ปีการศึกษา 2532

ระบบควบคุมเครื่องฟักไข่อิเลคทรอนิคส์แบบอัตโนมัติ

โดย

นาย นิรพันธ์ ลิ่มวานิชรัตน์ 28.1121

นาย พลาศิลป์ วิชิวานิเวศน์ 28.1153

อาจารย์ที่ปรึกษา

ดร. รัตติกร วรากูลศิริพันธุ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

023115 -8 ส.ค. 2532

ปริญญานิพนธ์ปีการศึกษา 2532

ภาควิชา อิเลคทรอนิคส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมเครื่องฟักไข่อิเลคทรอนิคส์แบบอัตโนมัติ

ผู้จัดทำ

1. นาย นิรพันธ์ ลิ่มวานิชรัตน์ 28.1121
2. นาย พลาศิลป์ วิชิวานิเวศน์ 28.1153

..... อาจารย์ที่ปรึกษา

(ดร. รัตติกร วราภูลศิริพันธุ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

023115

ระบบควบคุมเครื่องฟักไข่อิเลคทรอนิคส์แบบอัตโนมัติ

นาย นิรพันธ์ ลิ่มวานิชรัตน์
นาย พลาศิลป์ วิชิวานิเวศน์
ดร.รัตติกร วรากูลศิริพันธุ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2532

บทคัดย่อ

จุดประสงค์หลักของโครงงานนี้ คือ การออกแบบระบบควบคุมสำหรับตู้ฟักไข่และตู้เกิดลูกไก่ขนาดอุตสาหกรรมโดยสามารถควบคุมได้หลาย ๆ ตู้พร้อมกัน ประกอบด้วย เครื่องคอมพิวเตอร์ อุปกรณ์ที่ออกแบบสำหรับควบคุมตู้ฟักไข่และตู้เกิดลูกไก่ และโปรแกรมควบคุมการทำงาน หน้าที่ของระบบควบคุม คือ ควบคุมอุณหภูมิและความชื้นให้คงที่ การพลิกไข่ และการให้สัญญาณเตือน ใช้ พีที-100 ซึ่งมีค่าความต้านทานแปรตามอุณหภูมิ เป็นตัววัดอุณหภูมิ และความชื้น ใช้คอนเวอเตอร์แบบซัคเซสซีฟแอปพลอคซิเมชั่น ขนาด 8 บิท ในการเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล ใช้อินเทอรัพท์เพื่อให้โปรแกรมควบคุมการทำงานตามระยะเวลาที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTOMATIC CONTROLLER FOR ELECTRONICS EGG BROODER MACHINE

Niraphan Limwanitrat

Palasilp Vichivanives

Dr. Ruttikorn Varakulsiripunth Adviser

1989

Abstract

The purpose of this project is to design the controller for Industrail Incubators and Hatchers. The controller is composed of a personal computer , equipments designed for controlling Incubators and Hatchers , and a program. Functions of the controller are stabilizing temperature and humidity , turning eggs every hour and alarm. The sensor is PT-100 which its resistance varies with temperature to measure temperature and humidity. The 8 bit successive approximation analog-to-digital converter is used in converting analog signal to digital signal. The program uses periodic interrupt to perform functions.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ 1	บทนำ	1
บทที่ 2	ทฤษฎี	3
	2.1 หลักการฟักไข่	3
	2.2 วงจรขยายแตกต่าง	3
	2.3 วงจรเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล	4
	2.4 การอินเทอรัพท์ของเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที	7
	2.5 อินพุท-เอาท์พุทพอร์ทของเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที	13
	2.6 รายละเอียดสัญญาณสลอตบนซิสเต็มบอร์ดของเครื่องคอมพิวเตอร์ไอบีเอ็มพีซี	19
	2.7 การหาความชื้น	21
บทที่ 3	การออกแบบและการสร้าง	25
	3.1 หน้าที่ของระบบควบคุม	25
	3.2 อุปกรณ์ที่ถูกควบคุม	25
	3.3 อุปกรณ์ที่ต้องตรวจจับสถานะการทำงาน	26
	3.4 การแสดงผล	27
	3.5 การใช้แอดเดรสพอร์ทในฮาร์ดแวร์เพิ่มเติม	27
	3.6 โครงสร้างฮาร์ดแวร์เพิ่มเติม	29
	3.7 การขยายบัสมายังภายนอก	31
	3.8 การออกแบบการ์ดเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล ใช้กับการ์ดขยายบัสของระบบภายนอก	33
	3.9 การออกแบบการ์ดอินพุท-เอาท์พุทพอร์ท ใช้กับการ์ดขยายบัสของระบบภายนอก	37
	3.10 การออกแบบวงจรวัดอุณหภูมิและความชื้น	39
	3.11 การออกแบบโปรแกรมควบคุมการทำงาน	42
บทที่ 4	การใช้งานและผลการใช้งาน	53
	4.1 การต่อเครื่องคอมพิวเตอร์กับฮาร์ดแวร์สร้างเพิ่มเติมและการใช้งาน	53
	4.2 อุปกรณ์อื่น ๆ ที่ใช้ในการทำงาน	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การตั้งค่าอุณหภูมิและความชื้น 58
4.4 การใช้งาน 59
บทที่ 5 บทสรุปและวิจารณ์ 64
5.1 เกี่ยวกับระบบควบคุม 64
5.2 เกี่ยวกับจำนวนลูกไก่ที่ฟักได้ 64
5.3 การพัฒนา 64
ภาคผนวกที่ 1 ต้นฉบับของโปรแกรมควบคุมการทำงาน 66
กิตติกรรมประกาศ 67
บรรณานุกรม 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูป 2.1 วงจรขยายแตกต่าง 4

รูป 2.2 การวัดโดยใช้เทคนิคซัคเซสซีฟแอปพลอคซิเมชัน 5

รูป 2.3 บลอคไดอะแกรมของวงจรเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอลด้วยวิธีซัคเซสซีฟแอปพลอคซิเมชัน 6

รูป 2.4 บลอคไดอะแกรมของวงจรอินเทอรัพท์ในไอบีเอ็มพีซี 7

รูป 2.5 บลอคไดอะแกรมและการจัดเรียงขาของ 8259A 8

รูป 2.6 การใช้งานหน่วยความจำ 1024 ไบท์ล่างของไอบีเอ็มพีซีสำหรับการขออินเทอรัพท์ทั้ง 256 แบบ 12

รูป 2.7 การใช้ตารางเวคเตอร์ของ 8259A ในไอบีเอ็มพีซี 13

รูป 2.8 สัญญาณของบัสไซเคิลอ่านอินพุท-เอาท์พุทพอร์ท 14

รูป 2.9 สัญญาณของบัสไซเคิลเขียนอินพุท-เอาท์พุทพอร์ท 15

รูป 2.10 การใช้แอดเดรสของอินพุท-เอาท์พุทของไมโครโปรเซสเซอร์ 8088 16

รูป 2.11 การใช้งานที่ว่างของอินพุท-เอาท์พุทพอร์ทของไมโครโปรเซสเซอร์ 8088 บนเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที 16

รูป 2.12 หน้าที่ของพอร์ทบนซีสเต็มบอร์ด 17

รูป 2.13 การใช้งานส่วนที่สองของที่ว่างแอดเดรส 17

รูป 2.14 ส่วนประกอบและแนวคิดการขยายบัส 19

รูป 2.15 รายละเอียดสัญญาณบนสลอตของซิสเต็มบอร์ด 22

รูป 2.16 แกนต่าง ๆ ที่แสดงคุณสมบัติของความชื้นในอากาศบนไซโครเมตริกชาร์ท 23

รูป 2.17 ไซโครเมตริกชาร์ท 23

รูป 3.1 พัดลม หัวพ่นไอน้ำ ขดลวดความร้อนและท่อลมภายในตู้ฟักไข่ 26

รูป 3.2 โครงสร้างฮาร์ดแวร์เพิ่มเติม 30

รูป 3.3 วงจรการ์ดขยายระบบบัสภายในเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที 31

รูป 3.4 วงจรการ์ดขยายบัสของระบบภายนอกเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที 32

รูป 3.5 สัญญาณบนสลอต 30 ขาบนการ์ดขยายบัสของระบบภายนอกเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 3.6 โครงสร้างการ์ดเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล 35
รูป 3.7 วงจรของบลอคไดอะแกรม 1 และ 2 ในรูป 3.6 36
รูป 3.8 วงจรของบลอคไดอะแกรม 3 และ 4 ในรูป 3.6 37
รูป 3.9 โครงสร้างการ์ดอินพุท-เอาท์พุทพอร์ท 38
รูป 3.10 วงจรของบลอคไดอะแกรม 1 และ 2 ในรูป 3.9 39
รูป 3.11 วงจรของบลอคไดอะแกรม 3 และ 4 ในรูป 3.9 40
รูป 3.12 วงจรวัดอุณหภูมิและความชื้น 41
รูป 3.13 โฟลชาร์ทขั้นตอนการทำงานหลักของโปรแกรมตอบสนองการอินเทอรัพท์ 1C 43
รูป 3.13 (ต่อ) โฟลชาร์ทของ Temperature Process ในการทำงานหลัก 44
รูป 3.13 (ต่อ) โฟลชาร์ทส่วนแรกของ Humidity Process ในการทำงานหลัก 45
รูป 3.13 (ต่อ) โฟลชาร์ทส่วนหลังของ Humidity Process ในการทำงานหลัก 46
รูป 3.13 (ต่อ) โฟลชาร์ทของ Error Process ในการทำงานหลัก 47
รูป 3.14 โฟลชาร์ทส่วนแรกของโปรแกรมตอบสนองอินเทอรัพท์หมายเลข 0A 49
รูป 3.14 (ต่อ) โฟลชาร์ทส่วนหลังของโปรแกรมตอบสนองอินเทอรัพท์หมายเลข 0A 50
รูป 3.15 โฟลชาร์ทการทำงานของโปรแกรมหลัก 52
รูป 4.1 วงจรวัดอุณหภูมิและความชื้น (อยู่บริเวณตรงกลาง) เมื่อลง
บริ้นท์เซอร์กิทบอร์ด (Print Circuit Board) 53
รูป 4.2 วงจรสวิทช์ไฟกำลังสูง (อยู่บริเวณกลางรูป) เมื่อลงบริ้นท์เซอร์กิตบอร์ด 54
รูป 4.3 เครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ทีกับฮาร์ดแวร์ที่ออกแบบสร้างเพิ่มเติม
เพื่อใช้ในการฟักไข่ 54
รูป 4.4 กล่องบรรจุวงจรสวิทช์ไฟกำลังสูง (กล่องกลาง) พร้อมทั้ง
แอลอีดีแสดงการทำงานและแจ้งเตือน 55
รูป 4.5 จอภาพขณะกำลังทำงาน 56
รูป 4.6 รถใส่ไข่ที่จะเข้าตู้ฟักไข่ 56
รูป 4.7 รถใส่ไข่ที่จะเข้าตู้เกิดลูกไก่ 57
รูป 4.8 ภายในตู้เกิดลูกไก่ในขณะที่ยังไม่ใส่รถ 58
รูป 4.9 ภาพบนจอภาพที่ได้จากการสั่งพิมพ์จอภาพในขณะที่ทำการตั้งค่า 59
รูป 4.10 ภาพบนจอภาพ ขณะใช้งานควบคุมตู้ฟักไข่ เมื่อเวลา 11.59 นาฬิกา 60

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 4.11 ภาพบนจอภาพ ขณะใช้งานควบคุมตู้ฟักไข่ เมื่อเวลา 13.27 60
รูป 4.12 ภาพบนจอภาพ ขณะใช้งานควบคุมตู้ฟักไข่ เมื่อเวลา 20.21 นาฬิกา 61
รูป 4.13 ไข่ในตู้เกิดลูกไก่ที่ถ่ายมาจากตู้ฟักไข่ 62
รูป 4.14 ตัวลูกไก่ที่เกิดในรถเกิดลูกไก่คันหนึ่ง ในตู้เกิดลูกไก่ 62
รูป 4.15 ลูกไก่ที่เกิดมาในถาดหนึ่งของรถในรูป 4.14 63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง 2.1	แสดงถึงการใช้งานอินเทอร์พท์ในระดับต่าง ๆ ของไอบีเอ็มพีซี	7
ตาราง 2.2	ตัวอย่างความชื้นสัมพัทธ์หาได้จากไซโคเมตริกชาร์ท	24
ตาราง 3.1	การใช้แอดเดรสอินพุท-เอาท์พุทพอร์ทของการ์ดเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล	28
ตาราง 3.2	การใช้แอดเดรสอินพุท-เอาท์พุทพอร์ท	29
ตาราง 3.3	ความต้านทานของพีที-100 ที่อุณหภูมิต่าง ๆ	40
ตาราง 3.4	การทำงานในแต่ละครั้งของอินเทอร์พท์หมายเลข 1C (ฐาน 16)	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันเนื่องจากจำนวนประชากรเพิ่มขึ้นทำให้ต้องบริโภคอาหารเพิ่มขึ้น อาหารสัตว์ เช่น เนื้อไก่ก็มีความต้องการสูงขึ้นเช่นเดียวกัน ดังนั้นจึงเกิดความต้องการให้การผลิตไก่มีประสิทธิภาพสูงขึ้น โดยการใช้เครื่องฟักไข่ทำให้เปอร์เซนต์การเกิดเป็นไก่สูงกว่าการปล่อยให้เกิดเองตามธรรมชาติ และทำให้ต้นทุนการผลิตต่ำลงด้วย

เครื่องฟักไข่ที่มีประสิทธิภาพสูงต้องสั่งซื้อมาจากต่างประเทศด้วยราคาแพง ซึ่งถ้าสามารถสร้างเครื่องฟักไข่ที่มีประสิทธิภาพสูงนี้ขึ้นมาเองด้วยราคาที่ถูกกว่า ทำให้ไม่ต้องสั่งซื้อเครื่องฟักไข่จากต่างประเทศ และนำไปสู่การพึ่งพาตนเองทางเทคโนโลยี

ปริญญานิพนธ์ฉบับนี้กล่าวถึงรายละเอียดเครื่องฟักไข่ และระบบควบคุมเครื่องฟักไข่ที่ได้ออกแบบและทำการสร้างขึ้นมาเอง ซึ่งสามารถแบ่งเนื้อหาออกเป็นบทดังนี้

บทที่ 1 บทนำ กล่าวถึงวัตถุประสงค์ของปริญญานิพนธ์ และแนะนำเนื้อหาที่มีอยู่ในปริญญานิพนธ์ เพื่อให้ความสะดวกในการเลือกหารายละเอียดที่ต้องการได้ง่าย

บทที่ 2 ทฤษฎี กล่าวถึงหลักการฟักไข่ หลักการของวงจรขยายแบบแตกต่าง (Differential Amplifier) หลักการของวงจรเปลี่ยนสัญญาณไฟฟ้าอนาลอกเป็นดิจิตอล (Analog To Digital Converter) การอินเทอรัพท์ของเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที (Interrupt On IBM PC/XT Personal Computer) อินพุท-เอาท์พุทพอร์ทของเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที (Input/Output Port On IBM PC/XT Personal Computer) รายละเอียดสัญญาณสลอตบนซีสเต็มบอร์ดของเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที และการหาความชื้นสัมพัทธ์

บทที่ 3 การออกแบบและสร้าง กล่าวถึงหน้าที่ของระบบควบคุม อุปกรณ์ที่ถูกควบคุม อุปกรณ์ที่ต้องตรวจจับสถานะการทำงาน การแสดงผล การใช้แอดเดรสพอร์ทในฮาร์ดแวร์เพิ่มเติม โครงสร้างฮาร์ดแวร์เพิ่มเติม การขยายบัสมายังภายนอก การออกแบบการ์ดเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอลใช้กับการ์ดขยายบัสของระบบภายนอก การออกแบบการ์ดอินพุท-เอาท์พุทพอร์ทใช้กับการ์ดขยายบัสของระบบภายนอก การออกแบบวงจรวัดอุณหภูมิและความชื้น และการออกแบบโปรแกรมควบคุมการทำงาน

บทที่ 4 การใช้งานและผลการใช้งาน กล่าวถึง การต่อเครื่องคอมพิวเตอร์กับฮาร์ด

เอกสารนี้มีลิขสิทธิ์ สามารถใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แวร์สร้างเพิ่มเติมและการใช้งาน อุปกรณ์อื่น ๆ ที่ใช้ในการทำงาน การตั้งค่าอุณหภูมิและความชื้น การใช้งานและผลการใช้งาน

บทที่ 5 บทวิจารณ์และสรุป กล่าวถึงผลเกี่ยวกับระบบควบคุม เกี่ยวกับจำนวนลูกไก่ที่ฟักได้และขั้นตอนที่ควรจะพัฒนาต่อไป

ภาคผนวก เป็นซอร์สโปรแกรม (Source Program) ของโปรแกรมควบคุมการทำงานเครื่องฟักไข่ทั้งหมด

บรรณานุกรมแสดงรายชื่อหนังสืออ้างอิงสำหรับผู้ที่สนใจในรายละเอียดของหัวข้อต่าง ๆ และต้องการศึกษาโดยละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎี

2.1 หลักการฟักไข่

2.1.1 การเจริญเติบโตของเชื้อลูกไก่

การฟักไข่เป็นการทำให้เชื้อลูกไก่เจริญเติบโต การเจริญเติบโตจะเริ่มจากแม่ไก่ที่ได้รับการผสมพันธุ์จากตัวผู้ ออกไข่และนำไปฟัก ถ้าไข่นั้นมีความสมบูรณ์พันธุ์เชื้อลูกไก่ก็จะเจริญเติบโตทีละน้อย การเก็บรวบรวมไข่เพื่อนำไปฟักจะทำภายหลังผสมพันธุ์แบบฉีดเชื้อราว 20 ชั่วโมงก็อาจจะได้ไข่ที่มีเชื้อบริบูรณ์ แต่ถ้าจะให้ได้ผลดีที่สุดควรรออย่างน้อย 3 วัน

โดยปกติเชื้อลูกไก่จะทวีเซลล์เจริญขึ้นตั้งแต่อยู่ในตัวแม่ไก่ ขณะที่ไข่ออกจากตัวแม่ไก่เชื้อลูกไก่นี้จะมีเป็นจำนวนร้อยเรียงรายเป็นเซลล์ชั้นนอกและชั้นใน และอาจเริ่มมีเซลล์ชั้นกลางบ้างแล้ว ตราบที่ไข่ได้รับอุณหภูมิสูงกว่า 68 องศาฟาเรนต์ไฮต์เซลล์ก็จะทวีขึ้นเรื่อย ๆ หากได้รับอุณหภูมิหนาว ๆ ร้อน ๆ ไม่คงที่ของกลางวันกลางคืนไปเรื่อย ๆ การทวีเซลล์ก็จะมีอุปสรรค โอกาสที่เชื้อลูกไก่จะตายก็มีสูงขึ้น ซึ่งถ้ามีการรักษาอุณหภูมิและรักษาความสะอาดตลอดจนทำการฟักให้ถูกต้องเชื้อลูกไก่ก็จะเจริญต่อไปจนเป็นตัวลูกไก่ที่สมบูรณ์

2.1.2 สภาพที่เหมาะสมสำหรับการฟักไข่

เพื่อให้การฟักไข่มีประสิทธิภาพสูง สภาพแวดล้อมควรเป็นดังนี้

ในช่วง 18 วันแรกของการฟักไข่ ควรให้ไข่ได้รับอุณหภูมิประมาณ 99.5-100 องศาฟาเรนไฮต์ และความชื้นสัมพัทธ์ประมาณ 75 % ในช่วง 3 วันหลัง ควรให้ไข่ได้รับอุณหภูมิประมาณ 98-99 องศาฟาเรนไฮต์และความชื้นประมาณ 85 %

ควรมีการรักษาความสะอาด ฆ่าเชื้อโรคในบริเวณที่ทำการฟักไข่ และจัดให้อากาศถ่ายเทหมุนเวียนได้ดี

2.2 วงจรขยายแตกต่าง

ขาอินพุททั้งสองของโอเปอเรชันแนลแอมปริฟลายเอร์ (Operational Amplifier) คือ อินพุทบวกและอินพุทลบ ทำให้เกิดสัญญาณเอาท์พุทเท่ากันแต่กลับกัน โดยปกติออปแอมป์จะทำหน้าที่ขยายสัญญาณอินพุท 3 แบบด้วยกัน คือ สัญญาณอินพุทข้างเดียว (Single Ended Input Voltages) สัญญาณอินพุทแตกต่าง (Differential Input Voltages) และสัญญาณอินพุทร่วม (Common mode input voltages) สัญญาณอินพุทข้างเดียวเป็นสัญญาณระหว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราวนด์ (Ground) และขาอินพุทข้างใดข้างหนึ่ง สัญญาณอินพุทร่วมเป็นสัญญาณระหว่างขาอินพุททั้งสอง สัญญาณร่วมเป็นสัญญาณที่ปรากฏที่อินพุททั้งสองและมีค่าเท่ากัน ออปแอมป์ถูกออกแบบมาให้ขยายสัญญาณแตกต่าง และกำจัดสัญญาณร่วม ความสามารถในการกำจัดสัญญาณร่วมของออปแอมป์เรียกว่า คอมมอนโหมดรีเจคชันเรโช (Common Mode Rejection Ratio)

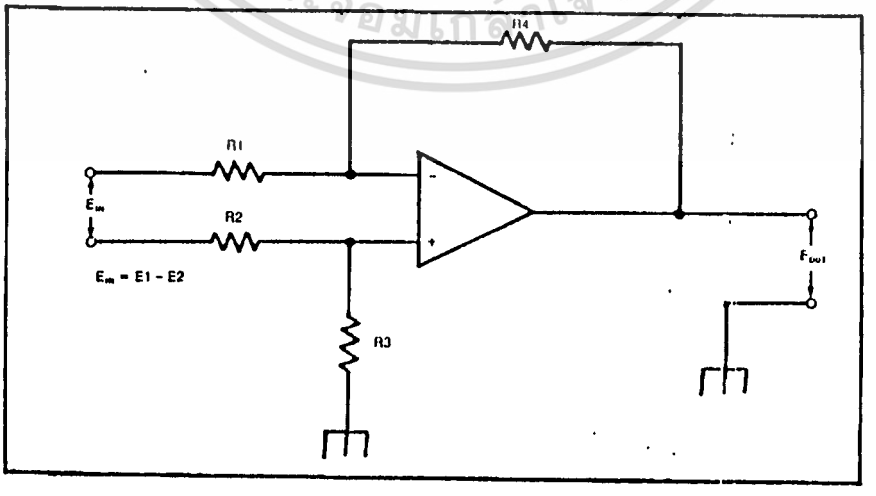
การที่ออปแอมป์สามารถกำจัดสัญญาณร่วมได้ ทำให้ออปแอมป์เหมาะสมสำหรับการสร้างเครื่องมือ เพราะว่าสามารถกำจัดสัญญาณรบกวนเมื่อมีสัญญาณรบกวนรุนแรงจากภายนอก ลองพิจารณากรณีวงจรขยายจำเป็นต้องต่อกับสัญญาณผ่านทางสาย ก็จะต้องชิลด์ (Shield) สายเมื่อต่อวงจรข้างเดียว ถึงอย่างนั้นก็ตามก็ยังอาจจะมีสัญญาณรบกวนจากแหล่งจ่ายไฟ (Power Supply) ได้ ในวงจรขยายแตกต่างสัญญาณรบกวนจะมีผลกับทั้งสองอินพุทเท่า ๆ กัน สัญญาณจะหักล้างกันหมดเพราะว่าทำให้เกิดเอาท์พุทเท่ากันแต่กลับขั้วกัน

รูป 2.1 แสดงวงจรขยายแตกต่างซึ่งใช้ออปแอมป์ 1 ตัว ความต้านทาน (Resister) R1 และ R2 เป็นความต้านทานอินพุท แต่ละตัวต่อไปยังขาอินพุทของออปแอมป์แต่ละข้าง โดยปกติในการใช้งานจะกำหนดให้ค่าความต้านทาน R1 เท่ากับ R2 และ R3 เท่ากับ R4 จะได้อัตราการขยายแรงดันของวงจรเป็น

$$Av = R4 / R1 = R3 / R2$$

ให้ R1 = R2 และ R3 = R4

ความต้านทานอินพุทบวกจะมีค่าเท่ากับความต้านทาน R2 และ ความต้านทาน R3 ส่วนความต้านทานอินพุทลบมีค่าเท่ากับความต้านทาน R1



รูป 2.1 วงจรขยายแตกต่าง

2.3 วงจรเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล

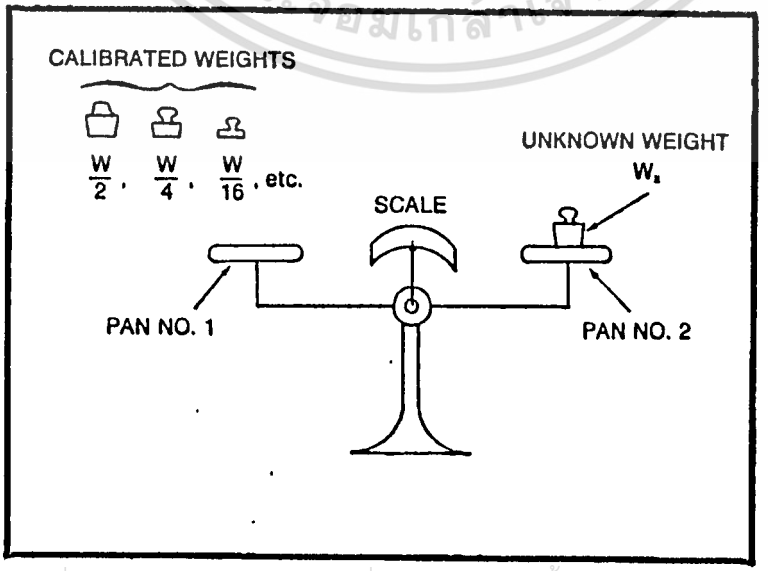
เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำหน้าที่เป็นสัญญาณอนาลอก (กระแส หรือ แรงดัน) เป็นสัญญาณดิจิตอลที่มีค่าเหมือนกัน วงจรจะให้เอาท์พุทเปลี่ยนไปในช่วง 1 บิทสำคัญน้อยที่สุด (Least Significant Bit) เมื่อสัญญาณอนาลอกเพิ่มขึ้นหรือลดลงในช่วงหนึ่ง ปกติคอมพิวเตอร์หรืออุปกรณ์ที่เป็นดิจิตอลจะใช้วงจรนี้ เพื่อใช้ติดต่อกับอุปกรณ์ที่เป็นอนาลอก

วิธีเปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิตอลมีหลายวิธีด้วยกัน แต่ขอกล่าวถึงหลักการของวงจรซัคเซสซีฟแอปพลอคซิเมชัน (Successive Approximation) ที่ใช้ในโครงงานนี้เท่านั้น

วงจรซัคเซสซีฟแอปพลอคซิเมชันใช้เทคนิคในการเปลี่ยนคล้ายกับวิธีที่ใช้วัดจำนวนทางกายภาพ นั่นก็คือ ทดสอบและผิดพลาด (Trial And Error) วงจรจะสร้างค่าเพื่อทดสอบ ทำการทดสอบ และทำการเปลี่ยนเอาท์พุทตามผลลัพท์ที่ได้รับจากการทดสอบ วงจรจะทดสอบต่อไปจนกว่าจะครบทุกบิทหรือจบการเปลี่ยน

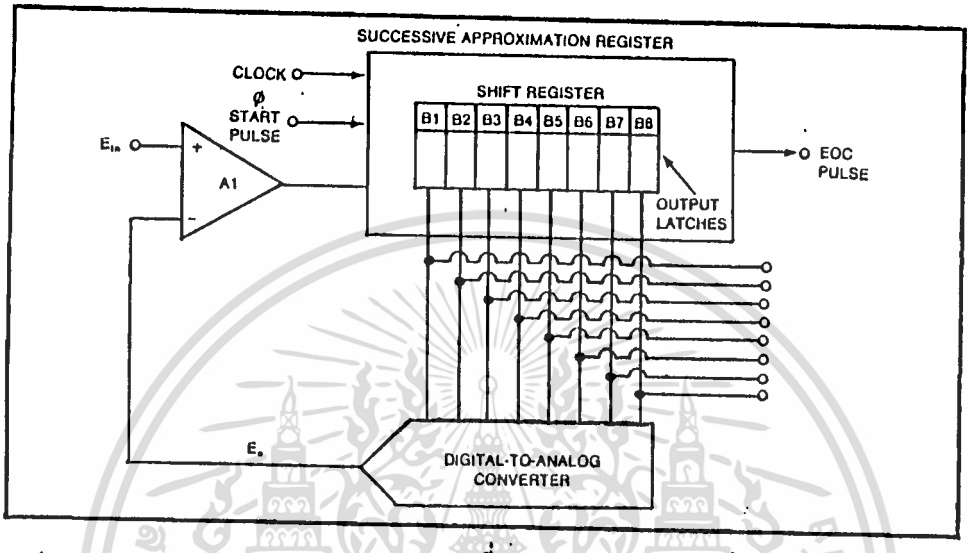
รูป 2.2 แสดงการวัดด้วยวิธีซัคเซสซีฟแอปพลอคซิเมชันในลักษณะเดียวกัน โดยมีตาชั่งและข้างหนึ่งมีน้ำหนักที่ไม่รู้ค่าวางอยู่ อีกข้างไว้สำหรับวางน้ำหนักทดสอบ เมื่อน้ำหนักทดสอบเท่ากับน้ำหนักที่ไม่รู้ค่าตาชั่งจะสมดุลพอดี เข็มชี้จะชี้ที่ตรงกลางของสเกล (Scale) เทคนิคในการวัดก็ไม่ยุ่งยาก โดยที่จะแบ่งน้ำหนักทดสอบให้มีความสัมพันธ์กันแบบกำลังสอง เช่น W/2 , W/4 , W/8 , W/16 เป็นต้น เริ่มวัดโดยวางน้ำหนักทดสอบที่หนักที่สุด (W/2) ไว้ที่แขนข้างหนึ่งของตาชั่ง และวางน้ำหนักที่ไม่รู้ค่าอีกแขนหนึ่ง ถ้าน้ำหนักที่ไม่รู้ค่าหนักกว่าน้ำหนักทดสอบเข็มชี้จะเอียงไปทางน้ำหนักที่ไม่รู้ค่า แต่ถ้าน้ำหนักทดสอบหนักกว่าเข็มก็จะชี้ไปทางน้ำหนักทดสอบ นี่ก็คือการทดสอบ ผลลัพท์ที่ได้ก็จะนำมาเพื่อใช้ตัดสินว่าควรเอาน้ำหนักทดสอบออกหรือไม่ก่อนที่จะทด



รูป 2.2 การวัดโดยใช้เทคนิคซัคเซสซีฟแอปพลอคซิเมชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สอบด้วยน้ำหนักทดสอบที่หนักน้อยกว่าถัดไป ในวงจรเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอลใช้ตัวเปรียบเทียบแรงดัน (Voltage Comparator) ในการทดสอบและผลลัพท์ที่ได้จะนำไปเซ็ท (Set) หรือรีเซ็ท (Reset) บิทที่อยู่ในรีจิสเตอร์ (Register) การทดสอบจะทำต่อไปจนกว่าตาชั่งจะสมดุล



รูป 2.3 บลอคไดอะแกรมของวงจรเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอลด้วยวิธีซัคเซสซีพแอปพลอคซิเมชัน

รูป 2.3 แสดงบลอคไดอะแกรมของวงจรเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอลด้วยวิธีซัคเซสซีพแอปพลอคซิเมชัน ส่วนประกอบที่สำคัญคือ ตัวเปรียบเทียบระดับแรงดัน ตัวเปลี่ยนสัญญาณดิจิตอลเป็นอนาลอก (Digital-to-anlog Converter) วงจรควบคุม (Control Logic) ชิฟท์รีจิสเตอร์ (Shift Register) และเอาท์พุทแลทช์ (Output Latches)

วงจรเริ่มทำงานเมื่อได้รับพัลซ์เริ่ม (Start Pulse) ทุกบิทมีค่าเป็นศูนย์และบิท B1 เป็นบิทสำคัญที่สุด (Most Significant Bit) เมื่อมีสัญญาณนาฬิกา (Clock) ลูกแรกหลังจากพัลซ์เริ่มบิท B1 จะมีค่าเป็น 1 ชั่วคราว ทำให้เอาท์พุทไปยังวงจรเปลี่ยนสัญญาณดิจิตอลเป็นอนาลอกเป็น 10000000 หรือครึ่งหนึ่งของค่าเต็ม ถ้าแรงดันอินพุท (E_{in}) มีค่ามากกว่าทำให้เอาท์พุทแลทช์บิท B1 เป็น 1 ถ้าแรงดันอินพุทมีค่าน้อยกว่า ทำให้บิท B1 ที่เอาท์พุทแลทช์เป็น 0 เพราะฉะนั้นบิทแรกจะมีค่าเป็น 1 ในกรณีแรก และ 0 ในกรณีหลัง จากนั้นก็จะเลื่อนมาเซ็ทบิทถัดมาและทำการทดสอบต่อไป แต่ละบิทจะถูกทดสอบและเอาท์พุทแลทช์จะเซ็ทหรือรีเซ็ทขึ้นอยู่กับผลลัพท์ของการทดสอบ เมื่อบิทสุดท้ายถูกทดสอบแล้ววงจรจะจบการทำงานในสัญญาณนาฬิกาลูกถัดมา ดังนั้นในการเปลี่ยนเป็นค่าดิจิตอล N บิทก็จะใช้เวลาในการเปลี่ยนเท่ากับ N + 1 สัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

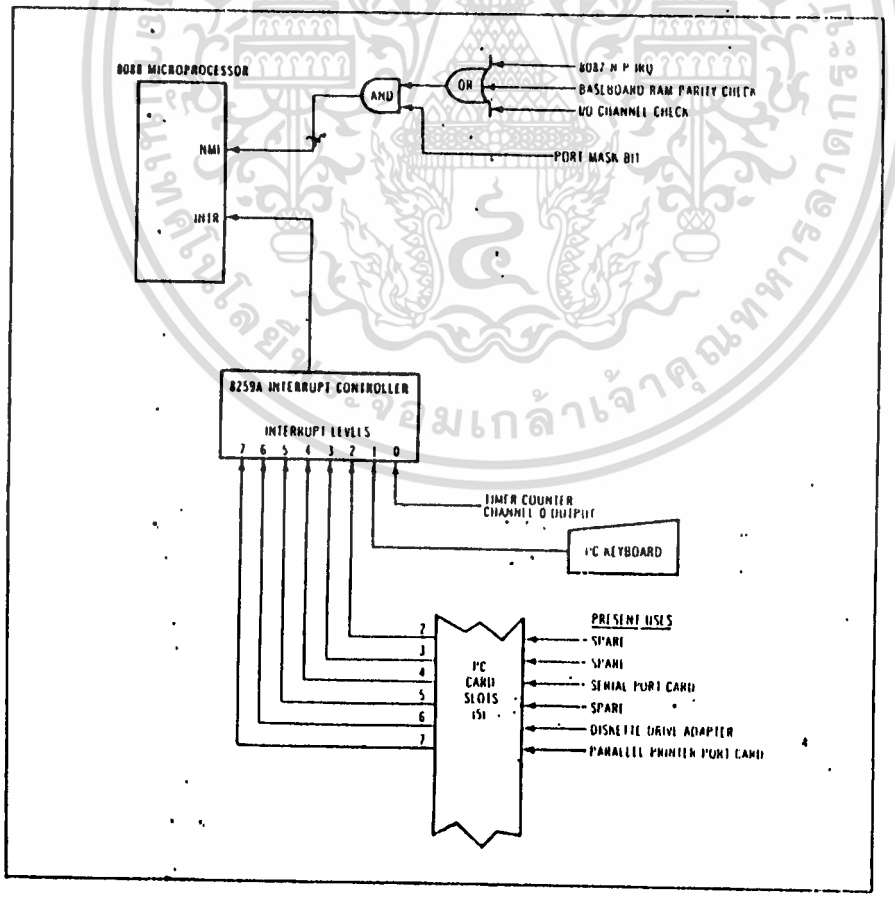
2.4 การอินเทอรัพท์ของเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที

2.4.1 การจัดอินเทอรัพท์ของไอบีเอ็มพีซี

การจัดอินเทอรัพท์ในไอบีเอ็มพีซี จะแบ่งเป็น 9 ระดับ โดยสามารถแบ่งออกได้เป็น 2 กลุ่ม คือ การอินเทอรัพท์แบบนอนมาส์คเคเบิล (Non-maskable Interrupt;NMI) และการอินเทอรัพท์แบบมาส์คเคเบิล (Maskable Interrupt;INT)

Interrupt Level.		Usage
Highest Level	NMI	Baseboard RAM parity, I/O channel check, numeric processor.
	IRQ 0	System timer output 8253-5 Channel 0.
	IRQ 1	Keyboard scan code interrupt.
Available in System Bus	IRQ 2	Not used at present.
	IRQ 3	Not used at present.
	IRQ 4	RS-232-C serial port.
	IRQ 5	Not used at present.
	IRQ 6	Diskette DRV status.
	IRQ 7	Parallel PRT port (not used in BIOS).

ตาราง 2.1 แสดงถึงการใช้งานอินเทอรัพท์ในระดับต่าง ๆ ของไอบีเอ็มพีซี



รูป 2.4 บลอคไดอะแกรมของวงจรอินเทอรัพท์ในไอบีเอ็มพีซี

สำหรับการอินเทอรัพท์แบบมาส์คเคเบิลนั้น ไอบีเอ็มพีซีจะใช้ชิพซัพพอร์ท (Chip

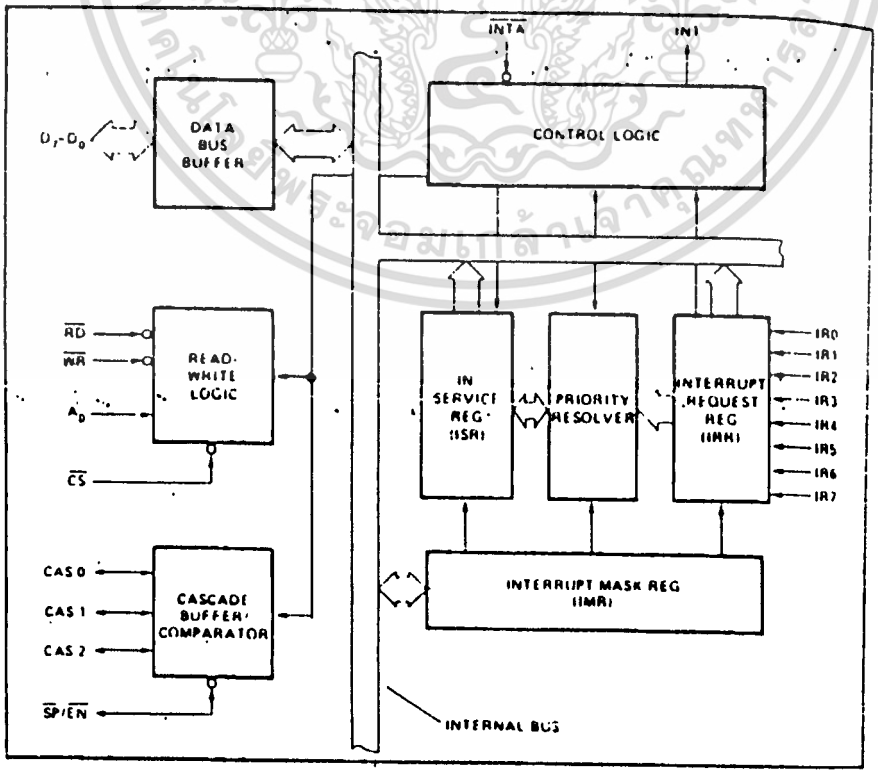
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Support) เบอร์ 8259A อินเทอรัพท์คอนโทรเลอร์ (Interrupt Controller) เข้าช่วยในการจัดอินเทอรัพท์แบบมาส์คเคเบิลของระบบ ซึ่งการใช้ 8259A เข้าช่วยนี้ ทำให้ไอบีเอ็มพีซี สามารถจัดแบ่งการขออินเทอรัพท์ในแบบมาส์คเคเบิลออกได้ถึง 8 ระดับ โดยผ่านทางแชนแนลการอินเทอรัพท์ทั้ง 8 แชนแนลของ 8259A คือ IR0-IR7

ดังนั้น จะเห็นได้ว่า ภายในไอบีเอ็มพีซี จะมีการจัดระดับของการขออินเทอรัพท์ไว้ถึง 9 ระดับ คือ ในการอินเทอรัพท์แบบนอนมาส์คเคเบิล 1 ระดับ และในการอินเทอรัพท์แบบมาส์คเคเบิลอีก 8 ระดับ แต่อย่างไรก็ตามก็ไม่สามารถที่จะเลือกใช้แชนแนลของการอินเทอรัพท์ได้หมดทั้ง 9 ระดับ ทั้งนี้ก็เพราะการขออินเทอรัพท์ในบางระดับนั้นถูกใช้งานโดยไอบีเอ็มพีซีอยู่แล้ว ดังนั้นในการที่จะออกแบบระบบอินเทอเฟสกับไอบีเอ็มพีซีได้นั้น จึงจำเป็นต้องทราบถึงการใช้งานอินเทอรัพท์ในระดับต่าง ๆ ของไอบีเอ็มพีซีด้วย สำหรับในตาราง 2.1 แสดงถึงการใช้งานอินเทอรัพท์ในระดับต่าง ๆ ของไอบีเอ็มพีซี สำหรับ รูปที่ 2.4 จะแสดงบลอคไดอะแกรมของวงจรในส่วนของการอินเทอรัพท์ในไอบีเอ็มพีซี

2.4.2 อินเทอรัพท์คอนโทรเลอร์ 8259A

8259A เป็นชิพซัพพอร์ทที่ช่วยจัดการเกี่ยวกับการขออินเทอรัพท์ของระบบ ซึ่งการใช้ 8259A นี้จะทำให้สามารถเพิ่มจำนวนระดับ (Channel) ของการขออินเทอรัพท์แบบมาส์คเค



รูป 2.5 บลอคไดอะแกรมและการจัดเรียงขาของ 8259A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เบิลได้มากถึง 8 ระดับ โดยแต่ละระดับจะมีลำดับความสำคัญที่แตกต่างกัน รูปที่ 2.5 จะแสดงถึงบลอคไดอะแกรม , การจัดเรียงขาและหน้าที่ของขาต่าง ๆ ของ 8259A

2.4.3 ขั้นตอนโดยย่อที่เกิดขึ้นในขบวนการอินเทอรัพท์

1) วงจรอินเทอร์เฟส (หรืออุปกรณ์ภายนอก) ส่งสัญญาณเพื่อขออินเทอรัพท์

2) 8259A ได้รับสัญญาณขออินเทอรัพท์ และเซ็ทบิท IRR ของแชนแนลที่ขออินเทอรัพท์นั้น

3) 8259A ส่งสัญญาณ INT ให้กับ 8088 เพื่อขออินเทอรัพท์

4) 8088 ทำการตอบรับการขออินเทอรัพท์ โดยการส่งพัลส์ลูกแรกของสัญญาณ INTA ให้กับ 8259A ซึ่งจะทำให้บิท ISR ของแชนแนลที่ขออินเทอรัพท์ (หรือแชนแนลที่มีลำดับความสำคัญสูงสุด ในกรณีที่มีการขออินเทอรัพท์พร้อมกันมากกว่า 1 แชนแนล) ถูกเซ็ท และบิท IRR ของแชนแนลนั้นก็จะถูกรีเซ็ทด้วย

5) 8088 ทำการส่งพัลส์ลูกที่สองของสัญญาณ INTA ให้กับ 8259A จากนั้น 8259A ก็จะส่งเวคเตอร์ของการอินเทอรัพท์ ออกมาบนบัสข้อมูล 8088 จะใช้ค่าเวคเตอร์นี้ไปเปิดตารางเวคเตอร์เพื่อหาตำแหน่งแอดเดรสเริ่มต้น (ค่าแอดเดรสนี้จะถูกโหลดให้กับรีจีสเตอร์ CS และ IP) ของโปรแกรมที่ตอบสนองต่อการขออินเทอรัพท์ที่เกิดขึ้น

6) 8088 จะทำการดิสเอเบิลการขออินเทอรัพท์ (แฟลก I จะถูกรีเซ็ทโดยอัตโนมัติ), เก็บข้อมูลในรีจีสเตอร์แฟลก (คือสถานะการทำงานของ 8088 นั่นเอง) ไว้ในสแตก, เก็บค่าแอดเดรสของคำสั่งถัดไป (คือ ค่าในรีจีสเตอร์ CS และ IP) ที่ 8088 จะกลับมาทำหลังเสร็จจากการทำงานในโปรแกรมตอบสนองต่อการอินเทอรัพท์แล้ว และทำการโหลดค่า CS และ IP ที่อยู่ในตำแหน่งแอดเดรสที่กำหนดโดยเวคเตอร์การอินเทอรัพท์จากตารางเวคเตอร์ ซึ่งจะเป็นผลให้ 8088 กระโดดไปทำงานในตำแหน่งที่กำหนดโดยค่าของ CS และ IP ที่โหลดเข้ามาใหม่นี้

7) โปรแกรมตอบสนองการอินเทอรัพท์สั่งให้ 8088 ทำการเก็บค่าของรีจีสเตอร์ต่าง ๆ ที่ถูกใช้ในโปรแกรมตอบสนองการอินเทอรัพท์นี้ไว้บนสแตก (คำสั่ง PUSH) เพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้นได้

8) ในกรณีที่ต้องการให้มีอินเทอรัพท์ซ้อนขึ้นได้ โปรแกรมตอบสนองการอินเทอรัพท์จะส่งคำสั่ง EOI ให้กับ 8259A และเซ็ทแฟลก I ของ 8088

9) 8088 เข้าสู่การทำงานในส่วนของโปรแกรมที่ตอบสนองต่ออุปกรณ์ที่ขออินเทอรัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

023115

10) หลังเสร็จจากการทำงาน ในโปรแกรมที่ตอบสนองต่ออุปกรณ์ที่ขออินเทอรัพท์แล้ว ต้องใช้คำสั่ง POP เพื่อดึงเอาค่าของรีจีสเตอร์ต่าง ๆ ที่เก็บไว้บนสแตก ออกมาให้กับรีจีสเตอร์เหล่านั้นตามเดิม

11) ภายในโปรแกรมตอบสนองการอินเทอรัพท์ จะสิ้นสุดด้วยคำสั่ง IRET (รีเทิน ฟรอมอินเทอรัพท์ (Retern From Interrupt) ซึ่งจะทำให้ 8088 ทำการเอนเนเบิลการขออินเทอรัพท์, นำเอาค่าของรีจีสเตอร์ CS, IP และ แฟลก ที่เก็บไว้บนสแตกมาคืนให้กับรีจีสเตอร์ทั้งสามตามเดิม ซึ่งจะทำให้ 8088 กลับเข้าสู่การทำงานในโปรแกรมเดิม ก่อนที่จะถูกอินเทอรัพท์ได้ตามปกติ

2.4.4 โปรแกรมตอบสนองการขออินเทอรัพท์

หลังจากการเกิดขั้นตอนต่าง ๆ ที่กล่าวถึงมาแล้ว 8088 จะเข้าสู่การทำงานในโปรแกรมตอบสนองการขออินเทอรัพท์ ที่มีแอดเดรสเริ่มต้นตรงกับแอดเดรสที่ 8088 คำนวณได้ ซึ่งในส่วนของโปรแกรมนี้จะทำให้ 8088 ทำงานเพื่อตอบสนองต่ออุปกรณ์ที่ขออินเทอรัพท์ อย่างไรก็ตาม ภายในโปรแกรมตอบสนองการอินเทอรัพท์นั้น นอกจากจะประกอบด้วยส่วนของโปรแกรมที่ตอบสนองต่ออุปกรณ์ที่ขออินเทอรัพท์โดยตรงแล้ว ยังจำเป็นที่จะต้องเพิ่มโปรแกรมอีกส่วนหนึ่งเข้าไป เพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้นได้ เมื่อ 8088 กลับเข้าสู่การทำงานในโปรแกรมเดิมที่ 8088 ทำอยู่ก่อนที่จะถูกอินเทอรัพท์ ซึ่งส่วนของโปรแกรมที่เพิ่มเข้าไปนี้จะช่วยในการแก้ปัญหาที่อาจเกิดขึ้นได้ โดยจะเพิ่มส่วนของโปรแกรมที่สั่งให้ 8088 ทำการเก็บค่าของรีจีสเตอร์ที่ถูกใช้งานในโปรแกรมตอบสนองการอินเทอรัพท์ไว้ ก่อนที่จะเริ่มการทำงานในส่วนของโปรแกรมที่ตอบสนองต่ออุปกรณ์ที่ขออินเทอรัพท์ และ ก่อนที่จะกลับเข้าสู่การทำงานในส่วนของโปรแกรมเดิมก็จะต้องทำการดึงเอาข้อมูลที่เก็บไว้นั้น คืนให้กับรีจีสเตอร์ต่าง ๆ เหล่านั้นด้วย ซึ่งโดยทั่วไปแล้วจะใช้คำสั่ง PUSH และ POP ในการเก็บและดึงข้อมูลออกจากหน่วยความจำ

บิท ISR ของ 8259A จะถูกเซ็ท เมื่อมีการอินเทอรัพท์เกิดขึ้น ซึ่งจะเป็นผลให้ 8259A ไม่รับการขออินเทอรัพท์ในแชนแนลที่มีลำดับความสำคัญต่ำกว่าหรือเท่ากับแชนแนลที่ขออินเทอรัพท์อยู่นี้ และ เนื่องจากภายในไอบีเอ็มพีซีไม่ได้โปรแกรมให้ 8259A อยู่ในโหมด AEOI ดังนั้นภายในโปรแกรมตอบสนองการอินเทอรัพท์นั้น เราจะต้องเพิ่มส่วนของโปรแกรมที่ทำการส่งคำสั่ง EOI ให้กับ 8259A ด้วยเพื่อรีเซ็ทบิท ISR ซึ่งโดยทั่วไปแล้วมักจะใส่คำสั่ง EOI นี้ไว้ก่อนที่จะสิ้นสุดโปรแกรมตอบสนองการอินเทอรัพท์ อย่างไรก็ตามในบางกรณี เราอาจจะต้องการให้ 8259A ยอมรับการขออินเทอรัพท์ซ้อนได้ ถึงแม้ว่า 8088 ยังไม่เสร็จจากการทำงานในโปรแกรม

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอบสนองการอินเทอรัพท์ก็ตาม ในกรณีเช่นนี้เราจะต้องใส่คำสั่ง EOI ไว้ในตำแหน่งอื่นของโปรแกรม (ขึ้นอยู่กับว่าเราต้องการให้ 8259A ยอมรับการขออินเทอรัพท์ซ้อนได้ในช่วงเวลาใดของโปรแกรมตอบสนองการอินเทอรัพท์)

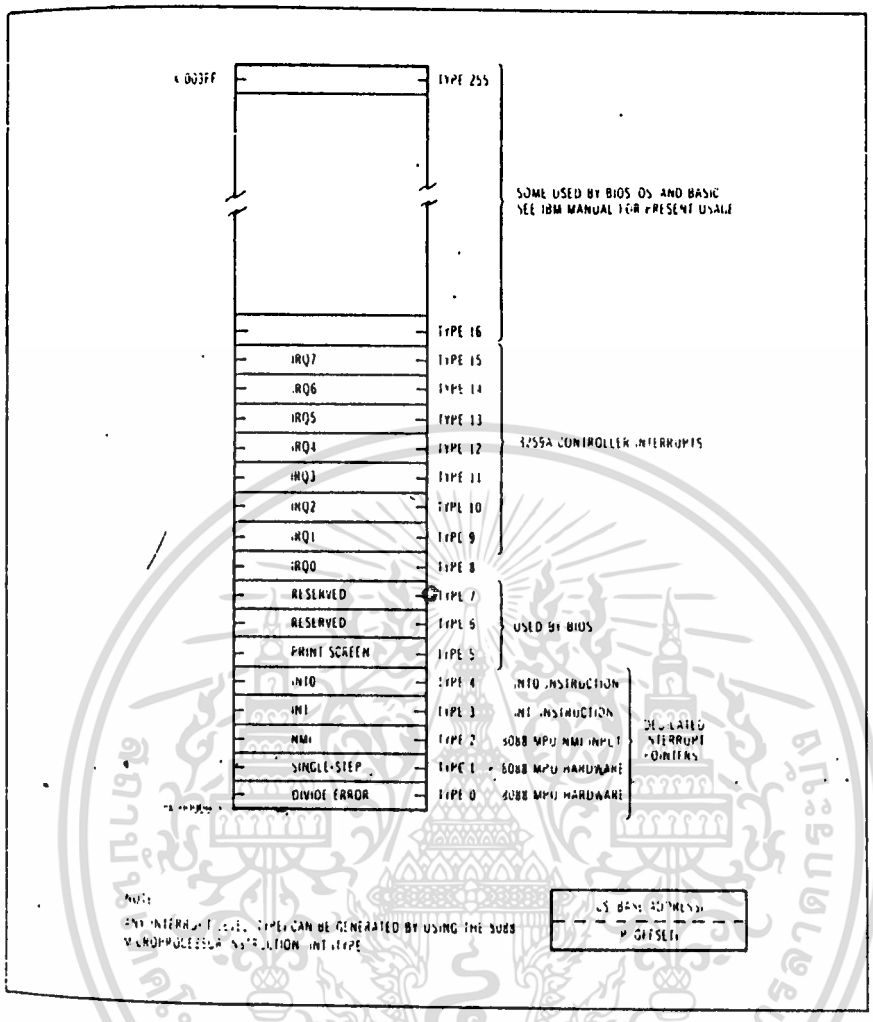
หลังจากที่ 8259A ได้รับคำสั่ง EOI แล้วก็จะรีเซ็ทบิท ISR ทำให้วงจรอินเทอร์เฟสหรืออุปกรณ์ภายนอกสามารถขออินเทอรัพท์ซ้อนเข้ามาได้ แต่ 8259A ยังไม่สามารถอินเทอรัพท์ 8088 ได้ เนื่องจากการขออินเทอรัพท์แบบมาส์คเคเบิล ยังคงถูก 8088 ดิสเอเบิลอยู่ (เมื่อถูกอินเทอรัพท์ 8088 จะดิสเอเบิลการขออินเทอรัพท์แบบมาส์คเคเบิลโดยอัตโนมัติ) ดังนั้นเราจะต้องเอนเนเบิลการขออินเทอรัพท์แบบมาส์คเคเบิลนี้ด้วย ซึ่งจะทำได้โดยการเซ็ทแฟลก I (แฟลกอินเทอรัพท์(Interrupt flag)) ให้เป็น 1 โดยใช้คำสั่ง STI ในกรณีที่เราต้องการเอนเนเบิลการขออินเทอรัพท์แบบมาส์คเคเบิล หลังจากที่ 8088 เสร็จจากการทำงานในโปรแกรมตอบสนองการอินเทอรัพท์นั้น ซึ่งจะทำให้ 8088 นำเอาค่าแฟลกที่เก็บไว้ในสแตกออกมาคืนให้กับรีจีสเตอร์แฟลก ซึ่งจะเป็นการเอนเนเบิลการขออินเทอรัพท์โดยอัตโนมัติ เนื่องจากแฟลก I ที่ถูกเก็บไว้บนสแตกนั้น จะต้องถูกเอนเนเบิลอยู่ก่อนแล้วจึงจะเกิดการอินเทอรัพท์ขึ้นได้

2.4.5 ตารางเวคเตอร์บนไอบีเอ็มพีซี

8088 ใช้หน่วยความจำ 1024 ไบท์ล่าง คือจาก 00000H จนถึง 003FFH สำหรับการอินเทอรัพท์ของระบบ ซึ่งจะมีการอินเทอรัพท์ได้ทั้งสิ้น 256 แบบ โดยหน่วยความจำส่วนนี้จะถูกใช้ในการเก็บแอดเดรสของโปรแกรมย่อยที่ตอบสนองต่อการอินเทอรัพท์ทั้ง 256 แบบนี้ และเนื่องจากการอ้างแอดเดรสในหน่วยความจำต้องกำหนดค่าให้กับรีจีสเตอร์ 2 ตัว คือ รีจีสเตอร์ IP และ CS ซึ่งเป็นรีจีสเตอร์ขนาด 16 บิททั้งคู่ ดังนั้นหน่วยความจำที่ใช้เก็บแอดเดรสสำหรับการอินเทอรัพท์แต่ละแบบจะมีจำนวน 4 ไบท์ คือ 2 ไบท์ใช้สำหรับเก็บค่าแอดเดรสที่จะป้อนให้กับ CS (คือ แอดเดรสเซกเมนท์) และอีก 2 ไบท์ใช้สำหรับเก็บค่าแอดเดรสที่จะป้อนให้กับ IP (คือแอดเดรสออฟเซ็ท) ซึ่งเมื่อนำแอดเดรสทั้งสองมาประกอบกันแล้ว ก็จะได้แอดเดรสของโปรแกรมที่ตอบสนองต่อการขออินเทอรัพท์ที่เกิดขึ้น ซึ่งเมื่อรวมทั้ง 256 แบบแล้ว ก็จะเป็นเนื้อที่ในหน่วยความจำเท่ากับ 256*4 หรือ 1024 ไบท์พอดี

ในกรณีที่มีการขออินเทอรัพท์เกิดขึ้น 8088 จะใช้ตารางนี้ เพื่อหาตำแหน่งแอดเดรสของโปรแกรมตอบสนองการอินเทอรัพท์ ซึ่งการอินเทอรัพท์ที่กล่าวถึงนี้อาจจะเป็นการอินเทอรัพท์ทางซอฟท์แวร์ (คำสั่ง INT) หรือ ฮาร์ดแวร์ภายนอกก็ได้ รูปที่ 2.6 จะแสดงการใช้งานหน่วยความจำ 1024 ไบท์ล่างของไอบีเอ็มพีซี สำหรับการขออินเทอรัพท์ทั้ง 256 แบบ

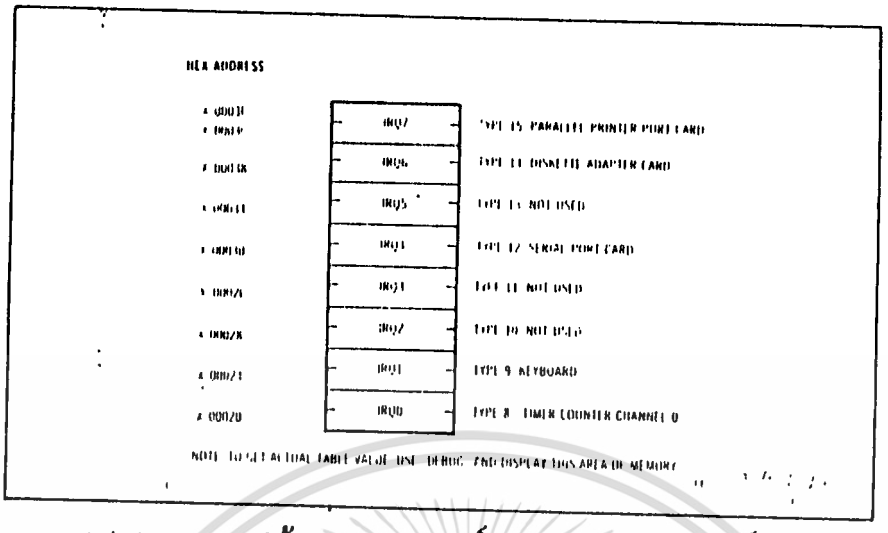
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.6 การใช้งานหน่วยความจำ 1024 ไบท์ล่างของไอบีเอ็มพีซีสำหรับการขออินเทอรัพท์ทั้ง 256 แบบ

จะเห็นได้ว่า แอดเดรสของตารางเวคเตอร์สำหรับไอบีเอ็มพีซี ในกรณีนี้จะอยู่ระหว่าง 00020H จนถึง 0003FH (0003CH+3) ในรูปที่ 2.7 จะแสดงการใช้ตารางเวคเตอร์ในแอดเดรส 00020H จนถึง 0003FH ของไอบีเอ็มพีซี (อินเทอรัพท์ แบบ 8-15) ซึ่งถูกกำหนดโดยโปรแกรมในส่วนของไบออส ตัวอย่างเช่น ค่าแอดเดรสของโปรแกรมตอบสนองการอินเทอรัพท์ใน level 0 (IRQ0) จะเก็บไว้ในแอดเดรส 00020H จนถึง 00023H ค่าแอดเดรสในที่นี้ จะประกอบด้วยค่าของรีจีสเตอร์ CS และ IP ของโปรแกรมตอบสนองการอินเทอรัพท์ โดยจะเก็บค่า CS ไว้ในแอดเดรส 2 ไบท์บน และค่า IP ไว้ในแอดเดรส 2 ไบท์ล่าง เช่น ในกรณีของอินเทอรัพท์ level 0 จะเก็บค่าแอดเดรสของโปรแกรมตอบสนองการอินเทอรัพท์ไว้ในตารางเวคเตอร์ โดยเก็บค่าของ CS ไว้ในแอดเดรส 00022H-00023H และค่าของ IP ไว้ในแอดเดรส 00020H-00021H เป็นต้น

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.7 การใช้ตารางเวคเตอร์ของ 8259A ในไอบีเอ็มพีซี

จากที่กล่าวถึงข้างต้นนั้น จะเห็นได้ว่า เมื่อ 8088 ได้แอดเดรสขนาด 10 บิทแล้ว 8088 ก็จะใช้ค่าแอดเดรสนี้ในการอ้างถึงข้อมูลในตารางเวคเตอร์ เพื่อหาแอดเดรสที่แท้จริงของโปรแกรมตอบสนองการอินเทอรัพท์ ตัวอย่างเช่น ในกรณีที่เกิดการขออินเทอรัพท์ใน level 3 (สัญญาณขออินเทอรัพท์เกิดขึ้นที่ขา IRQ3 ของ 8259A) เมื่อ 8259A ได้รับพัลส์ INTA ลูกที่สอง 8259A ก็จะส่งค่าเวคเตอร์ 0BH (หรือ 0000 1011 ในฐานสอง) ให้กับ 8088 จากนั้น 8088 จะคำนวณหาแอดเดรสขนาด 10 บิท โดยเพิ่มบิทที่มีค่าเป็น 0 จำนวน 2 บิทต่อท้ายค่าเวคเตอร์ที่ได้รับ ซึ่งจะทำให้ได้ค่าแอดเดรสขนาด 10 บิทเป็นๆ 02CH หรือ 0000 0010 1100 ในฐานสอง หลังจากนั้น 8088 จะใช้ค่าแอดเดรสนี้ในการเปิดตารางเวคเตอร์ โดยข้อมูลในตารางเวคเตอร์ในแอดเดรส 0002EH กับ 0002FH จะถูกใช้เป็นค่าของรีจีสเตอร์ CS

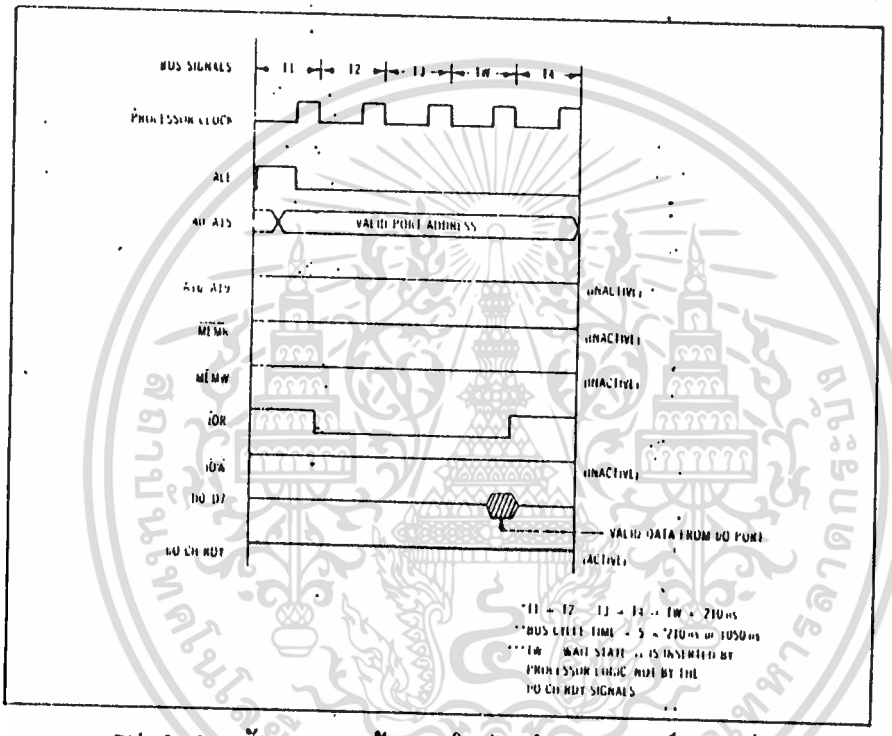
2.5 อินพุท-เอาท์พุทพอร์ทของเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที

2.5.1 บัสไซเคิลอ่านอินพุท-เอาท์พุทพอร์ท (I/O port read bus cycle)

ไมโครโปรเซสเซอร์ (Microprocessor) 8088 ทำงานคำสั่ง IN ทำให้เกิดบัสไซเคิลนี้ เพื่ออ่านข้อมูล (Data) จากพอร์ทไดพอร์ทหนึ่งในพอร์ททั้งหมดที่มีอยู่ บัสไซเคิลมีช่วงเวลาเท่ากับ 5 สัญญาณนาฬิกา (Clock) หรือประมาณ 1.05 ไมโครวินาที (Microsecond) บนเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ทีทำงานด้วยความเร็วสัญญาณนาฬิกา 4.77 เมกกะเฮิร์ท (MegaHertz) ในช่วงบัสไซเคิลอ่านอินพุท-เอาท์พุทพอร์ทไมโครโปรเซสเซอร์ 8088 ขับแอดเดรส (Address) 16 บิทของพอร์ทลงบนแอดเดรสบัสของระบบ (System Address Bus) ข้อสังเกตุ 4 บิทบนของแอดเดรสบัสไม่แอคทีฟ (Active) ในช่วงบัสไซเคิลนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่วงสัญญาณนาฬิกา T1 สัญญาณบัส ALE แอคทีพเป็นตัวบ่งว่าในแอดเดรสบัสมีข้อมูลแอดเดรสที่ถูกต้อง ช่วงสัญญาณนาฬิกา T2 สัญญาณควบคุมบัส (Bus Control Signal) IOR แอคทีพเป็นตัวบ่งว่าบัสไซเคิลนี้เป็นไซเคิลการอ่านอินพุท-เอาท์พุทพอร์ท และให้พอร์ทที่ถูกอ้างวางข้อมูลลงบนบัสข้อมูล (Data Bus) ไมโครโปรเซสเซอร์ 8088 ทำการสุ่ม (Sample) อ่านข้อมูลบนบัสข้อมูลในตอนเริ่มช่วงสัญญาณนาฬิกา T4 และสัญญาณบัส IOR ไม่แอคทีพ บัสไซเคิลสิ้นสุดเมื่อจบช่วงสัญญาณนาฬิกา T4 รูป 2.8 แสดงสัญญาณของบัสไซเคิลอ่านอินพุท-เอาท์พุทพอร์ท



รูป 2.8 สัญญาณของบัสไซเคิลอ่านอินพุท-เอาท์พุทพอร์ท

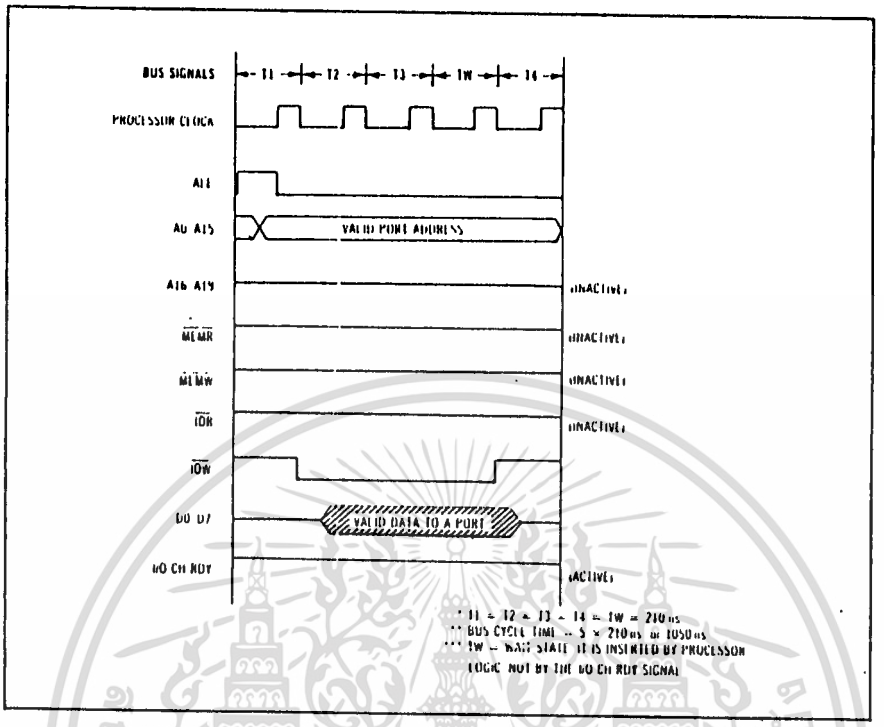
2.5.2 บัสไซเคิลเขียนอินพุท-เอาท์พุทพอร์ท (I/O Port Write Bus Cycle)

ไมโครโปรเซสเซอร์ (Microprocessor) 8088 ทำงานคำสั่ง OUT ทำให้เกิดบัสไซเคิลนี้ เพื่อเขียนข้อมูลจากไมโครโปรเซสเซอร์ 8088 ไปยังพอร์ทใดพอร์ทหนึ่งในพอร์ททั้งหมดที่มีอยู่ บัสไซเคิลมีช่วงเวลาเท่ากับ 5 สัญญาณนาฬิกาหรือประมาณ 1.05 ไมโครวินาที เพียงบิทที่ 0 ถึง 15 ของแอดเดรสบัสเท่านั้นที่ใช้ในการอ้างแอดเดรสของอินพุท-เอาท์พุทพอร์ท

ช่วงสัญญาณนาฬิกา T1 สัญญาณบัส ALE แอคทีพเป็นตัวบ่งว่าในแอดเดรสบัสมีข้อมูลแอดเดรสที่ถูกต้อง ช่วงสัญญาณนาฬิกา T2 ต่อมาสัญญาณควบคุมบัส IOW แอคทีพเป็นตัวบ่งว่าบัสไซเคิลนี้เป็นบัสไซเคิลเขียนอินพุท-เอาท์พุทพอร์ท และให้พอร์ทที่ถูกอ้างดึงข้อมูลจากบัสข้อมูล หลังจากช่วงสัญญาณนาฬิกา T2 ไม่นานไมโครโปรเซสเซอร์ 8088 ขับข้อมูลให้กับพอร์ทลงบนบัสข้อมูล ที่ช่วงเริ่มช่วงสัญญาณนาฬิกา T4 สัญญาณควบคุมบัส IOW ไม่แอคทีพ บัสไซเคิลสิ้นสุดเมื่อจบช่วงสัญ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฌาณนาฬิกา T4 รูป 2.9 แสดงสัญญาณของบัสไซเคิลเขียนอินพุท-เอาท์พุทพอร์ท



รูป 2.9 สัญญาณของบัสไซเคิลเขียนอินพุท-เอาท์พุทพอร์ท

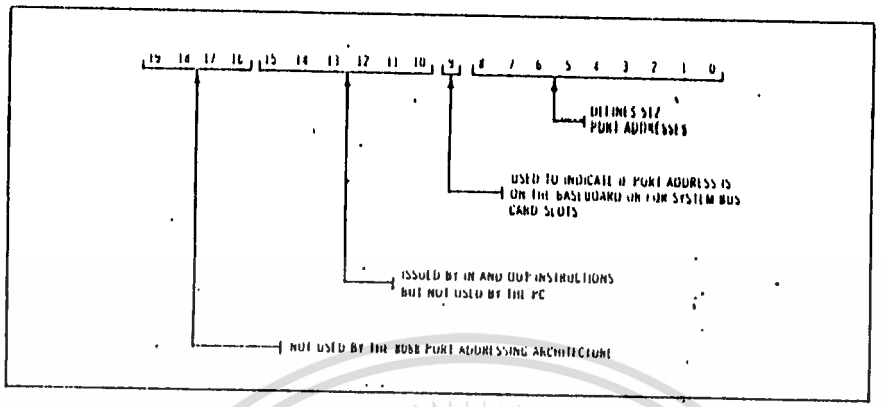
2.5.3 การอ้างพอร์ท (Port Address)

ด้วยการใช้ดิจิตอลอินพุทและเอาท์พุทพอร์ท เครื่องคอมพิวเตอร์สามารถควบคุมและรับรู้การทำงานของอุปกรณ์สนับสนุน และ อินพุท-เอาท์พุทอแดปเตอร์ (I/O Adapter) พอร์ทเหล่านี้สามารถอ้างได้ในที่ว่างของอินพุท-เอาท์พุทพอร์ทแอดเดรสของไมโครโปรเซสเซอร์ 8088 ไมโครโปรเซสเซอร์สามารถส่งข้อมูลไปยังพอร์ทเหล่านี้ด้วยคำสั่ง OUT และรับข้อมูลจากพอร์ทด้วยคำสั่ง IN โครงสร้างของไมโครโปรเซสเซอร์ 8088 สามารถให้ใช้งานอินพุท-เอาท์พุทพอร์ทแอดเดรสได้ทั้งหมด 65,536 พอร์ท แต่ในเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ทีไม่ได้ใช้แอดเดรสทั้งหมด ใช้เพียง 10 บิทล่างของแอดเดรสในการอ้างอุปกรณ์หรือพอร์ทเท่านั้น

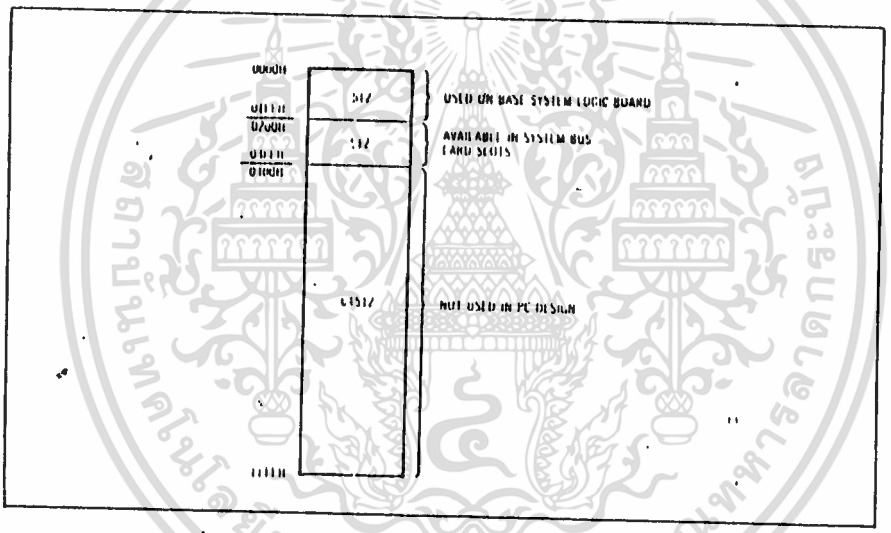
บิท 9 ในการอ้างแอดเดรส อินพุท-เอาท์พุทพอร์ทมีความหมายพิเศษดังนี้ เมื่อไม่แอคทีฟ บัสของระบบ (System Bus) ไม่สามารถรับข้อมูลจากสลอต (Slot) ที่อยู่บนซีสเต็มบอร์ด (System Board) และรับข้อมูลจากอุปกรณ์และอินพุท-เอาท์พุทพอร์ทบนซีสเต็มบอร์ดเท่านั้น เมื่อบิทนี้แอคทีฟ ถึงสามารถรับข้อมูลจากสลอตได้ หมายความว่า 1,024 พอร์ทที่สามารถใช้งานได้แบ่งออกเป็น 2 ส่วนเท่า ๆ กัน 512 พอร์ทแรกอยู่บนซีสเต็มบอร์ดเท่านั้น และอีก 512 พอร์ทอยู่บนสลอตเท่านั้น รูป 2.10 แสดงการใช้แอดเดรสของอินพุท-เอาท์พุทของไมโครโปรเซสเซอร์ 8088 รูป 2.11 แสดงการใช้งานที่ว่างของอินพุท-เอาท์พุทพอร์ทของไมโครโปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซสเซอร์ 8088 บนเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที



รูป 2.10 การใช้แอดเดรสของอินพุท-เอาท์พุทของไมโครโปรเซสเซอร์ 8088

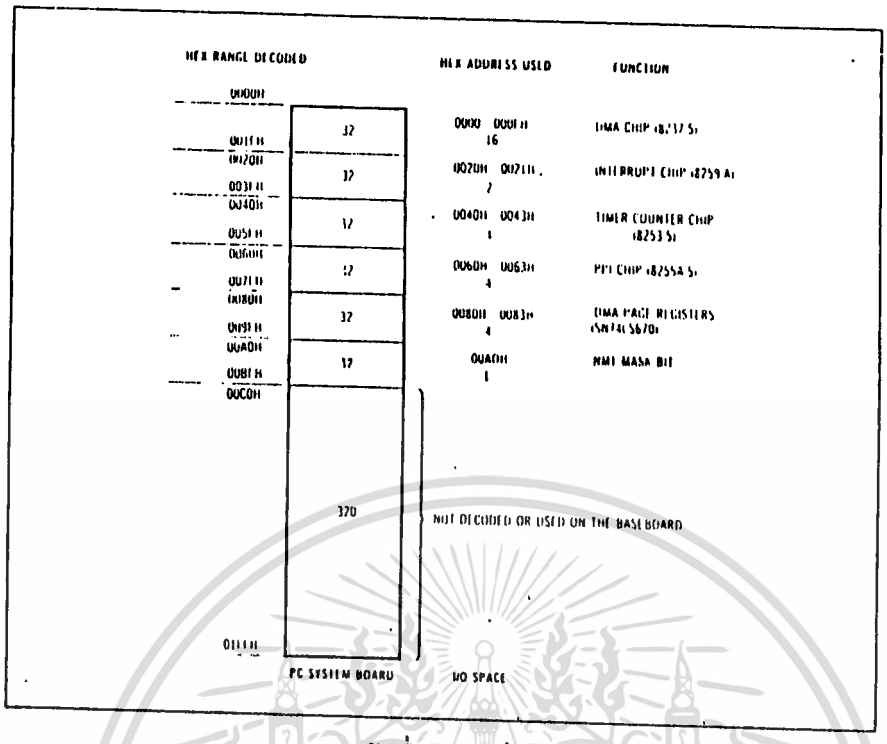


รูป 2.11 การใช้งานที่ว่างของอินพุท-เอาท์พุทพอร์ทของไมโครโปรเซสเซอร์ 8088 บนเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที

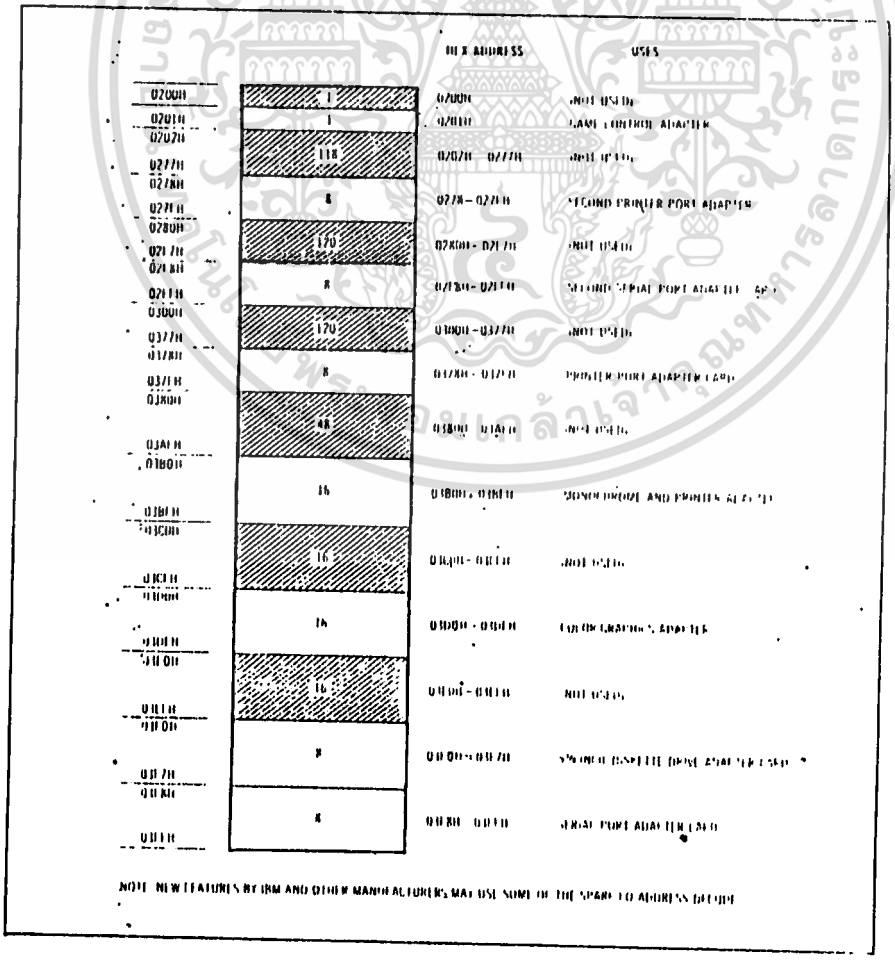
2.5.4 แผนภาพใช้งานแอดเดรสของอินพุท-เอาท์พุทพอร์ท (I/O Port Address Map)

แผนภาพใช้งานแอดเดรสของอินพุท-เอาท์พุทพอร์ท สามารถแบ่งออกเป็น 2 กลุ่ม กลุ่มแรกเป็นที่ว่างแอดเดรส 0000 Hex ถึง 01FF Hex หรือส่วนที่อยู่บนซีสเต็มบอร์ด แอดเดรสพอร์ทเหล่านี้ใช้อ้างอุปกรณ์สนับสนุนและอินพุท-เอาท์พุทของไมโครโปรเซสเซอร์ 8088 บนซีสเต็มบอร์ด รูป 2.12 แสดงหน้าที่ของพอร์ทบนซีสเต็มบอร์ด รูป 2.13 แสดงการใช้งานส่วนที่สองของที่ว่างแอดเดรส ในขณะที่แอดเดรส 200 Hex ถึง 3FF Hex ซึ่งสามารถอ้างถึงโดยผ่านทางบัสของระบบและบัสบนซีสเต็มบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.12 หน้าที่ของพอร์ทบนซิสเต็มบอร์ด



รูป 2.13 การใช้งานส่วนที่สองของที่ว่างแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.5 เทคนิคการอ้างแอดเดรสของอินพุท-เอาท์พุทพอร์ท (I/O Port Address Decoding Technique)

โดยการหาที่ว่างแอดเดรสและหากลุ่มของแอดเดรสที่ยังไม่มีการใช้งาน และสร้างวงจรอ้างที่เหมาะสมเพื่ออ้างถึงแอดเดรสหรือกลุ่มของแอดเดรสของอินพุท-เอาท์พุทพอร์ท

มีข้อกำหนดทางเวลา 2 ข้อในการถอดสัญญาณ (Decode) แอดเดรสพอร์ท ข้อแรกคือ เมื่อเริ่มบัสไซเคิลของอินพุท-เอาท์พุท ทำการถอดสัญญาณแอดเดรสล่าช้ามาก ซึ่งอาจเกิดหลังจากสัญญาณควบคุมบัส IOR และ IOW แอคทีฟแล้ว เนื่องจากบิทแอดเดรสของระบบเกิดการเปลี่ยนแปลง การถอดสัญญาณอาจถอดได้แอดเดรสพอร์ทอื่น เมื่อการถอดสัญญาณไม่ถูกต้อง ที่ถูกและ (And) กับสัญญาณควบคุมบัส อาจทำให้เกิดการเขียนหรืออ่านจากแอดเดรสพอร์ทที่ผิดในเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ทีสามารถให้การถอดสัญญาณให้ล่าช้ามากที่สุด 92 นาโนวินาที (Nanosecond)

ข้อสองคือ ขณะจบบัสไซเคิลของอินพุท-เอาท์พุทพอร์ท ถ้าสัญญาณควบคุมบัส IOW ล่าช้าและการถอดสัญญาณเร็ว อาจทำให้เกิดการเขียนข้อมูลไปที่แอดเดรสที่ถอดมาจากบัสไซเคิลถัดไปที่ตอนท้ายของสัญญาณเขียน ในเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ทีสามารถให้สัญญาณ IOW ซ้ำได้ไม่เกิน 200 นาโนวินาที แต่เวลาวิกฤตที่สุดคือ เวลาสัมพันธ์ของขอบหลังสัญญาณ IOW กับสัญญาณข้อมูลที่ถูกต้องบนบัสข้อมูล ถ้าสัญญาณ IOW ช้ามากกว่า 120 นาโนวินาที แอดเดรสพอร์ทอาจได้รับข้อมูลที่ไม่ถูกต้อง ในลักษณะเดียวกัน ถ้าสัญญาณ IOR ล่าช้า จะทำให้เวลาในการอ่านข้อมูลจากอินพุท-เอาท์พุทพอร์ทลดลง

2.5.6 การขยายบัส (Bus Extension)

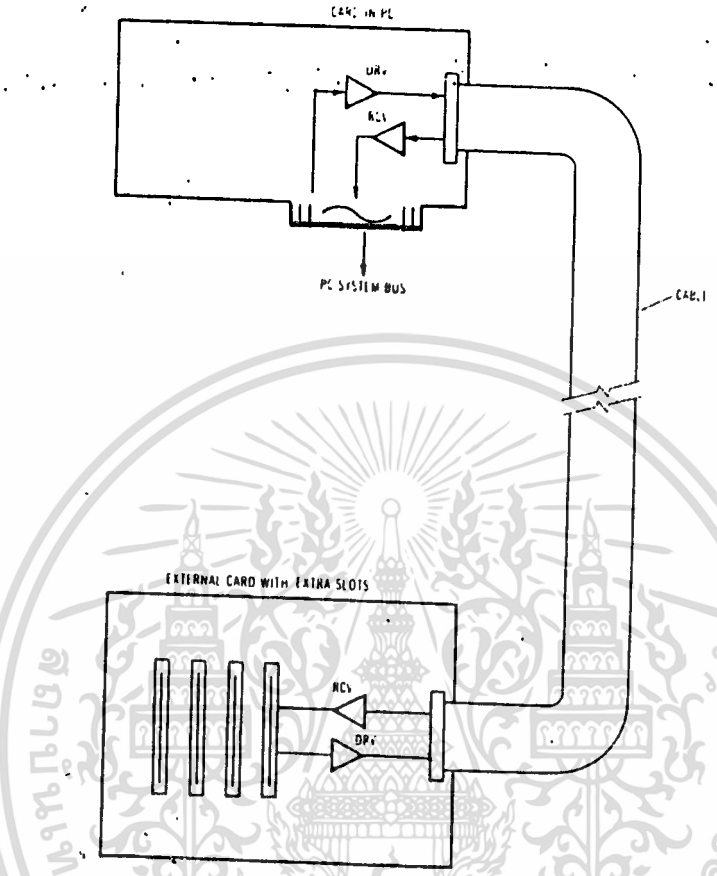
ในเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ทีมีจำนวนสลอตที่ว่างจำกัด และอาจไม่เหมาะสมกับความต้องการ จึงต้องขยายบัสของระบบไปยังภายนอก การขยายบัสทำให้การทำงานไม่มีข้อจำกัดเรื่องที่ว่างหรือกำลังจ่ายไฟ

แนวคิดพื้นฐานในการขยายบัสมี 2 ส่วน ส่วนแรกคือ มีการ์ด (Card) ซึ่งเสียบในสลอตบนซีสเต็มบอร์ด ทำหน้าที่รับและขับเอาท์พุทของการ์ด ซึ่งเสียบบนซีสเต็มบอร์ด และมีสลอตใหม่ให้ใช้เพิ่ม รูป 2.14 แสดงส่วนประกอบและแนวคิดการขยายบัส

หลักการที่สำคัญคือ การเพิ่มกำลังขับให้กับสัญญาณ และขับสัญญาณจากบัสไปยังสาย (Cable) ซึ่งสัญญาณโดยทั่วไปสามารถนำได้ไม่ยุ่งยาก หน้าที่หลักของวงจรขับคือ การลดค่าความจุ (Capacitance) ของสายไม่ให้มีผลกระทบในด้านการหน่วงเวลาหรือการเพี้ยนของสัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในเชิงธุรกิจ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ญาณกับบัสของระบบ และที่อีกข้างหนึ่งของสายซึ่งเป็นส่วนการ์ดสลอตใหม่ สำหรับสัญญาณและ



รูป 2.14 ส่วนประกอบและแนวคิดการขยายบัส

เพิ่มกำลังขับอีกเพื่อขับสลอตบนนี้ อินพุทจากการ์ดสลอตใหม่ก็ผ่านขบวนการเดียวกัน คือ เพิ่มกำลังขับเข้าไปในสาย รับเพิ่มกำลังขับและขับเข้าไปยังบัสของระบบ

2.6 รายละเอียดสัญญาณสลอตบนชิสเต็มบอร์ดของเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที

สัญญาณแสดงในรูปที่ 2.15 ซึ่งมีระดับสัญญาณเหมือนกับสัญญาณทีทีแอล (TTL) ทุกประการ โดยมีรายละเอียดดังนี้

2.6.1 OSC ออสซิลเลเตอร์ (Ocillator) เป็นสัญญาณนาฬิกาที่มีความเร็วสูงประมาณ 14.31818 เมกกะเฮิร์ทและมีดิวตี้ไซเคิล (Duty Cycle) 33% ทิศทางสัญญาณเอาท์พุท

2.6.2 CLK สัญญาณนาฬิกา เป็นสัญญาณนาฬิกาที่อ้างอิงในระบบ โดยการนำสัญญาณออสซิลเลเตอร์มาหาร 3 จึงมีความเร็วประมาณ 4.77 เมกกะเฮิร์ท และมีดิวตี้ไซเคิล 33% ทิศทางสัญญาณเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3 RESET DRV รีเซ็ทไดร์เวอร์ (Reset Driver) เป็นสัญญาณที่ใช้กำหนดค่า

เริ่มต้นทางตรรกศาสตร์ของระบบ สัญญาณนี้จะซิงค์ (Synchronous) กับสัญญาณนาฬิกา ช่วงเปลี่ยนสถานะจากสูงเป็นต่ำ สถานะการทำงานแอคทีฟไฮ (Active High)

2.6.4 A0-A19 สัญญาณนี้ใช้อ้างอิงตำแหน่งข้อมูลในหน่วยความจำหรือตำแหน่งพอร์ทภายในระบบ ในจำนวน 20 บิทจะเข้าถึงข้อมูล 1 เมกกะไบท์ จะแอคทีฟไฮ

2.6.5 D0-D7 เป็นสายข้อมูลบิท 0 ถึงบิท 7 สำหรับไมโครโปรเซสเซอร์ หน่วยความจำ และพอร์ทต่าง ๆ โดยทิศทางการทำงานไบไดเรกชั่น (Bidirection) สถานะการทำงานแอคทีฟไฮ

2.6.6 ALE แอดเดรสแลทช์เอนเนเบิล (Address Latch Enable) เป็นสัญญาณที่ 8288 บัสคอนโทรเลอร์ (Bus Controller) จับสัญญาณแอดเดรสที่ถูกต้องจากไมโครโปรเซสเซอร์ 8088 ทิศทางการทำงานเอาท์พุท สถานะการทำงานแอคทีฟไฮ

2.6.7 I/O CHCK ไอโอแชนแนลเช็ค (I/O Channel Check) ใช้เป็นเครื่องมือในการตรวจจับความผิดพลาดพาริตี้ (Parity Error) ของหน่วยความจำหรือพอร์ทต่าง ๆเมื่อสถานะเป็นโลว์ (Low) ไมโครโปรเซสเซอร์ก็จะรู้ว่าเกิดความผิดพลาดขึ้นในระบบ ทิศทางการทำงานอินพุท สถานะการทำงานแอคทีฟโล (Active Low)

2.6.8 I/O CHRDY ไอโอแชนแนลเรดี้ (I/O Channel Ready) โดยปกติสถานะการทำงานจะไฮอยู่ แสดงว่า พร้อม หากหน่วยความจำหรือพอร์ทเกิดทำงานช้า อุปกรณ์เหล่านั้นก็จะให้สัญญาณนี้เป็นโลว์ เพื่อให้ไมโครโปรเซสเซอร์เพิ่มเวลาในส่วนการอ่านหรือเขียนข้อมูลกับอุปกรณ์เหล่านั้น ทิศทางการทำงานอินพุท

2.6.9 IRQ2-IRQ7 อินเทอรัพท์รีเควส 2-7 (Interrupt Request 2-7) เป็นการจัดลำดับความสำคัญของสัญญาณอินเทอรัพท์ จากอุปกรณ์ต่าง ๆ ไปยังไมโครโปรเซสเซอร์ โดยเรียงลำดับความสำคัญต่ำสุดจาก IRQ7 ไปยังความสำคัญสูงสุด IRQ2 เมื่อเกิดอินเทอรัพท์ขึ้น กล่าวคือ สัญญาณเปลี่ยนสถานะจากโลว์เป็นไฮ ไมโครโปรเซสเซอร์ก็จะตอบสนองโดยการกระโดดไปทำงานในส่วนที่เรียกว่า อินเทอรัพท์เซอวิสรูทีน ทิศทางการทำงานอินพุท สถานะการทำงานแอคทีฟไฮ

2.6.10 IOR ไอโอรีดคอมมาน (I/O Read Command) เป็นสัญญาณที่ใช้ในการติดต่อกับพอร์ทต่าง ๆ เพื่อให้ข้อมูลจากพอร์ทเข้าดาต้าบัส (Data Bus) สัญญาณนี้ถูกสร้างขึ้นจากไมโครโปรเซสเซอร์หรือดีเอ็มเอคอนโทรเลอร์ (DMA controller) ทิศทางการทำงานเอาท์พุท สถานะการทำงานแอคทีฟโลว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.11 IOW ไอโอไรท์คอมมาน (I/O Write Command) เป็นสัญญาณที่ใช้ในการติดต่อกับพอร์ทต่าง ๆ เพื่อให้พอร์ทนั้นรับข้อมูลจากบัสข้อมูล สัญญาณนี้ถูกสร้างขึ้นจากไมโครโปรเซสเซอร์หรือดีเอ็มเอคอนโทรเลอร์ ทิศทางการทำงานเอาท์พุท สถานะการทำงานแอคทีฟโลว์

2.6.12 MEMR เมมโมรี่รีดคอมมาน (Memory Read Command) เป็นสัญญาณที่ใช้ในการติดต่อกับหน่วยความจำ เพื่อให้ข้อมูลเข้าสู่บัสข้อมูล สัญญาณนี้ถูกสร้างจากไมโครโปรเซสเซอร์หรือดีเอ็มเอคอนโทรเลอร์ ทิศทางการทำงานเอาท์พุท สถานะการทำงานแอคทีฟโลว์

2.6.13 MEMW เมมโมรี่รีดคอมมาน (Memory Read Command) เป็นสัญญาณที่ใช้ในการติดต่อกับหน่วยความจำ เพื่อให้ข้อมูลจากบัสข้อมูลเข้าสู่หน่วยความจำ สัญญาณนี้ถูกสร้างขึ้นจากไมโครโปรเซสเซอร์หรือดีเอ็มเอคอนโทรเลอร์ ทิศทางการทำงานเอาท์พุท สถานะการทำงานแอคทีฟโลว์

2.6.14 DRQ1-DRQ3 ดีเอ็มเอรีเควส 1 ถึง 3 (DMA Request 1-3) เป็นสัญญาณอินเทอร์รัพท์ที่เกิดจากอุปกรณ์ที่ต้องการให้เกิดขบวนการดีเอ็มเอ โดยเรียงลำดับความสำคัญต่ำสุดจาก DRQ3 ไปยังความสำคัญสูงสุด DRQ1 ทิศทางการทำงานอินพุท สถานะการทำงานแอคทีฟไฮ

2.6.15 DACK0-DACK3 ดีเอ็มเอแอคโนเลจ (DMA Acknowledge 0-3) เป็นสัญญาณตอบรับการอินเทอร์รัพท์ดีเอ็มเอคอนโทรเลอร์ ทิศทางการทำงานเอาท์พุท สถานะการทำงานแอคทีฟโลว์

2.6.16 AEN แอดเดรสเอนเนเบิล (Address Enable) เมื่อสัญญาณนี้เป็นโลว์ แอดเดรสจะถูกใช้ในขบวนการดีเอ็มเอ และใช้ในการติดต่อกับพอร์ทต่าง ๆ ทิศทางการทำงานเอาท์พุท

2.6.17 T/C เทอร์มินอลเคาท์ (Terminal Count) เมื่อเกิดขบวนการดีเอ็มเอ สัญญาณนี้จะเป็นไฮ ทิศทางการทำงานเอาท์พุท

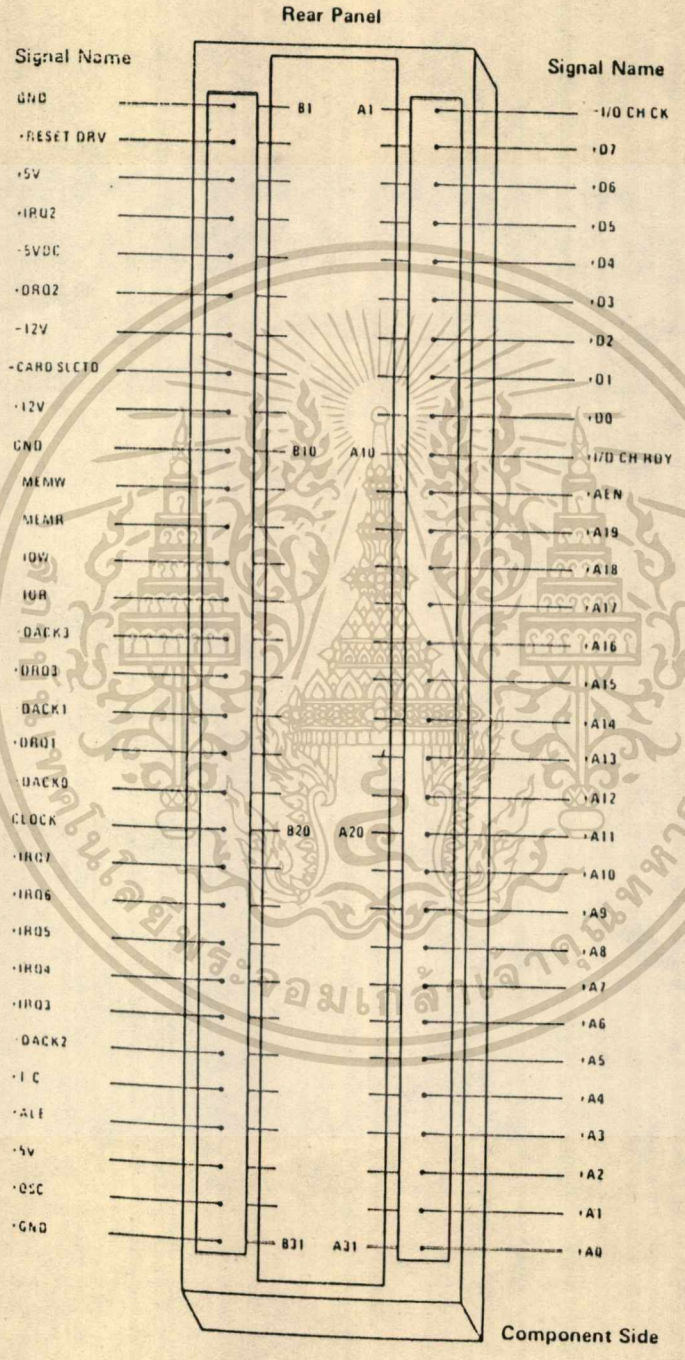
2.6.18 แรงดันต่าง ๆ และกราวด์ ในระบบส่วนขยายพอร์ทจะมีแรงดัน +5v อยู่ 2 ขา แรงดัน -5v อยู่ 1 ขา แรงดัน +12v อยู่ 1 ขา แรงดัน -12v อยู่ 1 ขา และกราวด์อยู่ 3 ขา

2.7 การหาความชื้น

ความชื้นในอากาศอยู่ในรูปของไอน้ำ ซึ่งอาจจะเป็นไอเปียกหรือไอดงก็ได้ อากาศสา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

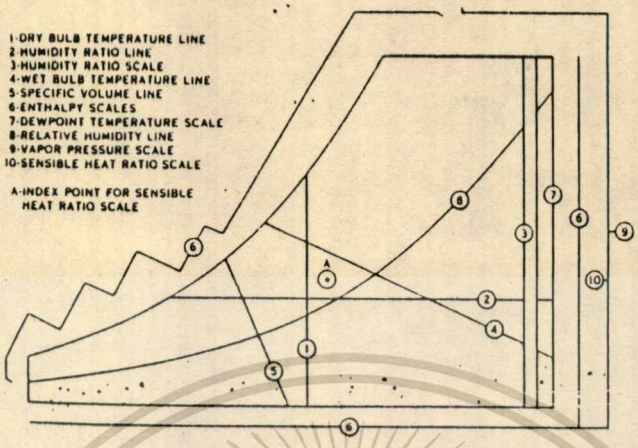
มารถจะอมความชื้นได้ปริมาณมากเมื่ออากาศมีอุณหภูมิสูง และความสามารถในการอมความชื้นของ



รูป 2.15 รายละเอียดสัญญาณบนสลอตของซิสเต็มบอร์ด

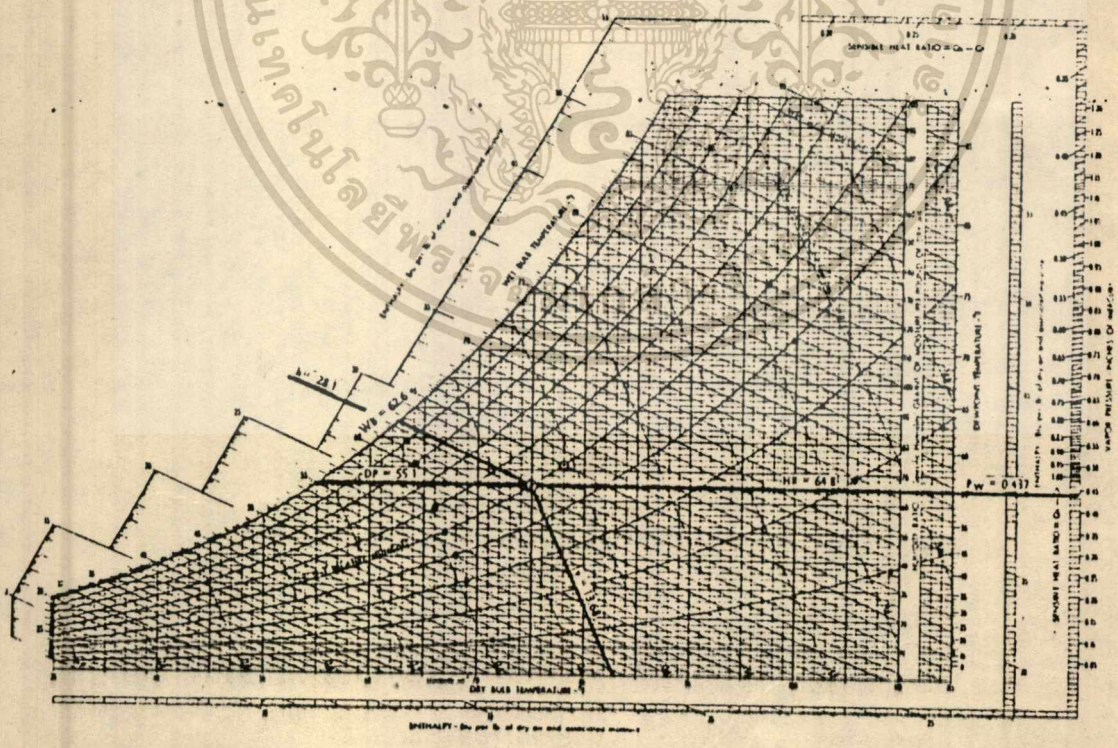
อากาศจะลดลง เมื่ออากาศมีอุณหภูมิต่ำ คุณสมบัติเกี่ยวกับความชื้นในอากาศสามารถหาได้จากไซ
โครเมตริกชาร์ท ซึ่งเป็นรูปหนึ่งของกราฟเขียนขึ้นมาจากการวิเคราะห์วิจัยทางวิศวกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.16 แกนต่าง ๆ ที่แสดงคุณสมบัติของความชื้นในอากาศบนไซโครเมตริกชาร์ท

การหาความชื้นสัมพัทธ์หาได้โดยจุดตัดระหว่างเส้นกราฟของอุณหภูมิดรายบัลย์ (Dry Bulb Temperature Line) กับเส้นกราฟของอุณหภูมิเว็ทบัลย์ (Wet Bulb Temperature Line) บนไซโครเมตริกชาร์ท ดังแสดงรูปไซโครเมตริกชาร์ทดังรูป 2.17



รูป 2.17 ไซโครเมตริกชาร์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความแตกต่าง (F) / อุณหภูมิ (F)	0.5	1.0	1.5	2.0	2.5
80.0	98	96	94	91	90
80.5	98	96	94	91	90
81.0	98	96	94	91	90
81.5	98	96	94	91	90
82.0	98	96	94	91	90
82.5	98	96	94	92	90
83.0	98	96	94	92	90
83.5	98	96	94	92	90
84.0	98	96	94	92	90
84.5	98	96	94	92	90
85.0	98	96	94	92	90
85.5	98	96	94	92	90
86.0	98	96	94	92	90
86.5	98	96	94	92	90
87.0	98	96	94	92	90
87.5	98	96	94	92	90
88.0	98	96	94	92	90
88.5	98	96	94	93	90
89.0	98	96	94	93	90
89.5	98	96	94	93	90
90.0	98	96	94	93	90
90.5	98	96	94	93	90

ตาราง 2.2 ตัวอย่างความชื้นสัมพัทธ์หาได้จากไซโคเมตริกชาร์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการสร้าง

3.1 หน้าที่ของระบบควบคุม

หน้าที่ของระบบควบคุม มีดังนี้

1) ควบคุมอุณหภูมิภายในตู้ให้คงที่ ปกติตู้ฟักไข่ประมาณ 100 องศาฟาเรนไฮต์ ตู้เกิดลูกไก่ประมาณ 98 องศาฟาเรนไฮต์

2) ควบคุมความชื้นภายในตู้ให้คงที่ ปกติตู้ฟักไข่ประมาณ 75% ตู้เกิดลูกไก่ประมาณ 85%

3) พลิกถาดไข่ทุก ๆ ชั่วโมง โดยการเอนถาดไข่ไปทางซ้ายหรือทางขวา เพื่อให้ไข่ได้รับความร้อนและความชื้นทั่วกันทั้งใบ พลิกเฉพาะไข่ในตู้ฟักไข่เท่านั้น

4) แสดงสถานะการทำงานของอุปกรณ์ต่าง ๆ แจ้งเตือนในกรณีไม่สามารถทำงานตามขั้นต้น หรืออุปกรณ์ทำงานไม่สมบูรณ์หรืออยู่ในสภาพที่ไม่ถูกต้อง

3.2 อุปกรณ์ที่ถูกควบคุม มีดังนี้

1) ขดลวดความร้อน (Heater)

2) วาล์ว (Valve) ปิดเปิดท่อน้ำก่อนไปยังหัวพ่นไอน้ำ

3) วาล์วท่อลมไปขับกระบอกลมของรถส่งไข่ในตู้ฟักไข่

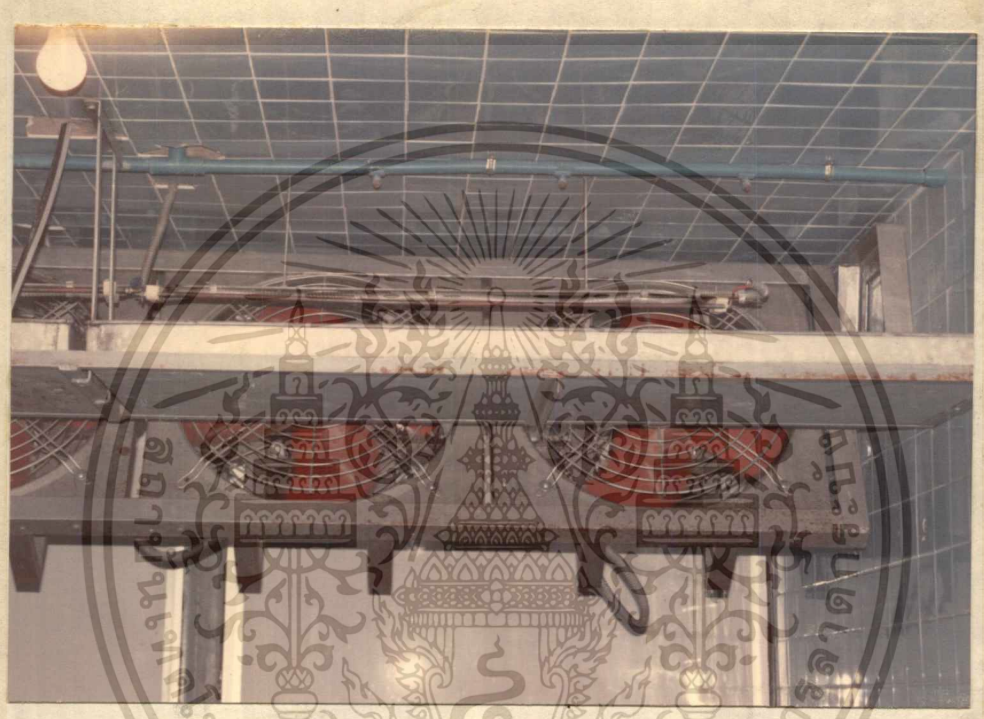
หน้าที่ของอุปกรณ์ที่ถูกควบคุมคือ ขดลวดความร้อนเป็นอุปกรณ์ให้ความร้อน ระบบจะควบคุมให้ขดลวดความร้อนทำงาน เมื่อวัดได้ว่าอุณหภูมิภายในตู้ฟักไข่หรือตู้เกิดลูกไก่ต่ำกว่าค่าที่ตั้งไว้ และหยุดทำงานเมื่อวัดอุณหภูมิภายในตู้ฟักไข่หรือตู้เกิดลูกไก่สูงกว่าค่าที่ตั้งไว้ ขดลวดความร้อนในตู้ฟักไข่มีขนาด 6,000 วัตต์ (Watt) และในตู้เกิดลูกไก่มีขนาด 1,600 วัตต์

วาล์วท่อน้ำเป็นอุปกรณ์ควบคุมการไหลของน้ำในท่อ ซึ่งจะไหลผ่านหัวพ่นไอน้ำ ระบบควบคุมให้วาล์วท่อน้ำเปิด เพื่อให้น้ำไหลผ่าน เมื่อวัดความชื้นภายในตู้ฟักหรือตู้เกิดลูกไก่ต่ำกว่าค่าที่ตั้งไว้ และให้วาล์วท่อน้ำปิดเมื่อวัดความชื้นภายในตู้ฟักไข่หรือตู้เกิดลูกไก่ได้สูงกว่าค่าที่ตั้งไว้ น้ำในท่อจะไหลผ่านปั๊มน้ำเพื่อให้ความดันในท่อน้ำเป็น 4-6 บาร์ (Bar) เพื่อให้เวลาน้ำที่พ่นออกจากหัวพ่นไอน้ำ มีลักษณะเป็นละอองฝอย สามารถกระจายได้อย่างทั่วถึง

วาล์วท่อลมเป็นอุปกรณ์ควบคุมการไหลของลมภายในท่อ ซึ่งจะไหลไปยังกระบอกลมใช้ในการพลิกถาดไข่ ใช้วาล์วท่อลม 2 ตัวในการพลิกถาดไข่ ตัวหนึ่งจะควบคุมให้ถาดไข่เอนไปทางซ้าย และอีกตัวควบคุมให้ถาดไข่พลิกไปทางขวา การใช้งานต้องให้ตัวหนึ่งเปิดและอีกตัวหนึ่งปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้ถาดไข่เอนไปในทิศทางที่ต้องการ ระบบควบคุม ควบคุมให้วาล์วท่อลมเปิดให้ถาดไข่พลิกไปทางซ้ายเมื่อครบ 1 ชั่วโมง ให้พลิกไปทางขวาเมื่อครบอีก 1 ชั่วโมงถัดไป และกลับให้พลิกไปทางซ้ายเมื่อครบ 1 ชั่วโมงต่อมา อย่างนี้ไปเรื่อย ๆ เพื่อทำให้ไข่บนรถในตู้ฟักไข่ได้รับความร้อนและความชื้นอย่างทั่วถึงทั้งใบ



รูป 3.1 พัดลม หัวพ่นไอน้ำ ขดลวดความร้อนและท่อลมภายในตู้ฟักไข่

3.3 อุปกรณ์ที่ต้องตรวจจับสถานะการทำงาน

1) ประตูของตู้ฟักไข่และตู้เกิดลูกไก่

2) พัดลมภายในตู้ฟักไข่และตู้เกิดลูกไก่

ในสภาพการใช้งานปกติ ประตูของทั้งตู้ฟักไข่และตู้เกิดลูกไก่ต้องปิดสนิท เพราะการเปิดประตูตู้ฟักไข่หรือตู้เกิดลูกไก่ ทำให้ความร้อนและความชื้นภายในตู้ถ่ายเทมายังภายนอกตู้ได้อย่างรวดเร็ว ทำให้อุณหภูมิและความชื้นลดลงไม่ได้ตามค่าที่ต้องการ เป็นผลเสียต่อการฟักไข่ ระบบควบคุมต้องตรวจสถานะของประตูตู้ฟักไข่และตู้เกิดลูกไก่อยู่เรื่อย ๆ ระหว่างการใช้งาน และแจ้งเตือนเมื่อมีการเปิดประตูตู้

พัดลมภายในตู้ฟักไข่และตู้เกิดลูกไก่ จะต้องทำงานเสมอ เพื่อกระจายความร้อนซึ่งเกิดจากขดลวดความร้อน และความชื้นซึ่งเกิดจากการพ่นละอองน้ำผ่านหัวพ่นไอน้ำให้ทั่วตู้ฟักไข่และตู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกิดลูกไก่ และช่วยหมุนเวียนอากาศภายในตู้ฟักไข่และตู้เกิดลูกไก่ ซึ่งเป็นสิ่งจำเป็นมาก ถ้าพัดลมไม่ทำงาน ทำให้ออกซิเจนมีไม่เพียงพอและเป็นอันตรายสำหรับไข่ที่ทำการฟักหรือลูกไก่ที่กำลังจะโต ระบบควบคุมต้องตรวจสภาวะการทำงานของพัดลมอยู่เรื่อย ๆ และแจ้งเตือนให้ทราบเมื่อพัดลมไม่ทำงาน

3.4 การแสดงผล

แสดงผลการทำงานของอุปกรณ์ที่ถูกควบคุมโดยระบบควบคุม ด้วยแอลอีดี (LED)

1) การทำงานของขดลวดความร้อน แสดงผลว่าทำงาน ไม่ทำงาน

2) การทำงานวาล์วท่อน้ำ แสดงผลว่าเปิดหรือปิด

3) การพลิกถาดไข่ไปทางขวา แสดงผลว่าถาดไข่อยู่ทางขวาหรือไม่อยู่

4) การพลิกถาดไข่ไปทางซ้าย แสดงผลว่าถาดไข่อยู่ทางซ้ายหรือไม่อยู่

แสดงการแจ้งเตือนเมื่อสภาพภายในตู้ฟักไข่หรือตู้เกิดลูกไก่ เป็นอันตรายต่อการใช้งาน ด้วยแอลอีดี

1) อุณหภูมิภายในตู้ฟักไข่หรือตู้เกิดลูกไก่ต่ำกว่าอุณหภูมิปลอดภัยต่ำสุด แสดงผลว่าต่ำกว่าหรือไม่

2) อุณหภูมิภายในตู้ฟักไข่หรือตู้เกิดลูกไก่สูงกว่าอุณหภูมิปลอดภัยสูงสุด แสดงผลว่าสูงกว่าหรือไม่

3) ความชื้นภายในตู้ฟักไข่หรือตู้เกิดลูกไก่ต่ำกว่าความชื้นปลอดภัยต่ำสุด แสดงผลว่าต่ำกว่าหรือไม่

4) ความชื้นภายในตู้ฟักไข่หรือตู้เกิดลูกไก่สูงกว่าความชื้นปลอดภัยสูงสุด แสดงผลว่าสูงกว่าหรือไม่

5) สถานะภาพของประตูตู้ฟักไข่หรือตู้เกิดลูกไก่ แสดงผลว่าประตูเปิดหรือปิด

6) สถานะภาพของพัดลมภายในตู้ฟักไข่หรือตู้เกิดลูกไก่ แสดงผลว่าพัดลมทำงานหรือไม่ทำงาน

แสดงค่าที่วัดได้ทางจอภาพ (Monitor) ของเครื่องคอมพิวเตอร์

1) ค่าอุณหภูมิภายในตู้ฟักไข่และตู้เกิดลูกไก่ เป็นองศาฟาเรนไฮต์

2) ค่าความชื้นสัมพัทธ์ในตู้ฟักไข่และตู้เกิดลูกไก่ เป็นเปอร์เซนต์

3) สถานะของถาดไข่ในตู้ฟักไข่ เป็นซ้ายหรือขวา

3.5 การใช้แอดเดรสพอร์ทในฮาร์ดแวร์เพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1 การใช้แอดเดรสอินพุท-เอาท์พุทพอร์ทของการ์ดเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล

คำสั่ง OUT มายังแอดเดรสพอร์ทเหล่านี้ในโปรแกรม เริ่มให้ทำการเปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

คำสั่ง IN ของแอดเดรสพอร์ทเหล่านี้ในโปรแกรม เป็นการอ่านค่าดิจิตอลที่ได้จากการเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอลด้วยคำสั่ง OUT

แอดเดรสอินพุท-เอาท์พุทพอร์ท (ฐาน 16)	สัญญาณอินพุท
02F3	อุณหภูมิดรายบัลย์ของตู้ฟักไข่หมายเลข 0
12F3	อุณหภูมิเวทบัลย์ของตู้ฟักไข่หมายเลข 0
22F3	อุณหภูมิดรายบัลย์ของตู้เกิดลูกไก่หมายเลข 1
32F3	อุณหภูมิเวทบัลย์ของตู้เกิดลูกไก่หมายเลข 1
42F3	อุณหภูมิดรายบัลย์ของตู้ฟักไข่หมายเลข 2
52F3	อุณหภูมิเวทบัลย์ของตู้ฟักไข่หมายเลข 2
62F3	อุณหภูมิดรายบัลย์ของตู้เกิดลูกไก่หมายเลข 3
72F3	อุณหภูมิเวทบัลย์ของตู้เกิดลูกไก่หมายเลข 3

ตาราง 3.1 การใช้แอดเดรสอินพุท-เอาท์พุทพอร์ทของการ์ดเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล

แอดเดรสพอร์ทข้างบนเป็นแอดเดรสพอร์ทที่ไม่มีการใช้งานจากการ์ดอื่น ๆ ที่ใช้กับเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที

3.5.2 การใช้แอดเดรสอินพุท-เอาท์พุทพอร์ทของการ์ดอินพุท-เอาท์พุทพอร์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอดเดรส อินพุท-เอาท์พุท พอร์ท (ฐาน 16)	คำสั่ง	บิทที่ 0 - 3	บิทที่ 4 - 7
82F3	OUT	ควบคุมวาล์วท่อลมสำหรับพลิกถาดไข่ไปทางขวาและแสดงสถานะการทำงาน	ควบคุมวาล์วท่อลมสำหรับพลิกถาดไข่ไปทางซ้ายและแสดงสถานะการทำงาน
92F3	OUT	ควบคุมขดลวดความร้อนและแสดงสถานะการทำงาน	ควบคุมวาล์ท่อน้ำและแสดงสถานะการทำงาน
A2F3	OUT	แจ้งเตือนอุณหภูมิต่ำ	แจ้งเตือนอุณหภูมิสูง
B2F3	OUT	แจ้งเตือนความชื้นต่ำ	แจ้งเตือนความชื้นสูง
C2F3	OUT	แจ้งเตือนประตูเปิด	แจ้งเตือนพัดลมไม่ทำงาน
D2F3	OUT	รับอินพุทสถานะของประตู	รับอินพุทสถานะของพัดลม

ตาราง 3.2 การใช้แอดเดรสอินพุท-เอาท์พุทพอร์ท

หมายเหตุ บิท 0 กับ 4 เป็นของตู้ฟักไข่หมายเลข 0

บิท 1 กับ 5 เป็นของตู้เกิดลูกไก่หมายเลข 1

บิท 2 กับ 6 เป็นของตู้ฟักไข่หมายเลข 2

บิท 3 กับ 7 เป็นของตู้เกิดลูกไก่หมายเลข 3

3.6 โครงสร้างฮาร์ดแวร์สร้างเพิ่มเติม

ฮาร์ดแวร์สร้างเพิ่มเติม สามารถแบ่งออกเป็นส่วน ๆ ได้ดังนี้

1) การขยายบัสมายังภายนอก

2) สร้างการ์ดสำหรับเปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิตอลใช้กับบัสภายนอก

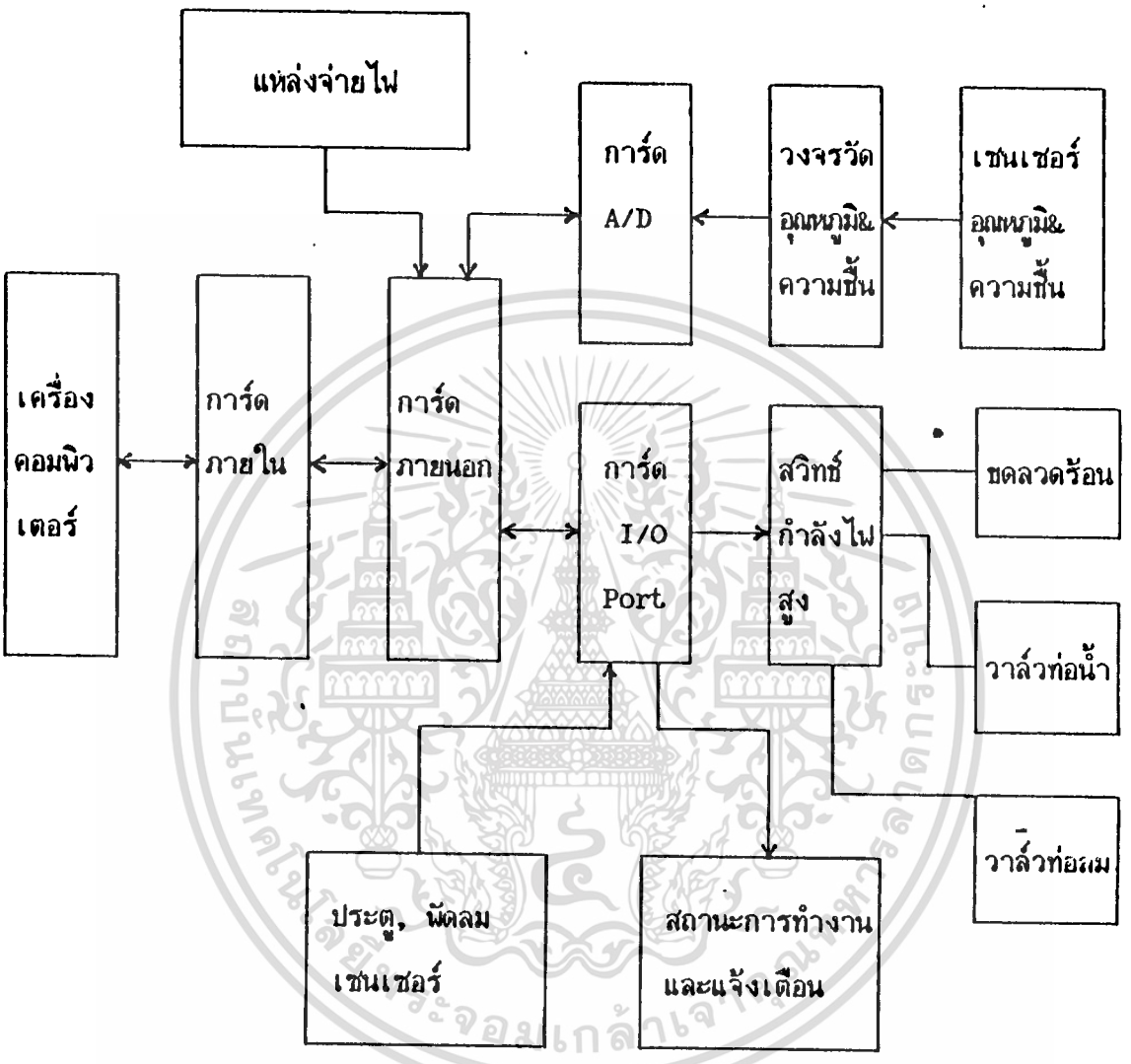
3) สร้างการ์ดสำหรับเป็นอินพุท-เอาท์พุทพอร์ทใช้กับบัสภายนอก

4) สร้างวงจรวัดอุณหภูมิและความชื้น

5) สร้างวงจรปิดเปิดไฟกำลังสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) สร้างแหล่งจ่ายไฟภายนอก



รูป 3.2 โครงสร้างฮาร์ดแวร์เพิ่มเติม

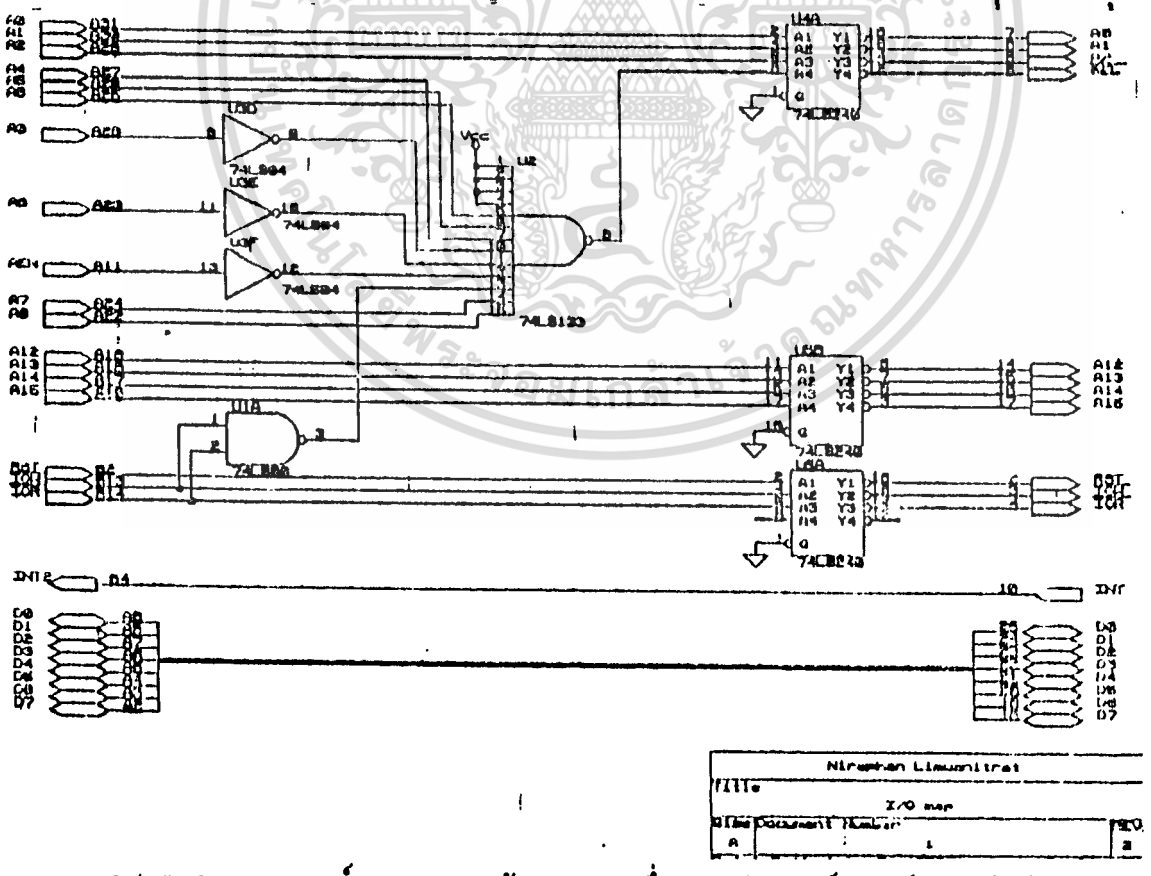
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การขยายบัสมายังภายนอก

เพื่อขยายบัสของระบบมายังภายนอก ซึ่งมีเนื้อที่ว่างมากกว่า และออกแบบให้ใช้สายระหว่างการ์ดภายในเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ทีกับการ์ดภายนอกได้ยาวประมาณ 3 ฟุต

3.7.1 การออกแบบการ์ดขยายบัสของระบบภายในเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที

จากการกำหนดการใช้แอดเดรสอินพุท-เอาท์พุทพอร์ทในหัวข้อ 3.5 ทำให้สามารถกำหนดสัญญาณบนสลอตเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ทีที่จะใช้ได้ดังนี้คือ แอดเดรสบิทที่ 0-9 และ 12-16 บัสข้อมูล, สัญญาณควบคุม IOR, IOW, AEN, สัญญาณ RTS, END และ INT2 หน้าที่ของการ์ดนี้ก็คือเพิ่มกำลังขับของสัญญาณเหล่านี้ก่อนส่งสัญญาณผ่านสายไปยังการ์ดภายนอกและรับสัญญาณผ่านสายจากการ์ดภายนอกมาเพิ่มกำลังขับก่อนส่งสัญญาณเข้าบัสของระบบ รูป 3.3 แสดงวงจรการ์ดขยายระบบบัสภายในเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที ซึ่งใช้ 74LS240 เป็นตัวเพิ่มกำลังขับของสัญญาณและรับสัญญาณผ่านสาย ปกติจะทำให้เกิดการล่าช้าของสัญญาณที่ถูกขับประมาณ 12 นาโนวินาที และช้าที่สุดไม่เกิน 18 นาโนวินาที ดังนั้นบัสไซเคิลจากการขยายระบบบัสจะเกิดการล่าช้า เวลาการเข้าถึงข้อมูล (Access Data) ของแอดเดรสอินพุท-เอาท์พุท



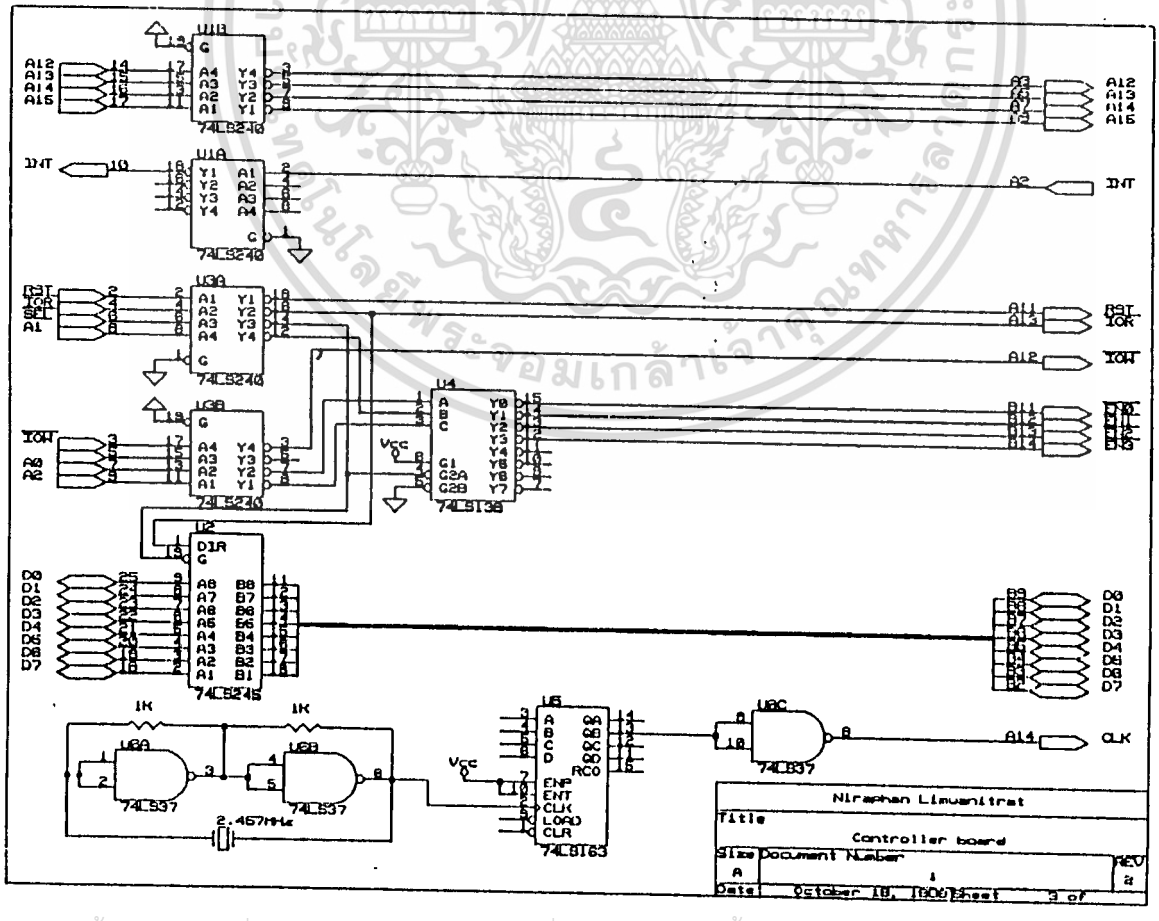
รูป 3.3 วงจรการ์ดขยายระบบบัสภายในเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที

พอร์ทจะน้อยลง แต่เนื่องจากการล่าช้ายังน้อย ดังนั้นจึงไม่มีผลกับการทำงาน สัญญาณจาก

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สลอตที่ถูกขับไปยังการ์ดสลอตภายนอกโดยตรง คือ สัญญาณแอดเดรสบิท 0 ถึง 2 และ 12 ถึง 15 สัญญาณ RTS, IOR และ IOW ส่วนสัญญาณแอดเดรสบิทที่ 4 ถึง 9, AEN, IOR และ IOW จะถูกนำมากอดสัญญาณ เป็นสัญญาณ SEL ก่อนส่งไปยังการ์ดสลอตนอก ซึ่งสัญญาณนี้แอคทีฟเมื่อมีการใช้คำสั่ง IN หรือ OUT มายังแอดเดรสอินพุท-เอาท์พุทพอร์ทที่แอดเดรส 10 บิทล่างอยู่ในช่วง 2F0-2F7 สัญญาณรับจากการ์ดภายนอกและเพิ่มกำลังขับก่อนส่งเข้าระบบบัสคือสัญญาณ INT2

สำหรับบัสข้อมูลจะไม่มีการเพิ่มกำลังขับก่อนส่งสัญญาณเข้าไปในสาย เนื่องจากบัสข้อมูลของระบบสามารถขับสัญญาณได้พอเพียง คงมีเพียงการต่อความต้านทานอนุกรมกับตัวเก็บประจุข้างความต้านทานต่อกับแหล่งจ่ายไฟ +5 โวลท์ และข้างตัวเก็บประจุต่อกับบัสข้อมูล ทั้งนี้เพื่อลดผลที่เกิดขึ้นเนื่องจากตัวเก็บประจุของสาย ซึ่งทำให้สัญญาณล่าช้าและเพี้ยน โดยเมื่อเกิดการเปลี่ยนแปลงของสัญญาณจากสูงเป็นต่ำ (High To Low) ตัวเก็บประจุจะลัดวงจรทำให้ความต้านทานต่อกับบัสข้อมูลอิมพีแดนซ์ (Impedance) มีค่าลดลง ประจุของระบบจะหมดไปกับส่วนของวงจรที่มีอิมพีแดนซ์ต่ำ ทำให้ผลของอันเดอร์ชูท (Undershoot) ลดลง



รูป 3.4 วงจรการ์ดขยายบัสของระบบภายนอกเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.2 การออกแบบการ์ดขยายบัสของระบบภายนอกเครื่องคอมพิวเตอร์ไอบีเอ็มพีซี

การ์ดภายนอกรับสัญญาณมาจากแอดเดรสบิทที่ 0-2, SEL, RST, IOR, IOW และ GND จากการ์ดภายใน ส่งสัญญาณ INT ไปยังการ์ดภายใน หน้าที่ของการ์ดภายนอกคือ เพิ่มกำลังให้สัญญาณที่รับเข้ามาและส่งไปให้การ์ดภายใน รูปที่ 3.4 แสดงวงจรการ์ดขยายบัสของระบบภายนอกเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที ซึ่งก็ใช้ 74LS240 เพิ่มกำลังและขับสัญญาณไปยังสลอต ซึ่งทำให้สัญญาณเกิดการล่าช้าเหมือนกับการ์ดขยายบัสของระบบภายใน สัญญาณแอดเดรสบิทที่ 12-15, RST, IOR และ IOW ถูกนำมาเพิ่มกำลังและขับไปยังสลอตโดยตรง ส่วนสัญญาณแอดเดรสบิทที่ 0-2 และสัญญาณ SEL ถูกนำมาถอดสัญญาณด้วย 74LS138 เพื่อให้ได้สัญญาณ EN0 ถึง EN7 หรืออีกอย่างก็คือ เพื่อเรียกช่วงใดช่วงหนึ่งของช่วงแอดเดรสอินพุท-เอาท์พุทของพอร์ท ซึ่งมีแอดเดรส 10 บิทล่างเป็น 2F0-2F7 (EN0 เท่ากับ แอดเดรส 10 บิทล่างเป็น 2F0 เป็นต้น) แต่ EN0-EN3 เท่านั้นที่จะไปยังสลอต มีการสร้างสัญญาณ CLK ไปยังสลอตขึ้นมาบนการ์ดสลอตภายนอกเอง เป็นสัญญาณนาฬิกาความถี่ประมาณ 640 กิโลเฮิร์ท ได้มาจากวงจรผลิตความถี่ซึ่งใช้คริสตัล (Crystal) ให้ความถี่ 2.457 เมกกะเฮิร์ท นำมาหารด้วยวงจรหาร 74LS163 สัญญาณที่รับเข้ามาเพิ่มกำลังและส่งไปยังการ์ดภายในคือสัญญาณ INT จากสลอตบนการ์ดภายนอกนี้ใช้ 74LS125 เป็นตัวเพิ่มกำลังและขับสัญญาณบัสข้อมูลระหว่างการ์ดภายในและภายนอก รวมทั้งควบคุมให้สัญญาณบัสข้อมูลสามารถผ่านไปยังการ์ดภายใน หรือจากบัสข้อมูลของการ์ดภายในมายังการ์ดภายนอก เฉพาะเมื่อมีการใช้ช่วงแอดเดรสอินพุท-เอาท์พุทพอร์ท 2F0-2F7 เท่านั้น รูป 3.5 แสดงสัญญาณบนสลอต 30 ขาของการ์ดขยายบัสของระบบภายนอกเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที

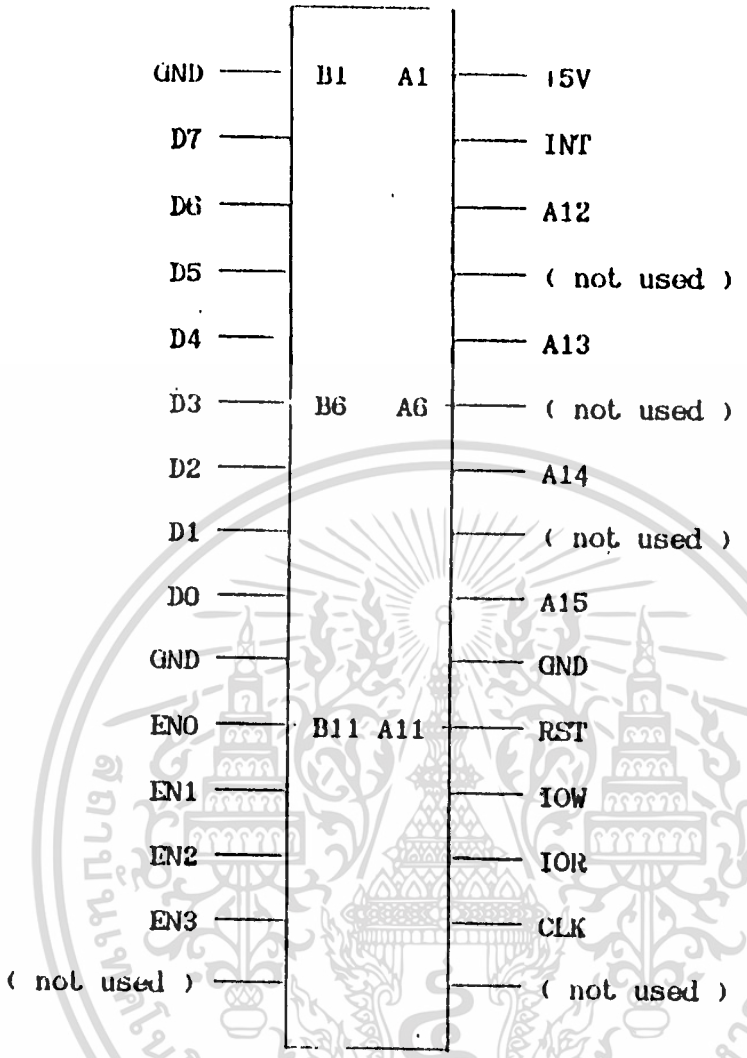
3.8 การออกแบบการ์ดเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล ใช้กับการ์ดขยายบัสของระบบภายนอก

รูป 3.6 แสดงให้เห็นบลอคไดอะแกรมของการ์ดเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล 4 ส่วนด้วยกันคือ

1) บลอคไดอะแกรมวงจรถอดสัญญาณ แอดเดรสบิทที่ 12 ถึง 15, CLK, IOW และ EN3 เพื่อสร้างเป็นสัญญาณ STA และสัญญาณ INT EN โดยที่สัญญาณ STA เป็นสัญญาณสั่งให้เริ่มทำการเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล สัญญาณ INT EN เป็นสัญญาณควบคุมให้สัญญาณ INT สั่งไปยังสลอตบนการ์ดสลอตได้

2) บลอคไดอะแกรมวงจรถอดสัญญาณ แอดเดรสบิทที่ 12 ถึง 15, IOR และ EN3 เพื่อสร้างเป็นสัญญาณ O/P EN และสัญญาณ INT DIS สัญญาณ O/P EN เป็นสัญญาณใช้สำหรับอ่านข้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



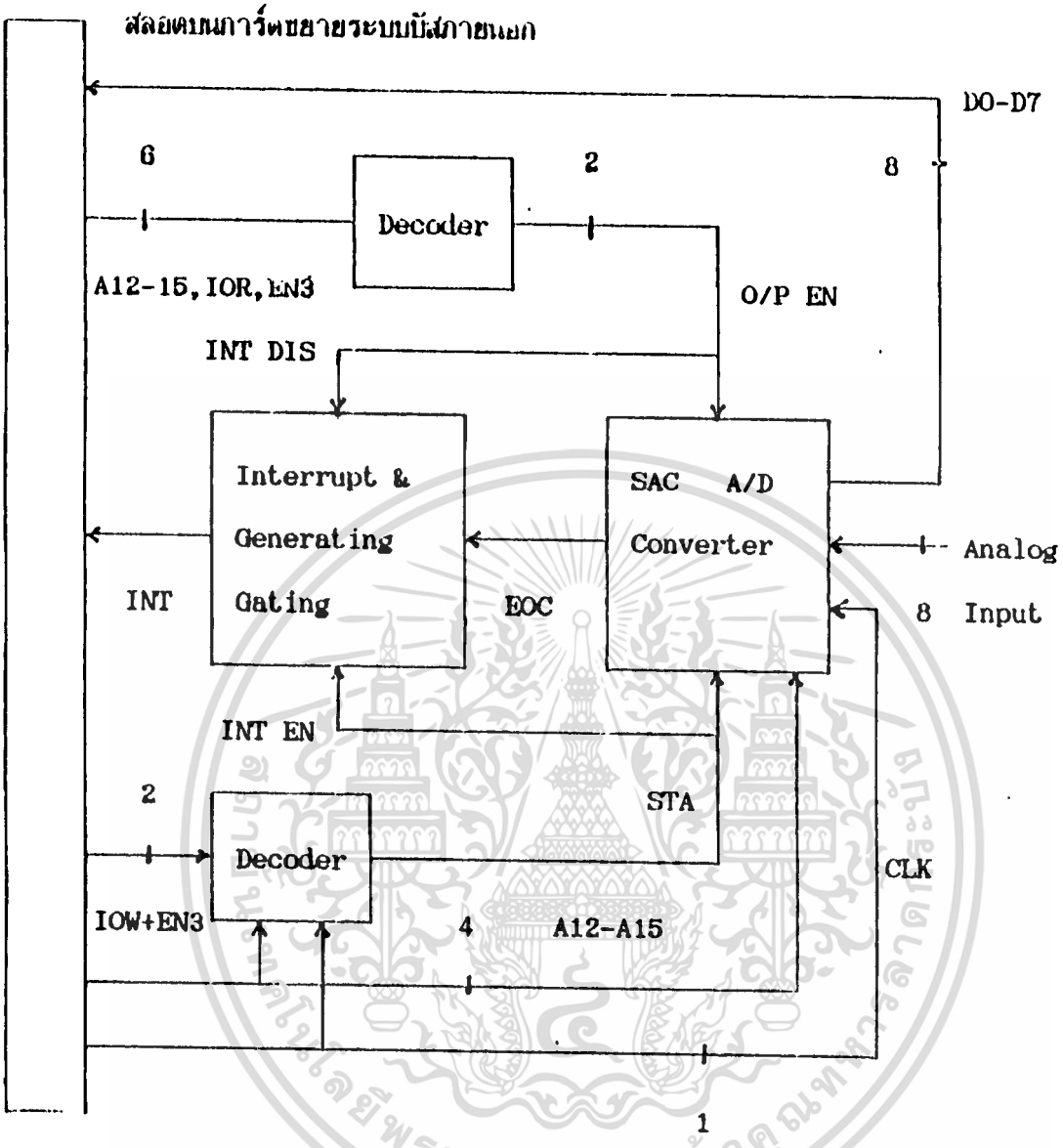
รูป 3.5 สัญญาณบนสล็อต 30 ขาบนการ์ดขยายบัสของระบบภายนอกเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ที

มูลดิจิตอลที่ได้จากการเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล สัญญาณ INT DIS เป็นสัญญาณควบคุมไม่ให้สัญญาณ INT ส่งไปยังสล็อตบนการ์ดขยายบัสของระบบภายนอก

3) บล็อคไดอะแกรมวงจรเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอลแบบซัคเซสซีฟแอพพลอคซิเมชั่น สัญญาณ STA แอคทีฟ ทำให้วงจรเริ่มเปลี่ยนสัญญาณอนาลอกอินพุท ซึ่งต่อกับช่องที่กำหนดโดยแอดเดรสบิทที่ 12 ถึง 15 ใช้สัญญาณ CLK เป็นสัญญาณนาฬิกาสำหรับวงจร เมื่อจบการเปลี่ยนสัญญาณก็จะสร้างสัญญาณ EOC ซึ่งเป็นสัญญาณบ่งว่าจบการเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล และเมื่อสัญญาณ O/P EN แอคทีฟ ก็จะขับข้อมูลที่ได้จากการเปลี่ยนสัญญาณครั้งสุดท้ายลงบนบัสข้อมูลบนสล็อตของการ์ดขยายบัสของระบบภายนอก

4) บล็อคไดอะแกรมวงจรสร้างและควบคุมสัญญาณอินเทอรัพท์ สัญญาณ EOC และ INT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



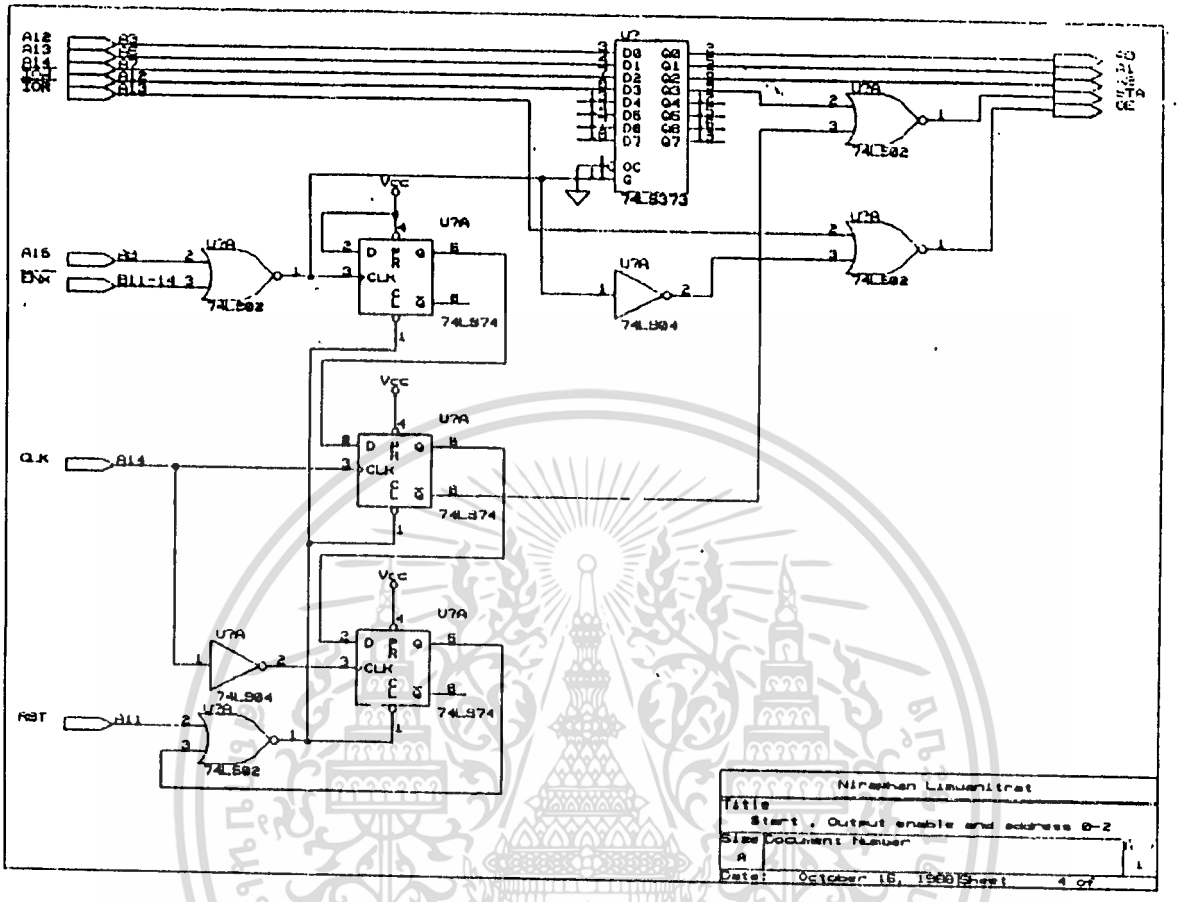
รูป 3.6 โครงสร้างการ์ดเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล

EN แอคทีฟ ก็จะสร้างสัญญาณ INT ซึ่งเป็นสัญญาณอินเทอรัพท์ไปยังสลอตบนการ์ดขยายบัสของระบบภายนอก สัญญาณ INT DIS แอคทีฟ จะปล่อยขา INT ลอย (Float) จากสลอตบนการ์ดขยายบัสของระบบภายนอก

รูป 3.7 แสดงวงจรบลอคไดอะแกรม 1 และ 2 ซึ่งสร้างสัญญาณ STA และ OE (เป็นสัญญาณเดียวกับ O/P EN) แอคทีฟไฮ สัญญาณแอดเดรสบิทที่ 15 แอคทีฟโลว์ นำไปผ่านนอร์เกท (NOR Gate) 74LS02 กับสัญญาณ EN3 จะได้สัญญาณแอคทีฟไฮ นำไปผ่านอินเวอเตอร์ก่อนนำไปผ่านนอร์เกทกับสัญญาณ IOR จะได้สัญญาณ OE แอคทีฟไฮ นำไปควบคุมให้ 74LS374 แลทช์ค่าแอดเดรสบิทที่ 12-14 และสัญญาณ IOW และนำไปเป็นฟลิบฟลอบ (Flip Flop) 74LS74 ซึ่งจะให้เอาท์พุทเป็นสัญญาณ CLK 1 ลูกเป็นสัญญาณแอคทีฟโลว์ นำไปผ่านนอร์

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในเชิงพาณิชย์ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกตกับสัญญาณ IOW จะได้สัญญาณ STA แอดทีพไฮ ที่ซิงโครนัส (Synchronous) กับสัญญาณ CLK



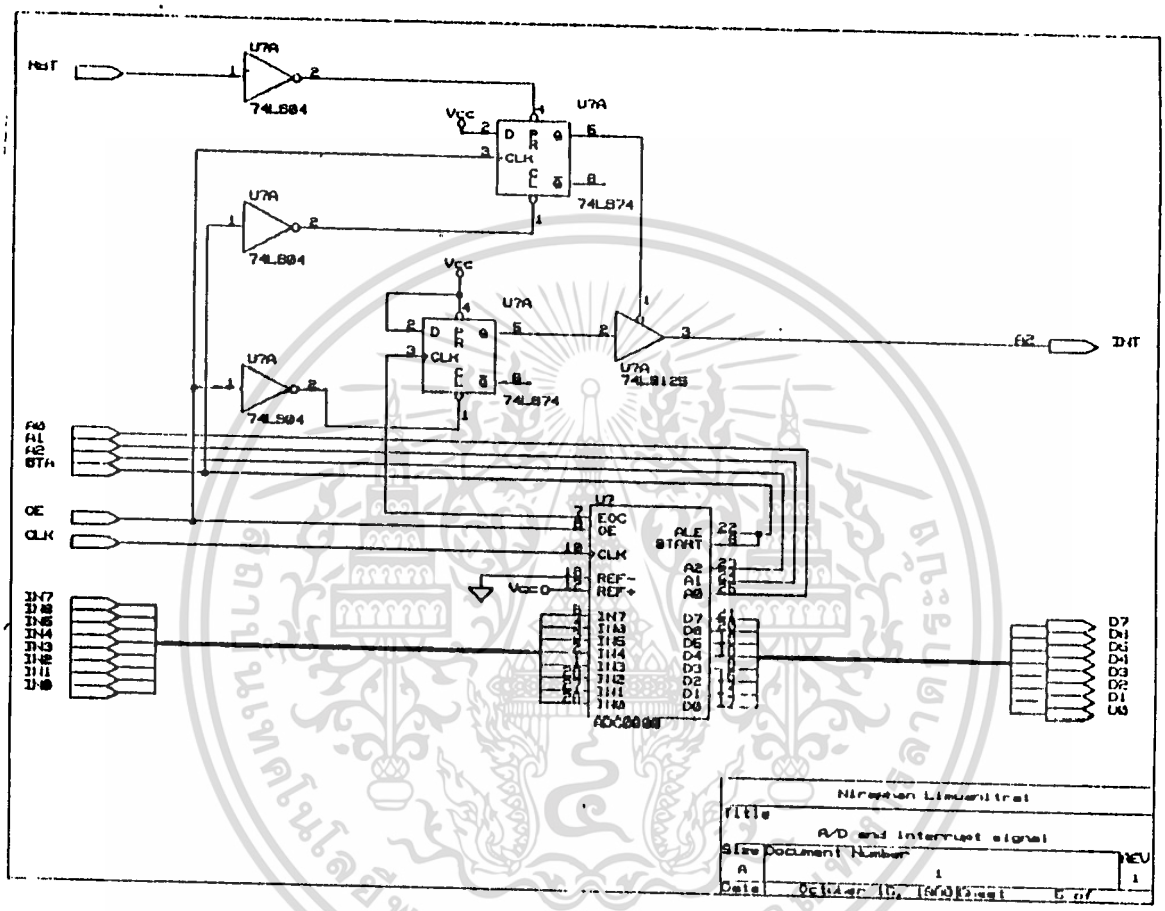
รูป 3.7 วงจรของบลอคไดอะแกรม 1 และ 2

รูป 3.8 แสดงวงจรของบลอคไดอะแกรม 3 และ 4 ใช้ตัวเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอลแบบซัคเซสซีฟแอปพลอคซิเมชั่น เบอร์ ADC 0809 ซึ่งสามารถรับสัญญาณอนาลอกอินพุทได้ 8 ช่อง ระดับสัญญาณอนาลอกอินพุทอยู่ในช่วง 0-5 โวลท์ ทำการเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอลด้วยวิธีการแบบซัคเซสซีฟแอปพลอคซิเมชั่นกินเวลาประมาณ 0.1 มิลิวินาที (Millisecond) ใช้สัญญาณ CLK ต้องการขา CLK เพื่อเป็นสัญญาณนาฬิกาสำหรับการทำงาน ให้สัญญาณดิจิตอลขนาด 8 บิท ใช้สัญญาณแอดเดรสบิทที่ 12-14 ต่อกับขา A0-A2 ในการเลือกช่องสัญญาณอนาลอกอินพุทที่จะทำการเปลี่ยนให้เป็นสัญญาณดิจิตอล สัญญาณ STA มาเข้าขา ALE และ START เพื่อสั่งให้ตัวคอนเวอเตอร์เริ่มทำงาน สัญญาณ OE ต่อกับขา OE เพื่อดึงข้อมูลดิจิตอลที่ได้จากการทำงานออกมาวางบนบัสข้อมูล บัสข้อมูลต่อกับขาก D0-D7 เพื่อใช้ในการส่งข้อมูล

สัญญาณ STA แอดทีพไฮ ทำให้เริ่มเปลี่ยนสัญญาณอนาลอกอินพุทซึ่งเลือกด้วยแอดเดรสบิทที่ 12-14 และ 74LS125 ยอมให้สัญญาณ INT ผ่านไปยังสลอตได้ เมื่อจบการทำงาน สัญญาณ EOC จะแอดทีพไฮ ผ่านไปยังวงจรสร้างสัญญาณ INT ซึ่งเป็นฟลิบฟลอบ 74LS74 สร้างสัญญาณอินเทอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับท์ไปยังสลอต และเมื่อสัญญาณ OE แอคทีฟไฮ ทำให้ข้อมูลดิจิตอลที่ได้จากการทำงานมาอยู่บนบัสข้อมูล 74LS125 ตัดสัญญาณ INT ออกจากสลอต และทำให้วงจรสร้างสัญญาณ INT สร้างสัญญาณ INT ไลว์



รูป 3.8 วงจรของบลอคไดอะแกรม 3 และ 4

3.9 การออกแบบการ์ดอินพุท-เอาท์พุทพอร์ทใช้กับการ์ดขยายบัสของระบบภายนอก

รูป 3.9 แสดงให้เห็นบลอคไดอะแกรมของการ์ดอินพุท-เอาท์พุทพอร์ท ซึ่งมี 4 ส่วน คือ

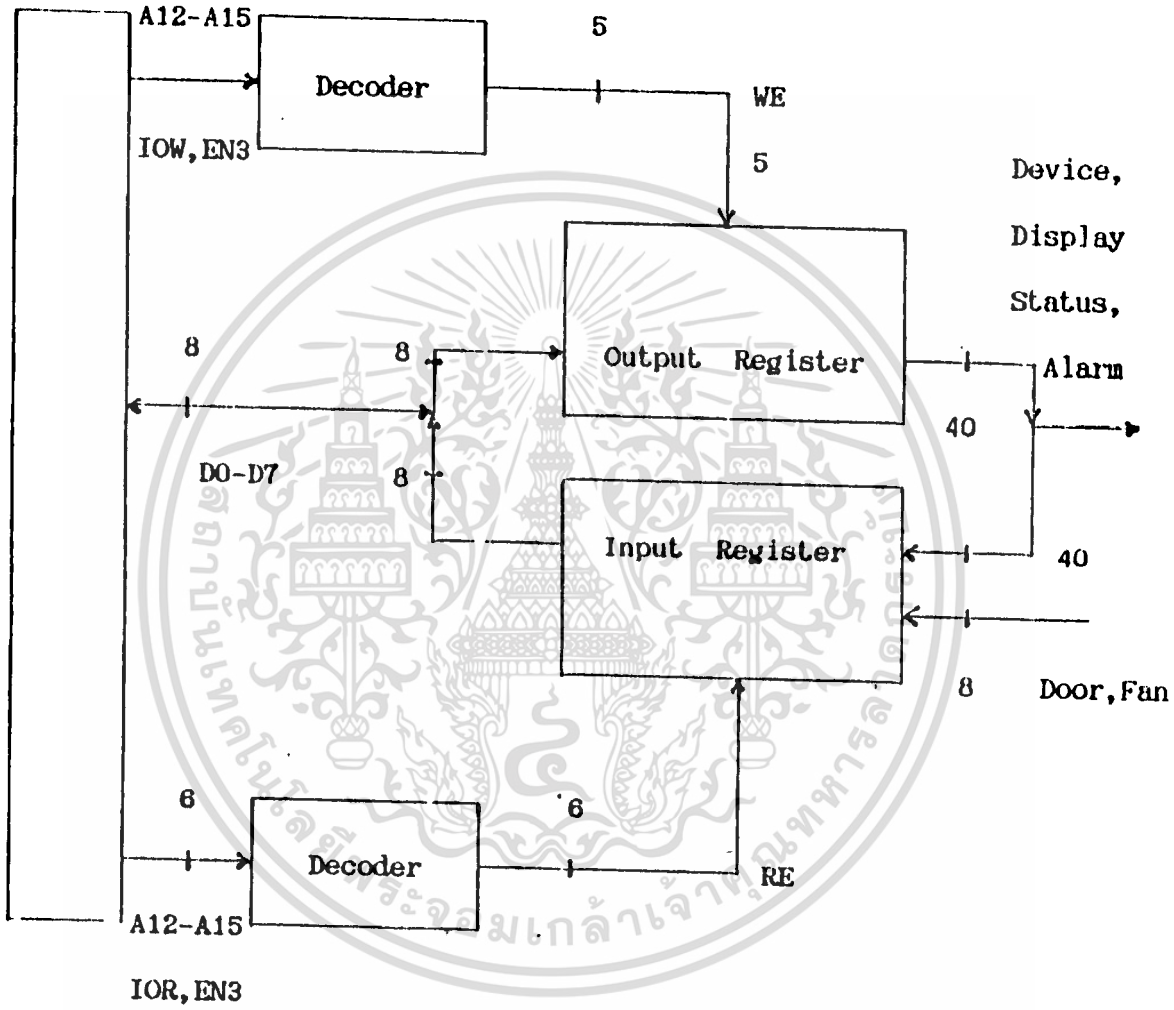
1) บลอคไดอะแกรมวงจรถอดสัญญาณ แอดเดรสบิทที่ 12-15, IOW และ EN3 เพื่อสร้างสัญญาณควบคุมการเขียน WE เอาท์พุทพอร์ท ที่แอดเดรสกำหนดไว้ในหัวข้อ 3.5

2) บลอคไดอะแกรมวงจรถอดสัญญาณ แอดเดรสบิทที่ 12-15, IOR และ EN3 เพื่อสร้างสัญญาณควบคุมการอ่าน RE อินพุทพอร์ท ที่แอดเดรสกำหนดไว้ในหัวข้อ 3.5

3) บลอคไดอะแกรมเอาท์พุทรีจิสเตอร์ ซึ่งรับสัญญาณควบคุมการเขียน WE และบัสข้อมูลเพื่อส่งค่าข้อมูลไปควบคุมการทำงานของอุปกรณ์ แสดงสถานะการทำงานและแจ้งเตือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

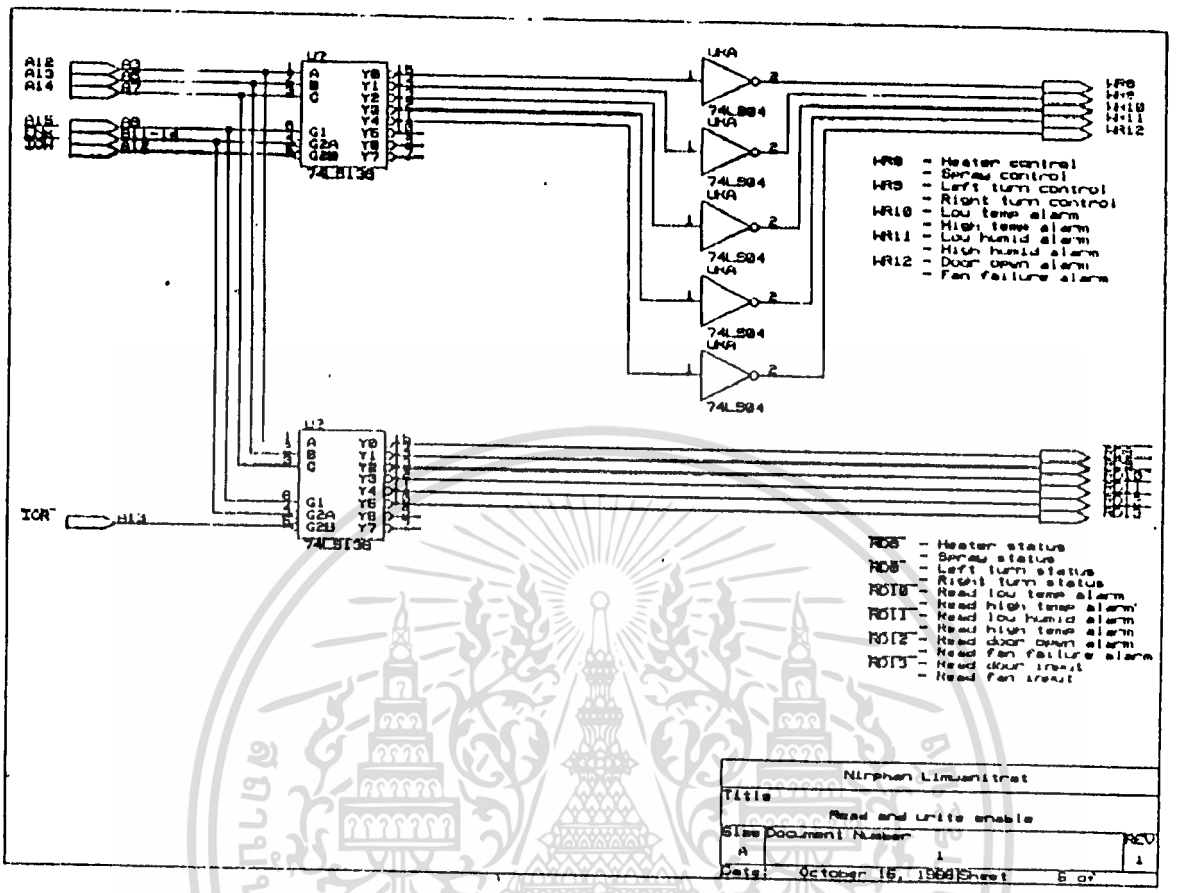
4) แผงไดอะแกรมอินพุทรีจิสเตอร์ ซึ่งรับสัญญาณควบคุมการอ่าน RE และบัสข้อมูลจากเอาท์พุทรีจิสเตอร์และตัวตรวจจับ เพื่อรับข้อมูลมาวางไว้บนบัสข้อมูลของการ์ดขยายบัสของระบบภายนอก



รูป 3.9 โครงสร้างการ์ดอินพุท-เอาท์พุทพอร์ท

รูป 3.10 แสดงวงจรของบลอคไดอะแกรม 1 และ 2 ทำหน้าที่สร้างสัญญาณควบคุมการเขียน WE แอดเดรสเอาท์พุทรีจิสเตอร์พอร์ท (ฐาน 16) 82F3, 92F3, A2F3, B2F3 และ C2F3 และแสดงสัญญาณควบคุมการอ่าน RE แอดเดรสอินพุทรีจิสเตอร์พอร์ท (ฐาน 16) 82F3, 92F3, A2F3, B2F3, C2F3 และ D2F3 สัญญาณควบคุมการเขียนแอคทีฟไฮ และสัญญาณควบคุมการอ่านแอคทีฟโลว์ โดยใช้ 74LS138 เป็นตัวถอดสัญญาณ สำหรับสัญญาณควบคุมการอ่านจะได้สัญญาณแอคทีฟโลว์ และใช้ 74LS138 เป็นตัวถอดสัญญาณสัญญาณควบคุมการเขียนแล้วนำไปผ่านอินเวอเตอร์ 74LS04 เพื่อให้สัญญาณแอคทีฟไฮ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.10 วงจรของบลอคไดอะแกรม 1 และ 2

รูป 3.11 แสดงวงจรของบลอคไดอะแกรม 3 และ 4 เอาท์พุทรีจิสเตอร์เป็น74LS378 และอินพุทรีจิสเตอร์เป็น 74LS245 สัญญาณควบคุมการเขียน WE ของเอาท์พุทรีจิสเตอร์ 74LS373 แอคทีฟไฮ ทำให้เอาท์พุทรีจิสเตอร์รับข้อมูลบนบัสข้อมูล มาไว้ในตัวมันเองและคงค่าไว้จนกว่าจะมีสัญญาณควบคุมการเขียนครั้งใหม่ และสัญญาณควบคุมการอ่าน RE ของอินพุทรีจิสเตอร์แอคทีฟโลว์ ทำให้อินพุทรีจิสเตอร์ 74LS245 รับข้อมูลมาส่งลงยังบัสข้อมูลบนสลอตของการ์ดขยายบัสของระบบภายนอก

3.10 การออกแบบวงจรวัดอุณหภูมิและความชื้น

ข้อกำหนดของวงจร

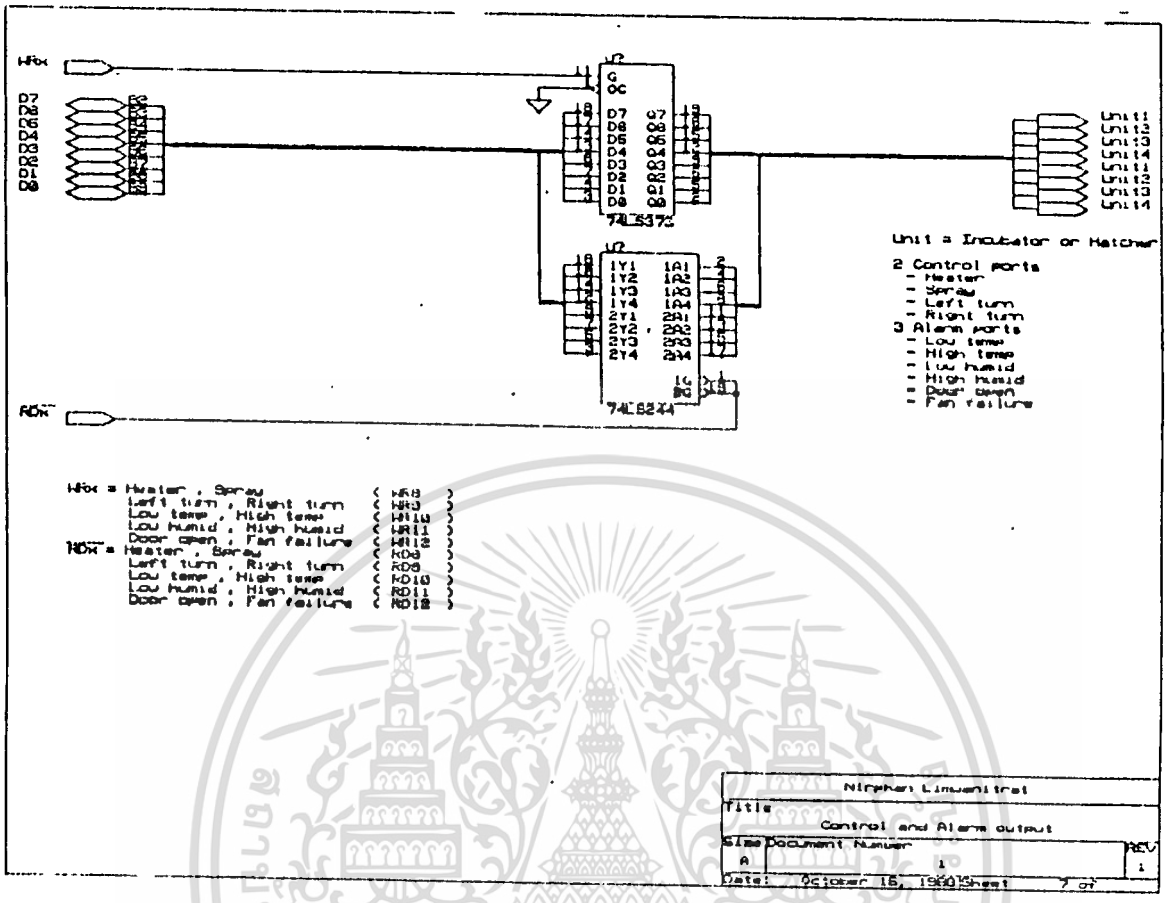
1) ใช้งานในช่วงอุณหภูมิ 25-40 องศาเซลเซียส

2) ให้เอาท์พุทเป็นสัญญาณอนาลอก 0-5 โวลท์

3) ไฟเลี้ยงวงจร 12 โวลท์และ -12 โวลท์

4) ใช้ พีที-100 เป็นตัวตรวจจับอุณหภูมิ ซึ่งเป็นตัวตรวจจับชนิดความต้านทาน มีค่าความต้านทานแปรผันโดยตรงกับอุณหภูมิ โดยความต้านทานของ พีที-100 ที่อุณหภูมิต่าง ๆ แสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.11 วงจรของบลอคไดอะแกรม 3 และ 4

ไว้ในตาราง 3.3

อุณหภูมิ(°C)	0	10	20	30	40	50	60	70
ความต้านทาน (โอห์ม)	100.00	103.97	107.93	111.87	115.81	119.73	123.64	127.54

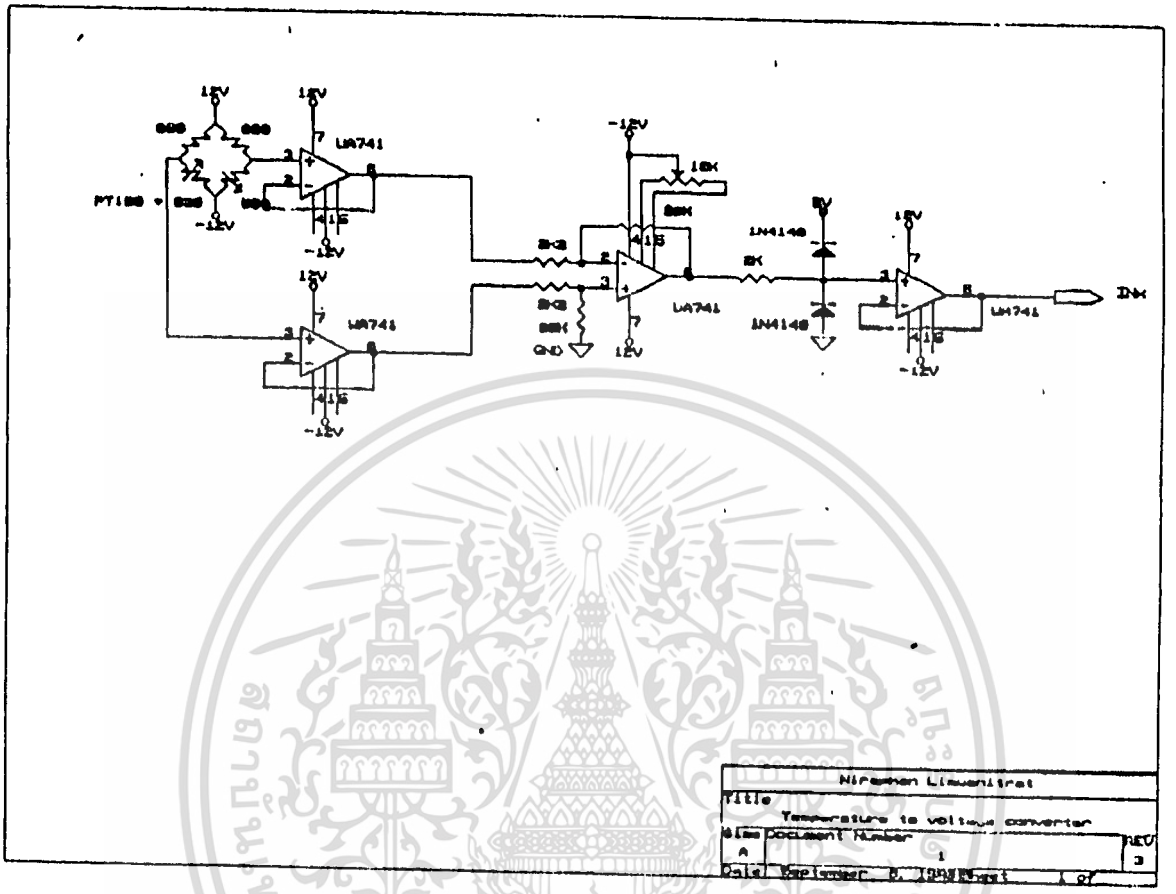
ตาราง 3.3 ความต้านทานของพีที-100 ที่อุณหภูมิต่าง ๆ

เนื่องจาก พีที-100 เป็นตัวตรวจจับแบบความต้านทาน จึงต้องให้กระแสไหลผ่านตัว พีที-100 เพื่อทำให้เกิดแรงดันตกคร่อม แรงดันตกคร่อมจะแปรผันตามอุณหภูมิด้วย แต่กระแสที่ไหลผ่านจะต้องไม่ทำให้เกิดความร้อนขึ้นที่ พีที-100 มากจนทำให้มีผลต่อการฯใช้งาน ซึ่งจะทำให้แรงดันมีค่าน้อยต้องนำไปขยายก่อนนำไปใช้งานต่อไป

ไพยการต่อ พีที-100 เป็นลักษณะบริดจ์ และนำไปขยายด้วยวงจรขยายความแตกต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้วงจรมีความยืดหยุ่นในการใช้งานจริงมากขึ้น



รูป 3.12 วงจรวัดอุณหภูมิและความชื้น

รูป 3.12 แสดงวงจรวัดอุณหภูมิและความชื้น โดยแขนข้างหนึ่งของวงจรบริดจ์เป็นความต้านทาน 680 โอห์มต่ออนุกรมกับ พีที-100 และความต้านทาน 330 โอห์ม และแขนอีกข้างต่อในลักษณะเดียวกันแต่เปลี่ยน พีที-100 เป็นความต้านทานปรับค่าได้ ซึ่งปรับให้มีค่าเท่ากับค่าความต้านทาน พีที-100 ที่อุณหภูมิ 25 องศาเซลเซียส แรงดันจากแขน พีที-100 จะเป็นแรงดันอินพุทบวกให้กับวงจรขยายความแตกต่าง ส่วนแรงดันของอีกแขนจะเป็นแรงดันอินพุทลบ โดยความแตกต่างของแรงดันอินพุทบวกและลบมีค่าเป็น 0 โวลท์ที่อุณหภูมิ 25 องศาเซลเซียส และ 0.136 โวลท์ที่ 40 องศาเซลเซียส ซึ่งต้องใช้วงจรขยายความแตกต่างซึ่งมีอัตราการขยายประมาณ 37 เท่าเพื่อให้แรงดันเป็น 5 โวลท์ที่ 40 องศาเซลเซียส แต่แรงดันอินพุทบวกและลบจะต้องผ่านวงจรบัฟเฟอร์ (Buffer) ทั้งนี้เพื่อไม่ให้อินพุทอิมพีแดนซ์ที่ไม่เท่ากันของสองอินพุทของวงจรขยายความแตกต่างมีผลกับการทำงานของวงจรบริดจ์ เนื่องจากวงจรขยายความแตกต่างให้แรงดันเอาท์พุทอยู่นอกช่วง 0-5 โวลท์ เมื่ออุณหภูมิใช้งานอยู่นอกช่วง 25-40 องศาเซลเซียส ดังนั้นจึงต้องนำแรงดันเอาท์พุทผ่านวงจรจำกัดแรงดันให้อยู่ในช่วง 0-5 โวลท์ก่อนส่งไปให้การ์ดเปลี่ยนสัญ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ญาณและเลยกเป็นดิจิตอลต่อไป

3.11 การออกแบบโปรแกรมควบคุมการทำงาน

3.11.1 โปรแกรมตอบสนองอินเทอรัพท์หมายเลข 1C (ฐาน 16)

เนื่องจากไม่จำเป็นต้องควบคุมอุปกรณ์ในการทำงานตลอดเวลา โปรแกรมให้การอินเทอรัพท์เป็นเวลาที่แน่นอนแก่เครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอกซ์ทีเข้าช่วยในการควบคุมการทำงานของอุปกรณ์ต่าง ๆ ตามกำหนดเวลา ซึ่งอินเทอรัพท์หมายเลข 1C (ฐาน 16) จะเกิดขึ้นประมาณ 18 ครั้งใน 1 วินาที

ในโปรแกรมจะกำหนดการทำงานให้กับอินเทอรัพท์ที่เกิดขึ้นทั้งหมดดังนี้

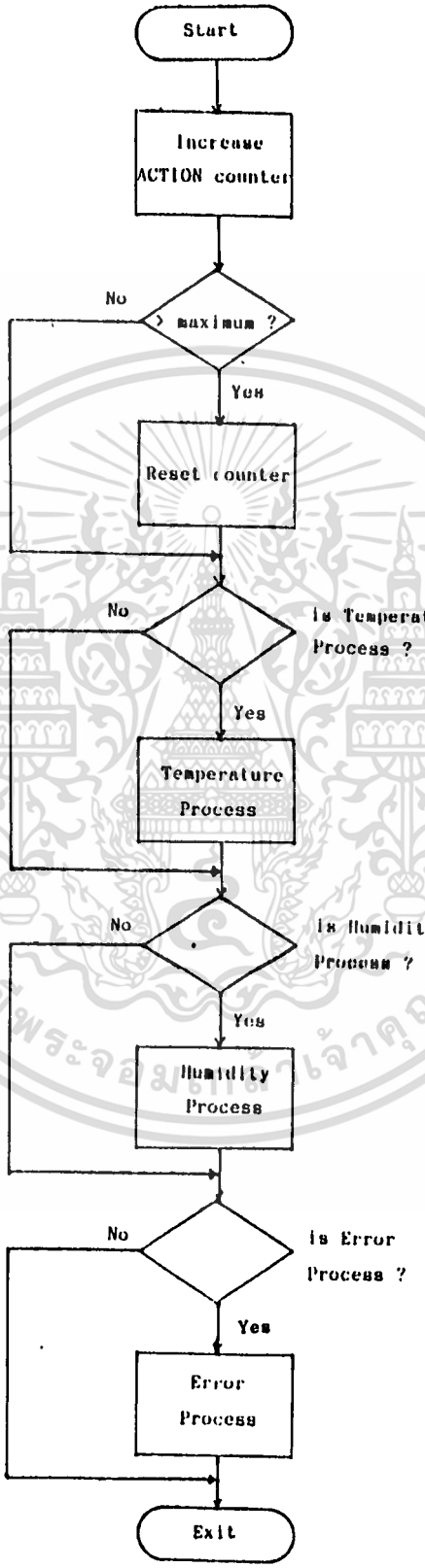
ครั้งที่	การทำงาน	ตู้ที่
1	อ่านอุณหภูมิดรายบัลย์ สถานะของประตูตู้และพัดลม	0
2	อ่านอุณหภูมิเวทบัลย์เมื่อครบเวลาหรือควบคุมวาล์วท่อน้ำ ควบคุมการพลิกถาดไข่เมื่อครบเวลา	0
3	อ่านอุณหภูมิดรายบัลย์ สถานะของประตูตู้และพัดลม	1
4	อ่านอุณหภูมิเวทบัลย์เมื่อครบเวลาหรือควบคุมวาล์วท่อน้ำ ควบคุมการพลิกถาดไข่เมื่อครบเวลา	1
5	อ่านอุณหภูมิดรายบัลย์ สถานะของประตูตู้และพัดลม	2
6	อ่านอุณหภูมิเวทบัลย์เมื่อครบเวลาหรือควบคุมวาล์วท่อน้ำ ควบคุมการพลิกถาดไข่เมื่อครบเวลา	2
7	อ่านอุณหภูมิดรายบัลย์ สถานะของประตูตู้และพัดลม	3
8	อ่านอุณหภูมิเวทบัลย์เมื่อครบเวลาหรือควบคุมวาล์วท่อน้ำ ควบคุมการพลิกถาดไข่เมื่อครบเวลา	3
9	ส่งเสียงเตือนที่เครื่องกำเนิดถ้าเกิดข้อผิดพลาด	0-3
10-18	ไม่มีการทำงาน	

ตาราง 3.4 การทำงานในแต่ละครั้งของอินเทอรัพท์หมายเลข 1C (ฐาน 16)

ซึ่งเมื่อทำงานครบ 18 ครั้ง โปรแกรมจะกลับไปทำยังครั้งที่ 1 ต่อไป โฟลชาร์ท (

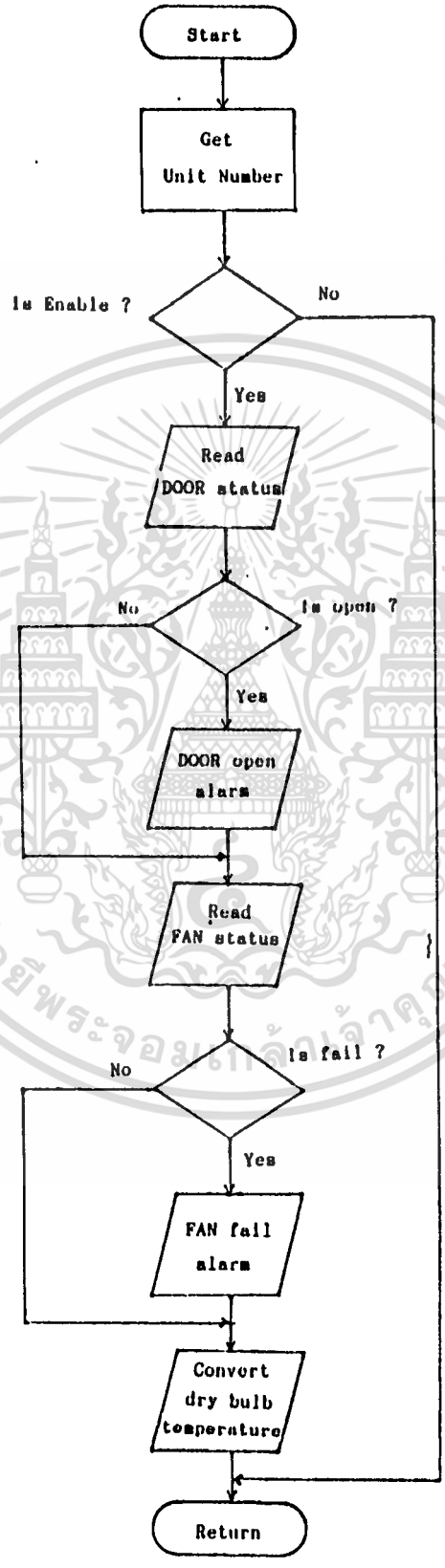
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flow Chart) ขั้นตอนการทำงานของโปรแกรมตอบสนองการอินเทอรัพท์ 1C ในรูปที่ 3.13



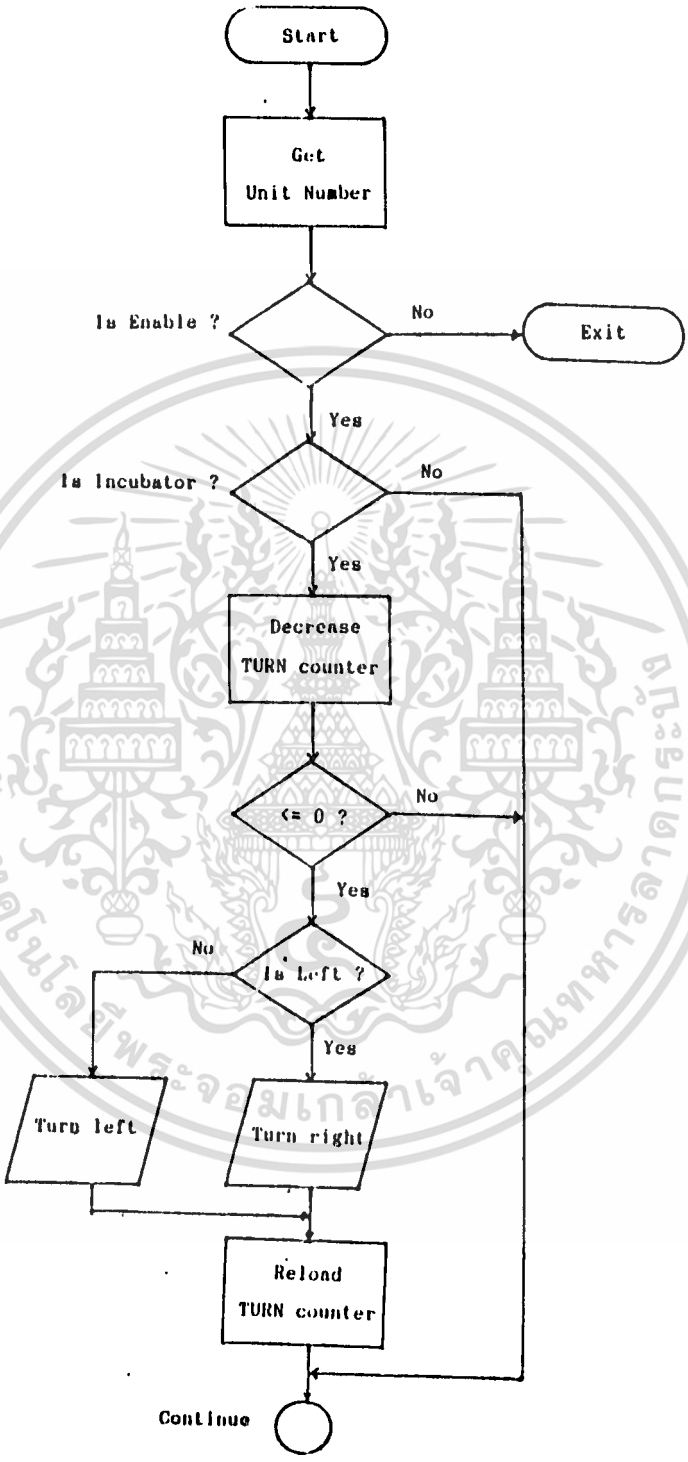
รูป 3.13 โฟลชาร์ทขั้นตอนการทำงานหลักของโปรแกรมตอบสนองการอินเทอรัพท์ 1C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



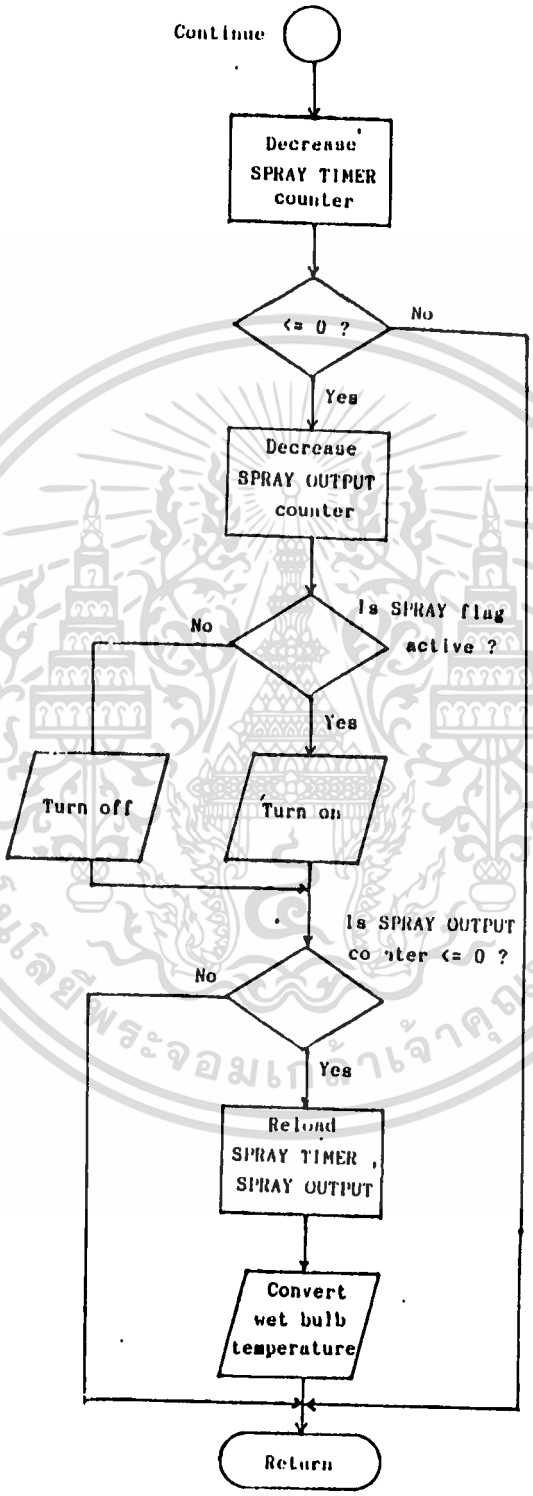
รูป 3.13 (ต่อ) แผนภาพการทำงานของ Temperature Process ในการทำงานหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



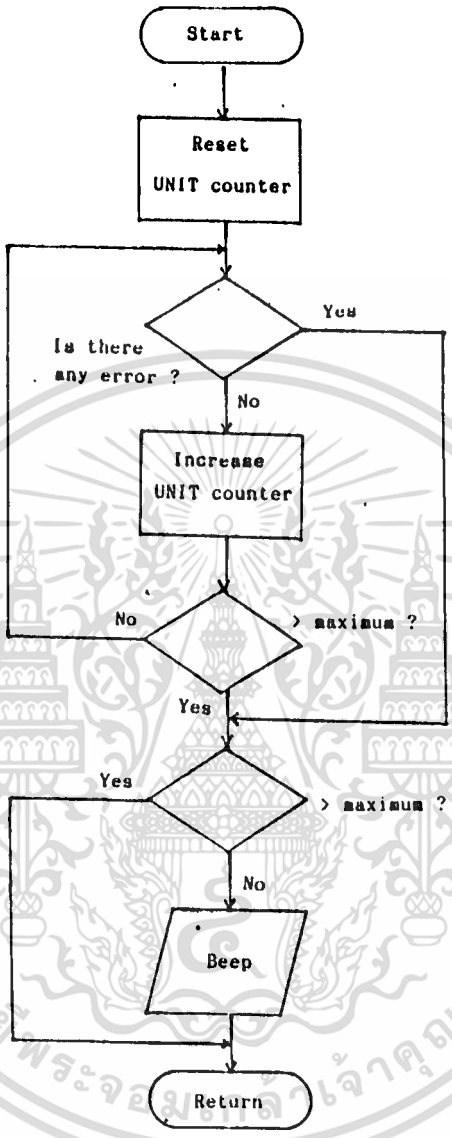
รูป 3.13 (ต่อ) โฟลชาร์ตส่วนแรกของ Humidity Process ในการทำงานหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.13 (ต่อ) ฟลอชาร์ทส่วนหลังของ Humidity Process ในการทำงานหลัก

เอกสารนี้มีไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.13 (ต่อ) โฟลชาร์ทของ Error Process ในการทำงานหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.11.2 โปรแกรมตอบสนองอินเทอรัพท์หมายเลข 0A (ฐาน 16)

การอ่านค่าอุณหภูมิดิจิตอลดรายบัลย์หรือเวทบัลย์ไม่สามารถอ่านได้ทันที ต้องรอให้ตัวเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล เปลี่ยนสัญญาณให้ และกินเวลาถึง 0.1 มิลลิวินาทีในการเปลี่ยนแต่ละครั้ง ซึ่งถ้าให้โปรแกรมวนรอ (Loop) เพื่ออ่านค่าที่เปลี่ยนได้ จะเป็นการสูญเปล่ากับการทำงานของเครื่องคอมพิวเตอร์ จึงใช้การอินเทอรัพท์เข้าช่วย โดยเมื่อตัวเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอลทำการเปลี่ยนสัญญาณเสร็จ ให้สร้างสัญญาณอินเทอรัพท์หมายเลข 0A (ฐาน 16) ผ่านจากระบบขยายบัสของระบบภายนอก ภายในโปรแกรมจะเตรียมโปรแกรมตอบสนองอินเทอรัพท์ หมายเลข 0A ไว้แล้วโดยมีการทำงานดังนี้

1) ตรวจสอบว่าเป็นการอ่านอุณหภูมิดรายบัลย์ หรือเวทบัลย์ และดูที่เท่าไหร่ ถ้าเป็นดรายบัลย์ให้ทำข้อ 2 และถ้าเป็นเวทบัลย์ให้ทำข้อ 5

2) อ่านอุณหภูมิดรายบัลย์จากตัวเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล

3) ควบคุมการทำงานของขดลวดความร้อน

4) เซ็ทบิทแสดงข้อผิดพลาด เมื่ออุณหภูมิดรายบัลย์สูงกว่าอุณหภูมิปลอดภัยสูงสุด หรือต่ำกว่าอุณหภูมิปลอดภัยต่ำสุด ข้ามไปข้อ 9

5) อ่านอุณหภูมิเวทบัลย์จากตัวเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล

6) หาค่าความชื้นสัมพัทธ์ จากอุณหภูมิดรายบัลย์และเวทบัลย์

7) เซ็ทบิทบอกการทำงานของวาล์วท่อน้ำ

8) เซ็ทบิทแสดงข้อผิดพลาด เมื่อความชื้นสัมพัทธ์ที่หาได้สูงกว่าความชื้นสัมพัทธ์ปลอดภัยสูงสุดหรือต่ำกว่าความชื้นสัมพัทธ์ปลอดภัยต่ำสุด

9) จบการทำงาน

รูป 3.14 แสดงโฟลชาร์ทการทำงานของโปรแกรมตอบสนองอินเทอรัพท์หมายเลข 0A โปรแกรมตอบสนองอินเทอรัพท์ทั้งหมดเขียนด้วยภาษาแอสแซมบลี้ (Assembly Language) เนื่องจากต้องการให้การทำงานเร็วที่สุด

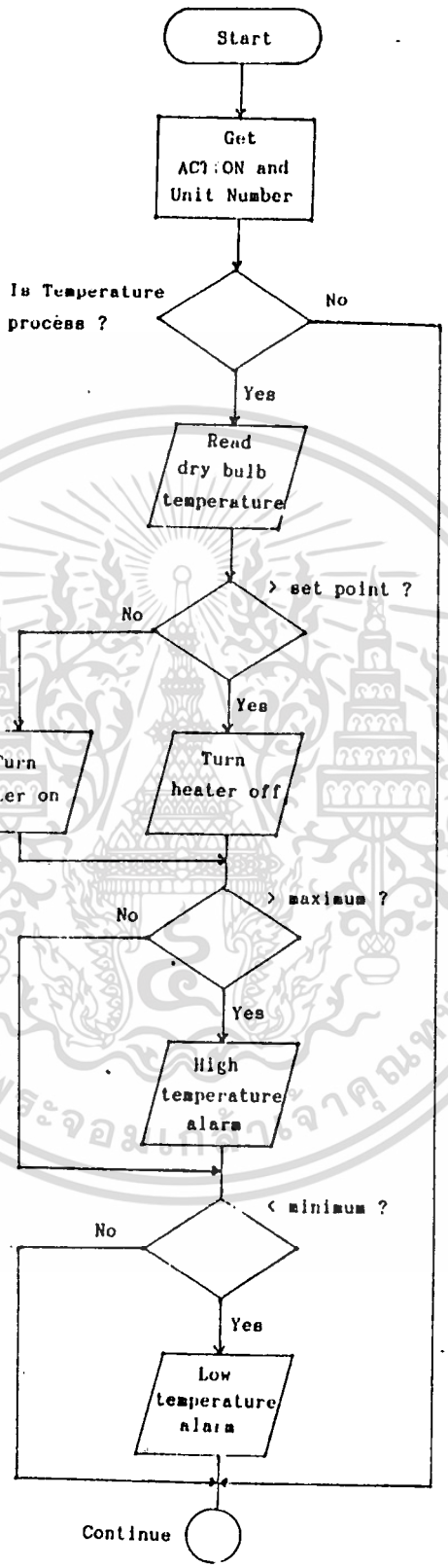
3.11.3 โปรแกรมการทำงานหลัก

หน้าที่ของโปรแกรมหลักมีดังนี้

1) ติดตั้งโปรแกรมตอบสนองอินเทอรัพท์หมายเลข 1C และ 0A (ฐาน 16) เข้าไปแทนที่โปรแกรมเดิม

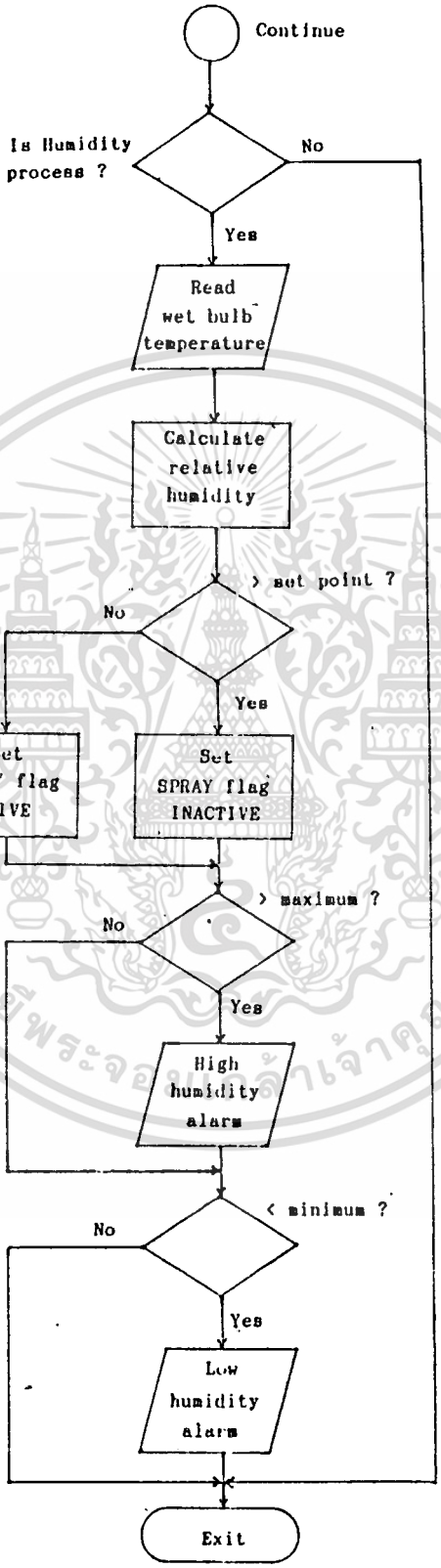
2) แสดงค่าของอุณหภูมิและความชื้นสัมพัทธ์ทางจอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.14 โฟลชาร์ทส่วนแรกของโปรแกรมคอมสนองอินเทอรัพท์หมายเลข 0A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.14 (ต่อ) โฟลชาร์ทส่วนหลังของโปรแกรมตอบสนองอินเทอรัพท์หมายเลข 0A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) แสดงสถานะการพลิกของถาดไข่ในตู้ฟักไข่ทางจอภาพ

4) ให้ตั้งค่าอุณหภูมิและความชื้นสัมพัทธ์ในการใช้งาน

รูป 3.15 แสดงโฟลชาร์ทการทำงานของโปรแกรมหลัก ซึ่งเขียนขึ้นด้วยภาษาซี (C Language)

3.11.4 โปรแกรมใช้งานอื่น ๆ

เพื่อช่วยการทำงานด้านอื่น ๆ ของโปรแกรมหลักให้สมบูรณ์ขึ้นเกี่ยวกับด้านจอภาพ ซึ่งก็มี

1) การกำหนดตำแหน่งของเคอร์เซอร์ (Cursor)

2) การเขียนประโยค (String) ของตัวหนังสือ

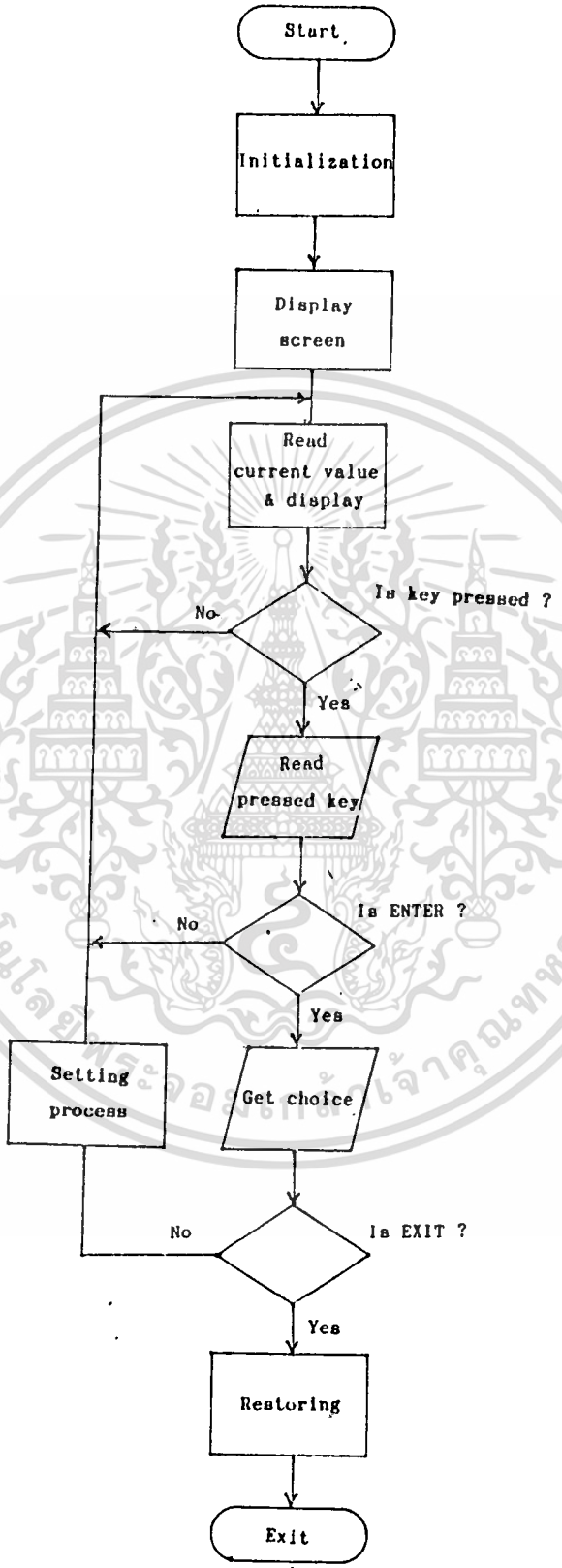
3) การเปลี่ยนตัวเลขให้เป็นตัวหนังสือ

4) การลบจอภาพ

5) การทำพูลดาวน์เมนู (Pull Down Menu)

ซึ่งเขียนเป็นโปรแกรมภาษาแอสแซมบลีและภาษาซี ส่วนโปรแกรมต้นฉบับทั้งหมดสามารถดูได้จากภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



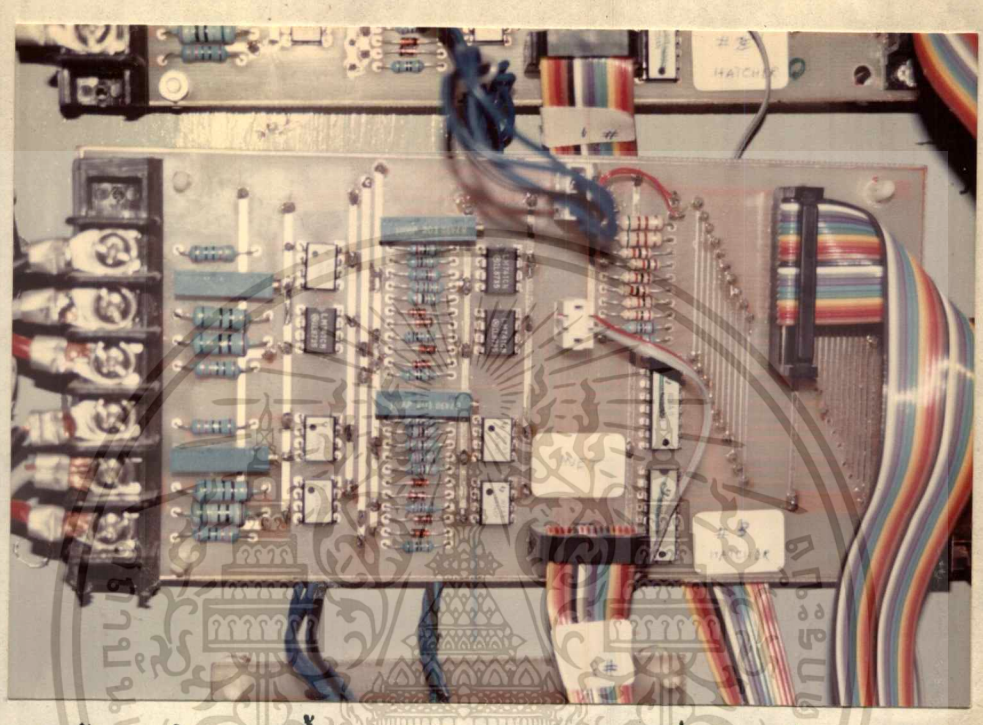
รูป 3.15 โฟลชาร์ทการทำงานของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การใช้งานและผลการใช้งาน

4.1 การต่อเครื่องคอมพิวเตอร์กับฮาร์ดแวร์สร้างเพิ่มเติมและการใช้งาน

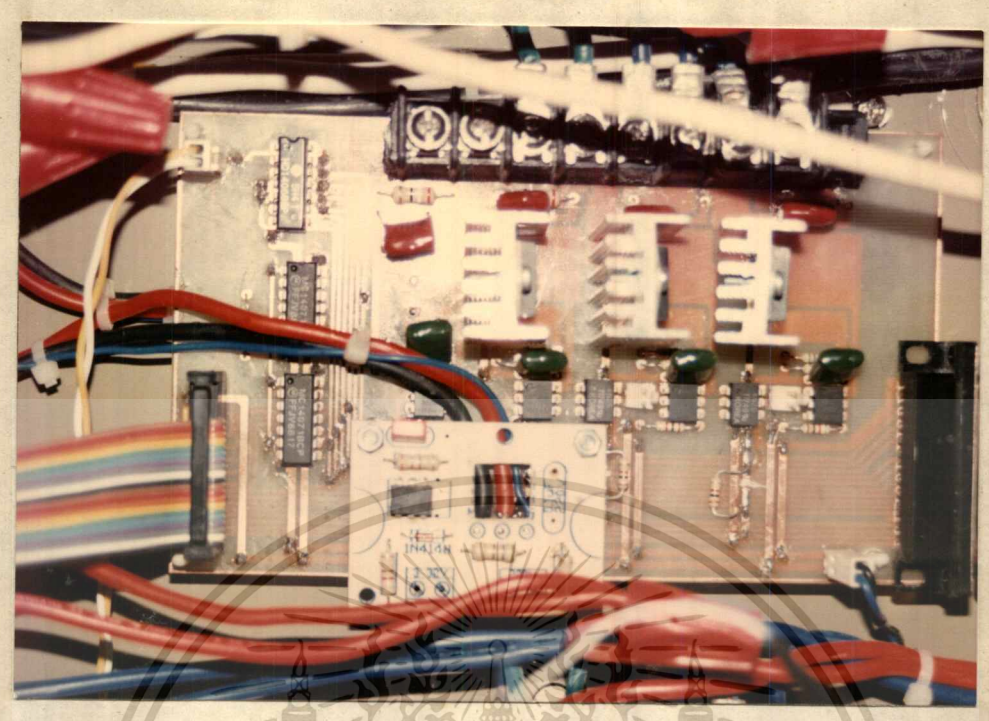


รูป 4.1 วงจรวัดอุณหภูมิและความชื้น (อยู่บริเวณตรงกลาง) เมื่อลงปริ้นท์เซอร์กิทบอร์ด (Print Circuit Board) แล้ว

รูป 4.1 แสดงวงจรวัดอุณหภูมิและความชื้น ทางซ้ายมือของรูปเป็นเทอมินัล (Terminal) สำหรับต่อไปยังตัววัดอุณหภูมิ พีที-100 สองตัวที่ติดตั้งอยู่ภายในตู้ ส่วนทางขวามือจะเป็นภาคขับสัญญาณสำหรับการแสดงสถานะและแจ้งเตือน

รูป 4.2 แสดงวงจรสวิทช์ไฟกำลังสูง (อยู่บริเวณกลางรูป) เมื่อลงปริ้นท์เซอร์กิตบอร์ด ทางด้านบนของรูปเป็นเทอมินัลสำหรับต่อขดลวดความร้อน วาล์วท่อน้ำ วาล์วท่อลมทางซ้าย และวาล์วท่อลมทางขวา ทางขวามือเป็นคอนเนคเตอร์ (Connector) แบบ ดีบี 25 (DB 25) เพื่อต่อสัญญาณควบคุมการทำงาน สัญญาณแสดงสถานะการทำงานและสัญญาณแจ้งเตือน ส่วนไตรแอค (Triac) สำหรับขดลวดความร้อนอยู่อีกกล่องหนึ่ง เนื่องจากต้องติดตั้งฮีทซิงค์ (Heat Sink)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.2 วงจรสวิทช์ไฟกำลังสูง (อยู่บริเวณกลางรูป) เมื่อลงปริ้นท์เซอร์กิตบอร์ด

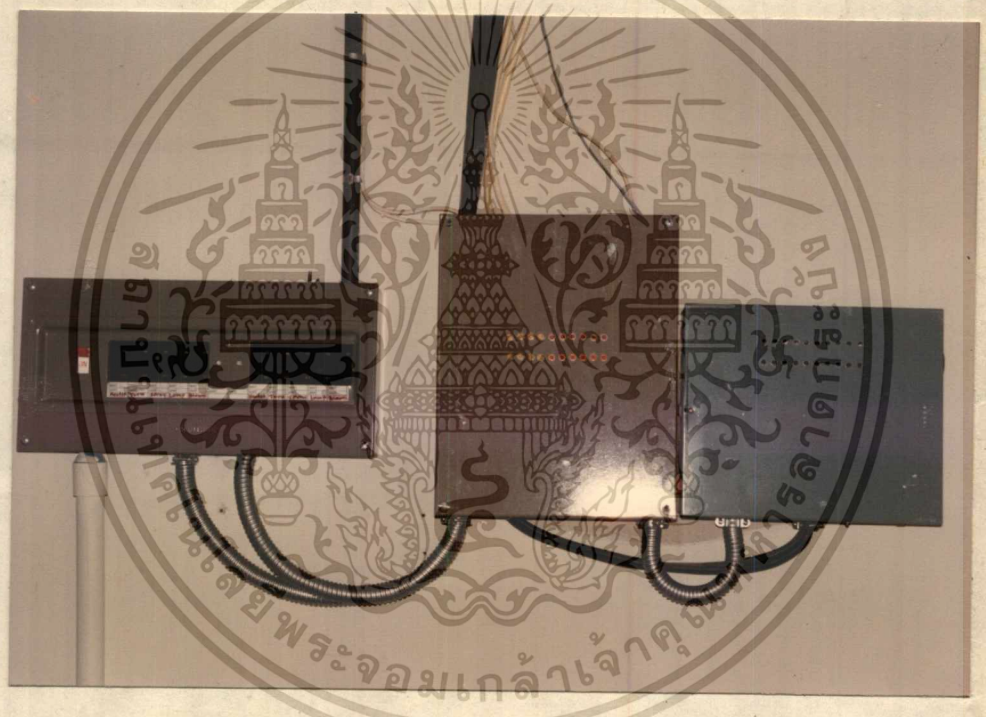
รูป 4.3 แสดงเครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ทีกับฮาร์ดแวร์ที่ออกแบบสร้างเพิ่มเติมเพื่อใช้ในการฟักไข่ ในภาพเป็นการต่อในขณะใช้งานจริง ตัวเครื่องคอมพิวเตอร์จะชิลด์ (Shield) ไว้ เพื่อไม่ให้เกิดสัญญาณรบกวน (Noise) มายังภายนอก ภายในเครื่องคอมพิว

รูป 4.3 เครื่องคอมพิวเตอร์ไอบีเอ็มพีซีเอ็กซ์ทีกับฮาร์ดแวร์ที่ออกแบบสร้างเพิ่มเติมเพื่อใช้ในการฟักไข่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เตอร์มีการ์ดขยายบัสของระบบ ส่วนในกล่องทางซ้ายมือมีการ์ดขยายบัสของระบบภายนอก การ์ดเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล การ์ดอินพุท-เอาท์พุทพอร์ท วงจรวัดอุณหภูมิและความชื้น พร้อมทั้งแอลอีดีแสดงสถานะการทำงานและแจ้งเตือน

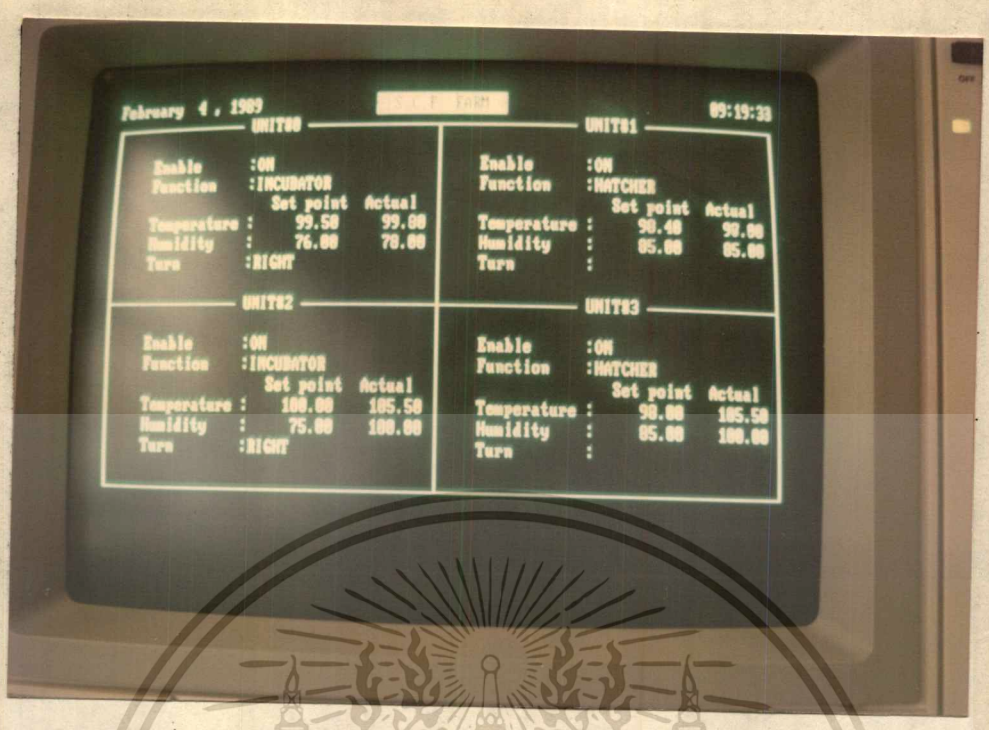
รูป 4.4 แสดงกล่องบรรจุวงจรสวิทช์ไฟกำลังสูง (กล่องกลาง) พร้อมทั้งแอลอีดีแสดงสถานะการทำงานและแจ้งเตือน กล่องทางซ้ายเป็นเซอร์กิตเบรกเกอร์ (Circuit Breakers) ไว้ตัดไฟในกรณีที่วงจรสวิทช์ไฟกำลังสูงลัดวงจร (Short Circuit) กล่องทางขวามือบรรจุไตรแอคซึ่งเป็นสวิทช์สำหรับขดลวดความร้อน 6,000 วัตต์ ซึ่งต้องติดฮีทซิงค์มีขนาดใหญ่พอสมควร



รูป 4.4 กล่องบรรจุวงจรสวิทช์ไฟกำลังสูง (กล่องกลาง) พร้อมทั้งแอลอีดีแสดงการทำงานและแจ้งเตือน

รูป 4.5 แสดงจอภาพขณะกำลังทำงาน ตู้หมายเลข 0 และ 1 เท่านั้นที่ต่ออุปกรณ์สำหรับทำงานครบสมบูรณ์ทุกอย่าง บนจอภาพจะแสดงวันที่ปัจจุบันทางข้างบนซ้ายมือของจอ เวลาปัจจุบันทางข้างบนขวามือของจอ จอจะตีกรอบแบ่งออกเป็น 4 ส่วน เพื่อแสดงรายละเอียดของแต่ละตู้ที่กำลังใช้งาน ซึ่งรายละเอียดก็มี สถานะการทำงานของตู้ (ทำงาน / ไม่ทำงาน) หน้าที่ของตู้ (ตู้ฟักไข่ / ตู้เกิดลูกไก่) อุณหภูมิที่ตั้งไว้ อุณหภูมิจริงภายในตู้ ความชื้นที่ตั้งไว้ ความชื้นจริงภายในตู้ สถานะของการพลิกถาดไข่ (เฉพาะตู้ฟักไข่เท่านั้น)

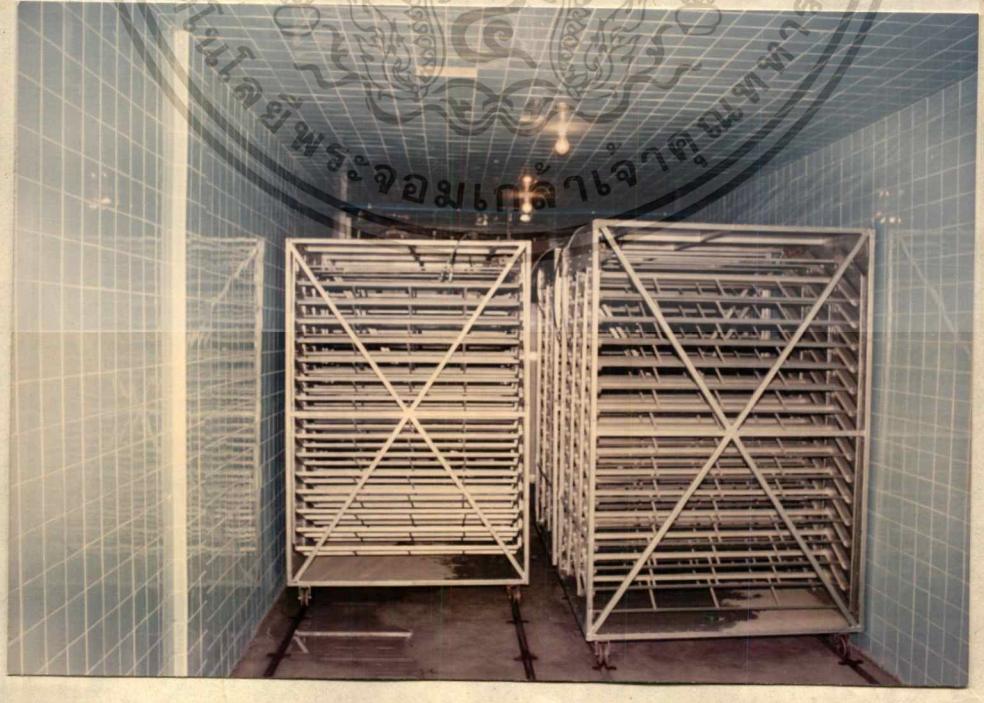
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.5 จอภาพขณะกำลังทำงาน

4.2 อุปกรณ์อื่น ๆ ที่ใช้ในการทำงาน

รูป 4.6 แสดงให้เห็นรถใส่ไข่ที่จะเข้าตู้ฟักไข่ โดยที่รถแต่ละคันจะติดกระบอกลมเพื่อพลิกรางวางถาดไข่ให้เอนไปทางซ้ายหรือขวา รถแต่ละคันแบ่งเนื้อที่ออกเป็น 15 ชั้น ในแต่ละชั้นมี 3 ราง ในแต่ละรางสามารถวางถาดไข่ขนาด 36 ฟองได้ 4 ถาด หรือเท่ากับว่ารถ 1 คัน



รูป 4.6 รถใส่ไข่ที่จะเข้าตู้ฟักไข่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถบรรทุกไข่ได้ 6,480 ฟอง ตู้ฟักไข่สามารถบรรจุรถได้ทั้งหมด 12 คัน หรือเท่ากับไข่ 77,760 ฟอง ไข่จะฟักในตู้ฟักไข่เป็นเวลา 18 วันก่อนจะถูกย้ายไปเข้าตู้เกิดลูกไก่ต่อไป

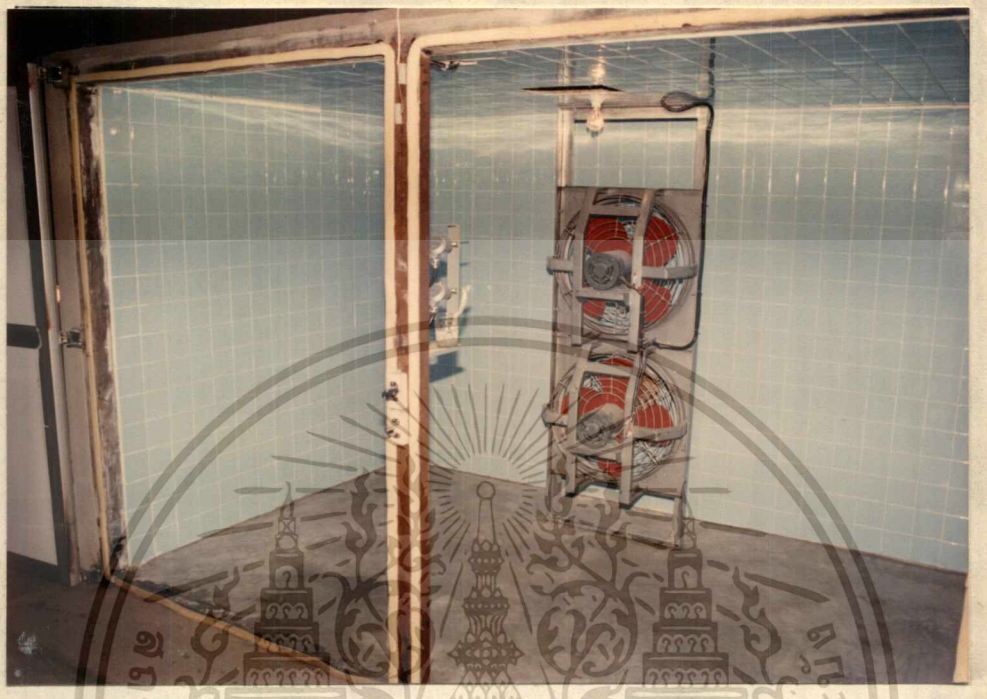
รูป 4.7 แสดงให้เห็นรถใส่ไข่ที่จะเข้าตู้เกิดลูกไก่ ไข่ที่ใส่จะเป็นไข่ที่ออกมาจากตู้ฟักไข่หลังจากฟักได้ 18 วันแล้ว โดยรถจะมีถาดสำหรับใส่ไข่ทั้งหมด 45 ถาด แต่ละถาดจะใส่ไข่เท่ากับจำนวนไข่บนรางของรถในตู้ฟักไข่ หรือเท่ากับไข่ 144 ฟอง ตู้เกิดลูกไก่ สามารถบรรจุรถสำหรับให้ลูกไก่เกิดได้ 2 คันซึ่งคิดเป็นจำนวนไข่ที่จะเกิด 12,960 ฟอง หลังจากไข่เข้าในตู้ 3 วัน ลูกไก่ก็จะออกจากไข่ ซึ่งในระหว่างนี้จะต้องใส่สารฆ่าเชื้อด่างทับทิม, ฟอร์มาลีน เข้าไปในตู้เพื่อฆ่าเชื้อด้วย เมื่อลูกไก่ขนแห้งแล้ว ก็สามารถย้ายไปเลี้ยงไก่ได้

รูป 4.7 รถใส่ไข่ที่จะเข้าตู้เกิดลูกไก่

รูป 4.8 แสดงให้เห็นภายในของตู้เกิดลูกไก่ในขณะที่ยังไม่ใส่รถ พัดลมจะติดอยู่ด้านหลังของตู้ตรงกลาง และหลังพัดลมจะเป็นขดลวดความร้อนและหัวพ่นน้ำ โดยพัดลมจะพัดให้ลมไปปะทะกับกำแพงด้านหลัง เพื่อให้ลมนำเอาความร้อนและความชื้นกระจายไปทั่วห้อง ที่เสาตรงกลางด้านหน้าของตู้ จะติดตัวตรวจจับสถานะของประตู ซึ่งเป็นแมกเนติกสวิทช์ (Magnetic Switch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

) ที่ข้างในจะเป็นแท่นสำหรับยึดตัวตรวจวัดอุณหภูมิสำหรับดรายบัลย์และเวทบัลย์ (พีที-100) มีถาดน้ำสำหรับเวทบัลย์



รูป 4.8 ภายในตู้เกิดลูกไก่ในขณะที่ยังไม่ใส่รถ

4.3 การตั้งค่าอุณหภูมิและความชื้น

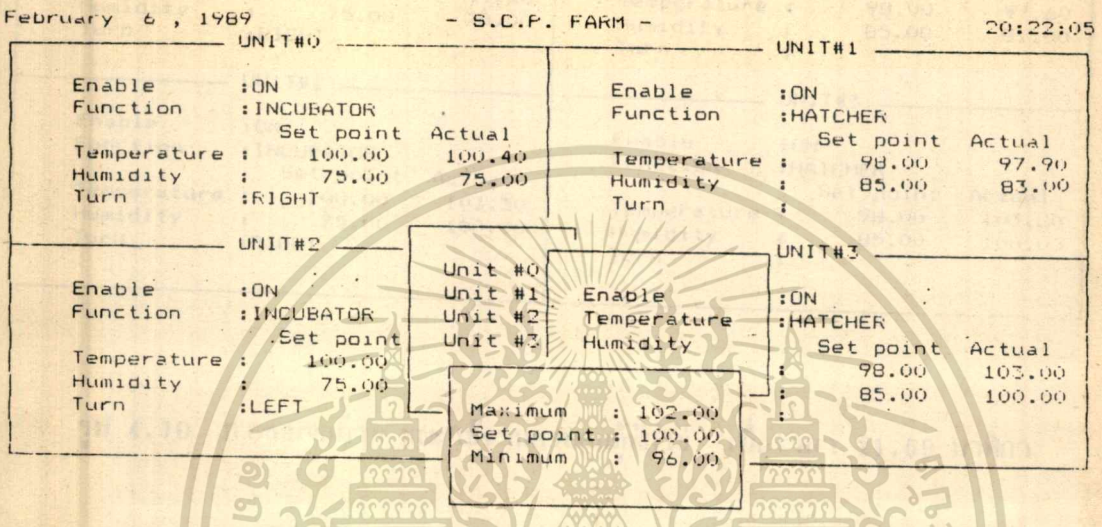
คนควบคุมการทำงานของเครื่องฟักไข่สามารถตั้งค่าอุณหภูมิและความชื้นภายในตู้ที่ต้องการได้ ทั้งนี้เพื่อให้อุณหภูมิและความชื้นภายในตู้ฟักไข่และตู้เกิดลูกไก่เหมาะสมกับการฟักไข่และการเกิดลูกไก่มากที่สุด แต่จะตั้งได้เพียงแค่ภายในช่วงที่ปลอดภัยเท่านั้น

โดยการกดคีย์ ENTER ระหว่างการทำงาน บนจอภาพจะขึ้นข้อความให้เลือกว่า จะตั้งค่าตู้หมายเลขที่เท่าไหร่ ซึ่งสามารถใช้ปุ่มลูกศรขึ้นและลงเพื่อเลื่อนข้อเลือก เมื่อถึงข้อเลือกที่ต้องการให้กด ENTER หรือหมายเลขตู้ จอภาพขึ้นข้อความให้เลือกว่าจะตั้งค่าอุณหภูมิหรือความชื้น ใช้วิธีการเดิมในการเลือกข้อเลือก จากนั้นบนจอภาพแสดงอุณหภูมิตั้งไว้ก่อนหน้าถ้าเลือกตั้งค่าอุณหภูมิ หรือแสดงความชื้นที่ตั้งไว้ก่อนหน้าถ้าเลือกตั้งค่าความชื้น พร้อมทั้งแสดงค่ามากที่สุดและค่าน้อยที่สุดที่สามารถตั้งได้ ค่าที่ตั้งจะต้องน้อยกว่าหรือเท่ากับค่ามากที่สุด และมากกว่าหรือเท่ากับค่าที่น้อยที่สุด การตั้งค่าใหม่ทำได้โดยการใช้ลูกศรทางขวาเพื่อเพิ่มค่าอุณหภูมิหรือความชื้น ใช้ลูกศรทางซ้ายเพื่อลดค่าอุณหภูมิหรือความชื้น

ค่าของอุณหภูมิที่เพิ่มขึ้นหรือลดลงในหนึ่งครั้งเท่ากับ 0.1 องศาฟาเรนไฮต์ ค่าของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความชื้นที่เพิ่มขึ้นหรือลดลงในหนึ่งครั้งเท่ากับ 1 เปอร์เซนต์ เมื่อได้ค่าอุณหภูมิและความชื้นที่ต้องการและต้องการให้เป็นค่าตั้งในการทำงานต่อไปให้กด ENTER แต่ถ้าต้องการยกเลิกให้กด ESC ค่าอุณหภูมิหรือความชื้นตั้งก็ยังคงเป็นค่าเดิม ในการออกจากการตั้งค่าในแต่ละหัวข้อเลือกให้ใช้ปุ่ม ESC รูป 4.9 แสดงภาพบนจอภาพที่ได้จากการสั่งพิมพ์จอภาพ (Print Screen) ในขณะที่ทำการตั้งค่า ซึ่งข้อเลือกจะขึ้นมาในลักษณะของหน้าต่าง (Window) ซ้อนกัน



รูป 4.9 ภาพบนจอภาพที่ได้จากการสั่งพิมพ์จอภาพในขณะที่ทำการตั้งค่า

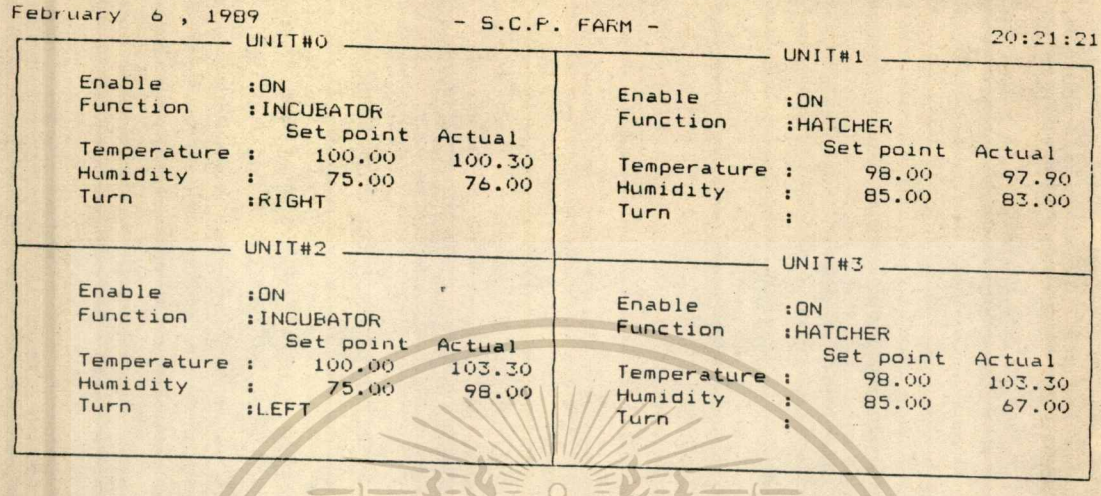
4.4 การใช้งาน

ตู้ฟักหมายเลข 0 และตู้เกิดลูกไก่ ได้เริ่มทำการฟักไก่ วันที่ 17 มกราคม พศ.2532 โดยใส่ไข่เต็มรถ 1 คัน ซึ่งไข่นี้บางส่วนจะผ่านการเก็บในห้องเย็นเพื่อเก็บรักษาไข่ที่อุณหภูมิประมาณ 21 องศาเซลเซียส บางส่วนจะไม่ผ่านห้องเย็น หลังจากนั้น 18 วัน ในวันที่ 4 กุมภาพันธ์ พศ.2532 นำไข่ออกจากตู้ฟักไข่ ออกมาคัดเอาเฉพาะไข่ที่เจริญพันธุ์เต็มที่ ๆ จะสามารถเกิดเป็นลูกไก่ได้และนำมาใส่ในรถสำหรับลูกไก่เกิด นำเข้าสู่ตู้เกิดลูกไก่ต่อไป และในวันที่ 6 กุมภาพันธ์ พศ.2532 เป็นวันที่ลูกไก่ออกจากไข่

4.5 ผลการใช้งาน

4.5.1 ด้านการควบคุมอุณหภูมิ และความชื้นภายในตู้ฟักไข่และตู้เกิดลูกไก่ จอภาพซึ่งแสดงอุณหภูมิและความชื้นภายในตู้ฟักไข่หมายเลข 0 และตู้เกิดลูกไก่หมายเลข 1 ด้วยการพิมพ์จอภาพซึ่งแสดงอุณหภูมิและความชื้นภายในตู้ต่าง ๆ เมื่อวันที่ 6 กุมภาพันธ์ พศ.2532 เมื่อเวลาประมาณ 11.59 นาฬิกา, 13.47 นาฬิกา และ 20.21 นาฬิกา ซึ่งแสดงในรูปที่ 4.10, 4.11 และ 4.12 ตามลำดับ แสดงให้เห็นว่า อุณหภูมิในตู้ผิดจากอุณหภูมิที่ตั้งไว้มากที่สุด 0.4 องศาฟา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.12 ภาพบนจอภาพ ขณะใช้งานควบคุมตู้ฟักไข่ เมื่อเวลา 20.21 นาฬิกา

4.5.2 ด้านการพลิกไข่ การแสดงสถานะการทำงานและแจ้งเตือนจากการใช้งาน การทำงานเหล่านี้ สามารถทำงานได้อย่างสมบูรณ์

4.5.3 จำนวนลูกไก่ที่ออกมาเป็นตัว

จากการนำไข่ที่ฟักมาเป็นเวลา 18 วัน นำมาคัดไข่ที่เจริญพันธุ์เต็มที่เพื่อนำไปใส่ตู้เกิดลูกไก่ พบว่าเป็นไข่ลมหรือไข่ที่เจริญพันธุ์ไม่เต็มที่ คิดเป็นเปอร์เซนต์ประมาณ 9 เปอร์เซนต์ของไข่ที่เข้าตู้ฟักไข่ทั้งหมด ไข่อยู่ในตู้เกิดลูกไก่ 3 วัน แล้วก็จะออกมาเป็นตัว ซึ่งคิดเป็นเปอร์เซนต์ได้ประมาณ 75 เปอร์เซนต์ของไข่ที่เข้าตู้ฟักไข่ทั้งหมด รูป 4.13 แสดงให้เห็นไข่ในตู้เกิดลูกไก่ที่ถ่ายมาจากตู้ฟักไข่ รูป 4.14 แสดงให้เห็นเป็นตัวลูกไก่ที่เกิดมาในรถเกิดลูกไก่คันหนึ่งที่เกิดขึ้นมาหลังจากอยู่ในตู้เกิดลูกไก่เป็นเวลา 3 วัน รูป 4.15 แสดงให้เห็นลูกไก่ที่เกิดมาในถาดหนึ่งของรถในรูป 4.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 4.13 ไข่ในตู้เกิดลูกไก่ที่ถ่ายมาจากตู้ฟักไข่

รูป 4.14 ตัวลูกไก่ที่เกิดในรถเกิดลูกไก่คันหนึ่ง ในตู้เกิดลูกไก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 4.15 ลูกไก่ที่เกิดมาในภาคหนึ่งของรถในรูป 4.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและวิจารณ์

5.1 เกี่ยวกับระบบควบคุม

จากผลการทดลองของการควบคุมอุณหภูมิและความชื้น ภายในตู้ฟักไข่และตู้เกิดลูกไก่ให้ผลเป็นที่น่าพอใจ โดยระบบควบคุมสามารถควบคุมอุณหภูมิภายในตู้ให้ผิดไปจากค่าที่ตั้งไว้ไม่ถึง 0.5 องศาฟาเรนไฮต์ และความชื้นผิดไปเพียงไม่เกิน 5 เปอร์เซนต์ ซึ่งการที่สามารถควบคุมให้ตัวแปรสำคัญของความสำเร็จในการฟักไข่ ให้เปลี่ยนแปลงไปเท่าที่จะมีผลทำให้ไข่มีโอกาสฟักออกมาเป็นตัวมากขึ้น ส่วนเรื่องเวลาในการพลิกถาดไข่ในตู้ฟักไข่ ก็ถือว่าเที่ยงตรง โดยทำการพลิกไข่ทุก ๆ ประมาณ 45 นาที ซึ่งจะมีผลทำให้ไข่ได้รับความร้อนและความชื้นทั่วกันทั้งใบ ทำให้ไข่มีโอกาสฟักออกเป็นตัวมากขึ้นเช่นกัน

5.2 เกี่ยวกับจำนวนลูกไก่ที่ฟักได้

จากการฟักครั้งแรก และให้ผลการฟักของลูกไก่เป็น 75 เปอร์เซนต์ ถือว่าเป็นผลที่น่าพอใจ เนื่องจากไข่ที่ทำการฟักมีสภาพแวดล้อมก่อนการฟักไม่เหมือนกัน และผลของการเกิดเป็นลูกไก่นอกจากจะขึ้นกับการควบคุมอุณหภูมิและความชื้นภายในตู้และการพลิกไข่แล้ว ยังขึ้นอยู่กับไข่ที่นำมาฟักว่าเป็นไข่ที่ผสมพันธุ์หรือไม่ การรักษาความสะอาดก็เป็นปัจจัยหนึ่งเช่นกัน ซึ่งต่อไปเมื่อการควบคุมปัจจัยอื่น ๆ ดีขึ้น ก็จะทำให้เปอร์เซนต์การเกิดของลูกไก่สูงขึ้นไปด้วย

5.3 การพัฒนา

ทางด้านฮาร์ดแวร์ โดยออกแบบเพื่อทำหน้าที่เพิ่มเติมดังนี้

1) ตรวจเช็คแหล่งจ่ายไฟกำลังสูง

2) ควบคุมการทำงานแหล่งกำเนิดไฟสำรอง (Generator) ในกรณีที่ไฟดับ

3) ควบคุมการทำงานที่ปิดช่องอากาศของตู้ และพัดลมดูดอากาศออกจากตู้

ทางด้านโปรแกรม โดยการแบ่งโปรแกรมออกเป็น 2 ส่วน คือ

1) ส่วนควบคุมการทำงานของอุปกรณ์ต่าง ๆ

2) ส่วนแสดงผลและตั้งค่า

โดยโปรแกรมในส่วนควบคุมการทำงานจะเป็นโปรแกรมที่คงอยู่คลอด (Resident Program) ส่วนการทำงานของโปรแกรมยังคงเป็นลักษณะเดิม การควบคุมก็จะเป็นลักษณะเดิมแต่สามารถใช้เครื่องคอมพิวเตอร์ทำงานโปรแกรมอื่น ๆ แต่ทั้งนี้โปรแกรมนั้นจะต้องไม่รบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรมควบคุม อาจเพิ่มการทำงานดังนี้

1) เพิ่มความสามารถในการรายงานออกทางเครื่องพิมพ์ (Printer) ตามระยะเวลาที่กำหนด

2) เพิ่มความสามารถในการเก็บข้อมูลลงดิสค์ (Disk) ตามระยะเวลาที่กำหนด

3) เพิ่มความสามารถในการควบคุมระยะไกล (Remote Control)

4) เพิ่มการป้องกันการใช้โดยไม่ได้รับอนุญาติ

5) สามารถตั้งค่าในระดับฮาร์ดแวร์สำหรับผู้ติดตั้งระบบ

6) แสดงรายละเอียดเกี่ยวกับตู้บนจอภาพมากขึ้น

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวกที่ 1

ในภาคผนวกที่ 1 แสดงต้นฉบับของโปรแกรมควบคุมการทำงาน ซึ่งเขียนด้วยภาษาแอสแซมบลี้ และภาษาซีโว้ดยละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*** Main program ***/

#include <stdio.h>
#include <conio.h>
#include <time.h>
#include "menu.h"
#include "key.h"

extern int  c_pos( int , int ) ;   /* set cursor position */
extern void prnchr( int , int ) ;  /* print a character with attribute */
extern int  clear( int , int , int , int , int ) ;
extern int  c_blink( void ) ;      /* set cursor blink */
extern int  c_blank( void ) ;      /* set cursor blank */
extern int  init( void ) ;         /* set up interrupt routine */
extern void release( void ) ;      /* release interrupt routine */

       int  prnstr( int , int , char * , int ) ; /* print character string */
       void scr1( int ) ;          /* display screen#1 */
       void scr2( int ) ;          /* display screen#2 */
       void scr3( int ) ;          /* display screen#3 */
       char *itoc( long , int ) ;  /* converts a long to a string */
       int  set( void ) ;          /* set temperature and humidity */

#define NORMAL  0x07
#define REVERSE 0x70
#define BRIGHT  0x0F
#define BL_NOR  0x17
#define BL_REV  0xF0
#define EXIT    4

char *month[] = { "January" , "February" , "March" , "April" , "May" , "June" ,
    "July" , "August" , "September" , "October" , "November" , "December" } ;

main()
    {
    struct tm *s_time ;
    time_t long_time ;
    int  *scr_c1 , *scr_c2 , result ;
    char *tstr ;

    c_blank() ;
    clear( 0 , 0 , 24 , 79 , NORMAL ) ;
    if( init() != 0 )
        return ;
    scr1( NORMAL ) ;
    scr2( 0 ) ;
    scr2( 1 ) ;
    scr2( 2 ) ;
    scr2( 3 ) ;
    for( ; ; )
        {
        if( kbhit() != 0 )
            {
            if( getch() == ENTER )
                {
                if( set() == EXIT )
                    break ;
                scr2( 0 ) ;
                scr2( 1 ) ;
                scr2( 2 ) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    scr2( 3 ) ;
                    }
               }
          else
               {
               time( &long_time ) ;
               s_time = localtime( &long_time ) ;
               c_pos( 0 , 0 ) ;
               printf( "%s %2d , %4d" , month[s_time->tm_mon] , s_time->tm_mday , s_time->tm_year + 1900 ) ;
               tstr = itoc( (long) s_time->tm_hour + 100 , 3 ) ;
               prnstr( 0,71, tstr+1 , NORMAL ) ;
               prnstr( 0,73, ":" , NORMAL ) ;
               tstr = itoc( (long) s_time->tm_min + 100 , 3 ) ;
               prnstr( 0,74, tstr+1 , NORMAL ) ;
               prnstr( 0,76, ":" , NORMAL ) ;
               tstr = itoc( (long) s_time->tm_sec + 100 , 3 ) ;
               prnstr( 0,77, tstr+1 , NORMAL ) ;
               scr3( 0 ) ;
               scr3( 1 ) ;
               scr3( 2 ) ;
               scr3( 3 ) ;
               }
          }
     release() ;
     clear( 0 , 0 , 24 , 79 , NORMAL ) ;
     c_pos( 0 , 0 ) ;
     c_blink() ;
     }


/*** Display screen#1 ***/

#include <stdio.h>
#include <stdlib.h>

extern int  c_pos( int , int ) ;
extern void prnchr( int , int ) ;
extern void repl( int , int , int ) ;

#define REVERSE  0x70
#define UPLEFT   0xda
#define UPRIGHT  0xbf
#define LOLEFT   0xc0
#define LORIGHT  0xd9
#define HOR      0xc4
#define HOR_D    0xc2
#define HOR_U    0xc1
#define VER      0xb3
#define VER_L    0xc3
#define VER_R    0xb4
#define CENTER   0xc5
#define MAXROW   24
#define MAXCOL   79

void scr1( at )
int  at ;
     {
     int row = 1 ;
     prnstr( 0 , 32 , "- S.C.P. FARM -" , REVERSE ) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    c_pos( row , 0 ) ;
    prnchr( UPLEFT , at ) ;
    c_pos( row , 1 ) ;
    repl( HOR , 38 , at ) ;
    c_pos( row , 39 ) ;
    prnchr( HOR_D , at ) ;
    c_pos( row , 40 ) ;
    repl( HOR , 38 , at ) ;
    c_pos( row , 78 ) ;
    prnchr( UPRIGHT , at ) ;
    for( row = 2 ; row < 10 ; row++ )
         {
         c_pos( row , 0 ) ;
         prnchr( VER , at ) ;
         c_pos( row , 39 ) ;
         prnchr( VER , at ) ;
         c_pos( row , 78 ) ;
         prnchr( VER , at ) ;
         }
    c_pos( row , 0 ) ;
    prnchr( VER_L , at ) ;
    c_pos( row , 1 ) ;
    repl( HOR , 38 , at ) ;
    c_pos( row , 39 ) ;
    prnchr( CENTER , at ) ;
    c_pos( row , 40 ) ;
    repl( HOR , 38 , at ) ;
    c_pos( row , 78 ) ;
    prnchr( VER_R , at ) ;
    for( row = 11 ; row < 19 ; row++ )
         {
         c_pos( row , 0 ) ;
         prnchr( VER , at ) ;
         c_pos( row , 39 ) ;
         prnchr( VER , at ) ;
         c_pos( row , 78 ) ;
         prnchr( VER , at ) ;
         }
    c_pos( row , 0 ) ;
    prnchr( LOLEFT , at ) ;
    c_pos( row , 1 ) ;
    repl( HOR , 38 , at ) ;
    c_pos( row , 39 ) ;
    prnchr( HOR_U , at ) ;
    c_pos( row , 40 ) ;
    repl( HOR , 38 , at ) ;
    c_pos( row , 78 ) ;
    prnchr( LORIGHT , at ) ;
    c_pos( 1 , 16 ) ;
    printf( " UNIT#0 " ) ;
    c_pos( 1 , 55 ) ;
    printf( " UNIT#1 " ) ;
    c_pos( 3 , 5 ) ;
    printf( "Enable      :" ) ;
    c_pos( 3 , 44 ) ;
    printf( "Enable      :" ) ;
    c_pos( 4 , 5 ) ;
    printf( "Function    :" ) ;
    c_pos( 4 , 44 ) ;
    printf( "Function    :" ) ;
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        c_pos( 5 , 20 ) ;
        printf( "Set point" ) ;
        c_pos( 5 , 31 ) ;
        printf( "Actual" ) ;
        c_pos( 5 , 59 ) ;
        printf( "Set point" ) ;
        c_pos( 5 , 70 ) ;
        printf( "Actual" ) ;
        c_pos( 6 , 5 ) ;
        printf( "Temperature :" ) ;
        c_pos( 6 , 44 ) ;
        printf( "Temperature :" ) ;
        c_pos( 7 , 5 ) ;
        printf( "Humidity    :" ) ;
        c_pos( 7 , 44 ) ;
        printf( "Humidity    :" ) ;
        c_pos( 8 , 5 ) ;
        printf( "Turn        :" ) ;
        c_pos( 8 , 44 ) ;
        printf( "Turn        :" ) ;
        c_pos( 10 , 16 ) ;
        printf( " UNIT#2 " ) ;
        c_pos( 10 , 55 ) ;
        printf( " UNIT#3 " ) ;
        c_pos( 12 , 5 ) ;
        printf( "Enable      :" ) ;
        c_pos( 12 , 44 ) ;
        printf( "Enable      :" ) ;
        c_pos( 13 , 5 ) ;
        printf( "Function    :" ) ;
        c_pos( 13 , 44 ) ;
        printf( "Function    :" ) ;
        c_pos( 14 , 20 ) ;
        printf( "Set point" ) ;
        c_pos( 14 , 31 ) ;
        printf( "Actual" ) ;
        c_pos( 14 , 59 ) ;
        printf( "Set point" ) ;
        c_pos( 14 , 70 ) ;
        printf( "Actual" ) ;
        c_pos( 15 , 5 ) ;
        printf( "Temperature :" ) ;
        c_pos( 15 , 44 ) ;
        printf( "Temperature :" ) ;
        c_pos( 16 , 5 ) ;
        printf( "Humidity    :" ) ;
        c_pos( 16 , 44 ) ;
        printf( "Humidity    :" ) ;
        c_pos( 17 , 5 ) ;
        printf( "Turn        :" ) ;
        c_pos( 17 , 44 ) ;
        printf( "Turn        :" ) ;
        }


/*** Display screen#2 and screen#3 ***/

extern int  c_pos( int , int ) ;  /* set cursor position */
extern void prnchr( int , int ) ; /* print a character with attribute */
extern int  t_status( int ) ;     /* return turn status */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
extern int  is_enble( int ) ;      /* return ZERO if enable */
extern int  is_incub( int ) ;      /* return ZERO if incubator */
extern int  temp_s( int ) ;        /* return set point temperature /10 */
extern int  temp_c( int ) ;        /* return current temperature /10 */
extern int  temp_o( int ) ;        /* return temperature offset */
extern int  humid_s( int ) ;       /* return set point humidity */
extern int  humid_c( int ) ;       /* return current humidity */

      char *ftoc( float , int , int ) ; /* converts floats to strings */
      int  prnstr( int , int , char * , int ) ; /* print character string */

#define NORMAL  0x07
#define REVERSE 0x70
#define BRIGHT  0x0F
#define BL_NOR  0x17
#define BL_REV  0xF0
#define MAXROW   24
#define MAXCOL   79
#define MAXUNIT  4

struct { int enrow ;
         int furow ;
         int efcol ;
         int terow ;
         int hurow ;
         int secol ;
         int cucol ;
         int turow ;
         int tucol } udis[MAXUNIT] = { { 3,4,18,6,7,22,32 ,8,18} ,
                                      { 3,4,57,6,7,61,71 ,8,57} ,
                                      { 12,13,18,15,16,22,32,17,18} ,
                                      { 12,13,57,15,16,61,71,17,57 } } ;

void scr2( un )
int  un ;
    {
    int  en , fu ;
    char *ts , *tc , *hs , *hc ;

    en = is_enble( un ) ;
    fu = is_incub( un ) ;
    prnstr( udis[un].enrow , udis[un].efcol , ( en == 0 ) ? "ON " : "OFF" , BRIGHT ) ;
    prnstr( udis[un].furow , udis[un].efcol , ( fu == 0 ) ? "INCUBATOR" : "HATCHER " , BRIGHT ) ;
    ts = ftoc( ((float) temp_s(un)/10 ) + ((float) temp_o(un)/2 ) ,6,2 ) ;
    prnstr( udis[un].terow , udis[un].secol , ts , BRIGHT ) ;
    tc = ftoc( ((float) temp_c(un)/10 ) + ((float) temp_o(un)/2 ) ,6,2 ) ;
    prnstr( udis[un].terow , udis[un].cucol , tc , BRIGHT ) ;
    hs = ftoc( (float) humid_s(un) ,6,2 ) ;
    prnstr( udis[un].hurow , udis[un].secol , hs , BRIGHT ) ;
    hc = ftoc( (float) humid_c(un) ,6,2 ) ;
    prnstr( udis[un].hurow , udis[un].cucol , hc , BRIGHT ) ;
    }

void scr3( un )
int  un ;
    {
    int  en ;
    char *ts , *tc , *hs , *hc ;
    if( ( en = is_enble( un ) ) != 0 )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            return ;
        ts = ftoc( ((float) temp_s(un)/10 ) + ((float) temp_o(un)/2 ) ,6,2 ) ;
        prnstr( udis[un].terow , udis[un].secol , ts , BRIGHT ) ;
        tc = ftoc( ((float) temp_c(un)/10 ) + ((float) temp_o(un)/2 ) ,6,2 ) ;
        prnstr( udis[un].terow , udis[un].cucol , tc , BRIGHT ) ;
        hs = ftoc( (float) humid_s(un) ,6,2 ) ;
        prnstr( udis[un].hurow , udis[un].secol , hs , BRIGHT ) ;
        hc = ftoc( (float) humid_c(un) ,6,2 ) ;
        prnstr( udis[un].hurow , udis[un].cucol , hc , BRIGHT ) ;
        if( is_incub( un ) == 0 )
            prnstr( udis[un].turow , udis[un].tucol , ( t_status( un ) ==0 ) ? "LEFT " : "RIGHT" , BRIGHT ) ;
        }


/*** Light bar menu ***/

#include <ctype.h>
#include "menu.h"
#include "key.h"

#define NORMAL   0x07
#define REVERSE  0x70

      int  prnstr( int , int , char * , int ) ;

int  menu( maxc , cch , ch)
int  maxc , cch ;
CHOICE ch[] ;
    {
    int  c , tc , k ;

    for( c = 0 ; c <= maxc ; c++ )
        prnstr( ch[c].row , ch[c].col , ch[c].str , NORMAL ) ;
    c = cch ;
    prnstr( ch[c].row , ch[c].col , ch[c].str , REVERSE ) ;
    for( ; ; )
        {
        k = getch() ;
        if( k == EXTKEY )
            {
            k = getch() ;
            tc = c ;
            switch( k )
                {
                case UARROW :
                    c = ( c > 0 ) ? --c : maxc ;
                    break ;
                case DARROW :
                    c = ( c < maxc ) ? ++c : 0 ;
                    break ;
                case PGUP :
                    c = 0 ;
                    break ;
                case PGDN :
                    c = maxc ;
                    break ;
                }
            if( c != tc )
                {
                prnstr( ch[tc].row , ch[tc].col , ch[tc].str , NORMAL ) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    prnstr( ch[c].row , ch[c].col , ch[c].str , REVERSE ) ;
                    }
                }
            else
                {
                if( ( k = toupper(k) ) == ENTER || k == ESCAPE )
                    {
                    if( k == ESCAPE )
                        c = ESCAPE ;
                    break ;
                    }
                else
                    {
                    for( tc = 0 ; tc <= maxc ; tc++ )
                        if( k == ch[tc].chr )
                            {
                            prnstr( ch[c].row , ch[c].col , ch[c].str , NORMAL ) ;
                            prnstr( ch[tc].row , ch[tc].col , ch[tc].str , REVERSE ) ;
                            c = tc ;
                            break ;
                            }
                    if( tc <= maxc )
                        break ;
                    }
                }
            }
        return( c ) ;
        }


    /*** Set screen ***/

    #include <stdio.h>
    #include "menu.h"
    #include "key.h"

    extern int  clear( int , int , int , int , int ) ;

          int  prnstr( int , int , char * , int ) ; /* print character string */
          int  box( int , int , int , int , int ) ; /* drawing box */
          int  *savescr( int , int , int , int ) ;  /* save screen */
          void restscr( int * ) ;                   /* restore screen */
          int  menu( int , int , CHOICE [] ) ;      /* pull-down menu */
          char *itoc( long , int ) ;  /* converts a long to a string */
          void settemp( int ) ;  /* set temperature */
          void sethumid( int ) ; /* set humidity    */

    #define NORMAL  0x07
    #define REVERSE 0x70
    #define BRIGHT  0x0F
    #define ENABLE  0
    #define TEMP    1
    #define HUMID   2

    CHOICE ch1[5] = { 11,31, '0' , " Unit #0 " ,
                      12,31, '1' , " Unit #1 " ,
                      13,31, '2' , " Unit #2 " ,
                      14,31, '3' , " Unit #3 " ,
                      15,31, 'E' , " Exit    " } ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CHOICE ch2[3] = { 12,41, 'E' , " Enable      " ,
                  13,41, 'T' , " Temperature " ,
                  14,41, 'H' , " Humidity    " } ;

static int result1 = 0 , result2 = 0 ;

int  set()
     {
     int  *scr_c1 , *scr_c2 , result ;

     scr_c1 = savescr( 9 , 29 , 17, 41 ) ;
     if( scr_c1 == NULL )
          return( 4 ) ;
     clear( 9 , 29 , 17 , 41 , NORMAL ) ;
     box( 9 , 29 , 17 , 41 , REVERSE ) ;
     for( ; ; )
          {
          result = menu( 4 , result1 , ch1 ) ;
          result1 = ( result != ESCAPE ) ? result : result1 ;
          if( result1 == 4 || result == ESCAPE )
               break ;
          else
               {
               scr_c2 = savescr( 10 , 39 , 16, 55 ) ;
               if( scr_c2 != NULL )
                    {
                    clear( 10 , 39 , 16 , 55 , NORMAL ) ;
                    box( 10 , 39 , 16 , 55 , REVERSE ) ;
                    for( ; ; )
                         {
                         result = menu( 2 , result2 , ch2 ) ;
                         result2 = ( result != ESCAPE ) ? result : result2 ;
                         if( result == ESCAPE )
                              break ;
                         switch( result )
                              {
                              case ENABLE :
                                   break ;
                              case TEMP :
                                   settemp( result1 ) ;
                                   break ;
                              case HUMID :
                                   sethumid( result1 ) ;
                                   break ;
                              }
                         }
                    restscr( scr_c2 ) ;
                    }
               }
          }
     restscr( scr_c1 ) ;
     return( result1 ) ;
     }


/*** Temperature and humidity setting ***/

#include  <stdio.h>
#include  "key.h"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define  NORMAL  0x07
#define  REVERSE 0x70
#define  BRIGHT  0x0F
#define  TRES    0.1
#define  HRES    1.0

extern int  temp_o( int )   ;       /* temperature offset    */
extern int  temp_s( int )   ;       /* temperature set point */
extern int  temp_ma( int )  ;       /* maximum temperature   */
extern int  temp_mi( int )  ;       /* minimum temperature   */
extern void stemp_s( int , int )  ; /* set (t) set point     */
extern int  humid_s( int )  ;       /* humidity set point    */
extern int  humid_ma( int ) ;       /* maximum humidity t    */
extern int  humid_mi( int ) ;       /* minimum humidity t    */
extern void shumid_s( int , int ) ; /* set (h) set point     */
extern int  clear( int , int , int , int , int ) ; /* clear screen */

      int  *savescr( int ,int , int , int ) ;
      void restscr( int * ) ;
      int  box( int , int , int , int ,  int ) ;
      int  prnstr( int , int , char * , int ) ;
      char *ftoc( float , int , int ) ;
      void scr2( int ) ;

void settemp( un )
int un ;
    {
    int   *scr_ct , key1 ;
    float toff , tset , tmax , tmin ;
    char  *tp ;

    scr_ct = savescr( 15,32,21,53 ) ;
    if( scr_ct == NULL )
        return ;
    toff = (float) temp_o( un )/2 ;
    tset = ((float) temp_s( un )/10 ) + toff ;
    tmax = ((float) temp_ma( un )/10 ) + toff ;
    tmin = ((float) temp_mi( un )/10 ) + toff ;
    clear( 15,32,21,53 , NORMAL ) ;
    box( 15,32,21,53 , REVERSE ) ;
    prnstr( 17,34, "Maximum   :" , NORMAL ) ;
    tp = ftoc( tmax ,6,2 ) ;
    prnstr( 17,46, tp , NORMAL ) ;
    prnstr( 18,34, "Set point :" , NORMAL ) ;
    tp = ftoc( tset ,6,2 ) ;
    prnstr( 18,46, tp , BRIGHT ) ;
    prnstr( 19,34, "Minimum   :" , NORMAL ) ;
    tp = ftoc( tmin ,6,2 ) ;
    prnstr( 19,46, tp , NORMAL ) ;
    for( ; ; )
        {
        key1 = getch() ;
        switch(key1)
            {
            case ENTER :
                tset -= toff ;
                tset *= 10 ;
                stemp_s( un , (int) tset ) ;
            case ESCAPE :
                restscr( scr_ct ) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                return ;
            case RARROW :
                if( tset <= tmax - TRES )
                    {
                    tset += TRES ;
                    tp = ftoc( tset ,6,2 ) ;
                    prnstr( 18,46, tp , BRIGHT ) ;
                    }
                break ;
            case LARROW :
                if( tset >= tmin + TRES )
                    {
                    tset -= TRES ;
                    tp = ftoc( tset ,6,2 ) ;
                    prnstr( 18,46, tp , BRIGHT ) ;
                    }
                break ;
            }
        }
    }

void sethumid( un )
int un ;
    {
    int   *scr_ch , key1 ;
    float hset , hmax , hmin ;
    char  *hp ;

    scr_ch = savescr( 15,32,21,53 ) ;
    if( scr_ch == NULL )
        return ;
    hset = (float) humid_s( un ) ;
    hmax = (float) humid_ma( un ) ;
    hmin = (float) humid_mi( un ) ;
    clear( 15,32,21,53 , NORMAL ) ;
    box( 15,32,21,53 , REVERSE ) ;
    prnstr( 17,34, "Maximum   :" , NORMAL ) ;
    hp = ftoc( hmax ,6,2 ) ;
    prnstr( 17,46, hp , NORMAL ) ;
    prnstr( 18,34, "Set point :" , NORMAL ) ;
    hp = ftoc( hset ,6,2 ) ;
    prnstr( 18,46, hp , BRIGHT ) ;
    prnstr( 19,34, "Minimum   :" , NORMAL ) ;
    hp = ftoc( hmin ,6,2 ) ;
    prnstr( 19,46, hp , NORMAL ) ;
    for( ; ; )
        {
        key1 = getch() ;
        switch(key1)
            {
            case ENTER :
                shumid_s( un , (int) hset ) ;
            case ESCAPE :
                restscr( scr_ch ) ;
                return ;
            case RARROW :
                if( hset <= hmax - HRES )
                    {
                    hset += HRES ;
                    hp = ftoc( hset ,6,2 ) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    prnstr( 18,46, hp , BRIGHT ) ;
                    }
                break ;
            case LARROW :
                if( hset >= hmin + HRES )
                    {
                    hset -= HRES ;
                    hp = ftoc( hset ,6,2 ) ;
                    prnstr( 18,46, hp , BRIGHT ) ;
                    }
                break ;
            }
        }
    }


/*** Draw box ***/

extern int  c_pos( int , int ) ;
extern void prnchr( int , int ) ;
extern void repl( int , int , int ) ;

#define NORMAL   0x07
#define REVERSE  0x70
#define UPLEFT   0xda
#define UPRIGHT  0xbf
#define LOLEFT   0xc0
#define LORIGHT  0xd9
#define HOR      0xc4
#define HOR_D    0xc2
#define HOR_U    0xc1
#define VER      0xb3
#define VER_L    0xc3
#define VER_R    0xb4
#define CENTER   0xc5
#define MAXROW   24
#define MAXCOL   79

int  box( ur , uc , lr , lc , at )
int  ur , uc , lr , lc , at ;
    {
    if( ur >= lr || uc >= lc )
        return( -1 ) ;
    if( lr > MAXROW || ur > MAXROW || lc > MAXCOL || uc > MAXCOL )
        return( -1 ) ;
    c_pos( ur , uc ) ;
    prnchr( UPLEFT , at ) ;
    if( lc - uc > 1 )
        {
        c_pos( ur , uc + 1 ) ;
        repl( HOR , lc - uc - 1 , at ) ;
        }
    c_pos( ur , lc ) ;
    prnchr( UPRIGHT , at ) ;
    if( lr - ur > 1 )
        {
        while( ++ur < lr )
            {
            c_pos( ur , uc ) ;
            prnchr( VER , at ) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                c_pos( ur , lc ) ;
                prnchr( VER , at ) ;
                }
            }
        c_pos( lr , uc ) ;
        prnchr( LOLEFT , at ) ;
        if( lc - uc > 1 )
            {
            c_pos( lr , uc + 1 ) ;
            repl( HOR , lc - uc - 1 , at ) ;
            }
        c_pos( lr , lc ) ;
        prnchr( LORIGHT , at ) ;
        return( 0 ) ;
        }


/*** Save and restore screen content ***/

#include  <stdio.h>
#include  <malloc.h>

#define  SCRSEG  0xB000
#define  NORMAL  0x07

extern int  readscr( unsigned int , unsigned int ) ;
extern void writscr( unsigned int , unsigned int , int ) ;
extern int  clear( int , int , int , int , int ) ;

int  *savescr( urow , ucol , lrow , lcol )
int  urow , ucol , lrow , lcol ;
    {
    int  *scrptr1 , *scrptr2 , buf , row , col ;

    buf = ( lrow - urow + 1 ) * ( lcol - ucol + 1 ) ;
    scrptr1 = scrptr2 = ( int * ) malloc( ( buf + 4 ) * sizeof( int ) ) ;
    if( scrptr1 != NULL )
        {
        *scrptr2 = urow ;
        scrptr2++ ;
        *scrptr2 = ucol ;
        scrptr2++ ;
        *scrptr2 = lrow ;
        scrptr2++ ;
        *scrptr2 = lcol ;
        scrptr2++ ;
        for( row = urow ; row <= lrow ; row++ )
            for( col = ucol ; col <= lcol ; col++ )
                {
                *scrptr2 = readscr( SCRSEG , ( row * 160 ) + ( col * 2 ) ) ;
                scrptr2++ ;
                }
        }
    return( scrptr1 ) ;
    }

void restscr( scrptr )
int  *scrptr ;
    {
    int  urow , ucol , lrow , lcol , row , col ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    urow = *(scrptr++) ;
    ucol = *(scrptr++) ;
    lrow = *(scrptr++) ;
    lcol = *(scrptr++) ;
    for( row = urow ; row <= lrow ; row++ )
        for( col = ucol ; col <= lcol ; col++ )
            writscr( SCRSEG , ( row * 160 ) + ( col * 2 ) , *(scrptr++) ) ;
    free( ( char * ) scrptr ) ;
    }


/*** Print string ***/

extern int  c_pos( int , int ) ;   /* set cursor position */
extern void prnchr( int , int ) ;  /* print a character with attribute */

#define MAXROW   24
#define MAXCOL   79

int  prnstr( r , c , ch , at )
int  r , c , at ;
char *ch ;
    {
    if( r > MAXROW || c > MAXCOL )
        return(-1) ;
    while( *ch != '\0' )
        {
        c_pos( r , c ) ;
        prnchr( *ch , at ) ;
        if( ++c > MAXCOL )
            if( ++r > MAXROW )
                return(-1) ;
        ch++ ;
        }
    return( 0 ) ;
    }


/*** Convert integer to string ***/

#include  <stdlib.h>

static char istr[50] ;

digit[10] = { '0','1','2','3','4','5','6','7','8','9' } ;

char *itoc( num , len )
long num ;
int  len ;
    {
    int  sign , i , j ;

    sign = ( num < 0 ) ? 1 : 0 ;
    num = ( sign == 1 ) ? -num : num ;
    istr[len] = '\0' ;
    for( i = len-1 ; i >= 0 ; i-- )
        istr[i] = ' ' ;
    istr[len-1] = '0' ;
    for( i = len-1 ; i >= sign && num > 0 ; i-- )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
        j = num % 10 ;
        num /= 10 ;
        istr[i] = digit[j] ;
        }
    if( sign == 1 )
        istr[i] = '-' ;
    if( num > 0 )
        for( i = len-1 ; i >= 0 ; i-- )
            istr[i] = '*' ;
    return( istr ) ;
    }


/*** Convert floating point to string ***/

#include  <stdlib.h>

static char nstr[50] ;

char *ftoc( num , len , pre )
float num ;
int   len , pre ;
    {
    int  sign , dec , i , j ;
    char *buf ;

    buf = fcvt( (double) num , pre , &dec , &sign ) ;
    nstr[i] = '\0' ;
    for( i = len-1 ; i >= 0 ; i-- )
        nstr[i] = ((dec>len-pre-1 && sign==0 ) || (dec>len-pre-2 && sign==1 )) ? '*' : ' ' ;
    nstr[len-pre-1] = '.' ;
    if( ( dec <= len-pre-1 && sign==0 ) || ( dec <= len-pre-2 && sign==1 ) )
        {
        if( dec == 0 )
            nstr[len-pre-2] = '0' ;
        else
            for( i = len-pre-2 , j = dec-1 ; j >= 0 ; j-- )
                nstr[i--]= *( buf + j ) ;
        if( sign == 1 )
            nstr[--i] = '-' ;
        for( i = len-pre , j = dec ; *( buf + j ) != '\0' ; j++ )
            nstr[i++] = *( buf + j ) ;
        }
    return( nstr ) ;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*** Screen function for C ***

DGROUP      group    _DATA

            public   _c_pos , _c_blink , _c_blank , _clear
            public   _repl , _prnchr

_TEXT       segment  byte public 'CODE'

            assume   cs:_TEXT , ds:DGROUP , ss: DGROUP , es:DGROUP

PAGEZERO    equ      0
C_STARTLINE equ      6
C_ENDLINE   equ      7
MAXROW      equ      24
MAXCOL      equ      79

_c_pos      proc     near
            push     bp                 ; use BP to access arguments
            mov      bp , sp
            push     bx                 ; save registers
            push     dx
            mov      dh , [bp+4]        ; read ROW argument
            mov      dl , [bp+6]        ; read COLUMN argument
            mov      bh , PAGEZERO      ; set current page to ZERO
            mov      ah , 2             ; function 2 set cursor position
            int      10h                ; call BIOS
            xor      ax , ax            ; return 0 if normal exit
            jmp      short cp_exit
cp_error:
            mov      ax , 0ffffh        ; return -1 if error exit
cp_exit:
            pop      dx                 ; restore registers
            pop      bx
            pop      bp
            ret
_c_pos      endp

_c_blink    proc     near               ; return nothing
            push     cx                 ; save register
            mov      ch , C_STARTLINE   ; cursor start line
            mov      cl , C_ENDLINE     ; cursor end line
            mov      ah , 1             ; function 1 set cursor type
            int      10h                ; call BIOS
            pop      cx                 ; restore register
            ret
_c_blink    endp

_c_blank    proc     near               ; return nothing
            push     cx                 ; save register
            mov      ch , 27h           ; bit 5 set indicate blanking
            mov      cl , 26h
            mov      ah , 1             ; function 1 set cursor type
            int      10h                ; call BIOS
            pop      cx                 ; restore register
            ret
_c_blank    endp

_prnchr     proc     near               ; return nothing
            push     bp                 ; use BP access argument
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            mov     bp , sp
            push    bx                  ; save registers
            push    cx
            mov     cx , 1              ; write 1 character
            mov     bh , PAGEZERO       ; set current page to ZERO
            mov     bl , [bp+6]         ; get ATTRIBUTE
            mov     al , [bp+4]         ; get CHARACTER
            mov     ah , 9              ; function 9 write char with att
            int     10h                 ; call BIOS
            pop     cx                  ; restore registers
            pop     bx
            pop     bp
            ret
_prnchr     endp

_repl       proc    near                ; return nothing
            push    bp                  ; use BP access argument
            mov     bp , sp
            push    bx                  ; save registers
            push    cx
            mov     cx , [bp+6]         ; write N character
            mov     bh , PAGEZERO       ; set current page to ZERO
            mov     bl , [bp+8]         ; get ATTRIBUTE
            mov     al , [bp+4]         ; get CHARACTER
            mov     ah , 9              ; function 9 write char with att
            int     10h                 ; call BIOS
            pop     cx                  ; restore registers
            pop     bx
            pop     bp
            ret
_repl       endp

_clear      proc    near
            push    bp                  ; use BP access argument
            mov     bp , sp
            push    bx                  ; save registers
            push    cx
            push    dx
            cmp     word ptr [bp+4] , MAXROW ; check validity
            jg      cl_error
            cmp     word ptr [bp+8] , MAXROW
            jg      cl_error
            cmp     word ptr [bp+6] , MAXCOL
            jg      cl_error
            cmp     word ptr [bp+10] , MAXCOL
            jg      cl_error
            mov     ch , [bp+4]         ; upper left ROW
            mov     cl , [bp+6]         ; upper left COLUMN
            mov     dh , [bp+8]         ; lower right ROW
            mov     dl , [bp+10]        ; lower right COLUMN
            mov     bh , [bp+12]        ; window ATTRIBUTE
            mov     al , 0              ; number of lines = 0 ( clear )
            mov     ah , 6              ; function 6 scroll up
            int     10h                 ; call BIOS
            xor     ax , ax             ; normal exit returns ZERO
            jmp     short cl_exit
cl_error:
            mov     ax , 0ffffh         ; error on exit returns -1
cl_exit:
            pop     dx                  ; restore registers
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            pop     cx
            pop     bx
            pop     bp
            ret
_clear      endp

_TEXT       ends

            end


;*** Read and write screen ram ***

            PUBLIC  _readscr , _writscr

_TEXT       SEGMENT BYTE PUBLIC 'CODE'
            ASSUME  CS:_TEXT

_readscr    proc    near
            push    bp
            mov     bp , sp
            push    bx
            push    ds
            mov     ax , [bp+4]  ; get screen segment
            mov     ds , ax
            mov     bx , [bp+6]  ; get screen offset
            mov     ax , [bx]    ; read character with attribute
            pop     ds
            pop     bx
            pop     bp
            ret
_readscr    endp

_writscr    proc    near
            push    bp
            mov     bp , sp
            push    bx
            push    ds
            mov     ax , [bp+4]  ; get screen segment
            mov     ds , ax
            mov     bx , [bp+6]  ; get screen offset
            mov     ax , [bp+8]  ; get character and attribute
            mov     [bx] , ax    ; write to screen
            pop     ds
            pop     bx
            pop     bp
            ret
_writscr    endp

_TEXT       ENDS

            END


;*** Initailize and release interrupt handler & reset output ***

DGROUP      group   _DATA

; 8259
PIC_MASK    equ     21h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PIC_EOI     equ     20h
EOI         equ     20h
INT_MASK    equ     04h
; 8255
PORT_B      equ     61h
SP_ON       equ     03h
SP_OFF      equ     not SP_ON
; CARD BASE ADDRESS
CARD_PORT   equ     82f3h
ID_PORT     equ     0e2f3h
ID_NO       equ     0aah
AD_PORT     equ     02f3h

            public  _init , _release
            extrn   int1C:far , int0A:far , unit_len:abs
            extrn   o_82f3:byte
            extrn   o_92f3:byte
            extrn   o_A2f3:byte
            extrn   o_B2f3:byte
            extrn   o_C2f3:byte

_TEXT       SEGMENT BYTE PUBLIC 'CODE'

            assume  cs:_TEXT , ds:DGROUP , ss: DGROUP , es:DGROUP

_init       proc    near
            push    bx
            push    cx
            push    dx
            push    ds
            push    es
in1:
            mov     dx , 82f3h
            mov     al , DGROUP:o_82f3
            out     dx , al
            add     dx , 1000h
            mov     al , DGROUP:o_92f3
            add     dx , 1000h
            out     dx , al
            mov     al , DGROUP:o_A2f3
            out     dx , al
            add     dx , 1000h
            mov     al , DGROUP:o_B2f3
            out     dx , al
            add     dx , 1000h
            mov     al , DGROUP:o_C2f3
            out     dx , al
            mov     ax , 351ch          ; function 35h get interrupt vector
            int     21h                 ; get vector 1C
            mov     DGROUP:int1C_off , bx ; save old address
            mov     DGROUP:int1C_seg , es
            mov     ax , 350ah
            int     21h                 ; get vector 0A
            mov     DGROUP:int0A_off , bx ; save old address
            mov     DGROUP:int0A_seg , es
            mov     ax , cs             ; get CODE SEGMENT
            mov     ds , ax
            mov     dx , offset int0A   ; get new int 0A address
            mov     ax , 250ah          ; function 25h set interrupt vector
            int     21h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
            mov     dx , offset int1C     ; get new int 1C address
            mov     ax , 251ch
            int     21h
            in      al , PIC_MASK
            and     al , not INT_MASK     ; enable int 0A
            out     PIC_MASK , al
            mov     ax , 0
in_exit:
            pop     es
            pop     ds
            pop     dx
            pop     cx
            pop     bx
            ret
_init       endp

_release    proc    near
            push    cx                    ; save registers
            push    dx
            push    ds
            in      al , PIC_MASK
            or      al , INT_MASK         ; disable interrupt 0A
            out     PIC_MASK , al
            mov     dx , AD_PORT
            in      al , dx
            in      al , PORT_B           ; read current value
            and     al , SP_OFF           ; speaker off
            out     PORT_B , al
            mov     dx , CARD_PORT        ; get base card port
            mov     cx , 5                ; for 5 consecutive ports
            mov     al , 0                ; reset all outputs
r1:
            out     dx , al
            add     dx , 1000h
            loop    r1
            mov     dx , DGROUP
            mov     ds , dx
            mov     dx , DGROUP:int1C_off  ; restore old service routines
            mov     ds , DGROUP:int1C_seg
            mov     ax , 251ch
            int     21h
            mov     dx , DGROUP
            mov     ds , dx
            mov     dx , DGROUP:int0A_off
            mov     ds , DGROUP:int0A_seg
            mov     ax , 250ah
            int     21h
            pop     ds                  ; restore registers
            pop     dx
            pop     cx
            ret
_release    endp

_TEXT       ENDS

_DATA       SEGMENT  WORD PUBLIC 'DATA'
int1C_off   dw       0      ; store 1C old service routine address
int1C_seg   dw       0
int0A_off   dw       0      ; store 0A old service routine address
int0A_seg   dw       0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
_DATA      ENDS

           END


;*** Handler for interrupt 1C and 0A ***

DGROUP     group    _DATA

           public   int1C , int0A
           public   o_82f3
           public   o_92f3
           public   o_A2f3
           public   o_B2f3
           public   o_C2f3
           extrn    unit_data:byte , unit_len:abs , humid_table:byte

_TEXT      SEGMENT  BYTE PUBLIC 'CODE'

           assume   cs:_TEXT , ds:DGROUP , ss: DGROUP , es:DGROUP

; * Define temperature base
TEMP_BASE  equ      160  ; 75 Farenheit
T_RANGE    equ      80   ; temperature range
TD_RANGE   equ      40   ; different temperature range
; * 8259
PIC_EOI    equ      20h
EOI        equ      20h
; * 8255
PORT_B     equ      61h
SP_ON      equ      03h
SP_OFF     equ      not SP_ON
; * Define enables
ON         equ      0h
OFF        equ      01h
; * Define turn status
LEFT       equ      00h
RIGHT      equ      01h
; * Define spray status
ACTIVE     equ      00h
INACTIVE   equ      01h
; * Define functions
INCUBATOR  equ      00h
HATCHER    equ      01h
; * Define actions
MAXACTION  equ      18
NOACTION   equ      00h
TEMP       equ      01h
HUMID      equ      02h
ERROR      equ      03h
; * Define units
MAXUNIT    equ      4
NO_0       equ      00h
NO_1       equ      01h
NO_2       equ      02h
NO_3       equ      03h
; * Define error flag and mask bits
LOW_TEMP   equ      01h        ; 1 if low temperature
HI_TEMP    equ      02h        ; 1 if high temperature
LOW_HUMID  equ      04h        ; 1 if low humidity
```

เอกสารนี้จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
HI_HUMID   equ      08h        ; 1 if high humidity
DOOR_OPEN  equ      10h        ; 1 if door opened
FAN_FAIL   equ      20h        ; 1 if fan fail
; * Define new error occured
OCCUR      equ      00h
NONE       equ      01h

DATA_STRUC  STRUC              ; data structure of each unit
 enable     db       ?        ; enable
 function   db       ?        ; function
                               ; * Temperature parameters
 temp_port  dw       ?        ; temperature port
 t_cur      db       ?        ; current temperature
 t_set      db       ?        ; temperature set point
 t_min      db       ?        ; minimum temperature
 t_max      db       ?        ; maximum temperature
                               ; * Humidity parameters
 humid_port dw       ?        ; humidity port
 h_cur      db       ?        ; actual humidity
 h_set      db       ?        ; humidity set point
 h_min      db       ?        ; minimum humidity
 h_max      db       ?        ; maximum humidity
 t_off      dw       ?        ; temperature offset
 h_base     dw       ?        ; humidity table base address
                               ; * Turn timer
 t_stat     db       ?        ; status
 t_ti_cnt   dw       ?        ; timer counter
 t_op_cnt   dw       ?        ; output counter
 t_ti_ti    dw       ?        ; timer time
 t_op_ti    dw       ?        ; output time
                               ; * Spray timer
 s_stat     db       ?        ; status
 s_ti_cnt   dw       ?        ; timer counter
 s_op_cnt   dw       ?        ; output counter
 s_ti_ti    dw       ?        ; timer time
 s_op_ti    dw       ?        ; output time
                               ; * Output ( port , mask )
 tl_port    dw       ?        ; turn left
 tl_mask    db       ?
 tr_port    dw       ?        ; turn right
 tr_mask    db       ?
 s_port     dw       ?        ; spray
 s_mask     db       ?
 h_port     dw       ?        ; heater
 h_mask     db       ?
                               ; * Alarm ( port , mask )
 lt_port    dw       ?        ; low temperature
 lt_mask    db       ?
 ht_port    dw       ?        ; high temperature
 ht_mask    db       ?
 lh_port    dw       ?        ; low humidity
 lh_mask    db       ?
 hh_port    dw       ?        ; high humidity
 hh_mask    db       ?
 do_port    dw       ?        ; doore opened
 do_mask    db       ?
 ff_port    dw       ?        ; fan failure
 ff_mask    db       ?
                               ; * Input ( port , mask )
 di_port    dw       ?        ; door
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
di_mask    db      ?
fi_port    dw      ?           ; fan
fi_mask    db      ?

err_flag   db      ?           ; error flag
err_mask   db      ?           ; error mask ( acknowledge )
DATA_STRUC ENDS

int1C      proc    far
           sti                                 ; enable interrupt
           push    ax                          ; save registers
           push    bx
           push    cx
           push    dx
           push    di
           push    si
           push    ds
           push    es
           mov     si , DGROUP                 ; get DGROUP segment address
           mov     ds , si
           mov     si , DGROUP:act_ctr         ; get action counter
           inc     si                          ; increase counter
           cmp     si , MAXACTION
           jb      ic1                         ; if equals maximum action
           xor     si , si                     ; reset counter
ic1:
           mov     DGROUP:act_ctr , si         ; save counter
           shl     si , 1                      ; multiply by 2
           mov     al , DGROUP:[si].action     ; get action
           cmp     al , TEMP                   ; check if temperature action
           jz      ic_temp                     ;       go temperature routine
           cmp     al , HUMID                  ; check if humidity action
           jnz     ic2
           jmp     ic_humid                    ;       go humidity routine
ic2:
           cmp     al , ERROR                  ; check if error action
           jnz     ic3
           jmp     ic_error                    ; go error routine
ic3:
           jmp     ic_exit                     ; else exit

ic_temp:
           xor     ch , ch
           mov     cl , DGROUP:[si+1].action   ; get unit number
           mov     bx , offset DGROUP:unit_data; get data base address
           and     cx , cx
           jz      ic_t2
ic_t1:
           add     bx , unit_len               ; add unit data length
           loop    ic_t1                       ; result data offset
ic_t2:
           cmp     byte ptr DGROUP:[bx].enable , OFF
           jnz     ic_t3
           jmp     ic_exit                     ; if disable then exit
ic_t3:
           mov     dx , DGROUP:[bx].di_port
           in      al , dx                     ; read door status
           and     al , DGROUP:[bx].di_mask
           jz      ic_t4                       ; 0 if open
           mov     dx , DGROUP:[bx].do_port
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mov     al , DGROUP:o_C2F3
        mov     ah , DGROUP:[bx].do_mask
        xor     ah , 0ffh
        and     al , ah
        mov     DGROUP:o_C2F3 , al
        out     dx , al
        mov     al , not DOOR_OPEN          ; door close
        and     DGROUP:[bx].err_flag , al   ; reset DOOR OPEN flag
        and     DGROUP:[bx].err_mask , al   ; reset DOOR MASK
        jmp     short ic_t5
ic_t4:
        mov     dx , DGROUP:[bx].do_port
        mov     al , DGROUP:o_C2F3
        or      al , DGROUP:[bx].do_mask
        mov     DGROUP:o_C2F3 , al
        out     dx , al
        mov     al , DOOR_OPEN              ; door open
        or      DGROUP:[bx].err_flag , al   ; set DOOR OPEN flag
ic_t5:
        mov     dx , DGROUP:[bx].fi_port
        in      al , dx                     ; read fan status
        and     al , DGROUP:[bx].fi_mask
        jz      ic_t6                       ; 0 if fail
        mov     dx , DGROUP:[bx].ff_port
        mov     al , DGROUP:o_C2F3
        mov     ah , DGROUP:[bx].ff_mask
        xor     ah , 0ffh
        and     al , ah
        mov     DGROUP:o_C2F3 , al
        out     dx , al
        mov     al , not FAN_FAIL           ; fan blow
        and     DGROUP:[bx].err_flag , al   ; reset FAN FAIL flag
        and     DGROUP:[bx].err_mask , al   ; reset FAN MASK
        jmp     short ic_t7
ic_t6:
        mov     dx , DGROUP:[bx].ff_port
        mov     al , DGROUP:o_C2F3
        or      al , DGROUP:[bx].ff_mask
        mov     DGROUP:o_C2F3 , al
        out     dx , al
        mov     al , FAN_FAIL               ; fan fail
        or      DGROUP:[bx].err_flag , al   ; set FAN FAIL flag
ic_t7:
        mov     dx , DGROUP:[bx].temp_port ; read current temperature
        out     dx , al
        jmp     ic_exit
ic_humid:
        xor     ch , ch
        mov     cl , DGROUP:[si+1].action    ; get unit number
        mov     bx , offset DGROUP:unit_data ; get data base address
        and     cx , cx
        jz      ic_h2
ic_h1:
        add     bx , unit_len               ; add unit data length
        loop    ic_h1                       ; result data offset
ic_h2:
        cmp     byte ptr DGROUP:[bx].enable , OFF
        jnz     ic_h3
        jmp     ic_exit                     ; if disable then exit
ic_h3:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        cmp     byte ptr DGROUP:[bx].function , INCUBATOR
        je      ic_h4                   ; jump if incubator
        jmp     ic_h8
ic_h4:
        cmp     word ptr DGROUP:[bx].t_ti_cnt , 0
        je      ic_h5
        dec     word ptr DGROUP:[bx].t_ti_cnt
        mov     dx , DGROUP:[bx].tr_port
        mov     al , DGROUP:o_82f3
        mov     ah , DGROUP:[bx].tr_mask
        xor     ah , 0ffh
        and     al , ah
        mov     ah , DGROUP:[bx].tl_mask
        xor     ah , 0ffh
        and     al , ah
        mov     DGROUP:o_82f3 , al
        out     dx , al
        jmp     ic_h8
ic_h5:
        cmp     byte ptr DGROUP:[bx].t_stat , LEFT
        jnz     ic_h6
        mov     byte ptr DGROUP:[bx].t_stat , RIGHT
        mov     dx , DGROUP:[bx].tr_port
        mov     al , DGROUP:o_82f3
        or      al , DGROUP:[bx].tr_mask
        mov     DGROUP:o_82f3 , al
        out     dx , al
        jmp     ic_h7
ic_h6:
        mov     byte ptr DGROUP:[bx].t_stat , LEFT
        mov     dx , DGROUP:[bx].tl_port
        mov     al , DGROUP:o_82f3
        or      al , DGROUP:[bx].tl_mask
        mov     DGROUP:o_82f3 , al
        out     dx , al
ic_h7:
        mov     ax , DGROUP:[bx].t_ti_ti
        mov     DGROUP:[bx].t_ti_cnt , ax
ic_h8:
        cmp     word ptr DGROUP:[bx].s_ti_cnt , 0
        je      ic_h11
        dec     word ptr DGROUP:[bx].s_ti_cnt
        cmp     word ptr DGROUP:[bx].s_op_cnt , 0
        je      ic_h9
        dec     word ptr DGROUP:[bx].s_op_cnt
        cmp     byte ptr DGROUP:[bx].s_stat , ACTIVE
        jne     ic_h9
        mov     dx , DGROUP:[bx].s_port
        mov     al , DGROUP:o_92f3
        or      al , DGROUP:[bx].s_mask
        jmp     short ic_h10
ic_h9:
        mov     dx , DGROUP:[bx].s_port
        mov     al , DGROUP:o_92f3
        mov     ah , DGROUP:[bx].s_mask
        xor     ah , 0ffh
        and     al , ah
ic_h10:
        mov     DGROUP:o_92f3 , al
        out     dx , al
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            jmp     ic_exit
ic_h11:
            mov     ax , DGROUP:[bx].s_ti_ti
            mov     DGROUP:[bx].s_ti_cnt , ax
            mov     ax , DGROUP:[bx].s_op_ti
            mov     DGROUP:[bx].s_op_cnt , ax
            mov     dx , DGROUP:[bx].humid_port
            out     dx , al
            jmp     ic_exit
ic_error:
            mov     DGROUP:err_occur , NONE
ic_exit:
            pop     es                          ; restore registers and exit
            pop     ds
            pop     si
            pop     di
            pop     dx
            pop     cx
            pop     bx
            pop     ax
            iret
int1C       endp

int0A       proc    far
            sti                                 ; enable interrupt again
            push    ax                          ; save registers
            push    bx
            push    cx
            push    dx
            push    si
            push    ds
            mov     ax , DGROUP                 ; get data segment
            mov     ds , ax
            mov     bx , DGROUP:act_ctr         ; get action counter
            shl     bl , 1                      ; multiply by 2
            xor     ch , ch
            mov     cl , DGROUP:[bx+1].action           ; get unit number
            cmp     byte ptr DGROUP:[bx].action , TEMP ; check action
            jz      ia_temp                     ; jump if temperature
            jmp     ia_humid
ia_1:
            jmp     ia_exit
ia_temp:
            mov     bx , offset DGROUP:unit_data
            and     cx , cx       ; test zero
            jz      ia_t2
           ia_t1:
            add     bx , unit_len
            loop    ia_t1
ia_t2:
            mov     dx , DGROUP:[bx].temp_port  ; read dry temperature
            in      al , dx
            push    ax
            mov     al , EOI                    ; acknowledge 8259A
            out     PIC_EOI , al
            pop     ax
            mov     DGROUP:[bx].t_cur , al      ; save current temperature
            xor     ah , ah
            mov     cl , DGROUP:[bx].t_set      ; get set point
            xor     ch , ch
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        cmp     ax , cx                 ; compare current & set point
        jl      ia_t3                   ; jump if temperature is lower
        mov     dx , DGROUP:[bx].h_port,
        mov     al , DGROUP:o_92f3      ; read current output
        mov     ah , DGROUP:[bx].h_mask
        xor     ah , 0ffh               ; turn heater off
        and     al , ah
        jmp     short ia_t4
ia_t3:
        mov     dx , DGROUP:[bx].h_port
        mov     al , DGROUP:o_92f3      ; read current output
        or      al , DGROUP:[bx].h_mask ; turn heater on
ia_t4:
        mov     DGROUP:o_92f3 , al
        out     dx , al
        mov     al , DGROUP:[bx].t_cur  ; get current temperature
        xor     ah , ah
        mov     cl , DGROUP:[bx].t_max  ; get maximum temperature
        xor     ch , ch
        cmp     ax , cx                 ; compare current & maximum
        jl      ia_t5
        mov     dx , DGROUP:[bx].ht_port
        mov     al , DGROUP:o_A2f3
        or      al , DGROUP:[bx].ht_mask
        mov     ah , DGROUP:[bx].lt_mask
        xor     ah , 0ffh
        and     al , ah
        mov     DGROUP:o_A2F3 , al
        out     dx , al
        or      byte ptr DGROUP:[bx].err_flag , HI_TEMP ; set flag
        jmp     ia_exit
ia_t5:
        mov     cl , DGROUP:[bx].t_min  ; get minimum temperature
        cmp     cx , ax                 ; compare current & minimum
        jl      ia_t6
        mov     dx , DGROUP:[bx].lt_port
        mov     al , DGROUP:o_A2f3
        or      al , DGROUP:[bx].lt_mask
        mov     ah , DGROUP:[bx].ht_mask
        xor     ah , 0ffh
        and     al , ah
        mov     DGROUP:o_A2F3 , al
        out     dx , al
        or      byte ptr DGROUP:[bx].err_flag , LOW_TEMP ; set flag
        jmp     ia_exit
ia_t6:          ; reset low and high error flag and mask
        mov     dx , DGROUP:[bx].ht_port
        mov     al , DGROUP:o_A2f3
        mov     ah , DGROUP:[bx].ht_mask
        xor     ah , 0ffh
        and     al , ah
        mov     ah , DGROUP:[bx].lt_mask
        xor     ah , 0ffh
        and     al , ah
        mov     DGROUP:o_A2F3 , al
        out     dx , al
        and     byte ptr DGROUP:[bx].err_flag , not HI_TEMP
        and     byte ptr DGROUP:[bx].err_mask , not HI_TEMP
        and     byte ptr DGROUP:[bx].err_flag , not LOW_TEMP
        and     byte ptr DGROUP:[bx].err_mask , not LOW_TEMP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            jmp     ia_exit
ia_humid:
            mov     bx , offset DGROUP:unit_data
            and     cx , cx       ; test zero
            jz      ia_h2
ia_h1:
            add     bx , unit_len
            loop    ia_h1
ia_h2:
            mov     dx , DGROUP:[bx].humid_port ; read wet temperature
            in      al , dx
            push    ax
            mov     al , EOI                    ; acknowledge 8259A
            out     PIC_EOI , al
            pop     ax
            push    ax
            mov     al , DGROUP:[bx].t_cur
            xor     ah , ah
            mov     cl , 5
            div     cl
            xor     ah , ah
            mov     cl , TD_RANGE
            mul     cl
            mov     si , ax
            pop     ax
            mov     cl , al
            xor     ch , ch
            mov     al , DGROUP:[bx].t_cur
            xor     ah , ah
            cmp     ax , cx
            jge     ia_h3
            mov     ax , cx
ia_h3:
            sub     ax , cx
            mov     cl , 5
            div     cl
            xor     ah , ah
            add     si , ax
            add     si , DGROUP:[bx].h_base
            mov     al , [si]
            mov     DGROUP:[bx].h_cur , al
            xor     ah , ah
            mov     cl , DGROUP:[bx].h_set
            xor     ch , ch                     ; get set point
            cmp     ax , cx
            jge     ia_h4                       ; jump if greater
            mov     byte ptr DGROUP:[bx].s_stat , ACTIVE
            jmp     short ia_h5
ia_h4:
            mov     byte ptr DGROUP:[bx].s_stat , INACTIVE
ia_h5:
            mov     cl , DGROUP:[bx].h_max      ; get maximum humidity
            xor     ch , ch
            cmp     ax , cx                     ; compare current & maximum
            jl      ia_h6
            mov     dx , DGROUP:[bx].hh_port
            mov     al , DGROUP:o_82F3
            or      al , DGROUP:[bx].hh_mask
            mov     ah , DGROUP:[bx].lh_mask
            xor     ah , 0ffh
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            and     al , ah
            mov     DGROUP:o_B2F3 , al
            out     dx , al
            or      byte ptr DGROUP:[bx] err_flag , HI_HUMID ; set flag
            jmp     ia_exit
ia_h6:
            mov     cl , DGROUP:[bx].h_min      ; get minimum humidity
            cmp     cx , ax                     ; compare current & minimum
            jl      ia_h7
            mov     dx , DGROUP:[bx].lh_port
            mov     al , DGROUP:o_B2F3
            or      al , DGROUP:[bx].lh_mask
            mov     ah , DGROUP:[bx].hh_mask
            xor     ah , 0ffh
            and     al , ah
            and     al , ah
            mov     DGROUP:o_B2F3 , al
            out     dx , al
            or      byte ptr DGROUP:[bx].err_flag , LOW_HUMID ; set flag
            jmp     ia_exit
ia_h7:              ; reset low and high error flag and mask
            mov     dx , DGROUP:[bx].hh_port
            mov     al , DGROUP:o_B2F3
            mov     ah , DGROUP:[bx].hh_mask
            xor     ah , 0ffh
            and     al , ah
            mov     ah , DGROUP:[bx].lh_mask
            xor     ah , 0ffh
            and     al , ah
            mov     DGROUP:o_B2F3 , al
            out     dx , al
            and     byte ptr DGROUP:[bx].err_flag , not HI_HUMID
            and     byte ptr DGROUP:[bx].err_mask , not HI_HUMID
            and     byte ptr DGROUP:[bx].err_flag , not LOW_HUMID
            and     byte ptr DGROUP:[bx].err_mask , not LOW_HUMID
ia_exit:
            pop     ds   ; restore registers and exit
            pop     si
            pop     dx
            pop     cx
            pop     bx
            pop     ax
            iret
int0A       endp

_TEXT       ENDS

_DATA       SEGMENT  WORD PUBLIC 'DATA'
o_82f3      db       0              ; output buffer
o_92f3      db       0
o_A2f3      db       0
o_B2f3      db       0
o_C2f3      db       0
err_occur   db       NONE           ; new error occured
act_ctr     dw       MAXACTION - 1 ; action counter.
unit_ctr    dw       0              ; unit counter
action      db       TEMP  , NO_0  ;
            db       HUMID , NO_0  ;
            db       TEMP  , NO_1  ;
            db       HUMID , NO_1  ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            db      TEMP  , NO_2  ;
            db      HUMID , NO_2  ;
            db      TEMP  , NO_3  ;
            db      HUMID , NO_3  ;
            db      ERROR , ?     ;
            db      NOACTION , ?  ;
            db      NOACTION , ?  ;
            db      NOACTION , ?  ;
            db      NOACTION , ?  ;
            db      NOACTION , ?  ;
            db      NOACTION , ?  ;
            db      NOACTION , ?  ;
            db      NOACTION , ?  ;
            db      NOACTION , ?  ;
_DATA       ENDS

            end


:*** Data and read data function ***

DGROUP      group   _DATA

public      _is_enble , _is_incub
public      unit_data , unit_len , _t_status
public      _temp_s , _temp_c , _temp_o , _temp_ma , _temp_mi , _stemp_s
public      _humid_s , _humid_c , _humid_ma , _humid_mi , _shumid_s
extrn       humid_table:byte

_TEXT       segment  byte public 'CODE'

            assume   cs:_TEXT , ds:DGROUP , ss: DGROUP , es:DGROUP

; * Define temperature base
TEMP_BASE   equ      160  ; 80 Farenheit
T_RANGE     equu     80   ; temperature range
TD_RANGE    equ      40   ; different temperature range

; * 8259
PIC_MASK    equ      21h
PIC_EOI     equ      20h
EOI         equ      20h
INT_MASK    equ      04h

; * Define enables
ON          equ      0h
OFF         equ      01h

; * Define turn status
LEFT        equ      00h
RIGHT       equ      01h

; * Define spray status
ACTIVE      equ      00h
INACTIVE    equ      01h

; * Define functions
INCUBATOR   equ      00h
HATCHER     equ      01h
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; * Define error flag and mask bits
LOW_TEMP   equ     01h  ; 1 if low temperature
HI_TEMP    equ     02h  ; 1 if high temperature
LOW_HUMID  equ     04h  ; 1 if low humidity
HI_HUMID   equ     08h  ; 1 if high humidity
DOOR_OPEN  equ     10h  ; 1 if door opened
FAN_FAIL   equ     20h  ; 1 if fan fail

DATA_STRUC  STRUC           ; data structure of each unit
 enable     db     ?        ; enable
 function   db     ?        ; function
                             ; * Temperature parameters
 temp_port  dw     ?        ; temperature port
 t_cur      db     ?        ; current temperature
 t_set      db     ?        ; temperature set point
 t_min      db     ?        ; minimum temperature
 t_max      db     ?        ; maximum temperature
                             ; * Humidity parameters
 humid_port dw     ?        ; humidity port
 h_cur      db     ?        ; actual humidity
 h_set      db     ?        ; humidity set point
 h_min      db     ?        ; minimum humidity
 h_max      db     ?        ; maximum humidity
 t_off      dw     ?        ; temperature offset
 h_base     dw     ?        ; humidity table base address
                             ; * Turn timer
 t_stat     db     ?        ; status
 t_ti_cnt   dw     ?        ; timer counter
 t_op_cnt   dw     ?        ; output counter
 t_ti_ti    dw     ?        ; timer time
 t_op_ti    dw     ?        ; output time
                             ; * Spray timer
 s_stat     db     ?        ; status
 s_ti_cnt   dw     ?        ; timer counter
 s_op_cnt   dw     ?        ; output counter
 s_ti_ti    dw     ?        ; timer time
 s_op_ti    dw     ?        ; output time
                             ; * Output ( port , mask )
 tl_port    dw     ?        ; turn left
 tl_mask    db     ?
 tr_port    dw     ?        ; turn right
 tr_mask    db     ?
 s_port     dw     ?        ; spray
 s_mask     db     ?
 h_port     dw     ?        ; heater
 h_mask     db     ?
                             ; * Alarm ( port , mask )
 lt_port    dw     ?        ; low temperature
 lt_mask    db     ?
 ht_port    dw     ?        ; high temperature
 ht_mask    db     ?
 lh_port    dw     ?        ; low humidity
 lh_mask    db     ?
 hh_port    dw     ?        ; high humidity
 hh_mask    db     ?
 do_port    dw     ?        ; doore opened
 do_mask    db     ?
 ff_port    dw     ?        ; fan failure
 ff_mask    db     ?
                             ; * Input ( port , mask )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  di_port    dw      ?          ; door
  di_mask    db      ?
  fi_port    dw      ?          ; fan
  fi_mask    db      ?
  err_flag   db      ?          ; error flag
  err_mask   db      ?          ; error mask ( acknowledge )
DATA_STRUC   ENDS

un_offset  proc    near
           mov     bx , offset DGROUP:unit_data
           mov     cx , [bp+4]   ; argument is UNIT NUMBER
           and     cx , cx
           jz      un2
un1:
           add     bx , unit_len ; calculate data offset
           loop    un1
un2:
           ret
un_offset  endp

_is_enble  proc    near
           push    bp
           mov     bp , sp
           push    bx
           push    cx
           call    un_offset
           mov     al , DGROUP:[bx].enable
           cmp     al , ON
           jnz     ie1
           xor     ax , ax       ; return ZERO if ON
           jmp     short ie_exit
ie1:
           mov     ax , 0ffffh   ; return NONZERO if OFF
ie_exit:
           pop     cx
           pop     bx
           pop     bp
           ret
_is_enble  endp

_is_incub  proc    near
           push    bp
           mov     bp , sp
           push    bx
           push    cx
           call    un_offset
           mov     al , DGROUP:[bx].function
           cmp     al , INCUBATOR
           jnz     is1
           xor     ax , ax       ; return ZERO if INCUBATOR
           jmp     short is_exit
is1:
           mov     ax , 0ffffh   ; return NONZERO if HATCHER
is_exit:
           pop     cx
           pop     bx
           pop     bp
           ret
_is_incub  endp
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
_temp_c    proc    near
           push    bp
           mov     bp , sp
           push    bx
           push    cx
           call    un_offset
           mov     al , DGROUP:[bx].t_cur
           xor     ah , ah       ; return current temperature (F) / 10
           pop     cx
           pop     bx
           pop     bp
           ret
_temp_c    endp

_temp_ma   proc    near
           push    bp
           mov     bp , sp
           push    bx
           push    cx
           call    un_offset
           mov     al , DGROUP:[bx].t_max
           xor     ah , ah       ; return maximum temperature (F) / 10
           pop     cx
           pop     bx
           pop     bp
           ret
_temp_ma   endp

_temp_mi   proc    near
           push    bp
           mov     bp , sp
           push    bx
           push    cx
           call    un_offset
           mov     al , DGROUP:[bx].t_min
           xor     ah , ah       ; return minimum temperature (F) / 10
           pop     cx
           pop     bx
           pop     bp
           ret
_temp_mi   endp

_temp_s    proc    near
           push    bp
           mov     bp , sp
           push    bx
           push    cx
           call    un_offset
           mov     al , DGROUP:[bx].t_set
           xor     ah , ah       ; return set point temperature (F) / 10
           pop     cx
           pop     bx
           pop     bp
           ret
_temp_s    endp

_stemp_s   proc    near
           push    bp
           mov     bp , sp
           push    bx
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            push    cx
            call    un_offset
            mov     ax , [bp+6]
            mov     DGROUP:[bx].t_set , al
            pop     cx
            pop     bx
            pop     bp
            ret
_stemp_s    endp

_temp_o     proc    near
            push    bp
            mov     bp , sp
            push    bx
            push    cx
            call    un_offset
            mov     ax , DGROUP:[bx].t_off ; return temperature (F) offset
            pop     cx
            pop     bx
            pop     bp
            ret
_temp_o     endp

_humid_c    proc    near
            push    bp
            mov     bp , sp
            push    bx
            push    cx
            call    un_offset
            mov     al , DGROUP:[bx].h_cur
            xor     ah , ah         ; return current humidity (%)
            pop     cx
            pop     bx
            pop     bp
            ret
_humid_c    endp

_humid_s    proc    near
            push    bp
            mov     bp , sp
            push    bx
            push    cx
            call    un_offset
            mov     al , DGROUP:[bx].h_set
            xor     ah , ah         ; return current humidity (%)
            pop     cx
            pop     bx
            pop     bp
            ret
_humid_s    endp

_shumid_s   proc    near
            push    bp
            mov     bp , sp
            push    bx
            push    cx
            call    un_offset
            mov     ax , [bp+6]
            mov     DGROUP:[bx].h_set , al
            pop     cx
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            pop     bx
            pop     bp
            ret
_shumid_s   endp

_humid_ma   proc    near
            push    bp
            mov     bp , sp
            push    bx
            push    cx
            call    un_offset
            mov     al , DGROUP:[bx].h_max
            xor     ah , ah         ; return maximum humidity (%)
            pop     cx
            pop     bx
            pop     bp
            ret
_humid_ma   endp

_humid_mi   proc    near
            push    bp
            mov     bp , sp
            push    bx
            push    cx
            call    un_offset
            mov     al , DGROUP:[bx].h_min
            xor     ah , ah         ; return minimum humidity (%)
            pop     cx
            pop     bx
            pop     bp
            ret
_humid_mi   endp

_t_status   proc    near
            push    bp
            mov     bp , sp
            push    bx
            push    cx
            call    un_offset
            mov     al , DGROUP:[bx].t_stat     ; return turn status
            xor     ah , ah
            pop     cx
            pop     bx
            pop     bp
            ret
_t_status   endp

_TEXT       ends

_DATA       segment  word public 'DATA'
; Unit#0
unit_data   db      ON              ; enable
            db      INCUBATOR       ; function
            dw      12f3h           ; temperature A/D port
            db      0               ; current temperature
            db      200             ; temperature set point
            db      160             ; minimum temperature ( 80 + 10 )
            db      220             ; maximum temperature ( 80 + 22 )
            dw      02f3h           ; humidity A/D port
            db      00h             ; current humidity
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            db      75              ; humidity set point
            db      70              ; minimum humidity
            db      80              ; maximum humidity
            dw      TEMP_BASE       ; base temperature
            dw      offset DGROUP:humid_table + ( ( TEMP_BASE - 140 ) * TD_RANGE )
            db      LEFT            ; turn status
            dw      0               ; counter
            dw      0
            dw      2700            ; turn timer
            dw      1
            db      INACTIVE
            dw      0               ; spray counter
            dw      0               ; output counter
            dw      30              ; spray timer
            dw      3               ; output duration
            dw      82f3h          ; turn left port
            db      01h             ; turn left mask
            dw      82f3h          ; turn right port
            db      10h             ; turn right mask
            dw      92f3h          ; spray port
            db      01h             ; spray mask
            dw      92f3h          ; heater port
            db      10h             ; heater mask
            dw      0a2f3h          ; low temperature port
            db      01h             ; low temperature mask
            dw      0a2f3h          ; high temperature port
            db      10h             ; high temperature mask
            dw      0b2f3h          ; low humidity port
            db      01h             ; low humidity mask
            dw      0b2f3h          ; high humidity port
            db      10h             ; high humidity mask
            dw      0c2f3h          ; door open port
            db      01h             ; door open mask
            dw      0c2f3h          ; fan failure port
            db      10h             ; fan failure mask
            dw      0d2f3h          ; door input port
            db      10h             ; door input mask
            dw      0d2f3h          ; fan input port
            db      01h             ; fan input mask
            db      00h             ; error flag
            db      0h              ; error mask
unit_len    equ     ( $ - unit_data )
; Unit#1
            db      ON
            db      HATCHER
            dw      52f3h
            db      00h
            db      180
            db      160
            db      200
            dw      42f3h
            db      00h
            db      85
            db      80
            db      90
            dw      TEMP_BASE
            dw      offset DGROUP:humid_table + ( ( TEMP_BASE - 140 )*TD_RANGE)
            db      LEFT
            dw      0
            dw      0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        dw      2700
        dw      1
        db      INACTIVE
        dw      7
        dw      3
        dw      30
        dw      3
        dw      082f3h
        db      02h
        dw      082f3h
        db      20h
        dw      092f3h
        db      02h
        dw      092f3h
        db      20h
        dw      0a2f3h
        db      02h
        dw      0a2f3h
        db      20h
        dw      0b2f3h
        db      02h
        dw      0b2f3h
        db      20h
        dw      0c2f3h
        db      02h
        dw      0c2f3h
        db      20h
        dw      0d2f3h
        db      20h
        dw      0d2f3h
        db      02h
        db      00h
        db      00h
: Unit#2
        db      ON
        db      INCUBATOR
        dw      32f3h
        db      0
        db      200
        db      160
        db      220
        dw      22f3h
        db      00h
        db      75
        db      70
        db      80
        dw      TEMP_BASE
        dw      offset DGROUP:humid_table + ( ( TEMP_BASE - 140 )*TD_RANGE)
        db      LEFT
        dw      900
        dw      1
        dw      2700
        dw      1
        db      INACTIVE
        dw      15
        dw      3
        dw      30
        dw      3
        dw      082f3h
        db      04h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                dw      082f3h
                db      40h
                dw      092f3h
                db      04h
                dw      092f3h
                db      40h
                dw      0a2f3h
                db      04h
                dw      0a2f3h
                db      40h
                dw      0b2f3h
                db      04h
                dw      0b2f3h
                db      40h
                dw      0c2f3h
                db      04h
                dw      0c2f3h
                db      40h
                dw      0d2f3h
                db      40h
                dw      0d2f3h
                db      04h
                db      00h
                db      0
; Unit#3
                db      ON
                db      HATCHER
                dw      72f3h
                db      00h
                db      180
                db      160
                db      200
                dw      62f3h
                db      00h
                db      85
                db      80
                db      90
                dw      TEMP_BASE
                dw      offset DGROUP:humid_table + ( ( TEMP_BASE - 140 )*TD_RANGE)
                db      LEFT
                dw      0
                dw      0
                dw      2700
                dw      10
                db      INACTIVE
                dw      22
                dw      3
                dw      30
                dw      3
                dw      082f3h
                db      08h
                dw      082f3h
                db      80h
                dw      092f3h
                db      08h
                dw      092f3h
                db      80h
                dw      0a2f3h
                db      08h
                dw      0a2f3h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            db      80h
            dw      0b2f3h
            db      08h
            dw      0b2f3h
            db      80h
            dw      0c2f3h
            db      08h
            dw      0c2f3h
            db      80h
            dw      0d2f3h
            db      80h
            dw      0d2f3h
            db      08h
            db      00h
            db      0
_DATA       ends

            end


:*** Relative humidity table ***

DGROUP      group   _DATA

            public  humid_table

T_RANGE     equ     80
TD_RANGE    equ     40

_DATA       segment word public 'DATA'
humid_table db  100 , 98 , 95 , 93 , 90 , 87 , 84 , 83 , 81 , 79 ; 70
            db   78 , 75 , 72 , 70 , 69 , 66 , 65 , 62 , 60 , 58
            db   56 , 52 , 51 , 50 , 48 , 45 , 44 , 42 , 41 , 38
            db   37 , 34 , 33 , 31 , 30 , 30 , 30 , 30 , 30 , 30
            db  100 , 98 , 95 , 93 , 90 , 89 , 85 , 84 , 81 , 79 ; 70.5
            db   78 , 75 , 72 , 70 , 69 , 66 , 65 , 62 , 60 , 59
            db   56 , 54 , 51 , 50 , 48 , 46 , 45 , 43 , 40 , 39
            db   36 , 34 , 33 , 32 , 30 , 30 , 30 , 30 , 30 , 30
            db  100 , 98 , 95 , 93 , 90 , 89 , 85 , 84 , 82 , 79 : 71
            db   78 , 75 , 72 , 70 , 69 , 67 , 65 , 62 , 60 , 59
            db   57 , 54 , 51 , 50 , 48 , 46 , 45 , 43 , 40 , 39
            db   37 , 35 , 33 , 32 , 31 , 31 , 31 , 31 , 31 , 31
            db  100 , 98 , 95 , 93 , 91 , 89 , 86 , 84 , 82 , 80 : 71.5
            db'  78 , 75 , 74 , 71 , 69 , 67 , 65 , 63 , 61 , 59
            db   57 , 55 , 53 , 51 , 49 , 48 , 46 , 44 , 42 , 40
            db   38 , 37 , 35 , 33 , 31 , 31 , 31 , 31 , 31 , 31
            db  100 , 98 , 95 , 93 , 90 , 89 , 86 , 84 , 82 , 80 ; 72
            db   78 , 76 , 74 , 71 , 69 , 67 , 65 , 63 , 61 , 60
            db   58 , 56 , 54 , 52 , 49 , 48 , 46 , 44 , 42 , 40
            db   38 , 37 , 35 , 33 , 31 , 31 , 31 , 31 , 31 , 31
            db  100 , 98 , 96 , 93 , 91 , 89 , 87 , 84 , 82 , 80 : 72.5
            db   78 , 76 , 74 , 71 , 69 , 68 , 65 , 63 , 61 , 60
            db   58 , 56 , 54 , 52 , 49 , 48 , 46 , 44 , 42 , 40
            db   39 , 38 , 35 , 33 , 31 , 31 , 31 , 31 , 31 , 31
            db  100 , 98 , 95 , 93 , 90 , 89 , 87 , 84 , 82 , 80 : 73
            db   78 , 76 , 74 , 71 , 70 , 68 , 65 , 63 , 61 , 60
            db   58 , 56 , 54 , 52 , 49 , 48 , 46 , 44 , 42 , 40
            db   39 , 38 , 35 , 33 , 31 , 31 , 31 , 31 , 31 , 31
            db  100 , 98 , 95 , 93 , 90 , 89 , 87 , 84 , 82 , 80 : 73.5
            db   78 , 76 , 74 , 71 , 70 , 68 , 65 , 63 , 61 , 60
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
db   58 , 56 , 54 , 52 , 49 , 48 , 46 , 44 , 42 , 40
db   39 , 38 , 35 , 33 , 31 , 31 , 31 , 31 , 31 , 31
db  100 , 98 , 95 , 93 , 90 , 89 , 87 , 84 , 82 , 80 ; 74
db   78 , 76 , 74 , 71 , 70 , 68 , 65 , 63 , 61 , 60
db   58 , 56 , 54 , 52 , 51 , 49 , 47 , 45 , 43 , 41
db   40 , 38 , 37 , 35 , 33 , 33 , 33 , 33 , 33 , 33
db  100 , 98 , 95 , 93 , 91 , 89 , 87 , 84 , 82 , 80 ; 74.5
db   78 , 76 , 74 , 71 , 70 , 68 , 65 , 63 , 61 , 60
db   58 , 56 , 54 , 52 , 51 , 50 , 47 , 46 , 43 , 42
db   40 , 38 , 37 , 35 , 33 , 33 , 33 , 33 , 33 , 33
db  100 , 98 , 95 , 93 , 91 , 89 , 87 , 84 , 82 , 80 ; 75
db   78 , 76 , 74 , 71 , 70 , 68 , 65 , 63 , 62 , 60
db   59 , 58 , 55 , 53 , 51 , 50 , 48 , 45 , 44 , 42
db   41 , 39 , 38 , 36 , 33 , 33 , 33 , 33 , 33 , 33
db  100 , 98 , 95 , 93 , 91 , 89 , 87 , 85 , 82 , 80 ; 75.5
db   78 , 76 , 75 , 72 , 70 , 68 , 67 , 65 , 63 , 60
db   59 , 58 , 55 , 53 , 51 , 50 , 48 , 46 , 44 , 43
db   41 , 40 , 38 , 36 , 33 , 33 , 33 , 33 , 33 , 33
db  100 , 98 , 95 , 93 , 91 , 89 , 87 , 85 , 83 , 80 ; 76
db   79 , 76 , 75 , 72 , 70 , 68 , 67 , 65 , 63 , 60
db   59 , 58 , 56 , 54 , 52 , 50 , 49 , 47 , 45 , 43
db   42 , 40 , 38 , 36 , 35 , 35 , 35 , 35 , 35 , 35
db  100 , 98 , 95 , 93 , 91 , 89 , 87 , 85 , 83 , 80 ; 76.5
db   79 , 77 , 75 , 73 , 70 , 69 , 67 , 65 , 63 , 61
db   60 , 58 , 56 , 54 , 52 , 51 , 49 , 47 , 45 , 44
db   42 , 41 , 39 , 37 , 35 , 35 , 35 , 35 , 35 , 35
db  100 , 98 , 95 , 93 , 91 , 89 , 87 , 85 , 83 , 80 ; 77
db   79 , 77 , 75 , 73 , 70 , 69 , 67 , 65 , 63 , 61
db   60 , 58 , 56 , 54 , 53 , 51 , 49 , 47 , 46 , 44
db   43 , 41 , 39 , 37 , 36 , 36 , 36 , 36 , 36 , 36
db  100 , 98 , 95 , 93 , 91 , 89 , 87 , 85 , 83 , 80 ; 77.5
db   79 , 77 , 75 , 73 , 71 , 69 , 67 , 66 , 63 , 61
db   60 , 58 , 56 , 55 , 53 , 51 , 50 , 48 , 46 , 44
db   43 , 41 , 39 , 38 , 36 , 36 , 36 , 36 , 36 , 36
db  100 , 98 , 95 , 93 , 91 , 89 , 87 , 85 , 83 , 80 ; 78
db   79 , 77 , 75 , 73 , 71 , 69 , 67 , 66 , 63 , 62
db   60 , 58 , 57 , 55 , 53 , 51 , 50 , 48 , 46 , 44
db   43 , 41 , 40 , 38 , 36 , 36 , 36 , 36 , 36 , 36
db  100 , 98 , 95 , 93 , 91 , 89 , 87 , 85 , 83 , 81 ; 78.5
db   79 , 77 , 75 , 73 , 71 , 70 , 68 , 66 , 63 , 62
db   60 , 58 , 57 , 55 , 53 , 51 , 50 , 48 , 47 , 45
db   43 , 42 , 40 , 39 , 37 , 37 , 37 , 37 , 37 , 37
db  100 , 98 , 95 , 93 , 91 , 89 , 87 , 85 , 83 , 81 ; 79
db   79 , 77 , 75 , 73 , 71 , 70 , 69 , 66 , 65 , 63
db   60 , 59 , 58 , 55 , 54 , 52 , 51 , 49 , 47 , 45
db   43 , 42 , 40 , 39 , 38 , 38 , 38 , 38 , 38 , 38
db  100 , 98 , 95 , 93 , 91 , 89 , 87 , 85 , 83 , 81 ; 79.5
db   79 , 77 , 75 , 73 , 71 , 70 , 69 , 66 , 65 , 63
db   60 , 59 , 58 , 55 , 54 , 52 , 51 , 49 , 47 , 46
db   44 , 42 , 41 , 40 , 38 , 38 , 38 , 38 , 38 , 38
db  100 , 98 , 95 , 93 , 91 , 89 , 87 , 85 , 83 , 81 ; 80
db   79 , 77 , 76 , 74 , 72 , 70 , 69 , 66 , 65 , 63
db   61 , 60 , 58 , 56 , 54 , 52 , 51 , 50 , 48 , 46
db   44 , 43 , 42 , 40 , 39 , 39 , 39 , 39 , 39 , 39
db  100 , 98 , 95 , 93 , 91 , 89 , 87 , 85 , 83 , 81 ; 80.5
db   79 , 78 , 76 , 74 , 72 , 70 , 69 , 66 , 65 , 63
db   61 , 60 , 58 , 56 , 54 , 53 , 51 , 50 , 48 , 46
db   44 , 43 , 42 , 40 , 39 , 39 , 39 , 39 , 39 , 39
db  100 , 98 , 95 , 93 , 91 , 89 , 87 , 85 , 83 , 81 ; 81
db   79 , 78 , 76 , 74 , 72 , 70 , 69 , 66 , 65 , 63
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น พร้อมทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
db   61 , 60 , 58 , 56 , 54 , 53 , 51 , 50 , 48 , 47
db   45 , 43 , 42 , 40 , 39 , 39 , 39 , 39 , 39 , 39
db  100 , 98 , 95 , 93 , 91 , 89 , 87 , 85 , 83 , 81 ; 81.5
db   79 , 78 , 76 , 74 , 72 , 70 , 69 , 67 , 65 , 63
db   61 , 60 , 58 , 56 , 55 , 53 , 51 , 50 , 48 , 47
db   45 , 43 , 42 , 40 , 39 , 39 , 39 , 39 , 39 , 39
db  100 , 98 , 96 , 94 , 91 , 90 , 88 , 85 , 83 , 81 ; 82
db   80 , 78 , 76 , 74 , 72 , 70 , 69 , 67 , 65 , 63
db   62 , 60 , 59 , 56 , 55 , 54 , 52 , 51 , 49 , 47
db   46 , 44 , 43 , 41 , 40 , 40 , 40 , 40 , 40 , 40
db  100 , 98 , 96 , 94 , 91 , 90 , 88 , 86 , 83 , 81 ; 82.5
db   80 , 78 , 76 , 74 , 72 , 70 , 69 , 67 , 65 , 64
db   62 , 60 , 59 , 57 , 55 , 54 , 52 , 51 , 49 , 48
db   46 , 44 , 43 , 41 , 40 , 40 , 40 , 40 , 40 , 40
db  100 , 98 , 96 , 94 , 92 , 90 , 88 , 86 , 84 , 82 ; 83
db   80 , 78 , 76 , 74 , 73 , 71 , 70 , 68 , 66 , 64
db   62 , 60 , 59 , 58 , 56 , 54 , 53 , 50 , 49 , 48
db   47 , 45 , 44 , 42 , 40 , 40 , 40 , 40 , 40 , 40
db  100 , 98 , 96 , 94 , 92 , 90 , 88 , 86 , 84 , 82 ; 83.5
db   80 , 78 , 76 , 75 , 73 , 71 , 70 , 68 , 67 , 65
db   63 , 61 , 59 , 58 , 56 , 55 , 53 , 51 , 49 , 48
db   47 , 46 , 44 , 43 , 41 , 41 , 41 , 41 , 41 , 41
db  100 , 98 , 96 , 94 , 92 , 90 , 88 , 86 , 84 , 82 ; 84
db   80 , 78 , 76 , 75 , 73 , 71 , 70 , 68 , 67 , 65
db   63 , 61 , 59 , 58 , 57 , 55 , 53 , 51 , 50 , 48
db   47 , 46 , 44 , 43 , 41 , 41 , 41 , 41 , 41 , 41
db  100 , 98 , 96 , 94 , 92 , 90 , 88 , 86 , 84 , 82 ; 84.5
db   80 , 78 , 77 , 75 , 73 , 71 , 70 , 68 , 67 , 65
db   63 , 61 , 59 , 58 , 57 , 55 , 53 , 52 , 50 , 49
db   47 , 46 , 44 , 43 , 41 , 41 , 41 , 41 , 41 , 41
db  100 , 98 , 96 , 94 , 92 , 90 , 88 , 86 , 84 , 82 ; 85
db   80 , 79 , 77 , 75 , 73 , 71 , 70 , 68 , 67 , 65
db   63 , 61 , 60 , 58 , 57 , 55 , 53 , 52 , 51 , 49
db   48 , 46 , 44 , 43 , 42 , 42 , 42 , 42 , 42 , 42
db  100 , 98 , 96 , 94 , 92 , 90 , 88 , 86 , 84 , 82 ; 85.5
db   80 , 79 , 78 , 75 , 73 , 71 , 70 , 68 , 67 , 65
db   63 , 61 , 60 , 59 , 57 , 55 , 53 , 52 , 51 , 49
db   48 , 46 , 45 , 44 , 42 , 42 , 42 , 42 , 42 , 42
db  100 , 98 , 96 , 94 , 92 , 90 , 88 , 86 , 85 , 82 ; 86
db   81 , 79 , 78 , 75 , 74 , 71 , 70 , 68 , 67 , 65
db   63 , 61 , 60 , 59 , 57 , 56 , 54 , 52 , 51 , 49
db   48 , 46 , 45 , 44 , 43 , 43 , 43 , 43 , 43 , 43
db  100 , 98 , 96 , 94 , 92 , 90 , 89 , 87 , 85 , 83 ; 86.5
db   81 , 79 , 78 , 76 , 74 , 72 , 70 , 69 , 67 , 65
db   63 , 62 , 60 , 59 , 58 , 56 , 54 , 53 , 51 , 50
db   48 , 46 , 45 , 44 , 43 , 43 , 43 , 43 , 43 , 43
db  100 , 98 , 96 , 94 , 92 , 90 , 89 , 87 , 85 , 83 ; 87
db   81 , 79 , 78 , 76 , 74 , 72 , 71 , 69 , 68 , 65
db   64 , 62 , 61 , 59 , 58 , 57 , 55 , 53 , 52 , 50
db   48 , 47 , 46 , 44 , 43 , 43 , 43 , 43 , 43 , 43
db  100 , 98 , 96 , 94 , 92 , 90 , 89 , 87 , 85 , 83 ; 87.5
db   81 , 79 , 78 , 76 , 74 , 72 , 71 , 69 , 68 , 66
db   64 , 62 , 61 , 59 , 58 , 57 , 55 , 53 , 52 , 50
db   48 , 47 , 46 , 44 , 43 , 43 , 43 , 43 , 43 , 43
db  100 , 98 , 95 , 94 , 93 , 90 , 88 , 86 , 84 , 82 ; 88
db   81 , 79 , 77 , 75 , 74 , 72 , 70 , 69 , 67 , 65
db   64 , 62 , 61 , 60 , 58 , 56 , 55 , 53 , 51 , 50
db   49 , 47 , 46 , 45 , 43 , 42 , 41 , 40 , 38 , 37
db  100 , 98 , 96 , 94 , 93 , 90 , 88 , 86 , 84 , 82 ; 88.5
db   81 , 79 , 77 , 75 , 74 , 72 , 70 , 69 , 67 , 65
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
db   64 , 62 , 61 , 60 , 58 , 56 , 55 , 53 , 52 , 51
db   49 , 47 , 46 , 45 , 43 , 42 , 41 , 40 , 38 , 37
db  100 , 98 , 96 , 94 , 93 , 90 , 88 , 86 , 85 , 83 ; 89
db   81 , 80 , 77 , 76 , 74 , 72 , 71 , 70 , 68 , 66
db   65 , 63 , 61 , 60 , 58 , 57 , 55 , 54 , 52 , 51
db   49 , 48 , 47 , 45 , 44 , 43 , 41 , 40 , 39 , 38
db  100 , 98 , 96 , 94 , 93 , 90 , 88 , 86 , 85 , 83 ; 89.5
db   81 , 80 , 77 , 76 , 74 , 72 , 71 , 70 , 68 , 66
db   65 , 63 , 62 , 60 , 58 , 57 , 56 , 54 , 52 , 51
db   50 , 48 , 47 , 46 , 45 , 43 , 42 , 40 , 39 , 38
db  100 , 98 , 96 , 94 , 93 , 90 , 89 , 87 , 85 , 83 ; 90
db   81 , 80 , 77 , 76 , 75 , 73 , 71 , 70 , 68 , 67
db   65 , 64 , 62 , 60 , 58 , 57 , 56 , 54 , 52 , 51
db   50 , 48 , 47 , 46 , 45 , 43 , 42 , 40 , 39 , 38
db  100 , 98 , 96 , 94 , 93 , 90 , 89 , 87 , 85 , 83 : 90.5
db   81 , 80 , 78 , 76 , 75 , 73 , 71 , 70 , 68 , 67
db   65 , 64 , 62 , 60 , 59 , 57 , 56 , 54 , 53 , 51
db   50 , 49 , 48 , 46 , 45 , 43 , 42 , 40 , 40 , 39
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 85 , 83 ; 91
db   81 , 80 , 78 , 76 , 75 , 73 , 71 , 70 , 68 , 67
db   65 , 64 , 62 , 61 , 59 , 58 , 56 , 55 , 53 , 52
db   50 , 49 , 48 , 46 , 45 , 44 , 42 , 40 , 40 , 39
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 85 , 83 : 91.5
db   81 , 80 , 78 , 76 , 75 , 73 , 71 , 70 , 68 , 67
db   65 , 64 , 62 , 61 , 60 , 58 , 56 , 55 , 54 , 52
db   51 , 49 , 48 , 47 , 46 , 44 , 43 , 41 , 40 , 39
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 85 , 83 : 92
db   81 , 80 , 78 , 76 , 75 , 73 , 72 , 70 , 69 , 67
db   66 , 64 , 62 , 61 , 60 , 58 , 56 , 55 , 54 , 52
db   51 , 50 , 48 , 47 , 46 , 44 , 43 , 42 , 40 , 39
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 85 , 83 ; 92.5
db   81 , 80 , 78 , 76 , 75 , 73 , 72 , 70 , 69 , 67
db   66 , 64 , 62 , 61 , 60 , 58 , 57 , 55 , 54 , 52
db   51 , 50 , 48 , 47 , 46 , 44 , 43 , 42 , 41 , 39
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 85 , 83 ; 93
db   82 , 80 , 78 , 77 , 75 , 73 , 72 , 70 , 69 , 68
db   66 , 64 , 63 , 61 , 60 , 58 , 57 , 55 , 54 , 53
db   51 , 50 , 49 , 48 , 46 , 45 , 43 , 42 , 41 , 40
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 85 , 83 ; 93.5
db   82 , 80 , 78 , 77 , 75 , 73 , 72 , 71 , 69 , 68
db   66 , 64 , 63 , 62 , 60 , 59 , 57 , 55 , 54 , 53
db   52 , 50 , 49 , 48 , 46 , 45 , 44 , 42 , 41 , 40
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 85 , 83 : 94
db   82 , 80 , 78 , 77 , 75 , 73 , 72 , 71 , 69 , 68
db   66 , 65 , 62 , 62 , 61 , 59 , 57 , 56 , 54 , 53
db   52 , 50 , 49 , 48 , 46 , 45 , 44 , 42 , 41 , 40
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 85 , 83 : 94.5
db   82 , 80 , 79 , 77 , 75 , 74 , 72 , 71 , 69 , 68
db   66 , 65 , 64 , 62 , 61 , 59 , 57 , 56 , 55 , 53
db   52 , 50 , 49 , 48 , 47 , 45 , 44 , 42 , 41 , 40
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 85 , 83 : 95
db   82 , 80 , 79 , 77 , 75 , 74 , 72 , 71 , 70 , 68
db   67 , 65 , 64 , 62 , 61 , 59 , 58 , 56 , 55 , 54
db   52 , 51 , 50 , 48 , 47 , 46 , 44 , 43 , 42 , 40
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 85 , 84 ; 95.5
db   82 , 80 , 79 , 77 , 75 , 74 , 72 , 71 , 70 , 68
db   67 , 65 , 64 , 62 , 61 , 59 , 58 , 56 , 55 , 54
db   52 , 51 , 50 , 48 , 47 , 46 , 44 , 43 , 42 , 41
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 86 , 85 : 96
db   83 , 81 , 79 , 78 , 76 , 74 , 73 , 71 , 70 , 68
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
db   67 , 65 , 64 , 62 , 61 , 60 , 59 , 57 , 55 , 54
db   52 , 51 , 50 , 49 , 47 , 46 , 45 , 44 , 42 , 41
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 86 , 85 ; 96.5
db   83 , 81 , 79 , 78 , 76 , 74 , 73 , 71 , 70 , 68
db   67 , 65 , 64 , 62 , 61 , 60 , 59 , 57 , 55 , 54
db   53 , 51 , 50 , 49 , 48 , 47 , 45 , 44 , 42 , 41
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 86 , 85 ; 97
db   83 , 81 , 80 , 78 , 76 , 75 , 73 , 71 , 70 , 68
db   67 , 65 , 64 , 62 , 61 , 60 , 59 , 57 , 56 , 54
db   53 , 52 , 50 , 49 , 48 , 47 , 45 , 44 , 42 , 41
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 86 , 85 ; 97.5
db   83 , 81 , 80 , 78 , 76 , 75 , 73 , 71 , 70 , 69
db   67 , 65 , 64 , 63 , 61 , 60 , 59 , 57 , 56 , 54
db   53 , 52 , 51 , 49 , 48 , 47 , 46 , 45 , 43 , 42
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 86 , 85 ; 98
db   83 , 81 , 80 , 78 , 76 , 75 , 73 , 71 , 70 , 69
db   67 , 66 , 64 , 63 , 62 , 60 , 59 , 57 , 56 , 55
db   54 , 52 , 51 , 50 , 49 , 47 , 46 , 45 , 43 , 42
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 86 , 85 ; 98.5
db   83 , 81 , 80 , 78 , 76 , 75 , 73 , 71 , 70 , 69
db   67 , 66 , 64 , 63 , 62 , 60 , 59 , 57 , 56 , 55
db   54 , 52 , 51 , 50 , 49 , 47 , 46 , 45 , 44 , 42
db  100 , 98 , 96 , 94 , 93 , 91 , 89 , 87 , 86 , 85 ; 99
db   83 , 81 , 80 , 78 , 76 , 75 , 73 , 71 , 70 , 69
db   67 , 66 , 65 , 63 , 62 , 60 , 59 , 58 , 57 , 55
db   54 , 53 , 51 , 50 , 49 , 47 , 46 , 45 , 44 , 43
db  100 , 98 , 97 , 95 , 93 , 91 , 89 , 87 , 86 , 85 ; 99.5
db   83 , 81 , 80 , 78 , 76 , 75 , 73 , 71 , 70 , 69
db   67 , 66 , 65 , 63 , 62 , 60 , 59 , 58 , 57 , 55
db   54 , 53 , 51 , 50 , 49 , 48 , 46 , 45 , 44 , 43
db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 87 , 86 , 85 ; 100
db   83 , 81 , 80 , 78 , 76 , 75 , 73 , 71 , 70 , 69
db   67 , 66 , 65 , 63 , 62 , 60 , 59 , 58 , 57 , 56
db   54 , 53 , 52 , 50 , 49 , 48 , 47 , 45 , 44 , 43
db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 87 , 86 , 85 ; 100.5
db   83 , 81 , 80 , 78 , 76 , 75 , 74 , 72 , 71 , 70
db   68 , 67 , 65 , 64 , 62 , 61 , 60 , 58 , 57 , 56
db   54 , 53 , 52 , 51 , 49 , 48 , 47 , 46 , 45 , 43
db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 87 , 86 , 85 ; 101
db   83 , 81 , 80 , 78 , 76 , 75 , 74 , 72 , 71 , 70
db   68 , 67 , 65 , 64 , 62 , 61 , 60 , 58 , 57 , 56
     db   55 , 53 , 52 , 51 , 49 , 48 , 47 , 46 , 45 , 43
db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 87 , 86 , 85 ; 101.5
db   83 , 82 , 81 , 79 , 77 , 76 , 74 , 73 , 71 , 70
db   68 , 67 , 65 , 64 , 63 , 61 , 60 , 59 , 57 , 56
db   55 , 54 , 52 , 51 , 50 , 48 , 47 , 46 , 45 , 44
db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 87 , 86 , 85 ; 102
db   83 , 83 , 81 , 79 , 77 , 76 , 74 , 73 , 71 , 70
db   68 , 67 , 65 , 64 , 63 , 61 , 60 , 59 , 57 , 56
db   55 , 54 , 52 , 51 , 50 , 48 , 47 , 46 , 45 , 44
db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 87 , 86 , 85 ; 102.5
db   83 , 82 , 81 , 79 , 77 , 76 , 75 , 73 , 71 , 70
db   68 , 67 , 65 , 64 , 63 , 61 , 60 , 59 , 58 , 56
db   55 , 54 , 53 , 51 , 50 , 49 , 48 , 46 , 45 , 44
db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 88 , 86 , 85 ; 103
db   83 , 82 , 81 , 79 , 77 , 76 , 75 , 73 , 71 , 70
db   69 , 67 , 66 , 64 , 63 , 61 , 60 , 59 , 58 , 56
db   56 , 54 , 53 , 51 , 50 , 49 , 48 , 46 , 45 , 44
db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 88 , 86 , 85 ; 103.5
db   83 , 82 , 81 , 79 , 77 , 76 , 75 , 73 , 71 , 70
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        db   69 , 67 , 66 , 64 , 63 , 61 , 60 , 59 , 58 , 57
        db   56 , 54 , 53 , 52 , 50 , 49 , 48 , 47 , 46 , 45
        db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 88 , 86 , 85 ; 104
        db   83 , 82 , 81 , 79 , 77 , 76 , 75 , 73 , 71 , 70
        db   69 , 67 , 66 , 64 , 63 , 61 , 60 , 59 , 58 , 57
        db   56 , 54 , 53 , 52 , 50 , 49 , 48 , 47 , 46 , 45
        db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 88 , 86 , 85 ; 104.5
        db   83 , 82 , 81 , 79 , 77 , 76 , 75 , 73 , 71 , 70
        db   69 , 67 , 66 , 64 , 63 , 61 , 60 , 59 , 58 , 57
        db   56 , 55 , 53 , 52 , 51 , 50 , 48 , 47 , 46 , 45
        db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 88 , 86 , 85 ; 105
        db   83 , 82 , 81 , 80 , 78 , 77 , 75 , 73 , 71 , 70
        db   69 , 67 , 66 , 64 , 63 , 61 , 60 , 59 , 58 , 57
        db   56 , 55 , 53 , 52 , 51 , 50 , 49 , 47 , 46 , 45
        db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 88 , 86 , 85 ; 105.5
        db   83 , 82 , 81 , 80 , 78 , 77 , 75 , 73 , 71 , 70
        db   69 , 67 , 66 , 64 , 63 , 61 , 60 , 59 , 58 , 57
        db   56 , 55 , 54 , 52 , 51 , 50 , 49 , 48 , 47 , 46
        db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 88 , 86 , 85 ; 106
        db   83 , 82 , 81 , 80 , 78 , 77 , 75 , 73 , 71 , 70
        db   69 , 67 , 66 , 64 , 63 , 61 , 60 , 59 , 58 , 57
        db   56 , 55 , 54 , 53 , 52 , 50 , 49 , 48 , 47 , 46
        db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 88 , 86 , 85 ; 106.5
        db   83 , 82 , 81 , 80 , 78 , 77 , 75 , 74 , 72 , 71
        db   70 , 68 , 67 , 65 , 64 , 62 , 61 , 59 , 58 , 57
        db   56 , 55 , 54 , 53 , 52 , 51 , 49 , 48 , 47 , 46
        db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 88 , 86 , 85 ; 107
        db   83 , 82 , 81 , 80 , 78 , 77 , 75 , 74 , 72 , 71
        db   70 , 68 , 67 , 65 , 64 , 62 , 61 , 59 , 58 , 57
        db   56 , 55 , 54 , 53 , 52 , 51 , 49 , 48 , 47 , 46
        db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 88 , 86 , 85 ; 107.5
        db   83 , 82 , 81 , 80 , 78 , 77 , 75 , 74 , 72 , 71
        db   70 , 68 , 67 , 65 , 64 , 62 , 61 , 60 , 53 , 57
        db   56 , 55 , 54 , 53 , 52 , 51 , 50 , 48 , 47 , 46
        db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 88 , 86 , 85 ; 108
        db   83 , 82 , 81 , 80 , 78 , 77 , 75 , 74 , 72 , 71
        db   70 , 68 , 67 , 65 , 64 , 62 , 61 , 60 , 59 , 57
        db   56 , 55 , 54 , 53 , 52 , 51 , 50 , 49 , 48 , 46
        db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 88 , 86 , 85 ; 108.5
        db   83 , 82 , 81 , 80 , 78 , 77 , 75 , 74 , 72 , 71
        db   71 , 68 , 67 , 65 , 64 , 62 , 61 , 60 , 59 , 58
        db   57 , 56 , 55 , 54 , 52 , 51 , 50 , 49 , 48 , 47
        db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 88 , 86 , 85 ; 109
        db   83 , 82 , 81 , 80 , 78 , 77 , 75 , 74 , 72 , 71
        db   70 , 68 , 67 , 65 , 64 , 62 , 61 , 60 , 59 , 58
        db   57 , 56 , 55 , 54 , 53 , 51 , 50 , 49 , 48 , 47
        db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 88 , 87 , 86 ; 109.5
        db   84 , 83 , 82 , 80 , 78 , 77 , 75 , 74 , 72 , 71
        db   70 , 68 , 67 , 65 , 64 , 62 , 61 , 60 , 59 , 58
        db   57 , 56 , 55 , 54 , 53 , 51 , 50 , 49 , 48 , 47
        db  100 , 98 , 97 , 95 , 94 , 92 , 90 , 88 , 87 , 86 ; 110
        db   84 , 83 , 82 , 80 , 78 , 77 , 75 , 74 , 72 , 71
        db   70 , 68 , 67 , 65 , 64 , 62 , 61 , 60 , 59 , 58
        db   57 , 56 , 55 , 54 , 53 , 51 , 50 , 49 , 48 , 47
_DATA   ends

        end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

งานออกแบบระบบควบคุมเครื่องฟักไข่สำเร็จได้ ผู้จัดทำขอขอบคุณ ดร.รัตติกร วรากูล ศิริพันธุ์ อาจารย์ที่ปรึกษา ซึ่งท่านได้กรุณาให้ความรู้และคำแนะนำซึ่งเป็นประโยชน์ต่อการทำงานเป็นอย่างยิ่ง คุณปรีชา สุดาทิศที่ได้ให้ความรู้เกี่ยวกับวิธีการฟักไข่ และอำนวยความสะดวกในด้านต่าง ๆ คุณสมศักดิ์ จันวันที่ได้ให้คำแนะนำปรึกษา นอกจากนี้ผู้จัดทำขอขอบคุณภาควิชาอิเลคทรอนิคส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ให้ความสะดวกในการใช้เครื่องมือทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. ชัยสวัสดิ์ เทียนวิบูลย์, การทำความเย็นและปรับอากาศ. กฤวิรรธน์, 415 หน้า, 2523

2. Lewis C. Eggebrecht, INTERFACE TO THE IBM PERSONAL COMPUTER. Howard W. Sams&co.,Inc, 246 p., 1983

3. IBM Corporation, Technical Reference Personal Computer XT System. IBM Computer Hardware, 512 p., 1983

4. Joseph J. Carr, Microcomputer Interfacing Handbook : A/D&D/A. TAB BOOKS Inc, 350 p., 1981

5. RAY DUNCAN, ADVANCE MS-DOS. Microsoft Press, 468 p., 1986

6. Microsoft Corporation,Microsoft Macro Assembler V4.0 for the MS-DOS operating System : User's guide. Microsoft press, 1985

7.Microsoft Corporation, Microsoft C Compiler V4.0 for the MS-DOS operating system : Code View and Language Reference. Microsoft press, 482 p., 1986

8.Microsoft Corporation, Microsoft C Compiler V4.0 for the MS-DOS operating system : Run-time Library Reference. Microsoft press, 420 p., 1986

9.Microsoft Corporation, Microsoft C Compiler V4.0 for the MS-DOS operating system : User's guide. Microsoft press, 444 p., 1986

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้