPI เพื่อทำการโปรแกรมข้อมูลในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาพอร์ต 2 (P2.0-P2.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วยส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (Float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

ขาพอร์ต 3 (P3.0-P3.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (Float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นต้นต่อไปนี้

P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรมหรือขา RxD

P3.1 ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรมหรือขา TxD

P3.2 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 หรือขา INT0

P3.3 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา INT1

P3.4 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0 หรือขา T0

P3.5 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง1 ขา T1

P3.6 ใช้เป็นขาสัญญาณ WR ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

P3.7 ใช้เป็นขาสัญญาณ RD ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

ขารีเซต (Reset) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซตสถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมทชีนไซเกิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

ขา ALE/PROG (Address Latch Enable/Program pulse input) เป็นขาที่ใช้ในการควบคุมการแลตซ์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีพรอม

ขา PSEN (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละแมชชีนไซเกิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใดๆออกมา

ขา EA/Vpp (External access Enable/Programming voltage input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรเลอร์ ถ้าหากขานี้เป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้เป็น "1" เป็นการเลือกให้ไมโครคอนโทรเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ ที่ขานี้ยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้เป็นขาอินพุตสำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม+12V

ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

2.2.2 การมัลติเพล็กซ์โดยใช้ไมโครคอนโทรลเลอร์

สัญญาณจากเซ็นเซอร์ต่างๆ ที่มีอยู่ด้วยกัน 6 สัญญาณนั้น คือ อุณหภูมิ ความชื้น ความดัน ลม น้ำฝน และความเข้มแสง โดยใช้เซ็นเซอร์ต่างเป็นตัวตรวจจับสัญญาณเหล่านี้อยู่ในรูปแบบพัลส์เมื่อถูกส่งเข้าไมโครคอนโทรลเลอร์เราจะต้องทำการนำสัญญาณพัลส์มารวมกันโดยกระบวนการที่รวมสัญญาณตัวอย่างของหลายๆสัญญาณลงเป็นลำดับตามเวลาหรือที่เรียกว่า การมัลติเพล็กซ์แบบแบ่งเวลา โดยสัญญาณอินพุทที่จะทำการมัลติเพล็กซ์ด้วยไมโครคอนโทรลเลอร์นั้นจะต้องทำการบัฟเฟอร์ไว้ในหน่วยความจำก่อน จึงจะสามารถส่งออกจากไมโครคอนโทรลเลอร์ทางพอร์ตการสื่อสารแบบอนุกรมทีละ 8 บิต

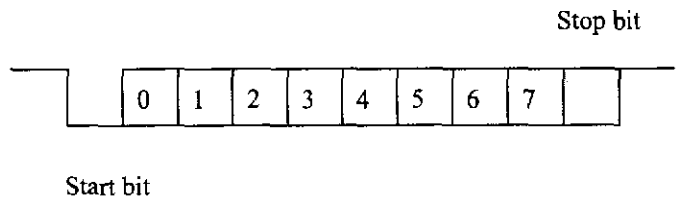
การใช้งานพอร์ทสื่อสารอนุกรมนั้นจะทำงานแบบฟูลดูเพล็กซ์ ซึ่งสามารถรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน โดยจะแบ่งเป็น 2 ขาในไมโครคอนโทรเลอร์ ในส่วนมัลติเพล็กซ์นี้เราจะใช้ขาส่ง TxD เพื่อส่งสัญญาณที่ทำการมัลติเพล็กซ์มาแล้ว สำหรับพอร์ทอนุกรมนี้ใช้เป็นบัฟเฟอร์สำหรับส่งข้อมูลให้ SBUF ด้วย และโหมดในการทำงานจะแบ่งออกเป็น4 โหมด ซึ่งแต่ละโหมดจะมีความเร็วและความเหมาะสมในการใช้งานที่ต่างกัน โดยเราจะเลือกใช้ที่โหมด1 ใช้ทั้งหมด 10 บิต ข้อมูล 8 บิต 1 start bit และ 1 stop bit และสามารถเปลี่ยนแปลงความเร็วในการส่งข้อมูลได้ โดยขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โฟลว์ของ Timer 1,2 ในการหาความเร็วดังสมการ

$$BaudrateMode1,3 = \frac{2^{SMOD} \times CPUOSC}{32 \times 12 \times [256 - (TH1)]}$$ โดยใช้ Timer 1 (2.5)

$$BuadrateMode1,3 = \frac{CPUOSC}{32 \times [65536 - (RCAP2H, RCAP2L)]}$$ โดยใช้ Timer 2 (2.6)

ส่วนในการจัดเฟรมนั้น จะแสดงได้ดังรูป

Sync Frame	Sync Frame	Sensor # 1	Sensor # 2	Sensor # 3	Sensor # 4	Sensor # 5	Sensor # 6



รูปที่ 2.15 แสดงการจัดเฟรมของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจากรูปแสดงการจัดการเฟรมข้อมูล โดย 2 เฟรมแรกจะเป็นเฟรมซิงโครไนซ์เพื่อที่จะสามารถเรียงลำดับข้อมูลได้อย่างถูกต้อง โดย 6 เฟรมต่อมาจะเป็นข้อมูลของเซนเซอร์ต่างๆ โดยเซนเซอร์จะใช้เฟรม 1 เฟรมในการส่งข้อมูลแต่ละครั้ง

2.3 ภาครับ-ส่งสัญญาณ RF

โครงงานนี้ได้ใช้โมดูลรับส่ง RS232 to RF Wireless (RF 2.4GHz) Converter เป็นชุด Signal Converter ในการแปลงสัญญาณระหว่าง RS232 กับ RF-Wireless โดยในโหมดการทำงานของการส่งข้อมูล (Transmitter) จะทำการรอรับข้อมูลจากพอร์ตสื่อสารอนุกรม RS232 จากขา RX แล้วแปลงเป็นสัญญาณความถี่ (GFSK) ส่งออกไปในอากาศ และในทางกลับกันในโหมดการทำงานแบบรับ (Receiver) ก็จะค่อยทำการตรวจจับข้อมูลในรูปสัญญาณความถี่ (GFSK) จากด้าน RF เพื่อแปลงกลับเป็นข้อมูล RS232 ส่งออกไปทางขา TX ได้ด้วย

ซึ่งจะเห็นได้ว่าชุดแปลงสัญญาณนั้น สามารถนำไปต่อใช้งานร่วมกับพอร์ตสื่อสารอนุกรม แบบ RS232 เพื่อใช้งานในลักษณะของการสื่อสารอนุกรมแบบไร้สาย (Wireless Transceiver) ได้โดยตรงโดยจะมีข้อดีกว่า คือ สามารถรับส่งข้อมูลกันได้ในระยะทางที่ไกลกว่า RS232 หลายเท่าตัว และประการสำคัญคือไม่จำเป็นต้องใช้สายสัญญาณที่เป็นตัวนำสัญญาณทางไฟฟ้าในการสื่อสารข้อมูลกัน ทำให้สามารถเปลี่ยนแปลงหรือเคลื่อนย้ายจุดรับส่งข้อมูลได้ตลอดเวลา ซึ่งถ้าเป็นการรับส่งข้อมูลด้วยระบบ RS232 แบบที่ใช้สายสัญญาณนั้นจะเกิดความยุ่งยากในการติดตั้งสายสัญญาณเป็นอย่างมากแต่อย่างไรก็ตามการรับส่งข้อมูลโดยใช้อากาศเป็นตัวกลางในการสื่อสารนั้น ก็มีข้อจำกัดบางประการเหมือนกัน โดยเฉพาะอย่างยิ่ง เรื่องความน่าเชื่อถือของข้อมูลที่รับส่งกัน ซึ่งมีโอกาสผิดพลาดหรือสูญหายได้เหมือนกัน เนื่องจากในการลำเลียงข้อมูลนั้นไม่ได้ใช้สายสัญญาณเป็นตัวกลางในการรับส่งข้อมูล แต่ใช้อากาศเป็นตัวกลางในการรับส่งข้อมูลแทน ซึ่งมีโอกาสที่ข้อมูลจะเกิดการรบกวนจากสัญญาณอื่นๆที่มีย่านความถี่ใกล้เคียงกันแล้วทำให้ข้อมูลผิดเพี้ยนไปได้บ้างเหมือนกัน ซึ่งระบบการจัดการข้อมูลของเครื่องแปลงสัญญาณนั้น มีระบบการเข้ารหัสและถอดรหัสข้อมูลที่มีความน่าเชื่อถืออยู่ในเกณฑ์ที่จัดว่าดี โดยข้อมูลแต่ละ Byte ที่มีการรับส่งกันนั้น จะมีการตรวจสอบความถูกต้องของข้อมูลให้ด้วยแล้ว โดยข้อมูลที่รับได้จากด้าน RF นั้นรับประกันได้ว่าเป็นข้อมูลที่มีความถูกต้องแน่นอน แต่อย่างไรก็ตามการรับส่งข้อมูลนั้นมีโอกาสผิดพลาดในเรื่องของการสูญหายของข้อมูลบ้างเหมือนกันเนื่องจากกลไกในการรับส่งข้อมูลนั้น จะมีการตรวจสอบข้อมูลทุก Byte ที่รับได้จาก RF เสมอ ซึ่งถ้าพบว่ามีความผิดพลาดเกิดขึ้นจะทิ้งข้อมูล Byte นั้นไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

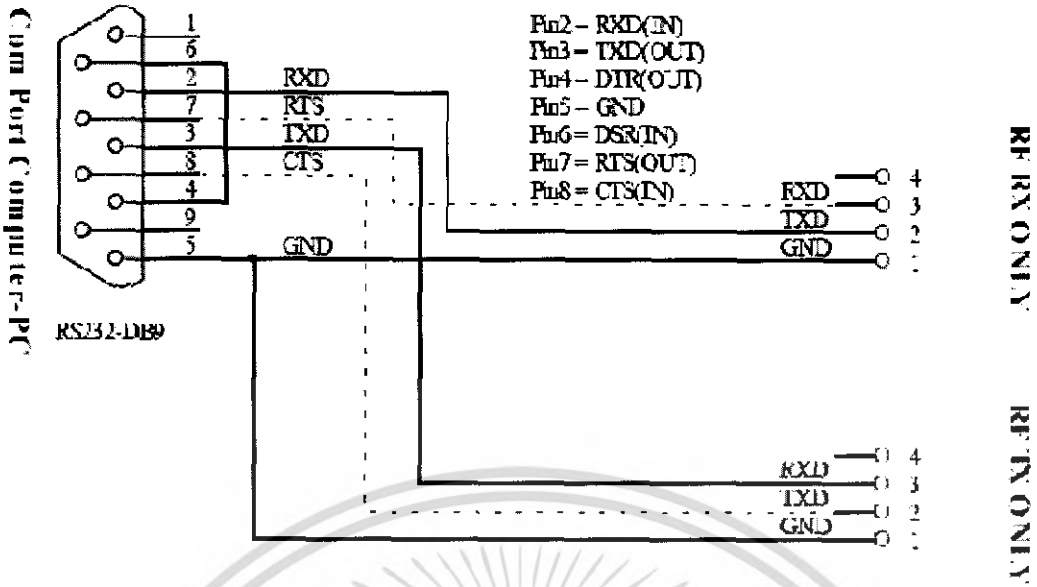
2.3.1 การทำงานแบบ RF Receive Only

เป็นการทำงานแบบทิศทางเดียว โดยการทำงานในโหมดนี้ จะเป็นการรอรับข้อมูลความถี่แบบ GFSK จากด้าน RF แล้วเปลี่ยนเป็นข้อมูลอนุกรมส่งออกไปทางขา TX (Transmit) ของ RS232 โดยการทำงานจะวนรอบอยู่เช่นนี้ไปตลอด ซึ่งในการใช้งาน ในโหมดนี้จะต้องนำสัญญาณ TX(Transmit) ไปต่อกับขาสัญญาณ RX (Receive) ของอุปกรณ์ด้านตรงข้าม (RS232 ของคอมพิวเตอร์ PC) โดยในโหมดนี้ การทำงานของขาสัญญาณ RX ด้าน RS232 ของเครื่องแปลงสัญญาณ จะถูกเปลี่ยนหน้าที่เป็นสัญญาณ CTS (Clear To Send) สำหรับใช้ตรวจสอบความพร้อมในการส่งข้อมูลไปให้อุปกรณ์ด้านตรงข้ามแทน ซึ่งในการใช้งานจะต้องนำสัญญาณนี้ไปต่อเข้ากับสัญญาณ RTS (Ready To Send) ของอุปกรณ์ด้านตรงข้าม โดยเครื่องแปลงสัญญาณจะทำการตรวจสอบสถานะของสัญญาณ RX ซึ่งในโหมดนี้เปรียบเสมือน CTS ว่ามีค่าเป็น "0" หรือไม่ โดยถ้าพบว่าเป็น "0" จึงจะส่งข้อมูลออกไปให้ทางขา TX แต่ถ้าพบว่าสถานะของขาสัญญาณนี้มีค่าเป็น "1" แสดงว่าอุปกรณ์ด้านตรงข้ามยังไม่พร้อมรับข้อมูลก็จะรอจนกว่าจะพบว่าสถานะของสัญญาณดังกล่าวมีค่าเป็น "0" จึงจะส่งข้อมูลออกไปให้ โดยเครื่องสัญญาณ จะสามารถจัดเก็บข้อมูลไว้ใน Buffer เพื่อรอการส่งได้สูงสุด 64 Byte เท่านั้น ซึ่งถ้าในระหว่างที่รอความพร้อมอยู่นั้น มีข้อมูลด้าน RF ส่งเข้ามาเกินกว่า 64 Byte จะทำให้ข้อมูลที่เกินมานั้นสูญหายไป

2.3.2 การทำงานแบบ RF Transmit Only

เป็นการทำงานแบบทิศทางเดียว โดยการทำงานในโหมดนี้จะมีลักษณะตรงกันข้ามกับ RF Receive Onlyกล่าวคือ เครื่องสัญญาณ จะทำหน้าที่รอรับข้อมูลจากขา RX (Receive) ด้าน RS232 แล้วเปลี่ยนเป็นข้อมูลแบบ GFSK ส่งออกไปทางด้าน RF โดยการใช้งานเครื่องในโหมดนี้ จะต้องนำสัญญาณ TX (Transmit) ซึ่งเป็นขาส่งข้อมูลจาก RS232 ของอุปกรณ์ด้านตรงข้ามมาต่อเข้ากับขา RX (Receive) ของเครื่องสัญญาณ ส่วนขาสัญญาณ TX จะถูกเปลี่ยนหน้าที่เป็น RTS (Ready To Send) เพื่อใช้แสดงสถานะความพร้อมในการรับข้อมูลจากด้าน RS232 ซึ่งในการใช้งานจะต้องนำสัญญาณ TX ซึ่งในขณะนี้เปรียบเสมือนกับ RTS นำไปต่อเข้ากับสัญญาณ CTS (Clear To Send) ของอุปกรณ์ด้านตรงข้าม เพื่อใช้ในการตรวจสอบความพร้อมในการรับข้อมูล โดยอุปกรณ์ด้านตรงข้ามจะต้องทำการตรวจสอบสถานะของสัญญาณ RTS นี้ เพื่อตรวจสอบความพร้อมในการรับข้อมูลของเครื่องแปลงสัญญาณด้วย โดยถ้าเครื่องแปลงสัญญาณพร้อมรับข้อมูลจาก RS232 มันจะส่งสัญญาณ RTSให้มีค่าเป็น "0" รอไว้ และเมื่อใดก็ตามที่การรับข้อมูลทางด้านของ RS232 มีจำนวนข้อมูลที่ยังไม่สามารถเปลี่ยนเป็น GFSK เพื่อส่งออกไปทางด้าน RF ได้ทันจนเกือบจะเต็ม Buffer แล้ว เครื่องแปลงสัญญาณจะทำการส่งสัญญาณ RTS ให้มีค่าเป็น "1" ออกไปบอกให้อุปกรณ์ด้านตรงข้ามทราบเพื่อจะได้หยุดการส่งข้อมูลออกมา โดยอุปกรณ์ด้านตรงข้ามจะต้องหยุดการส่งข้อมูลและรอจนกว่าสถานะของสัญญาณ RTS จะกลับเป็น "0" จึงจะเริ่มต้นส่งข้อมูลออกมาใหม่ ซึ่งหลังจากที่เครื่องแปลงสัญญาณส่งสัญญาณ RTS ด้วยค่า "1" ออกไปแล้ว จะยังคงสามารถรับข้อมูลได้เพิ่มเติมอีกไม่เกิน 16 Byte เท่านั้น ซึ่งถ้าอุปกรณ์ด้านตรงข้ามยังส่งข้อมูลต่อเนื่องมาอีกจนเกินขนาดของ Buffer ที่เครื่องแปลงสัญญาณจะรับไว้ได้จะทำให้ข้อมูลที่เกินมานั้นเกิดการสูญหาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 แสดงสายสัญญาณ RS232 เพื่อใช้เครื่องแปลงสัญญาณ ในโหมด RF Receive Only และ RF Transmit Only

2.4 ส่วนแสดงผลผ่านจอ LCD

โครงงานนี้เราจะใช้ โมดูล LCD ต่อเข้ากับไมโครคอนโทรลเลอร์ทางด้านเครื่องรับ โดยเลือกใช้จอขนาด 16 อักษร 2 บรรทัด ในโมดูล LCD จะมีส่วนประกอบหลักๆ 3 ส่วน ดังนี้

1.ตัวแสดงผล (Display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอ LCD

2.ตัวควบคุม (Controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของโมดูล LCD เช่น ลบจอภาพ แสดงตัวอักษร หรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิปควบคุม

3.ตัวขับ (Driver)เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด ชิปที่ใช้ทำหน้าที่เป็นตัวขับนี้ได้แก่ เบอร์ HD44100H และ MSM5259 เป็นต้น

ในการเขียนข้อมูลเพื่อควบคุมให้โมดูล LCD แสดงผลตามที่ผู้ใช้งานต้องการ ต้องส่งคำสั่ง แล้วกำหนดโหมดการทำงานให้แก่โมดูล LCD ก่อนจากนั้นจึงค่อยส่งข้อมูลที่ต้องการแสดงผล เนื่องจากบัสข้อมูลของโมดูล LCD มี 8 เส้นคือ D0-D7 และใช้เป็นทางผ่านของทั้งคำสั่งและข้อมูล ดังนั้นในการส่งคำสั่งและข้อมูลจึงต้องอาศัยการกำหนดสัญญาณลอจิกที่ขา RS ถ้าหากที่ขา RS ได้ลอจิก “0” หมายความว่าข้อมูลที่ป้อนให้แก่โมดูล LCD ขณะนั้นเป็นคำสั่ง ในทางตรงกันข้าม หากขา RS ได้รับลอจิก “1” ข้อมูลที่ป้อนให้ขณะนั้นเป็นข้อมูลที่ใช้ในการแสดงผล

เมื่อต้องการเขียนหรืออ่านข้อมูลใน CGRAM และ DDRAM เริ่มต้นต้องกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรส จากนั้นกำหนดให้ขา RS ได้เป็น “1” เพื่อแจ้งให้ตัวควบคุมภายในโมดูล LCD ทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติไม่ใช่คำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ในกรณีที่ต้องการอ่านข้อมูลต้องกำหนดให้ขา R/W เป็น “1” ข้อมูลขนาด 8 บิต ก็จะปรากฏบนบัสข้อมูล โดยข้อมูลที่อ่านออกมาได้จะเป็นข้อมูลจากแอดเดรสของ CGRAM หรือ DDRAM ตามที่ต้องการ

ในกรณีที่ต้องการเขียนข้อมูล เมื่อกำหนดแอดเดรสและป้อนลอจิก “1” ให้ขา RS แล้ว แล้วต้องกำหนด ให้ขา R/W เป็น “0” ข้อมูลที่อยู่บนบัสข้อมูลจะถูกเขียนลงในรีจิสเตอร์ DR จากนั้นจึงถ่ายทอดลงใน DDRAM ต่อไป

2.4.1 จังหวะการทำงานของ LCD โมดูล

ในการติดต่อกับโมดูล LCD จะต้องมีการหน่วงเวลาหลังจากที่ทำการส่งรหัสคำสั่งหรือข้อมูล เนื่องจากต้องรอให้คอนโทรลเลอร์ภายใน LCD โมดูล แปลความหมายของรหัสคำสั่งและทำงานตามคำสั่งให้เรียบร้อยก่อนจากนั้นจึงจะรับข้อมูลหรือดำเนินการต่อไป

ดังนั้น ในการใช้งานโมดูล LCD การเขียนโปรแกรมต้องมีโปรแกรมเพื่อหน่วงเวลารอให้โมดูล LCD พร้อมทำงานด้วย โดยเมื่อเริ่มจ่ายไฟให้แก่โมดูล LCD ต้องรอประมาณ 10 มิลลิวินาที เพื่อให้ โมดูล LCD ทำการเตรียมความพร้อมหรืออินิเชียล (Initial) หลังจากนั้นก็จะกำหนดลอจิกให้แก่ขา RS ของโมดูล LCD แล้วต้องหน่วงเวลาอีกประมาณ 2 มิลลิวินาทีเพื่อให้คอนโทรลเลอร์ในโมดูล LCD แปลความหมายของลอจิกที่ขา RS ว่าข้อมูลต่อไปที่จะได้รับนั้นเป็นรหัสคำสั่งหรือเป็นข้อมูลที่ต้องการแสดงผล จากนั้นจะเป็นการส่งข้อมูลมารอที่บัสข้อมูล D0-D7 (กรณีทำงานในโหมด 8 บิต) ขั้นตอนต่อไปจะเป็นการส่งสัญญาณพัลส์ไปที่ขา E เพื่ออีเนเบิลโมดูล LCD ให้รับข้อมูลจากบัสข้อมูลเข้าไป โดยพัลส์ที่ป้อนเข้าที่ขา E ของโมดูล LCD ต้องเป็นพัลส์ขอบขาขึ้น จากนั้นทำการหน่วงเวลา 2 มิลลิวินาที

ทั้งหมดที่กล่าวมาคือขั้นตอนและจังหวะในการทำงาน 1 รอบของโมดูล LCD จะเห็นได้ว่ามีโปรแกรมย่อยที่สำคัญอยู่ 3 โปรแกรมย่อยคือ โปรแกรมอินิเชียล LCD โปรแกรมหน่วงเวลา และโปรแกรมย่อยการส่งพัลส์เพื่ออีเนเบิลโมดูล LCD

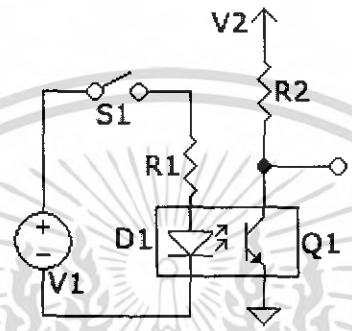
2.5 ส่วนควบคุมอุปกรณ์ไฟฟ้าโดยอัตโนมัติ

ส่วนนี้เป็นส่วนที่เชื่อมต่อมาจากส่วนของการรับค่าสัญญาณมาจากหน่วยประมวลผลของไมโครคอนโทรลเลอร์ที่ได้รับมาจากส่วนรับสัญญาณวิทยุ เมื่อไมโครคอนโทรลเลอร์ตรวจสอบว่าเซ็นเซอร์ตัวใดที่ตรวจจับสภาพแวดล้อมจนได้ค่าของการตรวจวัดถึงตำแหน่งของค่าที่จะทำให้ต้องมีการทำงานของอุปกรณ์อัตโนมัติ จากนั้นตัวไมโครคอนโทรลเลอร์จะทำการส่งสัญญาณควบคุมไปยังชุดสวิทช์ควบคุมอุปกรณ์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
62371

2.5.1 Opto Isolator

คืออุปกรณ์ที่มีหน้าที่เชื่อมต่อทางแสง (Opto Isolator) หรือ ที่เรียกอีกชื่อหนึ่งว่า ตัวเชื่อมต่อผ่านแสง (Opto Coupler) เป็นอุปกรณ์อิเล็กทรอนิกส์ที่ใช้การเชื่อมต่อกันทางแสงโดยใช้หลักการเปลี่ยนสัญญาณไฟฟ้าเป็นสัญญาณแสง และเปลี่ยนกลับจากสัญญาณแสงเป็นสัญญาณไฟฟ้าตามเดิม ใช้สำหรับการเชื่อมต่อสัญญาณระหว่างสองวงจรที่ต้องการแยกกันทางไฟฟ้าอย่างเด็ดขาดเพื่อป้องกันการรบกวนกันทางไฟฟ้า แบ่งออกเป็นหลายชนิดแต่ละชนิดจะประกอบด้วยตัวส่งแสงและตัวรับแสงที่เป็นอุปกรณ์อิเล็กทรอนิกส์ ชนิดต่างๆ เช่น ไดโอด ทรานซิสเตอร์ ไดแอก ไทรแอก เป็นต้น



รูปที่ 2.17 วงจรพื้นฐานของ Opto isolator

จากระบบสวิทช์ทรานซิสเตอร์ สวิทช์รีเลย์ และ Opto isolator เราสามารถประยุกต์ใช้กับโครงงานนี้ได้โดยให้เป็นส่วนควบคุมอุปกรณ์ไฟฟ้าซึ่งรีเลย์นั้นสามารถนำมาต่อควบคุมวงจรภายนอกได้หลากหลายแบบเช่น มอเตอร์ หรือ อุปกรณ์ไฟฟ้าที่ใช้กำลังไฟสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

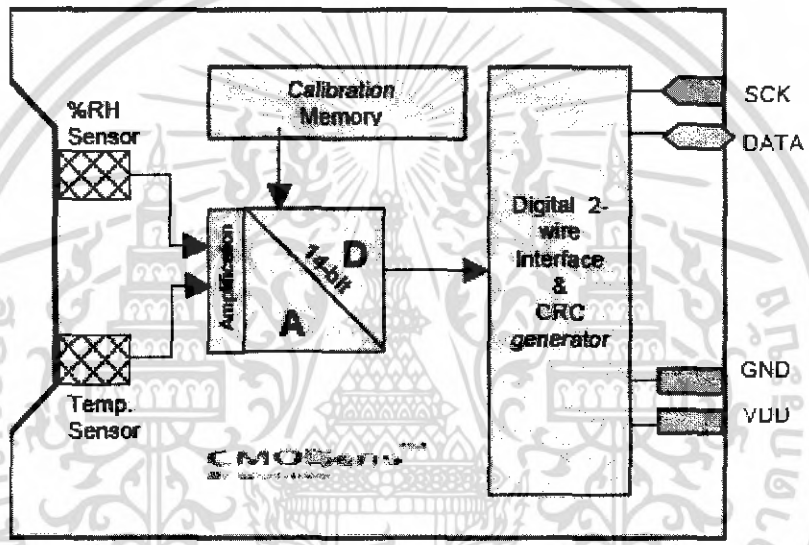
บทที่ 3

การคำนวณและการสร้าง

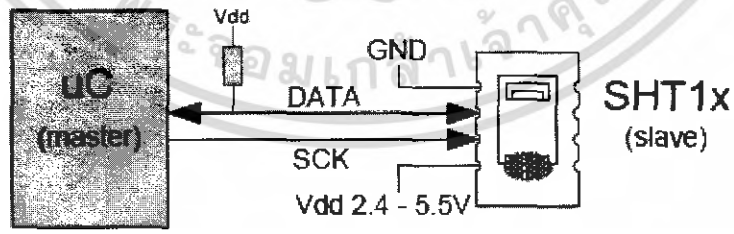
3.1 อุปกรณ์ตรวจสอบสัญญาณ

3.1.1 อุณหภูมิและ ความชื้น

การต่อวงจรต้องทำการเชื่อมต่อเซนเซอร์เข้ากับไมโครคอนโทรลเลอร์เพื่อรับสัญญาณ Serial input clock (SCK) มาเป็นสัญญาณควบคุมการทำงาน จากนั้นเซนเซอร์ทั้งสองก็จะทำการส่งผลเป็นสัญญาณอนาล็อกมาทำการแปลงเป็น A/D ก่อนส่งออกที่ขา DATA ไปยังไมโครคอนโทรลเลอร์ในรูปของสัญญาณดิจิตอล การทำงานทั้งหมดจะอยู่ภายในตัวเซนเซอร์ดังบล็อกไดอะแกรม

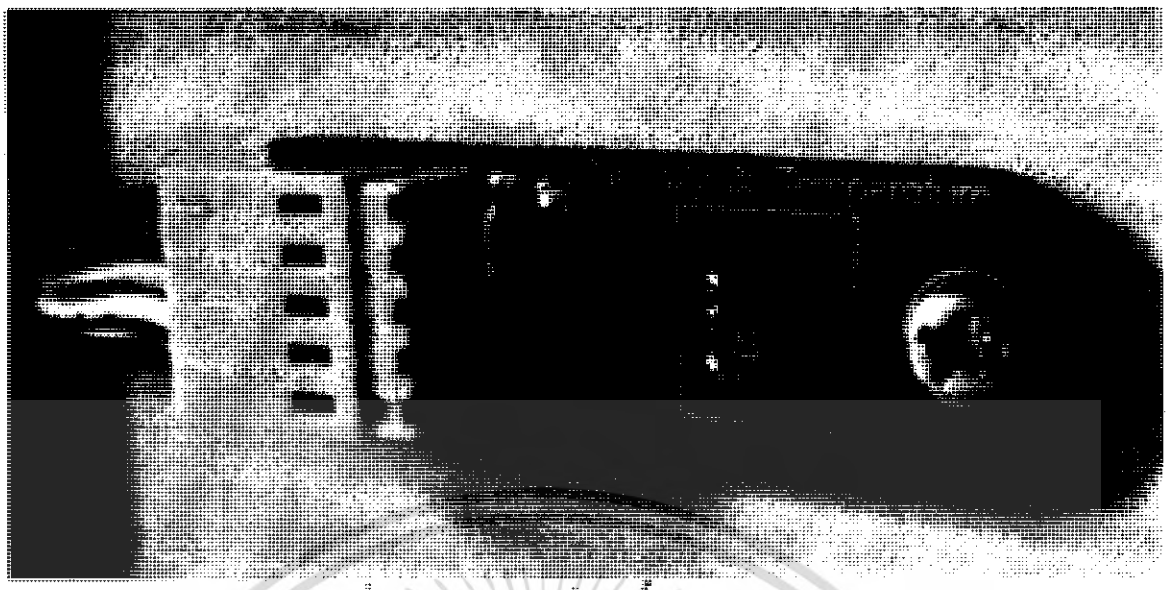


รูปที่ 3.1 บล็อกไดอะแกรมของเซ็นเซอร์อุณหภูมิและความชื้น



รูปที่ 3.2 แสดงการต่อเซนเซอร์วัดอุณหภูมิกับความชื้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงตำแหน่งของเซนเซอร์วัดอุณหภูมิ

หลังจากที่ไมโครคอนโทรลเลอร์รับค่าเข้ามาเราจะทำการแปลงค่าที่เซนเซอร์ส่งมาให้เป็นค่าทางฟิสิกส์โดยใช้สูตรตามคู่มือของเซนเซอร์

ค่าอุณหภูมิจะหาได้จาก

$$Temperature = d_1 + d_2 \bullet SO_T \tag{3.1}$$

ตารางที่ 3.1 Temperature conversion coefficient

VDD	d_1 [°C]	d_1 [°F]
5V	-40.00	-40.00
4V	39.75	39.50
3.5V	-39.66	-39.35
3V	-39.60	-39.28
2.5V	-39.55	-39.23

SO_T	d_2 [°C]	d_2 [°F]
14bit	0.01	0.018
12bit	0.04	0.072

ค่าความชื้นจะหาได้จาก

$$RH_{linear} = c_1 + c_2 \bullet SO_{RH} + c_3 \bullet SO_{RH}{}^2 \tag{3.2}$$

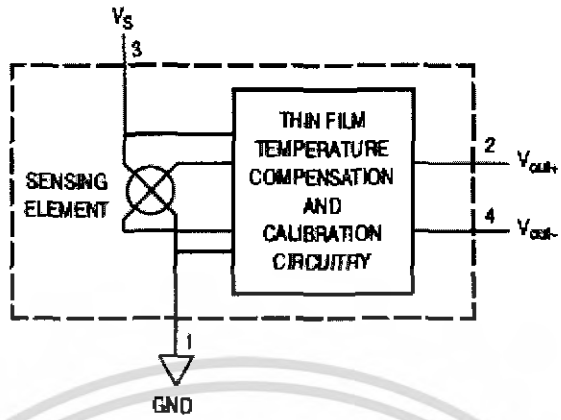
ตารางที่ 3.2 Humidity conversion coefficient

SO_{RH}	c_1	c_2	c_3
12 bit	-4	0.0405	$-2.8 * 10^{-6}$
8 bit	-4	0.648	$-7.2 * 10^{-4}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

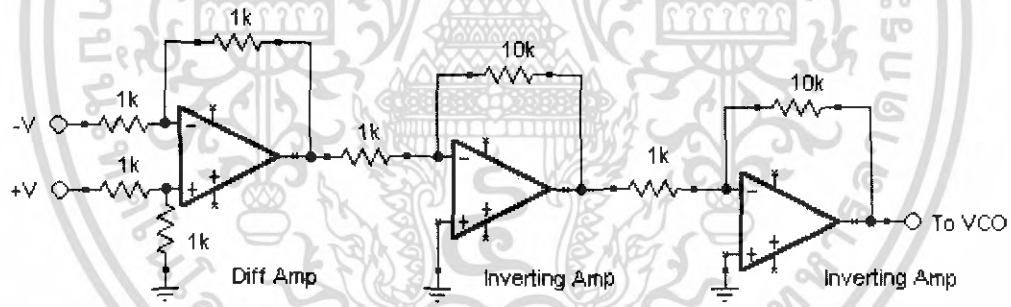
3.1.2 ความดัน

ไอซีตรวจสอบความดันจะมีวงจรภายในดังรูป



รูปที่ 3.4 วงจรภายในไอซีตรวจสอบแรงดัน

โดยเอาท์พุตของสัญญาณจะเป็นไฟตรงที่มีค่าอยู่ในย่าน 20mV-70mV โดยในการนำเอาไอซีนี้ไปใช้งานจริงเราจะนำเอาท์พุตที่ได้ไปขยายเพื่อเป็นอินพุตเพื่อป้อนให้แก่ วงจรวีซีโอ หรือ โวลต์เตจคอนโทรลออสซิเลเตอร์ ไปผลิตสัญญาณสี่เหลี่ยมที่มีความเปลี่ยนแปลงตามค่าสัญญาณไฟตรงจากไอซีวัดความดัน



รูปที่ 3.5 วงจร Op-Amp ขยายแรงดัน 100 เท่า

อัตราการขยายของวงจร Op-Amp นี้คือ 100 เท่า สามารถคำนวณได้จากสมการดังนี้

- พิจารณาที่โหนด a และโหนด b

$$\frac{Va - Vi1}{R1} + \frac{Va}{R2} = 0$$
$$\frac{Vb - Vi2}{R3} + \frac{Vb - Vo}{R4} = 0 \qquad (3.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ย้ายสมการหาค่า V_o ที่โหนดb

$$Vo = -\frac{R4}{R3}Vi2 + (1+\frac{R4}{R3})Vb \tag{3.4}$$

- ย้ายสมการหาค่า V_a ที่โหนด a

$$Va = \frac{R2}{R1+R2}Vi1 \tag{3.5}$$

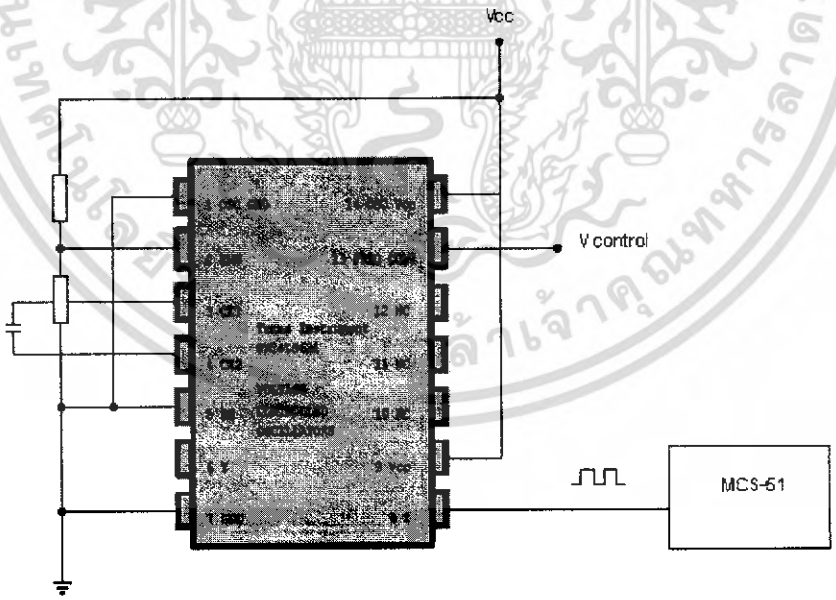
- จากคุณสมบัติของออปแอมป์ V_a=V_b

$$Va = Vb = \frac{R2}{R1+R2}Vi1 \tag{3.6}$$

- จะได้ค่า V_o ดังสมการ

$$Vo = -\frac{R4}{R3}Vi2 + (1+\frac{R4}{R3})(\frac{R2}{R1+R2})Vi1 \tag{3.7}$$

จากส่วนของวงจรขยายสัญญาณ สัญญาณที่ได้จะนำมาเป็นตัวควบคุมในส่วนของภาคผลิตความถี่โดยใช้ IC VCO เพื่อส่งให้กับไมโครคอนโทรลเลอร์ในการประมวลผลความถี่ต่อไป

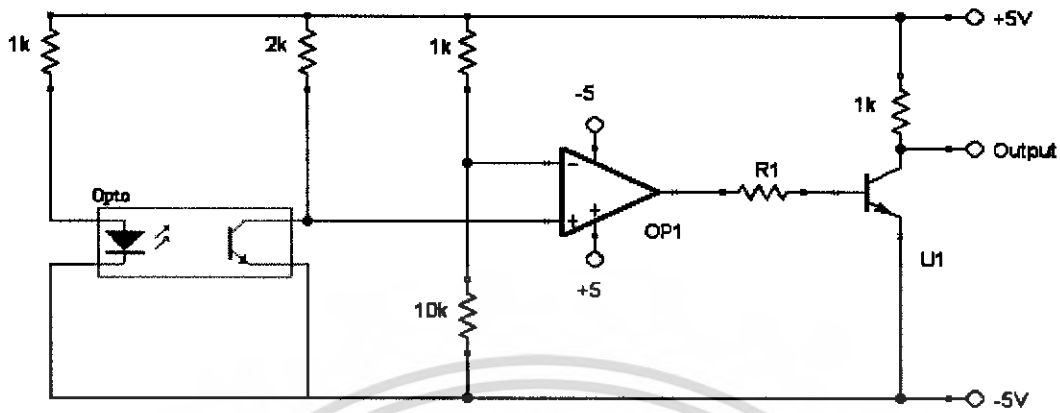


รูปที่ 3.6 แสดงวงจรโวลท์เตจคอนโทรลออสซิลเลเตอร์ ที่ใช้ในการสร้างสัญญาณพัลส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 ลม

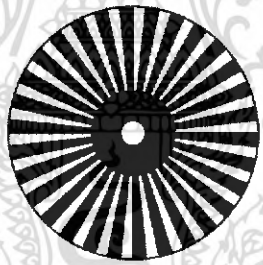
ในส่วนของวงจรตรวจสอบลม เราจะต่อวงจรตามรูป



รูปที่ 3.7 วงจรตรวจสอบความเร็วลม

ขั้นตอนการทำตัวรับลม

1.ออกแบบแผ่นเพลทตัดแสงเป็นรูปวงกลมโดยแบ่งส่วนสลับสีทั้งหมด 60 ส่วน ดังรูปด้านล่าง ให้มีขนาดเส้นผ่านศูนย์กลาง 3 เซ็นติเมตร โดยการตัดสติกเกอร์ให้เป็นรูปด้านล่างแล้วนำมาติดเข้ากับแผ่นพลาสติกใส แล้วทำการพ่นสีดำ ทำการตัดให้เป็นวงกลมตามแบบ จากนั้น ก็จะได้ส่วนของตัดแสงของวงจรทำงาน



รูปที่ 3.8 แผ่นตัดแสง

2.นำแบริ่งขนาดเส้นผ่านศูนย์กลาง 0.8 เซนติเมตร 2 ชิ้นมาตอกอัดลงบนแกนหมุนของส่วนเครื่องวัดลมซึ่งใช้แกนหมุนสแตนเลส มีเส้นผ่านศูนย์กลาง 0.8 เซนติเมตร

3.ในส่วนของลูกถ้วยหมุนนั้นออกแบบโดยใช้ลูกปิงปอง เส้นผ่านศูนย์กลาง 3.8 เซนติเมตร โดยทำการผ่าครึ่ง แล้วทำการเจาะรู บริเวณตำแหน่งขอบของลูกปิงปอง เพื่อติดตั้งแขนของแกนหมุน ที่ทำจากท่อพลาสติก ขนาดเส้นผ่านศูนย์กลาง 0.5 เซนติเมตร ความยาว 3.5 เซนติเมตร โดยทำอย่างละสามชิ้น

4.ในการทำตัวยึดแขนของลูกถ้วยหมุนนั้น ใช้ฝาขวดน้ำพลาสติกมาเจาะรูตรงกลางฝา เพื่อยึดเข้ากับแกนหมุน และเจาะรูด้านข้างเป็นจำนวนสามรู ให้แต่ละรูทำมุมกัน 120 องศา เพื่อยึดแขนของลูกถ้วยทั้งสาม

5.ทำการติดตั้งแผ่นตัดแสงเข้ากับแกนหมุนโดยการเจาะรูแผ่นตัดแสง แล้วนำมาติดตั้งกับแกนหมุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

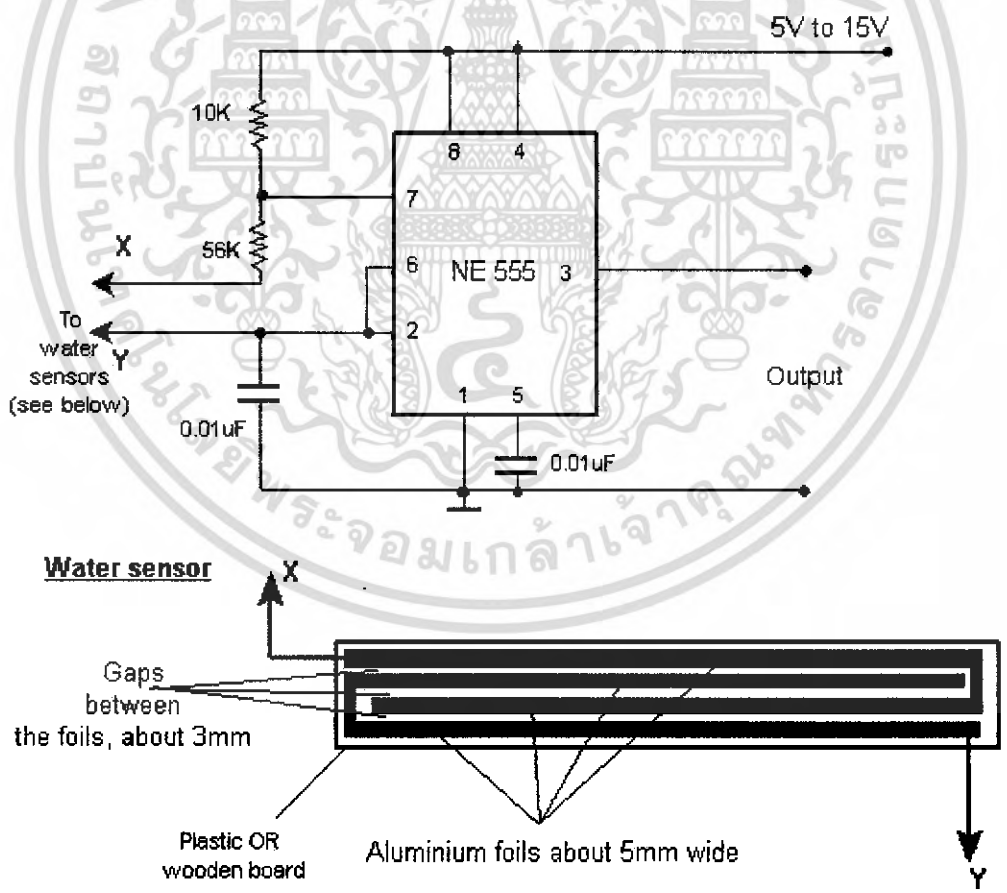
6.ยึดแบริ่งของแกนหมุนเข้ากับฐานของแผ่นวงจรที่ติดตั้ง ออปโต้ ตรวจจับแสง โดยให้แผ่นตัดแสงอยู่ระหว่างกลางของ ออปโต้ตรวจจับแสง

7.ประกอบแกนของลูกถ้วยเข้ากับฝาครอบ โดยใส่ตามจุดเจาะที่ได้กำหนดไว้แล้ว จากนั้นประกอบส่วนของลูกถ้วยหมุนเข้ากับแกนหมุนที่ทำไว้ก็เป็นอันเสร็จสมบูรณ์ สามารถประกอบบนฝากล่องเข้ากับส่วนแผ่นพลาสติกแสง ก่อนประกอบเข้ากับส่วนวงจรที่อยู่ภายในกล่อง

เมื่อประกอบอุปกรณ์ในส่วนเครื่องวัดความเร็วลมเรียบร้อยแล้วก็จะได้ส่วนของเครื่องวัด

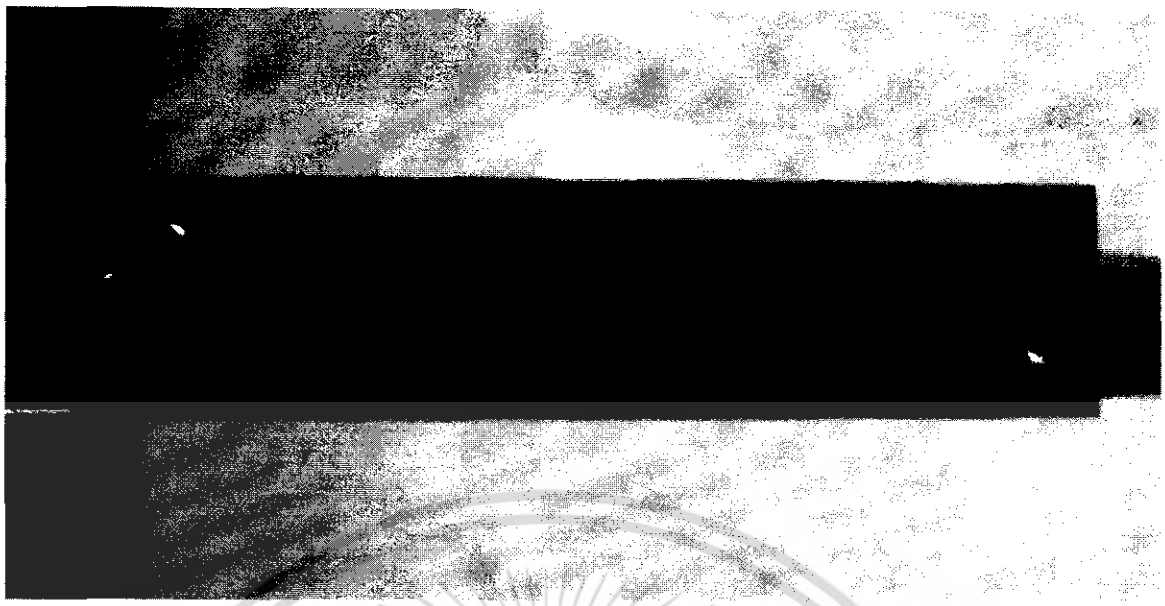
3.1.4 ฝน

ใช้การทำงานร่วมกันระหว่าง ไอซีไทม์เมอร์ 555 กับแผ่นทองแดงรับฝน โดยอาศัยหลักการที่ให้แผ่นทองแดงทำหน้าที่เสมือนตัวต้านทาน เมื่อน้ำตกลงมาบนแผ่นทองแดง จะส่งผลให้เกิดความต้านทานที่เปลี่ยนไป ทำให้วงจรAstable Multivibator ผลิตความถี่ที่แต่งกันออกมา ขณะที่ยังไม่มีน้ำจะไม่ผลิตความถี่ออกมาออกมาเพราะแผ่นฟอยส์เปรียบเสมือนว่ามีความต้านทานสูงมาก แต่เมื่อมีน้ำตกลงมาจะทำให้แผ่นทองแดงเสมือนว่ามีความต้านทานลดต่ำลง ทำให้สามารถรับรู้ได้ว่าเกิดฝนตกหรือไม่ วงจรทำงานดังรูป



รูปที่ 3.9 แสดงการต่อวงจรเซนเซอร์เข้ากับวงจรอะสเตเบิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงแผงเซนเซอร์น้ำฝน

จากการออกแบบวงจรกำเนิดความถี่เรา ออกแบบวงจรกำเนิดความถี่ให้มีความถี่สูงสุดไม่สูงมากนัก ประมาณ 1-1.5 KHz โดยการหาค่าความต้านทานต่ำสุดของเซ็นเซอร์ตรวจสอบฝนมาพิจารณาโดยใช้สมการ

$$R_2 = \frac{T_{off}}{0.7C_1} \tag{3.8}$$

$$R_1 = \frac{T_{on}}{0.7C_1} - R_2 \tag{3.9}$$

$$F = \frac{1}{T} = \frac{1}{T_{on} + T_{off}} = \frac{1.44}{(R_1 + 2R_2)C_1} \tag{3.10}$$

จากสมการR2 เราจะได้จากการวัด ส่วนค่าอื่นๆเราจะกำหนดตามความเหมาะสม โดยให้ T-on และ T-off มีค่าเท่ากัน

การสร้างตัวตรวจวัดฝน ทำโดยการนำแผ่นทองแดงมากัดลายปริ้นให้เป็นดังรูปที่ 3.10

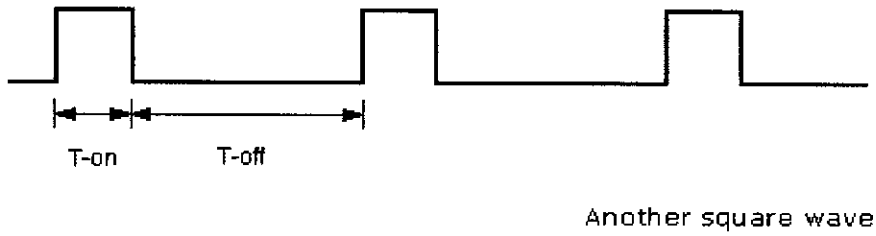
การคำนวณหา T-on, T-off และความถี่ มีการคำนวณดังนี้

$$T_{on} = 0.7(R_1 + R_2)C_1 \tag{3.11}$$

$$T_{off} = 0.7R_2C_1 \tag{3.12}$$

$$F = \frac{1}{T} = \frac{1}{T_{on} + T_{off}} = \frac{1.44}{(R_1 + 2R_2)C_1} \tag{3.13}$$

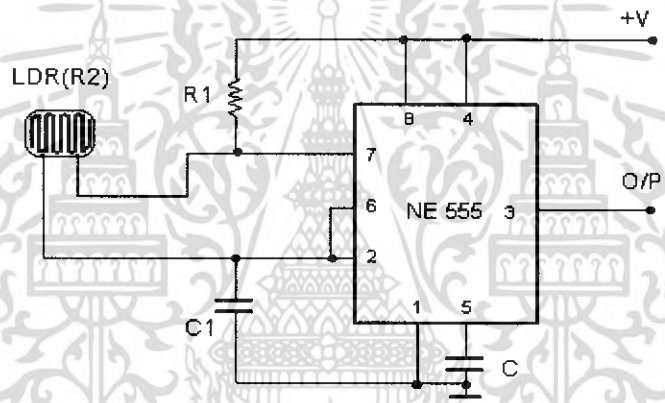
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงช่วงเวลา T-on และ T-off

3.1.5 แสงแดด

ใช้หลักการเดียวกับวงจรตรวจสอบฝน โดยการนำ LDR มาต่อที่ตำแหน่ง R2 และทำการคำนวณหาอุปกรณ์ต่อร่วม โดยความถี่เอ้าท์พุทประมาณ 1-1.5 kHz และทำการต่อดังรูป



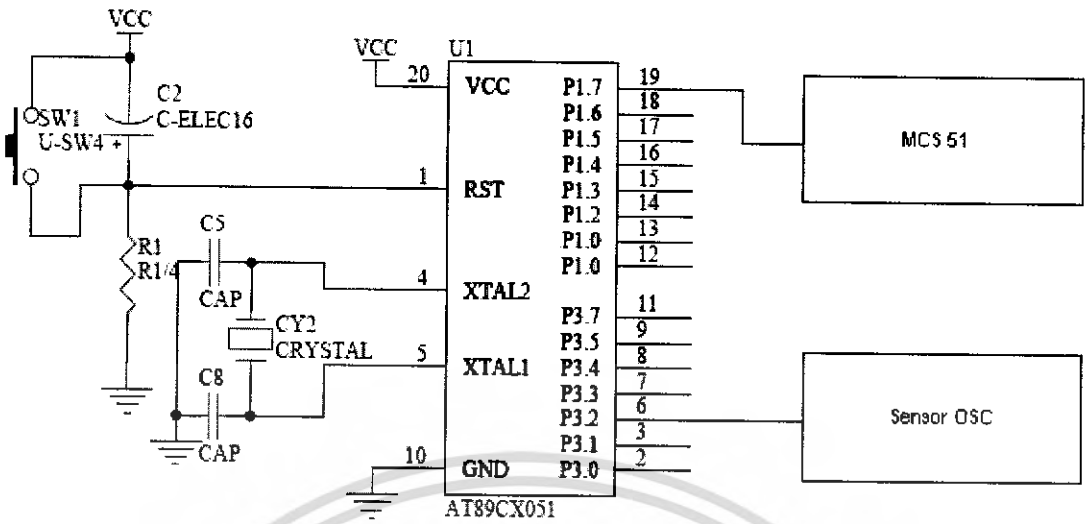
รูปที่ 3.12 การต่อ LDR เข้ากับ IC 555 เพื่อเป็น วงจร อะสเตเบิล มัลติไวเบรเตอร์

ในการพิจารณาการสร้างความถี่ที่ต้องการ ใช้หลักการพิจารณาความถี่เช่นเดียวกับส่วนตรวจสอบฝน โดยอุปกรณ์ต่อร่วมต่างๆ พิจารณาจากสมการเดียวกันกับเซนเซอร์ฝน

3.2 ส่วนนับและประมวลผลความถี่

ในส่วนนี้ จะใช้ไมโครคอนโทรลเลอร์ เบอร์ AT89C2051 ซึ่งเป็นไมโครคอนโทรลเลอร์ ตระกูล MCS-51 แต่มีขาเพียง 20 ขา ซึ่งเพียงพอต่อการทำงาน ในส่วนของการนับความถี่ โดยเมื่อได้ค่าความถี่ ก็จะส่งไปมัลติเพล็กยังไมโครคอนโทรลเลอร์ตัวต่อไป

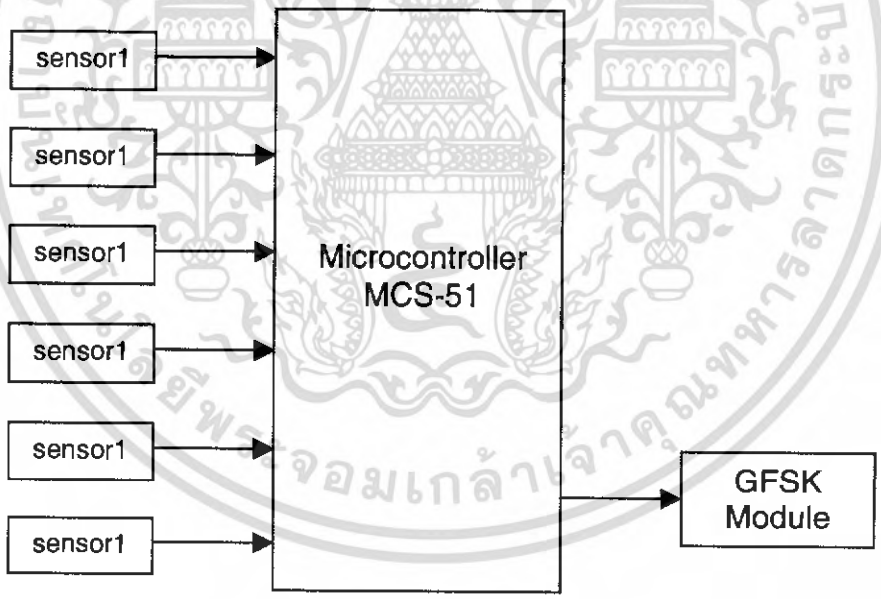
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงการต่อใช้งานในส่วนของการประมวลผลความถี่

3.3 ภาคมัลติเพล็กซ์ และ ดีมัลติเพล็กซ์

3.3.1 การมัลติเพล็กซ์โดยใช้ไมโครคอนโทรลเลอร์



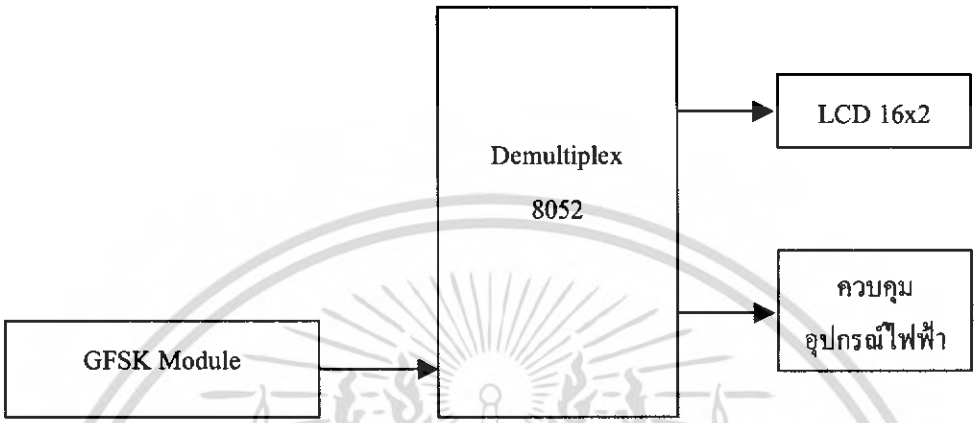
รูปที่ 3.14 แสดงการมัลติเพล็กซ์โดยใช้ไมโครคอนโทรเลอร์

ไมโครคอนโทรลเลอร์จะเป็นตัวรับสัญญาณต่างๆจากเซ็นเซอร์ทั้ง 6 แล้วนำมามัลติเพล็กซ์ทางลำดับเวลา (Time Division Multiplex:TDM) จากนั้นจะถูกส่งผ่านไปยังภาค FSK Modulator โดยเราจะใช้พอร์ท3 เป็น output โดยจะถูกส่งออกเป็นแบบสื่อสารอนุกรมซึ่งจะใช้ขา ในไมโครคอนโทรลเลอร์คือ TxD p3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งในการมัลติเพล็กซ์ข้อมูล จะต้องมีซิงโครไนซ์เฟรมเพื่อที่จะสามารถระบุข้อมูลได้อย่างถูกต้องที่ด้านรับ

3.3.2 การดีมัลติเพล็กซ์โดยใช้ไมโครคอนโทรลเลอร์



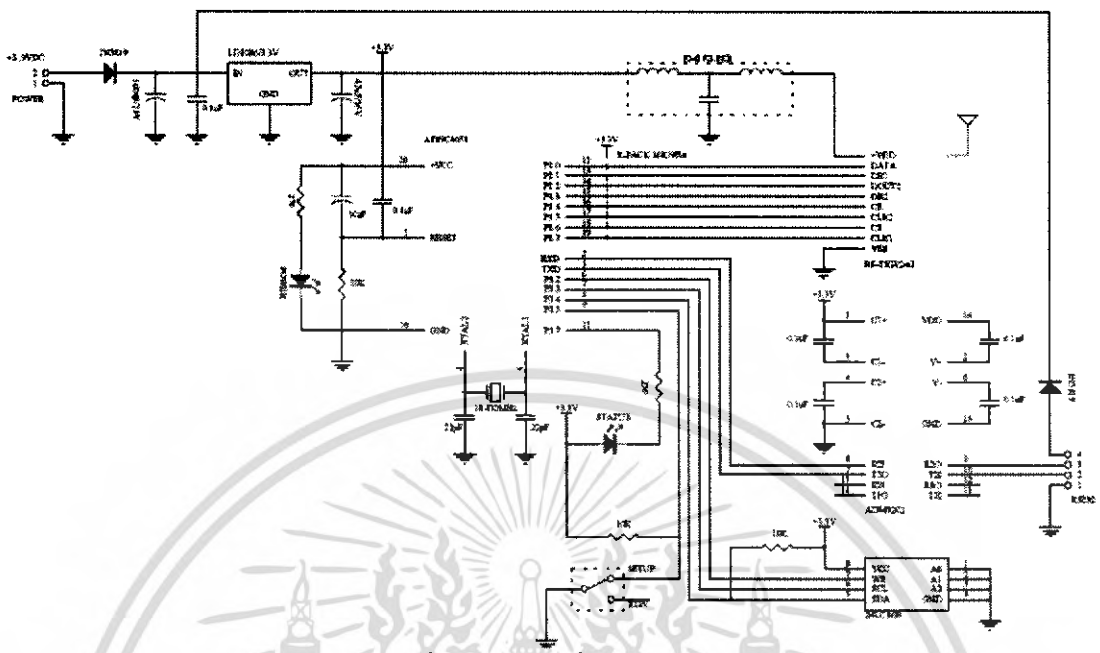
รูปที่ 3.15 แสดงการดีมัลติเพล็กซ์โดยใช้ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์จะทำหน้าที่ดีมัลติเพล็กซ์สัญญาณที่ถูกส่งมาจากภาคส่ง โดยรับสัญญาณจากภาค FSK Demodulation ซึ่งข้อมูลที่รับได้จะอยู่ใน SBUF ที่ขาคอนไมโครคอนโทรลเลอร์ RxD p3.0 เป็นขาสื่อสารพอร์ทอนุกรมทางด้านรับ จะต้องได้รับซิงโครไนซ์เฟรมเป็นอย่างแรก จึงจะสามารถแยกข้อมูลแต่ละเซนเซอร์ที่มาจากภาคส่งได้อย่างถูกต้อง หลังจากไมโครคอนโทรเลอร์ทำการดีมัลติเพล็กซ์และประมวลผลข้อมูลก็จะส่งสัญญาณออกไปอีก 2 สัญญาณ ดังนี้

1. ส่งไปยังจอแสดงผล LCD เพื่อให้ผู้ใช้งานที่อยู่ใกล้เครื่องรับได้ทราบค่าต่างๆจากเซนเซอร์
2. ส่งผ่านไปยังส่วนควบคุมอุปกรณ์ไฟฟ้า โดยส่วนนี้ผู้ใช้สามารถนำไปประยุกต์ใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 เครื่องรับ - ส่งวิทยุ RF



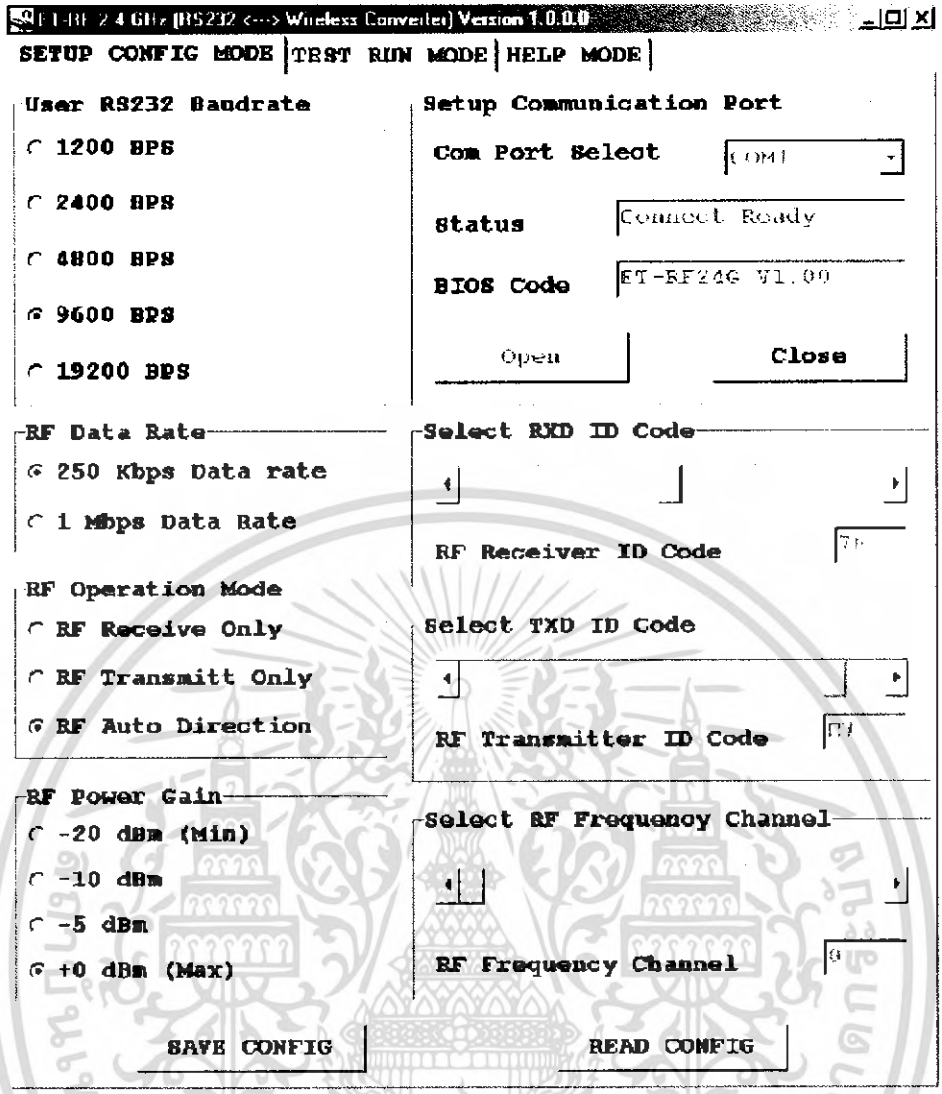
รูปที่ 3.16 วงจรเครื่องรับ-ส่งวิทยุ 2.4GHz

การทำงานของเครื่องแปลงสัญญาณนั้นจะแบ่งออกเป็น 2 โหมด ด้วยกัน โดยการกำหนดโหมดการทำงานของเครื่องรับส่งนั้นจะกระทำผ่าน Switch เลือกโหมด โดยการเลือกโหมดการทำงานนั้นจะต้องกระทำให้เสร็จเรียบร้อยก่อนการจ่ายไฟให้กับเครื่องแปลงสัญญาณด้วยเสมอ เนื่องจากการทำงานของเครื่องแปลงสัญญาณนั้นจะทำการตรวจสอบโหมดการทำงานของเครื่องจาก Switch เลือกโหมด เฉพาะในช่วงของการจ่ายไฟเลี้ยงให้เครื่องเริ่มต้นทำงานครั้งแรก (Power-ON) เท่านั้น ซึ่งการเปลี่ยนแปลงตำแหน่งการทำงานของ Switch เลือกโหมด หลังจากทำการจ่ายไฟให้ไปแล้ว จะไม่มีผลต่อการทำงานของเครื่องแต่อย่างใด โดยการทำงานนั้นจะมีอยู่ด้วยกันสองโหมดคือ

1. Run Mode
2. Setup Mode

ความแตกต่างของโหมดทั้งสองคือ Run mode จะเป็นโหมดที่เครื่องรับ-ส่งทำงานในส่วนของการรับและส่งข้อมูล แต่ในส่วนของ Setup mode นั้นคือการเปลี่ยนแปลงค่าภายในของโปรแกรมที่ได้กำหนดไว้ตั้งแต่ต้นว่าจะให้เครื่องแปลงสัญญาณทำหน้าที่เป็น Receiver only หรือว่าเป็น Transmitter only โดยวิธีการตั้งค่าต่างๆ นั้นเราจะใช้โปรแกรมในการตั้งค่า เมื่อมีการตั้งค่าแล้วโปรแกรมจะสามารถทำงานตามค่าที่ตั้งไว้ได้นับตั้งแต่ครั้งแรกที่จ่ายไฟเลี้ยงให้ตัววงจร และแม้ว่าจะทำการหยุดจ่ายไฟ หรือว่ามีการปิดเปิดเครื่องในครั้งต่อไปก็จะไม่สามารถเปลี่ยนแปลงข้อมูลที่ถูกตั้งไว้แล้ว เว้นแต่จะทำการเซ็ตโหมดการทำงานมาที่ Setup mode อีกครั้งแล้วทำการเปลี่ยนแปลงค่าภายใน จึงจะถือว่าข้อมูลนั้นได้เปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 แสดงรูปแบบการตั้งค่าภายในโดยใช้โปรแกรมในการตั้งค่า

User RS232 Baudrate ใช้สำหรับกำหนดค่าความเร็วในการรับส่งข้อมูลทางด้าน RS232 ของตัวเครื่อง ในขณะที่ทำงานอยู่ใน Run Mode ซึ่งสามารถกำหนดได้ 5 ค่าคือ

- 1200 BPS
- 2400 BPS
- 4800 BPS
- 9600 BPS
- 19200 BPS

RF Data Rate ใช้สำหรับกำหนดความเร็วในการรับส่งข้อมูลทางด้าน RF ของเครื่องแปลงสัญญาณซึ่งจะต้องกำหนดให้เครื่องแปลงสัญญาณทุกๆตัว ที่จะนำมาใช้ติดต่อสื่อสารกัน มีค่าอัตราความเร็วในการรับส่งข้อมูลด้าน RF หรือ RF Data Rate นี้มีค่าเท่ากันทั้งหมด ซึ่งถ้ากำหนดค่าความเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กันจะไม่สามารถรับส่งข้อมูลกันได้ ซึ่งค่าอัตราความเร็วในการส่งข้อมูลนี้จะมีผลต่อระยะทางการรับส่งข้อมูลด้วย ซึ่งถ้าใช้ความเร็วในการส่งสูง (1Mbps) จะทำให้รัศมีการรับส่งข้อมูลได้ระยะทางสั้นลง แต่ถ้าใช้ความเร็วในการรับส่งข้อมูลที่ช้าลง (250Kbps) จะทำให้ได้รัศมีการรับส่งไกลขึ้น โดยค่า RF Data Rate สามารถกำหนดได้ 2 ค่า คือ

- 250 Kbps
- 1 Mbps

RF Operation Mode ใช้สำหรับกำหนดโหมดการทำงานของ ET-RF24G V1.0 ซึ่งสามารถกำหนดหน้าที่การทำงานได้ 3 แบบ ด้วยกันคือ

1. RF Receive Only เป็นการกำหนดให้ เครื่องแปลงสัญญาณทำหน้าที่เป็นฝ่ายรอรับข้อมูลทางด้านRF เพื่อเปลี่ยนเป็นข้อมูลแบบ RS232 และส่งออกไปทางด้านขา TX ของ RS232 ตลอดเวลา

2. RF Transmit Only เป็นการกำหนดให้เครื่องแปลงสัญญาณทำหน้าที่เป็นฝ่ายรอรับข้อมูลทางด้าน RS232 จากขา RX เพื่อเปลี่ยนเป็นข้อมูลแบบ GFSK และส่งออกไปทางด้าน RF ตลอดเวลา

3. RF Auto Direction เป็นการกำหนดโหมดการทำงานแบบ Half Duplex 2 ทิศทาง ซึ่งสามารถสลับโหมดการทำงานระหว่างการรับและส่งข้อมูลได้เองโดยอัตโนมัติ โดยในโหมดการทำงานนี้เครื่องแปลงสัญญาณจะรอตรวจสอบข้อมูลทั้งจากด้าน RS232 และด้าน RF อยู่ตลอดเวลาโดยถ้าได้รับข้อมูลจากด้าน RS232 ก็จะทำการแปลงแล้วส่งออกทางด้าน RF จากนั้นก็จะกำหนดให้ด้าน RF กลับมาเป็นฝ่ายรอรับข้อมูลตามเดิม และเมื่อได้รับข้อมูลจากด้าน RF ก็จะแปลงเป็นข้อมูลแล้วส่งออกไปทางด้าน RS232 โดยอัตโนมัติ

RF Power Gain เป็นการกำหนดกำลังส่งของวงจร RF Power ที่ใช้ในการส่งข้อมูล โดยค่า +0dBmเป็นค่ากำลังส่งสูงสุด ส่วน –20dBm เป็นค่ากำลังส่งต่ำสุด โดยสามารถกำหนดได้ 4 ระดับคือ

- -20dBm (กำลังส่งต่ำสุด)
- -10dBm
- -5dBm
- +0dBm (กำลังส่งสูงสุด)

RXD ID Code เป็นรหัส ID Code ของเครื่องแปลงสัญญาณ ในโหมดของการรับข้อมูลจาก RF โดยเมื่อเครื่องแปลงสัญญาณด้านส่งจะทำการส่งข้อมูลออกไปทาง RF นั้นจะมีการระบุหมายเลข IDCode ของด้านรับรวมไปกับชุดข้อมูลด้วยเสมอ โดยเมื่อเครื่องแปลงสัญญาณที่อยู่ทางด้านรับทำการรับข้อมูลจากด้าน RF ได้ อันดับแรกมันจะทำการเปรียบเทียบรหัส ID Code ที่รวมมากับข้อมูลที่รับมาได้ว่าตรงกับรหัสของ RXD ID Code ที่กำหนดไว้ในตัวมันหรือไม่ ซึ่งถ้าถูกต้องก็จะแยกเอาเฉพาะส่วนของข้อมูลที่รับเข้ามาได้เพื่อเปลี่ยนเป็นข้อมูลแบบ RS232 แล้วส่งออกไปทางด้าน TX ของ RS232แต่ถ้ารหัส ID Code ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

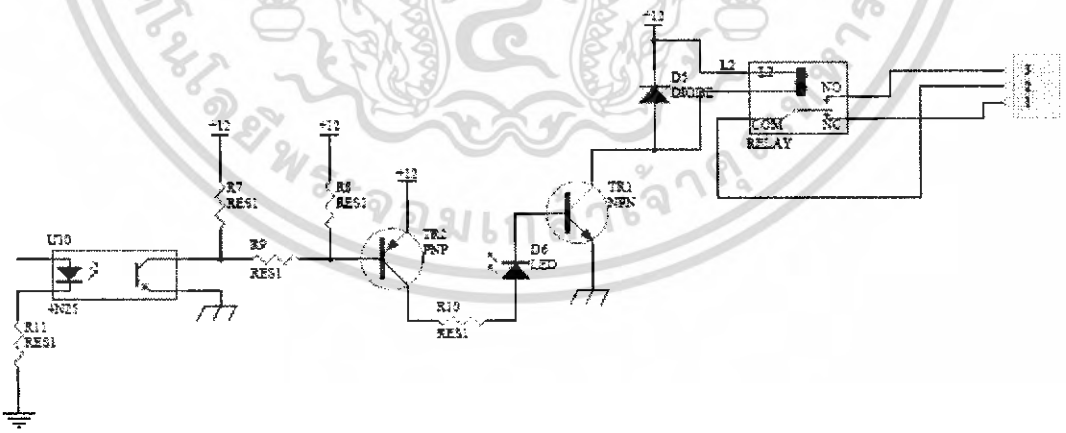
รับมาได้ไม่ตรงกับรหัส RXD ID Code ที่กำหนดไว้ เครื่องแปลงสัญญาณจะทิ้งข้อมูลชุดนั้นไปทันที โดยค่า RXD ID Code นั้นสามารถกำหนดได้ 256 ค่าในรูปแบบของเลขฐานสิบหก (00H-FFH)

TXD ID Code เป็นรหัส ID Code ปลายทางที่จะส่งข้อมูลไปหา โดยที่เครื่องแปลงสัญญาณที่ถูกกำหนดให้ทำหน้าที่เป็นฝ่ายส่งข้อมูลนั้น เมื่อมันสามารถรับข้อมูลจาก RS232 ได้แล้ว มันจะทำการนำเอาข้อมูลนั้นไปเข้ารหัสรวมกับ TXD ID Code ที่กำหนดไว้ แล้วส่งออกไปทางด้าน RF โดยรหัสของ TXD ID Code นี้หมายถึง รหัส RXD ID Code ของฝ่ายรับที่ต้องการส่งข้อมูลไปหานั่นเอง โดยค่า TXD ID Code นั้นสามารถกำหนดได้ 256 ค่าในรูปแบบของเลขฐานสิบหก (00H-FFH)

RF Frequency Channel เป็นการกำหนดค่าของช่องความถี่ที่จะใช้ในการรับส่งข้อมูลกัน โดยสามารถเลือกกำหนดช่องความถี่ได้สูงสุดมากถึง 125 ช่อง (0-124) โดยการที่เครื่องแปลงสัญญาณ จะทำการรับส่งข้อมูลกันได้นั้นจะต้องกำหนดช่องความถี่ที่ตรงกัน และ ใช้อัตราความเร็ว RF Data Rate ที่เท่ากันด้วย ซึ่งที่สามารถเลือกกำหนดช่องความถี่ RF Frequency Channel ได้นั้น จะมีประโยชน์เป็นอย่างมากในกรณีที่มีการใช้งานเครื่องแปลงสัญญาณจำนวนหลายๆกลุ่ม ในบริเวณพื้นที่ใกล้เคียงกัน โดยให้กำหนดช่องความถี่ของกลุ่ม ที่จะสื่อสารข้อมูลร่วมกันไว้ที่ช่องความถี่เดียวกัน ส่วนกลุ่มอื่นๆก็ให้เลือกกำหนดช่องความถี่ที่แตกต่างกันออกไป เพื่อลดปัญหาการรบกวนกัน

3.5 ส่วนควบคุมอุปกรณ์ไฟฟ้า

ในส่วนของการสร้างได้ทำให้ส่วนของการเตรียมความพร้อมโดยใช้วงจรรีเลย์ในการประยุกต์ใช้กับอุปกรณ์ไฟฟ้าต่างๆดังรูป

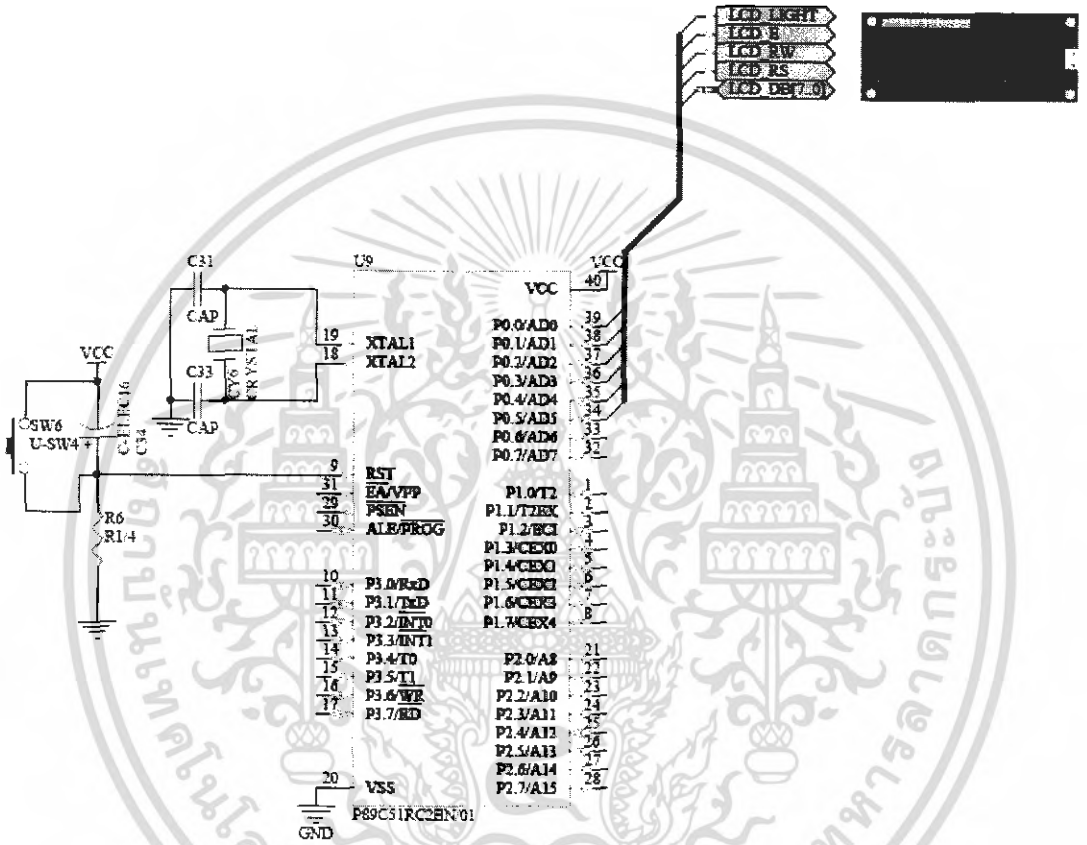


รูปที่ 3.18 แสดงการต่อวงจรรีเลย์สำหรับการควบคุมอุปกรณ์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ส่วนแสดงผลหน้าจอแอลซีดี

ในการต่อส่วน Display LCD เข้ากับคอนโทรลบอร์ด เราต่อเข้ากับขา I/O ของ 8255 โดยที่ได้เซต ให้พอร์ต A และ C เป็นเอาท์พุตพอร์ตเพื่อให้พอร์ต A เป็นพอร์ตข้อมูลคอยส่งข้อมูลต่าง ๆ ให้แก่ display LCD และให้พอร์ตสัญญาณ (Signal Port) ที่จะคอยควบคุมการทำงานต่าง ๆ ของ display LCD นี้ แสดงรายละเอียดการต่อดังรูปที่ 4



รูปที่ 3.19 การต่อ Display LCD เข้ากับชุดคอนโทรลบอร์ด

ขาที่ต่อใช้งานของส่วน Display LCD มีดังนี้

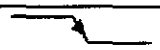
1. RS (register selection) เป็นขาเลือกรีจิสเตอร์ภายในซึ่งมีอยู่ 2 ตัว คือ IR (instruction register) และ DR (data register) โดยถ้าเป็น 1 เป็นการเลือกข้อมูล และถ้าเป็น 0 เป็นการเลือก Instruction

2. R/W (read / write) เป็นตัวเลือกว่าเขียนหรืออ่านข้อมูลจากตัว IC โดยอ่านข้อมูล = 1, เขียนข้อมูล = 0

3. E (enable signal) เป็นขากำหนดสภาพการรับเขียนอ่านข้อมูล แสดงลักษณะสัญญาณทั้ง RS, RW และ E ดังตารางที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 แสดงลักษณะสัญญาณ RS, RW และ E

RS	RW	E	การทำงาน
0	0		เขียนโค้ดคำสั่ง
0	1		อ่านแฟลกและตัวนับแอดเดรส
1	0		เขียนข้อมูล
1	1		อ่านข้อมูล

4. DB0-DB7 เป็นขารับส่งข้อมูลจากตัว IC

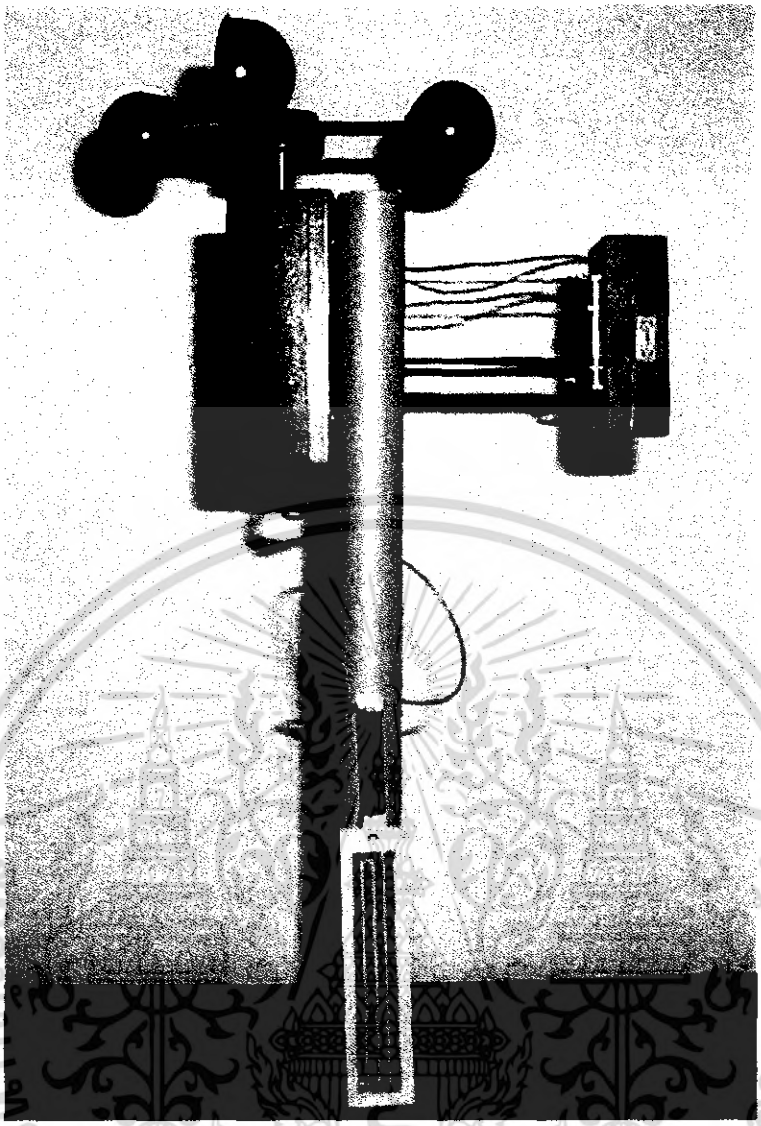
5. VDD ไฟเลี้ยงตัววงจร

6. VSS เป็นขา GND

7. VO เป็นขารับแรงดันในการขับ LCD ให้สว่างหรือมืด

รูปที่ 3.20 แสดงโมดูลภาครับ ส่วนควบคุมอุปกรณ์ไฟฟ้า และแสดงผลผ่านหน้าจอแอลซีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 แสดงเซนเซอร์ลม น้ำฝน และ แสงแดด

รูปที่ 3.22 แสดงโมดูลภาคส่งและส่วนประมวลผลความถี่จากเซนเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 เซนเซอร์ความดันอากาศ

ตารางที่ 4.1 แสดงผลการทดลองความดันอากาศที่อุณหภูมิ 25 องศาเซลเซียสเทียบกับความถี่

ความดัน (PSI)	ความถี่ (Hz)
1	55
1.5	80
2	105
2.5	130
3	152
3.5	181
4	200
4.5	225
5	240
5.5	270
6	300
6.5	330
7	360
7.5	395
8	425
8.5	455
9	490
9.5	525
10	560
10.5	595
11	625
11.5	660
12	695
12.5	725
13	760
13.5	795
14	825
14.5	860
15	895
15.5	920

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

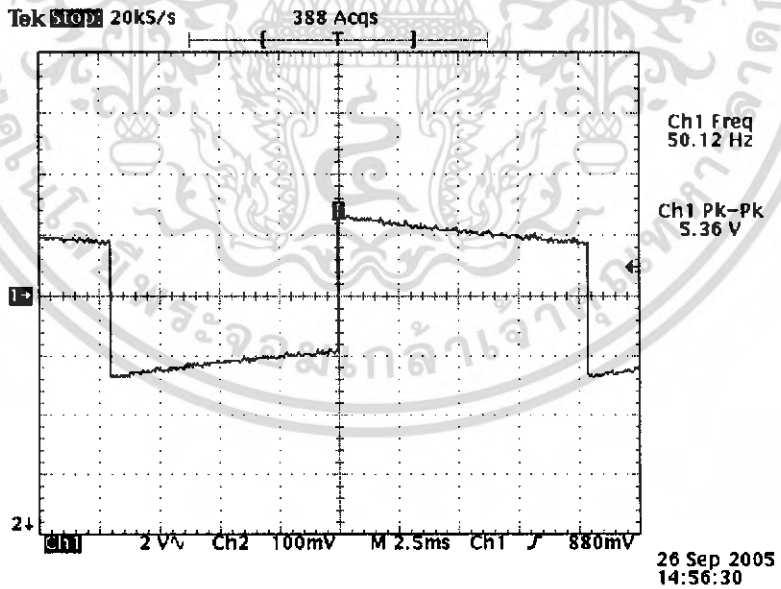
4.2 เซนเซอร์ความเร็วลม

ตารางที่ 4.2 แสดงความสัมพันธ์ระหว่างความเร็วลมและความถี่ของสัญญาณเอาท์พุตที่วัดได้

ความเร็วลม(m/s)	ความถี่ที่วัดได้ (Hz)	ความเร็วลม (m/s)	ความถี่ที่วัดได้ (Hz)
2	24	6	119.5
2.5	36	6.5	128.5
3	48.5	7	141.5
3.5	59.5	7.5	158.5
4	69	8	172.5
4.5	80.5	8.5	190
4.7	84	9	207
5	93.5	9.5	218.5
5.5	103.5	10	234.5

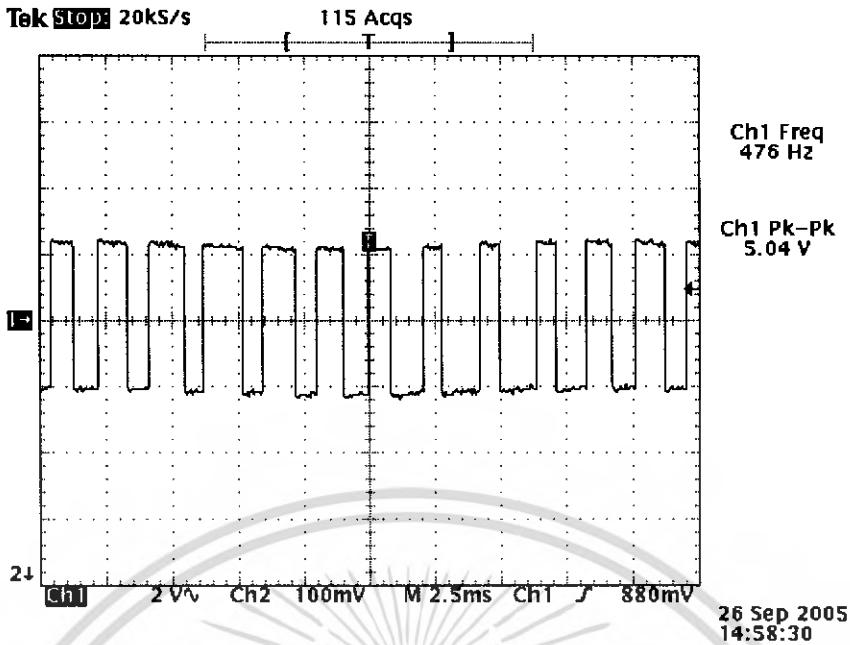
4.3 เซนเซอร์ตรวจจับปริมาณน้ำฝน

ตัวตรวจสอบฝน โดยรูป จะแสดงการแสดงผลของการกำเนิดความถี่ของไอซีไทม์เมอร์ 555 ที่สามารถเปลี่ยนค่าความถี่โดยการเปลี่ยนค่าความต้านทานที่ต่อเข้าที่ขา 7 ของไทม์เมอร์ โดยจะมีปริมาณน้ำฝนจะแปรผกผันกับความต้านทาน

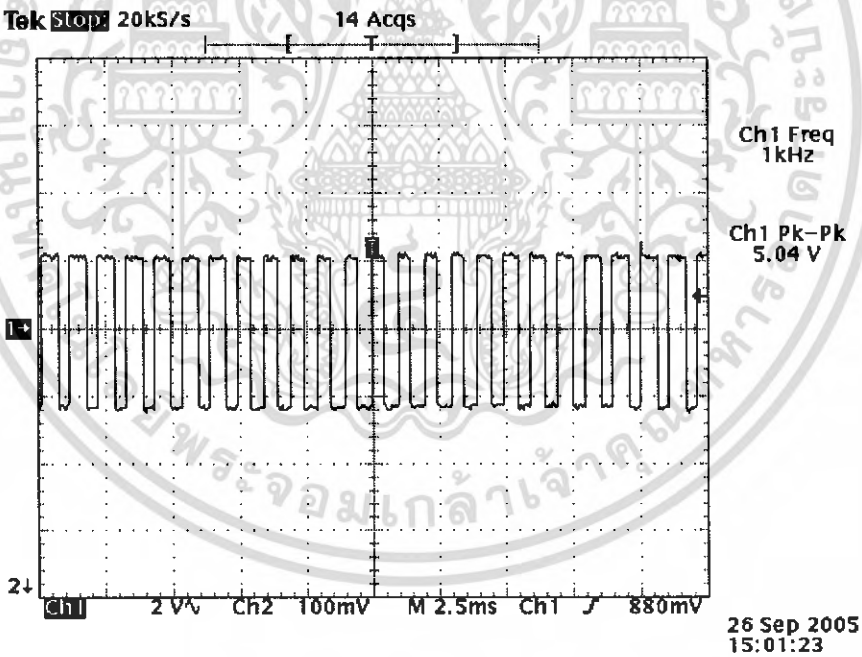


รูปที่ 4.1 แสดงค่าของความถี่เมื่อปริมาณน้ำฝนน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงค่าของความถี่เมื่อปริมาณน้ำฝนปานกลาง



รูปที่ 4.3 แสดงค่าของความถี่เมื่อปริมาณน้ำฝนมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 เซนเซอร์ความเข้มแสง

ตารางที่ 4.3 ค่าความเข้มแสงเทียบกับความถี่ที่สร้างขึ้นของวงจรอะสเตเบิล โดย LDR

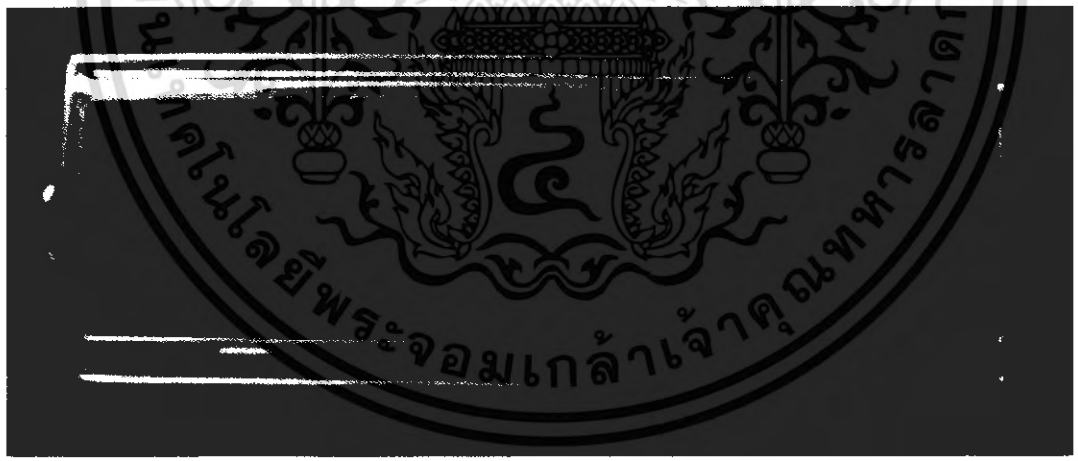
ความเข้มแสง (Lux)	ความถี่ (Hz)
10	30
20	75
30	120
40	170
50	220
60	270
70	300
80	340
90	380
100	410
110	440
120	470
130	490
140	510
150	530
160	550
170	555
180	570
190	585
200	600
210	610
220	620
230	630
240	640
250	655
260	660
270	670
280	673
290	677
300	680

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเข้มแสง (Lux)	ความถี่ (Hz)
320	690
340	710
360	720
380	730
400	740
420	750
440	760
460	765
480	770
500	780
520	785
540	790
560	795
580	800
600	805
620	810
640	813
660	815
680	820
700	825
720	828
740	830
760	835
780	840
800	842
820	843
840	845
860	847
880	849
900	853
920	855
940	857

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเข้มแสง (Lux)	ความถี่ (Hz)
960	859
980	861
1000	863
1250	880
1500	893
1750	902
2000	910
2250	916
2500	920
2750	924
3000	928
3500	933
4000	937
4100	938



รูปที่ 4.4 แสดงผลบนหน้าจอแอลซีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และบทสรุป

โครงงานเครื่องควบคุมอุปกรณ์อัตโนมัติตามสภาพแวดล้อมนี้สามารถทำงานได้จริงโดยใช้ผลที่ได้จากเซนเซอร์แต่ละตัว แล้วทำการแปลงสัญญาณเอาท์พุทของเซนเซอร์ให้กลายเป็นพัลส์ความถี่ที่สามารถส่งไปประมวลผลการทำงาน โดยไมโครคอนโทรลเลอร์ในการแปลงสัญญาณเอาท์พุทจะใช้ไมโครคอนโทรลเลอร์ เบอร์ AT89C2051 ซึ่งเป็นไมโครคอนโทรลเลอร์ขนาดเล็กมีราคาถูกก่อนนำผลที่เปลี่ยนเป็นสัญญาณพัลส์ความถี่ไปยังไมโครคอนโทรลเลอร์เบอร์AT89C52 ซึ่งเป็นไมโครคอนโทรลเลอร์ที่มีความสามารถในการรับเอาข้อมูลจากหลายอินพุทมาประมวลผลก่อนทำการส่งออกอากาศผ่านโมดูล RF 2.4 GHz และในด้านการรับข้อมูลก็จะใช้หลักการเดียวกัน โดยผลที่ได้จะนำไปควบคุมอุปกรณ์ไฟฟ้าส่วนหนึ่ง แสดงผลผ่านหน้าจอแอลซีดีส่วนหนึ่ง การนำผลไปควบคุมอุปกรณ์ไฟฟ้านั้นสามารถบังคับผลที่จะนำไปควบคุมอุปกรณ์โดยใช้สวิตช์กดติดปล่อยดับในการควบคุมค่าให้เป็นไปตามต้องการโดยไม่ต้องทำการเปลี่ยนแปลงโปรแกรมที่ใช้ในการควบคุมผล ส่วนผลที่ได้บนหน้าจอแอลซีดีจะระบุเป็นตัวเลขพร้อมอักษรตามหน่วยมาตรฐาน

ความสามารถในการรับส่งผลของเซนเซอร์นั้นยังอยู่ในระดับที่ต่ำเมื่อเทียบกับอุปกรณ์มาตรฐานที่ใช้ในการวัดทั่วไป เนื่องจากต้องทำการส่งข้อมูลผ่านสัญญาณ RF แล้วทำการส่งผลเข้าไมโครคอนโทรลเลอร์ซึ่งในจุดนี้เอง ที่ต้องคำนึกถึงความเร็วในการเปลี่ยนแปลงข้อมูลเป็นอย่างมาก เนื่องจากไมโครคอนโทรลเลอร์ไม่สามารถที่จะทำงานด้วยความถี่ของข้อมูลที่สูงจึงทำการกำหนดค่าการเปลี่ยนแปลงของข้อมูลที่ได้จากเซนเซอร์แต่ละตัวให้อยู่ในช่วง 0-1KHz ทำให้ผลที่จะแสดงบนเอาท์พุทต่างๆ นั้นจะไม่เปลี่ยนแปลงในทันทีที่มีการเพิ่มหรือลดค่าเอาท์พุทของเซนเซอร์ ในทางกลับกันจะเป็นผลดีต่อการทำงานของรีเลย์คือจะยังคงสถานะการทำงานเดิมเมื่อมีการเปลี่ยนแปลงของข้อมูลเพิ่มขึ้นหรือว่าลดลงสลับกันอย่างรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

-สมยศ จุณณะปิยะ.Microcontroller Application.กรุงเทพฯ: คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2546

-วิวัฒน์ กิรานนท์.วิศวกรรมการสื่อสาร Communication Engineering.กรุงเทพฯ: คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2546

-สุชาติ กังวารจิตต์.เครื่องรับส่งวิทยุและระบบวิทยุสื่อสาร:ซีเอ็ดยูเคชั่น จำกัด, กรุงเทพฯ, 2521

-อนันท์ คัมภิรานนท์.ทฤษฎีอิเล็กทรอนิกส์อุตสาหกรรม1.สกายบุ๊กส์,กรุงเทพ,2545

-www.electhai.com

-www.technicyaso.ac.th/teacher/chingchai/

-www.thaiio.com

-www.adisak51.com

-www.micro4dev.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

โปรแกรมที่ใช้ในการเปรียบเทียบค่าของความถี่ที่ได้จากเซนเซอร์กับปริมาณน้ำฝน (Level)

```
 #include <REG2051.H>
 #include<stdio.h>
 #include<intrins.h>
char  num;
unsigned char  round_count_low;
unsigned char  round_count_high;
unsigned int   f;
unsigned int   freq;
sbit start=P1^7;

void delay1(unsigned int j)
{
      for(j=j;j>1;j--)
            {
                  _nop_();
            }
}
//---------------------------
 void demsec (unsigned int count)
      {
        unsigned char i;
        while(count)
            {
              for(i=1,i<152;i++);
              count--;
            }
        }
//---------------------------
code unsigned char patt_c[16]{'0', '1', '2' , '3', '4', '5', '6',
'7', '8', '9', 'A','B','C','D','E','F' };

void print_dat2(unsigned int dat)
{
 printf("#");                          demsec(50);
 printf("%c",patt_c[(dat/10)%10]);     demsec(20);
 printf("%c",patt_c[dat%10])           demsec(20);
}
//---------------------------
void cal_rpm(void)
{
  unsigned long temp1;
  unsigned long temp2;

      temp2= round_count_low;

      temp2= round_count_high;

      temp2 = temp2*0x100;

      temp2 = temp2 | temp2;
 f=temp2;
   }

//----------------------------
void main(void)
{    delay1(1000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    demsec(100);
  SCON=0X52;    // rs232
 TMOD=0X21;    // auto relood timmer 0
 // PCON=0X80;     // rs232*2
TH1=0xfd;      //
      TR1=1;
   TH0=0x00;     // 100usec  11.059*2
   TL0=0x00;
   TR0=1;
   ET0=0;
   EX0=1;
   IT0=1;
   EA=1;  //----int a1
    while(1)
      {
      freq=9216/f;
          freq=(freq*100)+2;
   if(freq < 50) {num=1;}
   if(freq > 50) {num=2;}
   if(freq > 476) {num=3;}
   if(freq > 65000){num=0;}
      while(start==1);
      print_dat2(num);  demsec(10); printf("d");
      while(start==0);
        }
}
//int t0-----------------------------------------------
void Ex0(void) interrupt 0
{
       round_count_low=TL0;
         round_count_high=TH0;
         TL0=0x00;
         TH0=0x00;
         cal_rpm();
 }
//--int1-----------------------------------------------
void timer0(void) interrupt 1
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ในการเปรียบค่าความถี่ที่ได้จากเซนเซอร์ของแสงสว่างกับความเข้มแสง (kluk)

```
 #include <REG2051.H>
 #include<stdio.h>
 #include<intrins.h>
unsigned int num;
unsigned char  round_count_low,round_count_high;
unsigned int f,freq;
sbit start=P1^7;
void demsec (unsigned int count)
      {
            unsigned char i;
            while(count)
            {
            for(i=1;i<152;i++);
            count--;
          }
   }
 code unsigned char
patt_c[16]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','
E','F' };
void print_dat2(unsigned int dat)
{SBUF='#'; demsec(20);
 SBUF = patt_c[(dat/1000)%10];   demsec(20);
 SBUF = patt_c[(dat/100)%10];   demsec(20);
 SBUF = patt_c[(dat/10)%10];   demsec(20);
}
void cal_rpm(void)
{
      unsigned long temp1;
      unsigned long temp2;
      temp1 = round_count_low;
      temp2 = round_count_high;
      temp2 = temp2 * 0x100;
      temp2 = temp2 | temp1;
       f=temp2;
}
void main(void)
{
            SCON=0x52;
            TMOD=0x21;
            TH1=0xfd;
            TR1=1;
      TH0=0x00;
      TL0=0x00;
      TR0=1;
      ET0=0;
      EX0=1;
      IT0=1;
    while(1)
{
    freq=921600/f;
  if(freq < 938) {num=4100 -((freq - 937)*0.01);}
  if(freq < 937) {num=4000 -((freq - 933)*0.008);}
  if(freq < 933) {num=3500 -((freq - 928)*0.1);}
  if(freq < 928) {num=3000 -((freq - 920)*0.016);}
  if(freq < 920) {num=2500 -((freq - 916)*0.016);}
  if(freq < 916) {num=2250 -((freq - 910)*0.024);}
  if(freq < 910) {num=2000 -((freq - 902)*0.032);}
  if(freq < 902) {num=1750 -((freq - 893)*0.036);}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 if(freq < 893) {num=1500 -((freq - 880)*0.052);}
 if(freq < 880) {num=1250 -((freq - 863)*0.068);}
 if(freq < 863) {num=1000 -((freq - 800)*0.1);}

 if(freq == 844) {num= 820;}
 if(freq == 843) {num= 830;}
 if(freq == 842) {num= 800;}
 if(freq == 841) {num= 790;}

 if(freq < 841) {num= 790 -((freq - 835)*0.25);}
 if(freq < 835) {num= 760 -((freq - 828)*0.1);}
 if(freq < 828) {num= 720 -((freq - 825)*0.15);}
 if(freq < 825) {num= 700 -((freq - 820)*0.25);}
 if(freq < 820) {num= 680 -((freq - 815)*0.25);}
 if(freq < 815) {num= 660 -((freq - 810)*0.15);}
 if(freq < 810) {num= 620 -((freq - 780)*0.25);}
 if(freq < 780) {num= 500 -((freq - 770)*0.5);}
 if(freq < 770) {num= 480 -((freq - 765)*0.25);}
 if(freq < 765) {num= 460 -((freq - 760)*0.25);}
 if(freq < 760) {num= 440 -((freq - 710)*0.5);}
 if(freq < 710) {num= 340 -((freq - 690)*1);}
 if(freq < 690) {num= 320 -((freq - 680)*0.5);}
 if(freq < 680) {num= 300 -((freq - 677)*0.3);}
 if(freq < 677) {num= 290 -((freq - 673)*0.4);}
 if(freq < 673) {num= 280 -((freq - 655)*0.5);}
 if(freq < 620) {num= 250 -((freq - 610)*1);}
 if(freq < 610) {num= 210 -((freq - 600)*1);}
 if(freq < 600) {num= 200 -((freq - 570)*1.5);}
 if(freq < 570) {num= 180 -((freq - 555)*1.5);}
 if(freq < 555) {num= 170 -((freq - 550)*0.5);}

 if(freq < 550) {num= 160 -((freq -490)*2);}
 if(freq < 490) {num= 130 -((freq -470)*2);}
 if(freq < 470) {num= 120 -((freq -410)*0.16);}
 if(freq < 410) {num= 110 -((freq -380)*0.33);}
 if(freq < 380) {num= 90 -((freq - 340)*0.25);}
 if(freq < 340) {num= 80 -((freq - 300)*0.125);}
 if(freq < 300) {num= 70 -((freq - 170)*0.222);}
 if(freq < 170) {num= 40 -((freq - 75)*0.2);}
 if(freq < 75)  {num= 10 -((freq - 30)*0.3);}
 if(num > 65000){num=0;}

  while(start==1);
  print_dat2(num);demsec(10); SBUF='y';
  while(start==0);
   }
}
// int t0-----------------------------------------------------
void Ex0(void) interrupt 0
{
        round_count_low=TL0;
        round_count_high=TH0;
        TL0=0x00;
        TH0=0x00;
        cal_rpm();
}
//  int1 -----------------------------------------------------
void timer0(void) interrupt 1
{
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ในการเปรียบเทียบค่าระหว่างความถี่จากเซนเซอร์ลมกับความเร็วลม (m/s)

```
 #include<REG2051.H>
 #include<stdio.h>
 #include<intrins.h>
  char num;
unsigned char  round_count_low,round_count_high;
unsigned int f,freq;
sbit start=P1^7;
void delay1(unsigned int j){ for(j=j;j>1;j--){ _nop_(); }}
//-----------------------------
void demsec (unsigned int count)
      {
            unsigned char i;
            while(count)
            {
            for(i=1;i<152;i++);
            count--;
          }
   }
code unsigned char
patt_c[16]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','
E','F' };
void print_dat2(unsigned int dat)
{printf("#");       demsec(50);
 printf("%c",patt_c[(dat/10)%10]);   demsec(20);
 printf("%c",patt_c[dat%10]);        demsec(20);
}
void cal_rpm(void)
{
unsigned long temp1;
unsigned long temp2;

      temp1 = round_count_low;
      temp2 = round_count_high;
      temp2 = temp2 * 0x100;
      temp2 = temp2 | temp1;
    f=temp2;
}
void main(void)
{   delay1(1000);
    demsec(100);

   // CKCON =  CKCON | 1 ;          //clook *1
      SCON=0x52;     // rs232
    TMOD=0x21;     // auto relood timmer 0
   // PCON=0x80;     // rs232 * 2
      TH1=0xfd;      //11.059
      TR1=1;
//------int  timer0 set ----------------
   TH0=0x00;     // 100 usec  11.059 * 2
   TL0=0x00;
   TR0=1;
   ET0=0;
//------ int EX0 --P3.2-------------------
   EX0=1;
   IT0=1;

   EA=1;  //----int all
//-------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   while(1)
      {
          freq=9216/f;
          freq=(freq*100)+2;

   if(freq < 24){num=20;}
   if(freq > 36){num=25;}
   if(freq > 48){num=30;}
   if(freq > 59){num=35;}
   if(freq > 69){num=40;}
   if(freq > 80){num=45;}
   if(freq > 93){num=50;}
   if(freq > 103){num=55;}
   if(freq > 119){num=60;}
   if(freq > 128){num=65;}
   if(freq > 141){num=70;}
   if(freq > 158){num=75;}
   if(freq > 172){num=80;}

   if(freq > 190){num=85;}
   if(freq > 207){num=90;}
   if(freq > 218){num=95;}
   if(freq > 234){num=99;}

   if(freq > 65000){num=0;}
    while(start==1);
   print_dat2(num);demsec(10);     printf("e");
   while(start==0);
    }
}
// int t0-----------------------------------------------------
void Ex0(void) interrupt 0
{
         round_count_low=TL0;
         round_count_high=TH0;
         TL0=0x00;
         TH0=0x00;
         cal_rpm();
}
//--int1 -----------------------------------------------------
void timer0(void) interrupt 1
{
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ในการเปรียบเทียบค่าความถี่ที่ได้จากเซนเซอร์ความดันกับความดันอากาศ (ps)

```
 #include <REG2051.H>
 #include<stdio.h>
 #include<intrins.h>
unsigned int num;
unsigned char  round_count_low,round_count_high;
unsigned int f,freq;
sbit start=P1^7;
void demsec (unsigned int count)
      {
            unsigned char i;
            while(count)
            {
            for(i=1;i<152;i++);
            count--;
          }
   }

code unsigned char
patt_c[16]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','
E','F' };
void print_dat2(unsigned int dat)
{SBUF='#'; demsec(20);
 //SBUF = patt_c[(dat/1000)%10];   demsec(20);
 SBUF = patt_c[(dat/100)%10];   demsec(20);
 SBUF = patt_c[(dat/10)%10];   demsec(20);
 SBUF = patt_c[dat%10];        demsec(20);
}
void cal_rpm(void)
{ unsigned long temp1;
  unsigned long temp2;
      temp1 = round_count_low;
      temp2 = round_count_high;
      temp2 = temp2 * 0x100;
      temp2 = temp2 | temp1;
    f=temp2;
}
void main(void)
{
      SCON=0x52;    // rs232
   TMOD=0x21;    // auto relood timmer 0
      TH1=0xfd;     //11.059
      TR1=1;
//------int  timer0 set ----------------
   TH0=0x00;    // 100 usec  11.059 * 2
   TL0=0x00;
   TR0=1;
   ET0=0;
//------ int EX0 --P3.2-------------------
   EX0=1;
   IT0=1;
   EA=1;  //----int all
    while(1)
{
          freq=921600/f;
   if(freq > 970) {num=160;}
   if(freq < 970) {num=160;}
   if(freq < 920) {num=155;}
   if(freq < 895) {num=150;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if(freq < 860) {num=145;}
    if(freq < 825) {num=140;}
    if(freq < 795) {num=135;}
    if(freq < 760) {num=130;}
    if(freq < 725) {num=125;}
    if(freq < 695) {num=120;}
    if(freq < 660) {num=115;}
    if(freq < 625) {num=110;}

    if(freq < 595) {num=105;}
    if(freq < 560) {num=100;}
    if(freq < 525) {num=95;}
    if(freq < 490) {num=90;}
    if(freq < 455) {num=85;}
    if(freq < 425) {num=80;}
    if(freq < 395) {num=75;}

    if(freq < 360) {num=70;}
    if(freq < 330) {num=65;}
    if(freq < 300) {num=60;}
    if(freq < 270) {num=55;}
    if(freq < 240) {num=50;}
    if(freq < 225) {num=45;}
    if(freq < 200) {num=40;}
    if(freq < 181) {num=35;}
    if(freq < 152) {num=30;}
    if(freq < 130) {num=25;}
    if(freq < 105) {num=20;}
    if(freq < 80) {num=15;}
    if(freq < 55) {num=10;}

    if(num > 65000){num=0;}
   while(start==1);
    print_dat2(num);SBUF='a';
    while(start==0);
       }
}
// int t0-----------------------------------------------------
void Ex0(void) interrupt 0
{
         round_count_low=TL0;
         round_count_high=TH0;
         TL0=0x00;
         TH0=0x00;
         cal_rpm();
}
//--int1 -----------------------------------------------------
void timer0(void) interrupt 1
{
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการทำงานของภาคส่งและการมัลติเพล็กซ์

```
#include <REGX52.H>
#include <intrins.h>
#include <math.h>
#include <stdio.h>

typedef union
{ unsigned int i;
  float f;
} value;
enum {TEMP,HUMI};
#define noACK 0
#define ACK 1
#define STATUS_REG_W 0x06
#define STATUS_REG_R 0x07
#define MEASURE_TEMP 0x03
#define MEASURE_HUMI 0x05
#define RESET 0x1e
#define control P2_2
sbit DATA = P2^7;
sbit SCK =  P2^6;
sbit cpu1=P1^0;

sbit cpu2=P1^1;

sbit cpu3=P1^2;

sbit cpu4=P1^3;

//---------------------------232 rs--
 unsigned char d_232,num0,num1;
 unsigned char  num2,num3;
 unsigned char b_232[5];
 unsigned char dx_232;
 bit a,b,c,d,e,y;

 void dmsec (unsigned int count)
      {
            unsigned char i;
            while(count)
            {
            for(i=1;i<152;i++);
            count--;
          }
   }

//-------------------------------------------------------------------------
char s_write_byte(unsigned char value)
//-------------------------------------------------------------------------
{
  unsigned char i,error=0;
  for (i=0x80;i>0;i/=2)
  { if (i & value) DATA=1;
    else DATA=0;
    SCK=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    _nop_();_nop_();_nop_();
    SCK=0;
  }
  DATA=1;
  SCK=1;
  error=DATA;
  SCK=0;
  return error;
}

//-----------------------------------------------------------------------
char s_read_byte(unsigned char ack)
//-----------------------------------------------------------------------

{
  unsigned char i,val=0;
  DATA=1;
  for (i=0x80;i>0;i/=2)
  { SCK=1;
    if (DATA) val=(val | i);
    SCK=0;
  }
  DATA=!ack;
  SCK=1;
  _nop_();_nop_();_nop_();
  SCK=0;
  DATA=1;
  return val;
}

//-----------------------------------------------------------------------
void s_transstart(void)
//-----------------------------------------------------------------------

// generates a transmission start

//   _____         ________

// DATA:      |_______|

//     ___     ___

// SCK : ___|   |___|   |_____
{
   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;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//-----------------------------------------------------------------------
void s_connectionreset(void)
//-----------------------------------------------------------------------
// communication reset: DATA-line=1 and at least 9 SCK cycles followed
by transstart
//   _____________________________________________________________
________
// DATA:
|________|
//     _    _    _    _    _    _    _    _    _        ___     ___
// SCK : __| |__| |__| |__| |__| |__| |__| |__| |______|   |___|
|_______
{
  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
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//-----------------------------------------------------------------------
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;                  // set RS232 parameter
    TMOD = 0x20;

    TH1= 0xfd;// PCON |= 0x80;      // 19200

    TR1 =1;

      ET0=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    EA=1;
      ES=1;
}
float temp_dat,humi_dat;
//---------------------------------------------------------------------------

void calc_sth11(float *p_humidity ,float *p_temperature)

//---------------------------------------------------------------------------

//calculates temperature [¶C] and humidity [%RH]

//input :  humi [Ticks] (12 bit)

//    temp [Ticks] (14 bit)

//output:  humi [%RH]

//    temp [¶C]

{ const float C1=-4.0;                  // for 12 Bit

  const float C2=+0.04050;               // 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 [¶C]

  t_C=t*0.01-40;                        //calc. temperature from ticks to
[¶C]
  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 [¶C]
 *p_humidity=rh_true;
      //return humidity[%RH]

  temp_dat=t_c*10;
  humi_dat=rh_true*10;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//---------------------------------------------------------------
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_check()
//---------------------------------------------------------------------

// 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 [℃]

// 4. calculate dew point [℃]

// 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();

   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("%5.1f\t%5.1f\n",temp_val.f,humi_val.f);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
  //---------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!)
 //--------------------------------------------------------------------------

 //-Form Temp&Humidity Sensor
}
code unsigned char
patt_c[16]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E
','F' };
void print_dat2(unsigned int dat)
{printf("#");        dmsec(10);
 printf("%c",patt_c[(dat/10)%10]);   dmsec(20);
 printf("%c",patt_c[dat%10]);         dmsec(20);
}
void print_dat3(unsigned int dat)
{printf("#");        dmsec(10);
 printf("%c",patt_c[(dat/100)%10]);   dmsec(20);
 printf("%c",patt_c[(dat/10)%10]);   dmsec(20);
 printf("%c",patt_c[dat%10]);         dmsec(20);
}
void main()
{
      init_uart();
  printf("\n HOOOO ");
 //while ((c = getchar ()) == 'T')
  P1=0xff;
 //RAN=1;
  while (1)
  {
       main_check();
       printf("\n#");
      print_dat2(temp_dat/10);printf("b  ");dmsec(500);
      print_dat2(humi_dat/10);printf("c  ");          dmsec(500);

       cpu1=0;while(a==0); if(a==1){ num0 =(b_232[1]*10)+b_232[2]; a=0;}
cpu1=1;
       cpu2=0;while(b==0); if(b==1){ num1 =(b_232[1]*10)+b_232[2]; b=0;}
cpu2=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        cpu3=0;while(c==0);if(c==1){   num2
=(b_232[1]*100)+(b_232[2]*10)+b_232[3]; c=0;}  cpu3=1;
        cpu4=0;while(d==0); if(d==1){ num3
=(b_232[1]*100)+(b_232[2]*10)+b_232[3]; d=0;}  cpu4=1;
        ES=0;
        cpu1=0;dmsec(500);cpu1=1;
        cpu2=0;dmsec(500);cpu2=1;
        cpu3=0;dmsec(500);cpu3=1;
        cpu4=0;dmsec(500);cpu4=1;
        ES=0;
   }
}
//-------------------------------
Voidserial (void) interrupt 4
{
       if(RI==1)
       {
                RI=0;
              switch (SBUF)
              {
                 case'#':dx_232=0x00; break;
                   case'D':a=1;  break;
                   case'E':b=1;  break;
                   case'Y':c=1;  break;
                   case'A':d=1;  break;
              }
                 if(SBUF==0){} else{b_232[dx_232]=SBUF-48;SBUF=0;
dx_232++;}

          if(dx_232 > 6){dx_232=6;}
    }
  else{}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมภาครับ, การแสดงผลผ่านหน้าจอและส่วนควบคุมอุปกรณ์ไฟฟ้าอัตโนมัติ

```
#include <REGX52.H>
#include <stdio.h>
#include<intrins.h>
 unsigned char d_232;
 unsigned char b_232[5];
 unsigned char dx_232;
 bit a,b,c,d,e,y;
 sbit IPSDA       = P2^6;          //pin 6
 sbit IPSCL       = P2^7;          //pin 5
 unsigned int set;

sbit out1=P2^0;
sbit out2=P2^1;
sbit out3=P2^2;
sbit out4=P2^3;
sbit out5=P2^4;
sbit out6=P2^5;

sbit sw2=P1^0;
sbit sw3=P1^1;
sbit sw4=P1^2;
sbit sw5=P1^3;
sbit sw6=P1^4;
sbit sw7=P1^5;
sbit sw8=P1^6;
sbit sw9=P1^7;
unsigned int num5,num0,num1,num2,num3,num4;
void delay(unsigned int j){ for(j=j;j>1;j--){ _nop_(); }}
void delay1(unsigned int j){ for(j=j;j>1;j--){ _nop_(); }}
void dmsec (unsigned int count)
      {
    unsigned int i;
                  while (count)
      {
            i =115; while (i>0)i--;
                  count--;
                        }
      }
//-------------------------------------------------------
void ipdel (void)
{
    _nop_ ();
    _nop_ ();
    _nop_ ();
    _nop_ ();
    _nop_ ();
    _nop_ ();
    _nop_ ();
    _nop_ ();
    _nop_ ();
    _nop_ ();
}
void ipchigh (void)
 {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    IPSCL = 1;
    ipdel ();
}
void ipclow (void)
{
    IPSCL = 0;
    ipdel ();
}
void ipstart (void)
{
    IPSDA = 1;
    IPSCL = 1;
      _nop_();
      _nop_();
      _nop_();
      _nop_();
    IPSDA = 0;
    ipdel ();
    IPSCL = 0;
    IPSDA = 1;
}
void ipstop (void)
 {
    IPSDA = 0;
    IPSCL = 1;
    ipdel ();
    IPSDA = 1;
}

bit ipwrbyte (unsigned char dat)
{
      unsigned char i;
    bit outbit;
    for (i=1;i<=8;i++)
      {
        outbit = dat & 0X80;
        IPSDA = outbit;
        dat = dat << 1;
        ipchigh ();
        ipclow ();
    }
    IPSDA = 1;
    ipchigh ();
    outbit = IPSDA;
    ipclow ();
    return (outbit);
}

unsigned char iprdbyte (void)
{
    unsigned char i,dat;
    bit inbit;
    dat = 0;
    for (i=1;i<=8;i++)
      {
        ipchigh ();
        inbit = IPSDA;
        dat = dat << 1;
        dat = dat | inbit;
        ipclow ();
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    IPSDA = 1;
    ipchigh ();
    inbit = IPSDA;
    ipclow ();
    if (~inbit) dat = 0xff;
    return (dat);
}

bit eeprom_wd(unsigned char address,unsigned char dat)
{
    bit err;
    ipstart ();
    err = ipwrbyte(0xa0);

    if (err==0)
      {
        err = ipwrbyte (address);
            if(err==0)
            {
                  err = ipwrbyte (dat);
            }
    }
    ipstop ();
    return (err);
}

unsigned char eeprom_rd(unsigned char address)
{
    unsigned char dat;
    bit err=0;

    ipstart ();
    err = ipwrbyte(0xa0); //(addr);

    if (err==0)
      {
            err = ipwrbyte(address);
            if(err==0)
            {
                  ipstop();
                  delay1(10);
                  ipstart();

                  err = ipwrbyte(0xa1); //(addr);
                  dat = iprdbyte ();
            }
    }
       else dat = 0xff;
    ipstop ();
    return (dat);

}
unsigned long read_xx(unsigned char add)
{
      unsigned long temp=0X00;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      temp = temp + (eeprom_rd(add) *0X01000000);      temp = temp +
(eeprom_rd(add+1)*0X10000);
      temp = temp + (eeprom_rd(add+2)*0X0100);   temp = temp +
(eeprom_rd(add+3));
      return(temp);
}
void write_xx(unsigned char add,unsigned long dat)
{
      eeprom_wd(add+0,((dat / 0X10000000)& 0Xff));     delay(3000);
      eeprom_wd(add+1,((dat / 0x010000)& 0xff) );      delay(3000);
      eeprom_wd(add+2,((dat / 0X0100)& 0xff));  delay(3000);
      eeprom_wd(add+3,(dat & 0xff));            delay(3000);
}
//*****************************************************************
#define ON 1
#define OFF 0
sbit RS     = P0^0;
sbit ENABLE = P0^1;
sbit D4           = P0^2;
sbit D5           = P0^3;
sbit D6           = P0^4;
sbit D7           = P0^5;
//sbit BEEP   = P3^2;
bdata unsigned char buf;
sbit buf_0=buf^0;
sbit buf_1=buf^1;
sbit buf_2=buf^2;
sbit buf_3=buf^3;
sbit buf_4=buf^4;
sbit buf_5=buf^5;
sbit buf_6=buf^6;
sbit buf_7=buf^7;
//*****************************************************************
void hi(void) { D4=buf_4; D5=buf_5; D6=buf_6; D7=buf_7; }
void low(void){ D4=buf_0; D5=buf_1; D6=buf_2; D7=buf_3; }
void write_command(unsigned char dat)
{
      buf=dat;
      hi();
      RS=0;
      delay(5);
      ENABLE=1;
      delay(10);
      ENABLE=0;
      delay(10);
      low();
      ENABLE=1;
      delay(10);
      ENABLE=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        delay(10);
}

void write_data(unsigned char dat)
{
        buf=dat;
        hi();
        RS=1;
        delay(5);
        ENABLE=1;
        delay(10);
        ENABLE=0;
        delay(10);
        low();
        ENABLE=1;
        delay(10);
        ENABLE=0;
        delay(10);
}
void write_line1(void) { write_command(0X80); }
void write_line2(void) { write_command(0XC0); }
void print_lcd2(unsigned char dat[])
{       unsigned char count=0x00;
        char *string_ptr=dat;
        write_line2();
        while(*string_ptr != 0x00)//=NULL
        {
                write_data(*string_ptr);
                count++;
                string_ptr++;
        }
}
void print_lcd1(unsigned char dat[])
{       unsigned char count=0X00;
        char *string_ptr=dat;
        write_line1();
        while(*string_ptr != 0X00)//=NULL
        {
                write_data(*string_ptr);
                count++;
                string_ptr++;
        }
}
void initial_lcd(void)
{
        RS=0; D7=0; D6=0; D5=1; D4=1; delay(10); ENABLE=1; delay(100);
ENABLE=0; delay(50000);//33
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      RS=0; D7=0; D6=0; D5=1; D4=1; delay(10); ENABLE=1; delay(100);
ENABLE=0; delay(1000);
      RS=0; D7=0; D6=0; D5=1; D4=1; delay(10); ENABLE=1; delay(100);
ENABLE=0; delay(1000);//32
      RS=0; D7=0; D6=0; D5=1; D4=0; delay(10); ENABLE=1; delay(100);
ENABLE=0; delay(1000);
      RS=0; D7=0; D6=0; D5=1; D4=0; delay(10); ENABLE=1; delay(100);
ENABLE=0; delay(1000);//28
      RS=0; D7=1; D6=0; D5=0; D4=0; delay(10); ENABLE=1; delay(100);
ENABLE=0; delay(1000);
      RS=0; D7=0; D6=0; D5=0; D4=0; delay(10); ENABLE=1; delay(100);
ENABLE=0; delay(1000);//0C
      RS=0; D7=1; D6=1; D5=0; D4=0; delay(10); ENABLE=1; delay(100);
ENABLE=0; delay(1000);
      RS=0; D7=0; D6=0; D5=0; D4=0; delay(10); ENABLE=1; delay(100);
ENABLE=0; delay(1000);//06
      RS=0; D7=0; D6=1; D5=1; D4=0; delay(10); ENABLE=1; delay(100);
ENABLE=0; delay(1000);
      RS=0; D7=0; D6=0; D5=0; D4=0; delay(10); ENABLE=1; delay(100);
ENABLE=0; delay(1000);//01
      RS=0; D7=0; D6=0; D5=0; D4=1; delay(10); ENABLE=1; delay(100);
ENABLE=0; delay(1000);
      RS=1;
}
//---------------------------------------
code unsigned char
patt_c[16]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','
E','F' };
//---------------------------------------
void print_data (unsigned char add,dat)
{
write_command(add);
write_data(patt_c[(dat/10)%10]);
add++;
write_command(add);
write_data(patt_c[dat%10]);
}
void print_data1(unsigned char add,dat)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
write_command(add);
write_data(patt_c[(dat/10)%10]);
add++;write_command(add); write_data('.');add++;
write_command(add);
write_data(patt_c[dat%10]);
}
void print_data2(unsigned char add,dat)
{
write_command(add);
write_data(patt_c[dat%10]);
}
void print_data3(unsigned char add,dat)
{
write_command(add);
write_data(patt_c[(dat/100)%10]);
add++;
write_command(add);
write_data(patt_c[(dat/10)%10]);
add++;write_command(add); write_data('.');
add++;
write_command(add);
write_data(patt_c[dat%10]);
}
void print_data4(unsigned char add,dat)
{
write_command(add);
write_data(patt_c[(dat/100)%10]);
add++;write_command(add); write_data('.');
add++;write_command(add);
write_data(patt_c[(dat/10)%10]);
add++;
write_command(add);
write_data(patt_c[dat%10]);
}
//---------------------------------------
void print_dat(unsigned int dat)
{
printf("%c",patt_c[(dat/1000)%10]);
printf("%c",patt_c[(dat/100)%10]);
printf("%c",patt_c[(dat/10)%10]);
printf("%c",patt_c[dat%10]);

}
void main()
{
        delay(65000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
     ET0=1;

     TR0=1;
     EA=1;
      //--------------
       SCON=0X52;                    // set RS232 parameter
    TMOD = 0X20;
    TH1 = 0xfd;// PCON |= 0x80;      // 19200

    TR1 = 1;
//=============================================================================

 //---------------65535---------
    write_xx(00,33);

      num1=read_xx(00);

      delay(3000);
  //---------------255---------
      eeprom_wd(04,34);delay(3000);
      initial_lcd();

    delay(65000);

 EA=1;
//=============================================================================
 ES=0;
 Num3=1;
 while (1)
  {
   write_command(128);

   write_data('T'); print_data(129,num1);write_command(134);

   write_data('H'); print_data(135,num2); write_data('%');


if(num3==1){write_command(141);write_data('1');write_data('o');write_d
ata('w');}

if(num3==2){write_command(141);write_data('M');write_data('i');write_d
ata('d');}

if(num3==3){write_command(141);write_data('H');write_data('i');write_d
ata(' ');}

   write_command(192);

   write_data('W'); print_data1(193,num4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    print_data4(198,num5);
      print_data3(204,num0);
 /*
   write_command(197);
   write_data('L'); print_data4(num5); write_data('%');
   write_command(204);
   write_data('P'); print_data3(206,num0); write_data(' ');
  */
if(a==1){    ES=0; num0=(b_232[1]*100)+(b_232[2]*10)+b_232[3]; a=0;}  //
PSI
if(b==1){    ES=0; num1 =(b_232[1]*10)+b_232[2]; b=0;}
if(c==1){    ES=0; num2=(b_232[1]*10)+b_232[2]; c=0;}
if(d==1){    ES=0; num3 =(b_232[1]*10)+b_232[2]; d=0;}
if(e==1){    ES=0; num4 =(b_232[1]*10)+b_232[2]; e=0;}
if(y==1){    ES=0; num5 =(b_232[1]*100)+(b_232[2]*10)+b_232[3]; y=0;}  //
sang
      ES=1;
      printf("\n ");print_dat(num3);
      printf(" b0");print_dat(b_232[0]);printf(" b1
");print_dat(b_232[1]);
      printf(" b2");print_dat(b_2322[2]);printf(" b3
");print_dat(b_232[3]);

 if(sw2==0){ES=0; set=read_xx(12);}
 while(sw2==0)
     {
        if(sw9==0){set++;}
        if(sw8==0){set--;}
          write_command(128);write_data('s'); print_data(129,set);
          write_xx(12,set);
          dmsec(50);
       }
//-----------------------------------
if(sw4==0){ES=0; set=read_xx(16);}
 while(sw4==0)
     {
        if(sw9==0){set++;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(sw8==0){set--;}
        write_command(134);write_data('s'); print_data(135,set);
          write_xx(16,set);
          dmsec(50);
       }
//-------------------------------------
if(sw6==0){ES=0; set=read_xx(20);}
 while(sw6==0)
     {
        if(sw9==0){set++;}
        if(sw8==0){set--;}
        if(set > 3){set=1;}
        write_command(140);write_data('s');
          write_xx(20,set);
          dmsec(50);

if(set==1){write_command(141);write_data('1');write_data('o');write_da
ta('w');}

if(set==2){write_command(141);write_data('M');write_data('i');write_da
ta('d');}

if(set==3){write_command(141);write_data('H');write_data('i');write_da
ta(' ');}

       }write_command(140);write_data(' ');

//-----------------------------------
if(sw3==0){ES=0; set=read_xx(23);}
 while(sw3==0)
     {
        if(sw9==0){set++;}
        if(sw8==0){set--;}
        write_command(192);write_data('s'); print_data1(193,set);
          write_xx(23,set);
          dmsec(50);
       }
//-------------------------------------
if(sw5==0){ES=0; set=read_xx(27);}
 while(sw5==0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
     {
        if(sw9==0){set++;}
        if(sw8==0){set--;}
       write_command(198);
   //write_data('s'); print_data(199,set);

    print_data4(198,set);

         write_xx(27,set);
         dmsec(50);
       }
//-----------------------------------
if(sw7==0){ES=0; set=read_xx(31);}
 while(sw7==0)
     {
        if(sw9==0){set++;}
        if(sw8==0){set--;}
       write_command(205);
           print_data3(204,set);
         if(set>100){set=99;}
         write_xx(31,set);
         dmsec(50);
       }
//----------sys -------------------
 set=read_xx(12);dmsec(2);
 if(set > num1){out1=0;}else{out1=1;}
 set=read_xx(16);dmsec(2);
 if(set > num2){out2=0;}else{out2=1;}

 set=read_xx(20);dmsec(2);
 if(set > num3){out3=0;}if(set <=num3){out3=1;}
 set=read_xx(23);dmsec(2);
 if(set > num4){out4=0;}else{out4=1;}
 set=read_xx(27);dmsec(2);
 if(set > num5){out5=0;}else{out5=1;}
 set=read_xx(31);dmsec(2);
 if(set > num0){out6=0;}else{out6=1;}
 }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void timer0(void) interrupt 1
{
}
//-----------------------------------------
Voidserial (void) interrupt 4
{
      if(RI==1)
      {
               RI=0;
            switch (SBUF)
            {
                case'#':dx_232=0X00; break;
                  case'a':a=1;  break;
                  case'b':b=1;  break;
                  case'c':c=1;  break;
                  case'd':d=1;  break;
                  case'e':e=1;  break;
         case'y':y=1;  break;
            }
                if(SBUF==0){} else{b_232[dx_232]=SBUF-48;SBUF=0;
dx_232++;}
            if(dx_232 > 4){dx_232=5;}
   }
  else{}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้