สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องวัดวิเคราะห์สี

COLOR ANALYZER

โดย

นางสาว ธนัญธร ตรีรัตนกุล

นาย สรกิจ สรคุณพิพิธกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....

เลขทะเบียน 50183

วัน,เดือน,ปี 27 เม.ย. 2547

.b.....

.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชา ระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องวัดวิเคราะห์สี

ผู้จัดทำ 1. นางสาว ธนัญธร ตรีรัตนกุล

2. นาย สรกิจ สรคุณพิพิธกุล

..... อาจารย์ที่ปรึกษา

(รศ.ดร. วันชัย ริ้วรุจา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องวัดวิเคราะห์สี

นางสาว ธนัญธร ตรีรัตนกุล
นาย สรกิจ สรคุณพิพิธกุล
รศ.ดร. วันชัย ริ้วรุจา , อาจารย์ที่ปรึกษา
ปีการศึกษา 2545

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้เรียบเรียงขึ้นจากการพัฒนาอุปกรณ์ที่ใช้ในการวัดเปรียบเทียบสีของวัตถุ ซึ่งสามารถนำไปประยุกต์ใช้ประโยชน์ได้ในอุตสาหกรรมการผลิตต่างๆ และมีต้นทุนในการสร้างต่ำ

เครื่องวัดวิเคราะห์สีนี้ประกอบด้วย 3 ส่วน คือ

1. ส่วนอินพุท(Input Unit) – LDR(ตัวรับแสง)
2. ส่วนประมวลผล – PIC ไมโครคอนโทรลเลอร์(PIC Microcontroller)
3. ส่วนเอาท์พุท(Output Unit) – LCD

การวัดของเครื่องใช้คุณสมบัติการเปลี่ยนค่าความต้านทานของ LDR เมื่อได้รับแสงที่ส่องผ่านแผ่นกรองแสง ไมโครคอนโทรลเลอร์จะนำสัญญาณอินพุท(Input Signal)ที่ได้ไปประมวลผล และ แสดงค่าคลาดเคลื่อนที่คำนวณได้ออกทางจอ LCD ในรูปของเปอร์เซ็นต์เมื่อเทียบกับสีอ้างอิงที่กำหนดไว้ จากค่าคลาดเคลื่อนนี้เองจะทำให้รู้ว่าวัตถุนั้นมีสีผิดไปจากสีอ้างอิงมากน้อยอย่างไรและควรจะทำการแก้ไขหรือคัดเลือกอย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COLOR ANALYZER

Miss Thananthorn Threerattanakul
Mr. Sorakit Sorakunphiphitakul
Associate Professor Vanchai Riewruja , Advisor
2002

Abstract

This thesis has an objective to develop an instrument which can measure object's colors and display error values on the LCD for many utility applications in the manufacturing and its capital cost is low price.

The components of this COLOR ANALYZER are

1. Input Unit – LDR(Light Dependent Resistor)
2. CPU – PIC Microcontroller
3. Output Unit – LCD(Liquid Crystal Display)

The measurement used the reflection's quality. It can inspect the beam that reflects from the object's skin through the filters. The input signals will be sent to the microcontroller and then the microcontroller will calculate error values to display on the LCD by percentage. Finally, from the output values we can know the object's colors are over the limited values or not and what we should decide to do then.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญตาราง	IV
สารบัญรูป	V
บทที่ 1 บทนำ	1
1.1 ขอบเขตของการศึกษา	1
1.2 วัตถุประสงค์ของการสร้าง	1
1.3 ความเป็นมาของโครงงาน	2
บทที่ 2 ทฤษฎี	3
2.1 แสง (light)	3
2.1.1 แสงสีขาว	4
2.1.2 การสะท้อนแสง	6
2.2 สี (color)	7
2.2.1 แสงสี	8
2.2.2 สารสี	9
2.2.3 ช่วงความยาวคลื่นของสี	11
บทที่3 การออกแบบและการสร้าง	12
3.1 ตัวต้านทานไวแสง (Light dependent resistor)	12
3.2 ไมโครคอนโทรลเลอร์ตระกูล PIC	13
3.2.1 ระบบของ I/O ของ PIC 16F87x	15
3.2.1.1 พอร์ต A ใน PIC 16F87x	15
3.2.1.2 พอร์ต B ใน PIC 16F87x	17
3.2.1.3 พอร์ต C ใน PIC 16F87x	19
3.2.2 โครงสร้างหน่วยความจำ(memory)ของ PIC 16F87x	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การเปลี่ยนอนาลอกเป็นดิจิตอล (Analog-To-Digital Converter) 25
3.3 ตัวแสดงผลแบบผลึกเหลว Liquid Crystal Display (LCD) 30
3.3.1 ขาใช้งานโมดูล LCD 31
3.3.2 ความรู้เบื้องต้นเกี่ยวกับโมดูลแสดงผล LCD แบบอักขระ 32
3.3.3 โครงสร้างภายในของตัวควบคุมโมดูลแบบอักขระ 32
3.3.4 การติดต่อกับโมดูล LCD แบบอักขระ 33
3.3.5 ขั้นตอนการป้อนข้อมูล เพื่อการใช้งานโมดูล LCD 35
3.4 ภาษาที่ใช้สำหรับเขียนโปรแกรม PIC 37
3.4.1 ชนิดของโปรแกรม 38
บทที่4 การทดลองและผลการทดลอง 39
4.1 ส่วนที่1 (LDR) 39
4.2 ส่วนที่ 2 (ประมวลผล) 41
4.3 ส่วนที่3 (ส่วนแสดงผล) 44
4.4 ผลการทดลอง 44
บทที่ 5 บทสรุปและวิจารณ์ 50
5.1 ปัญหาและแนวทางแก้ไข 50
5.2 แนวทางการพัฒนา 50
5.3 บทสรุปและวิจารณ์ 51
ภาคผนวก
- วงจรเครื่องวัดวิเคราะห์สีด้วยไมโครคอนโทรลเลอร์ ตระกูล PIC MICROCONTROLLER 16F876
- โฟลชาร์ตโปรแกรม
- โปรแกรมภาษา C
- Data Sheet PIC MICROCONTROLLER 16F876
- Data Sheet MEMORY ORGANIZATION
- Data Sheet ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

กิตติกรรมประกาศ
บรรณานุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ 2.1 การดูดกลืนและสะท้อนแสงสีของสารสี 10

ตารางที่ 3.1 หน่วยความจำควบคุมเกี่ยวกับ A/ D 25

ตารางที่ 3.2 แสดงบิต7:ADFM 26

ตารางที่ 3.3 แสดงการแปลงสัญญาณให้ด้านซ้ายของ ADRESH 26

ตารางที่ 3.4 แสดงการแปลงสัญญาณให้ด้านขวาของ ADRESH 26

ตารางที่ 3.5 แสดงส่วนการใช้ ช่องสัญญาณ กับ พอร์ต 28

ตารางที่ 4.1 การทดลองหาค่าความต้านทาน เมื่อวัดความสว่างโดย Lux Meter ที่ความสว่างต่างกันและทำการหาค่า $\bar{A}$ ของ LDR ตัวที่ 1 39

ตารางที่4.2 การทดลองหาค่าความต้านทาน เมื่อวัดความสว่างโดย Lux Meter ที่ความสว่างต่างกันและทำการหาค่า $\bar{A}$ ของ LDR ตัวที่ 2 40

ตารางที่ 4.3 การทดลองหาค่าความต้านทาน เมื่อวัดความสว่างโดย Lux Meter ที่ความสว่างต่างกันและทำการหาค่า $\bar{A}$ ของ LDR ตัวที่ 3 40

ตารางที่ 4.4 การทดลองหาค่าความต้านทาน เมื่อวัดความสว่างโดย Lux Meter ที่ความสว่างต่างกันและทำการหาค่า $\bar{A}$ ของ LDR ตัวที่ 4 41

ตารางที่ 4.5 แสดงผลการวัดสีของวัตถุ ณ ตำแหน่งเดิมและความสว่างเท่ากัน 45

ตารางที่ 4.6 แสดงผลการวัดการผสมสีของสีขาวเพื่อให้ได้สีดำ 45

ตารางที่ 4.7 แสดงการเปรียบเทียบความคลาดเคลื่อนของสีอ้างอิงกับสีเปรียบเทียบ 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 ขนาดความยาวคลื่นของแสง 3

รูปที่ 2.2 รายละเอียดของคลื่นแม่เหล็กไฟฟ้าที่ความถี่หรือความยาวคลื่นต่างกัน 4

รูปที่ 2.3 แสงสีที่เกิดจากการหักเหของแสงสีขาวผ่านปริซึม 5

รูปที่ 2.4 ความสามารถของสายตามนุษย์ในการรับรู้การเปลี่ยนแปลงความเข้มของแสงสี 5

รูปที่ 2.5 ภาพรังสีตกกระทบและรังสีสะท้อน 6

รูปที่ 2.6 การสะท้อนของแสงบนวัตถุผิวเรียบและผิวไม่เรียบ 6

รูปที่ 2.7 การผสมแสงสี 8

รูปที่ 2.8 การผสมสารสี 9

รูปที่ 3.1 โครงสร้างและสัญลักษณ์ทางไฟฟ้า LDR 13

รูปที่ 3.2 ไมโครคอนโทรลเลอร์ตระกูล PIC ชนิด 8 ขา 13

รูปที่ 3.3 บล๊อกไดอะแกรมของขา RA3:RA0 และ RA5 15

รูปที่ 3.4 บล๊อกไดอะแกรมของขา RA/TOCKI 16

รูปที่ 3.5 บล๊อกไดอะแกรมของขา RB3:RB0 17

รูปที่ 3.6 บล๊อกไดอะแกรมของขา RB7:RB4 18

รูปที่ 3.7 บล๊อกไดอะแกรมของขา RC<0:2>และRC<5:7> 19

รูปที่ 3.8 บล๊อกไดอะแกรมของขา RC<3:4> 20

รูปที่ 3.9 ส่วนหน่วยความจำของPIC 16F87x 21

รูปที่ 3.10 แสดงพื้นที่ของหน่วยความจำของPIC 16F87x 23

รูปที่ 3.11 แสดงโครงสร้างของ A/D convertor ของพอร์ต A 28

รูปที่ 3.12 จอแสดงผลแบบผลึกเหลว(LCD) 30

รูปที่ 3.13 ขนาดของตัวแสดงผลแบบผลึกเหลว(LCD) 30

รูปที่ 3.14 แสดงหน้าที่การทำงานของขา(LCD) 31

รูปที่ 3.15 ไดอะแกรมการทำงานงานของชิปควบคุมโมดูล LCD แบบอักขระเบอร์ HD 44870 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.16 ตัวอย่างวงจรเพื่อใช้ในโมดูล LCD แบบ 8 บิต 34
รูปที่ 3.17 ตัวอย่างวงจรเพื่อใช้ในโมดูล LCD แบบ 4 บิต 35
รูปที่ 4.1 ตัวอย่างแสดงผลของเครื่องวัดสี 44
รูปที่ 4.2 เครื่องวัดวิเคราะห์สี(Color Analyzer) 48
รูปที่ 4.3 การสะท้อนแสงผ่านแผ่นกรองสีและLDR 49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันกระบวนการผลิตในอุตสาหกรรมต่างๆ มีการใช้เครื่องมือการคัดแยกและการตรวจวัดคุณภาพของผลิตภัณฑ์ซึ่งคุณภาพและความเร็วของการผลิตสำหรับโรงงานอุตสาหกรรมเป็นสิ่งสำคัญ ดังนั้นการนำเครื่องมือที่ทันสมัยมาใช้ในอุตสาหกรรมเป็นสิ่งจำเป็นแต่การนำเข้าเครื่องนั้นจะเป็นการนำเข้าจากต่างประเทศซึ่งจะมีราคาแพงและอาจจะไม่เหมาะสมกับผลิตภัณฑ์และกระบวนการผลิตภายในประเทศ

กระบวนการผลิตที่มีการคัดแยกและตรวจวัดคุณภาพของผลิตภัณฑ์ อาจจำเป็นต้องใช้อุปกรณ์อิเล็กทรอนิกส์เชิงแสงซึ่งเป็นการนำเอาอุปกรณ์ทางอิเล็กทรอนิกส์มาประยุกต์ใช้ ยกตัวอย่าง กระบวนการคัดแยกผลิตภัณฑ์ทางการเกษตรโดยใช้สีของผลิตภัณฑ์เป็นเกณฑ์ สีของเม็ดน้ำตาล สีของน้ำปลาหรือแม้กระทั่งกระบวนการพ่นสีของรถยนต์ซึ่งกระบวนการดังกล่าวจำเป็นต้องใช้อุปกรณ์อิเล็กทรอนิกส์เชิงแสงเข้ามาเป็นส่วนหนึ่งของเครื่องมือวัด

1.1 ขอบเขตการศึกษา

1. ศึกษาการใช้งานอุปกรณ์อิเล็กทรอนิกส์เชิงแสงรวมถึงวงจรที่ใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์เชิงแสง
2. ศึกษาการประมวลผลโดยใช้งานไมโครคอนโทรลเลอร์ ตระกูล PIC
3. ศึกษาการแสดงผลออกโดยใช้ ตัวแสดงผลแบบผลึกเหลว Liquid Crystal Display (LCD)
4. ศึกษาตัวอย่างการใช้งานเพื่อเปรียบเทียบกับงานจริง

1.2 วัตถุประสงค์ของการสร้าง

ในปัจจุบันวงจรและอุปกรณ์อิเล็กทรอนิกส์เกี่ยวกับแสงมีอยู่มาก เช่น อุปกรณ์ให้กำเนิดแสงสว่าง ได้แก่ ไดโอดเปล่งแสง(LED) หลอดทังสเตน หลอดนีออน หลอดฟลูออเรสเซนต์ และ หลอดซีนอน อุปกรณ์เหล่านี้สามารถเปล่งคลื่นแสงออกมา อุปกรณ์รับแสง เช่น ตัวต้านทานไวแสง Light dependent resistor (LDR) โฟโต้ไดโอด โฟโต้ทรานซิสเตอร์

ลักษณะการใช้งานของอุปกรณ์อิเล็กทรอนิกส์เชิงแสง

1. การควบคุมระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การนับด้วยความเร็วสูง
3. การคัดเลือกแบ่งชนิด
4. การตรวจสอบขนาด
5. การตรวจสอบอัตโนมัติ
6. การติดขัดของระบบสายพาน
7. การป้องกันอันตรายในการใช้เครื่อง
8. การจัดกลุ่มหรือกรองโดยอัตโนมัติ
9. การวัดปริมาณแสง

อุปกรณ์อิเล็กทรอนิกส์เชิงแสง เข้ามามีบทบาทอย่างมาก ดังนั้นจึงจำเป็นที่ต้องศึกษาว่าสามารถนำอุปกรณ์อิเล็กทรอนิกส์เชิงแสงมาประยุกต์ใช้ในกระบวนการอุตสาหกรรมได้อย่างไร

1.3 ความเป็นมาของโครงงาน

จากอุตสาหกรรมด้านการเกษตร การคัดแยกผลผลิตมีความจำเป็นในการคัดแยกของผลิตภัณฑ์เป็นเกณฑ์ในการคัดแยก เช่น การผลิตซอสมะเขือเทศ การคัดแยกผลมะเขือเทศที่จะใช้ในกระบวนการโดยมนุษย์เป็นผู้คัดแยกจะทำให้การคัดแยกเป็นมาตรฐานเดียวกันได้ยาก เนื่องจากการมองเห็นสีของมนุษย์แต่ละคนมีขอบเขตจำกัด อีกทั้งต้องใช้คนเป็นจำนวนมาก แรงงานต้องมีความชำนาญพิเศษ จากความเหนื่อยล้าอาจจะทำให้เกิดความผิดพลาดได้ อุตสาหกรรมการแยกสีเม็ดน้ำตาล การคัดแยกสีของเม็ดน้ำตาลเพื่อให้ได้คุณภาพเดียวกันอาจเป็นการยากถ้าใช้การคัดแยกจากคน อุตสาหกรรมการแยกคุณภาพสีของน้ำปลาก็เป็นอีกตัวอย่างหนึ่งที่จำเป็นต้องใช้เครื่องมือการคัดแยกเข้ามาคัดแยก การพ่นสีของรถยนต์การผสมสีและพ่นสีอาจจะทำให้เกิดความผิดพลาดได้ถ้าเกิดจากการมองเห็นสีของมนุษย์

ดังนั้นจากตัวอย่างที่กล่าวมา อุตสาหกรรมด้านการเกษตรหรือการพ่นสีของรถยนต์มีการนำเครื่องมือที่นำเอาอุปกรณ์อิเล็กทรอนิกส์เชิงแสงมาประยุกต์ใช้เพื่อให้ได้ประสิทธิภาพและความรวดเร็ว รวมทั้งได้ผลิตภัณฑ์ที่มีคุณภาพเป็นมาตรฐานซึ่งทำให้เกิดอุปกรณ์ที่เรียกว่า"เครื่องวัดวิเคราะห์สี(Color Analyzer)"

โครงงานนี้ผู้จัดทำได้ศึกษาเกี่ยวกับ เซ็นเซอร์วัดแสง เพื่อนำมาใช้ตรวจสอบ รวมถึงการศึกษาวงจรอิเล็กทรอนิกส์ที่ใช้ควบคู่ ศึกษาการใช้งานไมโครคอนโทรลเลอร์ ตระกูล PIC และศึกษาการแสดงผลออกหน้าจอให้เหมาะสมกับการใช้งานจริง

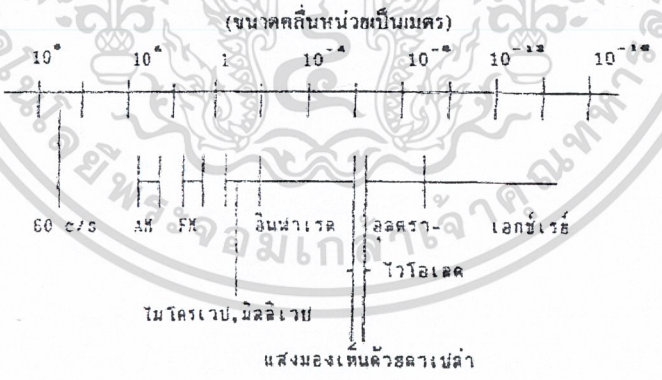
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี

เครื่องวัดวิเคราะห์สี(Color Analyzer)ใช้หลักการสะท้อนแสง การมองเห็นสี และการรวมสี ประกอบเข้าด้วยกัน เพื่อสามารถวิเคราะห์และแยกความแตกต่างของสีที่มองเห็นได้ จนกระทั่งผ่านกระบวนการวิเคราะห์จากเครื่องวัดวิเคราะห์สีได้ จากหลักการข้างต้นการศึกษาเรื่องคุณสมบัติของแสงและสีจึงมีความสำคัญอย่างมาก

2.1 แสง(Light)

แสงเป็นคลื่นสนามแม่เหล็กไฟฟ้า ฉะนั้นแสงจึงมิได้หมายถึงแสงที่มองเห็นด้วยตาเปล่าเท่านั้น แสงโดยทั่วไปมีอยู่สองประเภท คือ แสงที่สายตามนุษย์เรามองเห็น (visible rays) กับแสงที่สายตาเรามองไม่เห็น ซึ่งได้แก่ แสงจำพวก แกมม่า-เรย์ (gamma-rays) เอ็กซ์-เรย์ (x-rays) อุลตราไวโอเลต เรย์ (ultraviolet rays) และอินฟราเรด เรย์ (infrared rays) แสงเหล่านี้ต่างก็เป็นส่วนหนึ่งของพลังงานแม่เหล็กไฟฟ้าแต่มีความถี่และความยาวคลื่นแตกต่างกันออกไป ดังรูป



รูปที่ 2.1 ขนาดความยาวคลื่นของแสง

แสงที่ตาคนมองเห็นจะมีความยาวคลื่นอยู่ในช่วง 380 นาโนเมตร จนถึงประมาณ 780 นาโนเมตร คือ สี 7 สี คือ ม่วง คราม น้ำเงิน เขียว เหลือง ส้ม แดง

ส่วนแสงอุลตราไวโอเลต (ultraviolet) เป็นแสงที่มีความยาวคลื่นต่ำกว่าคลื่นสีม่วง และแสงอินฟราเรด(infrared) เป็นแสงที่มีความยาวคลื่นสูงกว่าคลื่นสีแดง

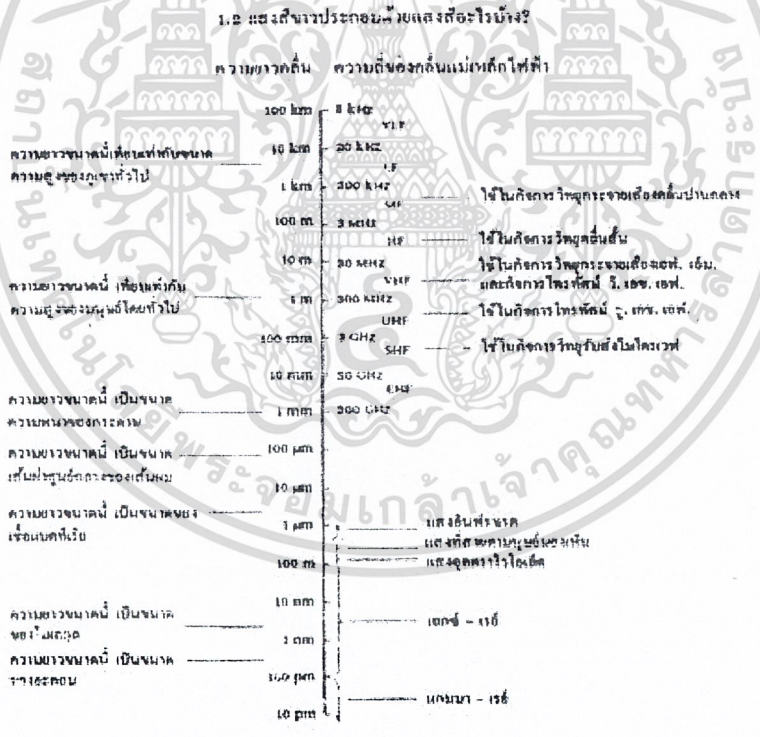
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสงที่เรามองเห็นนี้จะทำให้สายตาของคนเราได้รับความรู้สึก 2 ประการ ประการแรกคือความรู้สึกว่ามีแสงสว่างมากหรือน้อย (sensation of brightness) ความรู้สึกอีกประการหนึ่งคือทำให้รู้จักว่าเป็นแสงสีอะไร ความรู้สึกนี้เรียกว่า sensation of color

สำหรับความสว่างของแสงจะขึ้นอยู่กับความห่างระหว่างต้นกำเนิดแสง และฉากโดยจะเป็นส่วนกลับกำลังสองของความห่าง

2.1.1 แสงสีขาว

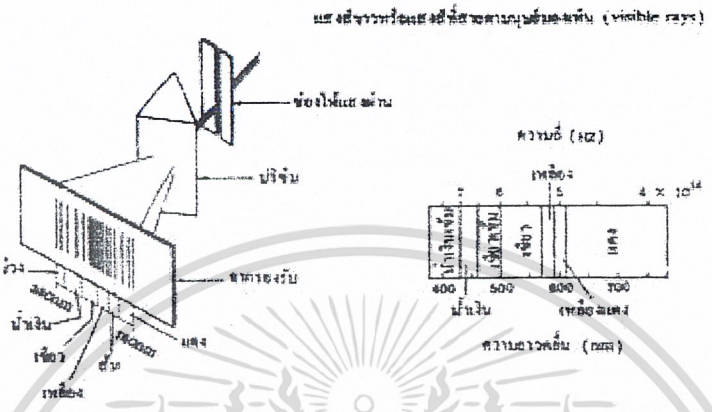
แสงสีที่มนุษย์มองเห็นมีอยู่ 2 ประการ คือแสงสีโดยตรงที่เกิดจากแหล่งกำเนิดแสงอย่างใดอย่างหนึ่งเช่นแสงจากดวงอาทิตย์เป็นต้น อีกประการหนึ่งก็คือแสงที่สะท้อนจากสิ่งอื่นวัตถุบางอย่างเมื่อเกิดการกระทบแสงสีจะมีการดูดซึมได้ดีแต่วัตถุบางอย่างจะสะท้อนแสงได้ดีด้วยเหมือนแสงสีขาวคือส่วนของคลื่นแม่เหล็กไฟฟ้าที่สายตามนุษย์มองเห็นตามที่แสดงไว้ในรูปที่ 2.2



รูปที่ 2.2 รายละเอียดของคลื่นแม่เหล็กไฟฟ้าที่มีความถี่หรือความยาวคลื่นต่างกัน

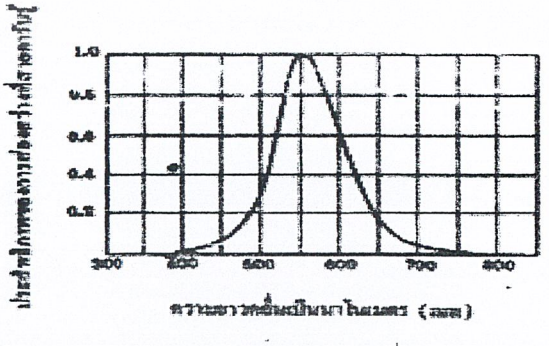
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสงสีขาวประกอบด้วยแสงสี 7 สีซึ่งอาจตรวจสอบทดลองดูได้โดยการให้แสงจากดวงอาทิตย์ ซึ่งเป็นแสงสีขาวผ่านปริซึม(prism)แล้วใช้ฉากรับตามที่แสดงไว้ในรูปที่ 2.3



รูปที่ 2.3 แสงสีที่เกิดจากการหักเหของแสงสีขาวผ่านปริซึม

การที่แสงต่างๆปรากฏขึ้นเช่นนี้ เป็นเพราะแสงสีแต่ละชนิดนั้นมีความยาวคลื่นไม่เท่ากัน แสงที่มีความยาวคลื่นมาก(เช่นสีแดง)ก็จะหักเหไปมาก ส่วนแสงที่มีความยาวคลื่นน้อย (เช่นแสงสีม่วง)ก็จะหักเหไปน้อย ฉะนั้น จึงได้แสงสีต่างๆ เรียงลำดับกันบนฉากที่รอรับเอาไว้ แสงสีที่เรียงลำดับกันเหล่านี้ เรียกว่า สเปกตรัม(spectrum) ซึ่งเรามักจะพบเห็นกันบ่อยๆ ในปรากฏการณ์ธรรมชาติหลังจากฝนตก ที่เรียกว่ารุ้งกินน้ำ อย่างไรก็ตามสายตามนุษย์ไม่สามารถไวต่อแสงสีได้เหมือนกันสม่ำเสมอ



รูปที่ 2.4 ความสามารถของสายตามนุษย์ ในการรับรู้การเปลี่ยนแปลงความเข้มของแสงสี

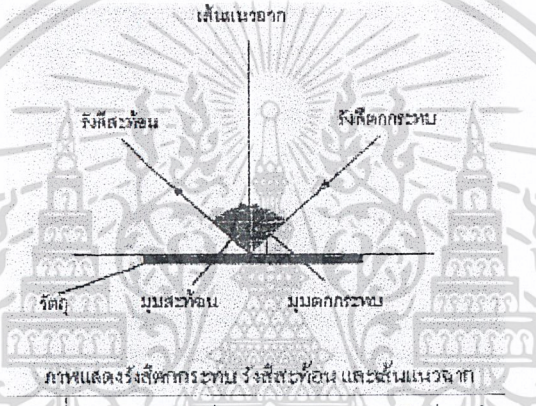
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่แสดงนี้ อาจกล่าวได้ว่าความรู้สึกของมนุษย์ที่มีต่อการเปลี่ยนแปลงความเข้มของแสงนั้น มีอยู่ความยาวคลื่นประมาณ 400 นาโนเมตร(ซึ่งเป็นส่วนหนึ่งของสีม่วง) จนถึงประมาณ 750 นาโนเมตร(ซึ่งเป็นส่วนหนึ่งของสีแดง) เท่านั้น

2.1.2 การสะท้อนแสง

การสะท้อนของแสงที่ผิวราบ

เมื่อรังสีของแสงตกกระทบผิววัตถุที่จุดใดก็ตาม ถ้าเราลากเส้นตั้งฉาก กับผิววัตถุนั้น เส้นตั้งฉากที่ลากนี้เรียกว่า เส้นแนวฉาก และเรียกมุมที่รังสีตกกระทบทำกับเส้นแนวฉากว่า มุมตกกระทบ มุมที่รังสีสะท้อนทำกับแนวฉาก เรียกว่า มุมสะท้อน

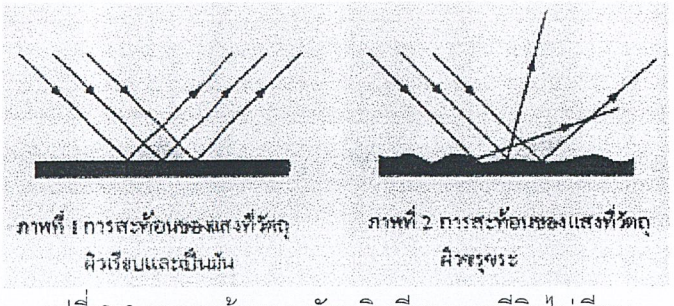


รูปที่ 2.5 ภาพรังสีตกกระทบ และ รังสีสะท้อน

กฎการสะท้อนของแสงมีดังนี้

1. รังสีตกกระทบ รังสีสะท้อน และเส้นแนวฉาก อยู่บนระนาบเดียวกัน
2. มุมตกกระทบเท่ากับมุมสะท้อน

วัตถุที่สะท้อนแสงได้ดีจะต้องมีผิวเรียบและเป็นมัน เช่น กระจกเงา จะทำให้เกิดการสะท้อนอย่างมีระเบียบ ดังภาพที่ 1 รูปที่ 2.6 แต่ถ้าวัตถุที่มีผิวไม่เรียบ จะเกิดการสะท้อนไม่มีระเบียบ ดังภาพที่ 2 รูปที่ 2.6 แต่การสะท้อนของแสงเป็นไปตามกฎการสะท้อนของแสง



รูปที่ 2.6 การสะท้อนบนวัตถุผิวเรียบและมีผิวไม่เรียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 สี(color)

สีเป็นสิ่งที่ทุกคนสนใจแต่การศึกษาเรื่องสีนั้นจะต่างกันแล้วแต่ว่าแต่ละสาขาวิชาชีพจะให้ความสำคัญในแง่ใด รวมทั้งการวิเคราะห์เกี่ยวกับสีเพื่อนำไปใช้

การจัดจำพวกสี

1. ฮิว(hue) ในการศึกษาเรื่องสเปกตรัม จะเห็นว่าแสงที่ผ่านเข้าไปในปริซึมแล้วกระจายเป็นสเปกตรัมซึ่งตาจะรู้สึกเป็นสีต่างๆแถบสีต่างๆในสเปกตรัมเรียกว่าฮิว เช่น แถบสีแดง ส้ม เหลือง เขียว น้ำเงิน ม่วง ในสเปกตรัมของแสงอาทิตย์จะมีฮิวไม่ครบทั้งหมด เช่น สีม่วงแดง สีม่วงแกมน้ำเงิน
2. ความเจิดจ้า(lightness) คือความรู้สึกติดตาติดใจของจำนวนแสงที่สะท้อนออกมาจากแสงสีแต่ละแสงสีซึ่งทำให้เกิดความรู้สึกว่า มีแสงผ่านเข้าตามากหรือน้อย
3. ความอิ่มตัว(saturation) คือความแตกต่างมากน้อยจากสีเทาที่มีความสว่างเดียวกันถ้ามีความอิ่มตัวมาก เรียกว่าเข้มและถ้ามีความอิ่มตัวน้อย เรียกว่าอ่อน ส่วนสีเทามีความอิ่มตัวเป็นศูนย์

คุณสมบัติ 3 ประการ คือ ฮิว, ความเจิดจ้า, ความอิ่มตัว เป็นสื่อใช้ในการจัดจำพวกสี

การวัดสี

เราตรวจสอบวัตถุที่มีสีโดยให้แสงผ่านเข้าไปในวัตถุนั้นหรือให้แสงสะท้อนจากผิวของมัน สีของวัตถุที่สังเกตเห็นได้นั้นนอกจากขึ้นอยู่กับลักษณะเฉพาะของวัตถุแต่ละชนิดแล้ว ยังขึ้นอยู่กับแสงที่ใช้ส่องสว่างอีกด้วยแสงที่ใช้เป็นมาตรฐานในการส่องบนวัตถุที่มีสี คือ แสงอาทิตย์

ค่าของ ฮิว และ ความอิ่มตัว ของวัตถุ(หรือเรียกว่า Chromaticity ของวัตถุ) บอกได้ด้วยเศษส่วนของพลังงานที่ตกกระทบจากการส่องสว่างมาตรฐานซึ่งวัตถุที่ยอมให้แสงผ่านหรือสะท้อนจากผิวในแต่ละช่วงของความยาวคลื่นเครื่องมือที่ใช้วัดเศษส่วนนี้เรียกว่า สเปกโตรโฟโตมิเตอร์ (Spectrophotometer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การมองเห็นสี

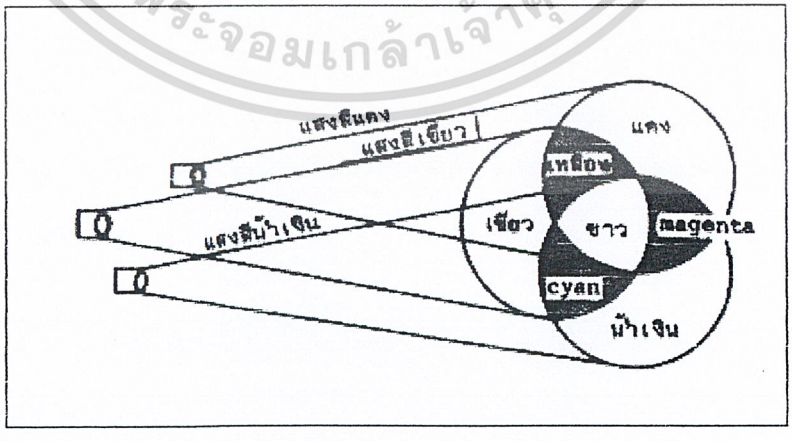
แสงสี แดง เขียว น้ำเงิน เรียกว่า แม่สี หรือ สีปฐมภูมิ (primary color) การมองเห็นสีนั้นอยู่ที่ว่าประสาทรับแสงสีชุดใดถูกกระตุ้นมากน้อยเพียงใด ถ้าประสาทรับสีแดง ถูกกระตุ้นเพียงประสาทเดียวก็จะมีความรู้สึกเห็นสีแดง ส่วนสีเขียว และ สีน้ำเงินก็เช่นกัน แต่ถ้าประสาทรับแสง 2 อันหรือ 3 อันถูกกระตุ้นพร้อมกัน เช่น ประสาทรับแสงสีแดงถูกกระตุ้นพร้อมกับสีเขียว ก็จะเห็นสีเป็นสีเหลือง(yellow)ประสาทรับแสงสีแดงถูกกระตุ้นพร้อมกับสีน้ำเงินจะเห็นเป็นสีม่วง(magenta) ถ้าประสาทรับแสงสีน้ำเงินถูกกระตุ้นพร้อมกับสีเขียวจะเห็นเป็นสีเขียวแกมน้ำเงิน(cyan)

2.2.1 แสงสี

1. แสงสีปฐมภูมิ คือ แสงสีหลักไม่สามารถเกิดจากการรวมกันของสีอื่น เป็นแสงสีเฉพาะตัว ได้แก่ สีน้ำเงิน สีแดง และสีเขียว
2. แสงสีทุติยภูมิ คือ แสงสีที่เกิดจากการรวมกันของแสงสีปฐมภูมิ ได้แก่ สีเหลือง(yellow) สีเขียวแกมน้ำเงิน (cyan) และสีม่วง (magenta)
3. แสงสีคู่เติมเต็ม คือ แสงที่รวมกันแล้วได้แสงสีขาว มี 3 คู่ คือ แดง+เขียวแกมน้ำเงิน เหลือง+น้ำเงิน และ ม่วง+เขียว

การผสมแสงสี

เมื่อฉายแสงสีขาวผ่านแก้วสีกรองแสง แล้วแสงสีแดง แสงสีเขียว และแสงสีน้ำเงิน ที่ไปตกบนฉากจะเกิดผลดังนี้



รูปที่ 2.7 การผสมแสงสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสงสีแดง + แสงสีเขียว = แสงสีเหลือง

แสงสีแดง + แสงสีน้ำเงิน = แสงสีม่วง (magenta)

แสงสีน้ำเงิน + แสงสีเขียว = แสงสีเขียวแกมน้ำเงิน(cyan)

แสงสีแดง + แสงสีเขียว + แสงสีน้ำเงิน = แสงสีขาว

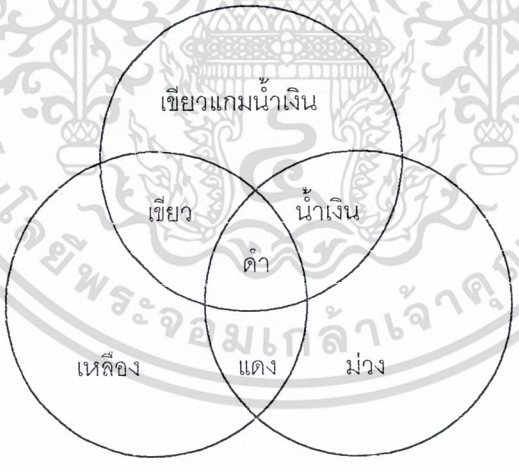
แสงสีแดง + แสงสีเขียวแกมน้ำเงิน(cyan) = แสงสีขาว

แสงสีเขียว + แสงสีม่วง(magenta) = แสงสีขาว

แสงสีน้ำเงิน + แสงสีเหลือง = แสงสีขาว

2.2.2 สารสี

1. สารสีปฐมภูมิ คือ สารสีหลักซึ่งไม่สามารถเกิดจากการรวมกันของสารสีอื่น ได้แก่ สีเหลือง (yellow) เขียวแกมน้ำเงิน(cyan) และสีม่วง(magenta)
2. สารสีทุติยภูมิ คือ สารสีที่เกิดจากการรวมกันของสารสีปฐมภูมิ ได้แก่ สีน้ำเงิน สีแดง และสีเขียว



รูปที่ 2.8 การผสมสารสี

3. สารสีคู่เติมเต็ม คือ สารสีที่รวมกันแล้วได้ สารสีดำมี 3 คู่ คือ ม่วง + เขียว , แดง + เขียวแกมน้ำเงิน และ เหลือง + น้ำเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารสี	ดูดกลืนแสงสี	สะท้อนแสงสี
ม่วง	เขียว	น้ำเงิน แดง
เหลือง	น้ำเงิน	แดง เขียว
เขียวแกมน้ำเงิน	แดง	เขียว น้ำเงิน
แดง	น้ำเงิน เขียว	แดง
เขียว	แดง น้ำเงิน	เขียว
น้ำเงิน	แดง เขียว	น้ำเงิน
ดำ	แดง เขียว น้ำเงิน	-
ขาว	-	แดง เขียว น้ำเงิน

ตารางที่ 2.1 การดูดกลืนและสะท้อนแสงสีของสารสี

หมายเหตุ แสงสีขาว ส่องวัตถุเห็นวัตถุเป็นสีเดิมเสมอ
แสงสีดำ ส่องวัตถุเห็นวัตถุเป็นสีดำเสมอ

ตัวอย่างเช่น สีทาบ้านที่เห็นเป็นสีเหลืองนั้น เกิดจากการสะท้อนแสงสีแดง เขียว ดูดกลืนแสงสีน้ำเงิน ซึ่งคือแสงสีเหลืองนั่นเอง ส่วนสีทาบ้านที่ผสมกันระหว่างสีเหลืองกับสีเขียวแกมน้ำเงินจะได้สีผสมทาบ้านเป็นสีเขียว เนื่องจากสีเขียวแกมน้ำเงิน ดูดกลืนแสงสีแดงแต่สะท้อนแสงสีเขียวและสีน้ำเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 ช่วงความยาวคลื่นของสี

ความยาวคลื่นของแสงที่มองเห็นได้ชัด แบ่งออกเป็น 8 ระยะ ตามคุณสมบัติของสี

ความยาวคลื่นของ(นาโนเมตร)	สี(color)
340-450	ม่วง
450-480	น้ำเงินเข้ม
480-510	น้ำเงิน
510-550	เขียว
550-575	เหลืองเขียว
575-585	เหลือง
585-620	ส้ม
620-760	แดง

สายตาคนเราไวต่อการเปลี่ยนแปลงมากที่สุด คือ แสงระหว่างสีเหลืองกับสีเขียวซึ่งมีความยาวคลื่นประมาณ 550 นาโนเมตร

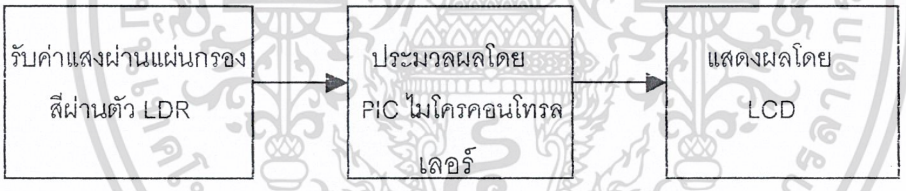
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่3
หลักการออกแบบและการสร้าง

เครื่องวัดวิเคราะห์สีเป็นเครื่องมือวัดสีของวัตถุต้นแบบ จะให้ค่าสีอ้างอิงของวัตถุ 3 สี คือ สีเหลือง (yellow) สีม่วง (magenta) และสีเขียวแกมน้ำเงิน(cyan) แล้วบันทึกค่าไว้แล้วนำวัตถุชิ้นต่อๆไปมาวัดทำการหาค่า ความคลาดเคลื่อน เมื่อเทียบกับวัตถุต้นแบบ เพื่อที่จะสามารถบ่งบอกได้ว่าวัตถุใดสามารถยอมรับได้

ตัวอย่างเช่น การคัดแยกมะเขือเทศ โดยการใช้เครื่องวัดวิเคราะห์สี นำมะเขือเทศที่เป็นต้นแบบมาทำการวัด หาค่า 3 สีออกมาแล้วนำนำมะเขือเทศผลอื่นๆที่ต้องการคัดคุณภาพมาวัดสี เมื่อทำการวัดมะเขือเทศที่เป็นต้นแบบมาวัดสีบันทึกค่าสีที่ได้และนำมะเขือเทศที่ต้องการวัด คัดคุณภาพมาวัดสีเช่นกัน เครื่องวัดจะทำการคำนวณหาค่าความคลาดเคลื่อน ของ 3 สี จะสามารถบอกคุณภาพของมะเขือเทศ จากการกำหนดค่าแตกต่างของสีระหว่างมะเขือเทศต้นแบบ กับมะเขือเทศที่ต้องการคัดคุณภาพของผู้ผลิตได้ เป็นต้น

หลักการของเครื่องวัดวิเคราะห์สี



ส่วนประกอบที่สำคัญของเครื่องมือวัดวิเคราะห์สี แบ่งออกเป็น 3 ส่วนหลัก คือ

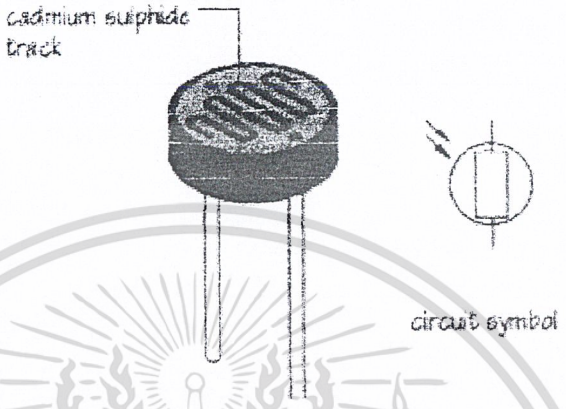
1. ตัวรับแสง LDR (Light dependent resistor)
2. ตัวประมวลผล PIC - MICROCONTROLLER
3. ตัวแสดงผล LCD (Liquid crystal display)

3.1 ตัวต้านทานไวแสง(Light dependent resistor)

ตัวต้านทานไวแสงหรือแอลดีอาร์ (ค่าความต้านทานที่แปรค่าได้กับแสง) โดยเมื่อป้อนพลังงานอย่างเพียงพอให้กับวาเลนซ์อิเล็กตรอนของสารกึ่งตัวนำจะทำให้เกิดการแตกตัวของโฮลและอิเล็กตรอนเกิดขึ้นมากมาย พลังงานที่จะทำให้โฮลและอิเล็กตรอนแตกตัวนี้ต้องเป็นพลังงานที่มาจากภายนอก เช่น แสง ความร้อน ปริมาณของพลังงานที่ต้องการทำให้เกิดการแตกตัวของอิเล็กตรอน - โฮล จะขึ้นอยู่กับชนิดของสารนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอลดีอาร์ส่วนใหญ่ทำมาจากสารกึ่งตัวนำจำพวกแคดเมียมซัลไฟต์หรือแคดเมียมเซเลไนด์ ซึ่งให้ผลตอบสนองต่อแสงได้ดีที่ความยาวคลื่นประมาณ 400 - 1000 นาโนเมตร ซึ่งเป็นแถบแสงที่มองเห็นพอดี

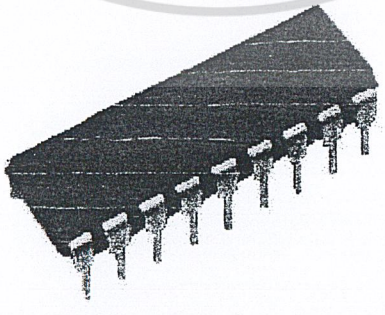


รูปที่ 3.1 โครงสร้างและสัญลักษณ์ทางไฟฟ้า LDR

ลักษณะการทำงานของแอลดีอาร์ จะเห็นว่าเมื่อแสงมีความเข้มมากๆมาตกกระทบ ก็จะทำให้ตัวมันสามารถนำกระแสไฟฟ้าได้ เป็นผลทำให้ความต้านทานในตัวมันลดลง ช่วงระหว่าง 2 $M\Omega$ (ที่มืด) 100 Ω (ห้องที่สว่าง)

ข้อดีการใช้งานของแอลดีอาร์ คือ มีความไวต่อแสงสูง ราคาถูก และมีขนาดเล็กมีการเปลี่ยนแปลงค่าได้กว้างแต่มีข้อเสียเมื่อเทียบกับอุปกรณ์รับแสงอื่นๆ คือ มีผลตอบสนองต่อความถี่ของแสงได้แคบ

3.2 ไมโครคอนโทรลเลอร์ตระกูล PIC



รูปที่ 3.2 ไมโครคอนโทรลเลอร์ตระกูล PICชนิด 18 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIC คือ ไมโครคอนโทรลเลอร์ อีกตระกูลหนึ่ง ย่อมาจากคำว่า Peripheral Interface Controller ซึ่ง แนวทาง ของ ไมโครคอนโทรลเลอร์ ตระกูลนี้ก็คือ รวมเอาทุกอย่างเอาไว้ในตัวของมันไม่ว่าจะเป็นโปรแกรมหน่วยความจำ(PROGRAM MEMORY),แรม(RAM),อีสคแวร์พรอม (EEPROM), ซีเรียล(SERIAL),ไอทูซี (I2C), PWM,อนาลอกเป็นดิจิตอล (A/D) ฯลฯ โดยไม่จำเป็นต้องต่ออุปกรณ์เสริมจากภายนอก ในตัวของ PIC จะมีฟังก์ชันที่ใช้ในการประมวลผล รวมทั้งหน่วยความจำ

ความเร็วของ PIC ภาคของความถี่สัญญาณนาฬิกา ปัจจุบันสามารถทำสัญญาณนาฬิกาได้ที่ 20 เมกกะเฮิร์ซ ซึ่งทำให้หนึ่งคำสั่งของ PIC ใช้เวลาเพียง 0.25 ไมโครวินาที แต่อย่างไรก็ตามได้มีบริษัทอื่นซื้อลิขสิทธิ์ PIC จาก microchip และได้สร้าง ชิป ที่มีความเร็วได้มากกว่าเดิมขึ้นไปอีก

หน่วยความจำของ PIC ในอดีตหน่วยความจำของ PIC จะค่อนข้างน้อย คืออยู่ระหว่าง 512 เวิร์ด ถึง 4กิโลเวิร์ด แต่ในปัจจุบัน บริษัท microchip ซึ่งเป็นเจ้าของ PIC ได้พัฒนาจนทำให้หน่วยความจำ ของ PIC มีขนาดเป็นหลายสิบกิโลไบต์ และมีทีท่าว่าจะขยายได้ใหญ่ขึ้นเรื่อยๆ ในเรื่องของการนับขนาดของหน่วยความจำของ PIC จะนับไม่เหมือนปกติ โดยที่ หนึ่งคำสั่งของ PIC จะมีขนาด 14 บิต ดังนั้นจะเรียกว่า 1 เวิร์ด ของ PIC จะมีขนาด 14 บิต เช่น PIC16F84A ระบุว่ามีหน่วยความจำ 1 K (ซึ่งหมายถึง 1 กิโลเวิร์ด ถ้าคำนวณเป็นแบบ 1 ไบต์ = 8 บิต จะได้ว่า 1 x 1,024 x 14 = 14,336 บิต ดังนั้นก็คือ 14,336/(8 x 1,024) = 1.75K ไบต์ สถาปัตยกรรมของ PIC ปัจจุบันมี 3 สายหลักๆ ในอดีตมีแค่สอง คือขึ้นต้นด้วย 16xxx,17xxx และล่าสุดคือ 18xxx คุณสมบัติที่เหนือกว่าเรียงจากน้อยสุดไปมากสุดคือ 16 -> 17 -> 18 คำสั่ง แอสแซมบลี ของ 17 และมี 18 จะมีมากกว่า 16 ทำให้เขียนโปรแกรมได้ง่ายกว่า ราคาก็จะสูงกว่าแต่ที่เป็นที่นิยมก็คือตระกูล 16xxx สถาปัตยกรรมของ PIC จะยึดถือการออกแบบที่ว่ารวมทุกอย่างไว้ใน ชิป ตัวเดียวโดยไม่ต้องต่ออุปกรณ์ใดๆ เพิ่มเติม ผลที่ตามมาก็คือแผ่นวงจรจะมีขนาดเล็ก และอุปกรณ์ที่ใช้จะไม่มาก บางงานอาจจะใช้แค่ PIC เพียงตัวเดียวโดยไม่ต้องใช้ ชิป อื่นมาเพิ่มเติม นี่คือคุณสมบัติพิเศษของ PIC ซึ่งปัจจุบันหลายบริษัทที่ผลิต ไมโครคอนโทรลเลอร์ ก็เริ่มจะหันมาเลียนแบบแนวทางนี้ แต่มีข้อเสีย ที่การรวมทุกอย่างไว้ใน ชิป เดียว ทำให้ โปรแกรมหน่วยความจำ และ ข้อมูลหน่วยความจำไม่สามารถขยายโดยใช้กับหน่วยความจำ ภายนอกได้ (ในทางทฤษฎี) ของจริงทำได้ แต่ต้องใช้เทคนิคซึ่งไม่นิยม PIC จึงเหมาะสำหรับงานเล็กๆ ไม่ใช้ในงานใหญ่ๆ ที่ต้องใช้การคำนวณ และหน่วยความจำ มากๆ

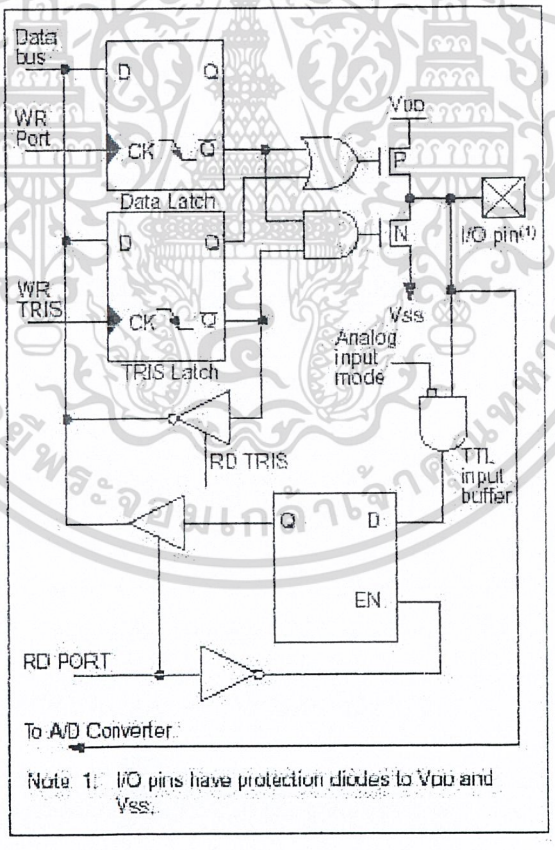
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเครื่องวัดวิเคราะห์สี (Color Analyzer) จำเป็นต้องมี อนาลอกเป็นดิจิตอลหรือ ADC(Analog to Digital converter) จำนวน 4 ช่องสัญญาณ(Channel) และต้องการ 3 พอร์ต ดังนั้น จึงสามารถเลือก PIC เบอร์ PIC 16F876 ซึ่งมี 5 ช่องสัญญาณ 10 บิต และมี 3 พอร์ต คือ พอร์ต A,B,C มีจำนวน 28 พิน ซึ่งถือว่าเพียงพอต่อความต้องการของเครื่องวัดวิเคราะห์สี (Color Analyzer)

3.2.1 ระบบบัสของ I/O ของ PIC 16F876

3.2.1.1 พอร์ต A ใน PIC 16F876

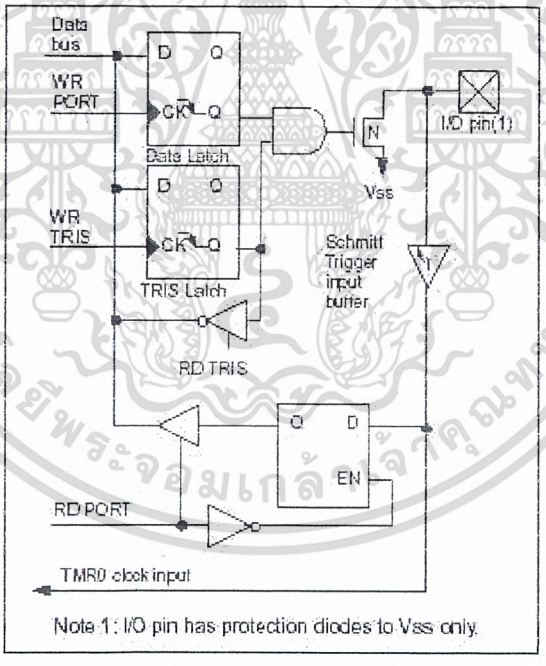
อินพุท/เอาท์พุท พอร์ตบางตัวของ PIC เป็นแบบ multiplexed ซึ่ง อาจเป็นทั้ง I/O หรือ peripheral features (เช่น A/D, Serial, IIC) ซึ่งเมื่อ ขาเหล่านี้ใช้งานในส่วน peripheral ก็จะไม่สามารถใช้งานในลักษณะของ I/O ได้ พอร์ต A และ TRISA รีจิสเตอร์



รูปที่ 3.3 บล็อกไดอะแกรมของขา RA3:RA0 และ RA5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ต A มีขนาด 6 บิต ซึ่งเป็น พอร์ต ที่เป็นได้ทั้ง อินพุท และ เอาท์พุท โดยต้องเลือกแบบใดแบบหนึ่ง สามารถเลือกได้จาก รีจีสเตอร์ ที่มีชื่อว่า TRISA ซึ่งถ้า TRISA บิต ถูก เซท(set) เป็น '1' พอร์ต A ที่มีหมายเลขบิตเดียวกันนั้นก็ทำงานเป็น อินพุท (ทำให้ พอร์ต นั้นอยู่ในสถานะ hi-impedance) ส่วนถ้า TRISA บิต ถูก เซท เป็น '0' พอร์ต A ที่มีหมายเลขบิตเดียวกันนั้นก็จะทำงานเป็น เอาท์พุท (พอร์ต จะอยู่ในสถานะ output latch) การอ่านค่า พอร์ต A รีจีสเตอร์ คือการอ่านค่าสถานะของขา พอร์ต A ในขณะนั้น ส่วนการเขียนค่าไปยัง พอร์ต A คือการเขียนไปยัง แลชท์(latch) ของพอร์ต ลักษณะการเขียนจะเป็นแบบ read-modify-write operations ซึ่งหมายความว่า ในการเขียนไปยัง พอร์ต จะเริ่มด้วยการ อ่านค่า พอร์ต นั้นมาก่อนแล้วทำการเปลี่ยนแปลงค่า จากนั้นก็ทำการเขียนกลับไปยัง พอร์ต-แลชท์ อีกครั้งหนึ่ง ขา RA4 จะ multiplexed กับ Timer0 module clock อินพุท ซึ่งจะเรียกรวมๆ ว่า RA4/T0CKI โดยที่ ขา RA4/T0CKI จะเป็นลักษณะ อินพุท แบบ Schmitt Trigger และ เอาท์พุท แบบ open drain. พอร์ต RA ทั้งหมด จะมี TTL อินพุท level และมี เอาท์พุท แบบ full CMOS drivers



รูปที่ 3.4 บล็อกไดอะแกรมของขา RA4/TOCKI

ส่วน พอร์ต A ขาอื่นๆ จะ multiplex กับ อนาลอกอินพุท และ Vref ของ A/D อินพุท ซึ่งการกำหนดการทำงานของแต่ละขา สามารถเลือกได้โดย เคลียร์(clear) หรือ เซทบิตควบคุม ใน ADCON1 รีจีสเตอร์

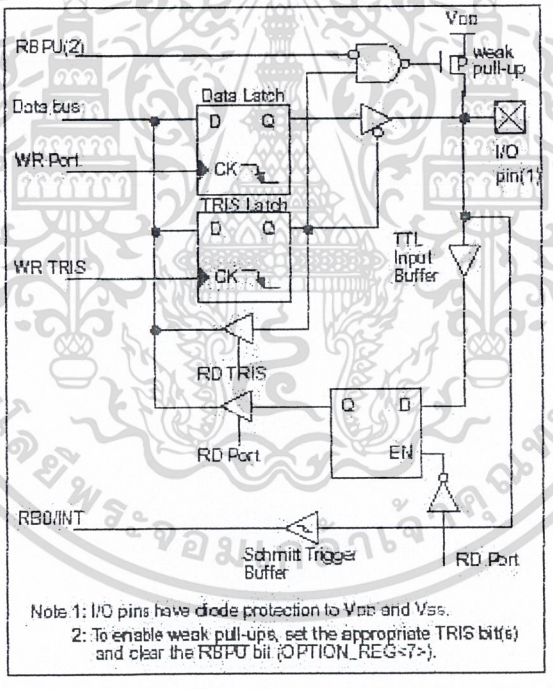
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ในขณะที่เกิด Power-on Reset ขาเหล่านี้จะถูก ระบุ ให้เป็น อนาลอกอินพุท และจะอ่านค่าได้เป็น '0' TRISA รีจีสเตอร์ มีหน้าที่ควบคุมว่าขา พอร์ต A ใดจะเป็น อินพุท/เอาท์พุท ในกรณีที่ใช้ พอร์ต A เป็น อนาลอกอินพุท TRISA รีจีสเตอร์ จะต้องถูก เซท

3.2.1.2 พอร์ต B ใน PIC 16F87X

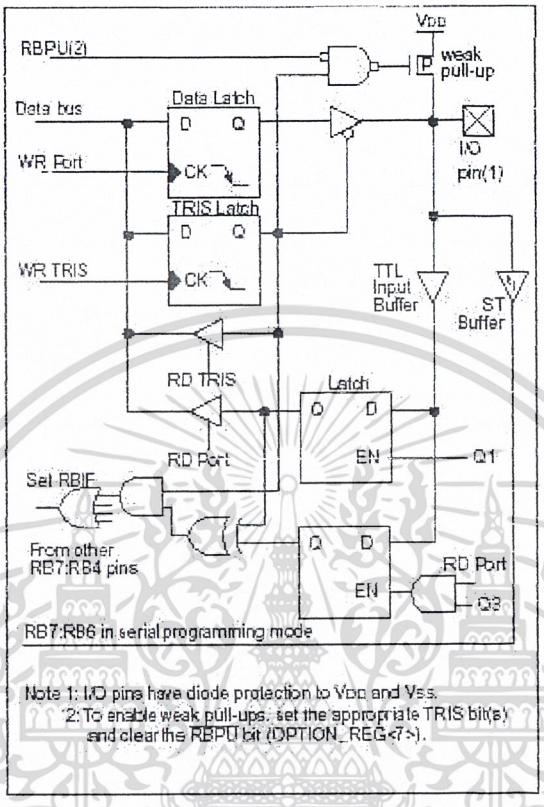
พอร์ต B เป็นลักษณะแบบพอร์ตแบบสองทิศทาง ซึ่ง รีจีสเตอร์ ที่จะเป็นตัวกำหนดว่า พอร์ต ใดจะเป็นแบบ อินพุท/เอาท์พุท จะถูกกำหนดโดย TRISB รีจีสเตอร์ ถ้า เซท TRISB บิต ใด (=1) พอร์ต B ที่บิตนั้นก็จะเป็น อินพุท ถ้า เคลียร์ TRISB บิต ใด (=0) พอร์ต B ที่บิตนั้นก็จะเป็น เอาท์พุท ขาสามขาของ พอร์ต B จะ multiplexed กับ Low Voltage Programming function ซึ่ง ได้แก่ RB3/PGM, RB6/PGC และ RB7/PGD



รูปที่ 3.5 บล็อกไดอะแกรมของขา RB3:RB0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50183



รูปที่ 3.6 บล็อกไดอะแกรมของขา RB7:RB4

พอร์ต B แต่ละ พอร์ต จะมี weak (พูลอัป)pull-up อยู่ภายใน (ถ้าต้องการ พูลอัป แข็งๆ ต้องต่อวงจรภายนอก) สามารถกำหนดว่าจะใช้ พูลอัป ภายในหรือไม่จากการ เซท หรือ เคลียร์ RBPU\ (OPTION รีจีสเตอร์ บิต 7) โดยถ้า เคลียร์ RBPU\ จะหมายถึง ทำการ ดีสเอเบิล พูลอัป ภายใน และถ้ากำหนดให้ พอร์ต B เป็น เอาท์พุท แล้ว พูลอัป จะถูก ดีสเอเบิล โดยอัตโนมัติ สำหรับ พอร์ต B นั้น ขา RB4-RB7 จะมี ลักษณะ เพิ่มเติมก็คือ การกำหนดให้เกิด อินเทอร์รัปพ์ เมื่อเกิดการเปลี่ยนแปลงของสถานะของสัญญาณไฟฟ้าที่ขา RB4-RB7 (โดยถ้าขาใดขาหนึ่งเกิดเปลี่ยนสถานะก็จะทำให้เกิด RB พอร์ต อินเทอร์รัปพ์เปลี่ยน ขึ้น ซึ่งจะทำให้ RBIF (INTCON.0) flag ถูก เซท โดยที่ อินเทอร์รัปพ์ ประเภทนี้สามารถทำการ “wake” microcontroller จากสถานะ sleep mode ได้ RBIF flag จะถูก Clear ได้ 2 กรณี คือ

1. ทำการอ่านหรือเขียน พอร์ต B

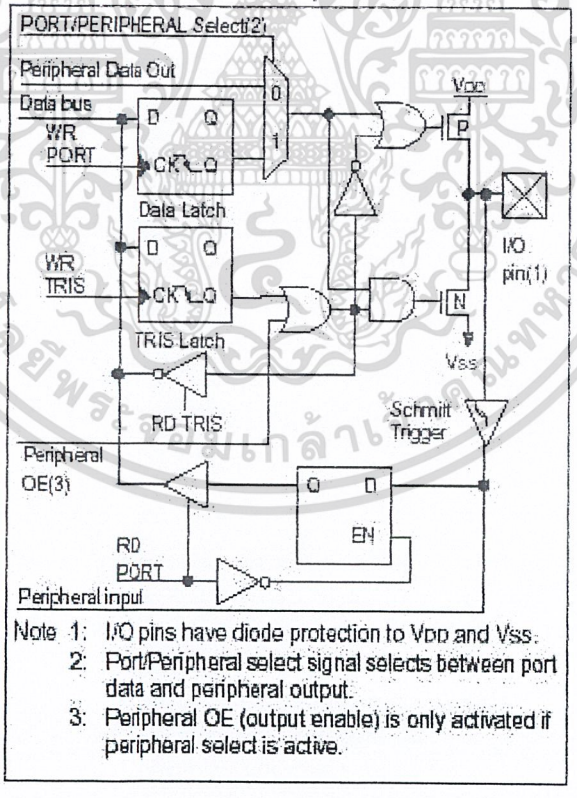
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ทำการ เคลียร์ RBIF flag โดยตรง

หากใช้ อินเทอร์รัปพ์ บน พอร์ต B เปลี่ยนน แล้วไม่ควรจะ อีนาเบิล พูลอัป ของ พอร์ต B

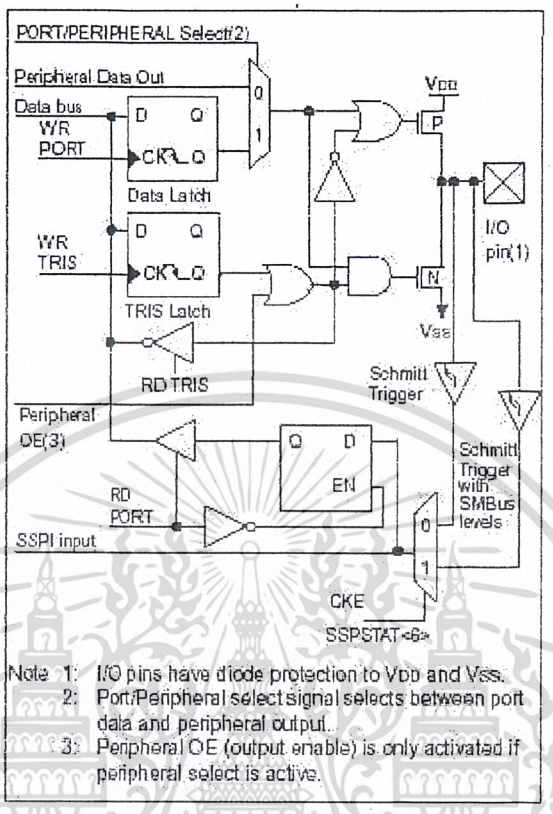
3.2.1.3 พอร์ต C ใน PIC 16F87X

พอร์ต C เป็นลักษณะแบบ พอร์ต แบบสองทิศทาง ซึ่ง รีจีสเตอร์ ที่จะเป็นตัวกำหนดว่า พอร์ต ใดจะเป็นแบบ อินพุท/เอาท์พุท จะถูกกำหนดโดย TRISC รีจีสเตอร์ ถ้า เซท TRISC บิต ใด (=1) พอร์ต B ที่บิตนั้นก็จะเป็น อินพุท ถ้า เคลียร์ TRISC บิต ใด (=0) พอร์ต C ที่บิตนั้นก็จะเป็น เอาท์พุท ที่ พอร์ต C จะมีคุณสมบัติเพิ่มเติม เช่น IIC, UART, SPI, PWM, CAPTURE ขึ้นอยู่กับการเลือกใช้งาน โดยเมื่อเราทำการ อีนาเบิล คุณสมบัติเพิ่มเติมต่างๆ ที่ พอร์ต C ต้องระวังในเรื่องของการตั้งค่า TRISC ของแต่ละขาของ พอร์ต C เพราะในการ อีนาเบิล คุณสมบัติบางตัวที่อยู่ที่พอร์ต C (เช่น UART) ตัวมันเองก็จะทำการเปลี่ยน บิต TRISC โดยอัตโนมัติ ดังนั้นไม่ควรที่จะตั้งค่า TRISC โดยตรงกับขาใดของ พอร์ต C ที่ทำการ อีนาเบิล คุณสมบัติเพิ่มเติม



รูปที่ 3.7 บล๊อกไดอะแกรมของขา RC<0:2>และ RC<5:7>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 บล๊อกไดอะแกรมของขา RC<3:4>

3.2.2 โครงสร้าง หน่วยความจำ(MEMORY) ของ PIC16F87x

หน่วยความจำ ใน PIC16F87X มีอยู่ 4 ประเภทคือ

1.โปรแกรมหน่วยความจำ(Program memory)

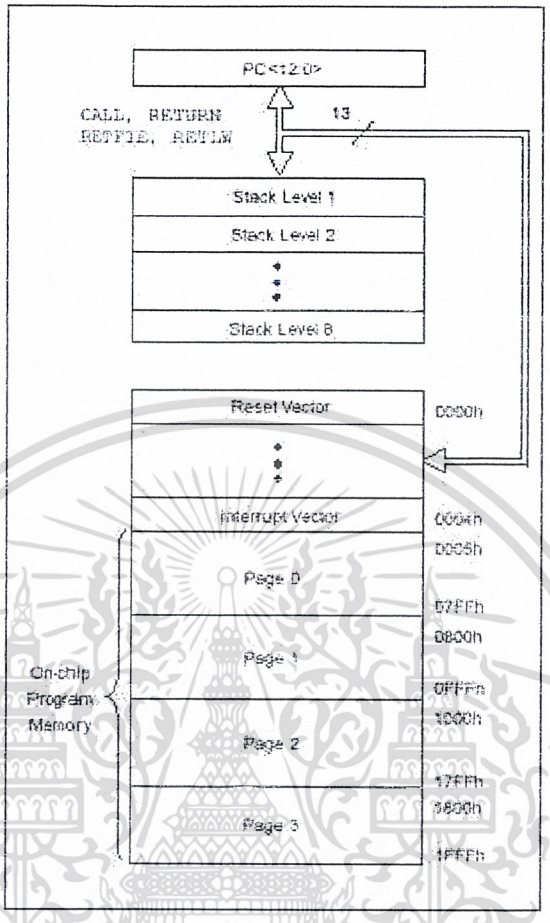
2.หน่วยความจำข้อมูล(Data memory)

3.หน่วยความจำ STACK

4.หน่วยความจำ EEPROM

โปรแกรมหน่วยความจำ PIC16F87x จะมีขนาดของโปรแกรมหน่วยความจำ ซึ่งสามารถอ้างได้ถึง 8กิโลไบต์ โดย PIC16F877/876 จะมีขนาดหน่วยความจำเท่ากับ 8K x 14 และ PIC16F873/874 มีขนาด 4K x 14 ซึ่ง ตำแหน่ง รีเซท-เวคเตอร์ จะอยู่ที่ 0000h และ อินเทอร์รัปพ์ เวกเตอร์ จะอยู่ที่ 0004h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ส่วนหน่วยความจำของ PIC 16F87X

รีจีสเตอร์ ที่เกี่ยวข้องกับ โปรแกรมหน่วยความจำของ PIC คือ PCL และ PCLATH ซึ่งก็คือ โปรแกรมนับ ต่ำ(LOW) และ สูง(HIGH) ไบต์ โดยที่ LOW ไบต์ จะมีขนาด 8 บิต ส่วน HIGH ไบต์ จะมีขนาด 5 บิต ซึ่งทำให้มีขนาดรวมกัน 13 บิต คือสามารถอ้างหน่วยความจำได้ 8 กิโลไบต์

PIC จะแบ่ง โปรแกรมหน่วยความจำ ออกเป็น ส่วน ซึ่งแต่ละ ส่วน ก็จะมีขนาด 2 กิโลไบต์ ซึ่งคำสั่ง CALL และ GOTO สามารถสั่งให้ โปรแกรมนับ กระโดดไปมาได้ในแต่ละส่วน เท่านั้นแต่ถ้าต้องการกระโดดจากส่วน หนึ่งไปยังอีก ส่วน หนึ่ง จะต้องไปควบคุม PCLATH<4:3> บิตที่อยู่ บิตแอดเดรส ที่ 12 และ 13 ให้ชี้ไปยัง ส่วน ที่ต้องการก่อน หลังจากนั้นจึงเรียกคำสั่ง CALL หรือ GOTO ตามอีกที

ส่วนที่ 0(Page 0) PCLATH<4:3> = 00

ส่วนที่ 1(Page1) PCLATH<4:3> = 01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

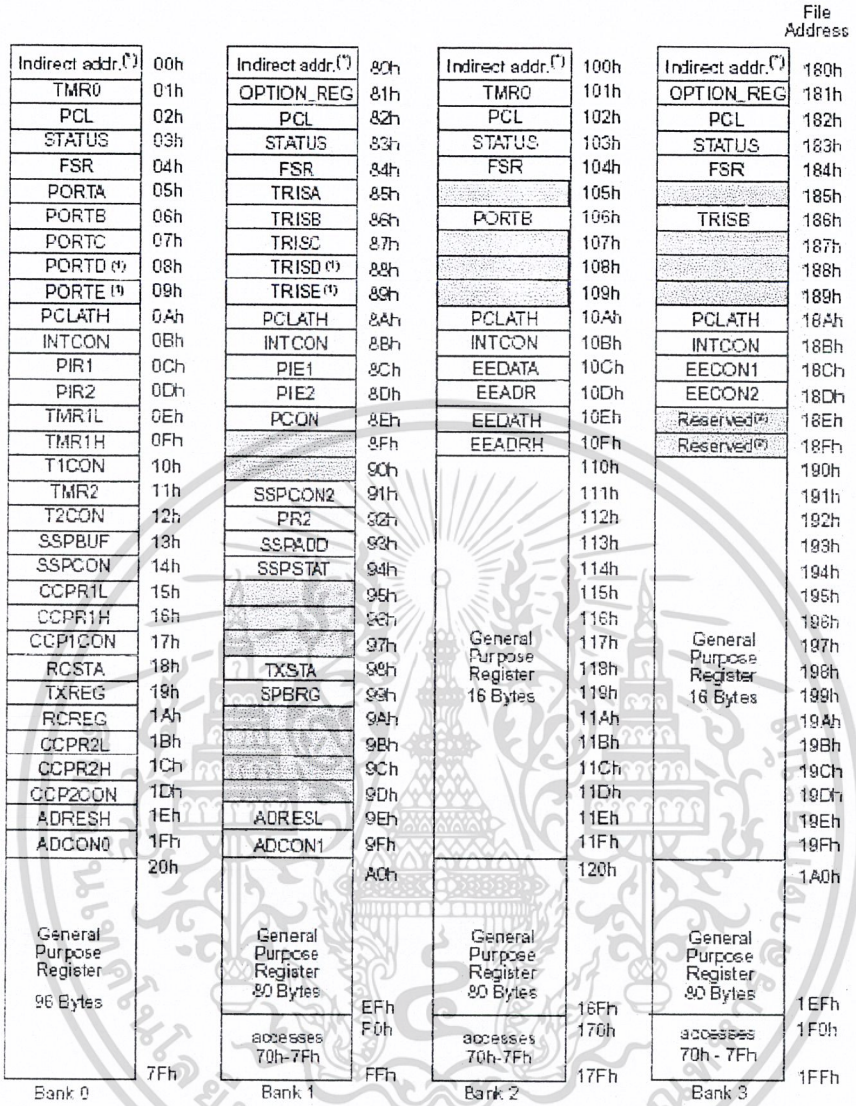
ส่วนที่ 2(Page2) PCLATH<4:3> = 10

ส่วนที่ 3(Page3) PCLATH<4:3> = 11

เมื่อใช้คำสั่ง CALL ไปที่ ช่วงการทำงาน(routine) ใด จะใช้คำสั่ง รีเทิร์น ในการกลับไป การ รีเทิร์นกลับนั้นไม่ต้องสั่ง PCLATH ให้ชี้ไปยัง ส่วน ก่อนหน้าที่จะเรียก CALL เพราะค่า ที่อยู่(address) ดังกล่าวจะถูกเก็บไว้ใน STACK เรียบร้อยแล้ว แต่สำหรับคำสั่ง GOTO เวลาข้าม ส่วน จะต้องสั่งให้ PCLATH ชี้ไปยัง ส่วน ที่จะไปทุกครั้ง การสร้าง มาโคร(marco) ไว้ที่หัวโปรแกรม เพื่อเอาไว้เรียกใช้ในการข้าม ส่วน ให้สะดวกขึ้น

หน่วยความจำข้อมูล ของ pic16F87X จะมีอยู่ 2 ส่วนคือ รีจีสเตอร์หน่วยความจำพิเศษ(Special register memory) และ รีจีสเตอร์ทั่วไป(General purpose register Special register) คือ รีจีสเตอร์ ที่ถูกใช้ โดย MCU และ Peripheral Modules เพื่อใช้ในการควบคุมการทำงานของ MCU โดย รีจีสเตอร์ จะมีชื่อเฉพาะอยู่แล้ว โดยสามารถเข้าถึงได้เหมือน แรม ปกติ General purpose register คือ รีจีสเตอร์ ที่สามารถนำไปใช้งานทั่วๆไปได้ ขึ้นอยู่กับผู้ใช้งาน รูปข้างล่างเป็นพื้นที่ของหน่วยความจำ ของ PIC16F87X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงพื้นที่ของหน่วยความจำ ของ PIC16F87X

สำหรับ PIC16F876 และ PIC16F877 จะมี รีจีสเตอร์ทั่วไป(general purpose register) อยู่ที่ address

20h – 7Fh

A0h – EFh ถ้าอ้าง ที่อยู่(address F0h – FFh) pic จะมองว่าเป็น 70h-7Fh

110h – 11Fh, 120h – 16Fh ถ้าอ้าง ที่อยู่(address 170h – 17Fh) pic จะมองว่าเป็น 70h-7Fh

190h – 19Fh, 1A0h – 1EFh ถ้าอ้าง ที่อยู่(address 1F0h – 1FFh) pic จะมองว่าเป็น 70h-7Fh

ส่วน PIC16F873 – PIC16F874 จะมี รีจีสเตอร์ทั่วไป(general purpose register) น้อยกว่า PIC16F876-PIC16F877 ซึ่งจะมีที่ตำแหน่ง หน่วยความจำ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

20h – 7Fh

A0h – FFh

ถ้าอ้าง ที่อยู่(address 120h – 17Fh) PIC จะมองว่าเป็น 20h-7Fh

ทำการ อ้างถึงโดยตรง(direct access) โดย รีจีสเตอร์ ที่เกี่ยวข้องถ้าอ้าง ที่อยู่(address 1A0h – 1FFh) PIC จะมองว่าเป็น A0h-FFh

ในการอ้าง หน่วยความจำ ของ รีจีสเตอร์ทั่วไป(general purpose register) ทำได้ 2 วิธีคือ แบบ อ้างถึงโดยตรง(direct access) กับ อ้างถึงทางอ้อม(indirect access)

อ้างถึงโดยตรง(direct acces ก็คือการอ้าง ที่อยู่(address) นั้นโดยตรงนั่นเองเช่น

อ่าน(Read)

movfw 0x21 ; อ่านข้อมูลที่อยู่ (address 0x20) ไปใส่ไว้ที่ w รีจีสเตอร์

เขียน(Write)

movwf 0x40 ; เขียนข้อมูลที่อยู่ใน w รีจีสเตอร์ ไปไว้ใน address 0x40

จะเห็นว่า หน่วยความจำข้อมูลของ PIC 16F87X จะ แบ่งเป็น 4 ช่อง(bank) ในการที่จะอ้าง หน่วยความจำ แบบ อ้างถึงโดยตรง (direct access) ที่ ช่อง(bank) ไหนจะต้องสั่งให้ mcu ชี้ไปที่ ช่อง(bank) นั้นเสียก่อน แล้วค่อยก็คือ STATUS รีจีสเตอร์ บิต 6 กับ 5 ซึ่งเรียกว่า RP1,RP0

RP1,RP0 > 00 เลือกช่อง(Bank0)

RP1,RP0 > 01 เลือกช่อง(Bank1)

RP1,RP0 > 10 เลือกช่อง(Bank2)

RP1,RP0 > 11 เลือกช่อง(Bank3)

เช่น

```
BSF STATUS,RP0 ; เลือก ไปยัง ช่อง(Bank1)

BCF STATUS,RP1 ;
```

access คือการอ้าง หน่วยความจำทั่วไป ของ ในตำแหน่งนั้นๆ โดยผ่าน รีจีสเตอร์ 2 ตัว คือ FSR และ INDF ซึ่ง FSR จะทำหน้าที่เหมือน ตัวชี้ที่อยู่(pointer address) ส่วน ข้อมูล(data) ในการอ่านและเขียน จะกระทำผ่านทาง INDF ตัวอย่างเป็นดังนี้

movlw 0x20 ; ใส่ค่า 20h ไปยัง w

movwf FSR ; ให้ ตัวชี้(pointer) ชี้ไปยังที่อยู่ (address 20h)

clrf INDF ; ทำการเคลียร์ ค่า ณ ตำแหน่งทางอ้อมใน หน่วยความจำที่ FSR ชี้อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการใช้ การอ้างถึงทางอ้อม(indirect access) สามารถอ้าง หน่วยความจำ ได้ทีละ 2 ช่อง(bank) คือ 0 กับ 1 หรือ 2 กับ 3 โดย รีจีสเตอร์ บิตที่ใช้ในการควบคุมคือ STATUS รีจีสเตอร์ บิต 7 (IRP) โดยที่ ถ้าบิตนี้เป็น

1 = Bank 2, 3 (100h – 1FFh)

0 = Bank 0, 1 (00h – FFh)

ดังนั้นต้องคอยระวังเวลาที่จะทำการอ้าง หน่วยความจำ ของ PIC ไม่ว่าจะเป็น ทางตรง(direct) หรือ ทางอ้อม(indirect) ก็ตาม เพราะมีโอกาสผิดพลาดเกิดขึ้นได้

3.2.3 การเปลี่ยนอนาลอกเป็นดิจิตอล (ANALOG-TO-DIGITAL CONVERTER(A/D))

สำหรับ A/D ใน ชิป แบบ 28 พิน จะมี 5 อินพุท แบบ 40 พิน จะมี 8 อินพุท รีจีสเตอร์ ที่เกี่ยวข้อง จะมีอยู่ 4 ตัวด้วยกันคือ ADRESH คือ ค่า รีจีสเตอร์ ผลลัพธ์ ไบต์ สูง ของการแปลงสัญญาณADRESL คือ ค่า รีจีสเตอร์ ผลลัพธ์ ไบต์ ต่ำ ของการแปลงสัญญาณADCON0 คือ รีจีสเตอร์ ควบคุมเกี่ยวกับ A/D ไบต์ ที่ 1ADCON1 คือ รีจีสเตอร์ ควบคุมเกี่ยวกับ A/D ไบต์ ที่ 2

อธิบาย รีจีสเตอร์ แต่ละตัวเพื่อความเข้า

ADCON0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/(DONE\)	-	ADON

ตารางที่ 3.1 หน่วยความจำควบคุมเกี่ยวกับ A/D

Bit7-6: ADCS1:ADCS0: เป็น บิต ที่ใช้เลือกสัญญาณนาฬิกาในการแปลง A/D

00 = Fosc/2 Fosc = ความถี่ของ สัญญาณนาฬิกา(crystal) ที่ใช้

01 = Fosc/8

00 = Fosc/32

11 = FRC (เลือกใช้ความถี่จากวงจร RC ที่อยู่ภายใน)

Bit5-3: CHS2:CHS0 เป็น บิต ที่ใช้เลือก ช่องสัญญาณ (channel) ของสัญญาณ A/D

000 = ช่องสัญญาณ 0, (RA0/AN0)

001 = ช่องสัญญาณ 1, (RA0/AN1)

000 = ช่องสัญญาณ 2, (RA0/AN2)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ไม่อนุญาตให้ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000 = ช่องสัญญาณ 3, (RA0/AN3)

000 = ช่องสัญญาณ 4, (RA0/AN4)

000 = ช่องสัญญาณ 5, (RA0/AN5) (ไม่มีใน mcu แบบ 28 pins)

000 = ช่องสัญญาณ 6, (RA0/AN6)

000 = ช่องสัญญาณ 7, (RA0/AN7)

บิต 2: GO/(DONE\): เป็นบิตที่ใช้ในการแสดงสถานะของการแปลง A/D

ถ้า ADON บิต ถูกเซท เป็น 1 แล้ว เมื่อบิตนี้เป็น

1 หมายถึง A/D กำลังอยู่ในช่วงการแปลงค่า (ให้ เซท บิต นี้ในการเริ่มต้นการแปลงสัญญาณ)

0 หมายถึง A/D ไม่ได้อยู่ในช่วงการแปลงค่า (บิตนี้จะ mcu จะ เคลียร์ เป็น 0 โดยอัตโนมัติ เมื่อทำการแปลงสัญญาณเสร็จ

บิต 1: ยังไม่ถูกใช้งาน

บิต 0: ADON: A/D On บิต (บอกสถานะของ A/D ในขณะนั้น)

1 = A/D convertor กำลังถูกใช้งาน

0 = A/D convertor ไม่ได้ถูกใช้งาน

ADCON1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0

ตารางที่3.2 แสดงบิต 7: ADFM:

1 = หลังจากแปลงสัญญาณ ให้ด้านซ้ายของ ADRESH เป็น 0

ADRESH	ADRESL
000000xx	xxxxxxxx

ตารางที่ 3.3 แสดงการแปลงสัญญาณให้ด้านซ้ายของ ADRESH

0 = หลังจากแปลงสัญญาณให้ด้านขวาของ ADRESL เป็น 0

ADRESH	ADRESL
xxxxxxxx	xx000000

ตารางที่ 3.4 แสดงการแปลงสัญญาณให้ด้านขวาของ ADRESH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 6-4: ไม่ได้ถูกใช้

บิต 3-0: PCFG3:PCFG0 เป็นตัว เซท คุณสมบัติต่างๆ ในการทำงาน A/D ให้กับ PIC โดยสามารถเลือกว่าจะใช้ VREF แยกต่างหากหรือจะใช้จาก VDD,VSS ก็ได้

PCFG3:PCFG0	AN7 RE2	AN6 RE1	AN5 RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ REFS
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2

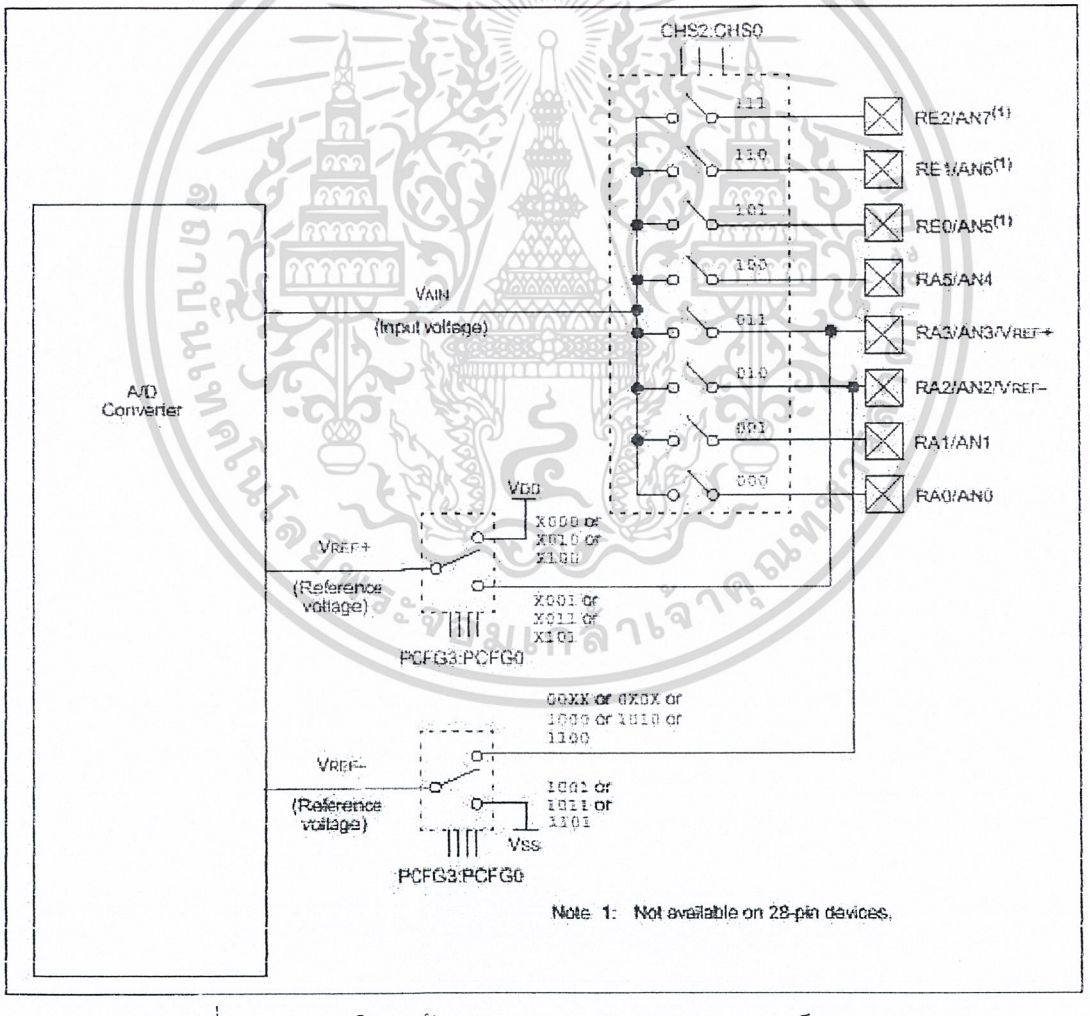
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

A หมายถึง อนาลอกอินพุท D หมายถึงดิจิตอล I/O ส่วน An7-AN5 จะไม่มีในตระกูลที่เป็น 28 ขา

ตารางที่ 3.5 แสดงส่วนการใช้ ช่องสัญญาณ กับ พอร์ต

เมื่อการแปลง A/D เสร็จสิ้น ผลลัพธ์ของการแปลง A/D จะมีขนาด 10 บิต ซึ่งจะเก็บอยู่ใน รีจีสเตอร์ 2 ตัวต่อกันคือ ADRESH:ADRESL ส่วน รีจีสเตอร์ บิต GO/DONE\ (ADCON0<2>) จะถูก เคลียร์ และ ADIF จะถูก เซท (A/D interrupt flag) บล๊อกไดอะแกรม ของ A/D



รูปที่ 3.11 แสดงโครงสร้าง A/D CONVERTOR ของ พอร์ต A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการใช้งาน A/D module

-เลือก เซทรูปแบบ ของ A/D โดย

-เลือก อนาลอกพิน/ แรงดันอ้างอิง ด้วย ADCON1

-เลือก A/D อินพุท ช่องสัญญาณ ด้วย ADCON0

-เลือก A/D conversion clock (ความถี่ของสัญญาณนาฬิกาที่จะใช้ใน A/D) จาก ADCON0

สั่งให้ A/D module ทำงาน ด้วย ADCON0

ถ้าต้องการใช้ A/D อินเทอร์รัปพ์ ต้อง เซท flag ต่างๆ ดังนี้

รอเวลาเพื่อให้ A/D module พร้อม (acquisition time) หาได้จากการคำนวณ

เริ่มทำการ เคลียร์ ADIF บิต

-เซท ADIE บิต

-เซท GIE บิต

A/D ด้วยการ เซท GO/DONE\ บิต

รอจนกว่าการแปลงA/D จะเสร็จสมบูรณ์

เขียนโปรแกรมวนลูป์ รอดู GO/DONE\ บิต จนกว่าจะ เคลียร์

รอ A/D อินเทอร์รัปพ์

อ่านค่าผลลัพธ์ของ A/D จาก ADRESH:ADRESL โดย เคลียร์ ADIF บิต หลังจาก การอ่านด้วยถ้าใช้ A/D อินเทอร์รัปพ์

หลังจากอ่านค่าเรียบร้อยแล้ว ต้องรอเป็นระยะเวลา 2TAD ก่อนที่จะทำการแปลง A/D อีกครั้งหนึ่ง

การเลือกสัญญาณนาฬิกา(A/D Conversion Clock)

เวลาที่ใช้ในการ แปลง A/D หนึ่งครั้งเรากำหนดให้เป็น TAD ในหนึ่งครั้งของการแปลง A/D PIC ต้องการอย่างน้อย 12TAD ต่อการแปลงแบบ 10 บิต ในการเลือกสัญญาณความถี่ของ A/D สามารถเลือกได้ 4 แบบคือ

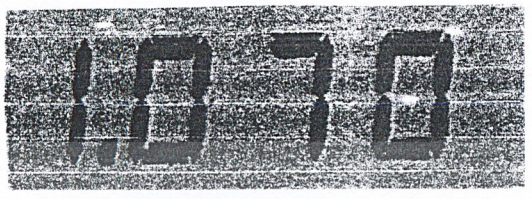
- 2Tosc
- 8Tosc
- 32Tosc
- Internal RC oscillator

ซึ่งค่าเหล่านี้เราเลือกได้จากการ เซท ค่า รีจิสเตอร์ (ADCS1:ADCS0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ตัวแสดงผลแบบผลึกเหลว Liquid crystal display(LCD)

LCD เป็นตัวเลขแสดงผลซึ่งใช้ผลึกเหลว ปัจจุบันนิยมใช้กันมาก เพราะกินไฟน้อยมากและสามารถสร้างรูปแบบของตัวเลขหรือตัวอักษรที่จะแสดงได้หลายรูปแบบ ดังรูป

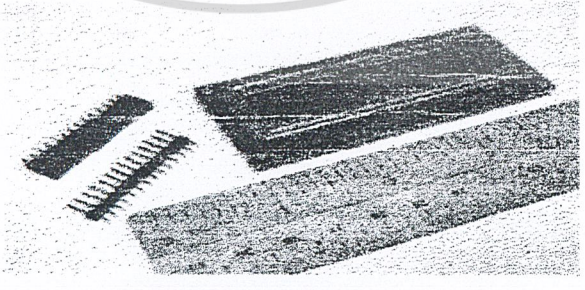


รูปที่ 3.12 จอแสดงผลแบบผลึกเหลว (LCD)

เป็นรูปแบบตัวแสดงผล LCD ปัจจุบันนาฬิกาข้อมูลดิจิตอล เครื่องดิจิตอล เครื่องคิดเลข ก็ใช้ LCD เป็นตัวแสดงผลทั้งหมดเมื่อยใช้ร่วมกับ ไอซี CMOS ใช้ถ่านปรอทขนาดเล็กก็สามารถทำงานได้นานได้นานถึง 1-2 ปี

โมดูล LCD มีด้วยกันหลายแบบแต่เห็นทั่วไปมี 3 แบบ คือ

1. แบบตัวเลข 7 ส่วน มักพบในเครื่องคิดเลขและนาฬิการะบบดิจิตอลขนาดเล็ก
2. แบบดอตเมตริกซ์ (dot – matrix) สามารถแสดงผลการทำงานให้ผู้ใช้งานทราบ เช่นเครื่องเลเซอร์พริ้นเตอร์ เครื่องรูดบัตรเครดิต จอแสดงผลของเครื่องเล่นวีดีโอ เครื่องเล่นเลเซอร์ดิสก์ เครื่องเล่นวีดีโอ และเครื่องมือแพทย์ เป็นต้น
3. แบบกราฟฟิก (Graphic) แบบนี้มีโครงสร้างเหมือนกับดอตเมตริกซ์แต่มีขนาดใหญ่กว่าสามารถแสดงอักษรและรูปภาพได้ มักพบในเครื่องเล่นเกมส์ จอภาพสำหรับโฆษณาประชาสัมพันธ์และในปัจจุบันมีแบบสีด้วยจึงสามารถนำไปใช้เป็นจอภาพสำหรับเครื่องรับโทรทัศน์ขนาดเล็ก เป็นต้น



รูปที่ 3.13 ขนาดของตัวแสดงผลแบบผลึกเหลว (LCD)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 **ขาใช้งานของโมดูล** LCD มีชื่อและหน้าที่ดังนี้

ขา กราวน์(GND) ขา 1 ต่อกับกราวด์ของวงจร

ขา $+V_{DD}$ (ขา 2) ต่อไฟเลี้ยง +5 V

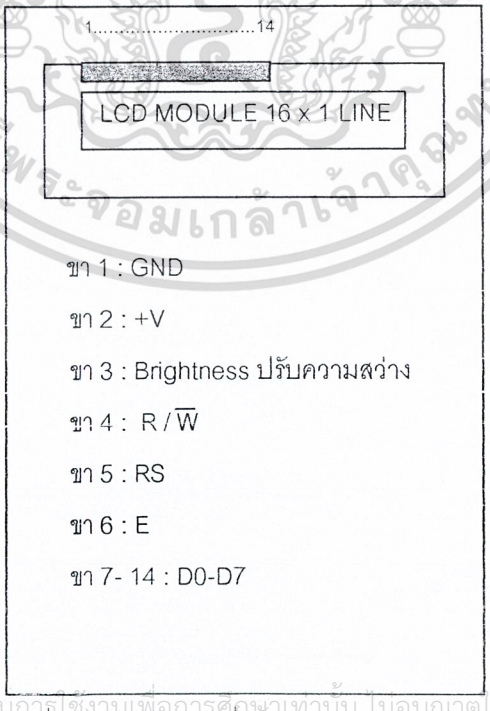
ขา V_O (ขา 3) เป็นขาสำหรับป้อนแรงดันเพื่อปรับความสว่างของหน้าจอแสดงผล

ขา RS (Register Select : ขา 4) เป็นขาสำหรับเลือกว่าต้องการติดต่อกับรีจิสเตอร์คำสั่ง (Data Register : DR)โดยถ้ากำหนดเป็น "0" จะเป็นการติดต่อกับรีจิสเตอร์คำสั่งและถ้าเป็น"1"จะเป็นการติดต่อกับรีจิสเตอร์ข้อมูลเพื่อนำข้อมูลไปแสดงผลบนหน้าจอ

ขา R/W (Read/Write control : ขา 5) เป็นขาเลือกการอ่านหรือเขียนข้อมูลกับโมดูล LCD ปกติขานี้จะต่อลงกราวด์เพื่อกำหนดให้เขียนข้อมูลไปยังโมดูล LCD เพียงทางเดียว เพื่อลดจำนวนขาที่ใช้ในการติดต่อ

ขา E (Enable : ขา 6) เป็นขาสำหรับการป้อนสัญญาณพัลส์เอ็นเอเบิลให้โมดูล LCD ทำงาน

ขา DO-D7 (Data : ขา 7- 14) เป็นขาข้อมูล 8 บิต โดยขา 7 คือ DO ไล่ไปตามลำดับจนถึงขา 14 คือ D7



รูปที่ 3.14 แสดหน้าที่การทำงานของขา LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 ความรู้เบื้องต้นเกี่ยวกับโมดูลแสดงผล LCD แบบอักขระ

มีส่วนประกอบหลักๆ 3 ส่วนดังนี้

1. ตัวแสดงผล (display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอ LCD

2. ตัวควบคุม (controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของโมดูล LCD เช่น ลบจอภาพ แสดงตัวอักษร หรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิปควบคุมโดยเฉพาะ ชิปที่นิยมใช้คือ เบอร์ HD44780 และ HD61830 โดย HD44780 จะใช้ควบคุมแบบอักขระ ส่วน HD61830ใช้ควบคุม LCD แบบกราฟิก

3. ตัวขับ (driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลที่กำหนด ชิปที่ใช้ทำหน้าที่เป็นตัวขับนี้ได้แก่ HD44100H และ MSM5259 เป็นต้น

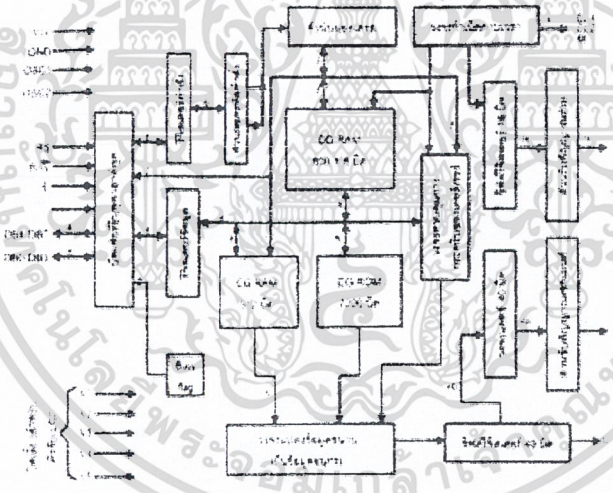
3.3.3 โครงสร้างภายในของตัวควบคุมโมดูล แบบอักขระ

ในการใช้งานโมดูล LCD จำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้างและคำสั่งที่ใช้ในการควบคุมให้ดีเสียก่อน โมดูล LCD ที่หาง่าย ราคาไม่แพงที่มีจำหน่ายในประเทศไทยและทั่วโลกนั้น นิยมใช้ชิปควบคุมของฮิตาชิเบอร์ HD44870 หรืออาจเป็นเบอร์อื่นแต่มีชุดคำสั่งและการทำงานที่เหมือนกันหรือที่เรียกว่า คอมแพตติเบิล (compattible) ดังนั้นจึงควรทำความรู้จักกับชิปควบคุมเบอร์นี้เสียก่อน ในรูป 3.15 เป็นบล็อกไดอะแกรมภายในของชิปควบคุม LCD เบอร์ HD44780 ประกอบด้วย

- บัฟเฟอร์อินพุทเอาท์พุท เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอกเพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม
- รีจิสเตอร์คำสั่ง (Instruction Register : IR)เป็นรีจิสเตอร์ใช้รับข้อมูลคำสั่งจากอุปกรณ์ภายนอกเพื่อนำไปควบคุมการแสดงผล
- รีจิสเตอร์ข้อมูล (Data Register : DR) เป็นรีจิสเตอร์ใช้รับข้อมูลจากอุปกรณ์ภายนอกเพื่อถ่ายทอดไปยังแรมเก็บข้อมูลแสดงผล หรือนำข้อมูลนั้นไปเพื่อสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร
- แรมเก็บข้อมูลแสดงผล (Display Data RAM : DDRAM)เป็นหน่วยความจำแรมทำหน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตาราง (Look up-table)ของตัวอักษรที่เก็บไว้ในหน่วยความจำรอมและแรมเก็บตัวอักษรเพื่อนำไปแสดงผลที่ตัวแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รอมเก็บตัวอักษร (Character Generator ROM : CGROM) เป็นหน่วยความจำรอมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านไปแสดงที่ตัวแสดงผลได้ มีขนาด 7,200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลใน DDRAM
- แรมเก็บตัวอักษร (Character Genertor RAM : CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอ มีขนาด 512 บิต การเขียนและอ่านค่าไปใช้นั้นทำได้เช่นเดียวกับ CGROM คือเขียนข้อมูลลงใน DDRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGRAMเอง
- แฟลก BUSY เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่าตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้นก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของแฟลก BUSY นี้เสียก่อน



รูป 3.15 ไดอะแกรมการทำงานของชิปควบคุมโมดูล LCD แบบอักขระเบอร์ HD44870

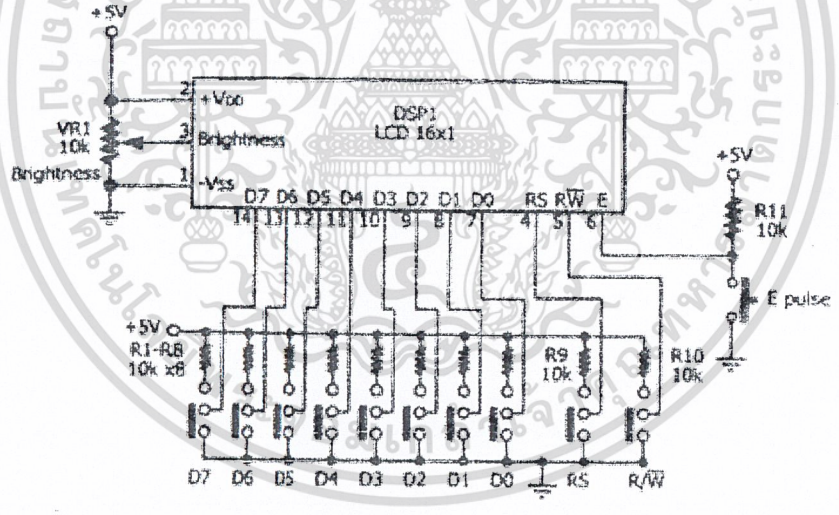
3.3.4 การติดต่อกับโมดูล LCD แบบอักขระ

มี 2 แบบ คือ 8 บิต และ 4 บิต โดยปกติใช้แบบแรก แต่ถ้ามีข้อจำกัดเรื่องจำนวนสายสัญญาณ ควรเลือกติดต่อแบบ 4 บิต ซึ่งมีขั้นตอนเพิ่มเล็กน้อย แต่ใช้สายสัญญาณเพียง 6-7 เส้น ในขณะที่แบบ 8 บิตใช้ 10-11 เส้น

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อแบบ 8 บิต

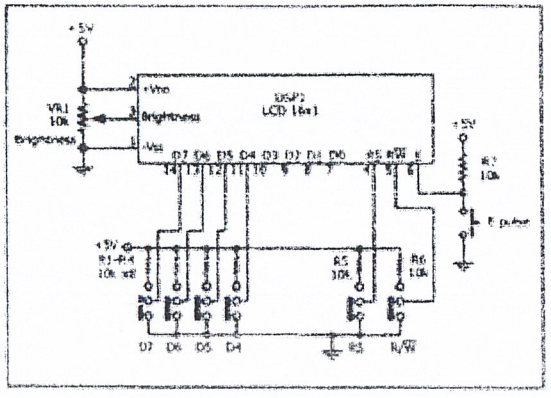
ในรูปที่ 3.16 แสดงการต่อวงจรเพื่อใช้งานโมดูล LCD อย่างง่ายที่สุด โดยขา D0-D7 ต่อกับแหล่งกำเนิดข้อมูลดิจิตอลขนาด 8 บิต ซึ่งอาจเป็นลอจิกสวิตช์หรือขาพอร์ตของไมโครคอนโทรลเลอร์ ต่อขา RS และ R / Wเข้ากับสวิตช์เพื่อเลือกป้อนลอจิก “0” และ “1” โดยขา R / W หากไม่มีความต้องการอ่านข้อมูลจากโมดูล LCD สามารถลดจำนวนสายสัญญาณลงได้โดยต่อขา R / W ลงกราวด์ ส่วนขา E ให้ต่อเข้ากับสวิตช์กดติดปล่อยดับที่มีการต่อตัวต้านทานพูลอัปเอาไว้เพื่อสร้างสัญญาณพัลส์อย่างง่าย โดยเมื่อยังไม่กดสวิตช์จะเกิดระดับลอจิก “1” และเมื่อกดสวิตช์แล้วปล่อย จะทำให้เกิดลอจิก”0” ขึ้นชั่วขณะ สัญญาณที่เกิดขึ้นมีลักษณะเป็นเหมือนพัลส์ลบเหตุที่ต้องสร้างสัญญาณเช่นนั้นเนื่องจากตัวควบคุมที่อยู่ภายในโมดูล LCD ต้องการสัญญาณพัลส์ลบในการกระตุ้นให้ทำงาน และที่ขา V_0 ต่อกับตัวต้านทานปรับค่าได้ 10kΩ เพื่อปรับความสว่างของจอแสดงผล



รูปที่ 3.16 ตัวอย่างวงจรเพื่อใช้โมดูล LCD แบบ 8 บิต

ในรูปที่ 3.17 แสดงการต่อวงจรเพื่อใช้งานโมดูล LCD ในโหมด 4 บิต สายสัญญาณข้อมูลใช้เพียง 4 เส้น คือ D4-D7 ส่วนสายสัญญาณควบคุมยังคงใช้ 3 เส้นคือ RS,R/W และ E หากต้องการลดสายสัญญาณควบคุมสามารถทำได้โดยต่อขา R/W ลงกราวน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 ตัวอย่างวงจรเพื่อใช้โมดูล LCD แบบ 4 บิต

3.3.5 ขั้นตอนการป้อนข้อมูลเพื่อใช้งานโมดูล LCD

1. เมื่อจ่ายไฟให้แก่โมดูล LCD แล้วต้องรอสักครู่หนึ่ง เพื่อให้โมดูล LCD ทำการเตรียมความพร้อมเพื่อเริ่มทำงานหรือเรียกว่า อินิเชีย (initial) ซึ่งใช้เวลาประมาณ 5 มิลลิวินาที ดังนั้นในทางปฏิบัติเมื่อเริ่มจ่ายไฟก็สามารถเริ่มป้อนข้อมูลได้เกือบทันที เพราะเวลา5 มิลลิวินาทีเร็วมาก

2. กำหนดสถานะลอจิกของขา R / W ว่าต้องการอ่านหรือเขียนข้อมูลกับโมดูล LCD ซึ่งปกติแล้วมักจะเป็นการเขียนมากกว่า จึงกำหนดให้ขา R / W นี้เป็นลอจิก "0"

3. กำหนดสถานะลอจิกของขา RS เพื่อเลือกว่า ต้องการติดต่อกับรีจิสเตอร์ข้อมูลหรือรีจิสเตอร์คำสั่ง

4. ป้อนข้อมูลให้แก่สายสัญญาณข้อมูลทั้ง 8 เส้น (D0-D7) ในกรณีติดต่อแบบ 8 บิต และ 4 บิต เส้น (D4-D7) ในกรณีติดต่อแบบ 4 บิต

5. ป้อนสัญญาณพัลส์เอ็นเอเบิล (E pulse) ให้แก่โมดูล LCD หลังจากที่โมดูล LCD ได้รับสัญญาณพัลส์เอ็นเอเบิล (E) โมดูล LCD จะทำงานตามคำสั่งและข้อมูลที่กำหนดทันที

กรณีเขียนคำสั่งไปยังโมดูล LCD

1. ทำให้ขา R / W เป็นลอจิก "0" โดยต่อลงกราวด์

2. ป้อนลอจิก "0" เข้าที่ขา RS และคงสถานะลอจิกนี้ไว้จนกว่าจะป้อนข้อมูลคำสั่งเสร็จสิ้นหรือต้องการเปลี่ยนไปป้อนข้อมูลแสดงผล

3. ป้อนข้อมูลคำสั่งเข้าทางขา D0-D7 ในกรณีติดต่อแบบ 8 บิต และ D4-D7 ในกรณีติดต่อแบบ 4 บิต

4. ป้อนพัลส์เอ็นเอเบิล 1 พัลส์เข้าที่ขา E ของโมดูล LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.หากยังคงต้องการเขียนคำสั่งไปยังโมดูล LCD สามารถดำเนินการต่อไปเลย โดยทุกครั้งที่ป้อนข้อมูลคำสั่งเสร็จ ต้องป้อนพัลส์เอ็นเอเบิล 1 พัลส์ให้แก่โมดูล LCD ที่ขา E เสมอ

กรณีเขียนข้อมูลแสดงผลไปยังโมดูล LCD

1. ทำให้ขา R / W เป็นลอจิก "0" โดยต่อลงกราวด์

2. ป้อนลอจิก "1" เข้าที่ขา RS และคงสถานะลอจิกนี้ไว้จนกว่าจะป้อนข้อมูลแสดงผลเสร็จสิ้นหรือต้องการเปลี่ยนไปป้อนคำสั่ง

3. ป้อนข้อมูลคำสั่งเข้าทางขา D0-D7 ในกรณีติดต่อแบบ 8 บิต และ D4-D7 ในกรณีติดต่อแบบ 4 บิต

4. ป้อนพัลส์เอ็นเอเบิล 1 พัลส์เข้าที่ขา E ของโมดูล LCD

5. หากยังคงต้องการเขียนข้อมูลแสดงผลไปยังโมดูล LCD สามารถดำเนินการต่อไปได้เลย โดยทุกครั้งที่ป้อนข้อมูลคำสั่งเสร็จ ต้องป้อนพัลส์เอ็นเอเบิล 1 พัลส์ให้แก่โมดูล LCD ที่ขา E เสมอ อย่างไรก็ตาม สามารถเปลี่ยนขนาดของข้อมูลที่ป้อนให้แก่โมดูล LCD ได้ตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ภาษาที่ใช้สำหรับเขียนโปรแกรม PIC

สำหรับ ไมโครคอนโทรลเลอร์ตระกูล PIC มีเครื่องมือทาง ซอฟแวร์ ที่ชื่อ MPLAB โดย Microchip Technology จากสหรัฐอเมริกาเป็นผู้พัฒนาขึ้น

MPLAB ได้รวบรวมเอาซอฟแวร์ที่สำคัญ ๆ ไม่ว่าจะเป็น เท็กซ์เอดิเตอร์ (text editor) สำหรับการเขียนโปรแกรม ตัวแปรภาษาแอสเซมบลีหรือที่เรียกว่าแอสเซมเบลอร์ (assembler) โปรแกรมจำลองการทำงานหรือซิมูเลเตอร์ (simulator) รวมทั้งโปรแกรมตรวจสอบการทำงานทางฮาร์ดแวร์อย่างดีบักเกอร์ (debugger) และอีมูเลเตอร์ (emulator) ซึ่งสองอย่างหลังต้องอาศัยฮาร์ดแวร์พิเศษด้วย

PCW PIC C Compile เป็นเครื่องมือที่ช่วยสำหรับการเขียนโปรแกรมด้วยภาษา C ซึ่งจะเป็นตัว Compile จาก .C เป็น .HEX เพื่อใช้ใน PIC ต่อไปซึ่งใน PCW จะรวม PCW (สำหรับ 14 บิตออปโค้ด) and PCB (สำหรับ 12 บิตออปโค้ด) ไว้ ซึ่งเป็นทางเลือกอีกทางสำหรับผู้เขียนโปรแกรม

อนึ่งสามารถเขียนโปรแกมได้โดย ภาษาเบสิกได้อีกทางเลือกหนึ่งซึ่งมีตัว compiler ของตัวมันเอง เช่นเดียวกับ PCW PIC C Compile

ภาษา C กับไมโครคอนโทรลเลอร์

เหตุที่เลือกใช้ภาษา C กับไมโครคอนโทรลเลอร์

1. ความสะดวกและความง่ายของการใช้ เนื่องจากภาษา C เป็นภาษาที่ใช้คีย์เวิร์ดมาตรฐานเพียง 28 ตัว ทำให้เรียนรู้ได้ง่าย และภาษามีโครงสร้างดีทำให้การเขียนโปรแกรมค่อนข้างง่ายเมื่อเทียบกับภาษาแอสแซมบลี

2. สามารถทำงานแอพพลิเคชั่นที่ซับซ้อน ภาษา C มีประโยคที่ใช้ควบคุมทิศทาง การประมวลผลแบบลอจิก รวมทั้งไลบรารี่ฟังก์ชั่นทางคณิตศาสตร์ เช่น คำสั่ง คูณ หาร และฟังก์ชั่นทางตรีโกณมิติ เช่น ไซน์ โคซายน์ เป็นต้น

3. ไลบรารี่ฟังก์ชั่น เราสามารถเขียนโปรแกรมที่มีขนาดใหญ่ขึ้นได้ง่ายด้วยความสามารถในการสร้างไลบรารี่ฟังก์ชั่นของภาษา C เช่น สร้างฟังก์ชั่นที่หลายแอพพลิเคชั่นสามารถใช้ร่วมกันได้และคอมไพล์ออกเป็นออปเจ็กไฟล์ เวลาที่ต้องการนำมาคอมไพล์หลาย ๆ ไฟล์ เราไม่ต้องนำซอร์สไฟล์ของไลบรารี่ฟังก์ชั่นเหล่านี้มาคอมไพล์ใหม่ คอมไพเลอร์มีโปรแกรมช่วยในการรวมเอาออปเจ็กไฟล์มาคอมไพล์ร่วมกับโปรแกรมหลักได้ทันที ทำให้ลดเวลาในการเขียนโปรแกรมที่มีขนาดใหญ่เป็นอันมาก

4. ง่ายต่อการปรับปรุงและบำรุงรักษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 ชนิดของโปรแกรม

MPASM

เป็น แอสแซมเบลอร์ ที่ทางบริษัท MICROCHIP เป็นที่นิยมสำหรับการใช้งานที่เขียนด้วยภาษาแอสแซมบลี สามารถหา ดาวน์โหลด ได้ที่ เวปไซต์ ของทาง MICROCHIP โดยจะติดมากับ MPLAB-IDE

MPLAB C17 และ MPLAB C18

เป็น C คอมไพล์เลอร์ (C compiler) ของทางค่าย MICROCHIP เอง สนับสนุน เฉพาะเบอร์ PIC17XXX และ PIC18XXX

HITECH-PIC C compiler

เป็น C คอมไพล์เลอร์ (C compiler) ของ HITECH มีประสิทธิภาพการ คอมไพล์ ให้เป็นภาษา แอสแซมบลี สูง (คือแปลงออกมาเป็น แอสแซมบลี แล้ว โค้ด ไม่ใหญ่มาก)

CCS C compiler

เป็น คอมไพล์เลอร์ ที่ประสิทธิภาพมาก นิยมใช้

PBASIC Compiler

มีโครงสร้างเป็นภาษา BASIC เป็นภาษานิยม เพราะเขียนง่าย เข้าใจง่าย ตัวอย่างมาก แต่ขนาดไฟล์ที่ คอมไพล์ ออกมาใหญ่ เหมาะกับโปรแกรม ที่ไม่ใหญ่และไม่ซับซ้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

หลักการทำงานของเครื่องวัดวิเคราะห์สี

แบ่งออกได้เป็น 3 ส่วน

1. ส่วน ตัวต้านทานไวแสง(LDR) ซึ่งถือว่าเป็น อินพุท รับค่าความเข้มแสงก่อนเข้าตัว PIC
2. ส่วนประมวลผล เป็นส่วนคำนวณค่าจาก อินพุท ที่รับเข้ามาเพื่อส่งค่าออก LCD
3. ส่วน LCD ซึ่งถือว่าเป็น เอาท์พุท มีหน้าที่แสดงผลที่ได้ออกมาหน้าจอ

4.1 ส่วนที่ 1 (LDR)

จากสูตร $R = AL^{-0.85}$ เมื่อ R = ค่าความต้านทาน (ohms)

A = ค่าคงที่ในช่วง 340 - $991x10^3$

L = ความสว่าง (Lux)

จากสมการ $R = AL^{-0.85}$ จะเห็นได้ว่า เมื่อนำไป วาดกราฟ จะได้กราฟไม่เป็นเชิงเส้น ฉะนั้นจึงต้องมีการทดลองหาค่าคงที่ A ค่าใดค่าหนึ่ง เพื่อแทนในสูตรและใช้ในการคำนวณต่อไป

ตารางที่4.1 การทดลองหาค่า ความต้านทาน เมื่อวัดความสว่างโดย Lux meter ที่ความสว่างต่างกันและทำการหาค่า $\overline{A}$

หาค่าคงที่ของ LDR ตัวที่ 1

LUX	R_1 (KΩ)	A_1
100	7.27	$364.30x10^3$
200	4.49	$400.17x10^3$
300	3.41	$429.29x10^3$
400	2.68	$434.64x10^3$
550	2.12	$453.25x10^3$
700	1.73	$455.03x10^3$
950	1.38	$467.60x10^3$
1150	1.16	$461.80x10^3$

$\overline{A_1} = 433.26 \times 10^3$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่4.2 การทดลองหาค่า ความต้านทาน เมื่อวัดความสว่างโดย Lux meter ที่ความสว่างต่างกันและทำการหาค่า $\overline{A}$

หาค่าคงที่ของ LDR ตัวที่ 2

LUX	R_2 (KΩ)	A_2
100	8.94	448.06×10^3
200	5.12	456.32×10^3
300	3.87	487.20×10^3
400	3.03	491.40×10^3
550	2.37	506.69×10^3
700	1.92	505.01×10^3
950	1.54	521.82×10^3
1150	1.28	509.57×10^3

$\overline{A_2} = 490.75 \times 10^3$

ตารางที่4.3 การทดลองหาค่า ความต้านทาน เมื่อวัดความสว่างโดย Lux meter ที่ความสว่างต่างกันและทำการหาค่า $\overline{A}$

หาค่าคงที่ของ LDR ตัวที่ 3

LUX	R_3 (KΩ)	A_3
100	8.36	418.99×10^3
200	4.83	430.47×10^3
300	3.61	454.47×10^3
400	2.83	458.97×10^3
550	2.24	478.90×10^3
700	1.80	473.44×10^3
950	1.42	481.15×10^3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1150	1.20	477.72×10^3

$$\overline{A}_3 = 459.26 \times 10^3$$

ตารางที่4.4 การทดลองหาค่า ความต้านทาน เมื่อวัดความสว่างโดย Lux meter ที่ความสว่างต่างกันและทำการหาค่า $\overline{A}$

หาค่าคงที่ของ LDR ตัวที่ 4

LUX	R_4 ($K\Omega$)	A_4
100	9.57	479.63×10^3
200	5.81	517.81×10^3
300	3.98	501.05×10^3
400	3.21	520.60×10^3
550	2.49	532.25×10^3
700	1.99	523.42×10^3
950	1.59	538.76×10^3
1150	1.33	529.48×10^3

$$\overline{A}_4 = 517.88 \times 10^3$$

จะได้ $\overline{A}_{\text{ทฤษ}} = 475.28 \times 10^3$ ดังนั้น $R = 475.28 \times 10^3 L^{-0.85}$

4.2 ส่วนที่ 2 (ประมวลผล)

เป็นส่วนคำนวณหาค่า ความคลาดเคลื่อน ของวัตถุ

ขั้นตอนการหาค่าความคลาดเคลื่อน

ขั้นที่ 1 หาค่าอ้างอิง

LDR จะมี 4 ตัว วัดสีวัตถุอ้างอิง โดย

LDR ตัวที่1 (ติดแผ่นกรองสีดำ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LDR ตัวที่2 (ติดแผ่นกรองสีเหลือง)

LDR ตัวที่3 (ติดแผ่นกรองสีม่วง)

LDR ตัวที่4 (ติดแผ่นสีกรองสีเขียวแกมน้ำเงิน)

จะได้ค่า R1 , R2, R3, R4 เข้าสมการ $R = 475.28x10^{3}L^{-0.85}$ เพื่อหาค่า LST[0], LST[1] , LST[2] , LST[3]

หมายเหตุ LST[0] คือ ค่าความสว่างอ้างอิง(L- Reference) ของสีดำ

LST[1] คือ ค่าความสว่างอ้างอิง(L- Reference) ของสีเหลือง

LST[2] คือ ค่าความสว่างอ้างอิง(L- Reference) ของสีม่วง

LST[3] คือ ค่าความสว่างอ้างอิง(L- Reference) ของสีเขียวแกมน้ำเงิน

แต่เนื่องจาก สีทั้ง 3 สีรวมกับเท่ากับสีดำ ดังนั้นถ้าให้ แสงสีดำ เทียบเป็น 100% สี 3 สีแต่ละสี จะเทียบได้เท่ากับ 33.3% ฉะนั้นต้องหาค่าคงที่ของแต่ละสีออกมา ดังนี้

$$q = \frac{LST[1]x\ 100}{LST[0]}$$; q คือ เปอร์เซนต์ของสีเหลืองเทียบกับสีดำ

แต่เปอร์เซนต์ที่ได้ต้องเป็น 33.3 % ดังนั้น

rate [1] = 33.3/q ; rate[1] คือ ค่าคงที่ของสีเหลือง

เช่นกัน สีม่วง และสีเขียวแกมน้ำเงิน สามารถคำนวณได้

rate[2] ; rate[2] คือ ค่าคงที่ของสีม่วง

rate[3] ; rate[3] คือ ค่าคงที่ของสีเขียวแกมน้ำเงิน

ขั้นที่ 2 หาค่าความคลาดเคลื่อน

นำ LDR วัดสีวัตถุเปรียบเทียบ โดย วัดตรงที่แสงเท่าเดิม

LDR 1 (ติดแผ่นกรองสีดำ)

LDR 2 (ติดแผ่นกรองสีเหลือง)

LDR 3 (ติดแผ่นกรองสีม่วง)

LDR 4 (ติดแผ่นสีกรองสีเขียวแกมน้ำเงิน)

จะได้ค่า R1 , R1, R3, R4 เข้าสมการ $R = 475.28x10^{3}L^{-0.85}$ เพื่อหาค่า LCP[0], LCP[1] , LCP[2] , LCP[3]

หมายเหตุ LCP[0] คือ ค่าความสว่างเปรียบเทียบ (L-Compare) ของสีดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LCP[1] คือ ค่าความสว่างเปรียบเทียบ (L- Compare) ของสีเหลือง

LCP[2] คือ ค่าความสว่างเปรียบเทียบ (L- Compare) ของสีม่วง

LCP[3] คือ ค่าความสว่างเปรียบเทียบ (L- Compare) ของสีเขียวแกมน้ำเงิน

ดังนั้นเปอร์เซนต์ ของ 3 สี (เหลือง ,ม่วง , เขียวแกมน้ำเงิน) เมื่อเทียบกับสีดำอ้างอิงเท่ากับ

$$\text{เปอร์เซนต์ของสีเหลือง} = \frac{LCP[1] \times 100}{LST[0]}$$

แต่ค่าที่ได้ต้องถูกคูณค่าคงที่ของแต่ละสีจากขั้นตอนที่1

ดังนั้น ค่าเปอร์เซนต์ที่แท้จริงของสีเหลืองเปรียบเทียบเมื่อเทียบกับสีเหลืองอ้างอิง

$$\text{คือ} \quad \frac{rate[1] \times Lcp[1]}{Lst[0]} \times 100$$

และค่าเปอร์เซนต์ที่แท้จริง ของสีม่วง , เขียวแกมน้ำเงิน เมื่อเทียบกับสีอ้างอิงของตัวเอง

$$\text{คือ} \quad \frac{rate[2] \times Lcp[2]}{Lst[0]} \times 100 \quad , \quad \frac{rate[3] \times Lcp[3]}{Lst[0]} \times 100$$

ฉะนั้น ค่าความคลาดเคลื่อน ของแต่ละสีเท่ากับ

$$ERROR[1] = 33.3 - \left[\frac{rate[1] \times Lcp[1]}{Lst[0]} \right] \times 100$$

$$ERROR[2] = 33.3 - \left[\frac{rate[2] \times Lcp[2]}{Lst[0]} \right] \times 100$$

$$ERROR[3] = 33.3 - \left[\frac{rate[3] \times Lcp[3]}{Lst[0]} \right] \times 100$$

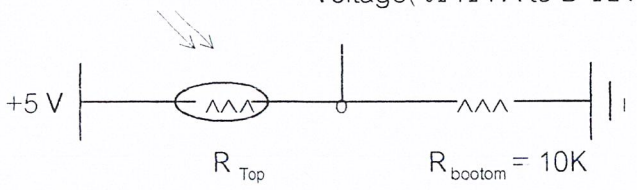
หมายเหตุ ERROR [1] คือความคลาดเคลื่อน ของสีเหลือง

ERROR [2] คือความคลาดเคลื่อน ของสีม่วง

ERROR [3] คือความคลาดเคลื่อน ของสีเขียวแกมน้ำเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>หมายเหตุ</u> การได้ค่าความต้านทาน จาก LDR ก็สามารถทำได้ ดังนี้

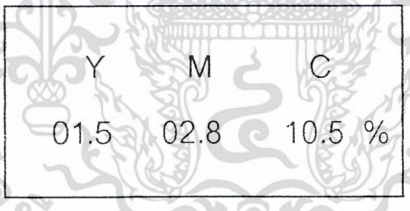


จาก Voltage Divider $V_o = \frac{R_{bottom}}{R_{bottom} + R_{top}} \times V_{in}$

จาก V_o ที่เข้า PIC เพื่อคำนวณก็สามารถหา $R_{TOP}(R_{LDR})$ ได้จากสูตรข้างต้น

4.3 ส่วนที่ 3 (ส่วนแสดงผล)

ส่วนแสดงผลจะแสดงออกหน้าจอ LCD โดยบอกค่า ความคลาดเคลื่อน ของสีทั้ง 3 จากที่คำนวณได้จากส่วนที่ 2 เป็น เปอร์เซ็นต์ ออกมา ตัวอย่างเช่น



รูปที่ 4.1 ตัวอย่างแสดงผลของเครื่องวัดวิเคราะห์สี

<u>หมายเหตุ</u> Y = Yellow M=Magenta C = Cyan บรรทัดที่ 2 คือ เปอร์เซ็นต์ ความคลาดเคลื่อน ของแต่ละสี

<u>4.4 ผลการทดลอง</u>

<u>วิธีที่ 1</u> ทดลองโดยการวัดสีของวัตถุ ณ ตำแหน่งเดิมเพื่อเปรียบเทียบสีของวัตถุนั้น ที่ความสว่างของแสงเท่าเดิม วิธีนี้เป็นการตรวจสอบการทำงานของเครื่องวัดวิเคราะห์สี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สีของวัตถุ	Y (yellow)	M(magenta)	C (cyan)
White	+ 00.7 %	+ 00.1 %	+00.0 %
Black	- 00.2 %	+ 00.2 %	- 00.1 %
Brown	-00.1 %	+ 00.0 %	+ 00.0 %
Red	- 00.2 %	+ 00.1 %	- 00.0 %

ตารางที่ 4.5 แสดงผลการวัดสีของวัตถุ ณ ตำแหน่งเดิม และ ความสว่างเท่ากัน

จากผลการทดลองที่ได้ สรุปได้ว่า การหาค่าความผิดพลาดของการวัดสีวัตถุ ณ ตำแหน่งเดิม ความสว่างเท่ากัน จะให้ค่าความคลาดเคลื่อนเป็น ± 00.0 % หรือ ใกล้เคียง ± 00.0 % มากที่สุด เพราะ L- Reference มีค่าใกล้เคียงกับ L-Compare

วิธีที่ 2 ทดลองการวัดสีของวัตถุโดยการผสมสี

หลักการคือ จากการที่ทราบว่า สีทั้ง 3 สี รวมกันจะเท่ากับสีดำ ดังนั้นทำงานวัด สีดำอ้างอิงไว้ แล้วนำไปวัดกับสีที่ต้องการผสมให้ได้สีดำ แล้วทำการ ผสมสีตามค่าความคลาดเคลื่อนที่อ่านได้จากเครื่องวัดวิเคราะห์สี เช่น นำวัด สีดำอ้างอิง และวัดสีขาว (เพื่อต้องการผสมให้ได้สีดำ)

การผสมสีครั้งที่	วัดสีขาว (เพื่อต้องการผสมสี)		
	Y	M	C
ยังไม่ผสม	- 07.6 %	-06.9 %	-0.54 %
1	-02.3 %	-04.6 %	-04.4 %
2	-00.5 %	-03.3 %	-01.8 %
3	-00.3 %	-01.6 %	-01.7 %
4	-00.0 %	-00.2 %	-00.8 %

ตารางที่ 4.6 แสดงการวัดการผสมสีของสีขาวเพื่อให้ได้สีดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ ค่าที่ได้ติดลบ บ่งบอกได้ว่า วัตถุที่เปรียบเทียบสีนี้อ่อนกว่าวัตถุอ้างอิง ดังนั้น จึงเป็นการเพิ่มสี

ค่าที่ได้เป็นบวก บ่งบอกได้ว่า วัตถุที่เปรียบเทียบมีสีเข้มกว่า ดังนั้นต้องทำการเจือจางให้สีน้อยลง

จากตารางการทดลองที่ 4.6 ในครั้งแรกวัดเปรียบเทียบกับสีขาว(สีขาวต้องการทำให้เป็นสีดำ) จะพบว่ามีค่าติดลบทั้ง 3 สี ดังนั้นจึงทำการเพิ่มสีทั้ง 3 สี ลงบนสีขาวในอัตราส่วนที่สีเหลืองมากที่สุด (ตามค่าความคลาดเคลื่อน) ในการผสมสีครั้งที่1 ให้ค่าสีทั้ง 3 ติดลบน้อยลง ดังนั้นจึงเพิ่ม 3 สีอีก (ในการอัตราส่วนสี magenta,cyan มากกว่า yellow) ในการผสมสีครั้งที่2 ให้ค่าสีทั้ง 3 สี ติดลบน้อยลงอีก ดังนั้นจึงเป็นสี 3 สีอีก(ในอัตราส่วนสี magenta มากที่สุด) ในการผสมสีครั้งที่ 3 ให้ค่าสีทั้ง 3 สี ติดลบน้อยลงมาก แต่ต้องทำการเพิ่มสีอีกทั้ง 3 สีอีก เพื่อให้ค่าใกล้เคียง ± 00.0 % มากที่สุด ฉะนั้นในการผสมสีครั้งที่ 4 จะให้ค่าสีใกล้เคียงสีดำมากที่สุด(ความคลาดเคลื่อนใกล้เคียง ± 00.0 %)

ประโยชน์จากการทดลองนี้สามารถเปรียบเทียบได้กับกระบวนการวัดเพื่อพ่นสีของรถยนต์

วิธีที่ 3 วัดวัตถุอ้างอิงและนำวัตถุที่ต้องการเปรียบเทียบหาค่าความคลาดเคลื่อนว่ามีค่ายอมรับได้หรือไม่ ตัวอย่างเช่น วัดสีแดง 1 เป็นอ้างอิง แล้วทำการวัดสีแดง 2 และสีแดง 3 เปรียบเทียบกับสีแดงอ้างอิง

2 1 3

วัตถุชิ้นที่	Y	M	C
สีแดง 2 เมื่อเทียบกับสีแดง	+ 02.1	+07.1	+ 03.6
สีแดง 3 เมื่อเทียบกับสีแดง	+ 02.5	- 02.0	- 01.3 %

ตารางที่4.7 แสดงการเปรียบเทียบความคลาดเคลื่อนของสีอ้างอิงกับสีเปรียบเทียบ

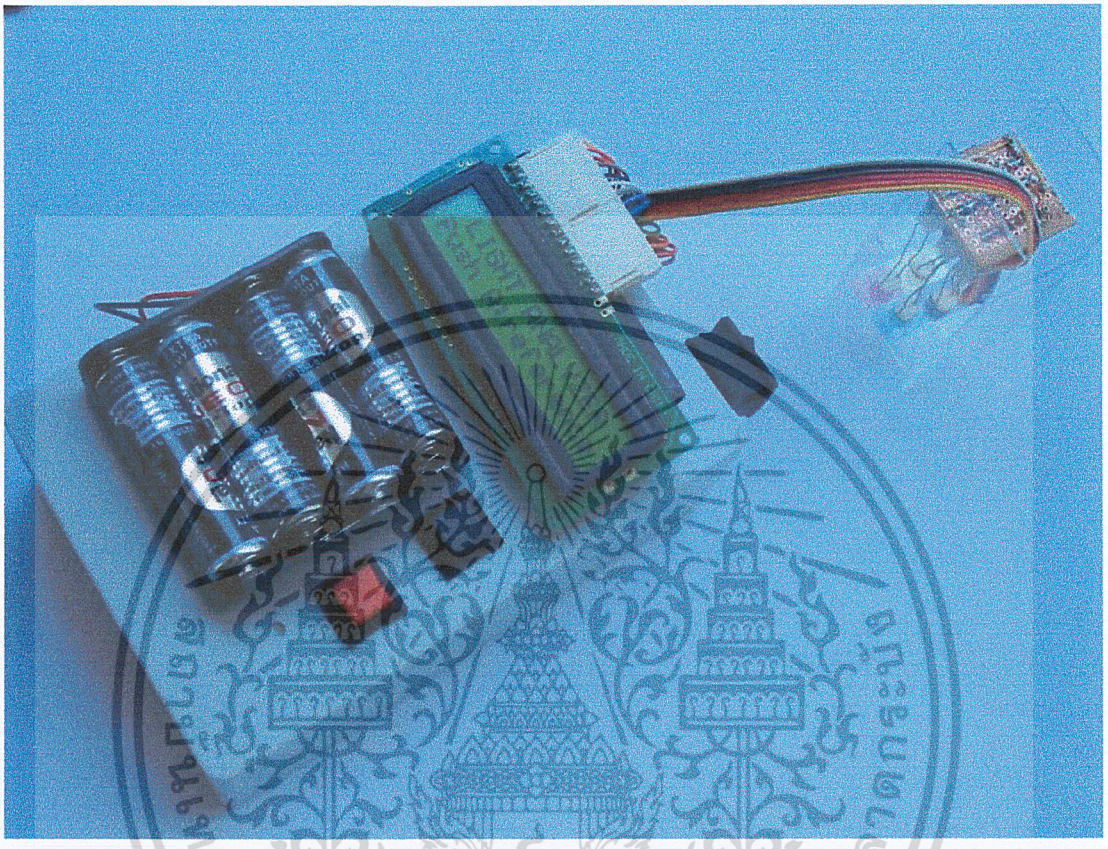
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองนี้เปรียบสีแดง1เป็นสีของมะเขือเทศอ้างอิงและสีแดง2สีแดง3เป็นสีของมะเขือเทศที่ต้องการวัดคุณภาพสีว่าแตกต่างกับมะเขือเทศอ้างอิงพอยอมรับได้หรือไม่(ค่าความแตกต่างขึ้นอยู่กับการกำหนดของผู้ใช้)

สมมติโรงงานแห่งหนึ่งกำหนดค่าความคลาดเคลื่อนของสีทั้ง3สีไว้ที่ ± 05.0 % ดังนั้นสีแดง2(มะเขือเทศที่เปรียบเทียบ) มีความคลาดเคลื่อนเกิน ± 05.0% จึงคัดมะเขือเทศชิ้นนี้ออกสีแดง3(มะเขือเทศที่เปรียบเทียบ) มีความคลาดเคลื่อนอยู่ในช่วง ±05.0% ดังนั้นมะเขือเทศชิ้นนี้จึงผ่านกระบวนการคัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพเครื่องวัดวิเคราะห์สี(Color Analyzer)

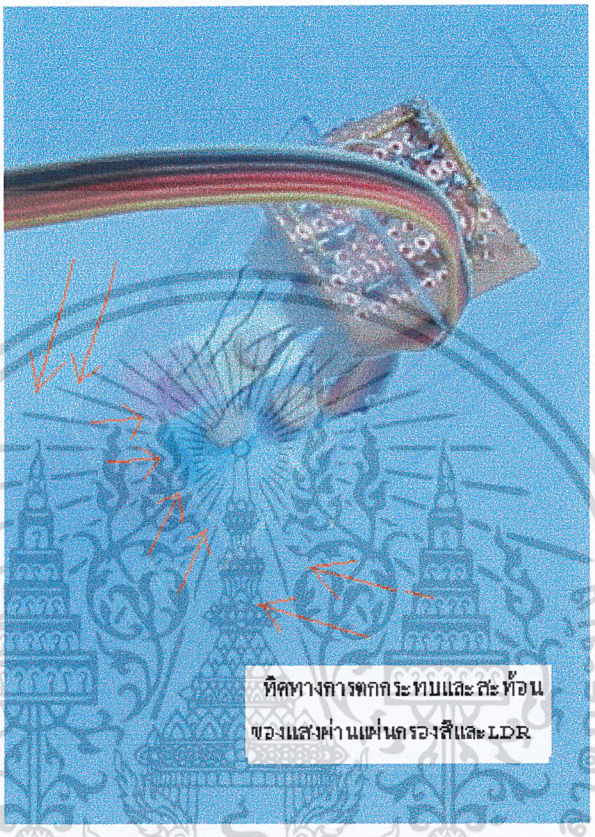


รูปที่ 4.2 เครื่องวัดวิเคราะห์สี(Color Analyzer)

จากรูปคือ เครื่องวัดวิเคราะห์สีภายนอกประกอบไปด้วย แหล่งจ่ายไฟ คือ แบตเตอรี่ , ปุ่มกดจำนวน 2 ปุ่ม , ปุ่ม A (สีแดง) , ปุ่ม B (สีดำ) , หน้าจอแสดงผลแบบผลึกเหลว(LCD) และส่วนหัวรับแสง ภายในประกอบไปด้วยวงจรการทำงานของเครื่องวัดวิเคราะห์สี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพการสะท้อนแสงผ่านแผ่นกรองสีและLDRของเครื่องวัดวิเคราะห์สี



รูปที่ 4.3 การสะท้อนแสงผ่านแผ่นกรองสีและLDR

จากรูปคือ ส่วนหัวรับแสงของเครื่องวัดวิเคราะห์สีผ่านตัวต้านทานไวแสง (LDR) ที่ติดแผ่นกรองสี 4 สี คือ สีดำ เหลือง ม่วง และเขียวแกมน้ำเงิน ลูกศรคือ แสงที่กระทบวัตถุและสะท้อนผ่านแผ่นกรองสีและ LDR เพื่อที่ LDR จะเปลี่ยนค่าความต้านทานเพื่อใช้ในการคำนวณต่อไป ลักษณะของวัตถุที่วัดจะต้องมีขนาดพื้นที่ผิวเท่ากับหรือใหญ่กว่าขนาดของส่วนหัวที่รับแสงของเครื่องวัดวิเคราะห์สี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและวิจารณ์

5.1 ปัญหาและแนวทางแก้ไข

การรับค่า แรงดัน จาก ตัวต้านทานไวแสง(LDR) เข้าสู่ อนาลอกเป็นดิจิตอล(Analog to Digital Converter) ของ ไมโครคอนโทรลเลอร์ตระกูล PIC การรับค่าจะมีการเปลี่ยนแปลงอย่างรวดเร็ว ในการรับค่าจาก อนาลอกแปลงเป็น ดิจิตอล เพียงครั้งเดียวจะได้ค่าที่ค่อนข้างไม่ถูกต้องต่อการคำนวณ ดังนั้นในโปรแกรมจึงมีส่วนการวนรับค่า แรงดัน เพื่อแปลงเป็น ดิจิตอล จำนวนหลายครั้ง แล้วนำมาเฉลี่ยเพื่อให้ได้ค่าที่ถูกต้องที่สุดในการคำนวณ

การหาค่าคงที่ของ ตัวต้านทานไวแสง(LDR) การศึกษาจะต้องทำการหาค่าคงที่ทั้ง 4 ตัว จะหาค่าคงที่เพียงตัวเดียวและสมมติว่าเท่ากันกับอีก 3 ตัวไม่ได้ เพราะการทำเครื่องมือวัดต้องอาศัยความถูกต้องที่สุดและ ตัวต้านทานไวแสง(LDR) แต่ละตัวจะมีค่าคงที่ไม่เท่ากัน ดังนั้นเพื่อความถูกต้องมากที่สุด จึงต้องหาค่าคงที่ทั้ง 4 ตัว

การทดลองเพื่อต้องการให้ได้ค่าถูกต้องมากที่สุด ต้องทำการวัดวัตถุหรือสิ่งที่ต้องการวัด ณ จุดที่มีแสงเท่ากันหรือใกล้เคียงกันมากที่สุด เพื่อความถูกต้องมากยิ่งขึ้น

5.2 แนวทางในการพัฒนา

เครื่องวัดวิเคราะห์สีสามารถจะพัฒนาให้มีคุณภาพตามที่ต้องการให้ดียิ่งขึ้นได้เพื่อให้เหมาะกับอุตสาหกรรมที่ต้องการนำไปใช้ประโยชน์นั้นๆ ยกตัวอย่างเช่น การเพิ่มจำนวนสีที่ต้องการวัดให้มีมากขึ้น ในอุตสาหกรรมที่ต้องการความละเอียดการเพิ่มฟังก์ชันหน้าที่การใช้งานให้มีมากขึ้น อาจมีการเพิ่ม โหมดการแสดงค่าอื่นๆ นอกจากการแสดงค่าความคลาดเคลื่อนออกมา เพื่อความสะดวกและเป็นประโยชน์มากยิ่งขึ้น การนำเครื่องวัดวิเคราะห์สีมาใช้กับ Lux meter เพื่อให้ได้ค่าแสงที่ถูกต้องมากยิ่งๆขึ้นได้ ตลอดจนพัฒนาโปรแกรมให้สามารถใช้งานง่าย เพื่อความเข้าใจ การนำเครื่องวัดวิเคราะห์สีไปประยุกต์ให้เข้ากับอุตสาหกรรมจะเป็นประโยชน์ที่สำคัญอย่างยิ่งต่อๆไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 บทสรุปและวิจารณ์

เครื่องวัดวิเคราะห์สีประกอบไปด้วย ส่วนหลักๆ 3 ส่วน

1. ส่วนที่1ตัวต้านทานไวแสง(LDR)คือรับค่าความเข้มแสงผ่านLDR4ตัวจากสี4สี คือ ดำ,เหลือง ม่วง,เขียวแกมน้ำเงิน โดยค่า ความต้านทานจะ เปลี่ยนเมื่อได้รับแสง
2. ส่วนที่ 2 (ประมวณผล) คือส่วนนำค่า ความต้านทานที่เปลี่ยนแปลงมาหาค่า ความสว่างอ้างอิงจากสมการและวัดค่าความสว่างที่ต้องการเปรียบเทียบเพื่อหาค่าความคลาดเคลื่อนออกมา
3. ส่วนที่ 3 (แสดงผล) คือส่วนที่แสดงค่าความคลาดเคลื่อนของทั้ง 3 สีออกมาเป็น ความคลาดเคลื่อน บนหน้าจอ LCD

อนึ่งผู้จัดทำได้เลือกใช้การเขียนโปรแกรมภาษา C และมี PCW เป็นโปรแกรม Compile เพื่อเปลี่ยน .C เป็น .H สาเหตุที่เลือกใช้ภาษา C เพราะ

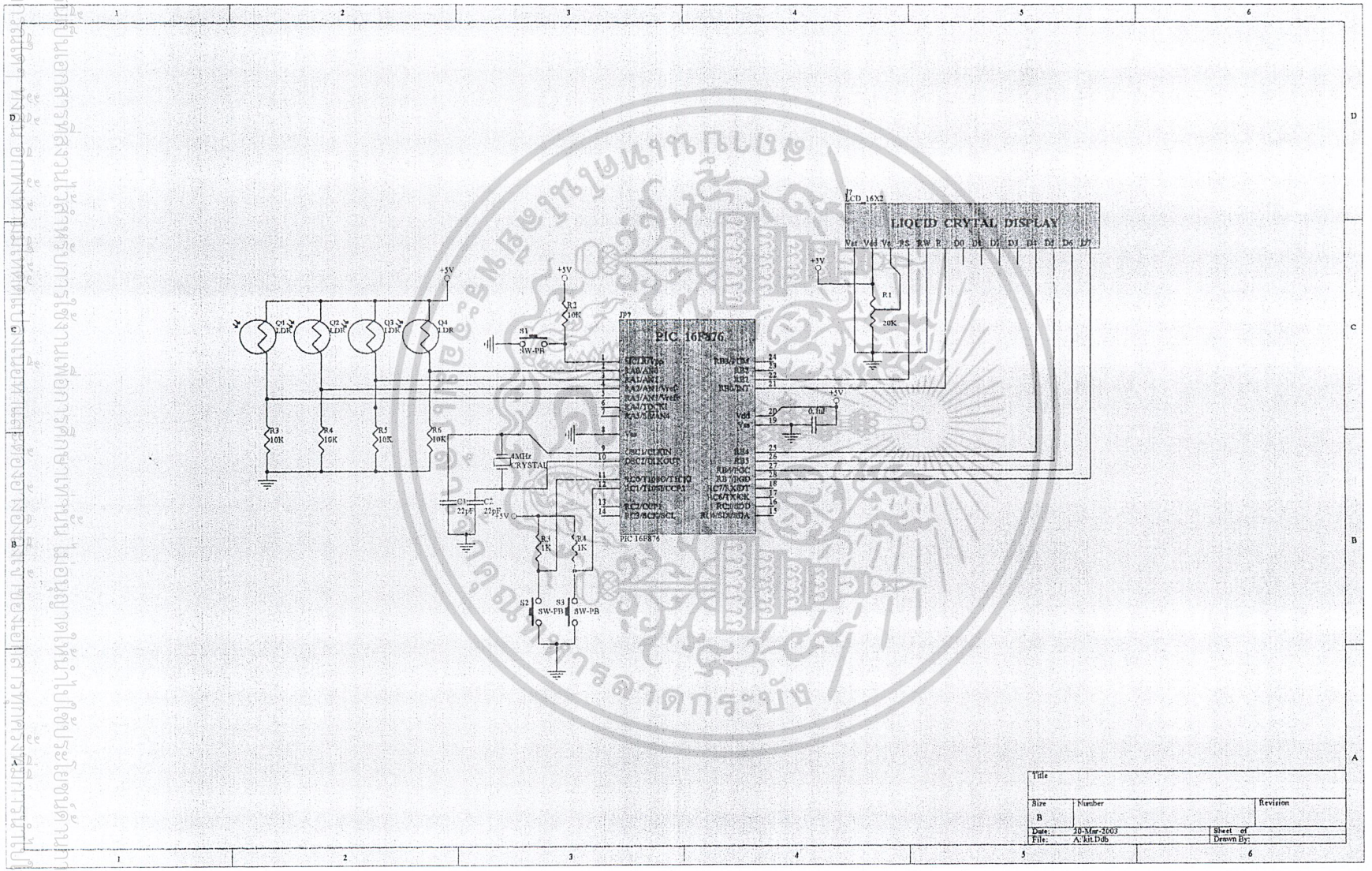
- ความสะดวกและง่ายในการใช้ สามารถเรียนรู้ได้ง่ายและภาษามีโครงสร้างดีทำให้การเขียนโปรแกรมค่อนข้างง่ายเมื่อเทียบกับภาษาแอสแซมบลี
- สามารถทำแอพพริเคชั่นที่ซับซ้อน ภาษา C มีประโยคที่ใช้ควบคุมทิศทาง การประมวลผลแบบลอจิกรวมทั้งไลบรารี่ฟังก์ชันทางคณิตศาสตร์ เช่น คำสั่ง คูณ หาร และฟังก์ชันทางตรีโกณมิติรวมทั้งการแก้สมการล๊อก(log) ฐาน10(จากสมการที่ใช้ $R = 475.28x10^{3} L^{-0.85}$ เป็นการแก้สมการ ล๊อก ฐาน10)

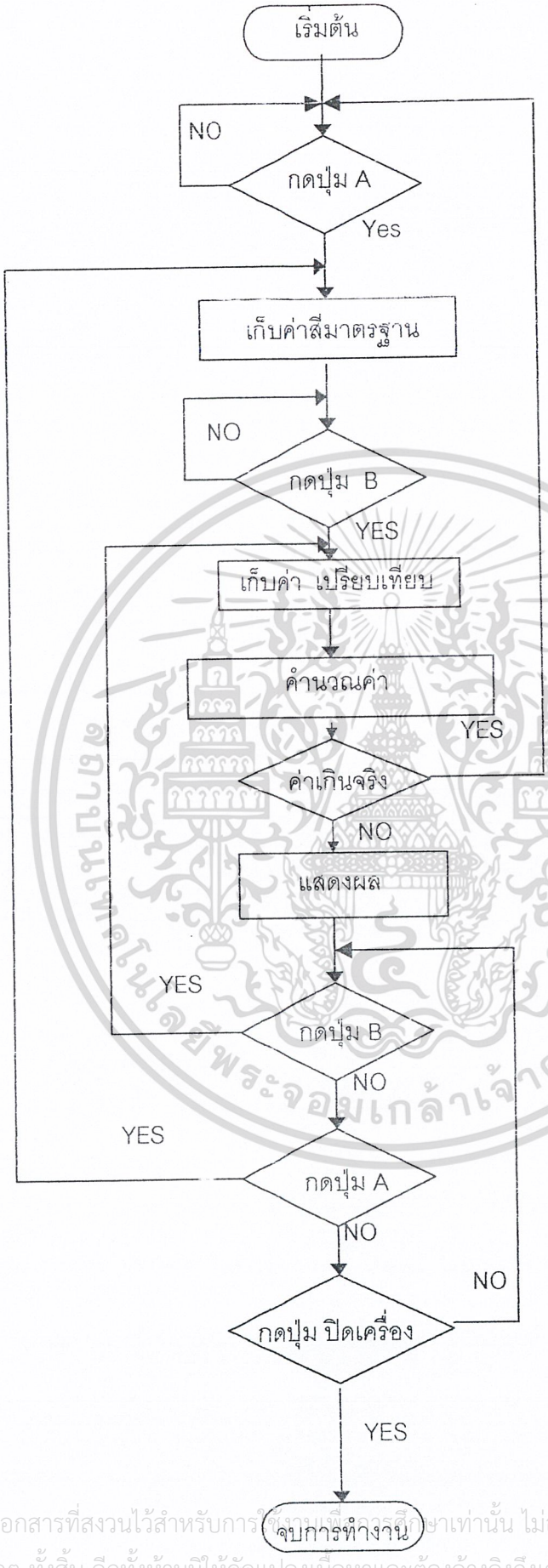
เครื่องวัดวิเคราะห์สีจัดเป็นเครื่องมือพื้นฐานในการพัฒนาเครื่องมือวัดให้มีมาตรฐานยิ่งๆขึ้นเพื่อนำไปใช้ในอุตสาหกรรมการวัด ตลอดจนคนไทยสามารถคิดค้น ประดิษฐ์ ได้อย่างเป็นมาตรฐาน เพื่อเป็นการลดการนำเข้าเครื่องมือวัดประเภทนี้จากต่างประเทศ อันจะทำให้เสียดุลการค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


LCD_16X2
LIQUID CRYTAL DISPLAY
Vss Vdd Vo RS RW E D0 D1 D2 D3 D4 D5 D6 D7
+5V
R1
20K
JP7
PIC 16F876
Q1 LDR
Q2 LDR
Q3 LDR
Q4 LDR
R3 10K
R4 10K
R5 10K
R6 10K
S1 SW-PB
R2 10K
4MHz CRYSTAL
C1 22pF
C2 22pF
0.1uf
R4 1K
R4 1K
S2 SW-PB
S3 SW-PB
Title
Size B
Number
Revision
Date: 10-Mar-2003
File: A:\kit.Ddb
Sheet of
Drawn By:



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <PRO4.H>
#include <LCD.C>
#include <math.h>

#bit sw1=7.0
#bit sw2=7.1
#byte port_c=0x07

#int_rtcc
rtcc_isr() {
        // Timer0 (RTCC) overflow
}

float average[4],sum,j[4],l[4],x[4],Lst[4],Lcp[4],rate[4],q,error[4];
long e_long[4];
int  x1[4],x2[4],x3[4],x4[4],x5[4],key,unit,val,i,num,set;
char k[16],n,n1,n2,n3;
short ref;


void read(int set)
{
  long tmp;
  float t1,value;
  for(i=set;i<4;i++)
    {
     set_adc_channel(i);
     value = 0 ;
   for(val=0; val<90; val++) {

     tmp = read_adc();
     tmp = tmp&0x03ff;
     t1 = tmp;
     t1 = t1*(5.0/1024.0);
     value = value + t1;
         }
     value = value / 90.0;
     average[i]= value;
     j[i] = 6.68-(log10((50000.0/average[i])-10000.0)/0.85);
     l[i] = exp(j[i]*log(10));
    }
}

void output(void)
{
  x[i] = error[i]*10;
  e_long[i] = x[i];
  x1[i] = e_long[i]/100;
     if (x1[i]>10)
         unit=10;
     else
         unit=9;
         x2[i] = x1[i]*10;
         x1[i] = x1[i]+0x30;
         x3[i] = e_long[i]/10;
         x4[i] = x3[i]-x2[i];
         x4[i] = x4[i]+0x30;
         x5[i] = e_long[i]%10;
         x5[i] = x5[i]+0x30;
```

เอกสารนี้เผยแพร่ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void cal_value(void)
{
   for (i=1;i<4;i++)
   {
   error[i] = 33.3-((rate[i]*Icp[i])/Ist[0])*100.0;
    if (error[1]<0.0)
        n1='-';
       if(error[2]<0.0)
        n2='-';
        if(error[3]<0.0)
         n3='-';
             else{n1='+';
                  n2='+';
                   n3='+';}
   output();
   if (unit==10)
     break;
  }
}
void display(void)
{
    if (unit==10)
    {
    lcd_putc("\f !! OVERLOAD !!   \n");
    lcd_putc(" Push A again");
    }
    else if (unit==9)
    {
    k[0]=n1;
    k[1]=x1[1];
    k[2]=x4[1];
    k[3]='.';
    k[4]=x5[1];
    k[5]=n2;
    k[6]=x1[2];
    k[7]=x4[2];
    k[8]='.';
    k[9]=x5[2];
    k[10]=n3;
    k[11]=x1[3];
    k[12]=x4[3];
    k[13]='.';
    k[14]=x5[3];
    k[15]='%';
    lcd_putc("\f  Y   M   C\n");
    for(num=0;num<16;num++)
      lcd_putc(k[num]);
    }
}

main() {
          setup_counters(RTCC_INTERNAL,RTCC_DIV_16);
          setup_port_a(ALL_ANALOG);
          setup_adc(ADC_CLOCK_INTERNAL);
          port_b_pullups(TRUE);
          enable_interrupts(RTCC_ZERO);
          enable_interrupts(GLOBAL);
          lcd_init();
 set_tris_c(0x03);
```

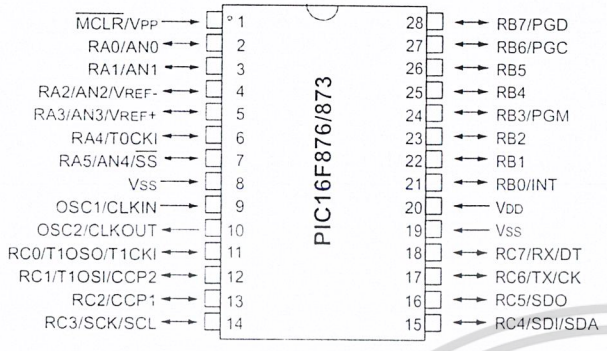
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
set = 8;
lcd_putc("\f LIGHT ANALYZER\n");
lcd_putc("Push A:reference");
ref = 0;

while(1)
{
port_c = 0x03;
delay_us(10);
key = port_c&0x03;

switch(key)
   {
    case(0x02) :
            set = 0;
            break;

    case(0x01) :
            set = 1;
            break;
   }

switch(set)
   {
    case 0:
        read(set);
        for (i=0;i<4;i++)
           Lst[i] = I[i];
        q = (Lst[1]*100.0)/(Lst[0]);
        rate[1] = 33.3/q;
        q = (Lst[2]*100.0)/(Lst[0]);
        rate[2] = 33.3/q;
        q = (Lst[3]*100.0)/(Lst[0]);
        rate[3] = 33.3/q;
        lcd_putc("\fRecord reference\n");
        lcd_putc(" Push B:compare ");
        set = 8;
        ref = 1;
        break;
    case 1:
        if(ref ==1)
        {
        read(set);
            for (i=1;i<4;i++)
          Lcp[i] = I[i];
            lcd_putc("\f Record compare\n");
            lcd_putc("SHOW ERROR VALUE");
            delay_ms(1000) ;
        cal_value();
        display();
        set=8;
        }
        else if (ref==0)
        {
            lcd_putc("\f No Record Ref!!\n");
        lcd_putc("  Push A before");
        set=8;
        }
        break;
   }
}
```

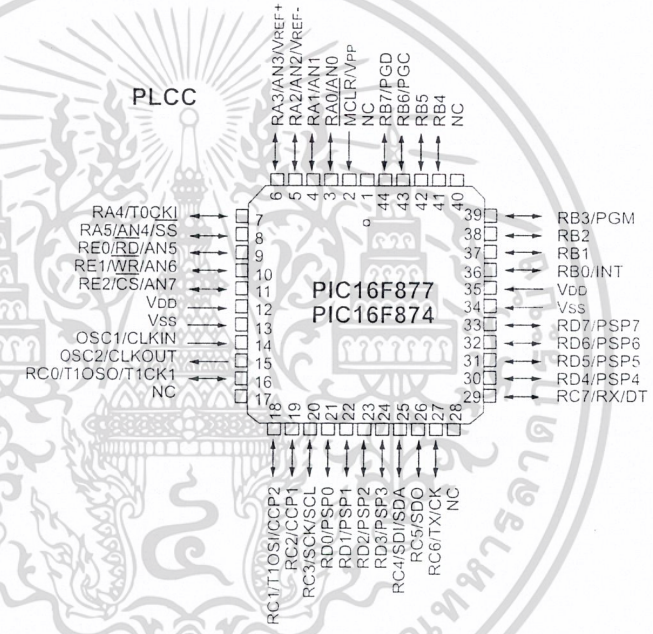
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Diagrams

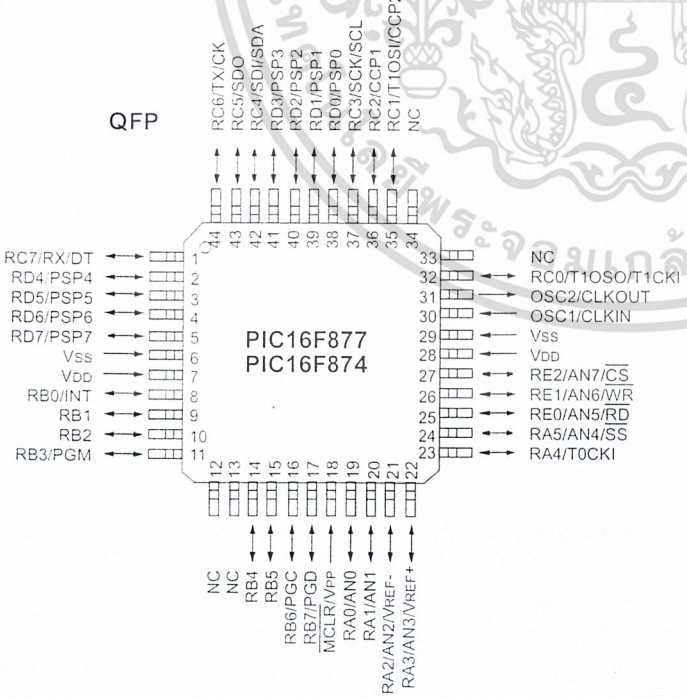
PDIP, SOIC



PLCC



QFP



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

© 2001 Microchip Technology Inc.

Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้จัดทำขึ้นสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.0 DEVICE OVERVIEW

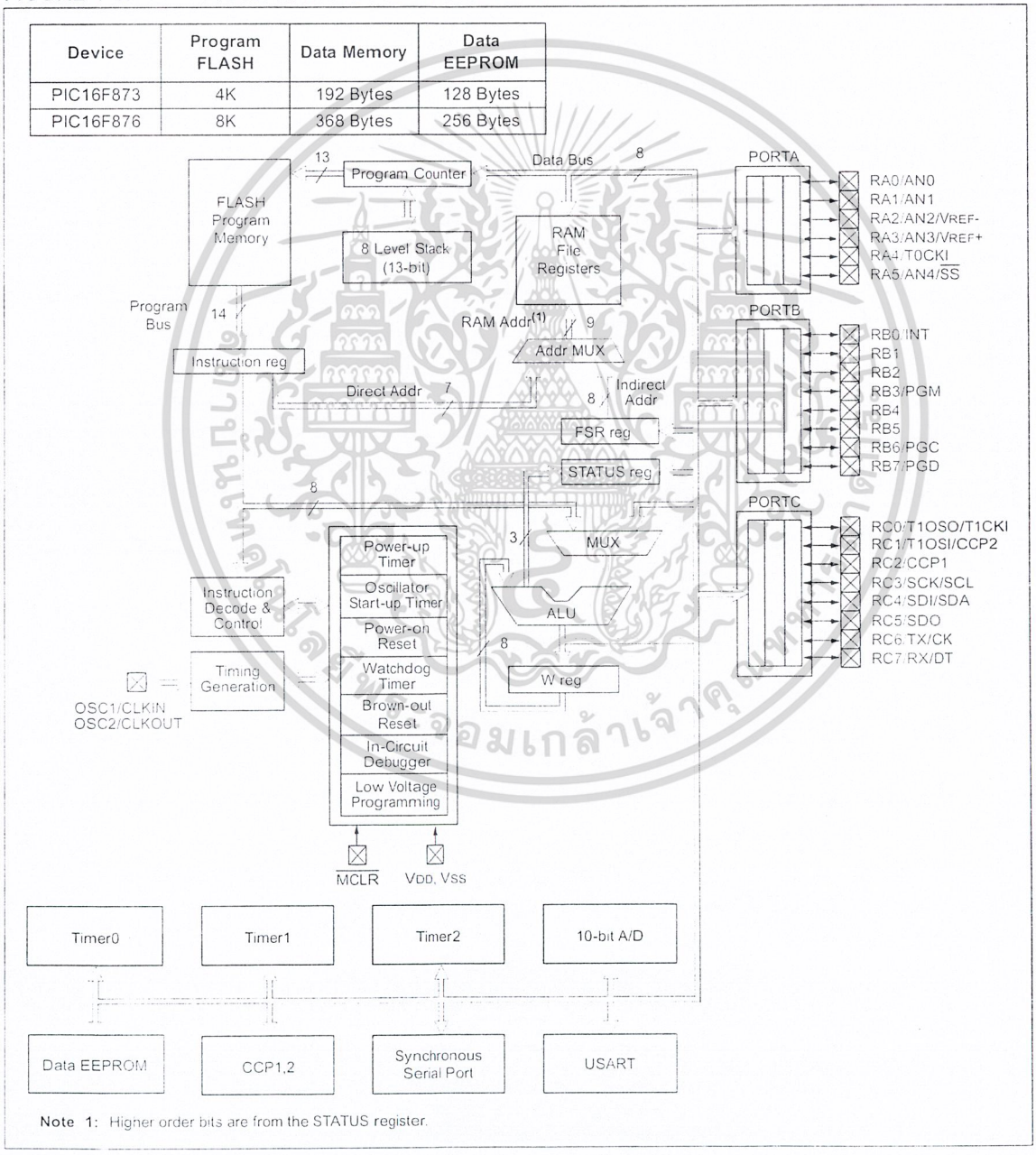
This document contains device specific information. Additional information may be found in the PICmicro™ Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip website. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

There are four devices (PIC16F873, PIC16F874, PIC16F876 and PIC16F877) covered by this data sheet. The PIC16F876/873 devices come in 28-pin packages and the PIC16F877/874 devices come in 40-pin packages. The Parallel Slave Port is not implemented on the 28-pin devices.

The following device block diagrams are sorted by pin number; 28-pin for Figure 1-1 and 40-pin for Figure 1-2. The 28-pin and 40-pin pinouts are listed in Table 1-1 and Table 1-2, respectively.

FIGURE 1-1: PIC16F873 AND PIC16F876 BLOCK DIAGRAM

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F873	4K	192 Bytes	128 Bytes
PIC16F876	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.

เอกสารนี้ถูกสร้างขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 1-1: PIC16F873 AND PIC16F876 PINOUT DESCRIPTION

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS(3)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	1	1	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
					PORTA is a bi-directional I/O port.
RA0/AN0	2	2	I/O	TTL	RA0 can also be analog input0.
RA1/AN1	3	3	I/O	TTL	RA1 can also be analog input1.
RA2/AN2/VREF-	4	4	I/O	TTL	RA2 can also be analog input2 or negative analog reference voltage.
RA3/AN3/VREF+	5	5	I/O	TTL	RA3 can also be analog input3 or positive analog reference voltage.
RA4/T0CKI	6	6	I/O	ST	RA4 can also be the clock input to the Timer0 module. Output is open drain type.
RA5/$\overline{SS}$/AN4	7	7	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
					PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	21	21	I/O	TTL/ST(1)	RB0 can also be the external interrupt pin.
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL	RB3 can also be the low voltage programming input.
RB4	25	25	I/O	TTL	Interrupt-on-change pin.
RB5	26	26	I/O	TTL	Interrupt-on-change pin.
RB6/PGC	27	27	I/O	TTL/ST(2)	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.
RB7/PGD	28	28	I/O	TTL/ST(2)	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.
					PORTC is a bi-directional I/O port.
RC0/T1OSO/T1CKI	11	11	I/O	ST	RC0 can also be the Timer1 oscillator output or Timer1 clock input.
RC1/T1OSI/CCP2	12	12	I/O	ST	RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	13	13	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/ PWM1 output.
RC3/SCK/SCL	14	14	I/O	ST	RC3 can also be the synchronous serial clock input/output for both SPI and I^2C modes.
RC4/SDI/SDA	15	15	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	16	16	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6/TX/CK	17	17	I/O	ST	RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	18	18	I/O	ST	RC7 can also be the USART Asynchronous Receive or Synchronous Data.
VSS	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.0 MEMORY ORGANIZATION

There are three memory blocks in each of the PIC16F87X MCUs. The Program Memory and Data Memory have separate buses so that concurrent access can occur and is detailed in this section. The EEPROM data memory block is detailed in Section 4.0.

Additional information on device memory may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

2.1 Program Memory Organization

The PIC16F87X devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. The PIC16F877/876 devices have 8K x 14 words of FLASH program memory, and the PIC16F873/874 devices have 4K x 14. Accessing a location above the physically implemented address will cause a wraparound.

The RESET vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 2-1: PIC16F877/876 PROGRAM MEMORY MAP AND STACK

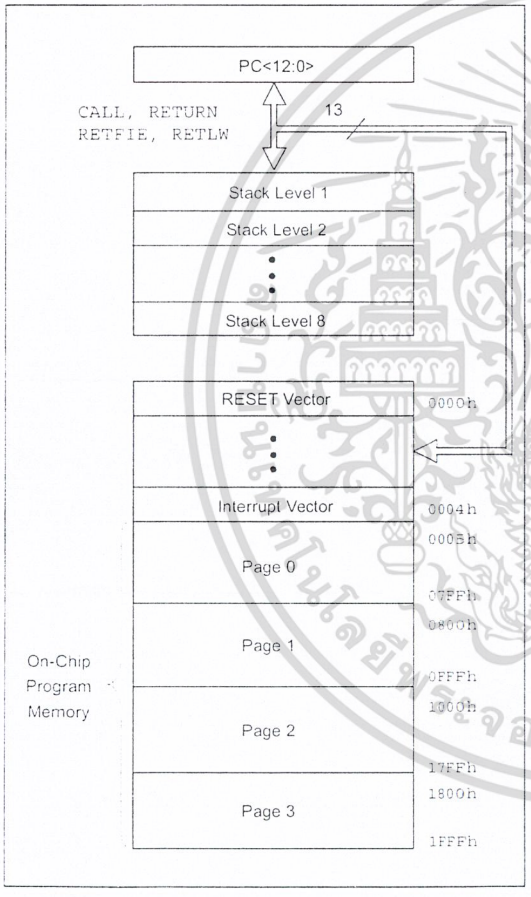
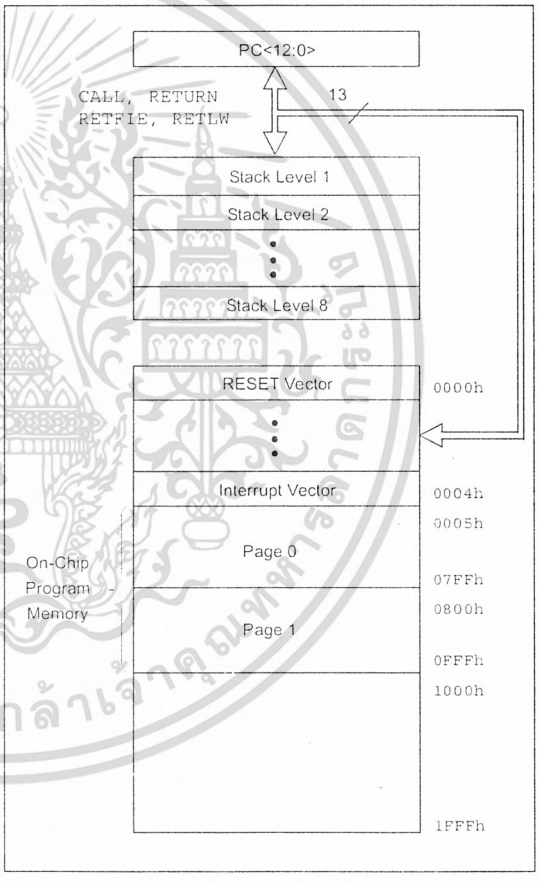


FIGURE 2-2: PIC16F874/873 PROGRAM MEMORY MAP AND STACK



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the Special Function Registers. Bits RP1 (STATUS<6>) and RP0 (STATUS<5>) are the bank select bits.

RP1:RP0	Bank
00	0
01	1
10	2
11	3

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some frequently used Special Function Registers from one bank may be mirrored in another bank for code reduction and quicker access.

Note:	EEPROM Data Memory description can be found in Section 4.0 of this data sheet.

2.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly, or indirectly through the File Select Register (FSR).

© 2001 Microchip Technology Inc.
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 2-3: PIC16F877/876 REGISTER FILE MAP

Bank 0	File Address	Bank 1	File Address	Bank 2	File Address	Bank 3	File Address
Indirect addr.(*)	00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD(1)	08h	TRISD(1)	88h		108h		188h
PORTE(1)	09h	TRISE(1)	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved(2)	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved(2)	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h	General Purpose Register 16 Bytes	117h	General Purpose Register 16 Bytes	197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Register 96 Bytes	20h 7Fh	General Purpose Register 80 Bytes	A0h EFh	General Purpose Register 80 Bytes	120h 16Fh	General Purpose Register 80 Bytes	1A0h 1EFh
accesses 70h-7Fh		accesses 70h-7Fh	F0h FFh	accesses 70h-7Fh	170h 17Fh	accesses 70h - 7Fh	1F0h 1FFh

☐ Unimplemented data memory locations, read as '0'.
* Not a physical register.

Note 1: These registers are not implemented on the PIC16F876.
2: These registers are reserved, maintain these registers clear.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ หาประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 2-4: PIC16F874/873 REGISTER FILE MAP

File Address	Bank 0	File Address	Bank 1	File Address	Bank 2	File Address	Bank 3
00h	Indirect addr.(*)	80h	Indirect addr.(*)	100h	Indirect addr.(*)	180h	Indirect addr.(*)
01h	TMR0	81h	OPTION_REG	101h	TMR0	181h	OPTION_REG
02h	PCL	82h	PCL	102h	PCL	182h	PCL
03h	STATUS	83h	STATUS	103h	STATUS	183h	STATUS
04h	FSR	84h	FSR	104h	FSR	184h	FSR
05h	PORTA	85h	TRISA	105h		185h	
06h	PORTB	86h	TRISB	106h	PORTB	186h	TRISB
07h	PORTC	87h	TRISC	107h		187h	
08h	PORTD(1)	88h	TRISD(1)	108h		188h	
09h	PORTE(1)	89h	TRISE(1)	109h		189h	
0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH
0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh	INTCON
0Ch	PIR1	8Ch	PIE1	10Ch	EEDATA	18Ch	EECON1
0Dh	PIR2	8Dh	PIE2	10Dh	EEADR	18Dh	EECON2
0Eh	TMR1L	8Eh	PCON	10Eh	EEDATH	18Eh	Reserved(2)
0Fh	TMR1H	8Fh		10Fh	EEADRH	18Fh	Reserved(2)
10h	T1CON	90h		110h		190h	
11h	TMR2	91h	SSPCON2				
12h	T2CON	92h	PR2				
13h	SSPBUF	93h	SSPADD				
14h	SSPCON	94h	SSPSTAT				
15h	CCPR1L	95h					
16h	CCPR1H	96h					
17h	CCP1CON	97h					
18h	RCSTA	98h	TXSTA				
19h	TXREG	99h	SPBRG				
1Ah	RCREG	9Ah					
1Bh	CCPR2L	9Bh					
1Ch	CCPR2H	9Ch					
1Dh	CCP2CON	9Dh					
1Eh	ADRESH	9Eh	ADRESL				
1Fh	ADCON0	9Fh	ADCON1				
20h	General Purpose Register 96 Bytes	A0h	General Purpose Register 96 Bytes	120h	accesses 20h-7Fh	1A0h	accesses A0h - FFh
7Fh		FFh		16Fh 170h		1EFh 1F0h	
				17Fh		1FFh	

☐ Unimplemented data memory locations, read as '0'.

* Not a physical register.

Note 1: These registers are not implemented on the PIC16F873.

2: These registers are reserved, maintain these registers clear.

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

© 2001 Microchip Technology Inc.

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 2-1.

The Special Function Registers can be classified into two sets: core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. Those related to the operation of the peripheral features are described in detail in the peripheral features section.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:
Bank 0											
00h(3)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
01h	TMR0	Timer0 Module Register								xxxx xxxx	47
02h(3)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26
03h(3)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	18
04h(3)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	29
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	31
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	33
08h(4)	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	35
09h(4)	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	36
0Ah(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
0Bh(3)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20
0Ch	PIR1	PSPIF(3)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	22
0Dh	PIR2	—	(5)	—	EEIF	BCLIF	—	—	CCP2IF	-r-0 0--0	24
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	52
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	52
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	51
11h	TMR2	Timer2 Module Register								0000 0000	55
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	55
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	70, 73
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	67
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	57
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	57
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	58
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	96
19h	TXREG	USART Transmit Data Register								0000 0000	99
1Ah	RCREG	USART Receive Data Register								0000 0000	101
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	57
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	57
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	58
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	116
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	$\overline{GO/DONE}$	—	ADON	0000 00-0	111

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.
2: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.
3: These registers can be addressed from any bank.
4: PORTD, PORTE, TRISD, and TRISE are not physically implemented on PIC16F873/876 devices; read as '0'.
5: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

เอกสารนี้มีขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:
Bank 1											
80h(3)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
81h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	19
82h(3)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26
83h(3)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	18
84h(3)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	29
86h	TRISB	PORTB Data Direction Register								1111 1111	31
87h	TRISC	PORTC Data Direction Register								1111 1111	33
88h(4)	TRISD	PORTD Data Direction Register								1111 1111	35
89h(4)	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	37
8Ah(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
8Bh(3)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20
8Ch	PIE1	PSPIE(2)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	21
8Dh	PIE2	—	(5)	—	EEIE	BCLIE	—	—	CCP2IE	-x-0 0--0	23
8Eh	PCON	—	—	—	—	—	—	$\overline{POR}$	$\overline{BOR}$	---- --qq	25
8Fh	—	Unimplemented								—	—
90h	—	Unimplemented								—	—
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	68
92h	PR2	Timer2 Period Register								1111 1111	55
93h	SSPADD	Synchronous Serial Port (I^2C mode) Address Register								0000 0000	73, 74
94h	SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	66
95h	—	Unimplemented								—	—
96h	—	Unimplemented								—	—
97h	—	Unimplemented								—	—
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	95
99h	SPBRG	Baud Rate Generator Register								0000 0000	97
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	—	Unimplemented								—	—
9Dh	—	Unimplemented								—	—
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	116
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	00-- 0000	112

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.

3: These registers can be addressed from any bank.

4: PORTD, PORTE, TRISD, and TRISE are not physically implemented on PIC16F873/876 devices; read as '0'.

5: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

© 2001 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:
Bank 2											
100h(3)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
101h	TMR0	Timer0 Module Register								xxxx xxxx	47
102h(3)	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	26
103h(3)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	18
104h(3)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
105h	—	Unimplemented								—	—
106h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	31
107h	—	Unimplemented								—	—
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
10Bh(3)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20
10Ch	EEDATA	EEPROM Data Register Low Byte								xxxx xxxx	41
10Dh	EEADR	EEPROM Address Register Low Byte								xxxx xxxx	41
10Eh	EEDATH	—	—	EEPROM Data Register High Byte						xxxx xxxx	41
10Fh	EEADRH	—	—	—	EEPROM Address Register High Byte					xxxx xxxx	41
Bank 3											
180h(3)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	19
182h(3)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26
183h(3)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	18
184h(3)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
185h	—	Unimplemented								—	—
186h	TRISB	PORTB Data Direction Register								1111 1111	31
187h	—	Unimplemented								—	—
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
18Bh(3)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	41, 42
18Dh	EECON2	EEPROM Control Register2 (not a physical register)								---- ----	41
18Eh	—	Reserved maintain clear								0000 0000	—
18Fh	—	Reserved maintain clear								0000 0000	—

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.

3: These registers can be addressed from any bank.

4: PORTD, PORTE, TRISD, and TRISE are not physically implemented on PIC16F873/876 devices; read as '0'.

5: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) Converter module has five inputs for the 28-pin devices and eight for the other devices.

The analog input charges a sample and hold capacitor. The output of the sample and hold capacitor is the input into the converter. The converter then generates a digital result of this analog level via successive approximation. The A/D conversion of the analog input signal results in a corresponding 10-bit digital number. The A/D module has high and low voltage reference input that is software selectable to some combination of VDD, Vss, RA2, or RA3.

The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode. To operate in SLEEP, the A/D clock must be derived from the A/D's internal RC oscillator.

The A/D module has four registers. These registers are:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register0 (ADCON0)
- A/D Control Register1 (ADCON1)

The ADCON0 register, shown in Register 11-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 11-2, configures the functions of the port pins. The port pins can be configured as analog inputs (RA3 can also be the voltage reference), or as digital I/O.

Additional information on using the A/D module can be found in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

REGISTER 11-1: ADCON0 REGISTER (ADDRESS: 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON
bit 7							bit 0

bit 7-6 **ADCS1:ADCS0:** A/D Conversion Clock Select bits
00 = Fosc/2
01 = Fosc/8
10 = Fosc/32
11 = FRC (clock derived from the internal A/D module RC oscillator)

bit 5-3 **CHS2:CHS0:** Analog Channel Select bits
000 = channel 0, (RA0/AN0)
001 = channel 1, (RA1/AN1)
010 = channel 2, (RA2/AN2)
011 = channel 3, (RA3/AN3)
100 = channel 4, (RA5/AN4)
101 = channel 5, (RE0/AN5)(1)
110 = channel 6, (RE1/AN6)(1)
111 = channel 7, (RE2/AN7)(1)

bit 2 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
If ADON = 1:
1 = A/D conversion in progress (setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)

bit 1 **Unimplemented:** Read as '0'

bit 0 **ADON:** A/D On bit
1 = A/D converter module is operating
0 = A/D converter module is shut-off and consumes no operating current

Note 1: These channels are not available on PIC16F873/876 devices.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.
0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.

bit 6-4 **Unimplemented:** Read as '0'

bit 3-0 **PCFG3:PCFG0**: A/D Port Configuration Control bits:

PCFG3: PCFG0	AN7(1) RE2	AN6(1) RE1	AN5(1) RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ Refs(2)
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

A = Analog input D = Digital I/O

Note 1: These channels are not available on PIC16F873/876 devices.
2: This column indicates the number of analog channels available as A/D inputs and the number of analog channels used as voltage reference inputs.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

The ADRESH:ADRESL registers contain the 10-bit result of the A/D conversion. When the A/D conversion is complete, the result is loaded into this A/D result register pair, the GO/$\overline{\text{DONE}}$ bit (ADCON0<2>) is cleared and the A/D interrupt flag bit ADIF is set. The block diagram of the A/D module is shown in Figure 11-1.

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs.

To determine sample time, see Section 11.1. After this acquisition time has elapsed, the A/D conversion can be started.

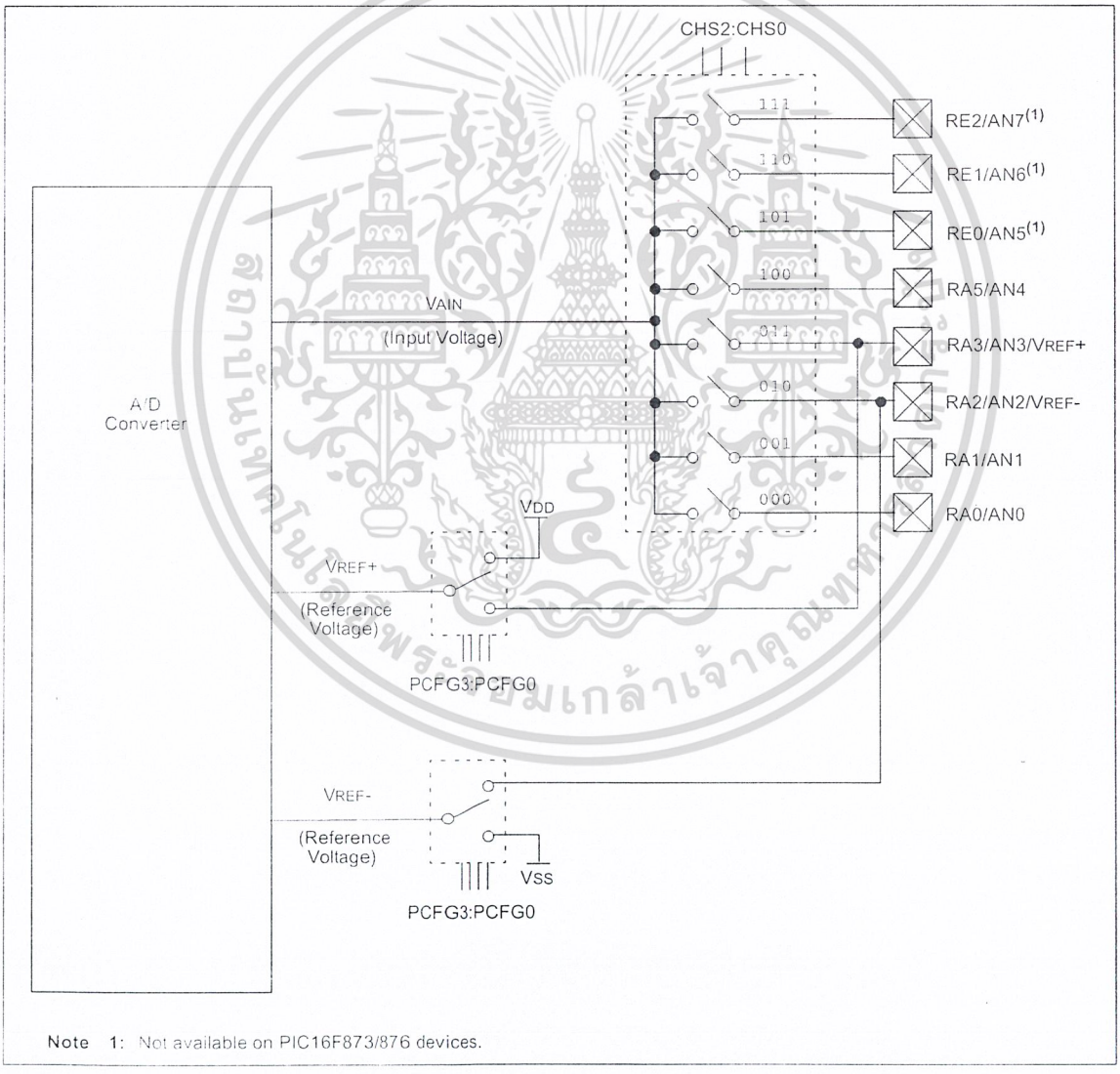
เอกสารนี้เผยแพร่เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

© 2001 Microchip Technology Inc.

These steps should be followed for doing an A/D Conversion:

1. Configure the A/D module:
   - Configure analog pins/voltage reference and digital I/O (ADCON1)
   - Select A/D input channel (ADCON0)
   - Select A/D conversion clock (ADCON0)
   - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
   - Clear ADIF bit
   - Set ADIE bit
   - Set PEIE bit
   - Set GIE bit
3. Wait the required acquisition time.
4. Start conversion:
   - Set GO/$\overline{\text{DONE}}$ bit (ADCON0)
5. Wait for A/D conversion to complete, by either:
   - Polling for the GO/$\overline{\text{DONE}}$ bit to be cleared (with interrupts enabled); OR
   - Waiting for the A/D interrupt
6. Read A/D result register pair (ADRESH:ADRESL), clear bit ADIF if required.
7. For the next conversion, go to step 1 or step 2, as required. The A/D conversion time per bit is defined as TAD. A minimum wait of 2TAD is required before the next acquisition starts.

FIGURE 11-1: A/D BLOCK DIAGRAM



Note 1: Not available on PIC16F873/876 devices.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.1 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 11-2. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), see Figure 11-2. **The maximum recommended impedance for analog sources is 10 kΩ.** As the impedance is decreased, the acquisition time may be decreased.

After the analog input channel is selected (changed), this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, Equation 11-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

To calculate the minimum acquisition time, TACQ, see the PICmicro™ Mid-Range Reference Manual (DS33023).

EQUATION 11-1: ACQUISITION TIME

TACQ = Amplifier Settling Time + Hold Capacitor Charging Time + Temperature Coefficient

= TAMP + TC + TCOFF
= 2µs + TC + [(Temperature -25°C)(0.05µs/°C)]
TC = CHOLD (RIC + RSS + RS) ln(1/2047)
= -120pF (1kΩ + 7kΩ + 10kΩ) ln(0.0004885)
= 16.47µs
TACQ = 2µs + 16.47µs + [(50°C -25°C)(0.05µs/°C)
= 19.72µs

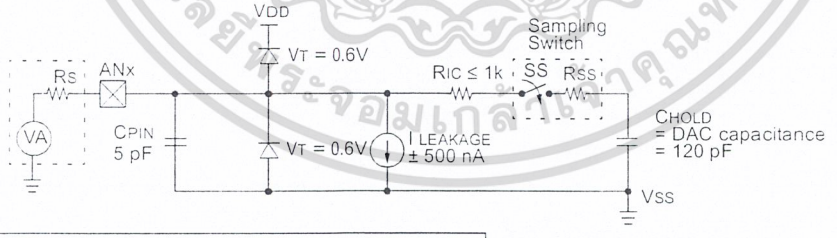
Note 1: The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

2: The charge holding capacitor (CHOLD) is not discharged after each conversion.

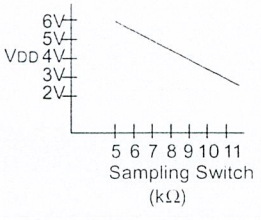
3: The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

4: After a conversion has completed, a 2.0TAD delay must complete before acquisition can begin again. During this time, the holding capacitor is not connected to the selected A/D input channel.

FIGURE 11-2: ANALOG INPUT MODEL



Legend
CPIN = input capacitance
VT = threshold voltage
I LEAKAGE = leakage current at the pin due to various junctions
RIC = interconnect resistance
SS = sampling switch
CHOLD = sample/hold capacitance (from DAC)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

© 2001 Microchip Technology Inc.

11.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires a minimum 12TAD per 10-bit conversion. The source of the A/D conversion clock is software selected. The four possible options for TAD are:

- 2TOSC
- 8TOSC
- 32TOSC
- Internal A/D module RC oscillator (2-6 µs)

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6 µs.

Table 11-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

TABLE 11-1: TAD vs. MAXIMUM DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (C))

AD Clock Source (TAD)		Maximum Device Frequency
Operation	ADCS1:ADCS0	Max.
2TOSC	00	1.25 MHz
8TOSC	01	5 MHz
32TOSC	10	20 MHz
RC(1, 2, 3)	11	(Note 1)

Note 1: The RC source has a typical TAD time of 4 µs, but can vary between 2-6 µs.
2: When the device frequencies are greater than 1 MHz, the RC A/D conversion clock source is only recommended for SLEEP operation.
3: For extended voltage devices (LC), please refer to the Electrical Characteristics (Sections 15.1 and 15.2).

11.3 Configuring Analog Port Pins

The ADCON1 and TRIS registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS2:CHS0 bits and the TRIS bits.

Note 1: When reading the port register, any pin configured as an analog input channel will read as cleared (a low level). Pins configured as digital inputs will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.

2: Analog levels on any pin that is defined as a digital input (including the AN7:AN0 pins), may cause the input buffer to consume current that is out of the device specifications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

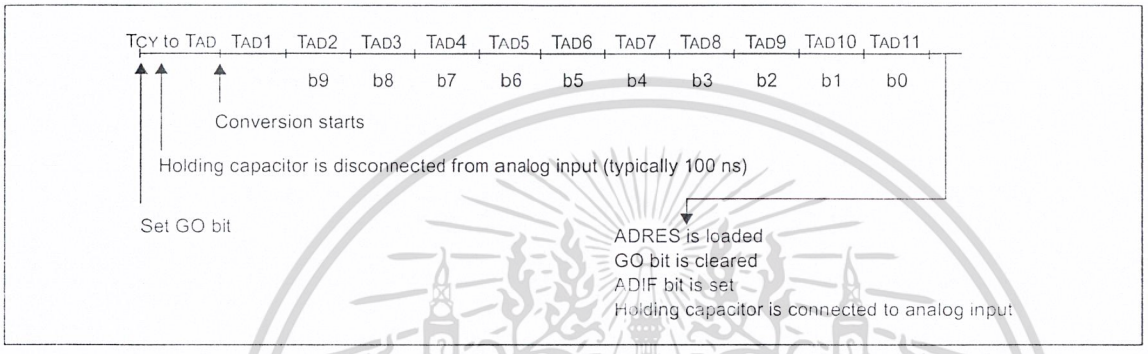
11.4 A/D Conversions

Clearing the GO/$\overline{\text{DONE}}$ bit during a conversion will abort the current conversion. The A/D result register pair will NOT be updated with the partially completed A/D conversion sample. That is, the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers). After the A/D conversion is aborted, a 2TAD wait is required before the next acquisition is started. After this 2TAD wait, acquisition on the selected channel is automatically started. The GO/$\overline{\text{DONE}}$ bit can then be set to start the conversion.

In Figure 11-3, after the GO bit is set, the first time segment has a minimum of TCY and a maximum of TAD.

Note:	The GO/$\overline{\text{DONE}}$ bit should **NOT** be set in the same instruction that turns on the A/D.

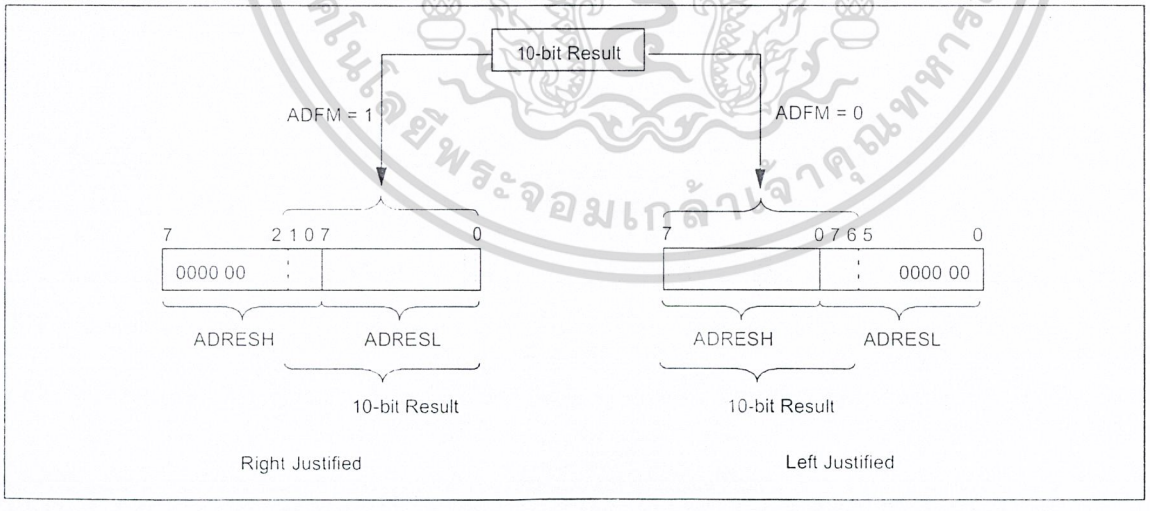
FIGURE 11-3: A/D CONVERSION TAD CYCLES



11.4.1 A/D RESULT REGISTERS

The ADRESH:ADRESL register pair is the location where the 10-bit A/D result is loaded at the completion of the A/D conversion. This register pair is 16-bits wide. The A/D module gives the flexibility to left or right justify the 10-bit result in the 16-bit result register. The A/D Format Select bit (ADFM) controls this justification. Figure 11-4 shows the operation of the A/D result justification. The extra bits are loaded with '0's'. When an A/D result will not overwrite these locations (A/D disable), these registers may be used as two general purpose 8-bit registers.

FIGURE 11-4: A/D RESULT JUSTIFICATION



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

© 2001 Microchip Technology Inc.

11.5 A/D Operation During SLEEP

The A/D module can operate during SLEEP mode. This requires that the A/D clock source be set to RC (ADCS1:ADCS0 = 11). When the RC clock source is selected, the A/D module waits one instruction cycle before starting the conversion. This allows the SLEEP instruction to be executed, which eliminates all digital switching noise from the conversion. When the conversion is completed, the GO/$\overline{\text{DONE}}$ bit will be cleared and the result loaded into the ADRES register. If the A/D interrupt is enabled, the device will wake-up from SLEEP. If the A/D interrupt is not enabled, the A/D module will then be turned off, although the ADON bit will remain set.

When the A/D clock source is another clock option (not RC), a SLEEP instruction will cause the present conversion to be aborted and the A/D module to be turned off, though the ADON bit will remain set.

Turning off the A/D places the A/D module in its lowest current consumption state.

Note:	For the A/D module to operate in SLEEP, the A/D clock source must be set to RC (ADCS1:ADCS0 = 11). To allow the conversion to occur during SLEEP, ensure the SLEEP instruction immediately follows the instruction that sets the GO/$\overline{\text{DONE}}$ bit.

11.6 Effects of a RESET

A device RESET forces all registers to their RESET state. This forces the A/D module to be turned off, and any conversion is aborted. All A/D input pins are configured as analog inputs.

The value that is in the ADRESH:ADRESL registers is not modified for a Power-on Reset. The ADRESH:ADRESL registers will contain unknown data after a Power-on Reset.

TABLE 11-2: REGISTERS/BITS ASSOCIATED WITH A/D

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on $\overline{\text{MCLR}}$, WDT
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON	0000 00-0	0000 00-0
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	--0u 0000
89h(1)	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction bits			0000 -111	0000 -111
09h(1)	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.
Note 1: These registers/bits are not available on the 28-pin devices.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สำเร็จลุล่วงได้ ด้วยคำแนะนำและคำปรึกษาจากท่าน รศ.ดร. วันชัย ริ้วรุจา ผู้ทำปริญญานิพนธ์รู้สึกทราบซึ้งในความอนุเคราะห์จากท่านและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณบิดา มารดา ครอบครัว และเพื่อนๆ ของข้าพเจ้าที่คอยให้กำลังใจมาโดยตลอด

ขอขอบคุณ นาย ธวัชชัย คำศรี ด้วยคำแนะนำและให้คำปรึกษาจากท่านทำให้ปริญญานิพนธ์เล่มนี้สำเร็จลุล่วง คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์นี้ ขอมอบแด่ผู้มีพระคุณทุกท่าน

นางสาว ธนัญธร ตรีรัตนกุล

นาย สรกิจ สรคุณพิพิธกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. " PHOTOCONDUCTIVE " EG&G optoelectronics Vactec Division
2. " PIC 16F87X datasheet " 2001 Microchip Technology Inc.
3. ดร. ธวัช เมฆสวรรค์ , "หนังสือเทคนิคการซ่อมเครื่องรับโทรทัศน์"
4. ณัฏฐพล วงศ์สุนทรชัย , ชัยวัฒน์ ลิ้มพรจิตรวิไล " แนะนำการใช้งานโปรแกรม MPLAB PIC -- MICRO development tool " , Index
5. ณัฏฐพล วงศ์สุนทรชัย , ชัยวัฒน์ ลิ้มพรจิตรวิไล "Akit – LCD คู่มือทดลองการใช้งานโมดูลแสดงผล LCD แบบอักขระ "

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้