

สำนักหอสมุดกอง พระจอมเกล้าลาดกระบัง

แบบจำลองโรงเพาะเลี้ยงพืชด้วยระบบคอมพิวเตอร์

Model of Computer control for Greenhouse



2/พ.
๑๖๒๖
๑๕๕๐

เลขหมู่.....**83078**
เลขทะเบียน.....
วัน,เดือน,ปี.....**- 5 ส.พ. 2551**

b. 11๑ 611๒๙
i.....

**ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Model of Computer control for Greenhouse

BY

MR. ANON BOONTARAM

MR. PROMLERE YENRUT AISIRI

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR THE DEGREE OF
BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	แบบจำลองโรงเพาะเลี้ยงพืชด้วยระบบคอมพิวเตอร์	
ชื่อนักศึกษา	นายอนนต์ บุญทาราม	รหัสประจำตัว 48015646
	นายพรหมเลิศ เข็นฤทัยศิริ	รหัสประจำตัว 48015648
อาจารย์ที่ปรึกษา	ผศ. อุทัย ศรีธีระวิโรจน์	
อาจารย์ที่ปรึกษาร่วม	ผศ.ดร.สมเกียรติ อุดมพระธนากุล	
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต	
	สาขาวิศวกรรมสารสนเทศ	
ภาควิชา	วิศวกรรมสารสนเทศ	
ปีการศึกษา	ปีการศึกษา 2550	

บทคัดย่อ

ในการทำโรงเรือนปลูกผักไร้ดิน ปังจี้ยต่างๆ ที่ส่งผลกระทบต่อการเจริญเติบโตของพืช อาทิ เช่น อุณหภูมิ ความชื้น แสงสว่าง และการระบายอากาศ เป็นต้น จำเป็นต้องมีการควบคุม เพื่อให้ผลผลิตออกมาได้ตามเป้าหมาย สามารถลดจำนวนผู้ดูแลโรงเพาะเลี้ยงและลดเวลาในการดูแลได้ ดังนั้นในโครงการนี้จึงได้เสนอการควบคุมปังจี้ยต่างๆ ดังที่กล่าวมาด้วยการควบคุมแบบวงจรถืออิเล็กทรอนิกส์และผ่านคอมพิวเตอร์ ในระบบที่ออกแบบนั้นสามารถตรวจจับอุณหภูมิและความชื้นภายในโรงเพาะปลูกพืชผ่านเครือข่าย และสามารถควบคุม แสงสว่าง พัดลมดูดอากาศ และการจำลองปั้งต่างๆ ให้สามารถนำไปใช้งานจริง สามารถควบคุมที่ตัวเครื่อง แสดงผลที่จอ LCD ได้ และควบคุมผ่านคอมพิวเตอร์พร้อมทั้ง แสดงผลบนหน้าจอกอมพิวเตอร์ ได้

Thesis Title Model of Computer control for Greenhouse
Student MR. ANON BOONTARAM ID. 48015646
MR. PROMLERE YENRUT AISIRI ID. 48015648
Advisor Asst. Prof. Uthai Sritheeravirojana
Asst. Prof. Dr. Somkait Udomhunsakul
Graduate Bachelor Degree of Information Engineering
Department Information Engineering
Academic Year 2007

ABSTRACT

In the making of soilless culture farmstead, there are several factors which cause effect to the plants, for example, temperature, humidity, light, and ventilation. Therefore, we have to control it in order to make output to the aimed target. It can decrease farmstead keepers also be managed in less time. This project presents the controlling factors of electronic control in the designed system of which temperature and humidity can be detected via internet work as well as controls the intensity of light, ventilation fan and imitates the pumps for the real situations to be displayed in LCD or computer monitor.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี ขอขอบพระคุณ ผศ. อุทัย ศรีธีระวิโรจน์ ผศ. ดร.สมเกียรติ อุดมธรรมากุล ที่ให้คำแนะนำและความช่วยเหลือที่ดีเสมอมาตลอดจนอีกทั้งยังได้ชี้แนะแนวทางในการแก้ปัญหาต่างๆในการทำงานวิจัยมาโดยตลอด

ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ที่คอยห่วงใยและให้การสนับสนุนในการศึกษา รวมทั้งขอขอบคุณญาติสนิทและพี่ๆทุกคนที่เป็นกำลังใจพร้อมทั้งให้ความช่วยเหลือในด้านต่างๆมาโดยตลอด

สุดท้ายขอขอบคุณเพื่อนๆห้อง 3F/3 เพื่อนชั้น 9 ทั้งต่อเนื่อง ภาคปกติและรับตรงและพี่เอ็ดส ที่คอยช่วยเหลือกันมาตลอด

นาย อนนต์ บุญทาราม
นาย พรหมเลิศ เข็นฤทัยศิริ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หัวข้อ	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	
สารบัญรูป	
สารบัญตาราง	
บทที่ 1 บทนำ	
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของโครงการ	2
1.4 ข้อมูลในการออกแบบ	2
1.5 ขั้นตอนและวิธีการดำเนินงาน	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	
2.1 โรงเรือนและพื้นฐานความต้องการของการปลูกพืชไร้ดิน	4
2.1.1 แนวความคิด	4
2.1.2 เหตุผลและทางเลือกในการลงทุนการปลูกพืชไร้ดิน	5
2.1.3 ไร้ดินแบบไฮโดรโพนิกส์	6
2.1.4 ไร้ดินแบบแอโร โพนิกส์	6
2.1.5 ประโยชน์ของการปลูกพืชโดยไม่ใช้ดิน	7
2.1.6 การปลูกพืชโดยไม่ใช้ดินกับการผลิตเชิงธุรกิจ	7
2.1.7 ความแตกต่างระหว่างการปลูกพืชบนดินตามธรรมชาติ กับการปลูกพืชโดยไม่ใช้ดิน	8
2.1.8 ปัญหาและแนวความคิดการปลูกพืชไร้ดินเชิงธุรกิจในประเทศไทย	9
2.2 ไมโครคอนโทรลเลอร์	13
2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51	14
2.2.1.1 คุณสมบัติของ P89C51RD2	14
2.2.2 ลักษณะการจัดการขามาคนอกของไมโครคอนโทรลเลอร์ MCS-51	16
2.2.3 ขาที่สำคัญของไมโครคอนโทรลเลอร์ MCS-51	16

สารบัญ (ต่อ)

หัวข้อ	หน้า
2.3 การอินเทอร์รัปต์ (Interrupt)	17
2.3.1 การจัดการอินเทอร์รัปต์ในไมโครคอนโทรลเลอร์ MCS-51	17
2.3.2 การเขียนโปรแกรมข้อยินเทอร์รัปต์	18
2.3.3 รีจิสเตอร์ที่เกี่ยวข้องกับการอินเทอร์รัปต์ในไมโครคอนโทรลเลอร์ MCS-51	18
2.3.4 แหล่งกำเนิดสัญญาณอินเทอร์รัปต์จากไมโครคอนโทรลเลอร์ MCS-51	20
2.3.4.1 สัญญาณอินเทอร์รัปต์จากภายนอก	20
2.3.4.2 การอินเทอร์รัปต์จากไทเมอร์/เคาน์เตอร์ 0 และ 1	20
2.3.4.3 การอินเทอร์รัปต์จากพอร์ตอนุกรม	21
2.3.4.4 การอินเทอร์รัปต์จากไทเมอร์/เคาน์เตอร์ 2	21
2.4 วงจร โซลิดสเตทรีเลย์ (Solid State Relay)	22
บทที่ 3 การออกแบบและสร้างระบบควบคุมการเลี้ยงพืชไร้ดิน	
3.1 ภาคตรวจจับอุณหภูมิความชื้นและภาคแสดงผล	23
3.1.1 การออกแบบตัวตรวจจับอุณหภูมิ	24
3.1.2 การออกแบบตัวตรวจจับความชื้น ความชื้น HIH-4000	25
3.1.3 การออกแบบวงจรภาคตรวจจับแสง	26
3.2 การออกแบบวงจรการเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล	27
3.3 การออกแบบวงจรภาครับการกดคีย์ข้อมูล	28
3.4 การออกแบบภาคประมวลผล	28
3.5 การออกแบบภาคเอาต์พุต	29
3.6 ออกแบบวงจรขับหลอด LED และ มอเตอร์	30
บทที่ 4 ผลการทดลอง	
4.1 การทดลองตรวจจับอุณหภูมิและความชื้น	31
4.2 การทดลองทางด้านโปรแกรม (Software)	34
4.3 การทดลองการกดปุ่มที่ตัวเครื่องและการแสดงผลที่จอ LCD	36
4.3.1 การทดลองการกดปุ่มที่ตัวเครื่อง	36

สารบัญ (ต่อ)

หัวข้อ	หน้า
4.3.2 การทดลองการแสดงผลที่จอ LCD	37
บทที่ 5 บทสรุปและวิจารณ์	
5.1 สรุปผลการดำเนินงาน	38
5.2 ปัญหาที่เกิดขึ้นและแนวทางแก้ไข	38
5.3 แนวทางพัฒนาในอนาคต	38
บรรณานุกรม	39
ภาคผนวก	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูป	หน้า
รูปที่ 2.1 ตารางการเปรียบเทียบการปลูกพืชด้วยดินและน้ำ	5
รูปที่ 2.2 แปลงปลูกพืชไฮโดร โพนิกส์	6
รูปที่ 2.3 โรงเพาะเลี้ยงพืชไฮโดร โพนิกส์	8
รูปที่ 2.4 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 เบอร์ P89C81RD2	15
รูปที่ 2.5 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51	16
รูปที่ 2.6 โซลิตเตอรีเลย์ (Solid State Relay)	22
รูปที่ 3.1 ภาพตรวจจับอุณหภูมิ, ความชื้น, แสดงผล และการสื่อสาร RS 232	23
รูปที่ 3.2 ตัวตรวจจับอุณหภูมิ	24
รูปที่ 3.3 ตัวตรวจจับความชื้นรูปที่ 3.4 วงจรภาคตรวจจับแสง	25
รูปที่ 3.4 วงจรภาคตรวจจับแสง	26
รูปที่ 3.5 ลักษณะของสัญญาณ	26
รูปที่ 3.6 บล็อกแสดงการแปลงสัญญาณอนาล็อกเป็นดิจิทัล	27
รูปที่ 3.7 การออกแบบวงจรภาครับการกดคีย์ข้อมูล	28
รูปที่ 3.8 ภาคประมวลผล	29
รูปที่ 3.9 ภาคเอาต์พุต	29
รูปที่ 3.10 วงจรขับ LED และ Motor	30
รูปที่ 4.1 ตู้ควบคุมอุณหภูมิและความชื้น (Versatile Environmental Test Chamber)	31
รูปที่ 4.2 รูปแบบจำลองโรงเพาะเลี้ยงพืช	32
รูปที่ 4.3 การทดลองทางด้านโปรแกรม (Software)	34
รูปที่ 4.4 การทดลองการกดปุ่มที่ตัวเครื่อง	36
รูปที่ 4.5 การทดลองการแสดงผลที่จอ LCD	37

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 ตัวอย่างของพืชที่สามารถปลูกโดยการปลูกพืชโดยไม่ใช้ดิน	10
ตารางที่ 2.2 เปรียบเทียบข้อดีและข้อเสียของการปลูกพืชโดยไม่ใช้ดิน	11
ตารางที่ 2.2(ต่อ) เปรียบเทียบข้อดีและข้อเสียของการปลูกพืชโดยไม่ใช้ดิน	12
ตารางที่ 4.1 การตรวจวัดอุณหภูมิ	33
ตารางที่ 4.2 การตรวจวัดความชื้น	33
ตารางที่ 4.3 การทดสอบกลุ่มต่างๆทางด้านโปรแกรม (Software)	35
ตารางที่ 4.4 การทดสอบปุ่มกดที่กดต้องควบคุม	36
ตารางที่ 4.5 การแสดงผลที่จอ LCD	37

บทที่ 1

บทนำ

1.1 ความเป็นมา

ในปัจจุบันนี้มนุษย์มีการใส่ใจสุขภาพมากขึ้นจะเห็นได้ว่า มีการบริโภคอาหารที่มีประโยชน์ต่อสุขภาพมากขึ้น และหันมาสนใจเกี่ยวกับการบริโภคพืชผักที่ปลอดสารพิษ จากในอดีตถึงปัจจุบันยังมีการปลูกผักในดินที่ใช้สารเคมีในการปราบศัตรูพืชซึ่งให้ผลผลิตการเกษตรที่ต้องใช้เวลาในการเจริญเติบโตของพืชช้ากว่าการปลูกไร้ดิน และมีสารพิษปนเปื้อน จึงได้หันมาศึกษาและปลูกพืชไร้ดินแบบปลอดสารพิษอยู่ในโรงเรือนเพื่อป้องกันแมลง ทำให้ไม่ต้องใช้สารเคมีในการฆ่าแมลง แต่การเพาะปลูกแบบนี้ยังไม่แพร่หลายมากนักเพราะว่างบประมาณในการลงทุนทำโรงเรือนยังมีราคาที่สูงอยู่ แต่การปลูกพืชไร้ดินนั้นมีข้อดีอยู่มาก ทั้งการประหยัดเวลาในการปลูก ประหยัดสารเคมีที่ใช้เร่งการเจริญเติบโต ไม่มีสารพิษที่มาจากสารเคมีในการปราบศัตรูพืช

การจัดการและการควบคุมโรงเรือนจึงมีความสำคัญมาก เพราะมีผลกระทบต่อผลผลิตของพืชในทางด้านอุณหภูมิความชื้นการให้แสงสว่าง การระบายความร้อนด้วยพัดลมดูดอากาศและเพิ่มความชื้นด้วยการพ่นละอองน้ำ แต่ที่ใช้กันอยู่ต้องควบคุมอุปกรณ์เหล่านี้ด้วยคนที่ดูแลโรงเรือนและต้องอยู่ที่โรงเรือนด้วย จึงเกิดแนวความคิดที่จะนำระบบอิเล็กทรอนิกส์มาใช้ในการควบคุมการทำงานของระบบ โรงเรือนได้ทั้งอยู่ที่โรงเรือนมีการแสดงผลทั้งอุณหภูมิและความชื้นที่ตัวเครื่องได้และอยู่ในระยะไกลได้โดยการควบคุมผ่านคอมพิวเตอร์และแสดงผลบนหน้าจอได้ จึงทำให้มีความสะดวกสบายมากขึ้นและสามารถควบคุมได้แม่นยำ

นับว่าเป็นเครื่องมือที่มีประโยชน์อย่างยิ่ง หากมีการส่งเสริมไปใช้ในการเกษตรสำหรับโรงเรือนปลูกพืชไร้ดิน เพราะจะทำให้มีความสะดวกมากขึ้น ลดจำนวนผู้ดูแลโรงเรือน ควบคุมให้มีผลผลิตที่ดีขึ้น ทำให้การปลูกพืชไร้ดินที่มีอยู่มีประสิทธิภาพที่ดีขึ้นมีผลผลิตมากขึ้น และส่งผลให้คุณภาพของผลผลิตได้มาตรฐานในระดับที่ตลาดต้องการ

1.2 วัตถุประสงค์

1. เพื่อศึกษาการเขียนโปรแกรมของไมโครคอนโทรลเลอร์
2. เพื่อศึกษาการควบคุมผ่านคอมพิวเตอร์
3. เพื่อประยุกต์การใช้งานของคอมพิวเตอร์ให้สามารถใช้งานร่วมกับไมโครคอนโทรลเลอร์

1.3 ขอบเขตของโครงการ

เป็นการนำไมโครคอนโทรลเลอร์มาใช้ในการควบคุมหลอดไฟฟ้า พัดลม หลอด LED เพื่อแสดงสถานะ แสดงผลอุณหภูมิและความชื้น สามารถควบคุมที่ตัวไมโครคอนโทรลเลอร์แสดงผลค่าอุณหภูมิและความชื้นออกที่จอ LCD ที่ตัวไมโครคอนโทรลเลอร์ได้ และในส่วนของ การเขียนโปรแกรมสามารถควบคุมด้วยระบบคอมพิวเตอร์ไปควบคุมหลอดไฟฟ้า พัดลม หลอด LED เพื่อแสดงสถานะ ที่อยู่ที่ตัวไมโครคอนโทรลเลอร์ได้ และยังสามารถรับค่าอุณหภูมิจากตัวตรวจจับอุณหภูมิและความชื้น (LM35 และ HIH-4000-003) มาแสดงผลที่จอคอมพิวเตอร์ได้ โดยการจำลองรูปแบบโรงเรือนเป็นแบบจำลอง

1.4 ข้อมูลในการออกแบบ

- สามารถแสดงผลอุณหภูมิที่ LCD และจอคอมพิวเตอร์ได้
- สามารถแสดงผลความชื้นที่ LCD และจอคอมพิวเตอร์ได้
- สามารถควบคุม หลอดไฟ หลอด LED พัดลมที่แบบจำลองได้ จากการควบคุมที่ตัวไมโครคอนโทรลเลอร์
- สามารถควบคุม หลอดไฟ หลอด LED พัดลมที่แบบจำลองได้ การควบคุมด้วยระบบคอมพิวเตอร์ได้

ในแบบจำลองโรงเรือนเลี้ยงพืชไร้ดินนี้จะจำลองสถานะของ ป้อนน้ำ สถานะปุ๋ย สถานะน้ำ ด้วย หลอด LED พัดลมใช้พัดลม AC 220 V เพื่อแสดงการระบายอากาศ หลอดไฟที่ใช้จำลอง เพื่อให้ความชื้นลดลงจะเป็นหลอดไฟ AC 220 V เพื่อจะให้นำไปประยุกต์การใช้งานจริงได้ และเห็นการทำงานในแบบจำลองได้ชัดเจนมากยิ่งขึ้น

1.5 ขั้นตอนและวิธีการดำเนินงาน

1. ศึกษาและหาข้อมูลจากแหล่งต่างๆที่เกี่ยวข้อง
2. กำหนดขอบเขตงานให้กระชับและตรงตามจุดประสงค์มากที่สุด
3. ศึกษาการเขียนโปรแกรมไมโครคอนโทรลเลอร์และการเขียนโปรแกรม Visual Studio C#. NET
4. ศึกษาการติดต่อตัวตรวจจับอุณหภูมิและความชื้นและการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์
5. ศึกษาลักษณะการใช้อุปกรณ์ต่างๆและโปรแกรมที่จำเป็นต้องใช้กับฮาร์ดแวร์
6. ออกแบบวงจรหลักและวงจรควบคุม
7. นำฮาร์ดแวร์มาทดลองเพื่อวิเคราะห์หาข้อผิดพลาด
8. นำซอฟต์แวร์มาทดลองใช้งานเพื่อหาเปอร์เซ็นต์ความผิดพลาดของระบบ
9. แก้ไขและปรับปรุงระบบให้มีความผิดพลาดที่น้อยที่สุด
10. บันทึกและสรุปผลการทดลอง

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

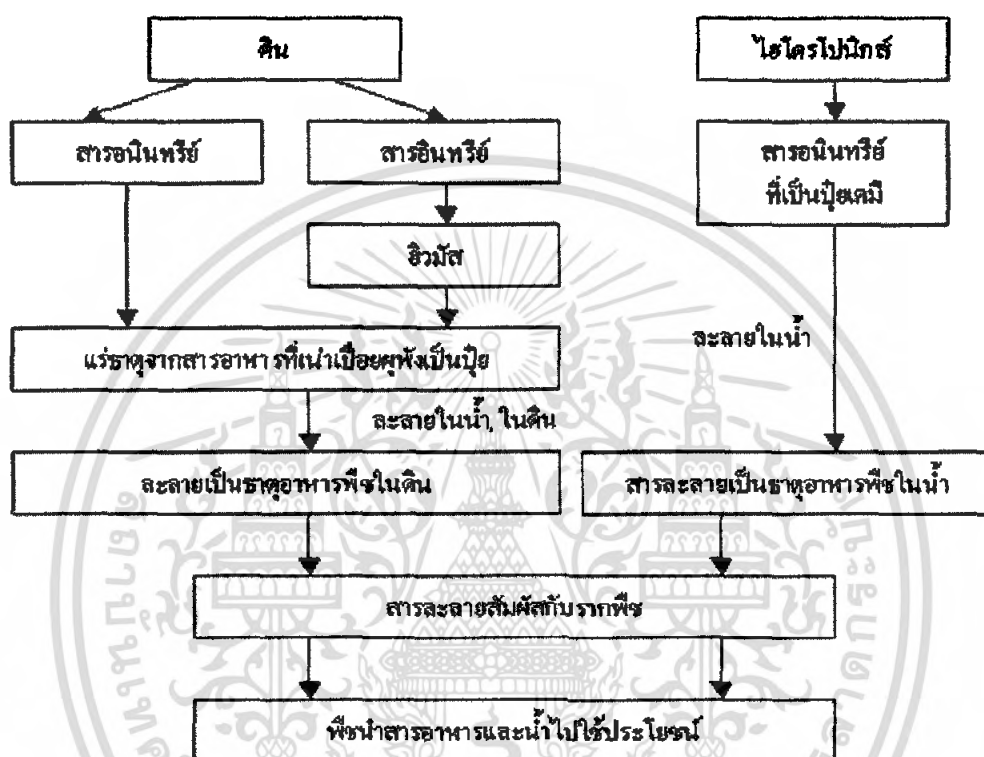
2.1 โรงเรือนและพื้นฐานความต้องการของการปลูกพืชไร้ดิน

จากปัญหาที่เกษตรกรประสบในการเพาะปลูกในประเทศไทย ทั้งปัญหาความแห้งแล้ง โรคพืชแมลงศัตรูพืชระบาด น้ำท่วม คุณภาพของผลิตผล และที่สำคัญที่สุดปัญหาการใช้สารพิษ เพื่อกำจัดแมลง ส่งผลทำให้ดินและระบบนิเวศน์ได้รับผลกระทบอย่างมาก รวมทั้งอันตรายที่เกิดจากสารพิษต่อทั้งเกษตรกรผู้ปลูกเอง และประชาชนผู้บริโภค ปัญหาของเกษตรกรที่เกิดขึ้นนี้ยังได้รับการดูแลแก้ไขน้อยมาก เนื่องจากการที่เกษตรกรขาดความรู้ งบประมาณ และแรงจูงใจ ซึ่งปัจจุบันรูปแบบการทำการเกษตรในโลกนี้ได้พัฒนาไปมากแล้ว ทั้งนี้มีสาเหตุมาจากการพัฒนาและปลูกฝังให้เยาวชนของชาติมีทัศนคติที่ดีต่ออาชีพเกษตรกรรม โดยใช้เทคโนโลยีการเกษตร และการจัดการฟาร์มในรูปแบบใหม่ๆ รวมถึงการสนับสนุนจากภาครัฐฯ ทั้งนี้เป็นที่ทราบกันด้วยปัจจัยหลายอย่าง เช่น สภาพของภูมิประเทศ ภูมิอากาศ ความวิริยะอุตสาหะ ความอดทน และที่สำคัญประชากรส่วนใหญ่ของประเทศประกอบอาชีพทางเกษตรกรรมอยู่แล้ว ความได้เปรียบด้านต่างๆ เหล่านี้ถ้าได้รับการส่งเสริมอย่างต่อเนื่องและมีเทคโนโลยีการผลิตที่มีประสิทธิภาพ เชื่อว่าต่อไปประเทศไทยจะเป็นแหล่งผลิตอาหารที่ได้คุณภาพที่สำคัญของโลกในไม่ช้า

2.1.1 แนวความคิด

การปลูกพืชผักแผนใหม่ โดยไม่ใช้ดินจนประสบความสำเร็จ และเป็นที่ยอมรับกันโดยทั่วไปได้พัฒนารูปแบบ ตลอดจนอุปกรณ์ที่ใช้ในการปลูกให้เหมาะสมกับสภาพภูมิอากาศ ลักษณะนิสัยการบริโภคการค้ำค่าในเชิงพาณิชย์ และความปลอดภัยของผู้บริโภค และที่สำคัญเทคโนโลยี อุปกรณ์ทั้งหมดผลิตขึ้นเองภายในประเทศ จึงทำให้ไม่ต้องพึ่งพาเทคโนโลยีจากต่างประเทศทำให้การลงทุนถูกลงมาก และสามารถแข่งขันได้กับต่างชาติ อีกทั้งยังสามารถปลูกผักได้ทุกชนิด และทุกสภาพอากาศทุกพื้นที่การเกษตร ขั้นตอนการปลูกก็ไม่ยุ่งยาก ดูแลรักษาง่าย ได้ผลผลิตที่แน่นอน และสูงกว่าการปลูกแบบไร้ดินของต่างประเทศ ผลผลิตที่ได้จะปราศจากสารพิษ 100% ปัจจุบันผู้บริโภคได้ตื่นตัวอย่างมาก เพราะเป็นที่ทราบกันดีว่า การปลูกพืชแบบดั้งเดิมไม่อาจหลีกเลี่ยงสารพิษที่เป็นอันตรายต่อร่างกายได้เพราะการระบาดของศัตรูพืช และการคือยาของศัตรูพืช ซึ่งได้ผลผลิตที่ปลอดสารพิษ เพื่อให้ทันกับความก้าวหน้าของเทคโนโลยีการผลิตพืชแผนใหม่ และพัฒนาเทคนิคใหม่ในการปลูกพืชไร้ดิน จนประสบความสำเร็จ กว้างขวางในวงการ

เกษตรแผนใหม่ และมีความพร้อมอย่างมากที่จะเผยแพร่เทคโนโลยีการเกษตรแผนใหม่เพื่อให้รองรับความต้องการของผู้บริโภคให้ทันและยกระดับความรู้ให้แก่เกษตรกรเพื่อความก้าวหน้าของวงการเกษตรประเทศไทยต่อไป



รูปที่ 2.1 ตารางการเปรียบเทียบการปลูกพืชด้วยดินและน้ำ

2.1.2 เหตุผลและทางเลือกในการลงทุนการปลูกพืชไร้ดิน

ด้วยความสนใจในความเป็นไปของวงการเกษตรทั้งเรื่องของผลผลิต วิธีการผลิต การจัดจำหน่าย การจัดการการผลิตและเทคโนโลยีการผลิตของประเทศไทย พบว่ามีหลายสิ่งหลายอย่างที่ยังไม่สมบูรณ์ยังขาดการพัฒนา หรือแนวคิดอีกมาก เพื่อรองรับกับความต้องการของผู้บริโภคอีกกลุ่มหนึ่ง หรือการใช้ปุ๋ย และสารกำจัดศัตรูพืชในประเทศที่สูงกว่าอีกหลายประเทศมากในขณะที่ผลผลิตที่ได้น้อยกว่ากันหลายเท่าตัวแม้แต่การใช้พื้นที่ หรือแรงงานในการเกษตรของบ้านเราก็ไม่คุ้มค่า

จากปัญหาหลายสิ่งหลายอย่างที่ข้าพเจ้าพบเห็น ทำให้เกิดมีแนวคิดว่าจะทดลองทำการปลูกพืชผักในวิธีการที่น่าจะเป็นไปได้ทั้งเรื่องของการจัดการ และใช้เทคโนโลยีที่การเกษตรแผนใหม่มา เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผสมผสาน โดยเริ่มจากการสำรวจข้อมูลด้านการตลาดทั้งในส่วนของผู้บริโภค ที่ได้จากการสำรวจพบว่ายังมีช่องทางที่จะทำตลาดได้อีกไม่น้อย

ส่วนที่นิยมปลูกกันมากที่สุด คือ การปลูกในน้ำ เรียกว่า ไฮโดร โพนิกส์ (Hydroponics) การปลูกพืชไร้ดินที่นิยมทำกันอยู่ขณะนี้ 2 ประเภท คือ แบบไฮโดร โพนิกส์ และแอโร โพนิกส์

2.1.3 ไร้ดินแบบไฮโดร โพนิกส์

ไฮโดร โพนิกส์ เป็นการปลูกพืชไร้ดิน ในรูปแบบของการปลูกพืชให้รากพืชแช่อยู่ในน้ำ หรือสารละลายธาตุอาหารพืช ไฮโดร โพนิกส์ เข้ามามีบทบาทเพื่อแก้ปัญหาของการปลูกพืชในดิน ซึ่งมีการใช้สารเคมีกำจัดศัตรูพืชและวัชพืช รวมทั้งเชื้อโรคพืชที่อาศัยอยู่ในดินทำให้เกิดสารพิษตกค้างในผลผลิตเป็นอันตรายต่อผู้บริโภคและตัวเกษตรกรผู้ปลูก นอกจากนี้ปลูกพืชในดินยังต้องใช้น้ำมาก ถ้าปราศจากแหล่งน้ำก็ก่อให้เกิดปัญหาในการเพราะปลูกอีกการปลูกพืชในดินต้องมีการเตรียมดิน ปรับสภาพดิน และต้องใช้ปุ๋ยเคมีสูตรต่าง ๆ ตามอายุพืช “ไฮโดร โพนิกส์” จึงเป็นระบบการปลูกพืชที่เข้ามาแก้ปัญหาดังกล่าว แต่ทั้งนี้ ไฮโดร โพนิกส์ก็เหมาะสมสำหรับพืชบางชนิดเท่านั้น



รูปที่ 2.2 แปลงปลูกพืชไฮโดร โพนิกส์

2.1.4 ไร้ดินแบบแอโร โพนิกส์

การปลูกพืชในระบบแอโร โพนิกส์ คล้าย ๆ กับไฮโดร โพนิกส์ แต่แทนที่รากพืชจะแช่อยู่ในน้ำยาซึ่งเป็นสารละลายธาตุอาหารพืช ก็ใช้วิธีการฉีดพ่นสารละลายธาตุอาหารพืชให้แก่พืชทางรากเป็นระยะ ๆ ตลอด 24 ชั่วโมงแทน วิธีการนี้พืชได้อาหารครบถ้วนและพอเพียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่การปลูกด้วยระบบแอโร โพนิกส์ ต้องใช้ระบบควบคุมการฉีดพ่นธาตุอาหารแบบอัตโนมัติ เช่นฉีดพ่นทุก ๆ 1 นาที ระยะเวลาฉีดพ่น1 นาที หยุด 1 นาที เป็นต้น วิธีการนี้ใช้น้ำน้อยมาก การปลูกพืชแบบแอโร โพนิกส์นี้ ความชื้นจากการฉีดพ่นสารละลายธาตุอาหารจะไปกระตุ้นให้รากพืชเจริญเติบโตอย่างสมบูรณ์ภายใน 10 วัน และต้นพืชโดยเฉพาะพืชผักสามารถเจริญเติบโตเก็บเกี่ยวได้ภายในระยะเวลาเพียง30วัน

2.1.5 ประโยชน์ของการปลูกพืชโดยไม่ใช้ดิน

การปลูกพืชแบบไม่ใช้ดินเป็นวิทยาการทางวิทยาศาสตร์และศิลปะผสมกันที่สามารถใช้ปลูกพืชได้ในทุกสถานที่โดยไม่มีขอบเขตจำกัด ไม่ว่าจะเป็นการปลูกจำนวนน้อยเพื่อบริโภคในครัวเรือนหรือการผลิตเชิงธุรกิจ เป็นวิธีที่เหมาะสมกับความต้องการสำหรับผู้ปลูกที่มีพื้นที่ปลูกน้อย เช่น แฟลต อพาร์ทเมนต์ จึงสามารถปลูกได้ในเมืองหลวงของเมืองที่แออัดคับแคบด้วยผู้คน เช่น ประเทศญี่ปุ่น ไต้หวัน เนเธอร์แลนด์ อังกฤษ เบลเยียม

สำหรับการปลูกแบบขนาดเล็กๆ เพื่อปลูกไว้ดูเล่นแบบสันทนาการ และมีอาหารจากการปลูกเพื่อบริโภคในครอบครัวจะ ไม่มีความยุ่งยากในการปลูก และดูแลรักษาเปรียบเหมือนกับการทำสวนตามปกติที่ให้ความเพลิดเพลิน การเรียนรู้เบื้องต้นในการปลูกพืช แต่ถ้าเป็นการปลูกแบบเชิงการค้าจะต้องมีการใช้เทคนิค หลักการต่างๆ ในการควบคุมการผลิตมากยิ่งขึ้น

2.1.6 การปลูกพืชโดยไม่ใช้ดินกับการผลิตเชิงธุรกิจ

วิธีการปลูกพืชโดยไม่ใช้ดินสามารถใช้ปลูกพืชได้หลายชนิด ทั้งนี้ขึ้นอยู่กับความยากง่ายของการปลูกพืชแต่ละชนิด ตั้งแต่ผัก ผลไม้ ไม้ดอก ไม้ประดับ พืชไม้เถา จนถึงพืชยืนต้น แต่การผลิตเชิงธุรกิจส่วนมากนิยมปลูกพวกพืชผัก ไม้ผลที่เป็นพืชที่เก็บเกี่ยวช่วงอายุสั้น ดังตัวอย่างที่แสดงให้เห็นในตารางต่อไปนี้



รูปที่ 2.3 โรงเพาะเลี้ยงพืชไฮโดรโปนิคส์

2.1.7 ความแตกต่างระหว่างการปลูกพืชบนดินตามธรรมชาติ กับการปลูกพืชโดยไม่ใช้ดิน

ปกติแล้วพืชจะเจริญเติบโตได้คือนั้น ต้องมีการเจริญเติบโตที่เหมาะสมคือ สภาพภูมิอากาศที่เหมาะสมเช่น แสงแดด อุณหภูมิ น้ำ ธาตุอาหารพืชที่มาจากดิน น้ำ และอากาศ (ออกซิเจน ไฮโดรเจน และ คาร์บอนไดออกไซด์) ทั้งที่รากส่วนเหนือดิน ซึ่งการที่พืชจะนำธาตุอาหารไปใช้ได้ นั้นจะเกี่ยวข้องกับความเป็นกรด-ด่าง (pH) ของดินการปลูกพืชบนดินโดยทั่วไปแม้ดินจะมีธาตุอาหาร และอากาศอันเป็นปัจจัยที่พืชต้องการนั้นมัก มีข้อเสีย คือ ดินจะไม่มีควบคุมสมบูรณ์ตามที่พืชต้องการ กล่าวคือ ดินจะมีคุณสมบัติที่ไม่แน่นอนแตกต่างกันไปตามสภาพพื้นที่ เช่น โครงสร้างของดิน ปริมาณธาตุอาหารหรือควบคุมสมบูรณ์ค่า pH ไม่เหมาะสม ยุ่งยากต่อการปรับปรุง และเสียค่าใช้จ่ายสูง ปัญหาเหล่านี้ทำให้ได้ผลผลิตที่ไม่แน่นอน

ส่วนการปลูกพืชโดยไม่ใช้ดินนั้นพืชจะได้รับสารละลายที่มีธาตุอาหารที่เรียกว่า สารละลายธาตุอาหารพืชที่ประกอบด้วยธาตุอาหารที่จำเป็นต่อพืช ที่อยู่ในรูปที่พืชสามารถนำไปใช้ได้ทันทีเพราะมีการปรับค่า EC และ pH ให้อยู่ในระดับที่เหมาะสมต่อพืชอยู่ตลอดเวลา อันที่จริงแล้วไม่มีความแตกต่างทางสรีระวิทยาของรากพืชที่จะนำธาตุอาหารทั้งจากการปลูกพืชบนดินตามธรรมชาติ หรือจากการปลูกพืชที่ไม่ใช้ดิน

ด้านการผลิตจากการศึกษาสภาพการผลิต ผลที่ได้รับไม่เป็นที่น่าพอใจกล่าวคือ การผลิตยังมีปัญหา และอุปสรรคอีกมากรวมถึงผลกระทบจากธรรมชาติ เช่น ปริมาณน้ำฝนที่มาก และน้อยเกินไป ปัญหาเรื่องคุณภาพผลผลิต และปัญหาทางด้านแรงงานรวมถึงต้นทุนด้านปุ๋ย และสารกำจัดศัตรูพืช

ด้านเทคโนโลยีปัจจุบันเทคโนโลยีที่ใช้ในการเกษตรแผนใหม่ในประเทศเป็นเทคโนโลยีจากต่างประเทศทั้งสิ้น รวมถึงอุปกรณ์ที่ใช้ และธาตุอาหารต่างๆ ที่จำเป็นต้องใช้ในการผลิตอีกทั้งยังไม่สามารถใช้ได้กับพืชผักที่นิยมบริโภคในประเทศ คือ สามารถปลูกได้แต่ผักสลัด เช่น RED OAK, GREEN OAK, COS และ BUTTER HEAD เป็นต้น ทำให้มีปัญหาทางด้านตลาดเพราะผู้บริโภคส่วนใหญ่ไม่ยอมรับประทาน

2.1.8 ปัญหาและแนวความคิดการปลูกพืชไร้ดินเชิงธุรกิจในประเทศไทย

ในปัจจุบันถ้าพิจารณาจากข้อมูล และสภาพความเป็นจริงแล้วสามารถกล่าวได้ว่าการปลูกพืชไร้ดินเชิงการค้าของประเทศไทยยังอยู่ในช่วงเริ่มต้นเท่านั้น เนื่องจากปัญหาและอุปสรรคต่างๆ รวมทั้งขาดข้อมูลทั้งด้านการผลิต การตลาด เทคโนโลยี และที่สำคัญที่สุดคือการบริหารจัดการที่ดี ทำให้การปลูกพืชไร้ดินในประเทศไม่มีความก้าวหน้าเท่าที่ควร ดังนั้น การที่จะพิจารณาหาทางผลิตพืชไร้ดินนั้นควรจะต้องทราบถึงแนวคิด หลักการ และแนวทางปฏิบัติให้ได้ เพื่อให้ประสบผลสำเร็จนั้นผู้ผลิตจะต้องเข้าใจถึงองค์ประกอบที่สำคัญๆ เช่น วัสดุอุปกรณ์ที่ใช้ สารอาหาร พันธุ์พืช ฯลฯ

ปัญหาการปลูกพืชไร้ดินในประเทศไทย สรุปได้ดังนี้

1. ขาดแหล่งข้อมูล และองค์ความรู้ที่จะค้นคว้า
2. ตลาดยังขาดความเข้าใจในผลผลิต
3. ขาดวัตถุดิบที่เหมาะสมในการผลิต
4. สภาพอากาศไม่เหมาะสม
5. ขาดการส่งเสริมจากภาครัฐอย่างจริงจัง
6. ขาดความรู้ความสามารถในการจัดการที่เหมาะสม

ปัญหานี้จัดว่าเป็นปัญหาใหญ่ และเกี่ยวข้องกับบุคลากรเนื่องจากการปลูกพืชไร่นาจะต้องเกี่ยวข้องกับโลกของปุ๋ยเคมี ดังนั้นการเรียนรู้นิเวศของปุ๋ย การเลือกใช้ การคำนวณ การผสมปุ๋ย รวมทั้งการควบคุมค่าต่างๆ ของปุ๋ยเป็นสิ่งที่หลีกเลี่ยงไม่ได้ ความรู้เกี่ยวกับดินพืช การเพาะกล้า การดูแล การจัดการศัตรูพืช การบำรุงรักษาระบบปลูก การเก็บผลผลิตให้ได้ตรงตามมาตรฐานของลูกค้า

ความรู้พื้นฐานเชิงช่างเกี่ยวกับการสร้างระบบปลูก ระบบการให้ปุ๋ย ระบบไฟฟ้า และวัสดุอุปกรณ์ที่เกี่ยวข้องกับการปลูก ความรู้เกี่ยวกับการตลาด เช่นการวิเคราะห์การตัดสินใจเลือกผลิตพืช กำหนดกลยุทธ์ในการผลิต การกำหนดราคาจำหน่าย การให้ความรู้แก่ผู้บริโภค และการเข้าถึงแหล่งจัดจำหน่ายซึ่งเป็นงานที่ทำทยละเอียดลออต้องอาศัยการฝึกฝนปฏิบัติอย่างจริงจังจึงจะประสบความสำเร็จได้

ตารางที่ 2.1 ตัวอย่างของพืชที่สามารถปลูกโดยการปลูกพืชโดยไม่ใช้ดิน

พืชผัก	ไม้ผล / ผักรับประทานผล	ไม้ดอก	พืชสมุนไพร	พืชอาหารสัตว์
มะเขือเทศ	ส้ม สตอร์เบอร์รี่	กุหลาบ	ว่านหางจระเข้	หญ้า
ผักกาดขาว	กล้วย แดงกวา	คาร์เนชั่น	พืชสวนครัว ต่างๆ	บาร์เล่ย์
คื่นฉ่าย	แดงแคนตาลูป ถั่วฝักยาว	เบญจมาศ		ข้าว โปดเลี้ยงสัตว์
ผักชี	พริก มะเขือ			

ตารางที่ 2.2 เปรียบเทียบข้อดีและข้อเสียของการปลูกพืชโดยไม่ใช้ดิน

ข้อดี	ข้อเสีย
<p>1. สามารถทำการเพาะปลูกพืชได้ในบริเวณพื้นที่ที่ดินไม่ดีหรือสภาพแวดล้อมไม่เหมาะสมต่อการเพาะปลูก</p> <p>2. ให้ผลผลิตต่อพื้นที่ปลูกสูงกว่า และสามารถทำการผลิตได้สม่ำเสมอ และต่อเนื่อง</p> <p>3. อัตราการใช้แรงงานเวลาในการปลูก และค่าใช้จ่ายต่ำกว่า</p> <p>4. ใช้น้ำ และธาตุอาหารได้อย่างประหยัด และมีประสิทธิภาพ เช่น ใช้น้ำลดลงถึง 10 เท่าตัวของการปลูกแบบธรรมดา</p> <p>5. ประหยัดเวลา และแรงงาน ในการเตรียมดิน และกำจัดวัชพืช</p> <p>6. ลดค่าใช้จ่ายที่เกี่ยวกับการใช้สารป้องกันและกำจัดแมลงได้ 100%</p>	<p>1. การลงทุนขั้นต้นสูงกว่าการปลูกบนดิน</p> <p>2. ผู้ปลูกต้องมีความรู้ความเข้าใจในเทคนิคการปลูกพืชแบบไร้ดินเป็นอย่างดี และมีประสบการณ์มากพอในการควบคุมดูแล</p> <p>3. ต้องการการควบคุมดูแลอย่างสม่ำเสมอ</p> <p>4. เป็นสิ่งใหม่สำหรับเกษตรกรที่ต้องใช้เวลาในการทำความเข้าใจ</p>

ตารางที่ 2.2 (ต่อ) เปรียบเทียบข้อดีและข้อเสียของการปลูกพืชโดยไม่ใช้ดิน

ข้อดี	ข้อเสีย
7. สามารถปลูกได้ในเมืองเพราะใช้พื้นที่น้อยทำให้ประหยัดค่าขนส่ง	
8. ผลผลิตมีคุณภาพ และไม่มีสารพิษตกค้าง และไม่มีปัญหาเกี่ยวกับศัตรูพืชที่เกิดจากดิน	
9. ผลผลิต คุณภาพ และราคา ดีกว่าการปลูกบนดินมาก เพราะสามารถควบคุมสภาพแวดล้อมต่างๆ ที่เกี่ยวข้องกับการเจริญเติบโตของพืชได้อย่างถูกต้องแน่นอนและรวดเร็ว	
10. ลดผลกระทบต่อสิ่งแวดล้อมด้านต่างๆ เช่น สารเคมีตกค้างในดิน การบุกรุกทำลายป่า เป็นต้น	
11. คนพิการก็สามารถทำการปลูกได้ เป็นการส่งเสริมอาชีพให้กับผู้ด้อยโอกาส	
12. เป็นทางเลือกให้กับเกษตรกรรุ่นใหม่	

2.2 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นอุปกรณ์ไอซี (IC: Integrated Circuit) ที่สามารถโปรแกรมการทำงานได้หลายครั้ง รับข้อมูลในรูปแบบสัญญาณดิจิทัลเข้าไปทำการประมวลผล แล้วส่งผลลัพธ์ข้อมูลดิจิทัลออกมาเพื่อนำไปใช้งานตามที่ต้องการได้

ไมโครคอนโทรลเลอร์หรืออาจจะเรียกอีกอย่างได้ว่าไมโครโพรเซสเซอร์ชิปเดี่ยว (Single-Chip Microprocessor) ซึ่งเป็นไมโครโพรเซสเซอร์ชนิดหนึ่ง เช่นเดียวกับหน่วยประมวลผลกลาง (CPU : Central Processing Unit) ที่ใช้ในคอมพิวเตอร์แต่ได้รับการพัฒนาแยกออกมาในภายหลัง เพื่อนำไปใช้ในวงจรด้านงานควบคุม คือ แทนที่ในการใช้งานจะต้องวางจอยักษ์ต่างๆเพิ่มเติม เช่นเดียวกับไมโครโพรเซสเซอร์ ก็จะทำการรวมวงจรที่จำเป็น เช่น หน่วยความจำ ส่วนอินพุต/เอาต์พุตบางส่วนเข้าไปในตัวไอซีเดียวกันและเพิ่มวงจรบางอย่างเข้าไปด้วยเพื่อให้มีความสามารถเหมาะกับการใช้งานควบคุม เช่น วงจรตั้งเวลา วงจรการสื่อสารอนุกรม เป็นต้น ดังนั้นไมโครคอนโทรลเลอร์สามารถจะทำงานเสมือนกับเป็นคอมพิวเตอร์เล็กๆเครื่องหนึ่งกล่าวโดยสรุปคือ

$$\text{Microcontroller} = \text{Microprocessor} + \text{Memory} + \text{I/O}$$

ปัจจุบันไมโครคอนโทรลเลอร์ถูกนำไปใช้อย่างกว้างขวาง โดยมักจะเป็นการนำไปประยุกต์ใช้ในระบบร่วมกับอุปกรณ์อื่นๆ (Embedded Systems) เพื่อใช้ควบคุมการทำงานบางอย่าง เช่น ใช้ในรถยนต์, เครื่องปรับอากาศ, เครื่องซักผ้าอัตโนมัติ เป็นต้น เพราะว่าไมโครคอนโทรลเลอร์มีข้อดีเหมาะสมต่อการใช้งานควบคุมหลายประการเช่น

1. ไอซีและระบบที่ได้มีขนาดเล็ก
2. ระบบที่ได้มีราคาถูกกว่าการใช้ชิปไมโครโพรเซสเซอร์
3. วงจรที่ได้จะมีความซับซ้อนน้อย ช่วยลดข้อผิดพลาดที่อาจจะเกิดขึ้นได้ในการต่อวงจร
4. มีคุณสมบัติเพิ่มเติมสำหรับงานควบคุม โดยเฉพาะซึ่งใช้งานได้ง่าย
5. ช่วยลดระยะเวลาในการพัฒนาระบบได้

ไมโครคอนโทรลเลอร์มีหลายยี่ห้อ หลายตระกูล และหลายเบอร์ด้วยกัน ซึ่งแต่ละเบอร์ก็จะมีโครงสร้างภายในและความสามารถในการทำงานที่แตกต่างกัน ทำให้เลือกใช้งานได้อย่างเหมาะสม

2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51

การเรียนรู้เพื่อพัฒนาไมโครคอนโทรลเลอร์ MCS-51 ที่เหมาะสมควรมีขนาดของหน่วยความจำโปรแกรมใหญ่พอสมควร ซึ่งในที่นี้ใช้ MCS-51 เบอร์ P89C81RD2 ของ Phillips เนื่องจากมีหน่วยความจำโปรแกรมแบบแฟลชมากถึง 64 กิโลไบต์ มีพอร์ตให้ใช้งานถึง 4 พอร์ต พร้อมไทมเมอร์ 3 ตัว มีโมดูล PCA สำหรับสร้าง PWM จำนวน 5 ช่อง และยังมีหน่วยความจำแรมพิเศษอีก 1 กิโลไบต์ และมีความสามารถเด่นคือ สามารถโปรแกรมหน่วยความจำผ่านพอร์ตอนุกรมในแบบ ISP (In-system Programming) ทำให้สามารถพัฒนาหรือแก้ไขโปรแกรมได้สะดวกขึ้น

2.2.1.1 คุณสมบัติทางเทคนิคของ P89C51RD2

เป็นไมโครคอนโทรลเลอร์ 8 บิต ที่เข้ากันได้กับไมโครคอนโทรลเลอร์แบบพื้นฐานหน่วยความจำโปรแกรมภายในเป็นแบบแฟลช ทำให้สามารถลบและเขียนใหม่ได้นับหมื่นครั้งจึงสามารถใช้งานในรูปแบบไมโครคอนโทรลเลอร์ชิปเดี่ยว (Single chip) โดยไม่ต้องใช้หน่วยความจำภายนอก ทำให้ใช้งานพอร์ตทั้งหมดได้อย่างครบถ้วน และขนาดของหน่วยความจำของ P89C51RD2 นี้มีขนาด 64 กิโลไบต์

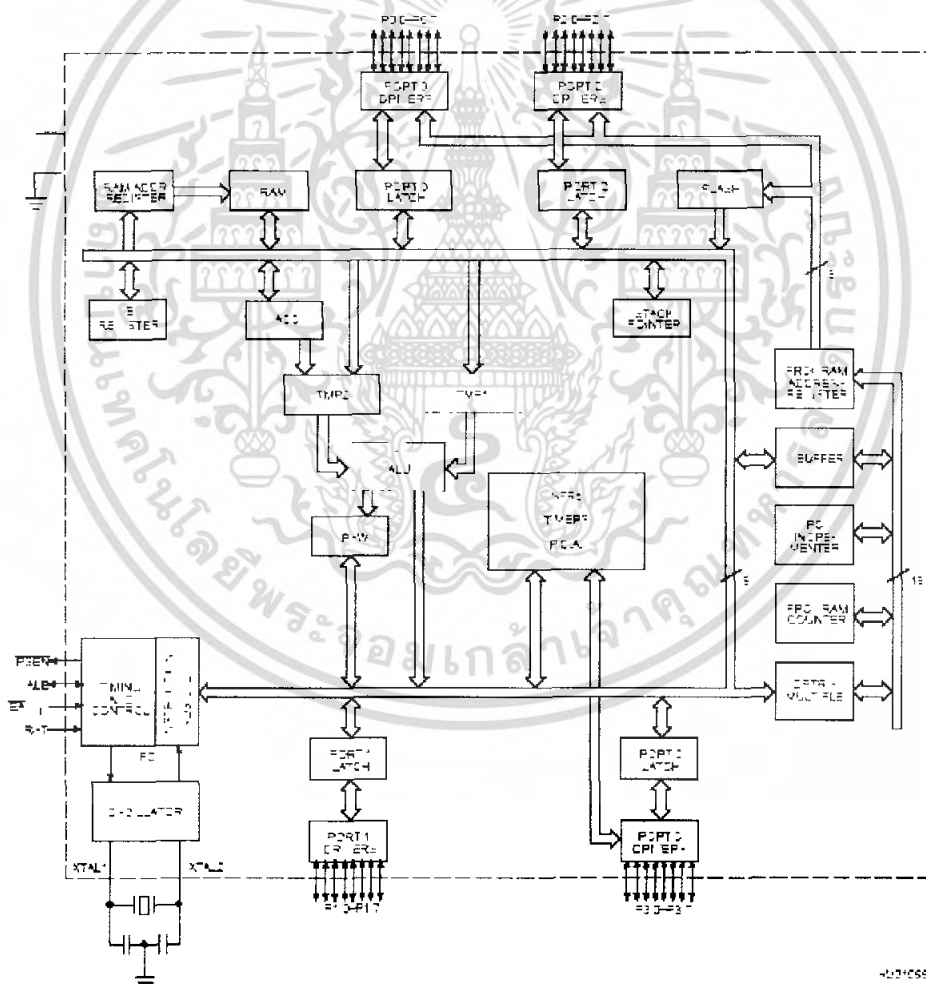
- หน่วยความจำข้อมูลภายใน 1 กิโลไบต์
- สามารถโปรแกรมข้อมูลลงในหน่วยความจำโปรแกรมในแบบวงจรหรือในระบบ (ISP : In-system programming)
- ความถี่สัญญาณนาฬิกาสูงสุด 33 เมกะเฮิร์ตซ์ ในกรณีทำงานด้วยสัญญาณนาฬิกาภายใน 12 ลูกต่อแมชชีนไซเคิลและ 20 เมกะเฮิร์ตซ์ ในกรณีทำงานด้วยสัญญาณนาฬิกาภายใน 6 ลูกต่อ Machine Cycle P89C51RD2BN ได้รับการกำหนดให้ทำงานเบื้องต้นในโหมดสัญญาณนาฬิกา 12 ลูกต่อ Machine Cycle สามารถเลือกเปลี่ยนเป็น 6 สัญญาณนาฬิกาต่อ Machine Cycle ได้ โดยสามารถเปลี่ยนกลับไปกลับมาได้ แตกต่างจาก P89C51RD2BN ที่ค่าเบื้องต้นเป็น 6 สัญญาณนาฬิกาต่อ Machine Cycle เมื่อเปลี่ยนเป็น 12 สัญญาณนาฬิกาแล้วจะเปลี่ยนกลับมาอีกไม่ได้
- ขาพอร์ต 8 บิต 4 พอร์ต แบบกึ่งสองทิศทาง (quasi-bidirectional) เป็นได้ทั้งอินพุตและเอาต์พุต
- อุปกรณ์เพอร์เฟอรัลภายในไมโครคอนโทรลเลอร์สามารถทำงานด้วยความเร็ว 12 สัญญาณนาฬิกาต่อ Machine Cycle ได้ แม้ว่าซีพียูจะทำงานด้วยความเร็ว 6 สัญญาณนาฬิกาต่อ Machine Cycle เป็นคุณสมบัติที่เพิ่มเติมเข้ามาในเบอร์ P89C51RD2BN

- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์

- ไทมเมอร์/เคาน์เตอร์ขนาด 16 บิต 3 ตัว (ไทมเมอร์ 0,1 และ 2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีรีจิสเตอร์ตัวชี้ตำแหน่งข้อมูลหรือ DPTR 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเทอร์รัปต์ได้ 7 ประเภท
- กำหนดนัยสำคัญของการตอบสนองอินเทอร์รัปต์ได้ 4 ระดับ
- สามารถติดต่อหน่วยความจำภายนอกได้สูงสุด 64 กิโลไบต์
- มีวอตช์ด็อกไทมเมอร์
- มีโมดูลวงจรนับโปรแกรมได้ (PCA : Programmable Counter Array) ซึ่งบรรจุ วงจรตรวจจับสัญญาณ (capture), เปรียบเทียบสัญญาณ (compare), วงจรมอดูเลชันทางความกว้างพัลส์(PWM) และวอตช์ด็อกไทมเมอร์ (watchdog timer)



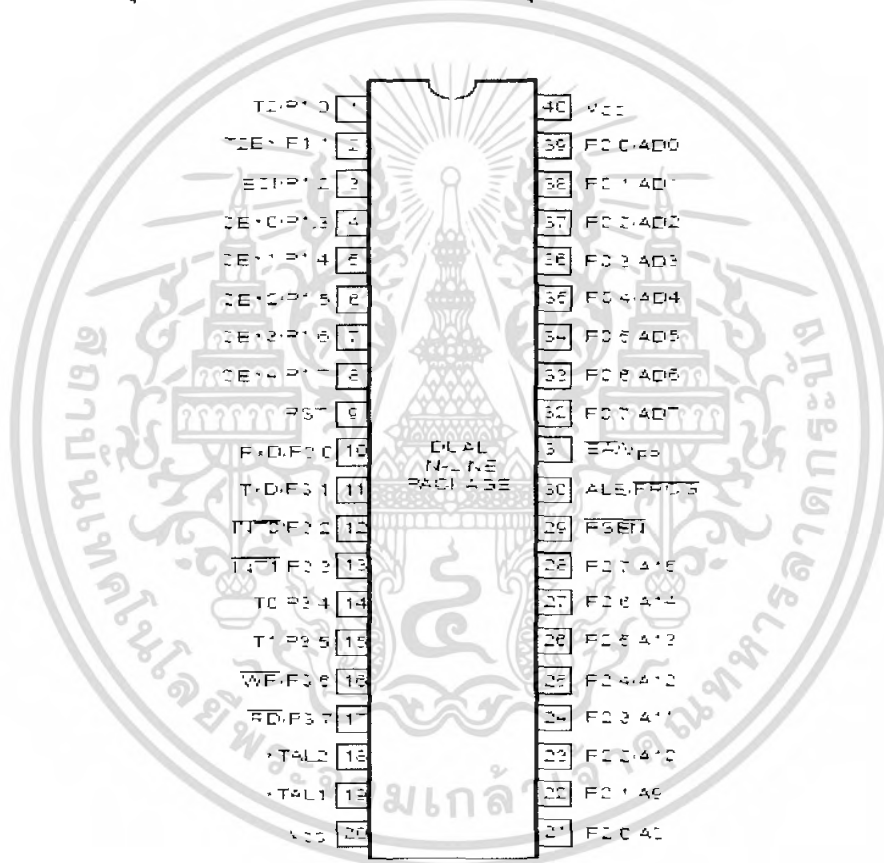
รูปที่ 2.4 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 เบอร์ P89C81RD2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ลักษณะการจัดขาภายนอกของไมโครคอนโทรลเลอร์ MCS-51

ลักษณะการจัดขาภายนอกของไมโครคอนโทรลเลอร์ MCS-51 จะมีการแบ่งกลุ่มการจัดขาออกเป็น 4 กลุ่มด้วยกัน คือ

- กลุ่มขาแหล่งจ่ายไฟเลี้ยง และสัญญาณนาฬิกา
- กลุ่มขาสำหรับการอ้างแอดเดรสและรับส่งข้อมูล
- กลุ่มขาที่ใช้ในการควบคุม
- กลุ่มขาพอร์ตใช้งานแบบขนานและอนุกรม



รูปที่ 2.5 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

2.2.3 ขาที่สำคัญของไมโครคอนโทรลเลอร์ MCS-51

1.ขา Vcc เป็นขารับแรงดันไฟกระแสตรง +5 VDC

2.ขา GND เป็นขาราวด์

3.พอร์ต 0 (P0.0-P0.7) เป็นพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทาง สำหรับใช้งานทั่วไป

4.พอร์ต 1 (P1.0-P1.7) เป็นพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทาง บางขาสำหรับใช้งานพิเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. พอร์ต 2 (P2.0-P2.7) เป็นพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทาง สำหรับใช้งานทั่วไป
6. พอร์ต 3 (P3.0-3.7) เป็นพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทาง บางขาสำหรับใช้งานพิเศษ
7. ขารีเซต (RST) ใช้สำหรับการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยการรีเซตจะต้องคงสถานะ high อย่างน้อยนาน 2 เมกซีไนซ์เกิลในขณะที่ออสซิลเลเตอร์กำลังทำงานอยู่
8. ขา ALE/PROG เป็นขาสัญญาณเพื่อทำหน้าที่ควบคุมการแลตช์ (Latch) ค่าตำแหน่งแอดเดรสไปต์ต่ำ (Address Latch Enable)
9. ขา PSEN (Program Store Enable) ทำหน้าที่เป็นสัญญาณสโตปเพื่ออ่านคำสั่งจากหน่วยความจำภายนอก เมื่อไมโครคอนโทรลเลอร์ประมวลผลคำสั่ง จากหน่วยความจำภายนอก ขานี้จะส่งสัญญาณสโตปจำนวน 2 ครั้งในแต่ละ Machine cycle แต่ในขณะที่ติดต่อกับหน่วยความจำข้อมูลภายนอกจะไม่มี การส่งสัญญาณสโตปแต่อย่างใด
10. ขา EA/Vcc (External Access Enable / Vcc) เป็นขาสำหรับการเลือกใช้หน่วยความจำโปรแกรมจากภายในหรือภายนอก โดยมีสถานะเป็น 0 และ 1 เลขานี้ยังทำหน้าที่รับแรงดันไฟสำหรับโปรแกรม (Vcc) ขนาด 12 โวลต์ เพื่อใช้ในระหว่างการโปรแกรมหน่วยความจำโปรแกรม (EPROM)
11. ขา XTAL1 และขา XTAL2 เป็นขาใช้งานของวงจรอินเวอร์ตติ้งออสซิลเลเตอร์แอมพลิไฟเออร์ (Inverting Oscillator Amplifier) สำหรับใช้ต่อร่วมกับคริสตัลภายนอก

2.3 การอินเตอร์รัปต์ (Interrupt)

การอินเตอร์รัปต์เป็นชื่อเรียกกระบวนการที่เข้ามาขัดจังหวะการทำงานโดยปกติของไมโครคอนโทรลเลอร์

2.3.1 การจัดการอินเตอร์รัปต์ในไมโครคอนโทรลเลอร์ MCS-51

เมื่อมีการอินเตอร์รัปต์ในไมโครคอนโทรลเลอร์ MCS-51 เกิดขึ้นและมีการอินเินาเบิล (enable) การตอบสนองการอินเตอร์รัปต์ไว้ กระบวนการหลังจากนั้นซีพียูจะทำการกระโดดไปยังแอดเดรสในหน่วยความจำที่กำหนดไว้เรียกตำแหน่งแอดเดรสนี้ว่า แอดเดรสอินเตอร์รัปต์เวกเตอร์ (Interrupt vector address) ดังนั้นจะต้องมีการเขียนโปรแกรมย่อยการบริการอินเตอร์รัปต์ไว้ที่แอดเดรสอินเตอร์รัปต์เวกเตอร์นี้ โดยค่าของแอดเดรสอินเตอร์รัปต์เวกเตอร์จะแตกต่างกันไปในการอินเตอร์รัปต์แบบต่างๆ ดังมีรายละเอียดต่อไปนี้

การอินเตอร์รัปต์ภายนอกที่ขา INT0 มีค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์อยู่ที่ 0003H

การอินเตอร์รัปต์ไทมเมอร์ 0 มีค่าแอดเดรสอินเตอร์รัปต์เวกเตอร์อยู่ที่ 000BH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลง หรือต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอินเทอร์รัปต์ภายนอกที่ขา INT1 มีค่าแอดเดรสอินเทอร์รัปต์แวกเตอร์อยู่ที่ 0013H
 การอินเทอร์รัปต์จากไทเมอร์ 1 มีค่าแอดเดรสอินเทอร์รัปต์แวกเตอร์อยู่ที่ 001BH
 การอินเทอร์รัปต์จากพอร์ตอนุกรม มีค่าแอดเดรสอินเทอร์รัปต์แวกเตอร์อยู่ที่ 0023H
 การอินเทอร์รัปต์จากไทเมอร์ 2 มีค่าแอดเดรสอินเทอร์รัปต์แวกเตอร์อยู่ที่ 002BH

2.3.2 การเขียนโปรแกรมย่อยอินเทอร์รัปต์

มีหลักการ โดยทั่วไปดังนี้

1. ต้องเริ่มต้นด้วยแอดเดรสอินเทอร์รัปต์แวกเตอร์เสมอ เพื่อให้การตรวจสอบการทำงานทำได้ง่ายและแยกส่วนของโปรแกรมย่อยนี้ออกจากโปรแกรมหลักหรือโปรแกรมย่อยอื่นๆอย่างชัดเจน ด้วยคำสั่ง `ORGxxxxH` (ค่าแอดเดรสอินเทอร์รัปต์แวกเตอร์)

2. เมื่อเข้าสู่โปรแกรมย่อย ควรเก็บค่าของรีจิสเตอร์หรือแฟล็กที่ใช้แสดงสถานะต่างๆที่ต้องมีการใช้งานในโปรแกรมย่อยบริการอินเทอร์รัปต์ไว้ในสแต็กเสียก่อน เพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้นต่อการทำงานทั้งของโปรแกรมบริการอินเทอร์รัปต์นี้และโปรแกรมหลักด้วยคำสั่ง `PUSH`

3. เมื่อเขียนโปรแกรมบริการอินเทอร์รัปต์เรียบร้อยแล้ว ให้ทำการคืนค่าของรีจิสเตอร์ที่นำมาใช้ในโปรแกรมบริการอินเทอร์รัปต์ด้วยคำสั่ง `POP` ยกเว้นรีจิสเตอร์ที่ต้องการนำผลการกระทำในโปรแกรมบริการอินเทอร์รัปต์นี้ไปใช้งาน ซึ่งในทางปฏิบัติจริง ไม่พบมากนักและไม่แนะนำให้เขียนโปรแกรมในลักษณะนี้

4. ปิดท้ายโปรแกรมย่อยบริการอินเทอร์รัปต์ด้วยคำสั่ง `RETI` เสมอ

2.3.3 รีจิสเตอร์ที่เกี่ยวข้องกับการอินเทอร์รัปต์ในไมโครคอนโทรลเลอร์ MCS-51

1. รีจิสเตอร์ IE (Interrupt Enable) ใช้ในการกำหนดว่าจะยอมให้มีการอินเทอร์รัปต์จากแหล่งใดได้บ้าง โดยมีรายละเอียดแต่ละบิตเป็นดังนี้

บิต 7

บิต 0

EA	X	ET2	ES	ET1	EX1	ET0	EX0
----	---	-----	----	-----	-----	-----	-----

EA ถ้าเป็นลอจิก “1” หมายความว่า ให้อินเทอร์รัปต์ได้

ET2 ถ้าเป็นลอจิก “1” จะอินาเบิ้ลไทเมอร์ 2 (ใช้กับเบอร์ที่มีไทเมอร์ 2)

ES ถ้าเป็นลอจิก “1” จะอินาเบิ้ลอินเทอร์รัปต์จากพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ET1 ถ้าเป็นลอจิก “1” จะอินเวิลไทเมอร์ 1

EX1 ถ้าเป็นลอจิก “1” จะอินเวิลสัญญาณอินเตอร์รัปต์จากภายนอกที่เข้ามาทางขา INT1

ET0 ถ้าเป็นลอจิก “1” จะอินเวิลไทเมอร์ 0

EX0 ถ้าเป็นลอจิก “1” จะอินเวิลสัญญาณอินเตอร์รัปต์จากภายนอกที่เข้ามาทางขา INTO

2.รีจิสเตอร์ IP (Interrupt Priority) ใช้กำหนดลำดับการอินเตอร์รัปต์ กรณีที่เกิดการอินเตอร์รัปต์จากหลายแหล่งพร้อมๆกัน

3.รีจิสเตอร์ TCON (Timer control) รีจิสเตอร์ตัวนี้นอกจากจะใช้ควบคุมไทเมอร์แล้วยังใช้ในการอินเตอร์รัปต์อีกด้วย โดยมีรายละเอียดดังนี้

บิต 7				บิต 0			
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1 เป็นบิตโอเวอร์โฟลว์ของไทเมอร์ 1 จะเป็นลอจิก “1”เมื่อไทเมอร์เกิดโอเวอร์โฟลว์ และบิตนี้สามารถอินเตอร์รัปต์ไมโครคอนโทรลเลอร์ได้ เมื่อทำโปรแกรมตอบสนองการอินเตอร์รัปต์จบบิต TF1 นี้จะกลับมาเป็นลอจิก “0”

TR1 ใช้เปิดไทเมอร์ 1

TF0 เหมือนกับ TF1 แต่ใช้กับไทเมอร์ 0

TR0 ใช้เปิดไทเมอร์ 0

IE1 เป็นบิตแสดงการอินเตอร์รัปต์ทางฮาร์ดแวร์ที่เข้ามาทาง INT1

IT1 ใช้ในการเลือกรูปแบบสัญญาณอินเตอร์รัปต์จากภายนอกที่เข้ามาทางขา INT1 ที่ต้องการตอบสนอง ถ้าเป็นลอจิก “0” หมายความว่า จะเกิดการอินเตอร์รัปต์เมื่อมีสัญญาณขอบขาลงเข้ามา ถ้าเป็นลอจิก “1” หมายความว่า จะเกิดการอินเตอร์รัปต์เมื่อมีระดับลอจิกต่ำ

IE0 ใช้งานเหมือนกับ IE1 แต่จะใช้กับ INTO

IT0 ใช้งานเหมือนกับ IT1 แต่จะใช้กับ INTO

2.3.4 แหล่งกำเนิดสัญญาณอินเทอร์รัปต์จากไมโครคอนโทรลเลอร์ MCS-51

2.3.4.1 สัญญาณอินเทอร์รัปต์จากภายนอก

เป็นการตรวจสอบสัญญาณที่เข้ามายังขา INTO และ INT1 หากตรงตามเงื่อนไขที่กำหนดก็จะทำให้เกิดการอินเทอร์รัปต์ขึ้น โดยการอินทิเนลการอินเทอร์รัปต์แบบนี้สามารถกระทำได้โดยการกำหนดค่าในรีจิสเตอร์ IE ที่บิต EX0 สำหรับสัญญาณอินเทอร์รัปต์ที่ขา INTO และบิต EX1 สำหรับสัญญาณอินเทอร์รัปต์ที่ขา INT1 และทำการเลือกเงื่อนไขของการตรวจสอบสัญญาณในรีจิสเตอร์ TCON ที่บิต IE0 สำหรับสัญญาณอินเทอร์รัปต์ที่ขา INTO และบิต IE1 สำหรับสัญญาณอินเทอร์รัปต์ที่ขา INT1

เงื่อนไขการตรวจสอบสัญญาณอินเทอร์รัปต์ที่ขา INTO และ INT1 มีด้วยกัน 2 ลักษณะ คือ

1. ตรวจสอบระดับลอจิก ถ้าบิต IEx ในรีจิสเตอร์ TCON เป็น “0” จะเกิดการอินเทอร์รัปต์จากภายนอกที่ขา INTO หรือ INT1 ได้เมื่อตรวจพบระดับลอจิกต่ำ หรือ “0” เมื่อเกิดการอินเทอร์รัปต์แล้วให้ดำเนินการทำให้สัญญาณที่ขาอินพุตกลับสู่ระดับลอจิก “1” ก่อนที่การบริการอินเทอร์รัปต์เสร็จสิ้นเพื่อป้องกันการเกิดอินเทอร์รัปต์ซ้อน

2 ตรวจสอบขอบขาของสัญญาณ ถ้าหากบิต IEx ในรีจิสเตอร์ TCON เป็น “1” จะเกิดการอินเทอร์รัปต์จากภายนอกที่ขา INTO หรือ INT1 ได้ต่อเมื่อตรวจพบการเปลี่ยนแปลงของสัญญาณที่ขา INTO หรือ INT1 จาก “1” เป็น “0” หรือตรวจสอบพบขอบขาลงของสัญญาณที่ป้อนมายังขา INTO หรือ INT1 และต้องมีกัรักษาสถานะลอจิก “0” นี้เป็นเวลาอย่างน้อย 1 Machine Cycle จึงถือว่าการอินเทอร์รัปต์อย่างสมบูรณ์

เมื่อเกิดการอินเทอร์รัปต์ขึ้น ซีพียูภายในไมโครคอนโทรลเลอร์จะกระโดดไปยังแอดเดรส 0003H สำหรับการอินเทอร์รัปต์ที่ขา INTO และ 0013H สำหรับการอินเทอร์รัปต์ที่ขา INT1

2.3.4.2 การอินเทอร์รัปต์จากไทเมอร์/คาน์เตอร์ 0 และ 1

แหล่งกำเนิดอินเทอร์รัปต์นี้จัดเป็นแหล่งกำเนิดอินเทอร์รัปต์ภายในแบบหนึ่ง โดยใช้การเกิดโอเวอร์โฟลว์ จากการนับค่าในไทเมอร์/คาน์เตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51 เมื่อไทเมอร์ 0 เกิดการโอเวอร์โฟลว์ ก็จะทำการเซตบิต TF0 ในรีจิสเตอร์ TCON และถ้าไทเมอร์ 1 เกิดการโอเวอร์โฟลว์ บิต TF1 ในรีจิสเตอร์ TCON จะได้รับการเซตค่าเช่นเดียวกัน

ค่าแอดเดรสอินเทอร์รัปต์แวกเตอร์ของการอินเทอร์รัปต์แบบนี้อยู่ที่ 000BH สำหรับไทเมอร์ 0 และ 001BH สำหรับไทเมอร์ 1

อย่างไรก็ตามการอินเทอร์รัปต์แบบนี้จะเกิดขึ้นหรือมีการตอบสนองก็ต่อเมื่อมีการอินทิเนลการอินเทอร์รัปต์ โดยการเซตบิต EA, ET0 และ ET1 ในรีจิสเตอร์ IE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4.3 การอินเทอร์รัปต์จากพอร์ตอนุกรม

แหล่งกำเนิดอินเทอร์รัปต์นี้จัดเป็นแหล่งกำเนิดอินเทอร์รัปต์ภายในแบบหนึ่ง เมื่อวงจรพอร์ตอนุกรมส่งหรือรับข้อมูลเสร็จสมบูรณ์ก็จะกำเนิดสัญญาณอินเทอร์รัปต์ขึ้น โดยการเซตบิต RI ในกรณีรับข้อมูล และบิต TI ในกรณีส่งข้อมูล RI และ TI อยู่ในรีจิสเตอร์ SCON

ค่าแอดเดรสอินเทอร์รัปต์แวกเตอร์ของการอินเทอร์รัปต์แบบนี้อยู่ที่ 0023H การอินเทอร์รัปต์แบบนี้สามารถแทนได้ด้วยการออร์กันของบิต RI และ TI

2.3.4.4 การอินเทอร์รัปต์จากไทเมอร์/เคาน์เตอร์ 2

แหล่งกำเนิดอินเทอร์รัปต์นี้จัดเป็นแหล่งกำเนิดอินเทอร์รัปต์ภายในแบบหนึ่ง โดยใช้การเกิดโอเวอร์โฟลว์ จากการนับค่าในไทเมอร์/เคาน์เตอร์ 2 หรือจากการแคปเจอร์หรือการตรวจจับสัญญาณที่ขา T2EX เมื่อเกิดการโอเวอร์โฟลว์ก็จะทำการเซตบิต TF2 ในรีจิสเตอร์ TCON และถ้าไทเมอร์ 2 สามารถตรวจจับการเปลี่ยนแปลงจากระดับ “1” เป็น “0” ที่ขา T2EX ได้ และบิต EXEN2 ในรีจิสเตอร์ T2CON ได้รับการเซตไว้ ก็จะทำให้บิต EXF2 ในรีจิสเตอร์ T2CON เซต เป็นการแจ้งว่าเกิดการอินเทอร์รัปต์เนื่องจากการแคปเจอร์ที่ไทเมอร์ 2 ค่าแอดเดรสอินเทอร์รัปต์แวกเตอร์ของการอินเทอร์รัปต์แบบนี้อยู่ที่ 002BH การอินเทอร์รัปต์แบบนี้สามารถแทนได้ด้วยการออร์กันของบิต TF2 และ EXF2

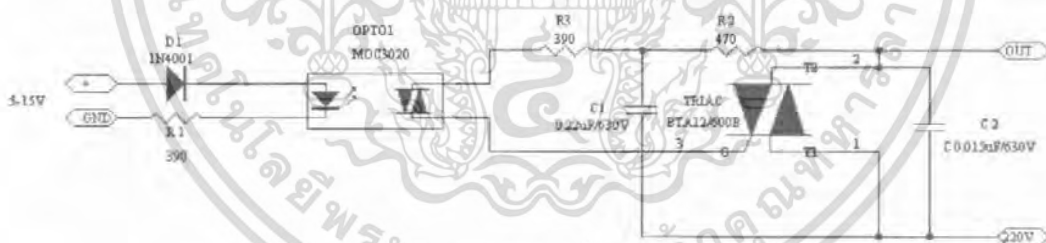
อย่างไรก็ตามการอินเทอร์รัปต์แบบนี้จะเกิดขึ้นหรือมีการตอบสนองก็ต่อเมื่อมีการอินาเบิลการอินเทอร์รัปต์ โดยการเซตบิต EA ในรีจิสเตอร์ IE จะตอบสนองในกรณีเกิดโอเวอร์โฟลว์เมื่อบิต EXEN2 ในรีจิสเตอร์ T2CON เป็น “0” และตอบสนองในกรณีเกิดแคปเจอร์เมื่อบิต EXEN2 ในรีจิสเตอร์ T2CON เป็น “1”

2.4 วงจรโซลิตเตทรีเลย์ (Solid State Relay)



รูปที่ 2.6 โซลิตเตทรีเลย์ (Solid State Relay)

วงจร โซลิตเตทรีเลย์เป็นวงจรที่ทำหน้าที่เสมือนเป็นสวิตช์อิเล็กทรอนิกส์ที่ประกอบด้วย ไตรแอกที่ทำงานคล้ายรีเลย์ แต่มีข้อดีกว่าก็คือ ใช้ไตรแอกทำหน้าที่เป็นสวิตช์แทนรีเลย์ ทำให้ไม่เกิดการสปาร์กที่หน้าคอนแทค อายุการใช้งานจึงนานกว่า ไม่มีหน้าคอนแทคให้เสียหายอายุการใช้งานยาวนานกว่า และยังใช้ไอซีเชื่อมต่อโยงทางแสง (OPTO COPLER) ในการแยกไฟสูงไฟต่ำเพื่อลดสัญญาณรบกวนจึงเหมาะแก่การนำมาใช้ในการควบคุมอุปกรณ์ไฟฟ้า



รูปที่ 2.7 รูปวงจรถอยโซลิตเตทรีเลย์ (Solid State Relay)

การทำงาน จากรูป ไอซี 1 เป็นไอซีเชื่อมต่อโยงทางแสง (OPTO COPLER) เพื่อแยกไฟสูงและไฟต่ำ R1 ควบคุมกระแสไฟให้กับ LED ในไอซีส่วนด้านไฟสูงแรงไฟสูงจะไหลผ่าน R2, R3 มาเข้าขา 6 แล้วออกทางขา 4 ไอซีไปเข้าขา G ของไตรแอกเพื่อทริกให้ไตรแอกทำงาน เนื่องจาก IC 1 เป็นไอซีแบบไตรแอกและทำงานที่แรงดัน 0 โวลต์ ดังนั้นจึงขจัดปัญหาการรบกวนและยืดอายุการใช้งานของไตรแอก

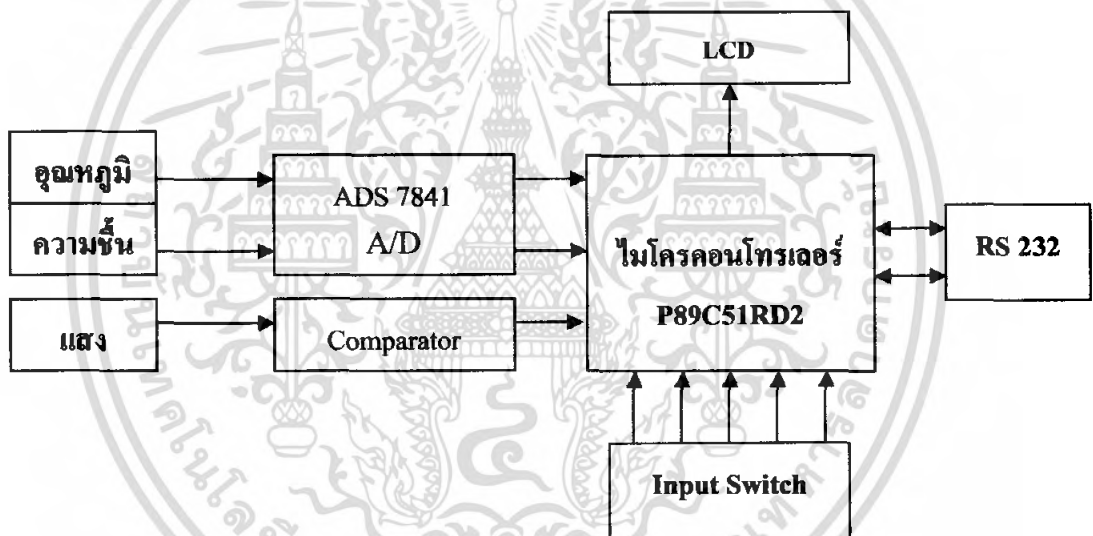
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและสร้างระบบควบคุมการเลี้ยงพืชไร้ดิน

3.1 ภาคตรวจจับอุณหภูมิ ความชื้น และภาคแสดงผล

ประกอบด้วย ตรวจจับอุณหภูมิและความชื้นใช้ IC เบอร์ LM35 และ HIH-4000 ตามลำดับ ซึ่งเป็น IC ที่สามารถตรวจจับอุณหภูมิ, ความชื้นและใช้ LDR เป็นเซ็นเซอร์รับแสง และมี ADS 7841 เป็นตัวแปลงจากสัญญาณอนาล็อกเป็น ดิจิตอล โดยในระบบจะถูกควบคุมไปด้วย ไมโครคอนโทรลเลอร์เบอร์ P89C51RD2 จะแสดงผลที่จอ LCD และ ส่งข้อมูลผ่านในแบบมาตรฐาน RS232 เข้าสู่เครื่องคอมพิวเตอร์ ผ่านพอร์ตอนุกรมเพื่อประมวลผล



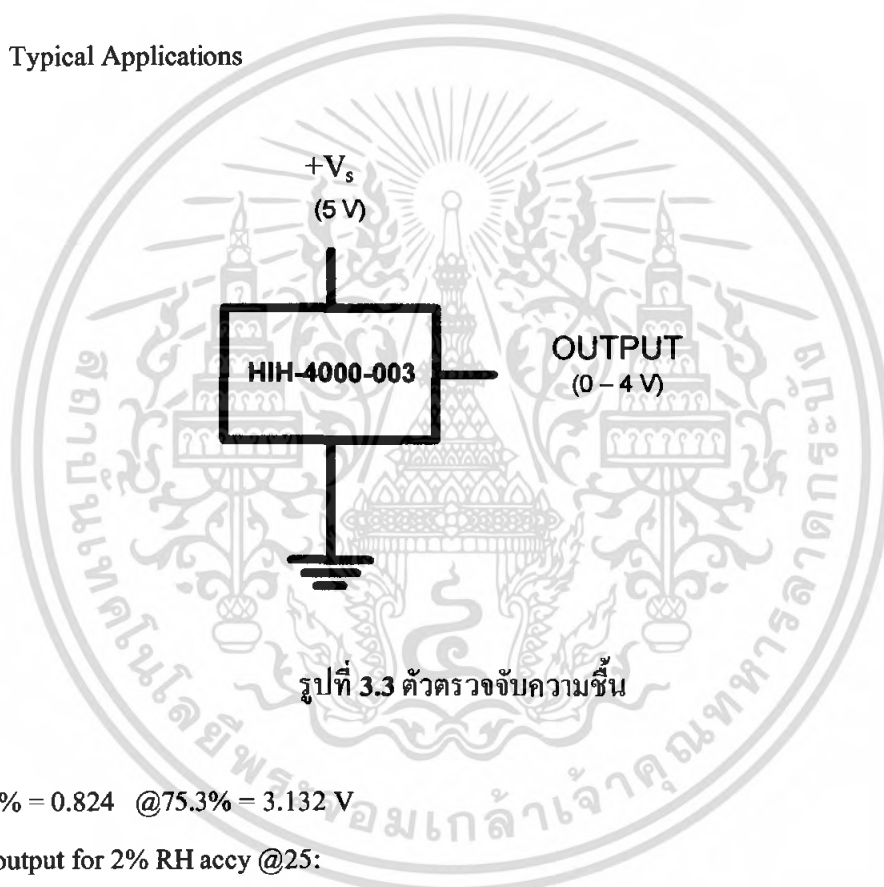
รูปที่ 3.1 ภาพตรวจจับอุณหภูมิ, ความชื้น, แสงแสดงผล และการสื่อสาร RS 232

3.1.2 การออกแบบตัวตรวจจับความชื้น ความชื้น HIH-4000

ตัววัดความชื้นตัวนี้จะมี V_{out} ที่ 0% = 0.824 โวลต์ แล้วจะมีความชันของความชื้นที่ 30.655 มิลลิโวลต์ ต่อ %RH ดังนั้น ถ้า RH เราจะหาได้จาก การนำเอา V_{out} มาลบด้วยค่าแรงดันเริ่มต้นที่ 0% คือ 0.824 มิลลิโวลต์ แล้วนำมาหาร 0.037 ดังสมการ

$$RH = (V_{out} - 0.824) / 0.0307 \quad (4.1)$$

Typical Applications



รูปที่ 3.3 ตัวตรวจจับความชื้น

$$V_{out} @0\% = 0.824 \quad @75.3\% = 3.132 \text{ V}$$

Linear output for 2% RH accy @25:

$$\text{Zero offset} = 0.824 \text{ V}$$

$$\text{Slope} = 30.655 \text{ mV} / \%RH$$

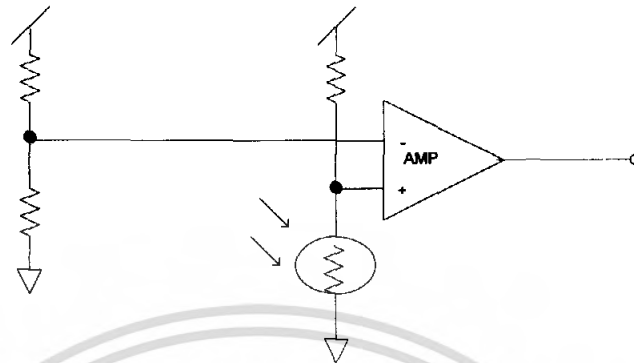
$$RH = (V_{out} - 0.824) / 0.0307$$

Ratio metric response for 0 to 100%RH:

$$V_{out} = S_{supply} * (0.1648 \text{ to } 0.7779)$$

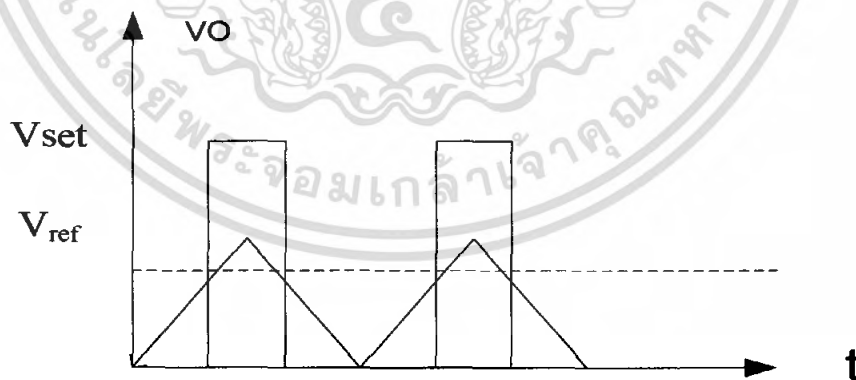
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 การออกแบบวงจรภาคตรวจจับแสง



รูปที่ 3.4 วงจรภาคตรวจจับแสง

ในการออกแบบวงจรภาคนี้เราได้ทำการออกแบบ โดยใช้หลักการของวงจรตรวจสอบระดับแรงดันที่เป็นบวกไม่กลับ หรืออนอินเวอร์ โดยมีแรงดันทางด้านอินพุต ได้จากการเปลี่ยนแปลงความต้านทานของ แอลดีอาร์ ในระดับแสงต่างๆ ซึ่งเกิดการเปลี่ยนแปลงแรงดันทางด้านอินพุต ที่เข้าทางขานอนอินเวอร์คิง และเราจะป้อนระดับแรงดันเทียบเคียง V_{ref} ป้อนให้ขาอินพุตลบ เมื่อแรงดันของอินพุตที่ได้จาก แอลดีอาร์ สูงเกินระดับแรงดันเปรียบเทียบจะทำให้แรงดันทางด้านเอาต์พุตใกล้เคียงกับแหล่งจ่ายไฟ



รูปที่ 3.5 ลักษณะของสัญญาณ

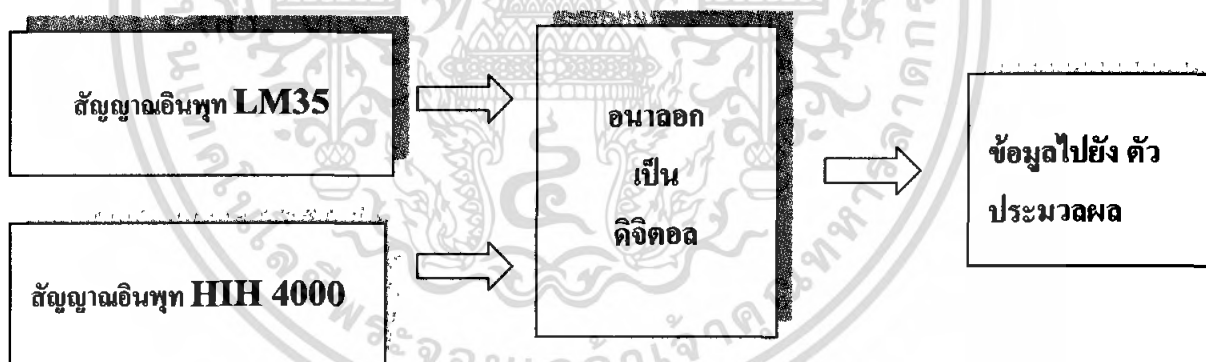
ในการออกแบบเราได้ตั้งค่าแรงดันอ้างอิงที่ 1.7 โวลต์ โดยการตั้งแรงดันอ้างอิงได้มาจากระดับของแสงในช่วงเย็น โดยความต้านทานของแอลดีอาร์ จะมีค่าเพิ่มมากขึ้นเมื่อได้รับแสงน้อยลง

ในช่วงที่เย็น แสงที่ได้รับจะทำให้เกิดความต้านทานประมาณ 25 กิโลโห์ม จึงนำค่านี้ม่ออกแบบเป็นวงจรเปรียบเทียบกับแรงดัน

$$V_o = V_{ref} = (5 \cdot 25k) / (50k + 25k) = 1.7 \text{ โวลต์} \quad (4.2)$$

3.2 การออกแบบวงจรการเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล

ในการออกแบบวงจรส่วนนี้เข้าทำการออกแบบด้วยการรับ อินพุต เป็นแรงดันจากตัวตรวจจับอุณหภูมิ โดยใช้ไอซีเบอร์ ADS 7841 ซึ่งทำหน้าที่แปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลขนาด 12 บิตหรือ 4096 ชั้น แล้วส่งข้อมูลออกไปเป็นสัญญาณอนุกรมและส่งข้อมูลออกไปภาคประมวลผลแบบขนานซึ่งการทำงานสามารถที่จะส่งข้อมูลไปยังตัวประมวลผลได้ถูกต้องแม่นยำ และตัวประมวลผลสามารถรับข้อมูลได้อย่างถูกต้องและเกิดความง่ายในการเขียนโปรแกรมควบคุม



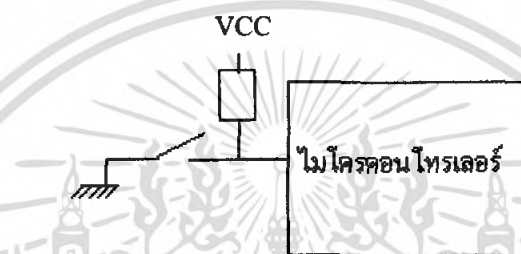
รูปที่ 3.6 บล็อกแสดงการแปลงสัญญาณอนาลอกเป็นดิจิตอล

ในการให้ข้อมูลแบบขนานนี้ ทำให้ง่ายต่อการออกแบบในการรับข้อมูลเขียนโปรแกรมบนตัวไมโครคอนโทรลเลอร์ ในการออกแบบวงจร การเปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิตอล เราต้องพิจารณาข้อมูลทางด้าน อินพุต และแรงดันอ้างอิง (V_{ref}) และอัตราการแปลงสัญญาณ ในการออกแบบ ซึ่งเพิ่มขึ้น 10 มิลลิโวลต์/องศา เพื่อที่สามารถแปลงค่าอุณหภูมิได้ในช่วง 0-125 องศาเซลเซียส และความชื้นจะมีค่า เริ่มต้นที่ 0% คือ 0.824 มิลลิโวลต์จะเพิ่มขึ้น 30.655 มิลลิโวลต์ ต่อ %RH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบวงจรภาครับการกดคีย์ข้อมูล

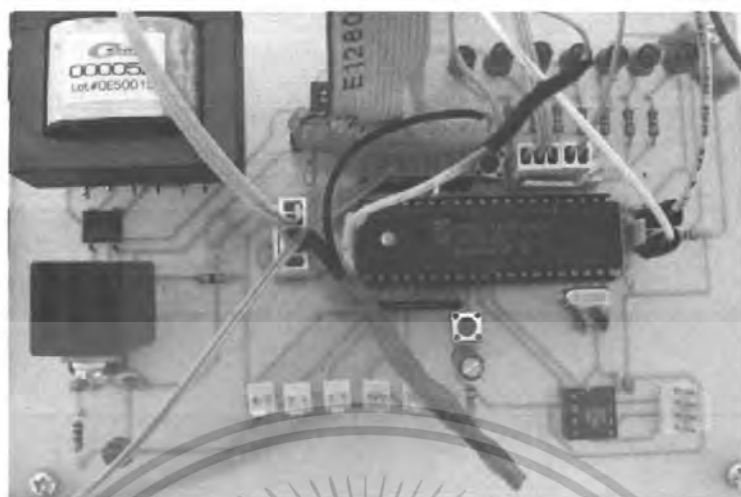
ในการใช้พอร์ทของ P89C51RD2 ถ้าเลือกเป็น พอร์ทอินพุต เราต้องมีการเขียนค่า 1 ลงไปใน พอร์ทที่ต้องการ เพื่อให้เกิดคุณสมบัติเป็น High Impedent ดังนั้นในการออกแบบวงจรภาครับอินพุตสามารถที่จะออกแบบโดยใช้แนวความคิดว่า เมื่อกดสวิตช์แล้ว วงจรจะได้รับค่า 0 และเมื่อมีการปล่อยสวิตช์วงจรรับค่า 5 โวลต์ จะเกิดขึ้นเมื่อมีการอ่านค่าจาก ไมโครคอนโทรเลอร์ (แต่ถ้าเซตพอร์ทเป็นพูลอัพภายในก็ไม่จำเป็นต้องมี ตัวต้านทานมาพูลอัพภายนอกก็ได้) ดังรูป



รูปที่ 3.7 การออกแบบวงจรภาครับการกดคีย์ข้อมูล

3.4 การออกแบบภาคประมวลผล

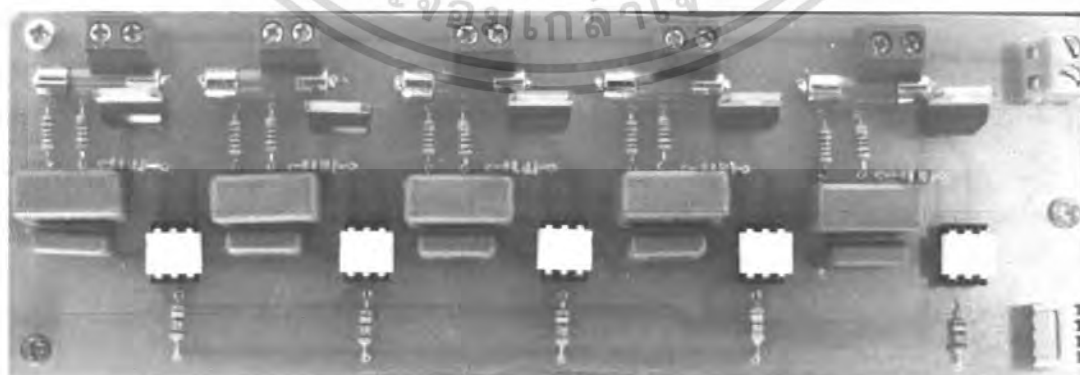
ภาคประมวลผลนี้จะใช้ไมโครคอนโทรเลอร์ P89c51RD2 เป็นตัวประมวลผลจากการรับข้อมูลจากเซ็นเซอร์ตัวตรวจจับอุณหภูมิและความชื้น มาประมวลผลออกทางจอ LCD และมี เซนเซอร์ตรวจจับแสง(LDR)เพื่อควบคุมหลอดไฟ AC 220 โวลต์ให้ทำงานตามความมืด-สว่างของแสง และรับอินพุตจากสวิตช์ ทั้งหมด 5 ตัว โดยสวิตช์ตัวแรกมีหน้าที่เป็นตัวปิด-เปิดสถานะน้ำ แสดงผลเป็นLED สีน้ำเงิน(ด้านล่าง) ตัวที่สองมีหน้าที่เป็นปุ่ม แสดงผลเป็นLED สีแดง ตัวที่สามทำหน้าที่เป็นตัวรีเซ็ตพ่นน้ำ แสดงน้ำ แสดงผลเป็นLED สีน้ำเงิน(ด้านบน) ตัวที่สี่ทำหน้าที่ปิด-เปิดหลอดไฟ AC 220 โวลต์ 4 ดวง ตัวที่ห้าทำหน้าที่เปิดปิดพัดลม AC 220 โวลต์ มีเพื่อทำหน้าที่ควบคุมภาคเอาต์พุต และยังมีวงจรสำหรับติดต่อกับ RS-232 เพื่อส่งค่าอุณหภูมิและความชื้นออกไปยังหน้าจอคอมพิวเตอร์ สามารถพิจารณาได้จาก DESIGN 1



รูปที่ 3.8 ภาคประมวลผล

3.5 การออกแบบภาคเอาต์พุต

ภาคเอาต์พุตนี้เป็นภาคที่ใช้ควบคุมอุปกรณ์ต่างๆ คือ หลอดไฟฟ้า 220 V จำนวนสี่ดวงเพื่อเพิ่มอุณหภูมิให้แบบจำลองโรงเรือน พัดลมดูดอากาศ 220 V จำนวน 1 ตัว ใช้สำหรับระบายอากาศออก หลอด LED แสดงสถานะของปั๊มตัวที่ 1 ปั๊มตัวที่ 2 สถานะของคราบน้ำไหลของน้ำ สถานการณ์ไหลของน้ำ และสถานะของการฉีดน้ำเพื่อลดอุณหภูมิของแบบจำลองโรงเรือน วงจรที่ใช้ควบคุมหลอดไฟฟ้า 220 V และพัดลมดูดอากาศ 220 V จะใช้วงจรโซลีสเตทริเลย์วงจรโซลิด ประกอบด้วยอุปกรณ์หลักๆคือ ไอซีเชื่อมต่อทางแสง, ไตรแอก วงจรนี้จะทำหน้าที่เหมือนสวิตช์อิเล็กทรอนิกส์ ใช้ควบคุมการเปิดปิดอุปกรณ์ไฟฟ้า สามารถพิจารณาได้จาก DESIGN 2

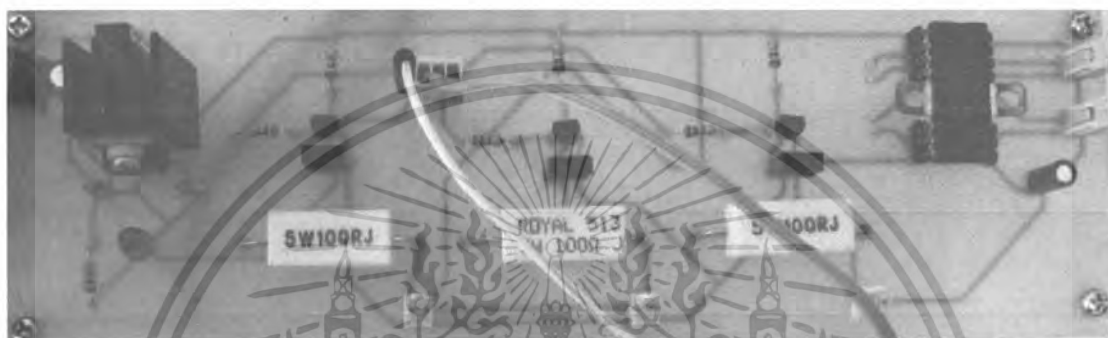


รูปที่ 3.9 ภาคเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ออกแบบวงจรขับหลอด LED และ มอเตอร์

จะเป็นวงจรที่ทำหน้าที่ขับหลอด LED ที่มีอยู่ในแบบจำลองโรงเพาะปลูกรูปลูกพืช ที่ใช้แสดงสถานะน้ำ ปุ๋ย และการฉีดน้ำ จะมีวงจรขับ LED 3 ชุด ใช้ขับ 1 ชุด / 20 ดวง ใช้ทรานซิสเตอร์ BD 139 และ 1815 ในการขับหลอด LED และวงจรขับมอเตอร์ใช้ไอซี TA 7279P ในการขับมอเตอร์ หนึ่งตัวที่ใช้ในการหมุนกล้อง webcam สามารถพิจารณาได้จาก DESIGN 3



รูปที่ 3.10 วงจรขับ LED และ Motor

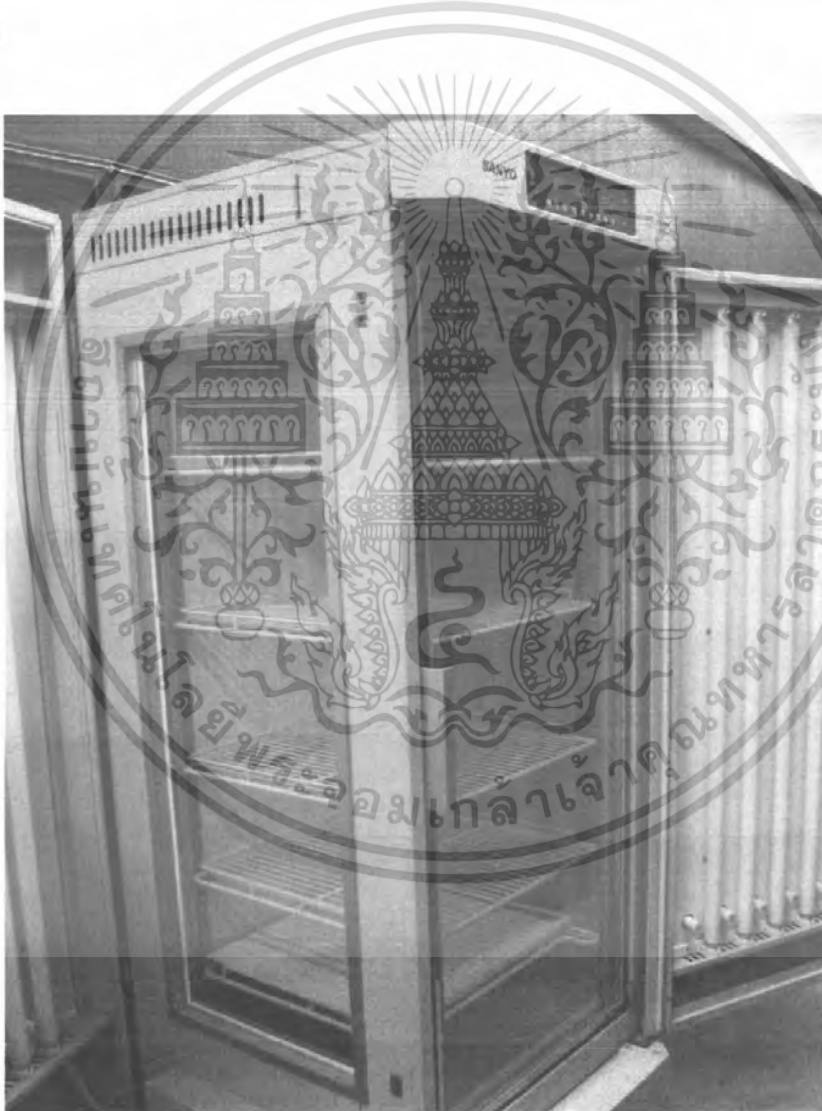
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง

4.1 การทดลองตรวจจับอุณหภูมิและความชื้น

ในการทดลองการตรวจจับอุณหภูมิและความชื้นนี้ ผู้ควบคุมอุณหภูมิและความชื้นของทางคณะเทคโนโลยีการเกษตร ได้รับความอนุเคราะห์ในการยืม ภาควิชาเทคโนโลยีการจัดการ ศัตรูพืช

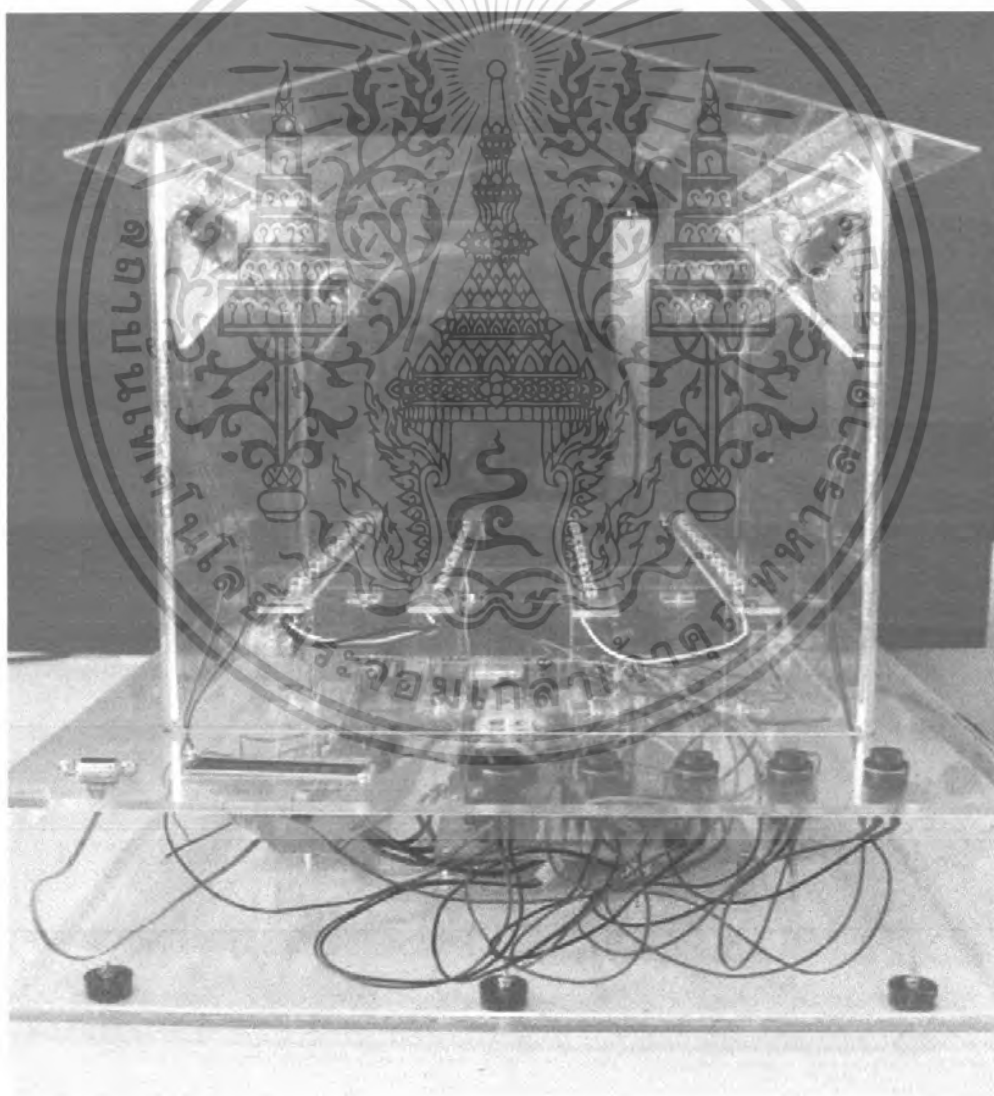


รูปที่ 4.1 ตู้ควบคุมอุณหภูมิและความชื้น (Versatile Environmental Test Chamber)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดลองในตู้ควบคุมอุณหภูมิและความชื้น โดยมีการทดลองนี้ตั้งอุณหภูมิที่ 30 องศาเซลเซียสถึง 40 องศาเซลเซียส ดังนั้นการทดลองนี้จะทำการทดลองอุณหภูมิเทียบกับ เทอร์โมมิเตอร์ และความชื้นเทียบกับ ตู้ควบคุมอุณหภูมิและความชื้น Versatile Environmental Test Chamber ที่ 40-90 %

1. ส่วนของอุณหภูมิ เราจะวัดตรวจจับที่ได้รับการออกแบบให้มีการแสดงผลเทียบค่ากับ เทอร์โมมิเตอร์
2. ส่วนของความชื้น เราจะวัดตรวจจับที่ได้รับการออกแบบให้มีการแสดงค่าความชื้น โดยเทียบกับตู้ควบคุมอุณหภูมิและความชื้น (Versatile Environmental Test Chamber)



รูปที่ 4.2 รูปแบบจำลองโรงเพาะเลี้ยงพืช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 การตรวจวัดอุณหภูมิ

อุณหภูมิ		
เทอร์โมมิเตอร์ (°C)	ค่าที่วัดได้ (°C)	% ความผิดพลาด
30	30.24	0.80
31	31.43	1.39
32	32.70	2.19
33	33.35	1.06
34	34.66	1.94
35	35.22	0.63
36	36.19	0.52
37	37.27	0.73
38	38.55	1.45
39	39.43	1.10
40	40.18	0.45

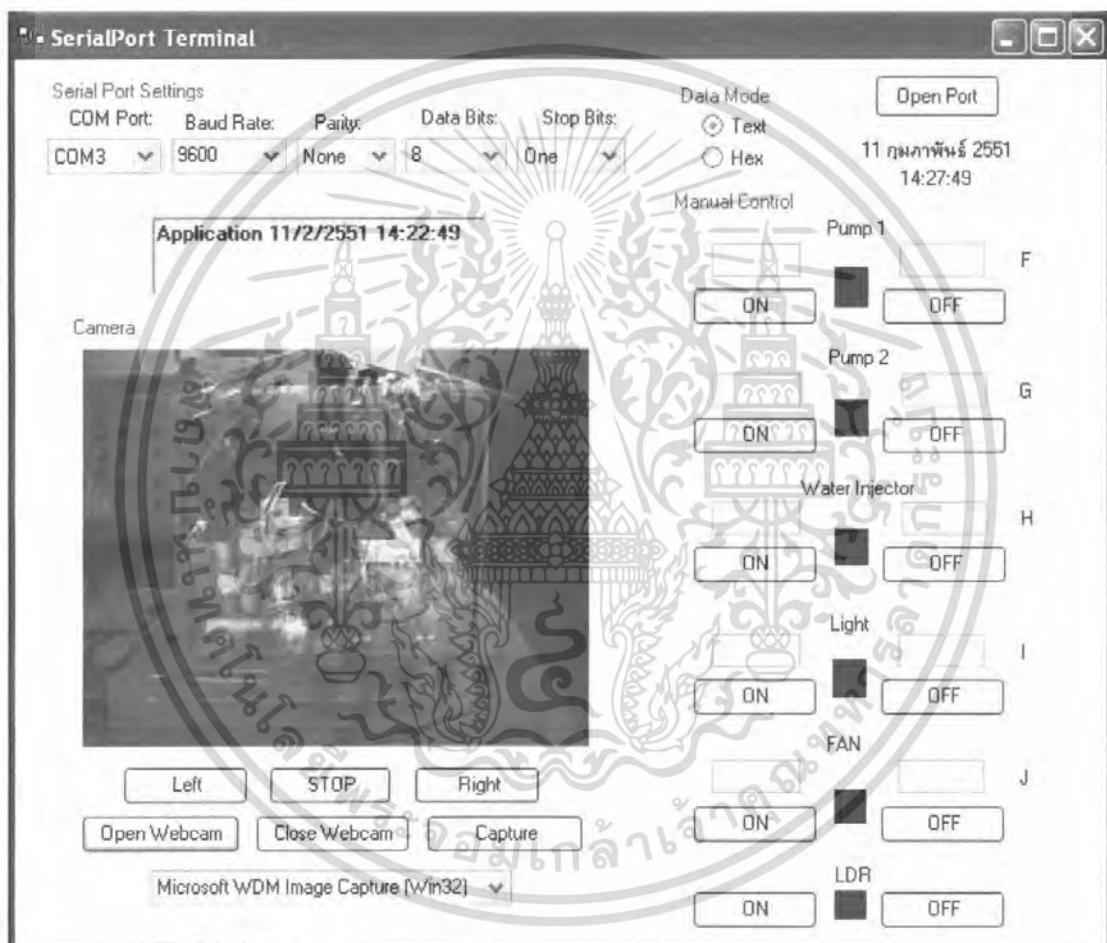
ตารางที่ 4.2 การตรวจวัดความชื้น

ความชื้น		
SANYO Versatile Environmental Test Chamber	ค่าที่วัดได้	% ความผิดพลาด
40	40.06	0.15
45	45.44	0.98
50	50.37	0.74
55	55.82	1.49
60	60.39	0.65
65	65.48	0.74
70	70.13	0.19
75	75.56	0.75
80	80.23	0.29
85	85.36	0.42
90	90.63	0.70

ในการทดลองตัวตรวจจับอุณหภูมิมีค่าความคลาดเคลื่อนน้อยในช่วงอุณหภูมิที่ทำการวัด เพราะเกิดจากไอซีเบอร์ LM 35 ที่ผู้ผลิตได้ออกแบบมาเพื่อใช้ทำการวัดอุณหภูมิโดยตรงและมีความเที่ยงตรงในการตรวจจับอุณหภูมิโดยมีค่าเพิ่มค่าแรงดันขึ้น 10 มิลลิโวลต์ต่อองศาเซลเซียส

ในการทดลองตัวตรวจจับความชื้นมีค่าความคลาดเคลื่อนน้อยเพราะเกิดจากไอซีเบอร์ HIH-4000 ที่ผู้ผลิตได้ออกแบบมาเพื่อทำการใช้การวัดความชื้น โดยตรงและมีความเที่ยงตรงในการตรวจจับค่าความชื้น

4.2 การทดลองทางด้านโปรแกรม (Software)



รูปที่ 4.3 การทดลองทางด้านโปรแกรม (Software)

เป็น โปรแกรมที่เขียนขึ้นมาเพื่อนำมาควบคุมแบบจำลองการเลี้ยงพืช ในโปรแกรมนี้อจะมีปุ่มควบคุม ดังนี้

Pump1 ใช้ควบคุมการเปิด-ปิด สถานะของน้ำแสดงผลที่ หลอดไฟ LED สีน้ำเงิน

Pump2 ใช้ควบคุมการเปิด-ปิด สถานะของปุ๋ยแสดงผลที่ หลอดไฟ LED สีแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Water Injector ใช้ควบคุมการเปิด-ปิด สถานะของการฉีดน้ำแสดงผลที่ หลอดไฟ LED สีน้ำเงิน

FAN ใช้ควบคุมการเปิด-ปิด พัดลม AC 220 V

Light ใช้ควบคุมการเปิด-ปิด หลอดไฟ AC 220 V

LDR กด ON เปิด โหมด LDR จะแสดงผลคือหลอดไฟ 220 V และหลอดไฟ Black Light ของ LCD เปิด – ปิด ตามแสงสว่างได้

กด OFF ปิด โหมด LDR หลอดไฟ Black Light ของ LCD เปิด – ปิด ตามแสงสว่างได้

ตารางที่ 4.3 การทดลองกดปุ่มต่างๆทางด้าน โปรแกรม (Software)

Bottom	สถานะ		การตั้งเวลา	
	ON	OFF	ON	OFF
Pump 1	✓	✓	✓	✓
Pump 2	✓	✓	✓	✓
Fan	✓	✓	✓	✓
Light	✓	✓	✓	✓
Water Injector	✓	✓	✓	✓

อุปกรณ์	Bottom		
LDR	ON*	OFF**	-
Motor	Left	Right	Stop
Webcam	Open Webcam	Close Webcam	Capture

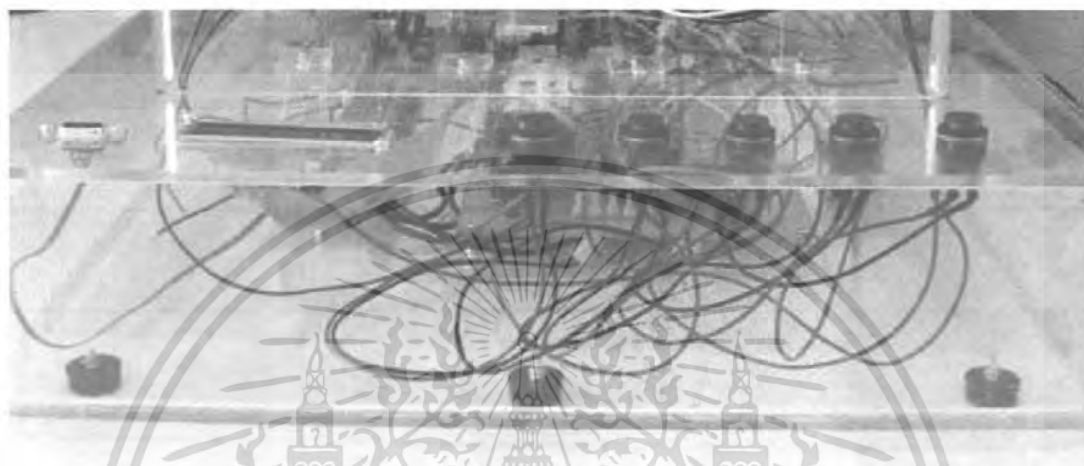
หมายเหตุ

* LDR ถ้ากด ON → หลอดไฟ 220 V และหลอดไฟ Black Light ของ LCD เปิด – ปิด ตามแสงสว่างได้

** LDR ถ้ากด OFF → หลอดไฟ Black Light ของ LCD เปิด – ปิด ตามแสงสว่างได้

4.3 การทดลองการกดปุ่มที่ตัวเครื่องและการแสดงผลที่จอ LCD

4.3.1 การทดลองการกดปุ่มที่ตัวเครื่อง



รูปที่ 4.4 การทดลองการกดปุ่มที่ตัวเครื่อง

ตารางที่ 4.4 การทดลองปุ่มกดที่กล่องควบคุม

Bottom	ON	OFF
ปุ่มที่ 1 (สถานะปุ่มโยย)	✓	✓
ปุ่มที่ 2 (สถานะ ปุ่ม น้ำ)	✓	✓
ปุ่มที่ 3(สถานะการฉีดน้ำ)	✓	✓
ปุ่มที่ 4 (หลอดไฟ 220V)	✓	✓
ปุ่มที่ 5 (พัดลม 220 V)	✓	✓

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 การทดลองการแสดงผลที่จอ LCD



รูปที่ 4.5 การทดลองการแสดงผลที่จอ LCD

ตารางที่ 4.5 การแสดงผลที่จอ LCD

T30.00°C	แสดงผลอุณหภูมิ
H77.61%	แสดงผลความชื้น
IP00000	ไม่มีการกดสวิตช์
IP10000	สวิตช์ตัวที่ 1 ทำงาน
IP01000	สวิตช์ตัวที่ 2 ทำงาน
IP00100	สวิตช์ตัวที่ 3 ทำงาน
IP00010	สวิตช์ตัวที่ 4 ทำงาน
IP00001	สวิตช์ตัวที่ 5 ทำงาน
OP00000	O/P ไม่ทำงาน
OP10000	สถานะปั๊มรูดทำงาน(LED แดง)
OP01000	สถานะปั้มน้ำทำงาน(LED สีน้ำเงิน)
OP00100	สถานะการฉีดน้ำทำงาน(LED สีน้ำเงิน)
OP00010	พัดลม 220 V ทำงาน
OP00001	หลอดไฟ 220V ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและวิจารณ์

5.1 สรุปผลการดำเนินงาน

โครงการนี้เป็นโครงการที่เกี่ยวกับทำแบบจำลองโรงเลี้ยงพืชด้วยคอมพิวเตอร์ ที่จะสามารถควบคุมและแสดงผลได้ทั้งในตัวเครื่องและที่หน้าจอคอมพิวเตอร์ การแสดงผลเป็นอุณหภูมิและความชื้น และการควบคุมจะสามารถควบคุม หลอด LED เพื่อแสดงสถานะ หลอด 220 V พัดลม 220 V และ Motor จะสามารถควบคุมได้จาก ที่คอมพิวเตอร์ได้ทีเดียว

จากการดำเนินงานสามารถควบคุมและแสดงผลได้ตามที่ออกแบบไว้

5.2 ปัญหาที่เกิดขึ้นและแนวทางแก้ไข

ปัญหา

วงจรที่นำไปขับ LED เพื่อแสดงสถานะ Pump ต่างๆ นั้น ไม่สามารถนำไปขับ ปั๊มน้ำได้จริง สามารถนำไปขับหลอด LED ได้เพียงอย่างเดียว แต่วงจร หลอดไฟ 220V และพัดลม 220 V สามารถนำไปใช้ได้จริง

แนวทางการแก้ไข

ทำวงจร วงจร โซลิดสเตทรีเลย์ (Solid State Relay) ที่ใช้กับหลอดไฟและพัดลม 220 V นำมาใช้แทนวงจรเดิม

5.3 แนวทางพัฒนาในอนาคต

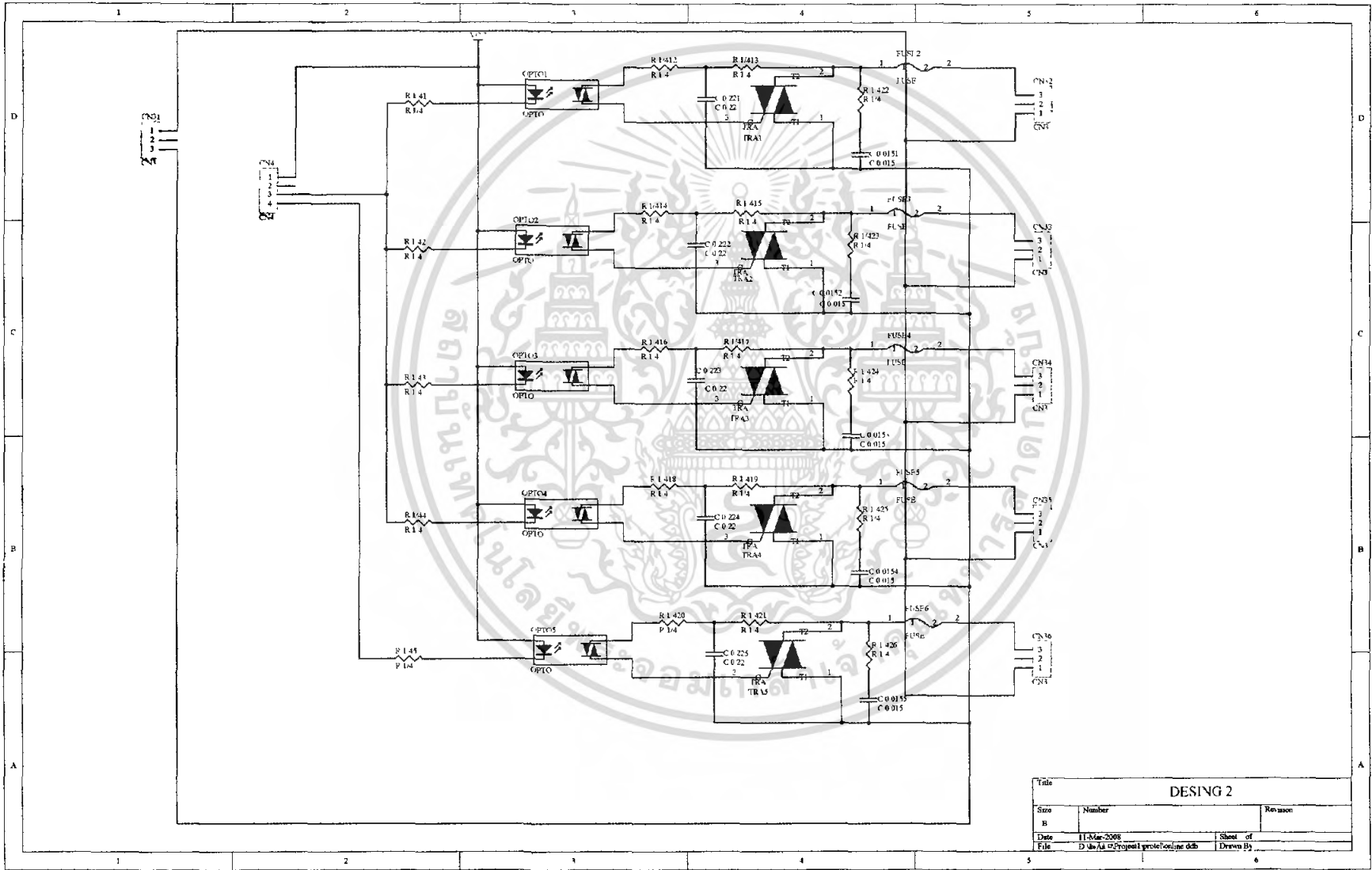
1. ทำให้สามารถไปใช้งานได้จริงที่โรงเพาะปลูก
2. เพิ่มเซ็นเซอร์อุณหภูมิและความชื้นให้มากขึ้น จะได้ตรวจเช็คได้หลายจุด
3. ทำให้สามารถควบคุมผ่านเว็บไซต์ได้

บรรณานุกรม

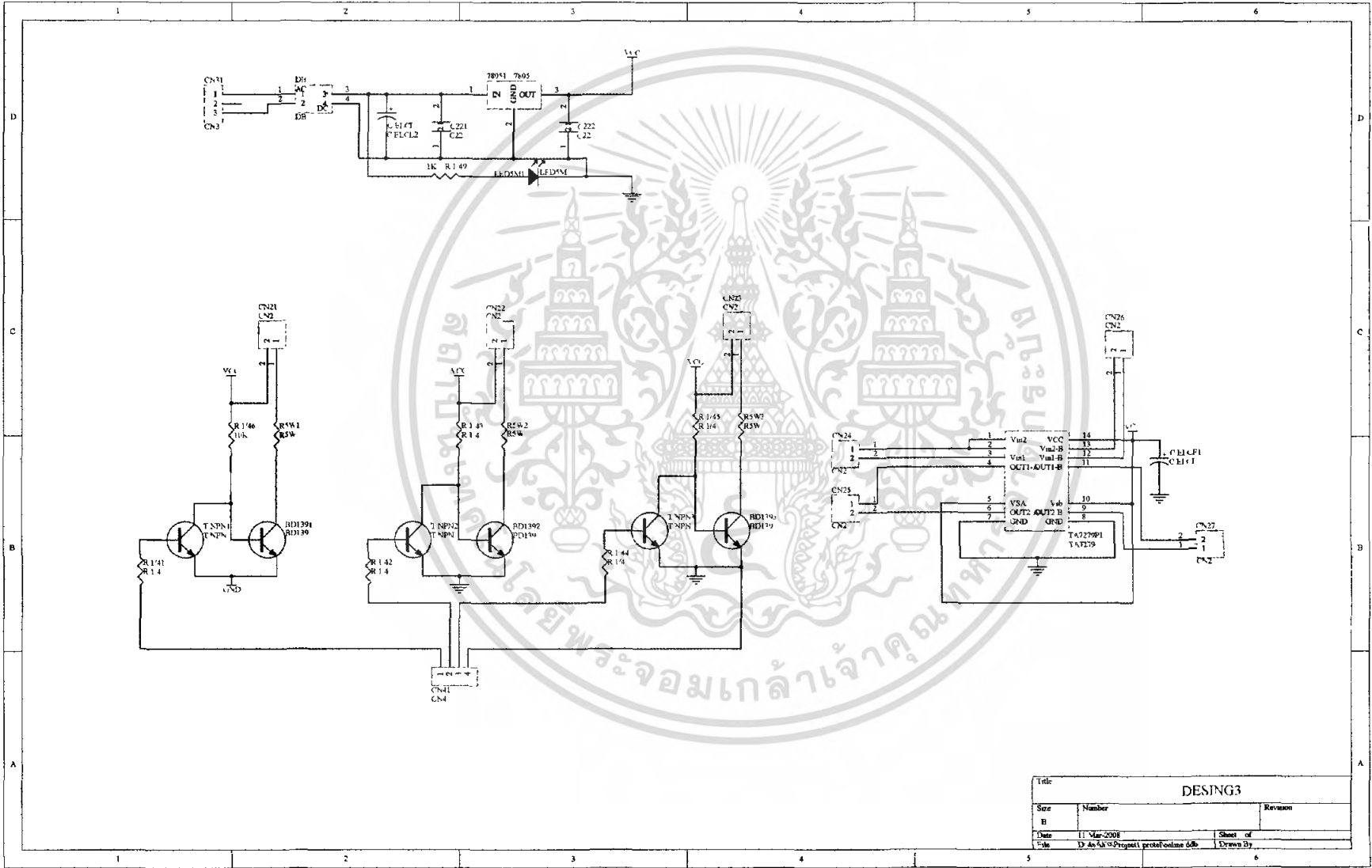
- [1] ประจัน พลังสันติสุข, “การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ AVR ด้วยภาษา C กับ WinAVR(C Compiler)”, แอพซอพพ์เนท, 2549
- [2] อ.ขจร อนุคิดย์, “การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ MCS-51 ด้วยภาษา C”, บริษัท เอ-บุ๊ก ดิสทริบิวชัน จำกัด, 2550
- [3] นคร ภัคศิชาติ, “การทดลองและใช้งานไมโครคอนโทรลเลอร์ MCS-51 ด้วยโปรแกรมภาษา C ฉบับ AT89Cx051”, บริษัท อิน โนเวตีฟ เอ็กเพอริเมนต์ จำกัด, 2549
- [4] ชีรบูลย์ หล่อวิเชียรรุ่ง, “คู่มือการเขียนโปรแกรมภาษา C เบื้องต้น สำหรับไมโครคอนโทรลเลอร์ MCS-51”, บริษัท อิน โนเวตีฟ เอ็กเพอริเมนต์ จำกัด, 2549
- [5] อุดม รานอก, “ภาษา C สำหรับงานควบคุมไมโครคอนโทรลเลอร์ MCS-51”, บริษัท ไอดีซี ดิสทริบิวเตอร์ เซ็นเตอร์ จำกัด, 2548
- [6] อภิชาติ กุฬลัป, “เริ่มต้นเขียนโปรแกรมติดต่อและควบคุมฮาร์ดแวร์ด้วย Visual Basic”, บริษัท เอ-บุ๊ก ดิสทริบิวชัน จำกัด, 2546
- [7] สัจจะ จรัสรุ่งรวีร์, “คู่มือ Visual C# 2005 ฉบับสมบูรณ์”, บริษัท เอ-บุ๊ก ดิสทริบิวชัน จำกัด, 2550



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title		
DESING 2		
Size	Number	Revision
R		
Date	11-May-2006	Sheet of
File	D:\u1a\2\project1\proj1\online.dcb	Drawn By



Title: DESING3		
Size: B	Number:	Revision:
Date: 11 Mar 2008	Sheet of	
File: D:\Asst\Project\procal\online\dd\	Drawn By:	

Code P89C51RD2

```
#include <REG51F.H>
#include <stdio.h>
#include <math.h>
#include "TCD.h"
#include "comport.h"
#include "ads7841.h"
////////// input
sbit SW1=P1^0;
sbit SW2=P1^1;
sbit SW3=P1^2;
sbit SW4=P1^3;
sbit SW5=P1^4;
unsigned char SW[6];
unsigned char SWO[6]; // old status
////////// Output
sbit Lo1=P2^7;
sbit Lo2=P2^6;
sbit Lo3=P2^5;
sbit Lo4=P2^4;
sbit Lo5=P2^3;
////////// Motor
sbit Mo1=P2^2;
sbit Mo2=P2^1;
//////////
sbit LDR=P1^6; // LDR
sbit BL_LCD=P1^5; // backlight LCD
sbit Buzzer=P1^7;
unsigned char LdrCon=0;
// unsigned int Temp;
unsigned char
Convert[10]={0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39};
// 0-9
unsigned int Ttemp = 0,Ttempl = 0;
unsigned int Htemp = 0,Htempl = 0;
unsigned int al,te;
float av,rv;
void ConLdr(void)
{
    if(LdrCon==1)
    {
        if(LDR==1) // dark
        {
            Lo4=0; // on lo4
        }
        else // light
        {
            Lo4=1; // off lo4
        }
    }
    else
    {
        if(SW[4]=='1') // on
        {
            Lo4=0; // on
            // printf("D");
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
{
    Lo4=1;    //// off
    // printf("I");
}
}

void ReadSwich(void)
{
    //////////////// input
    SW1=1;
    SW2=1;
    SW3=1;
    SW4=1;
    SW5=1;
    /* SW 1 on */
    if(SW1==0) {
        DISBUF[18]='1';
        SW[1]='1';
    }
    else { DISBUF[18]='0'; /* SW 1 off */
        SW[1]='0';
    }
    /* SW 2 on */
    if(SW2==0) {
        DISBUF[19]='1';
        SW[2]='1';
    }
    else {
        DISBUF[19]='0'; /* SW 2 off */
        SW[2]='0';
    }
    /* SW 3 on */
    if(SW3==0) {
        DISBUF[20]='1';
        SW[3]='1';
    }
    else {
        DISBUF[20]='0'; /* SW 3 off */
        SW[3]='0';
    }
    /* SW 4 on */
    if(SW4==0) {
        DISBUF[21]='1';
        SW[4]='1';
    }
    else {
        DISBUF[21]='0'; /* SW 4 off */
        SW[4]='0';
    }
    /* SW 5 on */
    if(SW5==0) {
        DISBUF[22]='1';
        SW[5]='1';
    }
    else {
        DISBUF[22]='0'; /* SW 5 off */
        SW[5]='0';
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    //////////////// control output
    ////// check SW1 change
    if(SW[1]!=SWO[1]){ SWO[1]=SW[1]; ////// new record

                                if(SW[1]=='1') ////// on
                                {
        Lo1=0;                    ////// on
                                //printf("A");
                                }
                                else
                                {
        Lo1=1; ////// off
                                //printf("F");
                                }
                                }
    ////// check SW2 change
    if(SW[2]!=SWO[2]){ SWO[2]=SW[2]; ////// new record
                                if(SW[2]=='1') ////// on
                                {
        Lo2=0;                    ////// on
                                // printf("B");
                                }
                                else
                                {
        Lo2=1; ////// off
                                // printf("G");
                                }
                                }
    ////// check SW3 change
    if(SW[3]!=SWO[3]){ SWO[3]=SW[3]; ////// new record
                                if(SW[3]=='1') ////// on
                                {
        Lo3=0;                    ////// on
                                //printf("C");
                                }
                                else
                                {
        Lo3=1; ////// off
                                // printf("H");
                                }
                                }
    ////// check SW4 change
    if(SW[4]!=SWO[4]){ SWO[4]=SW[4]; ////// new record
                                ConLdr();
                                }
    ////// check SW5 change
    if(SW[5]!=SWO[5]){ SWO[5]=SW[5]; ////// new record

                                if(SW[5]=='1') ////// on
                                {
        Lo5=0;                    //printf("E");
                                }
                                else ////// off
                                {
        Lo5=1;
                                //printf("J");
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้จัดทำนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
}
//////////
}
void KeyControl(char ky)
{
  if(ky =='A') ///// load 1 On
  {
    Lo1=0;
    DISBUF[27]='1'; /// on
    return;
  }
  if(ky =='B') ///// load 2 On
  {
    Lo2=0;
    DISBUF[28]='1'; /// on
    return;
  }
  if(ky =='C') ///// load 3 On
  {
    Lo3=0;
    DISBUF[29]='1'; /// on
    return;
  }
  if(ky =='D') ///// load 4 On
  {
    Lo4=0;
    DISBUF[30]='1'; /// on
    return;
  }
  if(ky =='E') ///// load 5 On
  {
    Lo5=0;
    DISBUF[31]='1'; /// on
    return;
  }
  //////////// command for off
  if(ky =='F') ///// load 1 Off
  {
    Lo1=1;
    DISBUF[27]='0'; /// on
    return;
  }
  if(ky ==-'G') ///// load 2 Off
  {
    Lo2=1;
    DISBUF[28]='0'; /// off
    return;
  }
  if(ky =='H') ///// load 3 Off
  {
    Lo3=1;
    DISBUF[29]='0'; /// off
    return;
  }
  if(ky =='I') ///// load 4 Off

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Lo4=1;
        DISBUF[30]='0'; /// off
        return;
    }
    if(ky =='J') ///// load 5 Off
    {
        Lo5=1;
        DISBUF[31]='0'; /// off
        return;
    }
    //////////////////////////////////// motor
    if(ky =='+')
    {
        Mo1=1;
        Mo2=0;
        return;
    }
    if(ky =='-')
    {
        Mo1=0;
        Mo2=1;
        return;
    }
    if(ky =='S')
    {
        Mo1=0;
        Mo2=0;
        return;
    }
    //////////////////////////////////// T H
    if(ky =='T') ///// show T
    {
        printf("%c%c.%c%c", DISBUF[1], DISBUF[2], DISBUF[4], DISBUF[5]);
        return;
    }

    if(ky =='Z') ///// show H
    {
        printf("%c%c.%c%c", DISBUF[10], DISBUF[11], DISBUF[13], DISBUF[14]);
        ;
        return;
    }

    if(ky =='t') /////test t
    {
        al=analog(2);
        av=al*0.001221;
        rv= al*0.1221;
        printf("AD = \t%d Step \t = %fV \t T = %f C\n",al,av,al-
675,rv);
        return;
    }
    if(ky =='h') /////test h
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        av=al*0.001221;
        rv=(al-675)*0.039832;
        printf("AD = \t%d Step \t = %fV \t offset = %d \t H = %f
%%\n",al,av,al-675,rv);
        return;
    }
    if(ky == 'Y') ///// on from LDR
    {
        LdrCon=1;    // control on by LDR
        ConLdr();
        return;
    }
    if(ky == 'X') ///// off
    {
        LdrCon=0;    // control off by LDR
        ConLdr();

        return;
    }
}
void ReadCommand(void)
{
    unsigned char c;
    if(RI)
    {
        c=SBUF;
        RI=0;
        KeyControl(c);
    }
}
void ReadDHIH4000(void) ///// hi
{
    unsigned int t;
    unsigned int ADCBUF[3];
    /*
    12 bit = 0-4095
    1 step = 5/4095 = 0.001221V
    HIH4000 zero offset = 0.824V = 675 step
    RH 100 % = 3.8895V - 3185.5 step
    *****
    1% = 25.105 step
    0.1% = 2.5105 step
    0.01% = 0.25105 step
    1 step = 0.03983270 %
    up * 100 for show lcd = 3.9832
    */
    /// h= (analog(3)-675) * 3.9832;
    /////////////// ac adc
    ADCBUF[2] =(analog(3)-675)* 3.9832;
    DelayMs(10);
    ADCBUF[1] = (analog(3)-675)* 3.9832;
    DelayMs(10);
    ADCBUF[0] = (analog(3)-675)* 3.9832;

    Htemp = ( ADCBUF[2]+ADCBUF[1]+ ADCBUF[0] ) / 3;
    if(abs(Htemp-Htempl)>50)    ///// more 1%

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        Htemp1=Htemp;          //////////////// set old temp

        //DISBUF[10]=Convert [Htemp/10000];
        t=Htemp%10000;
        DISBUF[10]=Convert [t/1000];
        t=t%1000;
        DISBUF[11]=Convert [t/100];
        t=t%100;
        DISBUF[13]=Convert [t/10];
        DISBUF[14]=Convert [t%10];
    }
}
void ReadLm35(void) ///// temp
{
    unsigned int t;
    unsigned int ADCBUF[3];
    /*
    temp - 10 mV / 1C
    12 bit = 0-4095
    1 step - 5/4095 = 0.001221V
    1 step = 0.1221 C
    up * 100 for show lcd = 12.21
    */
    ADCBUF[2] =analog(2)* 12.21;
    DelayMs(10);
    ADCBUF[1] = analog(2)* 12.21;
    DelayMs(10);
    ADCBUF[0] = analog(2)* 12.21;
    Ttemp = ( ADCBUF[2]+ADCBUF[1]+ ADCBUF[0] ) / 3;
    Ttemp =Ttemp-3; ////// jun
    if(abs(Ttemp-Ttemp1)>50) ////// more 1.8c
    {
        Ttemp1=Ttemp;          //////////////// set old temp
        //DISBUF[10]=Convert [Ttemp/10000];
        t=Ttemp%10000;
        DISBUF[1]=Convert [t/1000];
        t=t%1000;
        DISBUF[2]=Convert [t/100];
        t=t%100;
        DISBUF[4]=Convert [t/10];
        DISBUF[5]=Convert [t%10];
    }
}

void LdrControl(void)
{
    //////////////// control load and back light
    if(LDR==1) /// dark
    {
        BL_LCD=1; /// on back light LCD
    }
    else ///// light
    {
        BL_LCD=0; /// off back light LCD
    }
}

void CheckErr(void)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        //////////////// buzzer alarm
        //check trmp
        if(Ttemp >=4000 )          //temp over 40      C
        {
            Buzzer=1;  //on
            DelayMs(1000);
            Buzzer=0;    //off
            DelayMs(1000);
        }
        else if(Ttemp <=500 ) //temp down 5 C
        {
            Buzzer=1;  //on
            DelayMs(100);
            Buzzer=0;    //off
            DelayMs(1000);
        }
        else
        {
            Buzzer=0; //off
        }
        //////////////// check hi
        if(Htemp >=9000 ) //hi > 90%
        {
            Buzzer=1;  //on
            DelayMs(500);
            Buzzer=0;    //off
            DelayMs(1000);
        }
        else if(Htemp<=500 ) //hi down < 5%
        {
            Buzzer=1;  //on
            DelayMs(800);
            Buzzer=0;    //off
            DelayMs(1000);
        }
        else
        {
            Buzzer=0; //off
        }
    }
}
void main (void)
{
    P1=0xff;
    P2=0xff;
    LCD_Init();
    P1=0x00; //Clear Data
    Clear_DISBUF();
    //////////////// show t h
    DISBUF[0]='T';
    DISBUF[1]=' ';
    DISBUF[2]=' ';
    DISBUF[3]='.';
    DISBUF[4]=' ';
    DISBUF[5]=' ';
    DISBUF[6]=0x13;
    DISBUF[7]='C';
    DISBUF[8]='';
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DISBUF[9]='H';
DISBUF[10]=' ';
DISBUF[11]=' ';
DISBUF[12]='.';
DISBUF[13]=' ';
DISBUF[14]=' ';
DISBUF[15]='&';
////////// show 10
DISBUF[16]='I';
DISBUF[17]='P';
DISBUF[18]=' '; /// SW 1
DISBUF[19]=' '; /// SW 2
DISBUF[20]=' '; /// SW 3
DISBUF[21]=' '; /// SW 4
DISBUF[22]=' '; /// SW 5
DISBUF[23]=' ';
DISBUF[24]=' ';
DISBUF[25]='O';
DISBUF[26]='P';
DISBUF[27]='0'; //// Output 1
DISBUF[28]='0'; //// Output 2
DISBUF[29]='0'; //// Output 3
DISBUF[30]='0'; //// Output 4
DISBUF[31]='0'; //// Output 5
Lo1=1;
Lo2=1;
Lo3=1;
Lo4=1;
Lo5=1;
//old T=T;
//old H=H;
SWO[1]=SW[1]=0;
SWO[2]=SW[2]=0;
SWO[3]=SW[3]=0;
SWO[4]=SW[4]=0;
SWO[5]=SW[5]=0;
//////////
LDR=1; /// input check LDR
BL_LCD=0; /// off back light LCD
Buzzer=0; /// buzzer off
LCD_Display();
InitSerial();
while(1)
{
  ReadSwich();
  //sprintf(DISBUF, "asasas");
  ReadLm35(); /// temp
  ReadDHIH4000(); /// hi
  DelayMs(10);
  LdrControl();
  CheckErr();
  /*****/
  LCD_Display();
  ReadCommand();
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code C#

```
#region Namespace Inclusions
using System;
using System.Data;
using System.Text;
using System.Drawing;
using System.IO.Ports;
using System.Windows.Forms;
using System.ComponentModel;
using System.Collections.Generic;
using WebCamLib;
using System.Runtime.InteropServices;
using SerialPortTerminal.Properties;
using WIALib;
#endregion
namespace SerialPortTerminal
{
    #region Public Enumerations
    public enum DataMode { Text, Hex }
    public enum LogMsgType { Incoming, Outgoing, Normal, Warning, Error };
    #endregion
    public partial class frmTerminal : Form
    {
        #region Local Variables
        // The main control for communicating through the RS-232 port
        private SerialPort comport = new SerialPort();
        // Various colors for logging info
        private Color[] LogMsgTypeColor = { Color.Blue, Color.Green,
        Color.Black, Color.Orange, Color.Red };
        // Temp holder for whether a key was pressed
        private bool KeyHandled = false;
        #endregion
        #region Constructor
        public frmTerminal()
        {
            // Build the form
            InitializeComponent();
            // Restore the users settings
            InitializeControlValues();
            // Enable/disable controls based on the current state
            EnableControls();
            // When data is recieved through the port, call this method
            comport.DataReceived += new
            SerialPortReceivedEventHandler(port_DataReceived);
        }
        #endregion
        #region Local Methods

        /// <summary> Save the user's settings. </summary>
        private void SaveSettings()
        {
            Settings.Default.BaudRate = int.Parse(cmbBaudRate.Text);
            Settings.Default.DataBits = int.Parse(cmbDataBits.Text);
            Settings.Default.DataMode = CurrentDataMode;
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Settings.Default.Parity = (Parity)Enum.Parse(typeof(Parity),
cmbParity.Text);
        Settings.Default.StopBits =
(StopBits)Enum.Parse(typeof(StopBits), cmbStopBits.Text);
        Settings.Default.PortName = cmbPortName.Text;

        Settings.Default.Save();
    }

    /// <summary> Populate the form's controls with default settings.
</summary>
    private void InitializeControlValues()
    {
        cmbParity.Items.Clear();
cmbParity.Items.AddRange(Enum.GetNames(typeof(Parity)));
        cmbStopBits.Items.Clear();
cmbStopBits.Items.AddRange(Enum.GetNames(typeof(StopBits)));

        cmbParity.Text = Settings.Default.Parity.ToString();
cmbStopBits.Text = Settings.Default.StopBits.ToString();
cmbDataBits.Text = Settings.Default.DataBits.ToString();
cmbParity.Text = Settings.Default.Parity.ToString();
cmbBaudRate.Text = Settings.Default.BaudRate.ToString();
CurrentDataMode = Settings.Default.DataMode;

        cmbPortName.Items.Clear();
        foreach (string s in SerialPort.GetPortNames())
            cmbPortName.Items.Add(s);

        if (cmbPortName.Items.Contains(Settings.Default.PortName))
cmbPortName.Text = Settings.Default.PortName;
        else if (cmbPortName.Items.Count > 0) cmbPortName.SelectedIndex
= 0;
        else
        {
            MessageBox.Show(this, "There are no COM Ports detected on
this computer.\nPlease install a COM Port and restart this app.", "No
COM Ports Installed", MessageBoxButtons.OK, MessageBoxIcon.Error);
            this.Close();
        }
    }

    /// <summary> Enable/disable controls based on the app's current
state. </summary>
    private void EnableControls()
    {
        // Enable/disable controls based on whether the port is open or
not
        gbPortSettings.Enabled = !comport.IsOpen;

        if (comport.IsOpen) btnOpenPort.Text = "&Close Port";
        else btnOpenPort.Text = "&Open Port";
    }

    /// <summary> Send the user's data currently entered in the
'send' box.</summary>
    private void SendData()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        if (CurrentDataMode == DataMode.Text)
        {

        }
        else
        {
            try
            {

            }
            catch (FormatException)
            {

            }
        }
    }

    /// <summary> Log data to the terminal window. </summary>
    /// <param name="msgtype"> The type of message to be written.
</param>
    /// <param name="msg"> The string containing the message to be
    shown. </param>
    private void Log(LogMsgType msgtype, string msg)
    {
        rtfTerminal.Invoke(new EventHandler(delegate
        {
            rtfTerminal.SelectedText = string.Empty;
            rtfTerminal.SelectionFont = new
            Font(rtfTerminal.SelectionFont, FontStyle.Bold);
            rtfTerminal.SelectionColor = LogMsgTypeColor[(int)msgtype];
            rtfTerminal.AppendText(msg);
            rtfTerminal.ScrollToCaret();
        }));
    }

    private byte[] HexStringToByteArray(string s)
    {
        s = s.Replace(" ", "");
        byte[] buffer = new byte[s.Length / 2];
        for (int i = 0; i < s.Length; i += 2)
            buffer[i / 2] = (byte)Convert.ToByte(s.Substring(i, 2), 16);
        return buffer;
    }

    /// <summary> Converts an array of bytes into a formatted string
    of hex digits (ex: E4 CA B2)</summary>
    /// <param name="data"> The array of bytes to be translated into
    a string of hex digits. </param>
    /// <returns> Returns a well formatted string of hex digits with
    spacing. </returns>
    private string ByteArrayToHexString(byte[] data)
    {
        StringBuilder sb = new StringBuilder(data.Length * 3);
        foreach (byte b in data)
            sb.Append(Convert.ToString(b, 16).PadLeft(2, '0').PadRight(3,
            ' '));
        return sb.ToString().ToUpper();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#endregion
#region Local Properties
private DataMode CurrentDataMode
{
    get
    {
        if (rbHex.Checked) return DataMode.Hex;
        else return DataMode.Text;
    }
    set
    {
        if (value == DataMode.Text) rbText.Checked = true;
        else rbHex.Checked = true;
    }
}
#endregion
#region Event Handlers
private void lnkAbout_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    // Show the user the about dialog
    (new frmAbout()).ShowDialog(this);
}

private void frmTerminal_Shown(object sender, EventArgs e)
{
    Log(LogMsgType.Normal, String.Format("Application {0}\n",
DateTime.Now));
}
private void frmTerminal_FormClosing(object sender,
FormClosingEventArgs e)
{
    // The form is closing, save the user's preferences
    SaveSettings();
}
private void rbText_CheckedChanged(object sender, EventArgs e)
{ if (rbText.Checked) CurrentDataMode = DataMode.Text; }
private void rbHex_CheckedChanged(object sender, EventArgs e)
{ if (rbHex.Checked) CurrentDataMode = DataMode.Hex; }

private void cmbBaudRate_Validating(object sender,
CancelEventArgs e)
{ int x; e.Cancel = !int.TryParse(cmbBaudRate.Text, out x); }
private void cmbDataBits_Validating(object sender,
CancelEventArgs e)
{ int x; e.Cancel = !int.TryParse(cmbDataBits.Text, out x); }

private void btnOpenPort_Click(object sender, EventArgs e)
{
    // If the port is open, close it.
    if (comport.IsOpen) comport.Close();
    else
    {
        // Set the port's settings
        comport.BaudRate = int.Parse(cmbBaudRate.Text);
        comport.DataBits = int.Parse(cmbDataBits.Text);
        comport.StopBits = (StopBits)Enum.Parse(typeof(StopBits),
cmbStopBits.Text);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        comport.Parity = (Parity)Enum.Parse(typeof(Parity),
cmbParity.Text);
        comport.PortName = cmbPortName.Text;

        // Open the port
        comport.Open();
    }
    // Change the state of the form's controls
    EnableControls();

    // If the port is open, send focus to the send data box
    // if (comport.IsOpen) txtSendData.Focus();
}
private void port_DataReceived(object sender,
SerialDataReceivedEventArgs e)
{
    // This method will be called when there is data waiting in the
port's buffer
    // Determain which mode (string or binary) the user is in
    if (CurrentDataMode == DataMode.Text)
    {
        // Read all the data waiting in the buffer
        string data = comport.ReadExisting();

        // Display the text to the user in the terminal
        Log(LogMsgType.Incoming, data);
    }
    else
    {
        // Obtain the number of bytes waiting in the port's buffer
        int bytes = comport.BytesToRead;

        // Create a byte array buffer to hold the incoming data
        byte[] buffer = new byte[bytes];

        // Read the data from the port and store it in our buffer
        comport.Read(buffer, 0, bytes);

        // Show the user the incoming data in hex format
        Log(LogMsgType.Incoming, ByteArrayToHexString(buffer));
    }
}
private void txtSendData_KeyDown(object sender, KeyEventArgs e)
{
    // If the user presses [ENTER], send the data now
    if (KeyHandled = e.KeyCode == Keys.Enter) { e.Handled = true;
SendData(); }
}
private void txtSendData_KeyPress(object sender,
KeyPressEventArgs e)
{ e.Handled = KeyHandled; }
#endregion
private void frmTerminal_Load(object sender, EventArgs e)
{
    //.....Test Webcam.....//
    Device[] devices = DeviceManager.GetAllDevices();
    foreach (Device d in devices)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        cmddevice.Items.Add(d);
    }
    if (devices.Length > 0)
    {
        cmddevice.SelectedIndex = 0;
    }
    //.....End Test Webcam.....//
    timer1.Start();
    timer1.Enabled=true;
    this.label2.Text = "Pump 1";
    this.label3.Text = "Pump 2";
    this.label4.Text = "FAN";
    this.label5.Text = "SPORT";
    this.label6.Text = "Water Injector";
    this.button1.Text = "Lift";
    this.button16.Text = "STOP";
    this.button17.Text = "T";
    this.button2.Text = "Right";
    this.button3.Text = "ON";
    this.button4.Text = "OFF";
    this.button5.Text = "ON";
    this.button6.Text = "OFF";
    this.button7.Text = "ON";
    this.button8.Text = "OFF";
    this.button9.Text = "ON";
    this.button10.Text = "OFF";
    this.button11.Text = "ON";
    this.button12.Text = "OFF";
    this.pictureBox1.BackColor = Color.Black;
    this.pictureBox2.BackColor = Color.Black;
    this.pictureBox3.BackColor = Color.Black;
    this.pictureBox4.BackColor = Color.Black;
    this.pictureBox5.BackColor = Color.Black;
    this.pictureBox6.BackColor = Color.Black;
    label9.Text = "F";
    label10.Text = "G";
    label11.Text = "H";
    label12.Text = "I";
    label13.Text = "J";
}
private void lblSend_Click(object sender, EventArgs e)
{
}
private void button1_Click(object sender, EventArgs e)
{
    SerialPort porta = new SerialPort("COM4", 9600,
Parity.None, 8);
    porta.Open();
    porta.Write("+");
    Console.Beep(1000, 100);
    porta.Close();
}
private void button2_Click(object sender, EventArgs e)
{
    SerialPort porta = new SerialPort("COM4", 9600,
Parity.None, 8);
    porta.Open();
    porta.Write("-");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Console.Beep(1000, 100);
        porta.Close();
    }
    private void label2_Click(object sender, EventArgs e)
    {
    }
    private void label3_Click(object sender, EventArgs e)
    {
    }
    private void button3_Click(object sender, EventArgs e)
    {
        pictureBox2.BackColor = Color.Red;
        label9.Text = "A";
        SerialPort porta = new SerialPort("COM4", 9600,
Parity.None, 8);
        porta.Open();
        porta.Write("A");
        Console.Beep(1000, 100);
        porta.Close();
    }
    private void button4_Click(object sender, EventArgs e)
    {
        this.pictureBox2.BackColor = Color.Black;
        label9.Text = "F";
        SerialPort portf = new SerialPort("COM4", 9600,
Parity.None, 8);
        portf.Open();
        portf.Write("F");
        Console.Beep(500, 100);
        portf.Close();
    }
    private void pictureBox1_Click(object sender, EventArgs e)
    {
    }

    private void button6_Click(object sender, EventArgs e)
    {
        this.pictureBox3.BackColor = Color.Black;
        label10.Text = "G";
        SerialPort portg = new SerialPort("COM4", 9600,
Parity.None, 8);
        portg.Open();
        portg.Write("G");
        Console.Beep(1000, 100);
        portg.Close();
    }

    private void button5_Click(object sender, EventArgs e)
    {
        this.pictureBox3.BackColor = Color.Red;
        label10.Text = "B";
        SerialPort portb = new SerialPort("COM4", 9600, Parity.None,
8);
        portb.Open();
        portb.Write("B");
        Console.Beep(500, 100);
        portb.Close();
    }

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

private void groupBox2_Enter(object sender, EventArgs e)
{
}

private void button7_Click(object sender, EventArgs e)
{
    this.pictureBox4.BackColor = Color.Red;
    label11.Text = "C";
    SerialPort portb = new SerialPort("COM4", 9600,
Parity.None, 8);
    portb.Open();
    portb.Write("C");
    Console.Beep(500, 100);
    portb.Close();
}

private void button8_Click(object sender, EventArgs e)
{
    this.pictureBox4.BackColor = Color.Black;
    label11.Text = "H";
    SerialPort portg = new SerialPort("COM4", 9600,
Parity.None, 8);
    portg.Open();
    portg.Write("H");
    Console.Beep(1000, 100);
    portg.Close();
}

private void button9_Click(object sender, EventArgs e)
{
    this.pictureBox5.BackColor = Color.Red;
    label12.Text = "D";
    SerialPort portg = new SerialPort("COM4", 9600,
Parity.None, 8);
    portg.Open();
    portg.Write("D");
    Console.Beep(1000, 100);
    portg.Close();
}

private void button10_Click(object sender, EventArgs e)
{
    this.pictureBox5.BackColor = Color.Black;
    label12.Text = "I";
    SerialPort portg = new SerialPort("COM4", 9600,
Parity.None, 8);
    portg.Open();
    portg.Write("I");
    Console.Beep(1000, 100);
    portg.Close();
}

private void gbMode_Enter(object sender, EventArgs e)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    private void rtfTerminal_TextChanged(object sender, EventArgs
e)
    {
    }

    private void cmbBaudRate_SelectedIndexChanged(object sender,
EventArgs e)
    {
    }

    private void cmbParity_SelectedIndexChanged(object sender,
EventArgs e)
    {
    }

    private void button11_Click(object sender, EventArgs e)
    {
        this.pictureBox6.BackColor = Color.Red;
        label13.Text = "E";
        SerialPort portg = new SerialPort("COM4", 9600,
Parity.None, 8);
        portg.Open();
        portg.Write("E");
        Console.Beep(1000, 100);
        portg.Close();
    }

    private void button12_Click(object sender, EventArgs e)
    {
        this.pictureBox6.BackColor = Color.Black;
        label13.Text = "J";
        SerialPort portg = new SerialPort("COM4", 9600,
Parity.None, 8);
        portg.Open();
        portg.Write("J");
        Console.Beep(1000, 100);
        portg.Close();
    }

    private void label6_Click(object sender, EventArgs e)
    {
    }

    private void frmTerminal_MouseHover(object sender, EventArgs e)
    {
    }

    private void frmTerminal_MouseEnter(object sender, EventArgs e)
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private void timer1_Tick(object sender, EventArgs e)
{
    //.....Clock_Time.....//
    label8.Text = DateTime.Now.ToLongTimeString();

    //.....Date_DD/MM/YY.....//
    label7.Text = DateTime.Now.ToLongDateString();

    //.....PictureBox2.....//
    if (DateTime.Now.ToLongTimeString()
    ==textBox1.Text.ToString())
    {
        pictureBox2.BackColor = Color.Red;
        label9.Text = "A";
        SerialPort porta = new SerialPort("COM4", 9600,
    Parity.None, 8);
        porta.Open();
        porta.Write("A");
        porta.Close();
    }
    else if (DateTime.Now.ToLongTimeString() ==
    textBox2.Text.ToString())
    {
        pictureBox2.BackColor = Color.Black;
        label9.Text = "F";
        SerialPort portf = new SerialPort("COM4", 9600,
    Parity.None, 8);
        portf.Open();
        portf.Write("F");
        portf.Close();
    }
    //.....PictureBox3.....//
    if (DateTime.Now.ToLongTimeString() ==
    textBox3.Text.ToString())
    {
        pictureBox3.BackColor = Color.Red;
        label10.Text = "B";
        SerialPort portb = new SerialPort("COM4", 9600,
    Parity.None, 8);
        portb.Open();
        portb.Write("B");
        portb.Close();
    }
    else if (DateTime.Now.ToLongTimeString() ==
    textBox4.Text.ToString())
    {
        pictureBox3.BackColor = Color.Black;
        label10.Text = "C";
        SerialPort portg = new SerialPort("COM4", 9600,
    Parity.None, 8);
        portg.Open();
        portg.Write("G");
        portg.Close();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//.....PictureBox4.....//
if (DateTime.Now.ToLongTimeString() ==
textBox5.Text.ToString())
{
    pictureBox4.BackColor = Color.Red;
    label11.Text = "C";
}
else if (DateTime.Now.ToLongTimeString() ==
textBox6.Text.ToString())
{
    pictureBox4.BackColor = Color.Black;

    label11.Text = "H";
}

//.....PictureBox5.....//
if (DateTime.Now.ToLongTimeString() ==
textBox7.Text.ToString())
{
    pictureBox5.BackColor = Color.Red;
    label12.Text = "D";
}
else if (DateTime.Now.ToLongTimeString() ==
textBox8.Text.ToString())
{
    pictureBox5.BackColor = Color.Black;
    label12.Text = "I";
}

//.....PictureBox6.....//
if (DateTime.Now.ToLongTimeString() ==
textBox9.Text.ToString())
{
    pictureBox6.BackColor = Color.Red;
    label13.Text = "E";
}
else if (DateTime.Now.ToLongTimeString() ==
textBox10.Text.ToString())
{
    pictureBox6.BackColor = Color.Black;
    label13.Text = "J";
}

}

private void button13_Click(object sender, EventArgs e)
{
    Device d =
DeviceManager.GetDevice(cmddevice.SelectedIndex);
    d.ShowWindow(this.pictureBox1);
}

private void button14_Click(object sender, EventArgs e)
{
    Device d =
DeviceManager.GetDevice(cmddevice.SelectedIndex);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        d.Stop();
    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {

    }

    private void button15_Click(object sender, EventArgs e)
    {
        new CaptureForm().Show();
    }

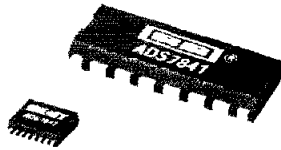
    private void button16_Click(object sender, EventArgs e)
    {
        SerialPort porta = new SerialPort("COM4", 9600,
Parity.None, 8);
        porta.Open();
        porta.Write("S");
        Console.Beep(1000, 100);
        porta.Close();
    }

    private void button17_Click(object sender, EventArgs e)
    {
        SerialPort porta = new SerialPort("COM4", 9600,
Parity.None, 8);
        porta.Open();
        porta.Write("T");
        Console.Beep(1000, 100);
        porta.Close();
    }

    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ADS7841

12-Bit, 4-Channel Serial Output Sampling ANALOG-TO-DIGITAL CONVERTER

FEATURES

- SINGLE SUPPLY: 2.7V to 5V
- 4-CHANNEL SINGLE-ENDED OR 2-CHANNEL DIFFERENTIAL INPUT
- UP TO 200kHz CONVERSION RATE
- ± 1 LSB MAX INL AND DNL
- GUARANTEED NO MISSING CODES
- 72dB SINAD
- SERIAL INTERFACE
- 16-PIN PDIP OR 16-LEAD SSOP PACKAGE
- ALTERNATE SOURCE FOR MAX1247

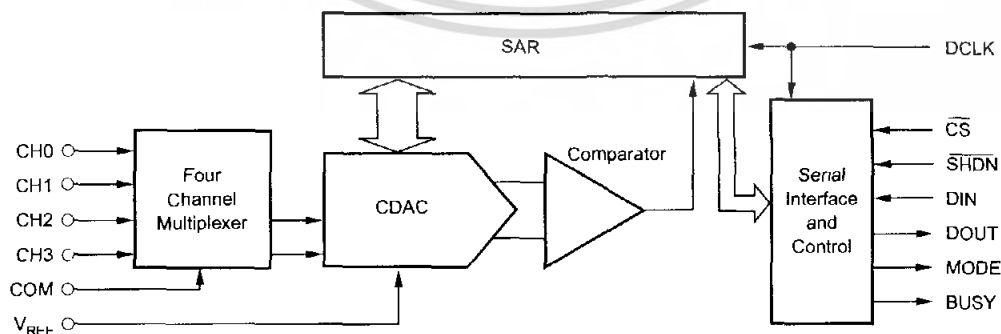
APPLICATIONS

- DATA ACQUISITION
- TEST AND MEASUREMENT
- INDUSTRIAL PROCESS CONTROL
- PERSONAL DIGITAL ASSISTANTS
- BATTERY-POWERED SYSTEMS

DESCRIPTION

The ADS7841 is a 4-channel, 12-bit sampling analog-to-digital converter (ADC) with a synchronous serial interface. The resolution is programmable to either 8 or 12 bits. Typical power dissipation is 2mW at a 200kHz throughput rate and a +5V supply. The reference voltage (V_{REF}) can be varied between 100mV and V_{CC} , providing a corresponding input voltage range of 0V to V_{REF} . The device includes a shutdown mode which reduces power dissipation to under 15 μ W. The ADS7841 is guaranteed down to 2.7V operation.

Low power, high speed, and on-board multiplexer make the ADS7841 ideal for battery operated systems such as personal digital assistants, portable multi-channel data loggers, and measurement equipment. The serial interface also provides low-cost isolation for remote data acquisition. The ADS7841 is available in a 16-pin PDIP or a 16-lead SSOP package and is guaranteed over the -40°C to $+85^{\circ}\text{C}$ temperature range.



International Airport Industrial Park • Mailing Address: PO Box 11400 Tucson, AZ 85734 • Street Address: 6730 S. Tucson Blvd., Tucson, AZ 85706 • Tel: (520) 746-1111 • Twx: 910-952-1111
Internet: <http://www.burr-brown.com/> • FAX Line: (800) 548-6133 (US/Canada Only) • Cable: BBRCORP • Telex: 066-6491 • FAX: (520) 889-1510 • Immediate Product Info: (800) 548-6132

SPECIFICATION: +5V

At $T_A = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$, $+V_{CC} = +5\text{V}$, $V_{REF} = +5\text{V}$, $f_{SAMPLE} = 200\text{kHz}$, and $f_{CLK} = 16 \cdot f_{SAMPLE} = 3.2\text{MHz}$, unless otherwise noted

PARAMETER	CONDITIONS	ADS7841E, P			ADS7841EB, PB			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
ANALOG INPUT								
Full-Scale Input Span	Positive Input - Negative Input	0		V_{REF}	*		*	V
Absolute Input Range	Positive Input	-0.2		$+V_{CC} + 0.2$	*		*	V
	Negative Input	-0.2		+1.25	*		*	V
Capacitance			25			*		pF
Leakage Current			± 1			*		μA
SYSTEM PERFORMANCE								
Resolution		12	12		12	*		Bits
No Missing Codes								Bits
Integral Linearity Error				± 2			± 1	LSB ⁽¹⁾
Differential Linearity Error			± 0.8			± 0.5	± 1	LSB
Offset Error				± 3			*	LSB
Offset Error Match			0.15	1.0		*	*	LSB
Gain Error				± 4			± 3	LSB
Gain Error Match			0.1	1.0		*	*	LSB
Noise			30			*		μVrms
Power Supply Rejection			70			*		dB
SAMPLING DYNAMICS								
Conversion Time				12			*	Clk Cycles
Acquisition Time		3			*			Clk Cycles
Throughput Rate				200			*	kHz
Multiplexer Settling Time			500			*		ns
Aperture Delay			30			*		ns
Aperture Jitter			100			*		ps
DYNAMIC CHARACTERISTICS								
Total Harmonic Distortion ⁽²⁾	$V_{IN} = 5\text{Vp-p}$ at 10kHz		-78	-72		-80	-76	dB
Signal-to-(Noise + Distortion)	$V_{IN} = 5\text{Vp-p}$ at 10kHz	68	71		70	72		dB
Spurious Free Dynamic Range	$V_{IN} = 5\text{Vp-p}$ at 10kHz	72	79		76	81		dB
Channel-to-Channel Isolation	$V_{IN} = 5\text{Vp-p}$ at 50kHz		120			*		dB
REFERENCE INPUT								
Range		0.1		$+V_{CC}$	*		*	V
Resistance	DCLK Static		5			*	*	$\text{G}\Omega$
Input Current	$f_{SAMPLE} = 12.5\text{kHz}$		40	100		*	*	μA
	DCLK Static		2.5	3		*	*	μA
DIGITAL INPUT/OUTPUT								
Logic Family			CMOS			*		
Logic Levels								
V_{IH}	$ I_{IH} \leq +5\mu\text{A}$	3.0		5.5	*		*	V
V_{IL}	$ I_{IL} \leq +5\mu\text{A}$	-0.3		+0.8	*		*	V
V_{OH}	$I_{OH} = -250\mu\text{A}$	3.5			*		*	V
V_{OL}	$I_{OL} = 250\mu\text{A}$			0.4			*	V
Data Format			Straight Binary			*		
POWER SUPPLY REQUIREMENTS								
$+V_{CC}$	Specified Performance	4.75		5.25	*		*	V
Quiescent Current			550	900			*	μA
	$f_{SAMPLE} = 12.5\text{kHz}$		300			*		μA
Power Dissipation	Power-Down Mode ⁽³⁾ , $\overline{CS} = +V_{CC}$			3			*	μA
				4.5			*	mW
TEMPERATURE RANGE								
Specified Performance		-40		+85	*		*	$^{\circ}\text{C}$

* Same specifications as ADS7841E, P

NOTE (1) LSB means Least Significant Bit. With V_{REF} equal to +5.0V, one LSB is 1.22mV. (2) First five harmonics of the test frequency. (3) Auto power-down mode (PD1 = PD0 = 0) active or $\overline{SHDN} = \text{GND}$

The information provided herein is believed to be reliable, however, BURR-BROWN assumes no responsibility for inaccuracies or omissions. BURR-BROWN assumes no responsibility for the use of this information and all use of such information shall be entirely at the user's own risk. Prices and specifications are subject to change without notice. No patent rights or licenses to any of the circuits described herein are implied or granted to any third party. BURR-BROWN does not authorize or warrant any BURR-BROWN product for use in life support devices and/or systems.



นี่เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ADS7841

2

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPECIFICATION: +2.7V

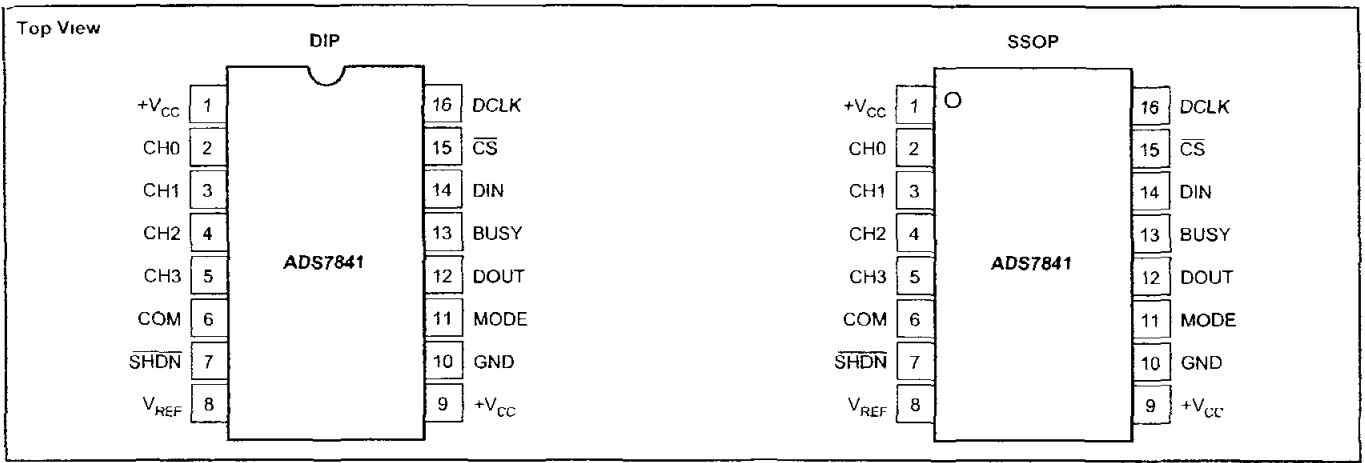
At $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$, $+V_{CC} = +2.7\text{V}$, $V_{REF} = +2.5\text{V}$, $f_{SAMPLE} = 125\text{kHz}$, and $f_{CLK} = 16 \cdot f_{SAMPLE} = 2\text{MHz}$, unless otherwise noted

PARAMETER	CONDITIONS	ADS7841E, P			ADS7841EB, PB			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
ANALOG INPUT								
Full-Scale Input Span	Positive Input - Negative Input	0		V_{REF}	*		*	V
Absolute Input Range	Positive Input	-0.2		$+V_{CC} + 0.2$	*		*	V
	Negative Input	-0.2		+0.2	*		*	V
Capacitance			25			*		pF
Leakage Current			± 1			*		μA
SYSTEM PERFORMANCE								
Resolution		12	12		12	*		Bits
No Missing Codes		12			12			Bits
Integral Linearity Error				± 2			± 1	LSB ⁽¹⁾
Differential Linearity Error			± 0.8			± 0.5	± 1	LSB
Offset Error				± 3			*	LSB
Offset Error Match			0.15	1.0		*	*	LSB
Gain Error				± 4			± 3	LSB
Gain Error Match			0.1	1.0		*	*	LSB
Noise			30			*		μV_{rms}
Power Supply Rejection			70			*		dB
SAMPLING DYNAMICS								
Conversion Time				12			*	Clk Cycles
Acquisition Time		3			*			Clk Cycles
Throughput Rate				125			*	kHz
Multiplexer Settling Time			500			*		ns
Aperture Delay			30			*		ns
Aperture Jitter			100			*		ps
DYNAMIC CHARACTERISTICS								
Total Harmonic Distortion ⁽²⁾	$V_{IN} = 2.5\text{V}_{p-p}$ at 10kHz		-77	-72		-79	-76	dB
Signal-to-(Noise + Distortion)	$V_{IN} = 2.5\text{V}_{p-p}$ at 10kHz	68	71		70	72		dB
Spurious Free Dynamic Range	$V_{IN} = 2.5\text{V}_{p-p}$ at 10kHz	72	78		76	80		dB
Channel-to-Channel Isolation	$V_{IN} = 2.5\text{V}_{p-p}$ at 50kHz		100			*		dB
REFERENCE INPUT								
Range		0.1		$+V_{CC}$	*		*	V
Resistance	DCLK Static		5			*	*	G Ω
Input Current	$f_{SAMPLE} = 12.5\text{kHz}$ DCLK Static		13	40		*	*	μA
			2.5			*	*	μA
			0.001	3		*	*	μA
DIGITAL INPUT/OUTPUT								
Logic Family			CMOS			*		
Logic Levels						*		
V_{IH}	$ I_{IH} \leq +5\mu\text{A}$	$+V_{CC} \cdot 0.7$		5.5	*		*	V
V_{IL}	$ I_{IL} \leq +5\mu\text{A}$	-0.3		+0.8	*		*	V
V_{OH}	$I_{OH} = -250\mu\text{A}$	$+V_{CC} \cdot 0.8$			*		*	V
V_{OL}	$I_{OL} = 250\mu\text{A}$			0.4			*	V
Data Format			Straight Binary			*		
POWER SUPPLY REQUIREMENTS								
$+V_{CC}$	Specified Performance	2.7		3.6	*		*	V
Quiescent Current			280	650		*	*	μA
	$f_{SAMPLE} = 12.5\text{kHz}$		220			*	*	μA
	Power-Down Mode ⁽³⁾ , $\overline{CS} = +V_{CC}$		3				*	μA
Power Dissipation				1.8			*	mW
TEMPERATURE RANGE								
Specified Performance		-40		+85	*		*	$^\circ\text{C}$

* Same specifications as ADS7841E, P

NOTE (1) LSB means Least Significant Bit. With V_{REF} equal to +2.5V, one LSB is 610mV. (2) First five harmonics of the test frequency. (3) Auto power-down mode (PD1 = PD0 = 0) active or SHDN = GND

PIN CONFIGURATIONS



PIN DESCRIPTIONS

PIN	NAME	DESCRIPTION
1	+V _{CC}	Power Supply, 2.7V to 5V
2	CH0	Analog Input Channel 0
3	CH1	Analog Input Channel 1
4	CH2	Analog Input Channel 2
5	CH3	Analog Input Channel 3
6	COM	Ground Reference for Analog Inputs. Sets zero code voltage in single-ended mode. Connect this pin to ground or ground reference point.
7	SHDN	Shutdown. When LOW, the device enters a very low power shutdown mode.
8	V _{REF}	Voltage Reference Input
9	+V _{CC}	Power Supply, 2.7V to 5V
10	GND	Ground
11	MODE	Conversion Mode. When LOW, the device always performs a 12-bit conversion. When HIGH, the resolution is set by the MODE bit in the CONTROL byte.
12	DOUT	Serial Data Output. Data is shifted on the falling edge of DCLK. This output is high impedance when CS is HIGH.
13	BUSY	Busy Output. This output is high impedance when CS is HIGH.
14	DIN	Serial Data Input. If CS is LOW, data is latched on rising edge of DCLK.
15	CS	Chip Select Input. Controls conversion timing and enables the serial input/output register.
16	DCLK	External Clock Input. This clock runs the SAR conversion process and synchronizes serial data I/O.

ABSOLUTE MAXIMUM RATINGS⁽¹⁾

+V _{CC} to GND	-0.3V to +6V
Analog Inputs to GND	-0.3V to +V _{CC} + 0.3V
Digital Inputs to GND	-0.3V to +6V
Power Dissipation	250mW
Maximum Junction Temperature	+150°C
Operating Temperature Range	-40°C to +85°C
Storage Temperature Range	-65°C to +150°C
Lead Temperature (soldering, 10s)	+300°C

NOTE (1) Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to absolute maximum conditions for extended periods may affect device reliability.



ELECTROSTATIC DISCHARGE SENSITIVITY

This integrated circuit can be damaged by ESD. Burr-Brown recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

PACKAGE/ORDERING INFORMATION

PRODUCT	MINIMUM RELATIVE ACCURACY (LSB)	DIFFERENTIAL NONLINEARITY (LSB)	SPECIFICATION TEMPERATURE RANGE	PACKAGE	PACKAGE DRAWING NUMBER ⁽¹⁾	ORDERING NUMBER ⁽²⁾	TRANSPORT MEDIA
ADS7841E	±2	68	-40°C to +85°C	16-Lead SSOP	322	ADS7841E	Rails
"	"	"	"	"	"	ADS7841E/2K5	Tape and Reel
ADS7841P	±2	68	-40°C to +85°C	16-Pin PDIP	180	ADS7841P	Rails
ADS7841EB	±1	70	-40°C to +85°C	16-Lead SSOP	322	ADS7841EB	Rails
"	"	"	"	"	"	ADS7841EB/2K5	Tape and Reel
ADS7841PB	±1	70	-40°C to +85°C	16-Pin PDIP	180	ADS7841PB	Rails

NOTES (1) For detailed drawing and dimension table, please see end of data sheet, or Appendix C of Burr-Brown IC Data Book. (2) Models with a slash (/) are available only in Tape and Reel in the quantities indicated (e.g., /2K5 indicates 2500 devices per reel). Ordering 2500 pieces of "ADS7841/2K5" will get a single 2500-piece Tape and Reel. For detailed Tape and Reel mechanical information, refer to Appendix B of Burr-Brown IC Data Book.

BURR-BROWN เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ADS7841

LM35

Precision Centigrade Temperature Sensors

General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of $\pm 1/4^\circ\text{C}$ at room temperature and $\pm 3/4^\circ\text{C}$ over a full -55 to $+150^\circ\text{C}$ temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only $60\ \mu\text{A}$ from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a -55° to $+150^\circ\text{C}$ temperature range, while the LM35C is rated for a -40° to $+110^\circ\text{C}$ range (-10° with improved accuracy). The LM35 series is available pack-

aged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear $+10.0\ \text{mV}/^\circ\text{C}$ scale factor
- 0.5°C accuracy guaranteeable (at $+25^\circ\text{C}$)
- Rated for full -55° to $+150^\circ\text{C}$ range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than $60\ \mu\text{A}$ current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only $\pm 1/4^\circ\text{C}$ typical
- Low impedance output, $0.1\ \Omega$ for $1\ \text{mA}$ load

Typical Applications

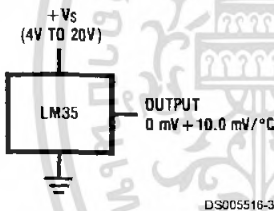
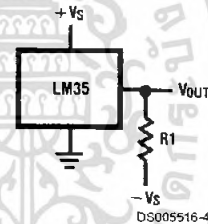


FIGURE 1. Basic Centigrade Temperature Sensor
($+2^\circ\text{C}$ to $+150^\circ\text{C}$)

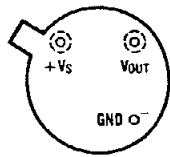


Choose $R_1 = -V_S/50\ \mu\text{A}$
 $V_{\text{OUT}} = +1,500\ \text{mV}$ at $+150^\circ\text{C}$
 $= +250\ \text{mV}$ at $+25^\circ\text{C}$
 $= -550\ \text{mV}$ at -55°C

FIGURE 2. Full-Range Centigrade Temperature Sensor

Connection Diagrams

**TO-46
Metal Can Package***

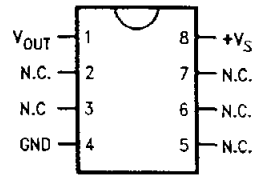


*Case is connected to negative pin (GND)

Order Number LM35H, LM35AH, LM35CH, LM35CAH or LM35DH

See NS Package Number H03H

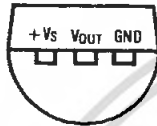
**SO-8
Small Outline Molded Package**



N C = No Connection

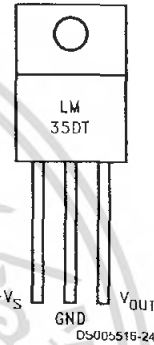
**Top View
Order Number LM35DM
See NS Package Number M08A**

**TO-92
Plastic Package**



**Order Number LM35CZ,
LM35CAZ or LM35DZ
See NS Package Number Z03A**

**TO-220
Plastic Package***



*Tab is connected to the negative pin (GND)

Note: The LM35DT pinout is different than the discontinued LM35DP.

**Order Number LM35DT
See NS Package Number TA03F**

Absolute Maximum Ratings (Note 10)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage	+35V to -0.2V
Output Voltage	+6V to -1.0V
Output Current	10 mA
Storage Temp.:	
TO-46 Package,	-60°C to +180°C
TO-92 Package,	-60°C to +150°C
SO-8 Package,	-65°C to +150°C
TO-220 Package,	-65°C to +150°C
Lead Temp.:	
TO-46 Package, (Soldering, 10 seconds)	300°C

TO-92 and TO-220 Package, (Soldering, 10 seconds)	260°C
SO Package (Note 12)	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 11)	2500V
Specified Operating Temperature Range: T_{MIN} to T_{MAX} (Note 2)	
LM35, LM35A	-55°C to +150°C
LM35C, LM35CA	-40°C to +110°C
LM35D	0°C to +100°C

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	$T_A = +25^\circ\text{C}$	± 0.2	± 0.5		± 0.2	± 0.5		°C
	$T_A = -10^\circ\text{C}$	± 0.3			± 0.3		± 1.0	°C
	$T_A = T_{MAX}$	± 0.4	± 1.0		± 0.4	± 1.0		°C
	$T_A = T_{MIN}$	± 0.4	± 1.0		± 0.4		± 1.5	°C
Nonlinearity (Note 8)	$T_{MIN} \leq T_A \leq T_{MAX}$	± 0.18		± 0.35	± 0.15		± 0.3	°C
Sensor Gain (Average Slope)	$T_{MIN} \leq T_A \leq T_{MAX}$	+10.0	+9.9, +10.1		+10.0		+9.9, +10.1	mV/°C
Load Regulation (Note 3) $0 \leq I_L \leq 1$ mA	$T_A = +25^\circ\text{C}$	± 0.4	± 1.0		± 0.4	± 1.0		mV/mA
	$T_{MIN} \leq T_A \leq T_{MAX}$	± 0.5		± 3.0	± 0.5		± 3.0	mV/mA
Line Regulation (Note 3)	$T_A = +25^\circ\text{C}$	± 0.01	± 0.05		± 0.01	± 0.05		mV/V
	$4V \leq V_S \leq 30V$	± 0.02		± 0.1	± 0.02		± 0.1	mV/V
Quiescent Current (Note 9)	$V_S = +5V, +25^\circ\text{C}$	56	67		56	67		μA
	$V_S = +5V$	105		131	91		114	μA
	$V_S = +30V, +25^\circ\text{C}$	56.2	68		56.2	68		μA
	$V_S = +30V$	105.5		133	91.5		116	μA
Change of Quiescent Current (Note 3)	$4V \leq V_S \leq 30V, +25^\circ\text{C}$	0.2	1.0		0.2	1.0		μA
	$4V \leq V_S \leq 30V$	0.5		2.0	0.5		2.0	μA
Temperature Coefficient of Quiescent Current		+0.39		+0.5	+0.39		+0.5	μA/°C
Minimum Temperature for Rated Accuracy	In circuit of <i>Figure 1</i> , $I_L = 0$	+1.5		+2.0	+1.5		+2.0	°C
Long Term Stability	$T_J = T_{MAX}$, for 1000 hours	± 0.08			± 0.08			°C

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35			LM35C, LM35D			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy, LM35, LM35C (Note 7)	$T_A = +25^\circ\text{C}$	± 0.4	± 1.0		± 0.4	± 1.0		$^\circ\text{C}$
	$T_A = -10^\circ\text{C}$	± 0.5			± 0.5		± 1.5	$^\circ\text{C}$
	$T_A = T_{\text{MAX}}$	± 0.8	± 1.5		± 0.8		± 1.5	$^\circ\text{C}$
	$T_A = T_{\text{MIN}}$	± 0.8		± 1.5	± 0.8		± 2.0	$^\circ\text{C}$
Accuracy, LM35D (Note 7)	$T_A = +25^\circ\text{C}$				± 0.6	± 1.5		$^\circ\text{C}$
	$T_A = T_{\text{MAX}}$				± 0.9		± 2.0	$^\circ\text{C}$
	$T_A = T_{\text{MIN}}$				± 0.9		± 2.0	$^\circ\text{C}$
Nonlinearity (Note 8)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	± 0.3		± 0.5	± 0.2		± 0.5	$^\circ\text{C}$
Sensor Gain (Average Slope)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	+10.0	+9.8, +10.2		+10.0		+9.8, +10.2	mV/ $^\circ\text{C}$
Load Regulation (Note 3) $0 \leq I_L \leq 1 \text{ mA}$	$T_A = +25^\circ\text{C}$	± 0.4	± 2.0		± 0.4	± 2.0		mV/mA
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	± 0.5		± 5.0	± 0.5		± 5.0	mV/mA
Line Regulation (Note 3)	$T_A = +25^\circ\text{C}$	± 0.01	± 0.1		± 0.01	± 0.1		mV/V
	$4\text{V} \leq V_S \leq 30\text{V}$	± 0.02		± 0.2	± 0.02		± 0.2	mV/V
Quiescent Current (Note 9)	$V_S = +5\text{V}, +25^\circ\text{C}$	56	80		56	80		μA
	$V_S = +5\text{V}$	105		158	91		138	μA
	$V_S = +30\text{V}, +25^\circ\text{C}$	56.2	82		56.2	82		μA
	$V_S = +30\text{V}$	105.5		161	91.5		141	μA
Change of Quiescent Current (Note 3)	$4\text{V} \leq V_S \leq 30\text{V}, +25^\circ\text{C}$	0.2	2.0		0.2	2.0		μA
	$4\text{V} \leq V_S \leq 30\text{V}$	0.5		3.0	0.5		3.0	μA
Temperature Coefficient of Quiescent Current		+0.39		+0.7	+0.39		+0.7	$\mu\text{A}/^\circ\text{C}$
Minimum Temperature for Rated Accuracy	In circuit of Figure 1, $I_L = 0$	+1.5		+2.0	+1.5		+2.0	$^\circ\text{C}$
Long Term Stability	$T_J = T_{\text{MAX}}$, for 1000 hours	± 0.08			± 0.08			$^\circ\text{C}$

Note 1: Unless otherwise noted, these specifications apply: $-55^\circ\text{C} \leq T_J \leq +150^\circ\text{C}$ for the LM35 and LM35A; $-40^\circ\text{C} \leq T_J \leq +110^\circ\text{C}$ for the LM35C and LM35CA, and $0^\circ\text{C} \leq T_J \leq +100^\circ\text{C}$ for the LM35D. $V_S = +5\text{Vdc}$ and $I_{\text{LOAD}} = 50 \mu\text{A}$, in the circuit of Figure 2. These specifications also apply from $+2^\circ\text{C}$ to T_{MAX} in the circuit of Figure 1. Specifications in **boldface** apply over the full rated temperature range.

Note 2: Thermal resistance of the TO-46 package is $400^\circ\text{C}/\text{W}$, junction to ambient, and $24^\circ\text{C}/\text{W}$ junction to case. Thermal resistance of the TO-92 package is $180^\circ\text{C}/\text{W}$ junction to ambient. Thermal resistance of the small outline molded package is $220^\circ\text{C}/\text{W}$ junction to ambient. Thermal resistance of the TO-220 package is $90^\circ\text{C}/\text{W}$ junction to ambient. For additional thermal resistance information see table in the Applications section.

Note 3: Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

Note 4: Tested Limits are guaranteed and 100% tested in production.

Note 5: Design Limits are guaranteed (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.

Note 6: Specifications in **boldface** apply over the full rated temperature range.

Note 7: Accuracy is defined as the error between the output voltage and $10\text{mV}/^\circ\text{C}$ times the device's case temperature, at specified conditions of voltage, current, and temperature (expressed in $^\circ\text{C}$).

Note 8: Nonlinearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the device's rated temperature range.

Note 9: Quiescent current is defined in the circuit of Figure 1.

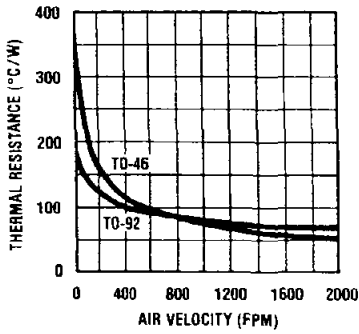
Note 10: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions. See Note 1.

Note 11: Human body model, 100 pF discharged through a $15 \text{ k}\Omega$ resistor.

Note 12: See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" or the section titled "Surface Mount" found in a current National Semiconductor Linear Data Book for other methods of soldering surface mount devices.

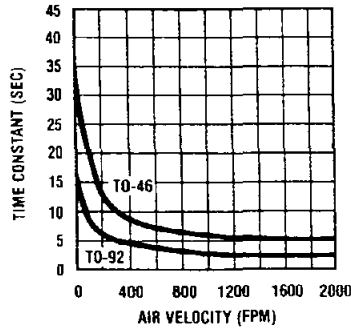
Typical Performance Characteristics

Thermal Resistance Junction to Air



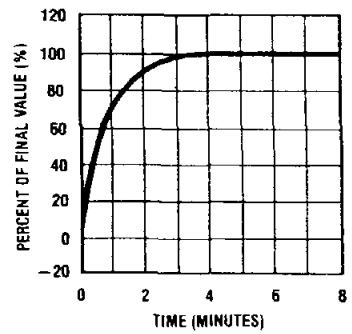
DS005516-25

Thermal Time Constant



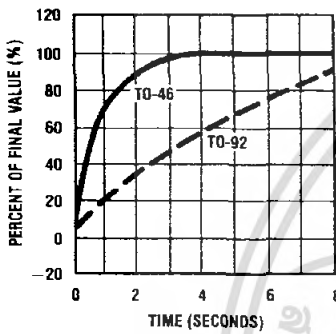
DS005516-26

Thermal Response in Still Air



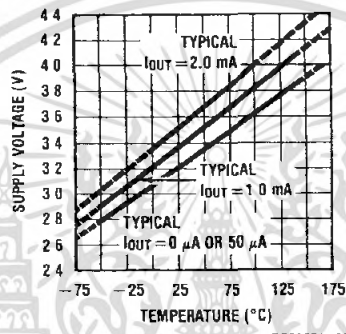
DS005516-27

Thermal Response in Stirred Oil Bath



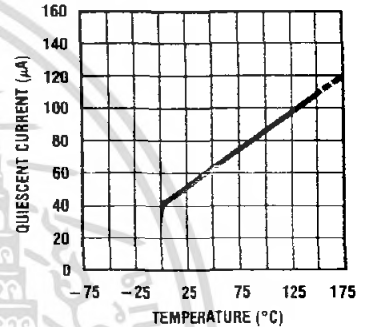
DS005516-28

Minimum Supply Voltage vs. Temperature



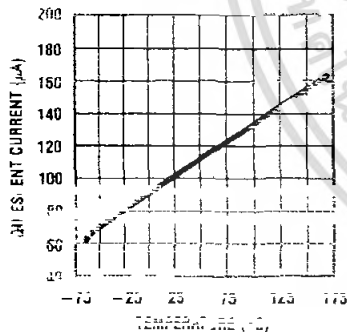
DS005516-29

Quiescent Current vs. Temperature (In Circuit of Figure 1.)



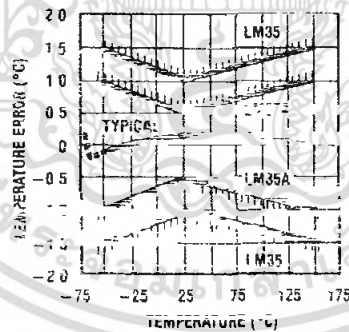
DS005516-30

Quiescent Current vs. Temperature (In Circuit of Figure 2.)



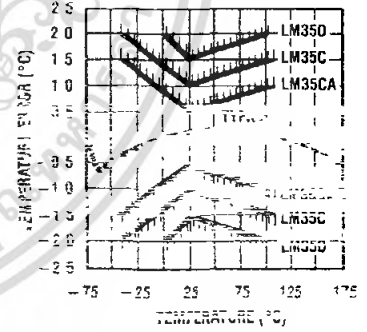
DS005516-31

Accuracy vs. Temperature (Guaranteed)



DS005516-32

Accuracy vs. Temperature (Guaranteed)



DS005516-33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



HIH-4000-001

HIH-4000 Series Integrated Circuitry Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP

Features

- Molded thermoset plastic housing with cover
- Linear voltage output vs %RH
- Laser trimmed interchangeability
- Low power design
- High accuracy
- Fast response time
- Stable, low drift performance
- Chemically resistant

Representative photograph, actual product appearance may vary.

Due to regional agency approval requirements, some products may not be available in your area. Please contact your regional Honeywell office regarding your product of choice.

Typical Applications

- Refrigeration
- Drying
- Meteorology
- Battery-powered systems
- OEM assemblies

Description

The HIH-4000 Series Humidity Sensors are designed specifically for high volume OEM (Original Equipment Manufacturer) users. Direct input to a controller or other device is made possible by this sensor's linear voltage output. With a typical current draw of only 200 μ A, the HIH-4000 Series is ideally suited for low drain, battery operated systems. Tight sensor interchangeability reduces or eliminates OEM production calibration costs. Individual sensor calibration data is available.

The HIH-4000 Series delivers instrumentation-quality RH (Relative Humidity) sensing performance in a low cost, solderable SIP (Single In-line Package). Available in two lead spacing configurations, the RH sensor is a laser trimmed, thermoset polymer capacitive sensing element with on-chip integrated signal conditioning. The sensing element's multilayer construction provides excellent resistance to most application hazards such as wetting, dust, dirt, oils and common environmental chemicals.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Honeywell

HIH-4000-001

HIH-4000 Series Integrated Circuitry Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP

Product Specifications	
Package Style	Solderable SIP
Termination Details	2,54 mm [0.100 in] Lead Pitch
Series Name	HIH-4000 Series
RH Accuracy	± 3.5% RH, 0-100 % RH non-condensing, 25 °C, 5 Vdc supply
RH Interchangeability	± 5% RH, 0-60% RH; ± 8% @ 60-100% RH Typical
RH Hysteresis	± 3% of RH Span Maximum
RH Repeatability	± 0.5% RH
RH response time, 1/e	15 s in slowly moving air @ 25 °C
RH Stability	± 0.2% RH Typical at 50% RH in 1 Year
Supply Voltage	4.0 Vdc to 5.8 Vdc
Supply Current	500 µA Max.
Operating Humidity Range	0 to 100% RH, non-condensing
Operating Temperature Range	-40 °C to 85 °C (-40 °F to 185 °F)
Temperature Compensation	True RH = Sensor RH / (1.0305 + 0.000044T - 0.0000011T ²) T in °C (True RH = Sensor RH / (0.9237 - 0.0041T + 0.000040T ²) T in °C)
Availability	Global
Comment	Light sensitive, shield from bright light.
UNSPSC Code	411121
UNSPSC Commodity	411121 Transducers

Honeywell

HIH-4000-001

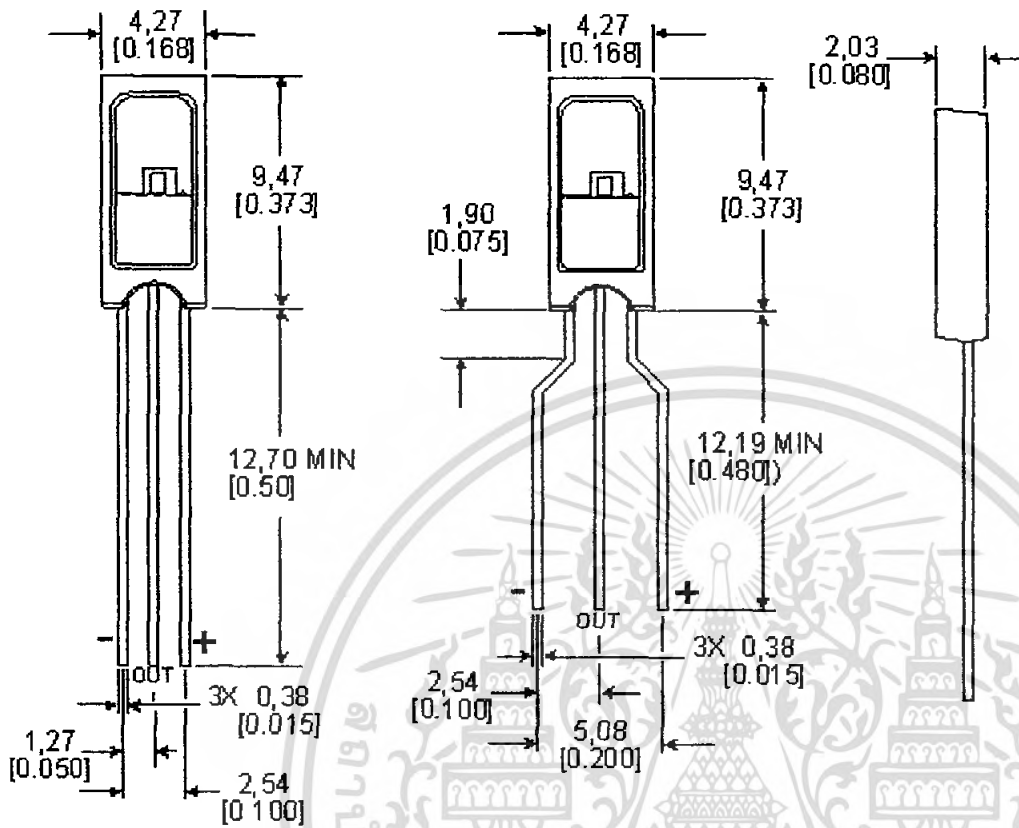
HIH-4000 Series Integrated Circuitry Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mounting Dimensions
For Reference Only [mm/in]

HIH-4000-002
HIH-4000-004

HIH-4000-001
HIH-4000-003



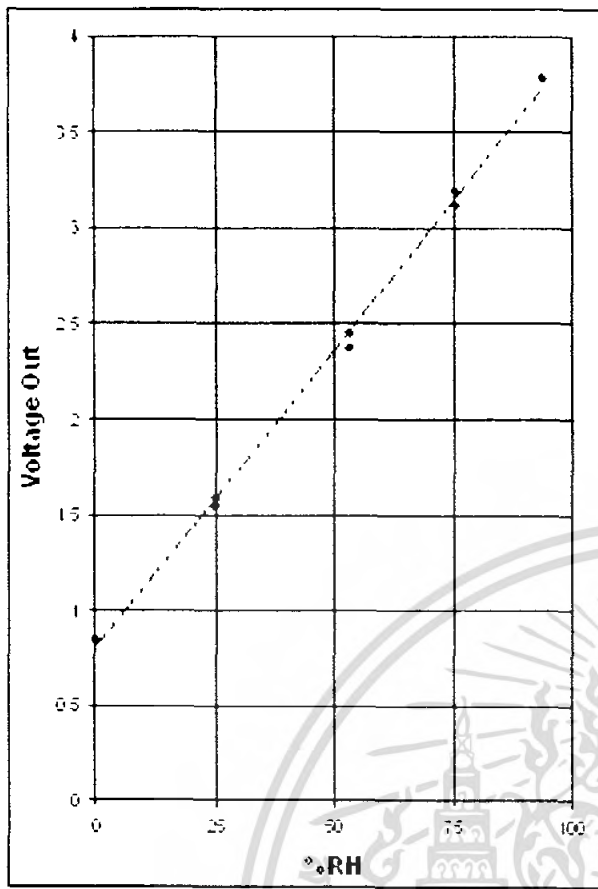
Honeywell

HIH-4000-001

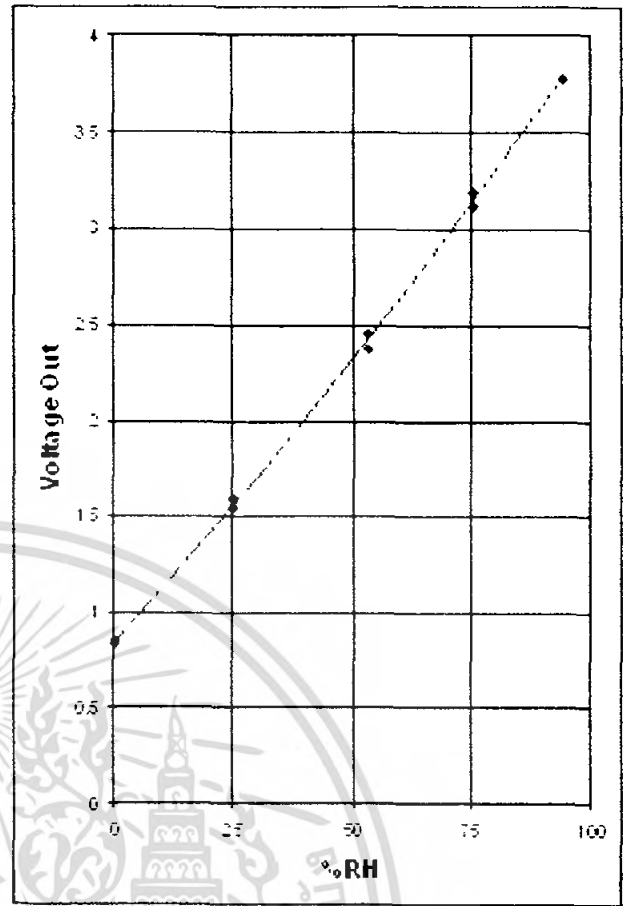
HIH-4000 Series Integrated Circuitry Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL BEST FIT STRAIGHT LINE



TYPICAL 2nd ORDER CURVE FIT



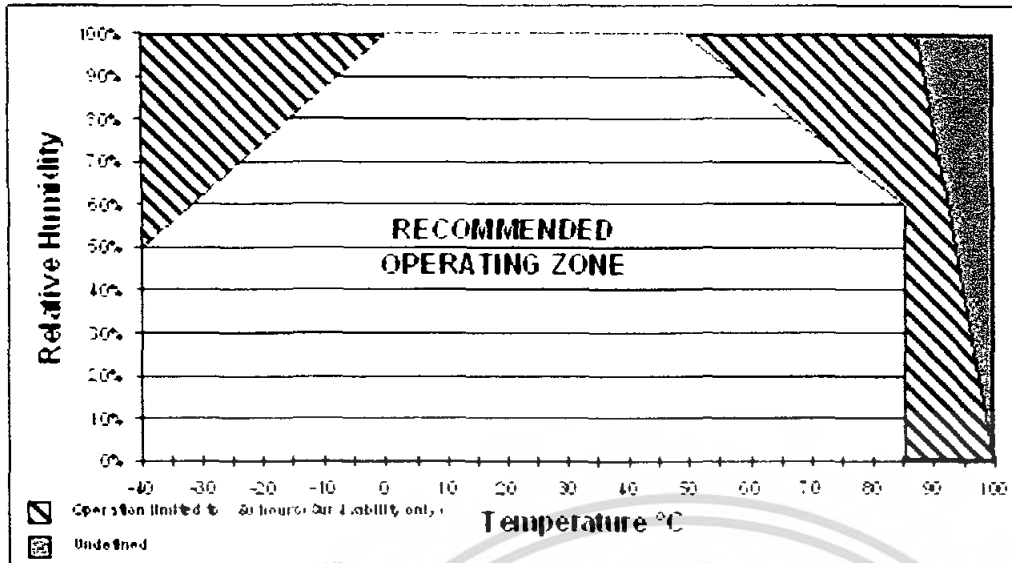
Honeywell

HIH-4000-001

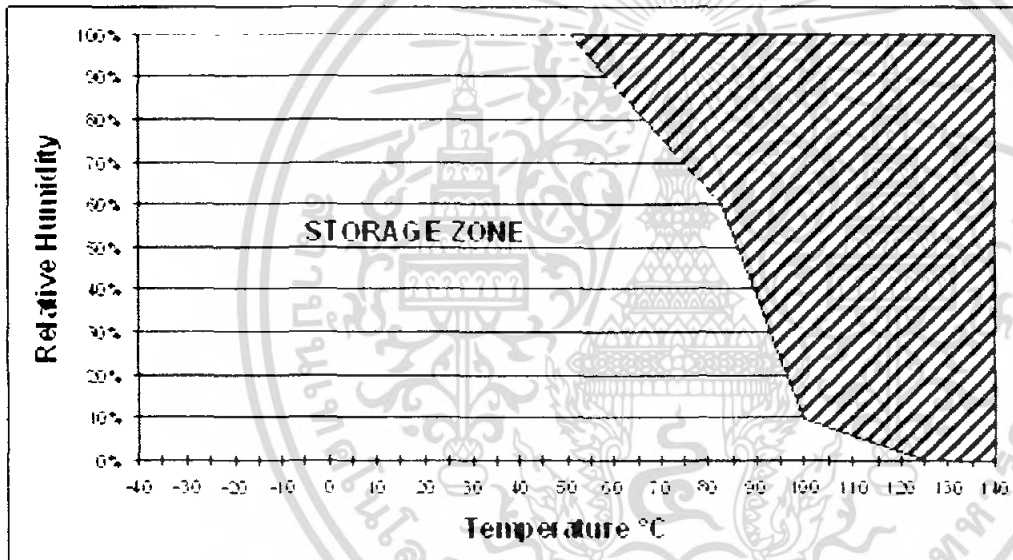
HIH-4000 Series Integrated Circuitry Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Recommended Operating Conditions



Storage Environment



Honeywell

HIH-4000-001

HIH-4000 Series Integrated Circuitry Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

⚠ WARNING

PERSONAL INJURY

DO NOT USE these products as safety or emergency stop devices, or in any other application where failure of the product could result in personal injury.

Failure to comply with these instructions could result in death or serious injury.

⚠ WARNING

MISUSE OF DOCUMENTATION

- The information presented in this product sheet (or catalog) is for reference only. DO NOT USE this document as product installation information.
- Complete installation, operation and maintenance information is provided in the instructions supplied with each product.

Failure to comply with these instructions could result in death or serious injury.

© Copyright Honeywell Inc. 1998-2004 All rights reserved.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้