สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

แบบจำลองการควบคุมฟาร์มระบบอิเล็กทรอนิกส์ผ่านเครือข่ายอินเตอร์เน็ต

MODEL OF E-FARM CONTROL VIA THE INTERNET

T104098

โดย

นายวิทวัส เจริญทวีทรัพย์

นางสาววชิราภรณ์ จอมศิลป์

นางสาวเกศินี ธรรมอมรรัตนา

เลขหมู่.....
เลขทะเบียน 104098
วัน,เดือน,ปี 2 0 ต.ค. 2552

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODEL OF E-FARM CONTROL VIA THE INTERNET

BY

MR. WITTAWAT JARONETAWEESUP

MISS. WACHIRAPORN CHOMSIN

MISS. KESINEE THAMAMORNRATTANA

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF

THE REQUIRMENT FOR THE DEGREE OF

BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUIT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	แบบจำลองการควบคุมฟาร์มระบบอิเล็กทรอนิกส์ผ่านเครือข่ายอินเตอร์เน็ต
Thesis Title	Model of E-Farm control via the Internet
ชื่อนักศึกษา	นางสาวเกศินี ธรรมอมรรัตนา รหัสนักศึกษา 48012072
	นางสาววชิราภรณ์ จอมศิลป์ รหัสนักศึกษา 48012109
	นายวิทวัส เจริญทวีทรัพย์ รหัสนักศึกษา 48012116
อาจารย์ที่ปรึกษา	ร.ศ.อุทัย ศรีธีระวิโรจน์
อาจารย์ที่ปรึกษาร่วม	ผศ.ดร.สมเกียรติ อุดมหรรษากุล
ระดับการศึกษา	ปริญญาตรี ปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	ปีการศึกษา 2551

ปริญญานิพนธ์ฉบับนี้ได้รับการอนุมัติเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตร์บัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

(ร.ศ.อุทัย ศรีธีระวิโรจน์)
อาจารย์ผู้ควบคุมวิทยานิพนธ์

ผศ.ดร.สมเกียรติ อุดมหรรษากุล
อาจารย์ผู้ควบคุมวิทยานิพนธ์

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	แบบจำลองการควบคุมฟาร์มระบบอิเล็กทรอนิกส์ผ่านเครือข่ายอินเตอร์เน็ต
Thesis Title	Model of E-Farm control via the Internet
ชื่อนักศึกษา	นางสาวเกศินี ธรรมอมรรัตนา รหัสนักศึกษา 48012072
	นางสาววชิราภรณ์ จอมศิลป์ รหัสนักศึกษา 48012109
	นายวิทวัส เจริญทวีทรัพย์ รหัสนักศึกษา 48012116
อาจารย์ที่ปรึกษา	ร.ศ.อุทัย ศรีธีระวิโรจน์
อาจารย์ที่ปรึกษาร่วม	ผศ.ดร.สมเกียรติ อุดมหรรษากุล
ระดับการศึกษา	ปริญญาตรี ปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	ปีการศึกษา 2551

บทคัดย่อ

โดยทั่วไปการที่จะให้ผลผลิตตามเป้าหมายในการทำฟาร์มเลี้ยงไก่ จำเป็นจะต้องมีระบบการควบคุมอุณหภูมิ ความชื้น แสงสว่าง และการระบายอากาศที่มีประสิทธิภาพ ดังนั้น โครงงานนี้จึงนำเสนอการควบคุมฟาร์มเลี้ยงไก่ด้วยวงจรอิเล็กทรอนิกส์ผ่านทางอินเตอร์เน็ต โครงงานนี้สามารถลดจำนวนผู้ดูแลฟาร์มและเวลาในการดูแล ในระบบที่ออกแบบนี้สามารถตรวจจับอุณหภูมิ ความชื้น การเปิดปิดแสงสว่าง การเปิดปิดพัดลมดูดอากาศ การเปิดปิดอาหารและน้ำที่ถูกควบคุมด้วยคอมพิวเตอร์ผ่านทางอินเตอร์เน็ต โครงงานนี้ลดต้นทุนในการจ้างคน ง่ายต่อการปรับปรุง และรองรับธุรกิจที่มีแนวโน้มที่จะขยายตัวเพิ่มมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Model of E-Farm control via the Internet
Student	
	Miss. Kesinee Thamamornrattana ID.48012072
	Miss. Wachiraporn Chomsin ID.48012109
	Mr. Wittawat Jaronetaweesup ID.48012116
Adevisor	Asst. Prof. Uthai Sritheeravirojana
	Asst. Prof.Dr.Somkait Udomhunsakul
Geaduate	Bachelor Degree of Information Engineering
Departtment	Information Engineering
Academic Year	2008

ABSTACT

In general, in order to achieve desired goal of farm products, it is necessary to have a completely effective system to control temperature, humidity, light and ventilation. Therefore, this project proposes a model of E-Farm control via the Internet. It can reduce the amount of staffs and time to manage the farm. This system can detect and adjust temperature, moisture, light, ventilation and irrigation. In addition, the system is connected to the internet that lead to some advantages such as efficiently reduces man power, easy to enhance and adjust the system and capable of expansion correspond to the coming of new technology.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี ขอขอบคุณ ผศ.อุทัย ศรีธีระวิโรจน์ ผศ.ดร. สมเกียรติ อุดมหรรษากุล ที่ให้คำแนะนำและความช่วยเหลือที่ดีเสมอมาตลอด พร้อมทั้งชี้แนะแนวทางในการแก้ปัญหาต่างๆในการทำงานมาโดยตลอด

ขอกราบพระคุณ คุณพ่อ คุณแม่ที่คอยห่วงใยและให้การสนับสนุนในการศึกษา และพี่ๆทุกคนที่เป็นกำลังใจ พร้อมให้ความช่วยเหลือในด้านต่างๆมาโดยตลอด

สุดท้ายขอบคุณเพื่อนๆห้อง 4FS เพื่อนทั้งต่อเนื่อง ภาคปกติ และพี่บอย พี่อาร์ที่คอยช่วยเหลือกันมาตลอด

นางสาวเกศินี ธรรมอมรรัตนา

นางสาววชิราภรณ์ จอมศิลป์

นายวิทวัส เจริญทวีทรัพย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หัวข้อ	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
สารบัญ	
สารบัญรูป	
สารบัญตาราง	
บทที่ 1 บทนำ	
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงงาน	2
1.4 ข้อมูลในการออกแบบ	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	
2.1 เนื้อหาเกี่ยวกับการเลี้ยงไก่ของแบ่งออกเป็น 2 หน่วย	3
2.2 การจัดการฟาร์มเลี้ยงไก่เนื้อ	3
2.2.1 พื้นฐานความต้องการเทคโนโลยีของฟาร์มเลี้ยงไก่ระบบปิด	3
2.2.2 โรงเรือนและระบบการเลี้ยง	4
2.2.3 ประเภทการเลี้ยงไก่เนื้อ	5
2.3 แนวคิดการจัดการเลี้ยงดูไก่เนื้อ	6
2.4. การเลี้ยงไก่แบบระบบอีแว๊ป (Evaporative Cooling System)	7
2.4.1. ส่วนประกอบที่สำคัญ	8
2.5 ประโยชน์การเลี้ยงไก่ระบบปิด (อีแว๊ป) Evap	15
2.6 ความแตกต่างระหว่างรูปแบบฟาร์มไก่เนื้อแบบโรงเรือนปิดและปิด	19
2.6.1 รูปแบบฟาร์มไก่เนื้อแบบโรงเรือนปิด	19
2.6.2 รูปแบบฟาร์มไก่เนื้อแบบโรงเรือนเปิด	24
2.7 สถาปัตยกรรมของซีพียู ARM 7	28
2.7.1 ประวัติความเป็นมาของซีพียู ARM	28
2.7.2 สถาปัตยกรรมซีพียู ARM7	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หัวข้อ	หน้า
2.7.3 ระบบการทำงานของ ARM7	30
2.7.3.1 ไปป์ไลน์	31
2.7.3.2 รีจิสเตอร์	30
2.7.3.3 รีจิสเตอร์เก็บสถานการณ์ทำงาน	32
2.8 ออกแบบโปรแกรมในส่วนของ ARM7 โดยนำโปรแกรมควบคุมการทำงานมาประยุกต์ใช้ร่วมกัน	34
2.8.1 เลือกใช้โปรแกรม Keil uVision3	35
2.9 การควบคุมคอมพิวเตอร์ผ่านอินเตอร์เน็ตโดยการ remote access	39
2.9.1 การติดตั้งโปรแกรม NTRConnect	40
2.9.2 วิธีใช้งาน NTRConnect	44
บทที่ 3 ฮาร์ดแวร์	
3.1 อุปกรณ์ที่ใช้	46
3.2 คุณสมบัติของบอร์ด	47
3.3 โครงสร้างของบอร์ด CP-JR ARM7 LPC2368	50
3.4 ไอซีตรวจสอบอุณหภูมิ (Temperature Sensors IC. :DS-1820)	54
3.4.1 ระบบการสื่อสารข้อมูลอนุกรมแบบหนึ่งสาย (1-Wire™ Serial Bus)	54
3.4.2 คุณสมบัติทางเทคนิคของระบบบัสหนึ่งสาย	54
3.4.3 รูปแบบการสื่อสารข้อมูลแบบหนึ่งสาย (1-Wire™ Serial Bus)	55
3.4.4 คุณสมบัติของไทม์สล็อต	56
3.4.5 คุณสมบัติของไอซีตรวจสอบอุณหภูมิ DS-1820	57
3.5 การใช้งานวงจรขับ LED แสดงผล	58
3.6 ภาคตรวจสอบอุณหภูมิ	59
บทที่4 การออกแบบ	
4.1 ภาพโดยรวมของระบบ	60
4.2 การออกแบบ	61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หัวข้อ	หน้า
4.3 การออกแบบการทำงานของตัวควบคุมอุณหภูมิ	62
4.3.1 Flow chart การทำงานของตัวควบคุมอุณหภูมิ	64
บทที่ 5 ผลการทดลอง	
5.1 การทดลองตรวจจับอุณหภูมิ	63
5.2 การทดลองด้านโปรแกรม (Software)	66
5.3 การทดลองการกดสวิตซ์ที่ตัวเครื่อง	67
บทที่ 6 บทสรุปและวิจารณ์	
6.1 สรุปผลการดำเนินงาน	69
6.2 ปัญหาที่เกิดขึ้นและแนวทางแก้ไข	69
6.3 แนวทางพัฒนาในอนาคต	69
บรรณานุกรม	70
ภาคผนวก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูป	หน้า
รูปที่ 2.1แผนผังองค์ประกอบหลักฟาร์มไก่เนื้อ แบบโรงเรือนปิด	19
รูปที่ 2.2โรงเรือนไก่เนื้อ แบบปิดด้านหน้า	20
รูปที่ 2.3โรงเรือนไก่เนื้อ แบบปิดด้านหลัง	21
รูปที่ 2.4 รูปด้านข้างโรงเรือนไก่เนื้อแบบปิด	22
รูปที่ 2.5 แผนผังองค์ประกอบหลักฟาร์มไก่เนื้อ แบบโรงเรือนเปิด	24
รูปที่ 2.6โรงเรือนไก่เนื้อ แบบเปิดด้านหน้า	25
รูปที่ 2.7โรงเรือนไก่เนื้อ แบบเปิดด้านหลัง	26
รูปที่ 2.8รูปด้านข้างโรงเรือนไก่เนื้อแบบเปิด	27
รูปที่ 2.9 ไปป์ไลน์ของ ARM7 ทั้ง 3 ขั้นตอน	30
รูปที่ 2.10 ความสัมพันธ์ระหว่างหน่วยความจำ รีจิสเตอร์และหน่วยประมวลผลของARM7	32
รูปที่ 2.11 ความหมายของบิตในรีจิสเตอร์สถานการณ์ทำงาน	33
รูปที่ 2.12 รูป Microcontroller ARM7 LPC 2148	34
รูปที่ 2.13 รูปแบบโปรแกรม Keil uVision3	35
รูปที่ 2.14 การกำหนดค่าตัวเลือกการแปลคำสั่งของโปรแกรม Keil uVision3	36
รูปที่ 2.15 การสร้างโปรเจคใหม่	37
รูปที่ 2.16 การกำหนด เบอร์ MCU ให้เป็น LPC2368 ของ Philips	37
รูปที่ 2.17 ขั้นตอนการก๊อปไฟล์ Startup ของ Keil	38
รูปที่ 2.18 กำหนดค่า X-TAL ให้กับโปรแกรม Keil	39
รูปที่ 2.19 การกำหนด Create Hex File ให้กับโปรแกรม Keil เท่านี้ก็จะสามารถเริ่มต้นเขียนโปรแกรม ใน ARM7 ได้	39
รูปที่ 2.20 การติดตั้งโปรแกรม	40
รูปที่ 2.21 การติดตั้งโปรแกรม	40
รูปที่ 2.22 การติดตั้งโปรแกรม	41
รูปที่ 2.23 การติดตั้งโปรแกรม	41
รูปที่ 2.24 การติดตั้งโปรแกรม	42
รูปที่ 2.25 การติดตั้งโปรแกรม	42
รูปที่ 2.26 การติดตั้งโปรแกรม	42
รูปที่ 2.27 การติดตั้งโปรแกรม	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

รูป	หน้า
รูปที่ 2.28 การติดตั้งโปรแกรม	43
รูปที่ 2.29 การติดตั้งโปรแกรม	44
รูปที่ 2.30 การใช้โปรแกรม NRT CONNECT	44
รูปที่ 2.30 การใช้โปรแกรม NRT CONNECT	44
รูปที่ 2.30 การใช้โปรแกรม NRT CONNECT	45
รูปที่ 3.1 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ ARM7 เบอร์ LPC2368	49
รูปที่ 3.2 โครงสร้างบอร์ด CP-JR ARM7 LPC2368	50
รูปที่ 3.3 แสดงตำแหน่งของอุปกรณ์ต่างๆในบอร์ด CP-JR ARM7 LPC2368	50
รูปที่ 3.4 แสดงวงจรไมโครคอนโทรลเลอร์	53
รูปที่ 3.5 แสดงลักษณะไอซีตรวจสอบอุณหภูมิ DS-1820	58
รูปที่ 3.6 วงจรรีเลย์	58
รูปที่ 3.7 วงจรประมวลผลอุณหภูมิ	59
รูปที่ 3.8 วงจรรีเลย์สวิตซ์	59
รูปที่ 4.1 Diagram แสดงภาพโดยรวมของโครงงาน	60
รูปที่ 4.2 Diagram ของระบบควบคุมอุณหภูมิ	61
รูปที่ 4.3 Flow Chartแสดงการทำงานของตัวควบคุมอุณหภูมิ	62
รูปที่ 5.1 รูปแสดงแบบจำลองฟาร์มเลี้ยงไก่	63
รูปที่ 5.2 แสดงการทำงานของไฟเมื่ออุณหภูมิต่ำกว่า 26 องศาเซลเซียส	65
รูปที่ 5.3 การทดลองทางด้านโปรแกรม	66
รูปที่ 5.4 การทดลองการกดสวิตซ์ที่ตัวเครื่อง	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 อาหารและคุณภาพอาหารไก่เนื้อ	12
ตารางที่ 2.2 ตารางมาตรฐานอาหาร	13
ตารางที่ 5.1 การตรวจวัดอุณหภูมิ	64
ตารางที่ 5.2 การทดลองปุ่มต่างๆทางด้านโปรแกรม (Software)	67
ตารางที่ 5.3 แสดงสวิตซ์ควบคุมที่ตัวเครื่องไปยังหลอดไฟ LED	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมา

ประเทศไทยเป็นประเทศที่มีการผลิตและส่งออกสินค้าการเกษตรเป็นรายได้หลักของประเทศ การเลี้ยงสัตว์ในประเทศได้มีการขยายตัวและพัฒนาเป็นอุตสาหกรรมการเกษตรขนาดใหญ่โดยรายได้ส่วนใหญ่ของประเทศ จึงได้จากการแปรรูปสินค้าอุตสาหกรรมการเกษตรและปศุสัตว์ ซึ่งเงื่อนไขการตลาดการค้าเสรีขององค์การค้าโลก(WTO) และกลุ่มประเทศยุโรป(E.U.) การเลี้ยงไก่เพื่อการบริโภคเนื้อ

ปัจจุบันการเลี้ยงไก่เนื้อเป็นธุรกิจสัตว์เลี้ยงที่สำคัญยิ่งของประเทศ เพื่อป้อนตลาดทั้งในและต่างประเทศ มีการพัฒนาการเลี้ยงหลากหลาย ทั้งผู้เลี้ยงรายย่อยที่รับจ้างเลี้ยงจากบริษัททำธุรกิจไก่โดยตรง โดยบริษัทเป็นผู้ประกันราคาขายให้ส่วนผู้เลี้ยงรายใหญ่ คือ บริษัทต่างๆที่ทำธุรกิจไก่และอาหารไก่คนไทยบริโภคเนื้อ ประมาณ 400-600 ล้านตัวต่อปี และส่งออกขายต่างประเทศมากกว่าปีละ 5800 ล้านตัว ไก่เป็นเนื้อสัตว์ที่ส่งออกขายต่างประเทศมากที่สุด จึงเป็นธุรกิจเกษตรที่มีความสำคัญยิ่งที่ผู้เลี้ยงควรศึกษาหาข้อมูลทั้งด้านการผลิต การตลาด การจัดการ การสุขาภิบาล (ลิขิต เอียดแก้ว, 2543: 78)

ทั้งนี้ทั้งนั้นการจัดการเลี้ยงไก่เนื้อมีจุดมุ่งหมายให้ไก่มีสุขภาพที่ดี มีต้นทุนในการเลี้ยงต่ำโตเร็ว ได้ขนาดตามที่ตลาดต้องการใช้ระยะเวลาในการเลี้ยงสั้นที่สุดและให้อาหารได้อย่างมีประสิทธิภาพฉะนั้นเกษตรกรผู้เลี้ยงไก่เนื้ออิสระจึงจำเป็นต้องมีความรู้ความเข้าใจและยอมรับเทคโนโลยีด้านการจัดการในการเลี้ยงไก่เนื้อ “การยอมรับเทคโนโลยีการเลี้ยงไก่เนื้อ” หมายถึงการที่เกษตรกรนำเอาเทคโนโลยีการเลี้ยงไก่เนื้อด้านทำเลที่ตั้ง โรงเรือน พันธุ์ อาหาร การจัดการและการสุขาภิบาลนำไปใช้ในการปฏิบัติจริงในการจัดการ

1.2 วัตถุประสงค์

1 เพื่อศึกษาการเขียนโปรแกรมของไมโครคอนโทรลเลอร์ ARM7TDMI-S Core

2 เพื่อศึกษาการควบคุมผ่านคอมพิวเตอร์และผ่านอินเตอร์เน็ต

3.เพื่อประยุกต์การใช้งานของคอมพิวเตอร์ให้สามารถใช้งานร่วมกับไมโครคอนโทรลเลอร์

4.ทำการศึกษางานวิจัยการจัดการฟาร์มเลี้ยงไก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงงาน

เป็นการนำไมโครคอนโทรเลอร์มาใช้ในการควบคุมหลอดไฟฟ้า พัดลมแสดงสถานะแทนการให้น้ำและอาหารของไก่ แสดงอุณหภูมิสมารถควบคุมอุณหภูมิโดยการเปิด-ปิดพัดลม และเขียนโปรแกรมสามารถควบคุมระบบคอมพิวเตอร์ไปควบคุมหลอดไฟฟ้า และ พัดลม และยังสามารถรับค่าอุณหภูมิมาแสดงค่าที่จอคอมพิวเตอร์ได้ โดยการจำลองรูปแบบโรงเลี้ยงไก่เป็นแบบจำลอง

1.4 ข้อมูลในการออกแบบ

- สามารถแสดงผลอุณหภูมิและควบคุมระบบให้มีอุณหภูมิลดลงได้ในระดับหนึ่ง
- สามารถควบคุม หลอดไฟ หลอด LED พัดลมที่แบบจำลองได้ จากการควบคุมที่ไมโครคอนโทรลเลอร์
- สามารถ ควบคุม หลอดไฟ หลอด LED พัดลมที่แบบจำลองได้ จากการควบคุมด้วยระบบคอมพิวเตอร์ได้

ในแบบจำลองโรงเรือนจะจำลองสถานะของอาหาร สถานะของน้ำ ด้วยหลอด LED พัดลมใช้พัดลม DC 12 Vเพื่อแสดงการระบายอากาศ หลอดไฟที่ใช้จำลองเพื่อเพิ่มอุณหภูมิจะเป็นหลอดไฟ DC 12 V เพื่อจะให้สามารถนำไปประยุกต์การใช้งานจริงได้ และเห็นการทำงานในแบบจำลองได้ชัดเจนมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 เนื้อหาเกี่ยวกับการเลี้ยงไก่ของแบ่งออกเป็น 2 หน่วย

1. การจัดการฟาร์มเลี้ยงไก่เนื้อ หมายถึง การกำหนดวิธีการปฏิบัติด้านการจัดการฟาร์มเลี้ยงไก่เนื้อ การจัดการด้านสุขภาพสัตว์ และการจัดการด้านสิ่งแวดล้อม เพื่อให้ได้ไก่เนื้อที่ถูกสุขลักษณะและเหมาะสมแก่ผู้บริโภค

2. ไมโครคอนโทรลเลอร์ ARM7TDMI-S Core ที่ใช้ควบคุมด้วยคอมพิวเตอร์ผ่านอินเตอร์เน็ตเพื่อฟาร์มเลี้ยงไก่

2.2 การจัดการฟาร์มเลี้ยงไก่เนื้อ

2.2.1 พื้นฐานความต้องการเทคโนโลยีของฟาร์มเลี้ยงไก่ระบบปิด

ปัจจุบันนิยมเลี้ยงไก่ลูกผสมซึ่งได้รับการปรับปรุงให้มีประสิทธิภาพการผลิตที่ดีขึ้นคือ โตเร็ว น้ำหนักสูงขึ้น ระยะเวลาการเลี้ยงสั้น น้ำหนักตัวสูงส่งตลาดได้เพิ่มขึ้นถึง 30% ในขณะที่อายุของไก่เนื้อลดลงกว่า 27% ซึ่งเป็นผลการคัดเลือกทางพันธุกรรมอย่างต่อเนื่อง ทำให้ปัจจุบันการเลี้ยงไก่เนื้อใช้ระยะเวลาเพียง 42-45 วัน ได้น้ำหนักส่งตลาด 18-20 กิโลกรัม และมีประสิทธิภาพการเปลี่ยนอาหารเพียง 2.0-2.1 ทั้งนี้ต้องขึ้นอยู่กับการจัดการที่ดีด้วย

เทคโนโลยี คือ Know-Now เป็นความรู้ที่ว่าจะทำอย่างไรเป็นความรู้ที่จะนำไปใช้ในทางปฏิบัติ สมิทธ์ คำเพิ่มพูล (2529 : 34-42) ได้กล่าวเกี่ยวกับเทคโนโลยีกับการเกษตรกรรมว่าผลผลิตทางการเกษตรไม่ว่าจะเป็นพืชหรือสัตว์ย่อมต้องอาศัยวิทยาศาสตร์และเทคโนโลยีในการเพิ่มผลผลิตให้ได้ปริมาณต่อหน่วยพื้นที่มากขึ้น และเป็นผลผลิตที่มีคุณภาพดีได้มาตรฐานสากลเพื่อให้สามารถแข่งขันในตลาดโลกได้ การใช้เทคโนโลยีสามารถให้ผลิตที่สูงและมีคุณภาพ

เทคโนโลยี (Technology) เป็นคำมาจากภาษากรีกที่ว่า Techne หมายถึงศิลปะ วิทยาศาสตร์ หรือทักษะ และภาษลาตินว่า Terere หมายถึง การสานหรือการสร้างเทคโนโลยีเกี่ยวกับเทคนิควิธีสมัยใหม่ช่วยในการทำงานให้มีประสิทธิภาพดีขึ้นและประสิทธิผลสูงสุด

กิดานันท์ มะลิทอง (2535 : 1-3) ได้ให้ความหมายของเทคโนโลยีโดยสรุปว่า เทคโนโลยีเป็นการนำเอาแนวความคิด หลักการ เทคนิค ความรู้ ระเบียบวิธี กระบวนการตลอดจนผลผลิตทางวิทยาศาสตร์ทั้งในด้านสิ่งประดิษฐ์และวิธีการปฏิบัติมาประยุกต์ใช้ในระบบงานเพื่อช่วยให้เกิดการเปลี่ยนแปลงในการทำงานให้ดีขึ้นและเพื่อเพิ่มประสิทธิภาพผลของงานนั้นให้ดีมากยิ่งขึ้น

เทคโนโลยี คือการประยุกต์ความรู้ทางวิทยาศาสตร์ไปใช้ในการผลิตหรือการดำเนินกิจกรรมต่างๆด้วยวิธีการใหม่ๆ เพื่อให้มีกิจกรรมนั้นมีประสิทธิภาพและเกิดประโยชน์ต่อมวลมนุษย์มากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 โรงเรือนและระบบการเลี้ยง

การเลือกทำเลที่ตั้งฟาร์ม

การพิจารณาเลือกทำเลที่ตั้งฟาร์ม ถือว่าเป็นเรื่องสำคัญมากอย่างหนึ่ง จึงควรพิจารณาเลือก ดังนี้

1. ราคาที่ดินไม่สูงมากนัก
2. เป็นที่ที่น้ำท่วมไม่ถึง
3. การคมนาคมสะดวกสำหรับขนส่งลูกไก่ อาหาร และไก่ใหญ่ เป็นต้น
4. แหล่งน้ำ ถือว่าสำคัญมากที่สุด จะต้องสะอาด ไม่กระด้าง ไม่เป็นกรดหรือด่าง และต้องเพียงพอตลอดปี

ระบบการเลี้ยงไก่เนื้อในเมืองไทย สามารถแบ่งได้เป็นประเภทใหญ่ๆ ดังนี้

1. ระบบโรงเรือนเปิด
2. ระบบโรงเรือนปิด (อีแว๊ป)

ระบบโรงเรือนเปิด สามารถแบ่งออกได้เป็น

1. เล้าบนพื้นดิน
2. เล้าลอย คือ เล้าที่สร้างแบบยกพื้น ซึ่งมีทั้งยกพื้นบนบก และยกพื้นในน้ำ

โรงเรือน

โดยทั่ว ๆ ไป การเลี้ยงไก่เนื้อในเชิงธุรกิจ จะสร้างโรงเรือนเป็นรูปสี่เหลี่ยมผืนผ้า ขนาดกว้าง 10-12 เมตร ความยาวไม่เกิน 100-120 เมตร ระบบโครงสร้างของโรงเรือนจะใช้วัสดุอะไรก็แล้วแต่ความเหมาะสมของแต่ละพื้นที่ โดยคำนึงถึงความคงทนและราคา เพราะจะต้องสร้างให้ถูกที่สุดและดีที่สุด สามารถเลี้ยงไก่ได้มากที่สุด ในเงินทุนที่เท่ากัน

เนื่องจากเมืองไทยเป็นเมืองร้อน แต่พันธุ์ไก่เนื้อเป็นพันธุ์เมืองหนาว ฉะนั้น การสร้างโรงเรือนจะต้องคำนึงถึงอุณหภูมิภายในโรงเรือน และระบายอากาศที่ดี

ดังนั้นโรงเรือนจะต้อง

1. สร้างตามแนวตะวันขึ้น-ตกดิน ไม่ขวางตะวัน
2. กรณีเล้าเปิด ควรมีจั่วหรือลูกหมุน เพื่อระบายอากาศ
3. สร้างบนเนิน หรือถมดินให้สูง 30-50 เซนติเมตร
4. ด้านข้างโรงเรือนห่างกันไม่ต่ำกว่า 20 เมตร และไม่มีวัชพืชขึ้นบังลม อากาศจะถ่ายเทผ่านโรงเรือนได้
5. การมีหญ้าเตี้ยๆ รอบโรงเรือน ช่วยลดความร้อนที่สะท้อนเข้าโรงเรือนได้ โรงเรือนที่ไม่มีระบบระบายความร้อนช่วย ไม่ควรกว้างเกิน 10 เมตร
6. ชายคาโรงเรือนสูงจากพื้นไม่น้อยกว่า 3 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. มีชายคายื่นออกไปพอสมควร เพื่อไม่ให้แสงแดดส่องเข้าไปในโรงเรือน ทำให้ไก่หนีแสงเข้าไปอยู่กลางเล้า
8. การติดม่านกรองแสง เสริมแนวชายคาออกไป จะช่วยลดรังสีความร้อนรอบๆ เล้าได้บ้าง ระวังอย่าให้ม่านกรองแสงห้อยลงมาบังลม จำทำให้ร้อนและอบอ้าวมากยิ่งขึ้น
9. การติดม่านกรองแสงเหนือหลังคา เพื่อลดความเข้มของแสงที่ตกลงบนหลังคา เป็นการลดความร้อนในโรงเรือนไก่ได้บางส่วน
10. การตีฝ้าหรือใช้วัสดุใต้หลังคา เพื่อป้องกันความร้อนจากหลังคาแผ่รังสีความร้อนลงมา

2.2.3 ประเภทการเลี้ยงไก่เนื้อ

ระบบการเลี้ยงไก่เนื้อ แบ่งออกเป็นประเภทใหญ่ ๆ ได้ 2 ประเภท ดังนี้คือ

ประเภทอิสระ

ค่าใช้จ่ายต่างๆ ผู้เลี้ยงจะลงทุนเองทั้งหมด เช่น ค่าก่อสร้างโรงเรือน อุปกรณ์ ค่าลูกไก่ อาหาร ยา เป็นต้น เมื่อผลผลิตออกมา เจ้าของก็จะต้องหาตลาดเอง ราคาขายก็ขึ้นอยู่กับกลไกการตลาด ผู้เลี้ยงประเภทนี้จะมีความเสี่ยงสูงกว่ากลุ่มผู้เลี้ยงแบบอื่นๆ ดังนั้น ผู้เลี้ยงจะต้องมีประสบการณ์มีข้อมูลการตลาด รู้จักวางแผน ความสำเร็จจึงอยู่ที่ราคาเป็นหลัก

ประเภทประกัน

การเลี้ยงไก่เนื้อประเภทนี้ สามารถแบ่งย่อยได้ 2 ประเภท

1. การเลี้ยงแบบประกันราคา

การเลี้ยงประเภทนี้ ผู้เลี้ยงจะต้องลงทุนสร้างโรงเรือน อุปกรณ์ ค่าจ้างแรงงาน และค่าใช้จ่ายอื่นๆ เช่น ค่าไฟ ค่าแกลบ ส่วนค่าลูกไก่ อาหาร ยารักษาโรค เกษตรกรทำสัญญากับบริษัทแล้วบริษัทจับไก่ใหญ่กลับ โดยหักค่าใช้จ่ายกัน ส่วนที่เหลือก็เป็นผลกำไรของเกษตรกร การเลี้ยงประเภทนี้เกษตรกรไม่มีอัตราเสี่ยงในเรื่องราคา เพราะได้ประกันราคาไว้กับบริษัท กำไรที่ได้จึงขึ้นกับฝีมือการเลี้ยง สำหรับราคาประกัน บริษัทตั้งราคาลูกไก่ ราคาอาหาร และราคาไก่ใหญ่ ให้ไม่ห่างกับราคาตลาดมากนัก

2. ประเภทรับจ้างเลี้ยง

การเลี้ยงประเภทนี้ เกษตรกรได้ตกลงกับบริษัทในเรื่องค่าจ้างเลี้ยงเท่านั้น โดยผลตอบแทนตั้งหลักเกณฑ์ไว้ เช่น จำนวนไก่รอดตาย น้ำหนักไก่ที่จับ โดยมากผลตอบแทนจะไม่เกินตัวละ 2 บาทการเลี้ยงประเภทนี้เกษตรกรจะไม่รับภาระการเสี่ยงในด้านต้นทุนและราคาจำหน่าย การเลี้ยงไก่เนื้อ ความเคลื่อนไหวราคาจะรวดเร็ว ความเสี่ยงสูง ฉะนั้น มือใหม่ควรเลือกเลี้ยงระบบประกันราคา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 แนวคิดการจัดการเลี้ยงดูไก่เนื้อ

กระทรวงเกษตรและสหกรณ์จึงกำหนดให้มีมาตรฐานฟาร์มเลี้ยงสัตว์ของประเทศไทย เป็นประกาศกระทรวงเกษตรและสหกรณ์ ลงวันที่ 3 พฤศจิกายน 2542 และวันที่ 15 พฤษภาคม 2545 ว่ามาตรฐานฟาร์มเลี้ยงไก่เนื้อของประเทศไทย พ.ศ. 2542

สินค้าปศุสัตว์ที่มีส่วนสำคัญต่อการผลิต เนื่องจากเนื้อสัตว์เป็นแหล่งอาหารโปรตีนของมนุษย์ ในปัจจุบันผู้บริโภคมีความรู้ และความต้องการในการบริโภคเนื้อสัตว์ ที่สะอาดและถูกสุขอนามัย และปราศจากสารตกค้าง (กระทรวงเกษตรและสหกรณ์. 2548 [online]) สถานการณ์การระบาดของโรคไข้หวัดนกมีการระบาดอย่างต่อเนื่องในเขตภูมิภาคเอเชีย จึงทำให้รัฐบาลไทยมีการกำกับดูแลในเรื่องของการจัดการฟาร์มเลี้ยงสัตว์เพิ่มสูงขึ้นเพื่อลดการระบาดของโรค ดังนั้น กระทรวงเกษตรและสหกรณ์ จึงได้มอบหมายให้กรมปศุสัตว์ดำเนินงานในเรื่องมาตรฐานฟาร์มอย่างต่อเนื่อง เพื่อให้รับรองมาตรฐานฟาร์มเลี้ยงสัตว์ในเรื่องของการจัดการฟาร์มเลี้ยงไก่เนื้อ

สินค้าปศุสัตว์ที่มีส่วนสำคัญต่อการผลิต เนื่องจากเนื้อสัตว์เป็นแหล่งอาหารโปรตีนของมนุษย์ ในปัจจุบันผู้บริโภคมีความรู้ และความต้องการในการบริโภคเนื้อสัตว์ ที่สะอาดและถูกสุขอนามัย และปราศจากสารตกค้าง (กระทรวงเกษตรและสหกรณ์. 2548 [online]) สถานการณ์การระบาดของโรคไข้หวัดนกมีการระบาดอย่างต่อเนื่องในเขตภูมิภาคเอเชีย จึงทำให้รัฐบาลไทยมีการกำกับดูแลในเรื่องของการจัดการฟาร์มเลี้ยงสัตว์เพิ่มสูงขึ้นเพื่อลดการระบาดของโรค ดังนั้น กระทรวงเกษตรและสหกรณ์ จึงได้มอบหมายให้กรมปศุสัตว์ดำเนินงานในเรื่องมาตรฐานฟาร์มอย่างต่อเนื่อง เพื่อให้รับรองมาตรฐานฟาร์มเลี้ยงสัตว์ในเรื่องของการจัดการฟาร์มเลี้ยงไก่เนื้อซึ่งการจัดการฟาร์มเลี้ยงไก่เนื้อ วันที่ 20 มกราคม 2549 โดยบัณฑิตศึกษา คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง มีผลวิจัยเกษตรว่า

การจดบันทึกผลการเลี้ยงพบว่า มีการยอมรับเทคโนโลยีด้านนี้ในระดับสูง

การตรวจการกินอาหารของไก่ พบว่า มีการยอมรับเทคโนโลยีด้านนี้ในระดับสูง

การเปลี่ยนวัสดุรองพื้น มีการยอมรับเทคโนโลยีด้านนี้ในระดับสูง

การใช้ระบบระบายอากาศในโรงเรือน มีการยอมรับเทคโนโลยีในระดับปานกลาง

การจัดการกับอาหารเหลือหรืออาหารที่ตกหล่น มีการยอมรับเทคโนโลยีด้านนี้ในระดับสูง

การจัดการกับการกกลูกไก่ มีการยอมรับเทคโนโลยีด้านนี้ในระดับปานกลาง

การจัดการกับซากไก่ตาย จากการสำรวจพบว่า เกษตรกร (ร้อยละ 50.00) ที่นำไปเลี้ยงสัตว์อื่น รองลงมา(ร้อยละ 25) ที่ทำการเผา หรือฝังดิน โดยเท่ากับเกษตรกรที่ขายหรือโยนทิ้งนอกฟาร์ม คิดเป็น (ร้อยละ 25)เช่นกัน เมื่อพิจารณาเป็นการยอมรับเทคโนโลยีด้านนี้ในระดับปานกลาง

ไก่เนื้อเป็นสัตว์ที่มีความไวต่อสภาพแวดล้อมมาก เทพ พงษ์พานิช (2527 : 310-311) ที่กล่าวว่า เกษตรกรสามารถเรียนรู้ รับรู้ ได้ดีก็ต่อเมื่อเขามีความปรารถนา และนำเอาความรู้นั้นไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปฏิบัติให้เกิดประโยชน์กับตัวของเกษตรกรเอง จนมีความชำนาญ และนอกจากนี้ยังมีปัจจัยอีกมากมายที่ทำให้เกษตรกรมีอัตราการยอมรับ มาก น้อย ช้า เร็ว แตกต่างกันไป เช่น การศึกษา เพศ อายุ ฐานะทางเศรษฐกิจ เป็นต้น

การศึกษาวิธีการบันทึกข้อมูลในการเลี้ยงไก่เนื้อของเกษตรกรผู้เลี้ยงไก่เนื้ออิสระพบว่า เกษตรกรมีการบันทึกทุกครั้งที่นำไก่มาเลี้ยง คิดเป็นร้อยละ (82.25) มีค่าเฉลี่ยเท่ากับ 2.75 จึงสรุปได้ว่า มีการยอมรับเทคโนโลยีในระดับสูง ซึ่งสอดคล้องกับ (อาวุธ ตันโช,2538 : 70) ที่กล่าวไว้ว่า ต้องมีการจดบันทึกผลการเลี้ยงและข้อมูลต่างๆ มีวัตถุประสงค์เพื่อให้ผู้เลี้ยงทราบถึงผลการดำเนินงานทุกขั้นตอน เมื่อเกิดการผิดพลาดจะสามารถหาสาเหตุได้อย่างรวดเร็วและแก้ไขได้ทันท่วงที

การปฏิบัติก่อนที่จะเข้าโรงเรือนเลี้ยงไก่เนื้อของเกษตรกรผู้เลี้ยงไก่เนื้ออิสระ พบว่า เกษตรกรมีการเปลี่ยนชุด และผ่านน้ำยาฆ่าเชื้อทุกครั้ง กับการไม่เปลี่ยนชุดแต่ผ่านน้ำยาฆ่าเชื้อมีจำนวนมากเท่ากันคือ ร้อยละ (37.5) และมีค่าเฉลี่ยเท่ากับ 2.00 จึงสรุปได้ว่า เกษตรกรมีการยอมรับเทคโนโลยีระดับปานกลาง (ปฐม เลาหะเกษตร, 2540 :128) ที่กล่าวไว้ว่า การจัดการสุขภาพสัตว์ว่า บุคลากรเข้า-ออก บริเวณเลี้ยงสัตว์จะต้องเปลี่ยนชุดที่ฟาร์มจัดเตรียมไว้ให้

2.4. การเลี้ยงไก่แบบระบบอีแว๊ป (Evaporative Cooling System)

การเลี้ยงไก่แบบระบบอีแว๊ป หรือการเลี้ยงแบบระบบปรับอากาศ เป็นการควบคุมอุณหภูมิภายในโรงเรือนให้เหมาะสมกับความต้องการของไก่ในอายุนั้นๆ โดยอาศัยหลักการดูดอากาศร้อนภายในโรงเรือนออก แล้วทำให้อากาศเย็นที่ผ่านการลดอุณหภูมิ ให้ต่ำลงด้วยชุดทำความเย็น เข้ามาแทนที่ ซึ่งส่งผลให้อุณหภูมิภายในโรงเรือนเย็นลง เหมาะกับการเจริญเติบโตของไก่ช่วงอายุนั้นๆ

โดยทั่วไประบบปรับอากาศ (อีแว๊ป) สามารถปรับอุณหภูมิให้ลดลงได้ประมาณ 6-9 องศาเซลเซียส หรืออุณหภูมิประมาณ 28 องศาเซลเซียส ความชื้นภายในโรงเรือน 75-85% ทำให้สามารถเลี้ยงไก่ได้เพิ่มขึ้นจากไม่เกิน 8ตัว / ตารางเมตร เป็น 12-13 ตัว/ตารางเมตร

โรงเรือนระบบอีแว๊ปเป็นโรงเรือนระบบปิด เพื่อให้อากาศสามารถผ่านเข้าภายในโรงเรือนได้ทางชุดทำความเย็นเพียงทางเดียว และอากาศถูกดูดออก โดยพัดลมดูดอากาศเพียงทางเดียวเช่นกัน อุณหภูมิและความชื้นถูกควบคุมโดยชุดเทอร์โมสแตท

2.4.1.ส่วนประกอบที่สำคัญ

1. โรงเรือน

โรงเรือนระบบอีแว๊ป จำเป็นต้องเป็นระบบปิด เพื่อให้อากาศผ่านเข้า-ออกทางเดียว โดยทั่วไปโรงเรือนระบบเปิดสามารถดัดแปลงเป็นโรงเรือนระบบปิดได้ โดยการติดตั้งชุดฝ้าเพดานและชุดผ้าม่านด้านข้าง โดยใช้ผ้าพาสติก พี.วี.ซี หรือ พี.อี. เพราะทำการติดตั้งง่ายและต้นทุนในการติดตั้งต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.พัดลมดูดอากาศ

พัดลมดูดอากาศทำหน้าที่ดูดอากาศร้อนในโรงเรือนออกในขณะเดียวกันก็จะดูดอากาศเย็นที่ผ่านการลดอุณหภูมิจากชุดทำความเย็นเข้ามาในโรงเรือนแทนที่อากาศที่ถูกดูดออกไป ทำให้อุณหภูมิในโรงเรือนลดลง

โดยทั่วไป พัดลมดูดอากาศที่ใช้มี 2 ขนาด คือ ขนาด 48 นิ้ว และ ขนาด 36 นิ้ว มีความสามารถในการดูดลม 22,000 CFM และ 11,000 CFM ตามลำดับ (ค่า CFM คือค่าประสิทธิภาพของพัดลม โดยคิดจากเส้า กว้าง(ฟุต)xยาว(ฟุต)xสูง(ฟุต) =CFM)

3. ชุดทำความเย็น

ชุดทำความเย็นที่นิยมใช้ในประเทศไทยมีอยู่ 2 แบบ คือ

1. แบบเยื่อกระดาษ (คูลลิ่ง แพด) มีลักษณะเป็นแท่งกระดาษแข็ง มีรูคล้ายรังผึ้ง มีหลายขนาด ในสภาวะน้ำปกติ ประกอบกับการบำรุงรักษาดี เยื่อกระดาษจะมีอายุการใช้งาน 5 ปี อุณหภูมิ 25-28 องศาเซลเซียส
2. แบบพ่นหมอก (แบบมุ้งเขียว) จะใช้หัวพ่นหมอกพ่นน้ำให้เป็นละออง โดยจะมีชุดโครงมุ้งช่วยดักไอน้ำ ทำให้สัมผัสอากาศได้ดียิ่งขึ้น สามารถลดอุณหภูมิได้ 26-28 องศาเซลเซียส ต้นทุนการผลิตต่ำ อายุการใช้งานยาวนาน การดูแลรักษาง่าย

4.ปั๊มน้ำ

ปั๊มน้ำ ทำหน้าที่ดูดน้ำจากบ่อ เพื่อมาหล่อเย็นชุดคูลลิ่ง แพด หรือพ่นหมอก สำหรับแบบหัวพ่นหมอกเครื่องปั๊มน้ำจะถูกควบคุมด้วยเทอร์โมสแตท

5.ถังเก็บน้ำ

ถังเก็บน้ำ ใช้สำหรับเก็บพักน้ำที่ใช้หล่อเย็นชุดความเย็นโดยทั่วไปใช้ขนาด 1,000-1,500 ลิตร อาจเป็นถังที่ก่อด้วยคอนกรีต หรือถังเก็บน้ำที่มีขายทั่วไป

6.ชุดควบคุมอุณหภูมิ

ทำหน้าที่ควบคุมอุณหภูมิภายในโรงเรือนให้อยู่ในเกณฑ์ที่ตั้งไว้โดยจะควบคุมการเปิด-ปิด ของพัดลมและปั๊มน้ำ ควรติดตั้งไว้ตรงจุดกึ่งกลางของโรงเรือน เพื่อให้สามารถวัดอุณหภูมิโดยเฉลี่ยของโรงเรือนได้

7.เครื่องปั่นไฟ

ไฟฟ้าสำรองมีความจำเป็นอย่างยิ่ง ทุกฟาร์มที่เลี้ยงระบบอีแว๊ป ต้องมีเครื่องปั่นไฟสำรองเพื่อใช้ในกรณีไฟฟ้าที่ใช้ประจำมีปัญหา มิฉะนั้นหากไฟฟ้าดับจะเกิดความเสียหายอย่างใหญ่หลวง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ที่จำเป็นในการเลี้ยง

1. อุปกรณ์การกกไก่ ขึ้นอยู่กับว่าจะกกแบบไหน ไฟฟ้า แก๊ส หรือกระโจม เป็นต้น
2. ที่ให้น้ำสำหรับไก่เล็ก (กระติกน้ำ)
3. ที่ให้น้ำสำหรับไก่ใหญ่ เช่น รางน้ำอัตโนมัติ ถังพลาสติกอัตโนมัติ เป็นต้น
4. ถาดอาหารสำหรับลูกไก่
5. ถังอาหารหรือที่ให้อาหารอัตโนมัติ
6. รถเข็นอาหาร
7. ถังละลายยาและวิตามิน
8. อ่างยาฆ่าเชื้อ เพื่อจุ่มเท้าก่อนเข้าเล้า
9. เครื่องปั๊มน้ำ สำหรับฉีดยาฆ่าเชื้อทำความสะอาดเล้า หรือฉีดน้ำไก่ขณะร้อน

การเตรียมโรงเรือนก่อนนำลูกไก่เข้าเลี้ยงเมื่อขายไก่ใหญ่ออก

1. รีบเก็บขี้ไก่ออกจากเล้าให้เร็วที่สุด
2. ล้างเล้า ทำความสะอาดอุปกรณ์ทุกชิ้นด้วยผงซักฟอก
3. อุปกรณ์ทุกชนิดให้จุ่มยาฆ่าเชื้อ ทิ้งไว้ให้แห้ง แล้วนำเข้าเก็บ
4. โรงเรือนพ่นยาฆ่าเชื้อ โรยปูนขาวที่พื้นเล้าและบริเวณรอบ ๆ เล้าแล้วพักเล้าไว้
5. ก่อนนำไก่เข้าเลี้ยง 3 วัน ควรฉีดยาพ่นฆ่าเชื้ออีกครั้ง โรยปูนขาวทิ้งไว้ให้แห้ง
6. อุปกรณ์ที่ให้น้ำและให้อาหาร ควรทำความสะอาดอีกครั้ง

เมื่อไก่ถึงฟาร์ม

จัดอุปกรณ์สำหรับกกไก่และเลี้ยงไก่เล็กไว้ให้พร้อม โดยเฉพาะเรื่องความอบอุ่น **น้ำดื่ม** ส่วนอาหาร ยังไม่จำเป็นในการลงไก่วันแรก ควรให้เสร็จก่อนไก่ถึงฟาร์มอย่างน้อย 1 ชั่วโมง

การเลี้ยงไก่ในช่วงเล็กถือว่าเป็นช่วงที่สำคัญที่สุด อาจกล่าวได้ว่าเป็นช่วงวิกฤติเลยก็ว่าได้ ผลสำเร็จของการเลี้ยงไก่เป็นผลพวงมาจากการเลี้ยงไก่ในระยะกก หรือการดูแลสัตว์วัยอ่อนที่ดี เนื่องจากสาเหตุไก่แตกไซส์ ได้ก้นเล้าเยอะ ตัดวัคซีนยาก หรือการป่วยระยะท้ายๆ ส่วนใหญ่ก็มาจากการดูแลไก่ในช่วงเล็กไม่ดี เช่น วัสดุรองพื้นเปียกชื้น เหม็นอับ มีกลิ่นก๊าซแอมโมเนีย ที่ทำให้อาหารไม่เพียงพอ เป็นต้น

เทคนิคการลงไก่ในวันแรก

การลงไก่ในวันแรก หลายคนไม่เคยให้ความสำคัญกับสิ่งนี้เลย โดยลืมนึกไปว่าช่วงนี้ลูกไก่ต้องปรับตัวอย่างมาก จากการใช้ชีวิตในเปลือกไข่ อุณหภูมิพอเหมาะ มาเป็นการใช้ชีวิตบนพื้นดิน ที่ต้องกินอาหารเอง น้ำบางตัวยังไม่รู้จัก ทำให้เราต้องคอยช่วยเหลือ การเคลื่อนย้ายลูกไก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโรงฟักไปยังฟาร์ม ลูกไก่ยังกระทบกับสิ่งแวดล้อมไม่เหมาะสมอีก ไก่สูญเสียน้ำและเกลือแร่ ไก่เครียดจากการขนย้าย อากาศร้อน ลม เป็นต้น การลงไก่วันแรก ทางฟาร์มจะต้องมีความพร้อมมากที่สุดเพื่อลดความเครียดที่เกิดกับลูกไก่ให้เหลือน้อยที่สุด

ลูกไก่แรกเกิดต้องการอะไร

1. ความอบอุ่น ทำอย่างไรถึงให้อุณหภูมิพอเหมาะกับความต้องการของลูกไก่
2. วัสดุรองพื้น จะต้องใหม่ แห้ง สะอาด
3. น้ำและอาหาร จะต้องใหม่ เพียงพอ และสะอาด
4. ภูมิคุ้มโรค จะต้องทำวัคซีนป้องกันโรคที่จำเป็น คือนิวคาสเซิลหลอดลมอักเสบและกัมโบโร

การให้ความอบอุ่นกับไก่ (กกไก่) มีหลายแบบ

1. การกกด้วยหลอดไฟฟ้าใช้หลอดไฟฟ้า 60-100 วัตต์ จำนวน 2-3 ดวงต่อวงกกสามารถกกไก่ได้ประมาณ 500 ตัว
2. การกกไก่ด้วยแก๊ส หัวกกแก๊สเป็นอุปกรณ์ที่ให้ความร้อนโดยตรง ทำให้สามารถกกไก่ได้ประมาณ 1000-1500 ตัวต่อวงกก
3. การกกไก่แบบกระโจม เป็นวิธีใหม่ที่ประหยัดค่าไฟฟ้าได้มากกว่า 50% สามารถกกได้ไม่เกิน 5000 ตัว/กระโจม

หลักการโดยรวมของการกกไก่

1. โรงเรือน โดยเฉพาะพื้นที่กกจะต้องสะอาด มีความอบอุ่นเหมาะสมกับไก่แรกเกิด (95 องศาฟาเรนไฮต์) เตรียมพร้อมก่อนไก่มาถึงอย่างน้อย 1 ชั่วโมง
2. ต้องให้ไก่ได้รับน้ำ (กินน้ำ) ให้เร็วที่สุด ต้องคอยดูแลลูกไก่ตัวที่กินน้ำไม่เป็น โดยช่วยจับปากจุ่มน้ำ
3. วันแรกที่ไก่มาถึง น้ำที่ให้ไก่กินควรละลายอิเลคโตรไลท์ (เกลือแร่) ด้วย
4. ที่ให้น้ำและอาหารควรมีปริมาณเพียงพอ ในวันแรกควรเพิ่มที่ให้น้ำ 50-100%
5. การขยายก็ควรทำทุก ๆ 2-3 วัน ไม่ควรขยายเร็วหรือช้าเกินไป
6. ควรดูแลวัสดุรองพื้นด้วย ไม่ให้เปียกแฉะหรือมีกลิ่นแก๊สมาก เพราะแก๊สแอมโมเนียเป็นสาเหตุโน้มนำ ทำให้เกิดปัญหาโรคระบบทางเดินหายใจ กดภูมิคุ้มกัน
7. การกกไก่ อุณหภูมิในการกกจะต้องไม่สูงมาก เพราะสูงมากจะทำให้ไก่สูญเสียน้ำจากร่างกาย ไก่แห้งเหมือนการอบไก่ ส่วนอุณหภูมิต่ำ ไก่ก็มีปัญหาการใช้ไข่แดงไม่หมด เกิดการท้องมานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ช่วงเวลาที่สำคัญที่สุด คือ หลังเที่ยงคืนไปแล้ว ซึ่งเป็นช่วงที่อากาศเย็น เราควรตรวจเช็คระบบการกกไก่ว่าเก็บอุณหภูมิได้หรือไม่
9. การกก คือ การรักษาอุณหภูมิกลางวันกับกลางคืนไม่ให้ห่างกันมาก เหมาะสมกับช่วงอายุนั้นๆ

การกกแบบกระโจม

การกกแบบกระโจม เป็นการกกไก่แบบใหม่ที่ได้ผลดีต่อสุขภาพไก่ และช่วยลดค่าใช้จ่าย (ค่าไฟฟ้าหรือค่าแก๊ส)ลงมากกว่าครึ่ง การกกแบบนี้ได้ทดลองการพัฒนาแก้ไขจุดบกพร่อง จนได้รูปแบบที่เหมาะสมและดีที่สุด โดยคุณกฤษดา หิรัญรัตนากร (เฮียแก้ว เจ้าของกฤษดาฟาร์ม) โดยยึดหลักดังนี้ คือ

1. กักเก็บอุณหภูมิได้ สามารถควบคุมอุณหภูมิได้ง่าย พอเหมาะกับไก่อายุนั้นๆ
2. การทำงานสะดวก รวดเร็ว แต่ได้ประสิทธิภาพ
3. การถ่ายเทอากาศดี แต่ลมไม่โกรก
4. ปริมาณก๊าซเสียน้อยที่สุด เช่น แอมโมเนีย คาร์บอนไดออกไซด์
5. ไม่คับแคบจนเกินไป การขยายกกก็เป็นไปตามอายุไก่ระยะนั้น
6. ความสว่างทั่วบริเวณกก ไม่มีมุมมืดให้ไก่หลบนอน
7. ปริมาณน้ำและอาหารที่ให้ มีมากเพียงพอต่อความต้องการ

อาหารและการให้อาหาร

การเลี้ยงไก่เนื้อเชิงธุรกิจ อาหารที่ใช้ส่วนใหญ่เป็นอาหารสำเร็จรูปจากบริษัท ซึ่งประกอบด้วยวัตถุดิบต่าง ๆ ที่ให้สารอาหารตามที่ไก่ต้องการในช่วงอายุนั้นๆ

ในอาหารไก่มีอะไรบ้าง

1. **อาหารประเภทแป้ง** (อาหารพลังงาน) เป็นวัตถุดิบที่ให้โปรตีนค่อนข้างต่ำ แต่ไก่จะใช้เป็นแหล่งพลังงานของร่างกาย เช่น
   - ปลายข้าว ให้โปรตีน 8% พลังงาน 3500 กิโลแคลอรี/กิโลกรัม
   - รำละเอียด ให้โปรตีน 12% พลังงาน 2710 กิโลแคลอรี/กิโลกรัม
   - ข้าวโพด ให้โปรตีน 8% พลังงาน 3370 กิโลแคลอรี/กิโลกรัม
   - ข้าวฟ่าง ให้โปรตีน 11% พลังงาน 3250 กิโลแคลอรี/กิโลกรัม
   - มันสำปะหลัง ให้โปรตีน 2% พลังงาน 3500 กิโลแคลอรี/กิโลกรัม
2. **อาหารประเภทโปรตีน**
   - ปลาป่น มีโปรตีนเฉลี่ยประมาณ 50-60%
   - การถั่วเหลือง มีโปรตีนประมาณ 44%
   - การเมล็ดยางพารา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กากปาล์มน้ำมัน กากเนื้อในเมล็ดปาล์ม ชนิดกะเทาะเปลือก มีคุณค่าทางอาหารดีที่สุด มีโปรตีน 16-18%

3. **อาหารประเภทพลังงานสูง** เช่น น้ำมันพืช ไขมันสัตว์ เป็นต้น ใส่ไปเพื่อปรับระดับพลังงานและลดการฟุ้งกระจาย
4. **อาหารประเภทวิตามินและแร่ธาตุ** ใส่ลงไปเพื่อสมดุลวิตามินและแร่ธาตุ
5. **สารเสริมอื่น ๆ** เช่น กรดอะมิโนสังเคราะห์ สารกันเชื้อรา ยากันบิด

อาหารสัตว์ที่ดี จะต้องมีสารอาหารครบถ้วนและสมดุลในทุก ๆ ด้านจึงจะทำให้สัตดำรงชีวิตอยู่ได้ และให้ผลผลิตดี

สารอาหารกับภูมิคุ้มกันในสัตว์

อาหารคุณภาพดีจะทำให้สัตว์สุขภาพแข็งแรงคือไม่เป็นโรคง่ายหรือติดเชื้อง่าย ร่างกายสามารถต้านทานโรคได้ดี เหตุที่ต้านทานโรคได้ดี เนื่องจากร่างกายมีภูมิคุ้มกันโรคนั่นเอง สรุปคือ อาหารช่วยทำให้เกิดภูมิคุ้มกันโรค ช่วยให้ระบบภูมิคุ้มกันของร่างกายสัตว์ทำงานได้ดีขึ้น

คุณภาพและมาตรฐานของอาหารไก่เนื้อ

1. ส่วนประกอบทางเคมี อาหารไก่เนื้อที่มีคุณภาพและผ่านมาตรฐานนั้น ควรมีส่วนประกอบทางเคมีดังนี้ คือ

ชนิดของไก่เนื้อ	คุณค่าทางเคมี			
	โปรตีนไม่น้อยกว่าร้อยละ	ไขมันไม่น้อยกว่าร้อยละ	กากไม่มากกว่าร้อยละ	ความชื้นไม่มากกว่าร้อยละ
ไก่เนื้อแรกเกิด-อายุ 3 สัปดาห์	21	3	5	13
ไก่เนื้ออายุ 3-6 สัปดาห์	19	3	5	13
ไก่เนื้ออายุ 6 สัปดาห์ขึ้นไป	17	3	5	13

ตารางที่ 2.1 อาหารและคุณภาพอาหารไก่เนื้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. อาหารไก่เนื้อที่ดีต้องไม่เป็นอาหารดังนี้ คืออาหารสัตว์ปลอมปน คืออาหารสัตว์ที่ใช้วัตถุอื่นไม่ตรงกับที่ขึ้นทะเบียนไว้ ยกเว้นวัตถุซึ่งอาจมีได้โดยธรรมชาติ ตามที่รัฐมนตรีประกาศกำหนด

อาหารผิดมาตรฐาน คือ อาหารสัตว์ที่มีคุณภาพไม่ถูกต้องตามมาตรฐานกำหนด

อาหารเสื่อมคุณภาพ ได้แก่ อาหารไก่เนื้อที่มีลักษณะ ดังนี้คือ

- อาหารสัตว์ที่หมดอายุไปจากที่แสดงไว้ในฉลาก
- อาหารสัตว์ที่เสื่อมคุณภาพ ซึ่งมีลักษณะดังนี้ คือ
  - ก. มีเชื้อซัลโมแนลล่า (Salmonella spp.)
  - ข. มีเชื้อแบคทีเรียปริมาณมากกว่า 8x10 โคโลนี/1 กรัมน้ำหนักอาหาร
  - ค. มีปริมาณอะฟล่าท็อกซิน มากกว่า 100 ไมโครกรัม/ กิโลกรัม
- อาหารสัตว์ที่บรรจุในภาชนะต้องห้ามตามมาตรา 6(7)

อาหารสัตว์ที่มิได้ขึ้นทะเบียน

อาหารสัตว์ที่อธิบดีสั่งเพิกถอนทะเบียน

3. อาหารไก่

อาหารที่ผลิต จะต้องมีฉลากแจ้งคุณภาพทางเคมีที่ภาชนะบรรจุ สถานที่ผลิต วัน เดือน ปีที่ผลิต และวันหมดอายุ

เบอร์อาหาร	ระยะเวลาที่ใช้	ระดับโปรตีน	จำนวนอาหารที่ใช้ต่อ 1000 ตัว
ระยะแรก	อายุ 1-28 วัน	21%	30 ถุง
ระยะที่ 2	อายุ 28-45 วัน	19%	60 ถุง
ระยะสุดท้าย	5-7 วันก่อนจับขาย	17%	30 ถุง

ตารางที่ 2.2 ตารางมาตรฐานอาหาร

หลักมาตรฐานการให้อาหารไก่ ปริมาณอาหารที่ให้ไก่กินต่อ 100 ตัว คือ 12% หมายถึง ไก่จำนวน 100 ตัว ให้กินอาหาร 12 ถุง น้ำหนักอาหารถุงละ 30 กิโลกรัม รวมน้ำหนักอาหารทั้งหมดเท่ากับ 360 กิโลกรัม ไก่จะได้น้ำหนัก 1.9 กิโลกรัม/ตัว โดยประมาณ

โดยแบ่งระยะอาหารตามอายุไก่ ดังนี้

อาหารระยะแรก 3 ถุง

ระยะที่ 2 จำนวน 6 ถุง

ระยะที่ 3 จำนวน 2 ถุง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยอาหารแต่ละระยะมีระดับโภชนาการแตกต่างกัน เพียงพอสำหรับไก่อายุนั้น ๆ โดยทั่วๆ ไปอาหารระยะแรกกับระยะที่ 2 จะมีการผสมยาป้องกันบิดด้วย ส่วนระยะสุดท้าย จะไม่ใส่ยากันบิด เพื่อป้องกันการตกค้างของยาที่จะมาถึงผู้บริโภค ฉะนั้นระยะนี้จึงเป็นช่วงที่เสี่ยงต่อการเกิดบิด ดังนั้นจะต้องดูแลสภาพพื้นโรงเรือนให้แห้ง ดูแลเรื่องสุขาภิบาลให้ดี

การเปลี่ยนอาหารจากระยะแรกเป็นระยะที่สอง จะอยู่ในช่วงวันที่ 18-20 ของอายุไก่ ในระยะนี้เป็นระยะที่ไก่ได้รับความเครียดหลายทาง เช่น ความเครียดที่เกิดจากการเปลี่ยนอาหาร บางครั้งอาจเกิดจากการติดหวัดจากการทำวัคซีนครั้งแรก (อายุ 7-10 วัน) ยังไม่หายดี ระยะนี้บางโปรแกรมวัคซีนจะต้องทำวัคซีนนิวคาสเซิล เพื่อกระตุ้นภูมิอีกครั้ง (Booter) และช่วงนี้เป็นช่วงที่ไก่อาจแสดงอาการโรคกัมโบโรหรือบิด จะเห็นได้ว่าความเครียดมีโอกาสเกิดได้หลายทาง

ดังนั้น ความเครียดเนื่องจากการเปลี่ยนอาหาร ซึ่งเราควบคุมได้ ก็ควรจะเลื่อนออกไปให้พ้นระยะวิกฤตินี้ จึงควรเปลี่ยนปริมาณการให้อาหารเป็น ระยะแรก 4% ระยะที่สอง 6% และระยะสุดท้าย 2% เพื่อจะได้เปลี่ยนอาหารหลัง 20 วัน บางคนเข้าใจว่าการให้อาหารระยะสองจะทำให้ไก่โตดีกว่า ทั้งที่จริงคุณค่าทางโภชนาการของอาหารระยะที่สองยังต่ำกว่าระยะแรกเสียอีก

ถ้าพูดถึงต้นทุนที่เพิ่มขึ้นจากการใช้อาหารระยะแรกจะเพิ่มขึ้นมาก เช่น ถ้าคิดจากไก่ 10,000 ตัว อาหารระยะแรกส่วนมากราคาจะห่างจากระยะที่สองประมาณ 10 บาท (แต่ระดับโปรตีนในอาหารห่างกันถึง 2% โปรตีน) การเพิ่มระยะแรกไป 1% เท่ากับเพิ่มอาหารระยะแรกไป 100 ถุงใน 10,000 ตัว ราคาอาหารเพิ่มถุงละ 10 บาท ค่าอาหารที่ต้องเพิ่มขึ้นแค่ 10,000บาท/ไก่ 10,000ตัว ถือว่าน้อยมากเมื่อเทียบกับความเครียดที่เกิดขึ้นจากการเปลี่ยนแปลงอาหารในช่วงนั้น ซึ่งได้ระดับโปรตีนที่เพิ่มขึ้นอีก 2%

2.5 ประโยชน์การเลี้ยงไก่ระบบปิด (อีแว๊ป) Evap

(Closednoused envir onmental control By computerized system with evaporative cooling system) เล้าไก่ที่มีผนังเล้าปิดมิดชิด อากาศจะผ่านเข้ามาในโรงเรือนโดยผ่านแผ่นรังผึ้ง cooling pad + ระบบทำความเย็นด้วยการระเหยของน้ำ Evaporative cooling Wind chilled effect ผลพลอยได้ ลดปริมาณเชื้อโรคที่แพร่ระบาดทางอากาศ

เพื่อปริมาณ ประสิทธิภาพ คุณภาพ มีการออกแบบ + จัดการอย่างถูกต้อง คุ้มค้าการลงทุน

- เพิ่มจำนวนไก่ต่อหนึ่งหน่วยพื้นที่
- ลดความร้อนที่เกิดจาก สิ่งแวดล้อมภายนอก
- ถ่ายเทความร้อนและก๊าซเสียออกจากตัวไก่ ฝุ่น และความชื้นออกนอกโรงเรือน
- ให้ออกซิเจน ไก่สบาย เจริญเติบโต และอัตราการเนื้อดีขึ้น
- ผลผลิตสูงขึ้น ประสิทธิภาพไก่เนื้อสูงขึ้น คุณภาพของผลผลิตดีขึ้น
- ย่นระยะเวลาในการเลี้ยง จาก 49 วันเป็น 43-44 วันทำให้เพิ่มจำนวนรุ่นต่อปีได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- น้ำหนักเมื่อส่งตลาดสูงกว่าแบบ เป็นในทางบวกในปริมาณเท่ากัน
- ป้องกันแมลงและศัตรูที่มารบกวนได้
- การจัดการเรื่องแสงสว่าง ประสิทธิภาพ
- อัตราการตายต่ำกว่า
- ลดจำนวนผู้ดูแลฝูงไก่ลง

ปัญหาและแนวคิดการจัดการฟาร์มเชิงธุรกิจในประเทศไทย

ไม่ว่าจะเป็นบริษัทเล็กหรือใหญ่ หรือจะเป็นผู้เลี้ยงอิสระรายย่อยก็ตาม การเลี้ยงไก่ก็มีเป้าหมายอยู่ที่กำไรสูงสุด พื้นฐานของการจัดการฟาร์มทำให้ผู้เลี้ยงประสบความสำเร็จในการเลี้ยงไก่ตัวอย่าง เช่น ถ้าไก่เครียดเพราะอยู่ในเล้าที่สกปรก มีความชื้นสูง สิ่งรองพื้นเปียกชื้นมีแอมโมเนียสูง อาหารและน้ำคุณภาพไม่ดีน้ำไม่พอให้ไก่กินจะเกิดโรคต่างๆสูง เพื่อจุดประสงค์คือได้ไก่ส่งตลาดในเวลาที่กำหนด ได้น้ำหนักตามเป้าหมาย และต้นทุนการผลิตต่ำสุด และการจัดการฟาร์มที่ดีอย่างสม่ำเสมอเป็นสิ่งจำเป็นที่สุดของการเลี้ยงไก่เนื้อให้ได้กำไร

ปัญหาต่าง ๆ เกี่ยวกับการกกลูกไก่เนื้อ

1. ลูกไก่แสดงอาการหอบมาก ๆ ง่วงเหงา เซื่องซึม เมื่อเดินทางมาถึงฟาร์ม ปัญหานี้เชื่อว่าเกิดจากการขนส่งลูกไก่ขณะที่อากาศร้อนเกินไป

การแก้ไข ผู้เลี้ยงช่วยเหลือไก่ โดยการให้ลูกไก่ได้กินน้ำตาลหรือเกลือแร่ โดยการกระจายภาชนะให้น้ำอย่างทั่วถึง ปัญหานี้บางครั้งอาจเกิดจากการขนส่งลูกไก่เป็นระยะเวลายาวนาน

2. ลูกไก่สุมรวมกันภายใต้เครื่องกก เมื่อเดินทางมาถึงฟาร์ม แสดงอาการไม่สนใจน้ำและอาหาร ปัญหานี้อาจเกิดจากการขนส่งขณะอากาศหนาวเย็นมาก

การแก้ไข โดยการเพิ่มอุณหภูมิกกให้สูงขึ้นอีก 1-2 องศาเซลเซียส หรือลดระดับเครื่องกกไก่ให้ต่ำลง และกระตุ้นให้ลูกไก่กินน้ำและอาหารได้ง่ายขึ้น โดยการขยับกระติกน้ำและถาดอาหารมาไว้ใกล้ตัวไก่

3. ลูกไก่แสดงอาการอ้าปากหายใจ เมื่อเดินทางมาถึงฟาร์ม ปัญหานี้เชื่อว่าเกิดจากการรมควันในโรงฟักมากเกินไป หรืออาจเกิดจากโรคแอสเปอร์จิโลซิส ปัญหานี้อาจเกิดจากโรงฟังและการติดเชื้อจากล่องไก่ ติดเชื้อจากเครื่องกกไก่ ติดเชื้อจากโรงเรือนทำความสะอาดไม่ดี

การแก้ไข โดยการขจัดสาเหตุที่เป็นแหล่งก่อให้เกิดการติดเชื้อใช้วัสดุรองพื้นที่ใหม่และสะอาด ละลายคอปเปอร์ซัลเฟต (จุนสี) ในน้ำให้ไก่กิน ฉีดพ่นคอปเปอร์ซัลเฟตเข้มข้น 1% ในโรงเรือนเพื่อฆ่าเชื้อรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ลูกไก่แสดงอาการกระจายตัว ไม่รวมกันและหายใจลำบก เมื่อเดินทางมาถึงฟาร์ม ปัญหานี้เชื่อว่าเกินจากการขาดออกซิเจนระหว่างการฟักให้ผู้เลี้ยงแจ้งไปยังบริษัทที่ผลิตลูกไก่ หรือบางครั้งอาจเกิดการขนส่งในเส้นทางที่มีฝุ่นมาก ทำให้ลูกไก่ได้รับออกซิเจนไม่เพียงพอ
5. ลูกไก่สุมกันมุมใดมุมหนึ่งของกรง ปัญหานี้เกิดจากสภาพลมโกรก

การแก้ไข โดยการปิดทิศทางลมที่เป็นสาเหตุทำให้ลมโกรก

6. ลูกไก่หนีออกไปอยู่ข้าง ๆ วงกก ปัญหานี้เกิดจากสภาพอุณหภูมิภายในวงกกร้อนเกินไป ผู้เลี้ยงควรยกเครื่องกกให้สูงขึ้น หรือปรับอุณหภูมิของเครื่องกกให้ลดลง ลูกไก่นอนสุมรวมกันภายในตู้เครื่องกกในระหว่างการกก ปัญหานี้เกิดจากสภาพอุณหภูมิภายในวงกกต่ำเกินไป

การแก้ไข โดยการปรับอุณหภูมิเครื่องกกให้สูงขั้น ถ้าไม่เพียงพอก็ให้ลดเครื่องกกไก่ให้ต่ำลง หรือเลื่อนวงกกให้แคบเข้า แต่ต้องระวังอย่าให้ลมโกรกเข้า หรืออาจเกิดจากปัญหาการถ่ายเทอากาศน้อยเกินไป

7. ลูกไก่อยู่ในสภาพไม่ดี อ่อนแอ ขาเหี่ยวแห้ง ลูกไก่ชอบขึ้นบนรางน้ำ สาเหตุเกิดมาจากลูกไก่ขาดน้ำด้วย

การแก้ไข ให้เพิ่มภาชนะที่ให้น้ำอีก 50% พร้อมทั้งผสมอิเลคโตรไลท์หรือวิตามินลงไปในน้ำด้วย

8. ลูกไก่มีอัตราการตายสูงในช่วง 3-5 วันแรก ตัวผอม แสดงว่าลูกไก่อดอาหาร สาเหตุเนื่องจากอุณหภูมิภายในเครื่องกกสูงหรือต่ำเกินไป

การแก้ไข โดยการเพิ่มที่ให้น้ำและอาหารอีก และต้องแน่ใจว่าอุณหภูมิภายในวงกกถูกต้องแล้ว

9. ลูกไก่มีอัตราการตายสูงในช่วงอายุ 3-5 วันแรก ลูกไก่แสดงอาการก้นเปียกแฉะ ตาเปียกแฉะ จมูกเปียก สาเหตุอาจเกิดจากอุณหภูมิภายในวงกกสูงเกินไปหรือเกิดจากการติดเชื้อแบคทีเรีย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การแก้ไข ให้ยาปฏิชีวนะกับลูกไก่

10. ลูกไก่แสดงอาการไอ จาม ในช่วงอายุ 2-3 วัน ปัญหานี้เกิดจากไก่ได้รับเชื้อหรือสภาพแวดล้อมไม่เหมาะสมในวันแรกของการลงไก่หรือเกิดจากปฏิกิริยาของวัคซีนที่ทำในวันแรก

การแก้ไข ให้ยาปฏิชีวนะ และช่วยเรื่องการสุขาภิบาลที่ดี

11. ลูกไก่อายุ 7 วัน เริ่มแสดงออกถึงความไม่สม่ำเสมอของฝูงแต่สภาพอื่น ๆ ปกติ สาเหตุอาจเกิดจากพื้นที่ที่ใช้ในการกกกว้างเกินไป อุณหภูมิไม่สม่ำเสมอภายในวงกก มักจะเกิดกับเล้าที่มีการขยายกกเร็วเกินไปหรือโรคกลุ่มอาการดูดซึมอาหารบกพร่อง (เอ็ม.เอ.เอส.)

ในช่วงท้ายของระยะกกลูกไก่ ลูกไก่มารวมกันแออัด อันเนื่องมาจากวัสดุรองพื้นเปียก ชื้น อับ มีกลิ่นแอมโมเนีย ทำให้ไก่ย้ายไปอยู่ที่ที่วัสดุรองพื้นแห้ง จุดนี้จะทำให้ไก่มีอาการเครียด มีโอกาสติดเชื้อได้ง่าย ดังนั้นผู้เลี้ยงจะต้องทำให้วัสดุรองพื้นแห้งอยู่เสมอเพื่อให้ไก่อยู่สบาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
104098

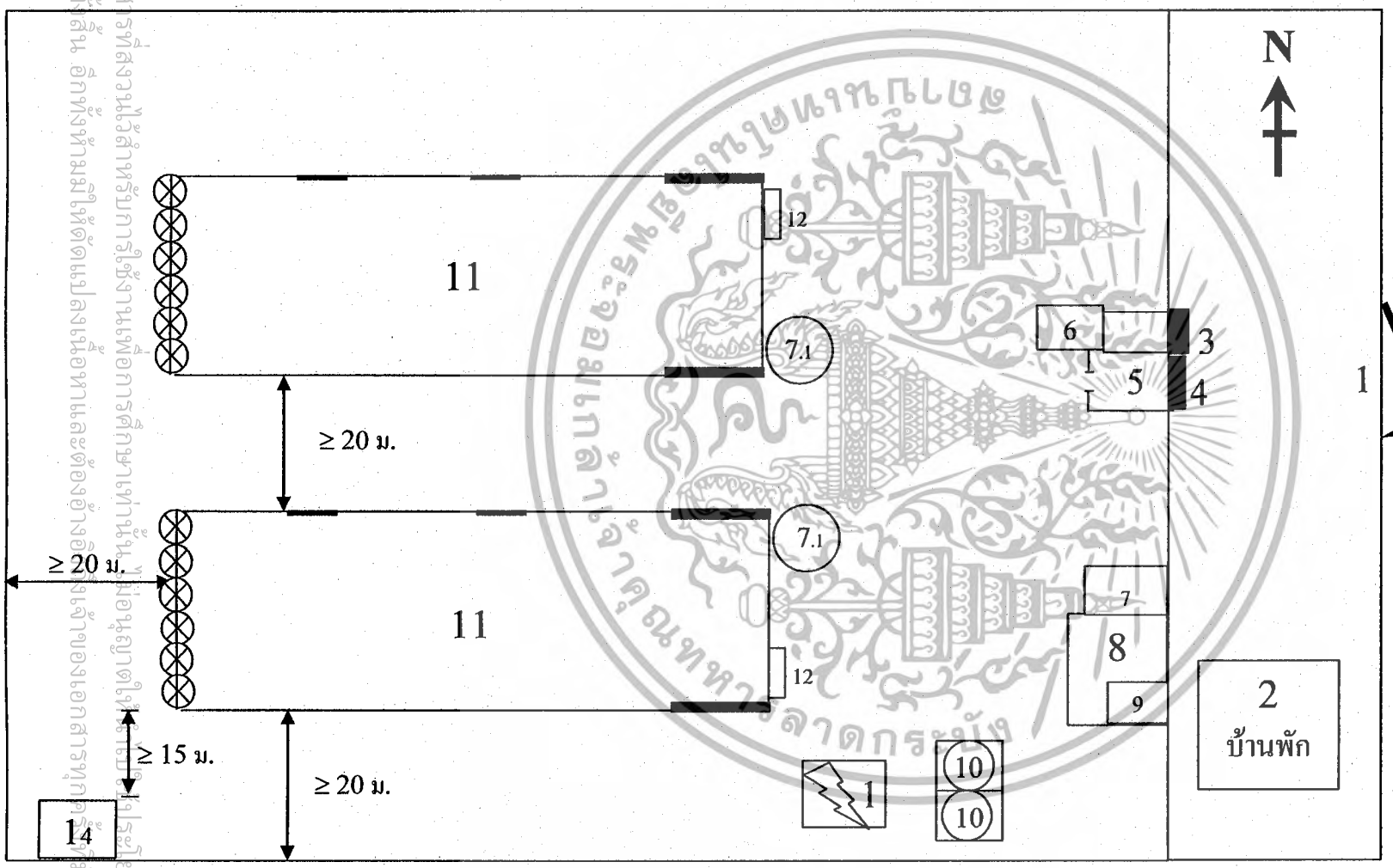
2.6 ความแตกต่างระหว่างรูปแบบฟาร์มไก่เนื้อแบบโรงเรือนปิดและปิด

2.6.1รูปแบบฟาร์มไก่เนื้อแบบโรงเรือนปิด

1. รั้วควรอยู่ห่างจากโรงเรือนอย่างน้อย 20 เมตร ความสูงรั้ว 1.5 เมตร สามารถป้องกันสัตว์อื่นเข้า-ออกได้
2. ประตูทางเข้าฟาร์มให้แยกสำหรับบุคคลเข้า-ออกและยานพาหนะ
   - ประตูทางเข้าฟาร์มสำหรับบุคคล ต้องมีห้องอาบน้ำเปลี่ยนเสื้อผ้า/รองเท้า และสเปรย์ฆ่าเชื้อโรค
   - ประตูทางเข้าฟาร์มสำหรับยานพาหนะ มีเครื่องพ่นน้ำยาฆ่าเชื้อโรค (ชนิดแรงดันสูง/เครื่องปั้มปี๋) หรือบ่อน้ำยาฆ่าเชื้อโรค หรือโรงพ่นน้ำยาฆ่าเชื้อโรค
3. โรงเก็บอาหาร / อุปกรณ์
   - แข็งแรง ถาวร มิดชิด สะอาด สามารถอบหรือรมควันฆ่าเชื้อโรคได้ และป้องกันนก หนู หรือแมลงได้
   - มีชั้นวางอาหารสูงจากพื้นไม่น้อยกว่า 10 เซนติเมตร
   - ตู้ยาอยู่ในห้องเก็บอุปกรณ์และสามารถล็อคได้
4. ลักษณะโรงเรือน
   - สร้างด้วยวัสดุที่คงทนถาวร
   - มีอ่างน้ำยาจุ่มเท้าหน้าประตูทางเข้าฟาร์ม
   - รอบโรงเรือนรัศมี 1 เมตร ให้เทพื้นซีเมนต์และมีรางระบายน้ำขนาดกว้าง 30 เซนติเมตร
5. บ้านพักอยู่นอกพื้นที่เลี้ยงสัตว์
6. พื้นที่ทำลายซากควรอยู่ท้ายฟาร์มและอยู่ห่างจากโรงเรือนอย่างน้อย 15 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนผังองค์ประกอบหลักฟาร์มไก่เนื้อ แบบโรงเรือนปิด



1. ประตูรั้ว ชั้นนอก
2. บ้านพัก อยู่นอกพื้นที่เลี้ยงสัตว์
3. ประตูทางเข้าฟาร์มสำหรับบุคคล
4. ประตูทางเข้าฟาร์มสำหรับยานพาหนะ
5. โรงสปรย์ฆ่าเชื้อยานพาหนะ
6. ห้องอาบน้ำ เปลี่ยนเสื้อผ้าและรองเท้า
7. โรงเก็บอาหาร หรือ
   7.1 ไซโล
8. โรงเก็บอุปกรณ์
9. ตู้เก็บยา
10. ถังเก็บน้ำ
11. โรงเรือนเลี้ยงไก่
12. อ่างน้ำยาจุ่มเท้าหน้าโรงเรือน
13. โรงเก็บเครื่องสำรองไฟฟ้า
14. พื้นที่ทำลายซาก

<u>หมายเหตุ</u> รูปแบบแผนผังฟาร์มอาจเปลี่ยนแปลงได้ตามความเหมาะสมของพื้นที่ แต่ต้องคงองค์ประกอบหลักไว้รวมถึงระยะห่างระหว่างโรงเรือน โรงเรือนกับรั้ว โรงเรือนกับพื้นที่ทำลายซาก

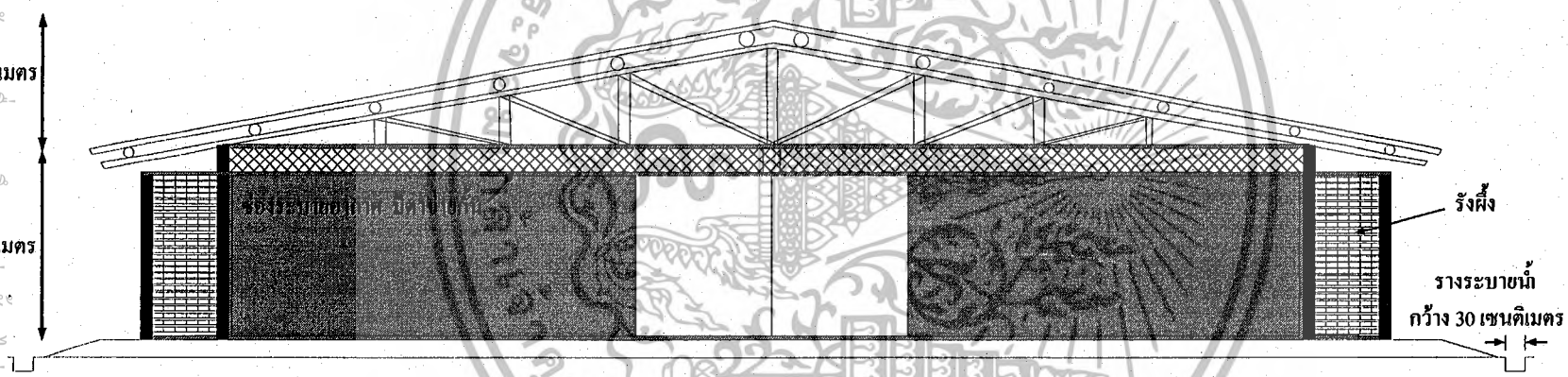
รูปที่ 2.1 แผนผังองค์ประกอบหลักฟาร์มไก่เนื้อ แบบโรงเรือนปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือแสวงหาประโยชน์ทางการค้า ในรูปแบบใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

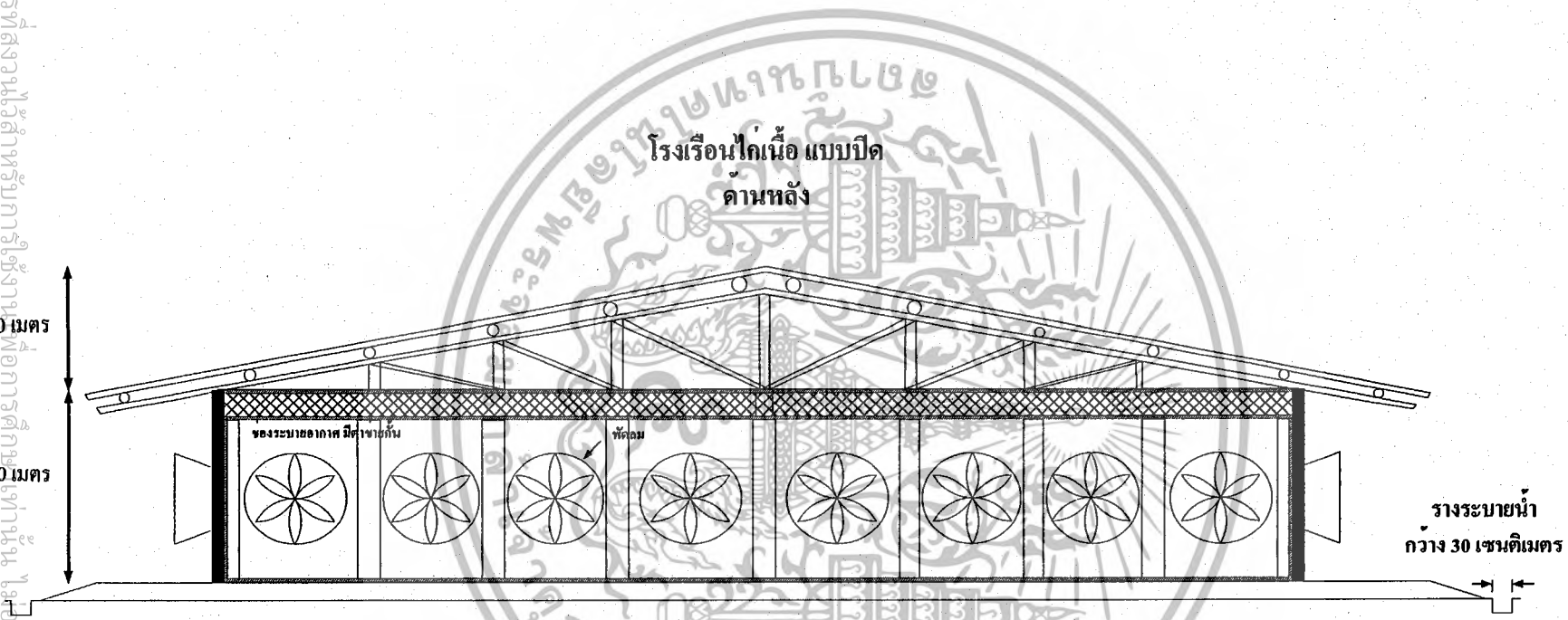
โรงเรือนไก่เนื้อ แบบปิด

ด้านหน้า



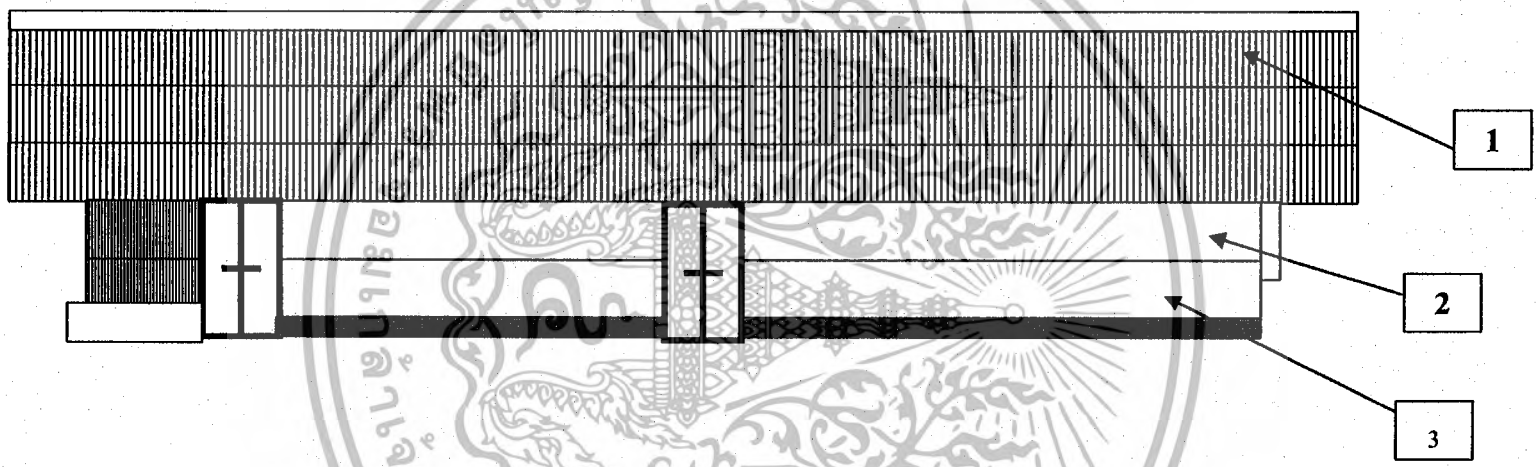
รูปที่ 2.2 โรงเรือนไก่เนื้อ แบบปิดด้านหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 โรงเรือนไก่เนื้อ แบบปิดด้านหลัง

รูปด้านข้างโรงเรือนไก่เนื้อแบบปิด



1. กระเบื้อง / สังกะสี
2. ผ้าม่านพลาสติก
3. ผนังปูนสูงจากพื้น 30 เซนติเมตร

รูปที่ 2.4 รูปด้านข้างโรงเรือนไก่เนื้อแบบปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

2.6.2รูปแบบฟาร์มไก่เนื้อแบบโรงเรือนเปิด

1. รั้วควรอยู่ห่างจากโรงเรือนอย่างน้อย 30 เมตร ความสูงรั้ว 1.5 เมตร สามารถป้องกันสัตว์อื่นเข้า-ออกได้

2. ประตูทางเข้าฟาร์มให้แยกสำหรับบุคคลเข้าออกยานพาหนะ

- ประตูทางเข้าฟาร์มสำหรับบุคคลต้องมีห้องอาบน้ำเปลี่ยนเสื้อผ้า/รองเท้าและสเปรย์ฆ่าเชื้อโรค
- ประตูทางเข้าฟาร์มสำหรับยานพาหนะ มีเครื่องพ่นน้ำยาฆ่าเชื้อโรค (ชนิดแรงดันสูง / เครื่องปั้มปี๋) หรือบ่อน้ำยาฆ่าเชื้อโรค หรือโรงพ่นน้ำยาฆ่าเชื้อโรค

3. โรงเก็บอาหาร / อุปกรณ์

- แข็งแรง ถาวร มิดชิดสะอาด สามารถอบรมควันฆ่าเชื้อโรคได้ สามารถป้องกันนก หนู หรือแมลงได้
- มีชั้นวางสูงจากพื้นไม่น้อยกว่า 10 เซนติเมตร
- ตู้ยาอยู่ในห้องเก็บอุปกรณ์และสามารถล็อคได้

4. ลักษณะโรงเรือน

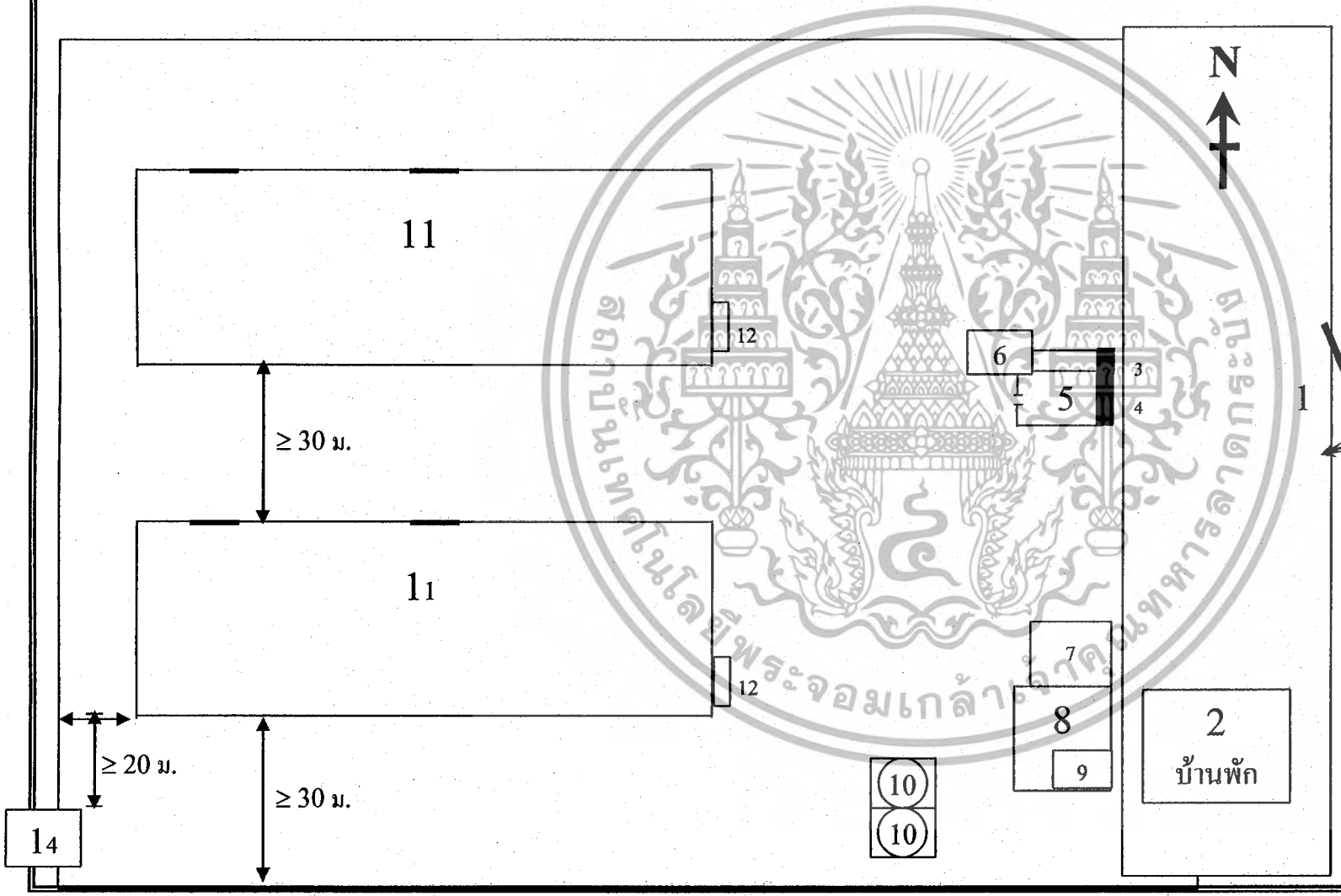
- สร้างด้วยวัสดุที่คงทนถาวร
- มีอ่างน้ำยาจุ่มเท้าหน้าประตูทางเข้าโรงเรือน
- ต้องติดตาข่ายขนาด 4 หุน (½ นิ้ว) ล้อมรอบโรงเรือนตั้งแต่หลังคาจรดพื้นเพื่อป้องกันไม่ให้สัตว์พาหะนำโรคเข้า โรงเรือนได้
- รอบโรงเรือนรัศมี 1 เมตร ให้เทพื้นซีเมนต์และมีรางระบายน้ำขนาดกว้าง 30 เซนติเมตร

5. บ้านพักอยู่นอกพื้นที่เลี้ยงสัตว์

6. พื้นที่ทำลายซากควรอยู่ท้ายฟาร์มและอยู่ห่างจากโรงเรือนอย่างน้อย 20 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนผังองค์ประกอบหลักฟาร์มไก่เนื้อ แบบโรงเรือนเปิด



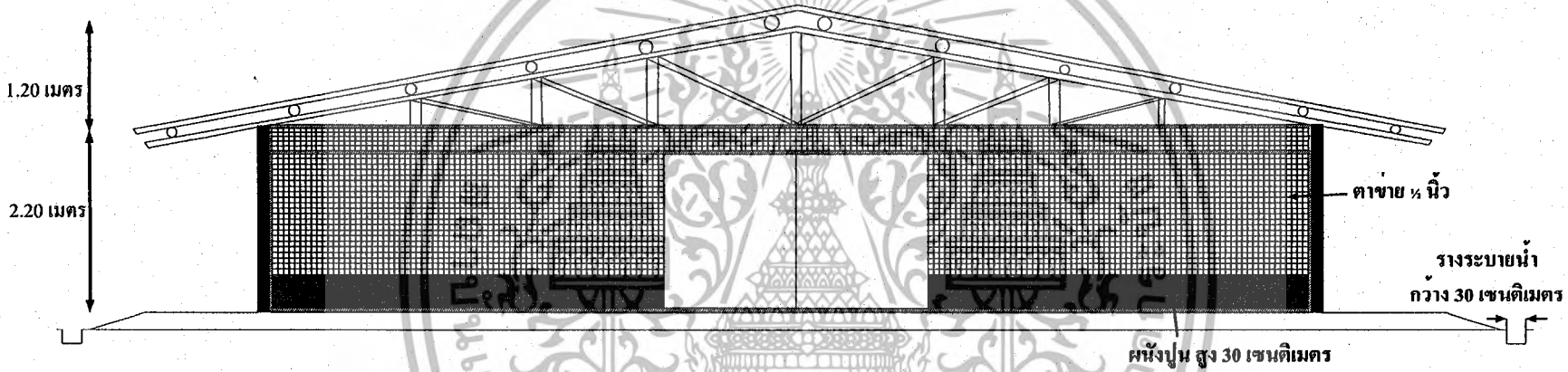
1. ประตูรั้ว ชั้นนอก
2. บ้านพัก อยู่นอกพื้นที่เลี้ยงสัตว์
3. ประตูทางเข้าฟาร์มสำหรับบุคคล
4. ประตูทางเข้าฟาร์มสำหรับยานพาหนะ
5. โรงสปรย์ฆ่าเชื้อยานพาหนะ
6. ห้องอาบน้ำ เปลี่ยนเสื้อผ้าและรองเท้า
7. โรงเก็บอาหาร
8. โรงเก็บอุปกรณ์
9. ตู้เก็บยา
10. ถังเก็บน้ำ
11. โรงเรือนเลี้ยงไก่
12. อ่างน้ำยาจุ่มเท้าหน้าโรงเรือน
13. พื้นที่ทำลายซาก

<u>หมายเหตุ</u> รูปแบบแผนผังฟาร์มอาจเปลี่ยนแปลงได้ตามความเหมาะสมของพื้นที่ แต่ต้องคงองค์ประกอบหลักไว้รวมถึงระยะห่างระหว่างโรงเรือน โรงเรือนกับรั้ว โรงเรือนกับพื้นที่ทำลายซาก

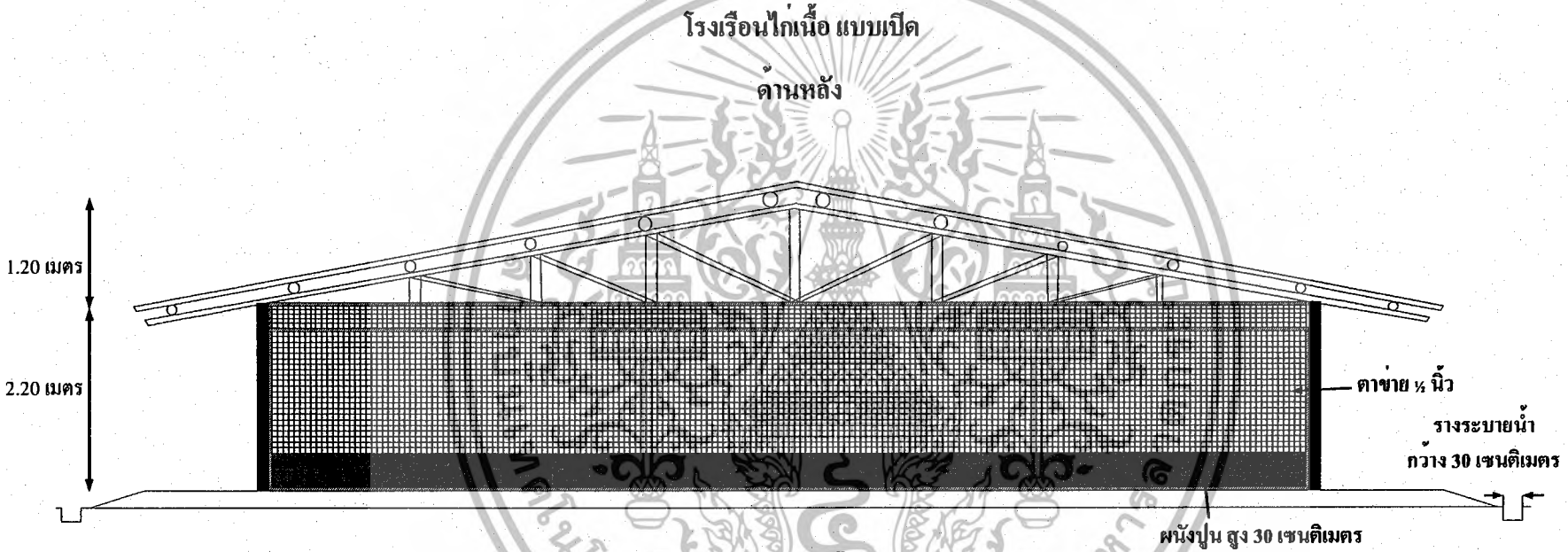
รูปที่ 2.5 แผนผังองค์ประกอบหลักฟาร์มไก่เนื้อ

โรงเรือนไก่เนื้อ แบบเปิด

ด้านหน้า

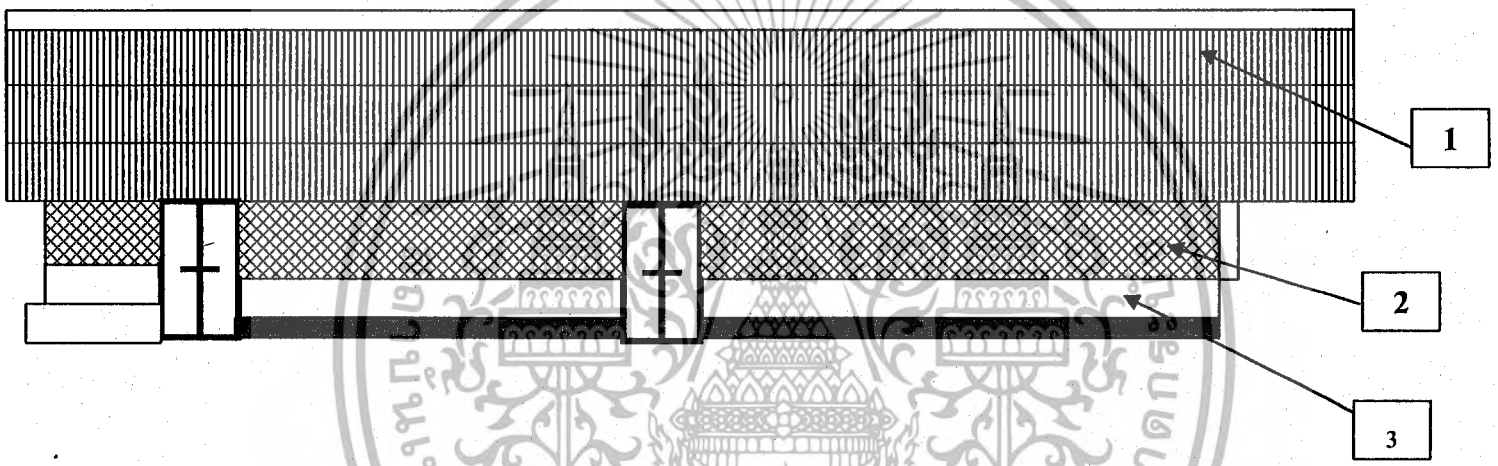


รูปที่ 2.6 โรงเรือนไก่เนื้อ แบบเปิดด้านหน้า



รูปที่ 2.7 โรงเรือนไก่เนื้อ แบบเปิดด้านหลัง

รูปด้านข้างโรงเรือนไก่เนื้อแบบเปิด



1. กระเบื้อง / สังกะสี
2. ตาข่ายขนาด 4 หุน (½ นิ้ว)
3. ผนังปูนสูงจากพื้น 30 เซนติเมตร

รูปที่ 2.8 รูปด้านข้างโรงเรือนไก่เนื้อแบบเปิด

2.7 สถาปัตยกรรมของซีพียู ARM 7[1]

2.7.1 ประวัติความเป็นมาของซีพียู ARM

บริษัท Acorn Computer Limited ในเมืองแคมบริดจ์ ประเทศอังกฤษเป็นบริษัทแรกที่ออกแบบซีพียู ARM ในช่วงเดือนตุลาคม ค.ศ. 1983 ถึงเมษายน ค.ศ.1985 โดยสร้างต้นแบบของซีพียู ARM ในวันที่ 26 เมษายน ค.ศ.1985 จากนั้นได้ถูกส่งไปผลิตที่ บริษัท VLSI Technology Inc ประเทศสหรัฐอเมริกา[16]

บริษัท Acorn เป็นบริษัทชั้นนำผู้ผลิตเครื่องไมโครคอมพิวเตอร์ BBC microcomputer ที่ได้รับความนิยมแพร่หลายอย่างมากในประเทศอังกฤษตั้งแต่เริ่มวางจำหน่ายในเดือน มกราคม ค.ศ.1982 เครื่อง BBC microcomputer เป็นเครื่องที่ทำงานโดยใช้ซีพียู 8 บิต เบอร์ 6502

หลังจากที่ประสบความสำเร็จกับเครื่อง BBC microcomputer แล้วทางบริษัท Acorn ได้คิดพัฒนาซีพียูสำหรับเครื่องไมโครคอมพิวเตอร์ยุคถัดไป โดยช่วงที่กำลังพัฒนาในปี ค.ศ. 1983 นั้น ซีพียู 16 บิตที่มีใช้งานแบบ CISC (Complex Instruction Set Computer) ที่มีความซับซ้อนมาก ทำงานช้ากว่าหน่วยความจำที่มีในยุคนั้น ทางวิศวกรของบริษัทจึงคิดออกแบบซีพียูเพื่อใช้งานทางการค้าเอง ซึ่งถ้าออกแบบทั้งหมดจะต้องใช้เวลาและแรงงานมาก ทางวิศวกรผู้ออกแบบจึงได้นำมาซีพียู Berkeley RISC I ซึ่งเป็นซีพียูแบบ RISC (Reduce Instruction Set Computer) ขนาด 32 บิต ที่ถูกออกแบบโดยนักศึกษาปริญญาโทของมหาวิทยาลัย Berkeley ที่เป็นซีพียูแบบง่ายๆ ไม่ซับซ้อน ใช้งานด้านการศึกษามาพัฒนาต่อให้เป็นซีพียู 32 บิตอเนกประสงค์เพื่อใช้งานทางการค้า ซีพียู ARM มีข้อดีในเรื่องสถาปัตยกรรมที่ไม่ซับซ้อน ประหยัดพื้นที่ในการผลิตชิปซีพียู กินพลังงานน้อย โดยที่ยังคงมีสมรรถนะที่สูง

หลังจากที่ออกแบบ และทดสอบจนพอใจแล้ว ได้มีการก่อตั้งบริษัท Advanced RISC Machines Ltd. ขึ้นในเดือนพฤศจิกายนปีค.ศ.1990 ในประเทศอังกฤษ โดยเป็นบริษัทร่วมทุนระหว่างบริษัท Apple Computer, Acorn Computer Group และ VLSI Technology โดยบริษัท Apple และ VLSI Technology เป็นผู้ลงทุนโดยบริษัท Acorn เป็นผู้อนุญาตให้ใช้เทคโนโลยีและโอนทีมวิศวกรผู้ออกแบบจำนวน 12 คน

ซีพียู ARM รุ่นแรกที่ออกจำหน่ายในปี 1991 เป็นซีพียูตระกูล ARM6 จัดว่าเป็นซีพียู RISC 32 บิตแบบฝังตัว (embedded RISC) ตัวแรกของโลก นับเป็นการสร้างมาตรฐานใหม่ให้กับวงการซีพียู 32 บิต หลังจากนั้นก็มีบริษัทผู้ผลิตซีพียูไมโครโปรเซสเซอร์ และไมโครคอนโทรลเลอร์ทั่วโลกอื่นๆ เข้าร่วมผลิตซีพียูตระกูล ARM ในปัจจุบันมีซีพียู ARM ต่างๆ ดังนี้ ARM7, ARM9, ARM9E, ARM10 และ ARM11

---

[1] จากที่มา http://www.ett.co.th/product/ARM/CP_JR_ARM7_USB_LPC2148.htm
เพื่อให้เกิดความสะดวกสำหรับผู้อ่านในการทบทวนพื้นฐานจึงอ้างถึง บทความต้นฉบับดังกล่าว ณ ที่นี้อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในยุคแรกซีพียู ARM เป็นซีพียู RISC 32 บิต การทำงานจำเป็นต้องต่อกับหน่วยความจำและอุปกรณ์ภายนอก เมื่อมีบริษัทผู้ผลิตไมโครคอนโทรลเลอร์จำนวนมากได้นำลิขสิทธ์ของซีพียู ARM ไปพัฒนาต่อ ได้มีการเพิ่มหน่วยความจำภายในทั้ง ROM และ RAM และเพิ่มโมดูลอุปกรณ์เสริมต่างๆ เช่น วงจรสื่อสารแบบอนุกรม วงจรแปลงดิจิตอลเป็นแอนะลอก ฯลฯ ทำให้กลายเป็นไมโครคอนโทรลเลอร์แบบ 32 บิตที่กินพลังงานต่ำ สามารถทำงานได้โดยใช้ชิปไอซีเพียงตัวเดียวโดยไม่ต้องต่ออุปกรณ์เพิ่มเติมภายนอก ทำให้มีการนำไปใช้ในงานคอมพิวเตอร์ฝังตัว (embedded computer) มากขึ้น

2.7.2 สถาปัตยกรรมซีพียู ARM7

สถาปัตยกรรม ARM7 เป็นซีพียูแบบ RISC ขนาด 32 บิต ภายในมีบัสขนาด 32 บิตตัวเดียวที่ใช้สำหรับรับส่งข้อมูล และคำสั่ง ชุดคำสั่งจะมีขนาด 32 บิตคงที่ ในขณะที่ข้อมูลสามารถเลือกได้ว่าจะมีขนาด 8, 16 หรือ 32 บิต โดยแสดงแกนกลาง (core) ของซีพียู ARM7

โครงสร้างของ ARM7 จะเป็นแบบที่เรียบง่าย มีชุดคำสั่งไม่มากนัก ประหยัดพื้นที่สารกึ่งตัวนำที่ใช้สร้าง ประหยัดพลังงาน

สถาปัตยกรรมของ ARM7 จะเป็นแบบ load-and-store ในการประมวลผลข้อมูลใดๆต้องกระทำผ่านทางรีจิสเตอร์ เริ่มต้นด้วยการโหลดค่าจากหน่วยความจำเก็บในรีจิสเตอร์นำค่ามาประมวลผล เสร็จแล้วจะเขียนค่าเก็บในหน่วยความจำดังเดิม

รีจิสเตอร์ของ ARM7 ที่ใช้งานได้สำหรับผู้ใช้มีทั้งหมด 16 ตัวคือ R0-R15 โดยทุกตัวมีขนาด 32 บิต โดย R0-R12 เป็น รีจิสเตอร์ทั่วไปที่ไม่ได้กำหนดหน้าที่การทำงานพิเศษ ส่วน R12 ทำหน้าที่เป็น stack pointer (sp) R14 ทำหน้าที่เป็น link register (LR) และ R15 ทำหน้าที่เป็น Program Counter (PC)

2.7.3 ระบบการทำงานของ ARM7

ขั้นตอนการออกแบบชุดคำสั่งของ ARM7 นั้นเป็นชนิด RISC (reduce instruction set computer) ที่มีการออกแบบให้มีคำสั่งขนาดเล็ก ทำให้สามารถประมวลผลได้รวดเร็ว โดยเฉพาะใน ARM7 นี้คำสั่งที่ออกแบบนั้นสามารถทำงานได้เสร็จเพียงใน 1 รอบการทำงาน (single cycle) เท่านั้น

2.7.3.1 ไปป์ไลน์

ไปป์ไลน์(pipeline)หรือการทำงานแบบสายท่อของ ARM7 นั้นมีการออกแบบไปป์ไลน์ของชุดคำสั่งเอาไว้ 3 สเตจ (stage) คือ ระยะของการอ่านชุดคำสั่ง(fetch)ระยะการถอดรหัสของชุดคำสั่ง (decode)และระยะของการทำงานตามชุดคำสั่ง(execute)ซึ่งการออกแบบด้วยวิธีการนี้มีข้อดีคือทำให้สามารถเรียกคำสั่งได้หลายคำสั่งมาซ้อนกันได้ ดังรูปที่ 2-9 จะเห็นว่าในช่วงแรกของการทำงานรอบที่ 1 นั้นจะมีการอ่านชุดคำสั่งที่ 1 เข้าไปสู่ไปป์ไลน์ หลังจากนั้นเมื่อทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถอดรหัสชุดคำสั่งที่ 1 ระยะการอ่านจะว่างจึงทำการอ่านชุดคำสั่งที่ 2 เมื่อทำการทำตามชุดคำสั่งที่ 1 อยู่นั้นชุดคำสั่งที่ 2 ก็จะถูกถอดรหัสและชุดคำสั่งที่ 3 จะถูกอ่านเข้ามา และเมื่อทำไปเรื่อยๆจะเห็นว่าจะสามารถทำงานได้ 4 คำสั่งโดยใช้การทำงานเพียง 2 รอบเท่านั้น

	ชุดคำสั่งที่ 1	ชุดคำสั่งที่ 2	ชุดคำสั่งที่ 3	ชุดคำสั่งที่ 4
การทำงานรอบที่ 1	Fetch	รอ	รอ	รอ
	Decode	Fetch	รอ	รอ
	Execute	Decode	Fetch	รอ
การทำงานรอบที่ 2	ทำงานเสร็จ	Execute	Decode	Fetch
	ทำงานเสร็จ	ทำงานเสร็จ	Execute	Decode
	ทำงานเสร็จ	ทำงานเสร็จ	ทำงานเสร็จ	Execute
	ทำงานเสร็จ	ทำงานเสร็จ	ทำงานเสร็จ	ทำงานเสร็จ

รูปที่ 2.9 ไปป์ไลน์ของ ARM7 ทั้ง 3 ขั้นตอน

2.7.3.2 รีจิสเตอร์

รีจิสเตอร์ (Register) เป็นหน่วยความจำที่อยู่ในหน่วยประมวลผล และตัว ARM7 มีสถาปัตยกรรมการทำงานแบบ โหลดและสโตร์ (load and store) ซึ่งหมายความว่า ข้อมูลที่จะใช้ในการประมวลผลนั้นจะต้องถูกนำเข้ามาเก็บในรีจิสเตอร์แล้วจึงทำการประมวลผลและเมื่อได้ทำการประมวลผลเสร็จ ผลลัพธ์ที่ได้นั้นจะถูกนำไปเก็บเอาไว้ในรีจิสเตอร์ ซึ่งสรุปการทำงานได้ 3 ขั้นตอนดังนี้

ขั้นตอนที่ 1 ทำการโหลดข้อมูลมาเก็บในรีจิสเตอร์

ขั้นตอนที่ 2 ทำการประมวลผลจากข้อมูลในรีจิสเตอร์

ขั้นตอนที่ 3 นำผลลัพธ์ที่ได้เก็บในรีจิสเตอร์ปลายทาง

เช่น ถ้าต้องการหาผลบวกของ M1 กับ M2 แล้วนำผลลัพธ์มาเก็บใน M3 สามารถนำมาเขียนเป็น 3 ขั้นตอนได้ดังนี้

ขั้นตอนที่ 1 ทำการโหลด M1 ไปเก็บในรีจิสเตอร์ (สมมุติว่าเป็น R1)

ทำการโหลด M2 ไปเก็บในรีจิสเตอร์ (สมมุติว่าเป็น R2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 2 เรียกคำสั่งบวก นั้นคือ ADD R3, R1, R2 (สมมุติว่าเก็บผลลัพธ์ ใน R3) คำสั่งนี้มีความหมายว่า R3 = R1 + R2

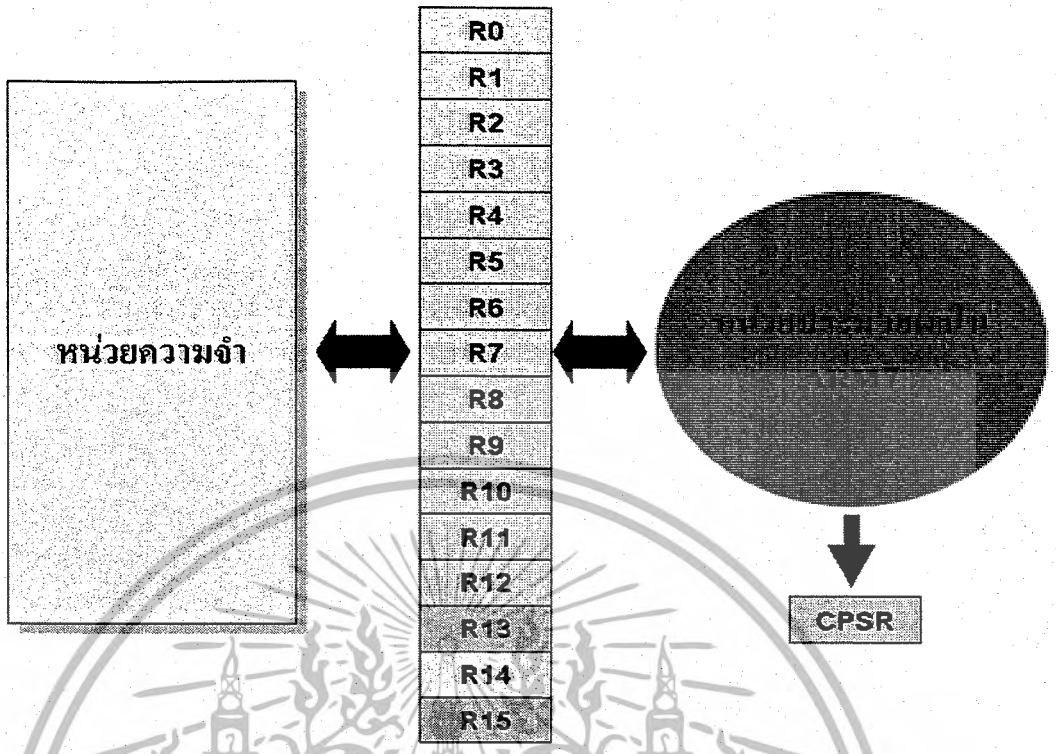
ขั้นตอนที่ 3 โอนข้อมูลจากรีจิสเตอร์ (R3) มาเก็บใน M3

รีจิสเตอร์สำหรับผู้ใช้ของหน่วยประมวลผล ARM7 นั้นจะมีขนาด 32 บิต ซึ่งมีทั้งหมด 17 ตัวคือ R0 ถึง R15 และ CPSR (current program status register) โดยจำแนกกลุ่มทำงานได้ดังนี้

- R0 ถึง R12 สามารถใช้งานได้ตามที่ผู้ใช้หรือเขียนโปรแกรมต้องการ
- R13 ถูกใช้เป็นที่เก็บค่าตัวชี้ไปยังหน่วยความจำสแตก (stack pointer)
- R14 หรือเรียกว่า LR (link register) ถูกใช้เป็นที่เก็บตำแหน่งของชุดคำสั่งถัดไปที่จะต้องประมวลผลก่อนที่จะเกิดการเรียกโปรแกรมย่อย (call) ซึ่งเป็นชุดคำสั่งที่จะต้องนำมาทำงานหลังจากที่โปรแกรมย่อยนั้นทำงานเสร็จแล้ว
- R15 หรือเรียกว่า PC (program counter) ถูกใช้เป็นที่เก็บตำแหน่งของคำสั่งถัดไปที่จะถูกนำมาประมวลผล
- CPSR ใช้เป็นที่เก็บสถานการณ์ทำงานของคำสั่งที่ถูกประมวลผลไปหลังสุด ซึ่งนิยมเรียกรีจิสเตอร์นี้กันว่า เรจิสเตอร์สถานะ หรือ รีจิสเตอร์แฟล็ก (flag register)

จากรีจิสเตอร์ที่กล่าวมานี้ สามารถนำมาเขียนความสัมพันธ์ระหว่างกันได้ดังรูปที่ 2-11 ซึ่งจะเห็นว่าหน่วยความจำนั้นจะต้องถ่ายโอน/แลกเปลี่ยนข้อมูลกับรีจีสเตอร์ โดยการประมวลผลนั้น ตัวประมวลผลจะใช้ข้อมูลจากรีจิสเตอร์ประกอบการทำงาน และเมื่อทำงานเสร็จจะบันทึกสถานะของการทำงานจากคำสั่งปัจจุบันเก็บเอาไว้ใน CPSR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 ความสัมพันธ์ระหว่างหน่วยความจำ รีจิสเตอร์และหน่วยประมวลผลของARM7

2.7.3.3 รีจิสเตอร์เก็บสถานการณ์ทำงาน

รีจิสเตอร์เก็บสถานการณ์ทำงานของ ARM7 จะมีขนาด 32 บิตมีหน้าที่รายงานและใช้ควบคุมการทำงานของหน่วยประมวล ARM7 นั้นคือ เมื่อหน่วยประมวลผลทำงานเสร็จจะเก็บสถานการณ์ทำงานใน CPSR แล้วผู้เขียนโปรแกรมสามารถนำค่าเหล่านี้มาใช้เป็นการควบคุมโปรแกรมที่เขียนเพื่อสั่งงานหน่วยประมวลผล หรืออาจจะกำหนดสถานะบางอย่างเพื่อให้หน่วยประมวลผลทำงานภายใต้ภาวะ (mode) ที่ผู้เขียนโปรแกรมกำหนด

โครงสร้างของรีจิสเตอร์ตัวนี้เป็นดังรูปที่ 2-11 แบ่งได้ 2 กลุ่ม คือ

1. กลุ่มรายงานสถานการณ์ทำงาน ได้แก่

- N (negative) เป็น 1 เมื่อผลการทำงานก่อให้เกิดค่าลบ
- Z (zero) เป็น 1 เมื่อผลการทำงานทำให้เกิดค่าศูนย์
- C (carry) เป็น 1 เมื่อผลการทำงานทำให้เกิดการทดค่า
- V (overflow) เป็น 1 เมื่อผลการทำงานเกิดล้นของข้อมูล

2. กลุ่มกำหนดการทำงานของหน่วยประมวลผล ได้แก่

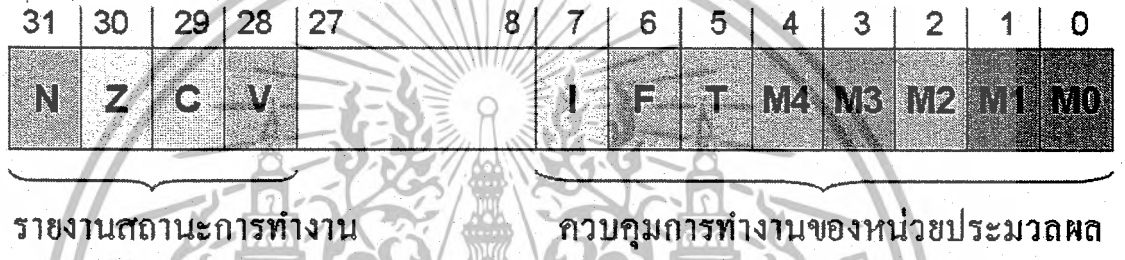
- I (interrupt enable) กำหนดให้เป็น 1 เมื่อต้องการดักสัญญาณขัดจังหวะแบบ IRQ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- F (fast interrupt enable) กำหนดให้เป็น 1 เมื่อต้องการดักสัญญาณขัดจังหวะแบบ FIQ

- T (thumb instruction set) กำหนดเป็น 1 เมื่อต้องการให้ทำงานชุดคำสั่งแบบ 16 บิต ซึ่งจะทำให้ ARM7 สามารถประมวลผลคำสั่งได้ 2 คำสั่งในการทำงานเพียง 1 รอบการทำงาน

- M4, M3, M2, M1, M0 ใช้ในการกำหนดภาวการณ์ทำงาน ซึ่งมีทั้งหมด 7 โหมด (mode) ที่มีผลต่อการใช้งาน รีจิสเตอร์แตกต่างกันไป แต่อย่างไรก็ดี ในการทำงานแบบโหมดผู้ใช้ (user mode) จะมีการทำงานเหมือนกับที่ได้อธิบายเอาไว้ในหัวข้อรีจิสเตอร์

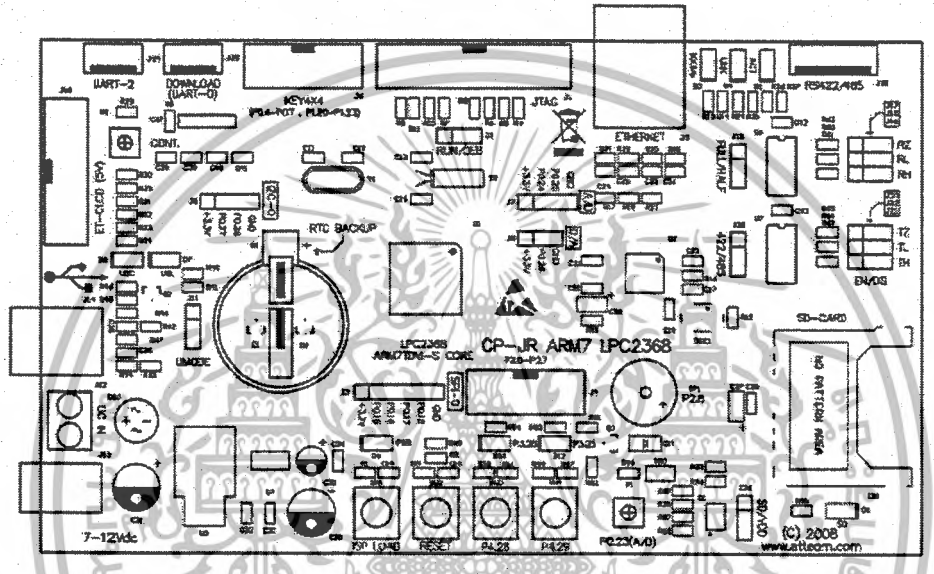


รูปที่ 2.11 ความหมายของบิตในรีจิสเตอร์สถานการณ์ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 ออกแบบโปรแกรมในส่วนของ ARM7 โดยนำโปรแกรมควบคุมการทำงานมาประยุกต์ใช้ร่วมกัน

โดยเลือกใช้ Microcontroller ARM7 LPC 2368 ของ บริษัทอีทีที ซึ่งมีคุณสมบัติและรูปแบบดังนี้สามารถ RUN ได้ทั้งในแบบ 16 BIT และ 32 BIT ออกแบบบอร์ดให้ใช้ในงานควบคุมต่างๆ สามารถ IN-CIRCUIT DOWNLOAD โปรแกรมเข้าหน่วยความจำภายในได้โดยตรงทาง PORT RS232



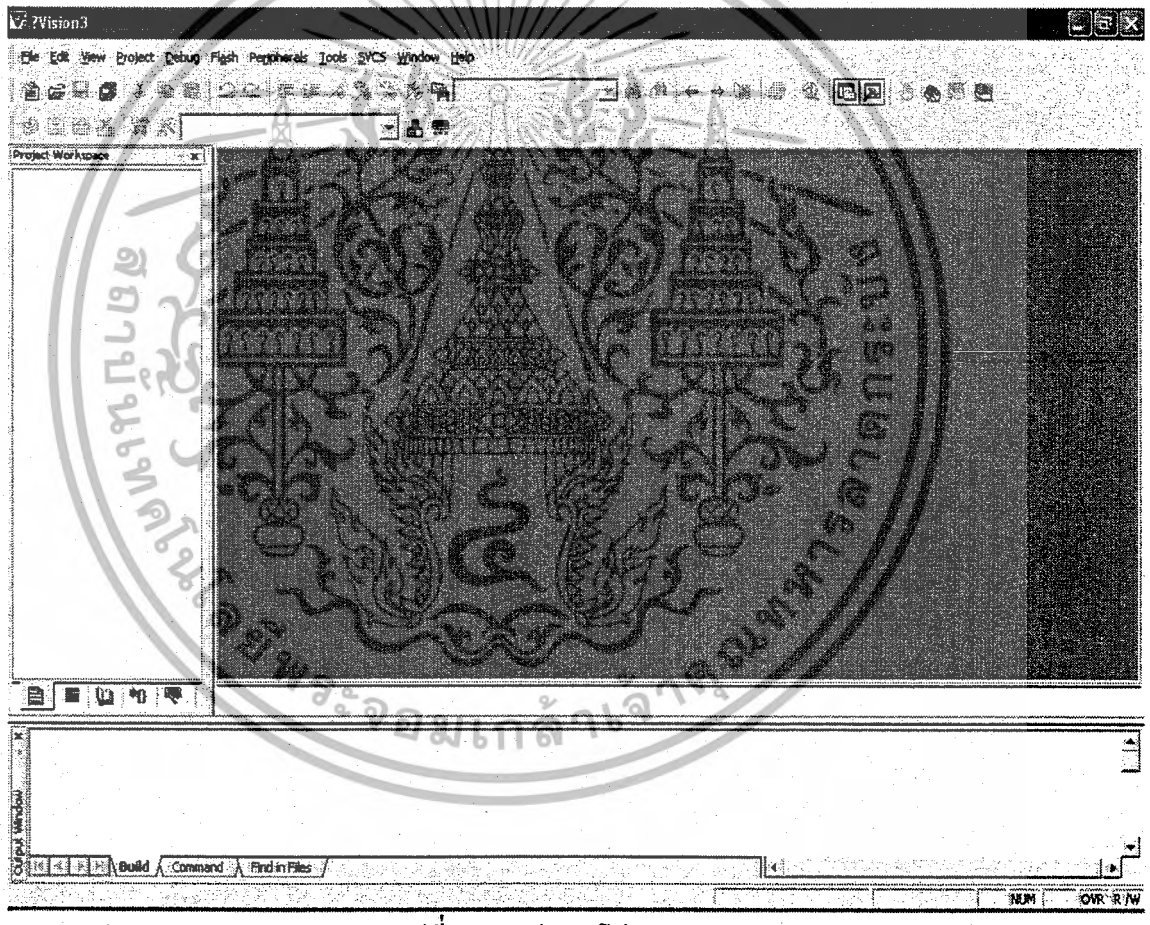
รูปที่ 2.12 รูป Microcontroller ARM7 LPC 2148

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.1 เลือกใช้โปรแกรม Keil uVision3

ตัวอย่างการใช้ Keil uVision3 ในการสร้าง Project File ของ Keil ARM ในที่นี้จะขอแสดงแนวทางการเขียนโปรแกรมภาษาซี โดยใช้ Keil-CARM ในการแปลคำสั่งภายใต้ โปรแกรม Text Editor ของ Keil (Keil uVision3) โดยจะขออธิบายถึงเฉพาะวิธีการกำหนดค่า Option สำหรับ เชื่อมโยงคำสั่งในการสั่งแปลโปรแกรมด้วย Keil-CARM ผ่านทาง Keil uVision3 เท่านั้น ส่วนรายละเอียดคำสั่งและ การใช้งานฟังก์ชั่นต่างๆในการเขียนโปรแกรมของ Keil-CARM นั้นขอให้ผู้ใช้ศึกษาจากคู่มือคำสั่งของ Keil-C ARM เอง โดยวิธีการกำหนดค่าตัวเลือกของ Keil uVision3 ให้ใช้งานกับ Keil-CARM นั้นมีขั้นตอนพอสังเขปดังนี้คือ

1. เปิดโปรแกรม Keil uVision3 ซึ่งเป็นโปรแกรม Text Editor ของ Keil-CARM ใช้สำหรับใช้ในการเขียนโปรแกรมที่เป็น Source Code ภาษาซี โดยจะมีลักษณะดังรูป

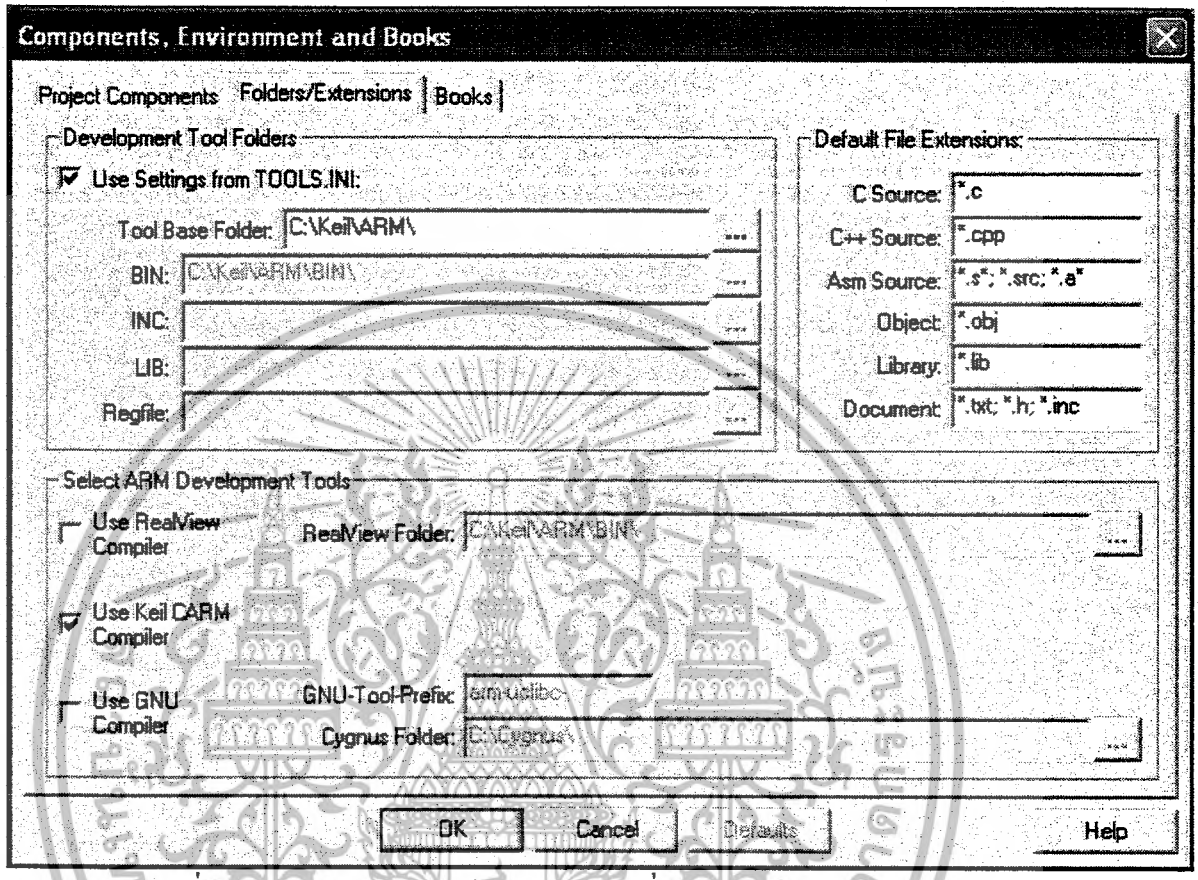


รูปที่ 2.13 รูปแบบโปรแกรม Keil uVision3

2. ทำการกำหนดค่าตัวเลือกในการแปลคำสั่งของ uVision3 ให้ใช้งานกับโปรแกรม Keil uVision3 และ Keil-CARM โดยให้เลือกคลิกเมาส์ที่เมนูคำสั่ง Project → Components, Environment, Books... จากนั้นให้เลือกค่าตัวเลือก สำหรับกำหนดการใช้งาน Complier จากหัวข้อ Select ARM Development Tools ซึ่งจะมีค่าตัวเลือกอยู่ 3 แบบคือ Use Keil-CARM Tools ,Use GNU Tools และ Use ARM Tools โดยให้เลือกเป็น "Use Keil ARM Tools" จากนั้นให้ทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

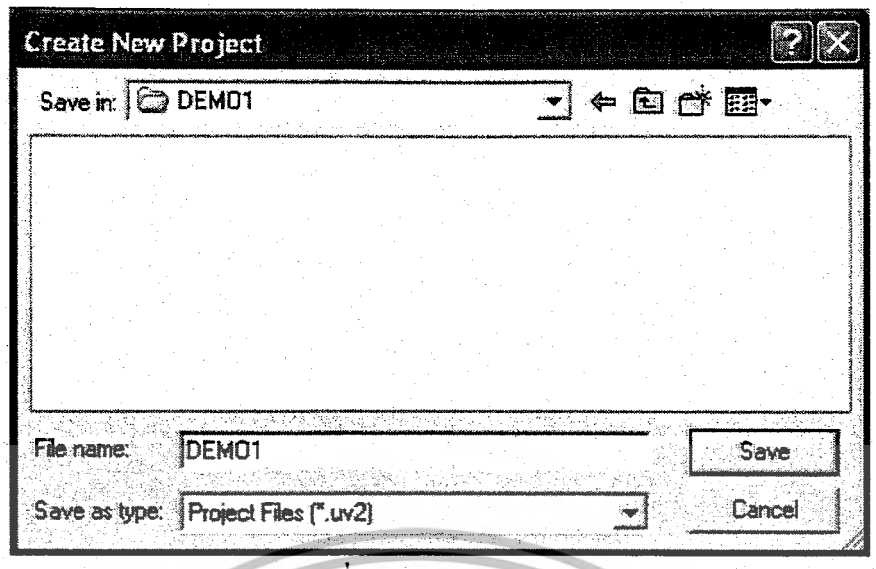
กำหนดตำแหน่ง Folder สำหรับเก็บค่าตัวเลือกการทำงานของโปรแกรม Keil ARM ซึ่งตามปกติแล้วจะเป็น "C:\Keil\ARM\" แต่ถ้าติดตั้ง Keil ไว้ที่อื่นก็ต้องปรับเปลี่ยนให้ถูกต้องตามความเป็นจริงด้วยดังรูป



รูปที่ 2.14 การกำหนดค่าตัวเลือกการแปลคำสั่งของโปรแกรม Keil uVision3

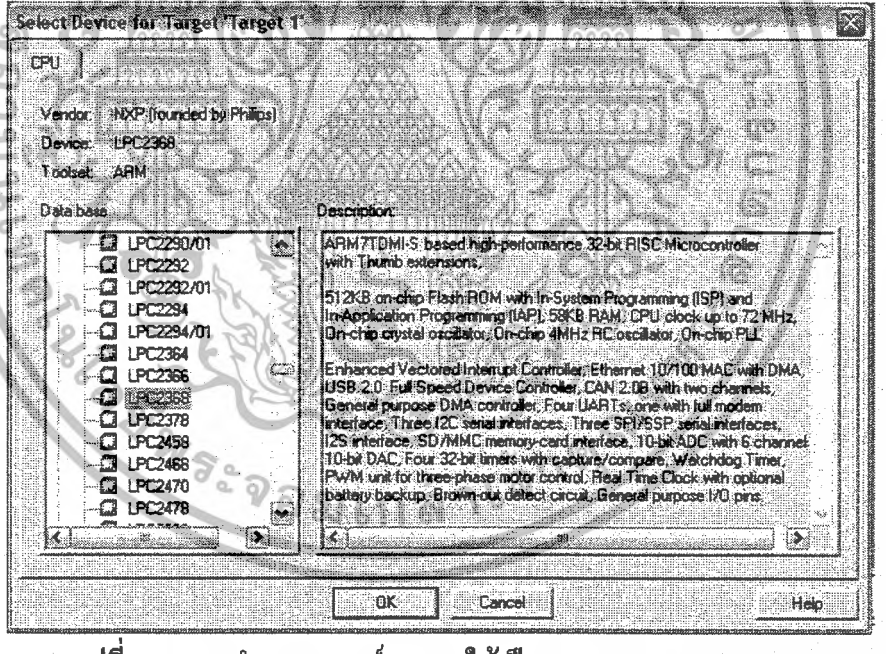
3. ทำการสร้าง Project File ขึ้นมาใหม่ โดยเรียกเมนูคำสั่ง Project ⟶ New Project จากนั้นให้เลือกกำหนดหรือ สร้างตำแหน่ง Folder ที่จะบันทึก Project File พร้อมกับกำหนดชื่อ Project File ตามต้องการ เช่น ถ้าต้องการสร้าง Project File ชื่อ DEMO1 โดยเก็บไว้ใน Folder ชื่อ DEMO1 ก็สามารถกำหนดตำแหน่ง Folder และชื่อ Project File ได้เอง โดยเมื่อกำหนดชื่อในช่อง File name แล้วให้เลือก Save เพื่อบันทึก Project File ไว้ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 การสร้างโปรเจคใหม่

หลังจากกำหนดชื่อและสั่ง Save Project File แล้ว โปรแกรมจะรอให้ผู้ใช้ทำการกำหนด เบอร์ MCU ที่จะใช้งานใน Project ที่สั่ง Save นั้น ซึ่งในกรณีที่ใช้งานกับบอร์ด CP-JR ARM7 USB-LPC2368 นั้น ให้เลือกกำหนดเบอร์ของ MCU เป็น LPC2368 ของ Philips แล้วเลือก OK ดังรูป

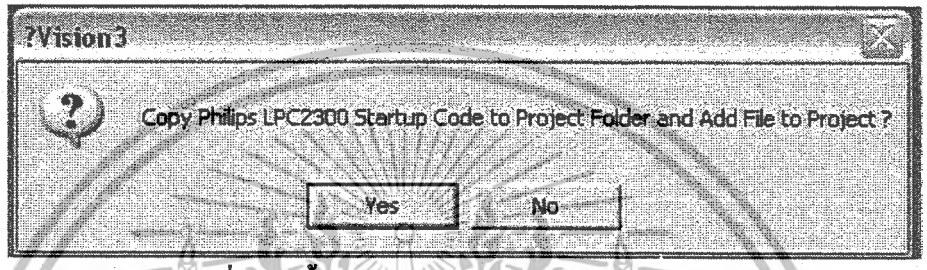


รูปที่ 2.16 การกำหนด เบอร์ MCU ให้เป็น LPC2368 ของ Philips

หลังจากเลือกกำหนดเบอร์ของ MCU เป็นที่เรียบร้อยแล้ว ในขั้นตอนนี้โปรแกรมจะรอให้ผู้ใช้ยืนยันว่า ต้องการจะทำการ Copy ไฟล์ Startup ของ Keil เพื่อใช้งานกับ MCU ของ Philips มาใช้ใน Project ด้วยหรือไม่ โดยStartup ไฟล์จะเป็นส่วนของการกำหนดค่าเริ่มต้นการทำงานให้กับ MCU เช่น การกำหนดค่า Stack และการ กำหนดค่าการทำงานให้กับ Phase-Lock-Loop ต่างๆ ก่อนที่จะเริ่มต้นทำงานตามโปรแกรมที่เราเขียนขึ้น ไม่เช่นนั้นแล้วโปรแกรมที่เรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขียนขึ้นมานั้นจะต้องเพิ่มคำสั่งในการเตรียมการทำงานส่วนเหล่านี้ให้ MCU เองทั้งหมดแต่เนื่องจากไฟล์ Startup ของ Keil-ARM นั้น เป็นไฟล์ภาษาแอสแซมบลี้ ซึ่งกำหนดค่าการทำงานไว้กับชุดพัฒนาของ Keil เอง ดังนั้นข้อกำหนดและการกำหนดค่าบางอย่างจะมีความแตกต่างกันอยู่กับค่าที่ต้องการสำหรับบอร์ด "CP-JR ARM7 USB-LPC2368" ไม่สามารถใช้งานไฟล์ Startup ได้ทันที ต้องมีการแก้ไขค่าตัวเลือกใหม่ดังนั้นก่อนที่จะใช้โปรแกรม Keil-CARM ในการแปลคำสั่งให้นั้น ผู้ใช้จะต้องเข้าไปแก้ไขไฟล์ Startup ใหม่โดยต้องกำหนดรูปแบบให้ถูกต้องตรงกับความต้องการของบอร์ดด้วย ดังนั้นในที่นี้ขอแนะนำให้เลือก "No" เพื่อไม่ให้ Keil uVision3 ทำการ Copy ไฟล์ Startup ของ Keil-CARM มาใช้ใน Project นี้ด้วย



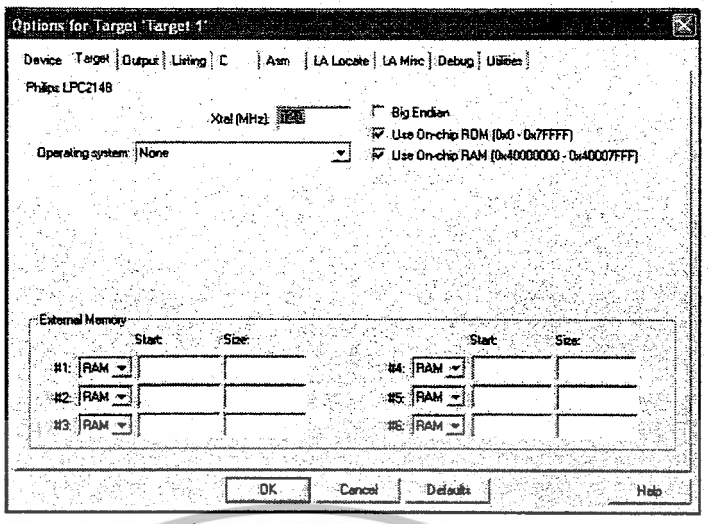
รูปที่ 2.17 ขั้นตอนการก๊อปไฟล์ Startup ของ Keil

4. ให้ทำการ Copy File ชื่อ "Startup.s" ที่ทาง อีทีที จัดเตรียมไว้ใน CD-ROM ซึ่งเก็บไว้ใน Example ชื่อ "Startup.s" มาไว้ในตำแหน่ง Folder เดียวกันกับ Project File ที่สร้างขึ้นมาใหม่นี้ โดยไฟล์ "Startup.s" จะเป็นไฟล์ซึ่งบรรจุคำสั่งภาษาแอสแซมบลี้ของ ARM7 สำหรับทำหน้าที่กำหนดค่า เริ่มต้นการทำงานที่จำเป็นให้กับ MCU ซึ่งได้แก่การ กำหนดค่า Stack ให้กับ MCU การ Initial Phase-Lock-Loop การกำหนดค่าให้กับ MAM Function และการกำหนดตำแหน่ง Vector ต่างๆของ MCU สำหรับใช้งานร่วมกับบอร์ด "CP-JR ARM7 USB-LPC2368" ซึ่งถ้าสั่ง Add ไฟล์ "Startup.s" จาก Keil หรือ Copy ไฟล์ดังกล่าวมาจากแหล่ง อื่นๆ อาจมีการทำงานของโปรแกรมใน Startup ไม่เหมือนกัน ซึ่งจะส่งผลต่อการทำงานของโปรแกรมด้วย

5. ให้ทำการกำหนดค่า Option ของ Project File โดยเลือกเมนูคำสั่ง Project ⟶ Option for Target 'Target 1'จากนั้นเลือกที่ Tab ของ Target เพื่อกำหนดค่าของ MCU Target โดยให้กำหนดดังนี้

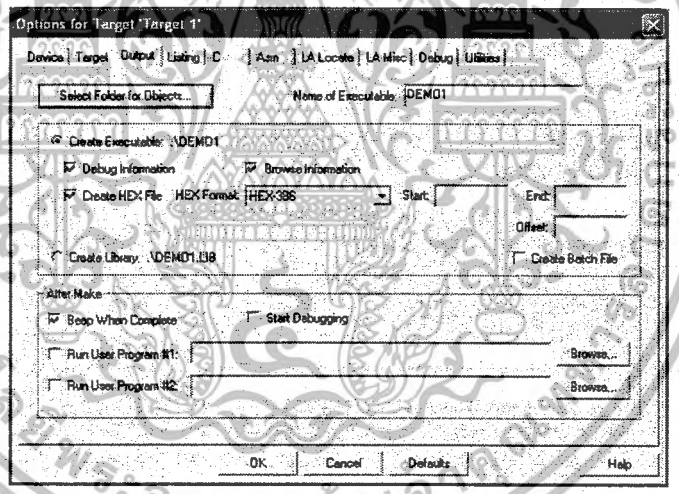
5.1 X-TAL ให้กำหนดเป็น 12 MHz พร้อมกับเลือกกำหนดให้ใช้หน่วยความจำที่มีอยู่ใน MCU เป็นเงื่อนไขในการแปลโปรแกรมของ Keil-CARM ด้วย ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 กำหนดค่า X-TAL ให้กับโปรแกรม Keil

5.2 Output ให้เลือกคลิกเมาส์ที่ค่าตัวเลือก Create HEX File พร้อมกับเลือกกำหนดรูปแบบของ Hex ให้เป็นแบบ HEX-386 แล้วเลือก OK ดังรูป



รูปที่ 2.19 การกำหนด Create Hex File ให้กับโปรแกรม Keil เท่านี้ก็จะสามารถเริ่มต้นเขียนโปรแกรม ใน ARM7 ได้

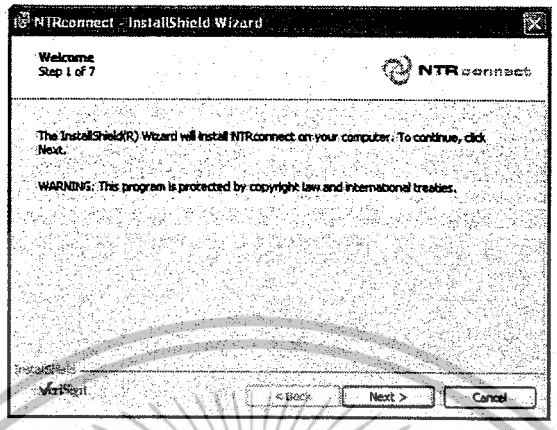
2.9 การควบคุมคอมพิวเตอร์ผ่านอินเตอร์เน็ตโดยการ remote access

โดยเราจะทำผ่านโปรแกรม NTRconnect Free 2.0 ซึ่งเป็นโปรแกรมของเว็บไซต์ www.ntrconnect.com ซึ่งโปรแกรม **NTRConnect** คือโปรแกรมสำหรับควบคุมเครื่องคอมพิวเตอร์จากที่อื่น เช่น ควบคุมเครื่องคอมพิวเตอร์ที่ฟาร์มจากที่บ้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

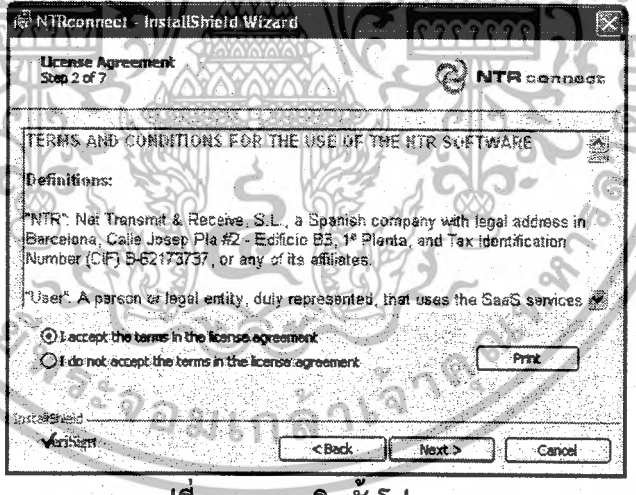
2.9.1 การติดตั้งโปรแกรม NTRConnect

ให้ทำการ double click ไอคอนเพื่อทำการติดตั้งโปรแกรม



รูปที่ 2.20 การติดตั้งโปรแกรม

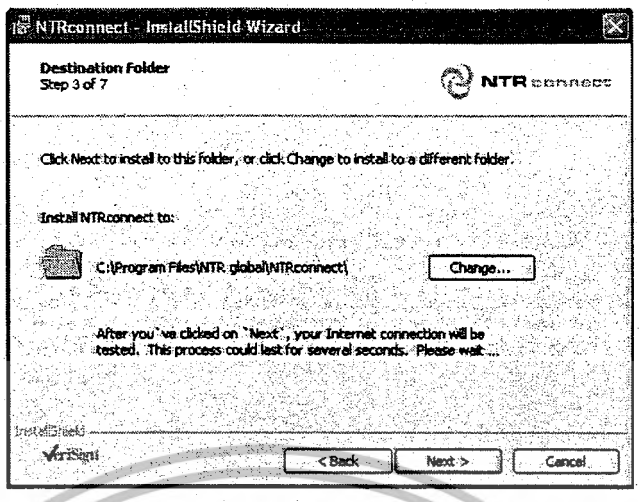
กดปุ่ม Next เพื่อไปหน้าต่อไป



รูปที่ 2.21 การติดตั้งโปรแกรม

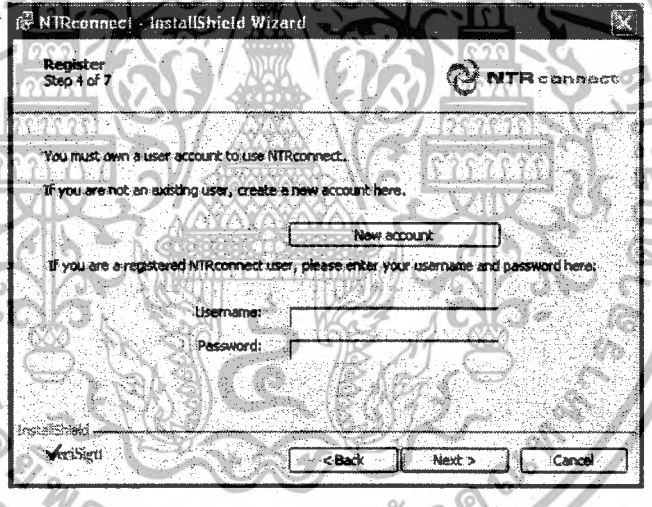
เลือก I accept the terms in the license agreement จากนั้นกดปุ่ม Next

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 การติดตั้งโปรแกรม

เลือกโฟลเดอร์ที่ต้องการติดตั้งโปรแกรม จากนั้นกดปุ่ม Next

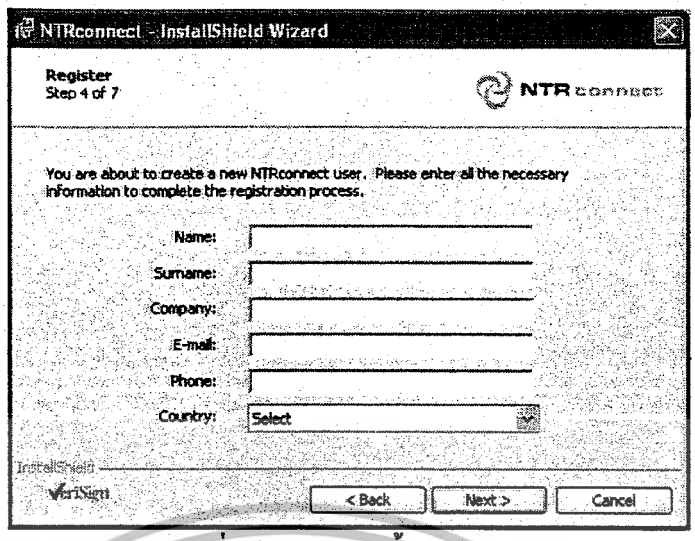


รูปที่ 2.23 การติดตั้งโปรแกรม

หากเคยสมัครสมาชิก สามารถใส่ข้อมูลชื่อและรหัสผ่านเก่าได้เลย

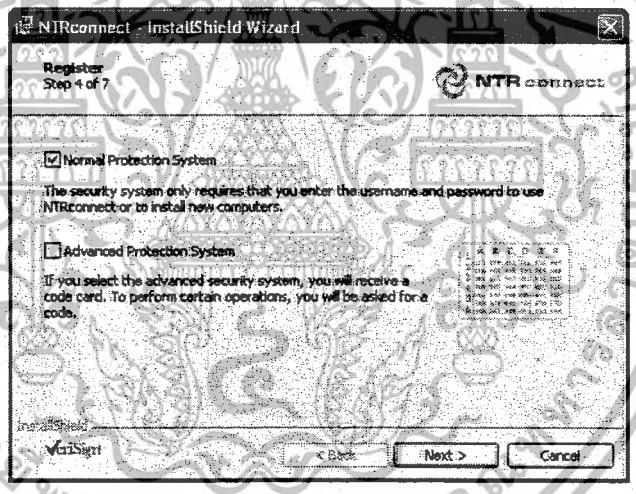
หากยังไม่เคยสมัครสมาชิกให้คลิกที่ New Account

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



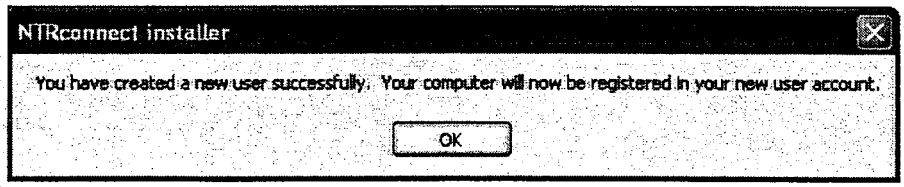
รูปที่ 2.24 การติดตั้งโปรแกรม

หน้านี้เป็นการสมัครสมาชิกใหม่ โดยการกรอกชื่อ, นามสกุล, ชื่อบริษัท, อีเมลล์, เบอร์โทรศัพท์ และเลือกประเทศ หลังจากนั้นกดปุ่ม Next



รูปที่ 2.25 การติดตั้งโปรแกรม

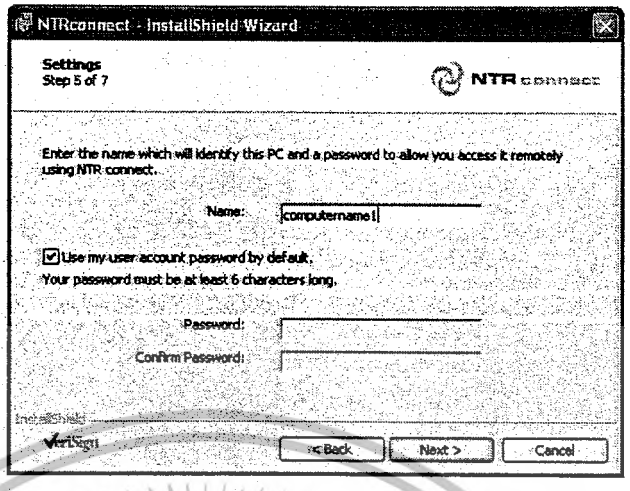
เลือกรูปแบบการเชื่อมต่อโดยแบบแรก จะใช้ user และ password ในการเข้าควบคุมระยะไกล ส่วนแบบที่สอง เราจะได้ภาพซึ่งไว้ใช้เป็นรหัสผ่านในการใช้งาน



รูปที่ 2.26 การติดตั้งโปรแกรม

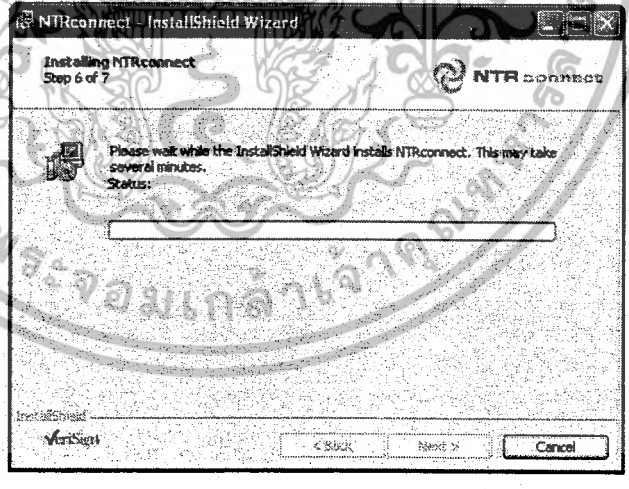
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมบอกว่าได้สร้างผู้ใช้งานสำเร็จแล้ว



รูปที่ 2.27 การติดตั้งโปรแกรม

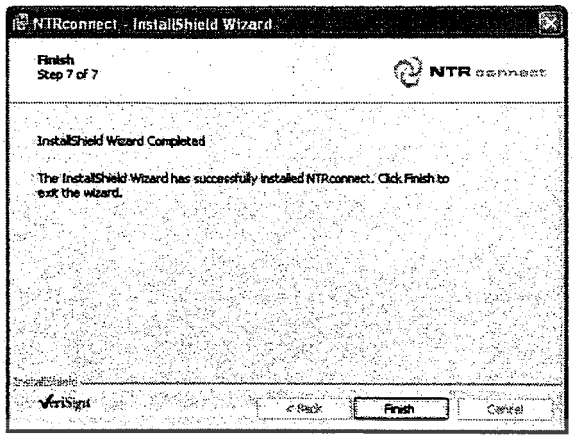
ในช่อง Name ให้ใส่ชื่อเรียกคอมพิวเตอร์ของคุณ ซึ่งชื่อจะปรากฏเมื่อคุณเข้าใช้งานโปรแกรม NTRConnect (มีประโยชน์หากติดตั้ง NTRConnect ไว้หลายเครื่อง) จากนั้นคลิกเครื่องหมายถูกหน้า Use my account password by default หากต้องการใช้ชื่อและรหัสที่สมัครไว้ในการใช้งาน หากต้องการใช้ชื่ออื่นในการใช้งานให้คลิกเครื่องหมายถูกออกแล้ว ใส่ชื่อและรหัสผ่านในช่องที่กำหนด จากนั้นกดปุ่ม Next



รูปที่ 2.28 การติดตั้งโปรแกรม

รอให้โปรแกรมติดตั้งจนเสร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 การติดตั้งโปรแกรม

จากนั้นกดปุ่ม Finish (จบการติดตั้ง)

2.9.2 วิธีใช้งาน NTRConnect

หลังจากเราได้ ติดตั้ง NTRConnect แล้ว จะได้ ไอคอน ของ NTRConnect ปรากฏขึ้นที่ Icon Tray ดังรูป

รูปที่ 2.30 การใช้โปรแกรม NTRConnect

ไอคอนนี้บอกว่าติดตั้งโปรแกรม NTRConnect ไว้ในเครื่องนี้ และสามารถควบคุมเข้ามาเครื่องนี้ได้จากระยะไกล

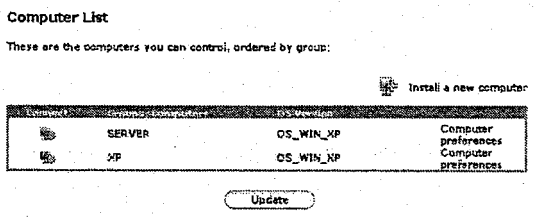
การควบคุม (remote control)

1. เข้าไปที่เว็บไซต์ www.ntrconnect.com

รูปที่ 2.31 การใช้โปรแกรม NTRConnect

ใส่ชื่อและรหัสผ่านที่สมัครไว้จากนั้นกดปุ่ม Log in

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.32 การใช้โปรแกรม NTRConnect

ให้คลิกที่รูปคอมพิวเตอร์ที่มีวงกลมสีเขียว (แสดงว่า online หากสีแดงคือ offline) เมื่อเชื่อมต่อเข้าไป ก็จะสามารถควบคุม (remote control) เครื่องนั้นๆได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ฮาร์ดแวร์

3.1 อุปกรณ์ที่ใช้

Microcontroller 32 bit CP-JR ARM7 LPC2368

CP-JR ARM7 LPC2368 เป็นบอร์ดไมโครคอนโทรเลอร์ในตระกูล ARM7TDMI-S Core ซึ่งเลือกใช้ไมโครคอนโทรเลอร์ 16/32-Bit ขนาด 100 Pin(LQFP) แบบใช้พลังงานต่ำเป็น MCUประจำบอร์ดซึ่งบอร์ดนี้เลือกใช้ MCU เบอร์ LPC2368 ของ Philips (NXP) โดยการออกแบบโครงสร้างของบอร์ดนั้นจะเน้นเรื่องของการจัดวางอุปกรณ์พื้นฐานที่จำเป็นต่อการนำไปประยุกต์ใช้งานและศึกษาทดลองขั้นพื้นฐานรวมไว้อย่างครบถ้วน เช่น LED แสดงสถานะของ Output Logic และ Push Button Switch สำหรับสร้างสัญญาณ Logic เพื่อทดสอบการทำงานของ Input หรือ Volume ปรับค่าแรงดัน เพื่อใช้ทดสอบการทำงานของ A/D รวมถึงวงจรขับเสียงโดยใช้ Mini-Speaker สำหรับสร้างเสียง Beep ต่างๆ เป็นต้น

นอกจากวงจรขั้นพื้นฐานดังกล่าวข้างต้นแล้ว บอร์ด CP-JR ARM7 LPC2368 ยังได้ออกแบบวงจรสำหรับประยุกต์ใช้งานขั้นสูงจัดเตรียมไว้ให้ใช้งานด้วย เช่น

- วงจรเชื่อมต่อกับ USB ซึ่งรองรับการเชื่อมต่อกับ USB 2.0 ได้ ซึ่งผู้ใช้สามารถนำบอร์ดไปพัฒนา
  เป็น USB Device แบบต่างๆ ได้โดยสะดวก
- วงจรเชื่อมต่อกับการ์ดหน่วยความจำ ซึ่งสามารถใช้งานได้กับการ์ดหน่วยความจำแบบ SD Card
  และการ์ดหน่วยความจำแบบ MMC Card
- วงจรเชื่อมต่อ Ethernet LAN แบบ 10/100Mb สามารถเชื่อมต่อกับเครือข่าย LAN แบบ Ethernet ได้ทั้งระบบ 10Mb และ 100Mb
- วงจรเชื่อมต่อกับ Dot-Matrix LCD แบบ Character พร้อมวงจรปรับความสว่าง
- วงจรสื่อสารข้อมูลแบบ RS232 พร้อม Line Driver จำนวน 2 ช่อง
- วงจรสื่อสารข้อมูลแบบ RS422/485 ทั้งแบบ Half-Duplex และ Full-Duplex

นอกเหนือจากนี้แล้วยังมี GPIO ต่างที่ว่างไว้ให้ผู้ใช้ออกแบบใช้งานร่วมกับอุปกรณ์อื่นๆ ได้เองตามความเหมาะสม สำหรับวิธีการพัฒนาโปรแกรมของบอร์ดนั้นก็มีความอ่อนตัวเป็นอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาก กล่าวคือสามารถใช้การพัฒนาแบบ ISP Download ผ่านทางพอร์ตสื่อสารอนุกรม RS232 และการพัฒนาแบบJTAG โดยใช้งานร่วมกับ ARM JTAG มาตรฐาน สำหรับ Download และ Debug ได้โดยง่าย ซึ่งจะเห็นได้ว่าโครงสร้างโดยรวมของบอร์ด CP-JR ARM7 LPC2368 จะมีความหลากหลาย และครบถ้วนพอสมควรเหมาะที่จะใช้เป็นบอร์ดทดลองเรียนรู้ และนำไปประยุกต์ดัดแปลงสร้างเป็น Application ใช้งานในด้านต่างๆได้มากมาย

3.2 คุณสมบัติของบอร์ด

1. ใช้ MCU ตระกูล ARM7TDMI-S เบอร์ LPC2368 ของ Philips(NXP) ซึ่งเป็น MCU ขนาด 16/32Bit

2. ภายใน MCU มีหน่วยความจำโปรแกรมแบบ Flash ขนาด 512KB, Static RAM ขนาด 58KB

3. ใช้ Crystal 12.00 MHz โดย MCU สามารถประมวลผลด้วยความเร็วสูงสุดที่ 72 MHz เมื่อใช้งาน

ร่วมกับ Phase-Locked Loop (PLL) ภายในตัว MCU เอง

4. มีวงจร RTC(Real Time Clock) พร้อม XTAL ค่า 32.768KHz และ Battery Backup

5. รองรับการโปรแกรมแบบ In-System Programming (ISP) และ In-Application Programming

(IAP) ผ่านทาง On-Chip Boot-Loader Software ทางพอร์ต UART-0 (RS232)

6. มีวงจรเชื่อมต่อกับ JTAG ARM ขนาด 20 Pin มาตรฐาน เพื่อทำการ Debug แบบ Real Time ได้

7. Power Supply ใช้แรงดันไฟฟ้า 7-12 VAC/DC โดยใช้ขั้วต่อแบบ Terminal และ DC-Jack พร้อม

วงจร Bridge Rectifier และ Regulate +5V/800mA และ +3V3/3A

8. มีวงจร USB มาตรฐาน 2.0 แบบ Full Speed ภายในตัว (USB Function มี 32 End Point)

9. มีวงจรเชื่อมต่อ Ethernet LAN 10/100Mb โดยใช้ขั้วต่อแบบ RJ45 มาตรฐาน จำนวน 1 ช่อง

10. มีวงจรเชื่อมต่อการ์ดหน่วยความจำแบบ SD Card หรือ MMC Card จำนวน 1 ช่อง

11. มีวงจรสื่อสาร RS232 โดยใช้ขั้วต่อแบบ 4-PIN มาตรฐาน ETT จำนวน 2 ช่อง

12. มีวงจรสื่อสารอนุกรม RS422/485 โดยใช้ขั้วต่อแบบ 6-PIN มาตรฐาน ETT จำนวน 1 ช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. มีวงจรเชื่อมต่อ Dot-Matrix LCD พร้อมวงจรปรับความสว่าง ใช้ขั้วต่อ 14 Pin มาตรฐาน ETT

14. มีวงจร Push Button Switch จำนวน 3 ชุด พร้อมสวิตช์ RESET

15. มีวงจร LED แสดงสถานะเพื่อทดลอง Output จำนวน 2 ชุด

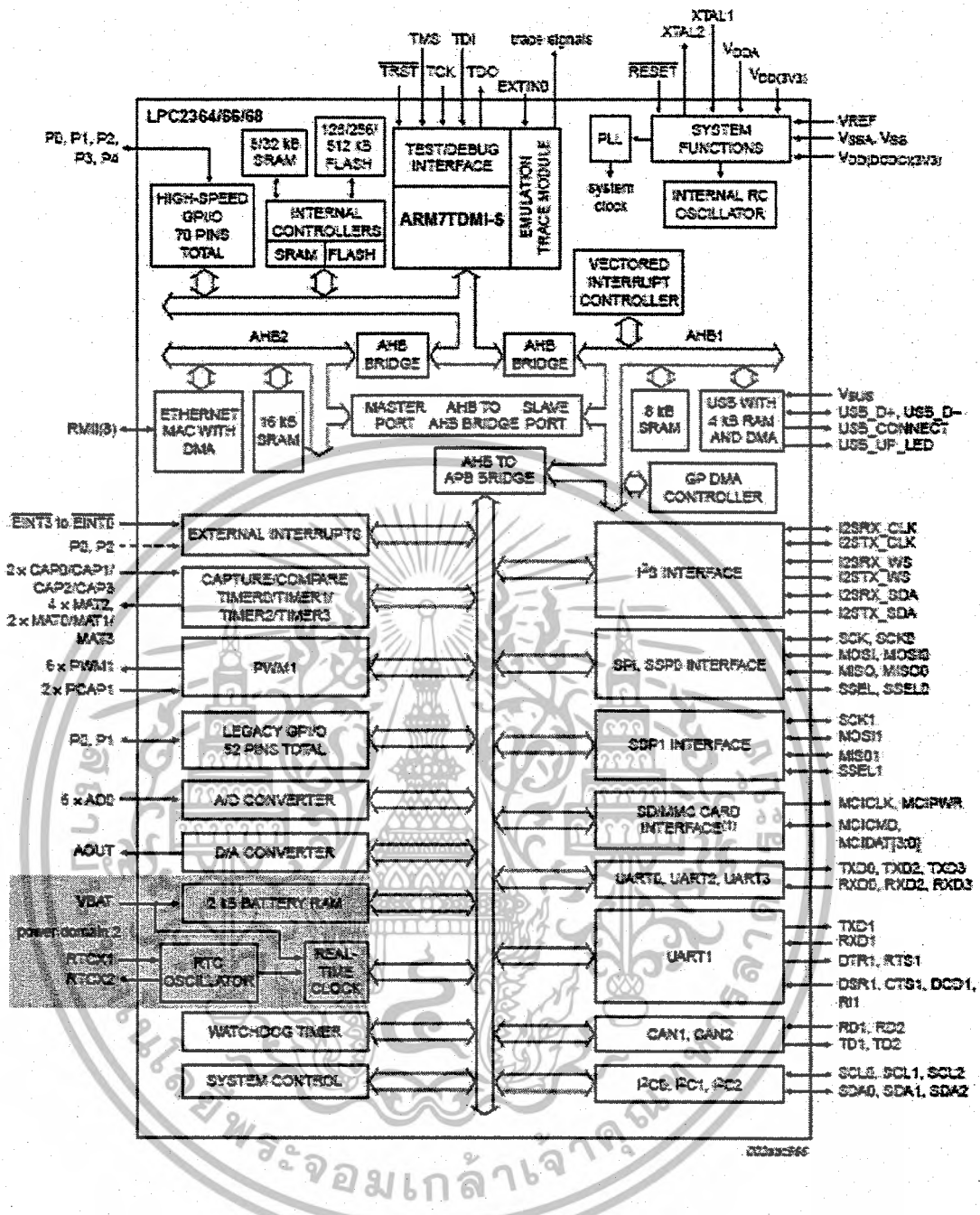
16. มีวงจร สร้างแรงดัน 0-3V3 โดยใช้ตัวต้านทานปรับค่าได้สำหรับทดสอบ A/D จำนวน 1 ชุด

17. มีวงจรกำเนิดและขับเสียง Beep โดยใช้ Mini Speaker จำนวน 1 ชุด

18. มี 25 Bit GPIO อิสระ สำหรับประยุกต์ต่างๆ เช่น A/D,D/A,I2C,SPI และ Input / Output

- Header 10Pin IDE (P2[0..7]) สำหรับ GPIO หรือ Full-Duplex Serial UART
- Header 10Pin IDE (P0[4..7],P1[20..23]) สำหรับ GPIO หรือ Matrix Key ขนาด 4x4
- 3 Pin Header(P0[26]) สำหรับ GPIO หรือ D/A
- 4 Pin Header(P0[24..25]) สำหรับ GPIO หรือ A/D
- 4 Pin Header(P0[27..28]) สำหรับ GPIO หรือ I2C Bus
- 6 Pin Header(P0[15..18] สำหรับ GPIO หรือ SPI Bus

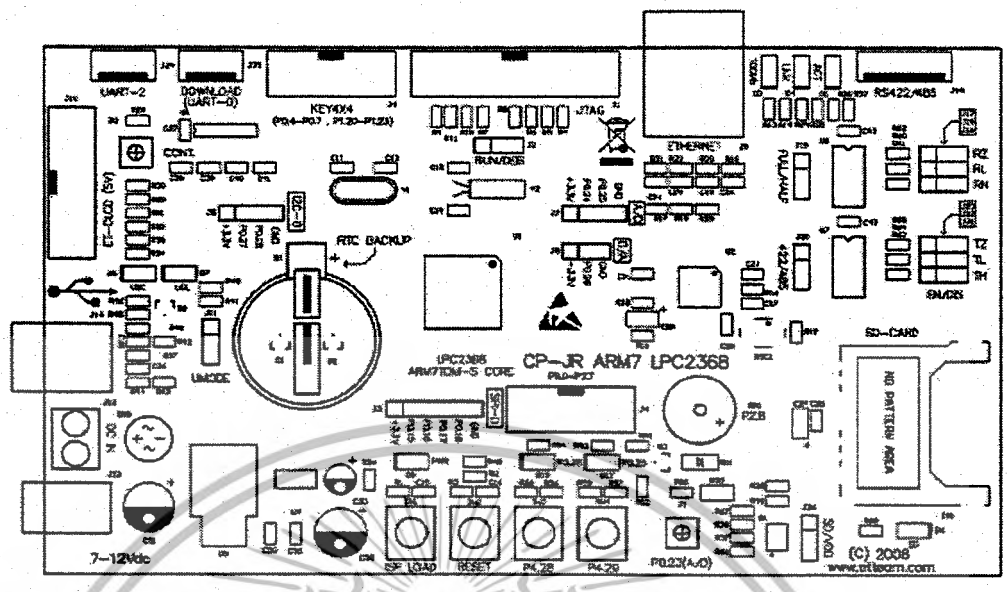
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



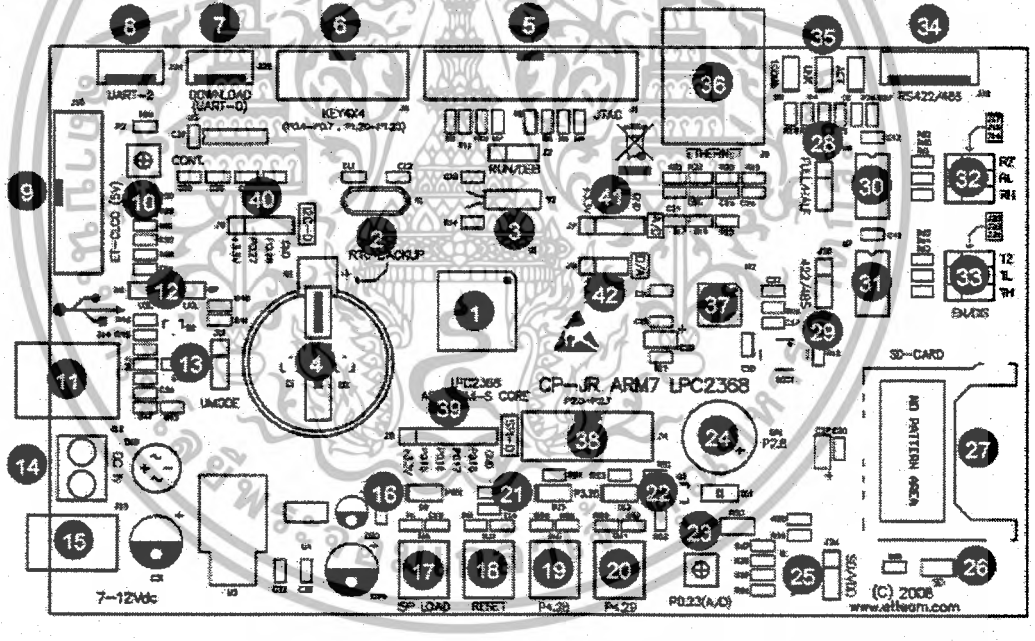
รูปที่ 3.1 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ ARM7 เบอร์ LPC2368

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 โครงสร้างของบอร์ด CP-JR ARM7 LPC2368



รูปที่ 3.2 โครงสร้างบอร์ด CP-JR ARM7 LPC2368



รูปที่ 3.3 แสดงตำแหน่งของอุปกรณ์ต่างๆในบอร์ด CP-JR ARM7 LPC2368

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

• **หมายเลข 1 คือ** MCU เบอร์ LPC2368 (100Pin LQFP)

• **หมายเลข 2 คือ** Crystal ค่า 12 MHz สำหรับใช้เป็นฐานเวลาระบบให้ MCU

• **หมายเลข 3 คือ** Crystal ค่า 32.768KHz สำหรับฐานเวลาให้ RTC ภายในตัว MCU

• **หมายเลข 4 คือ** Battery ขนาด 3V สำหรับ Backup ค่าให้กับระบบ RTC

• **หมายเลข 5 คือ** ขั้วต่อ JTAG ARM สำหรับ Debug แบบ Real Time

• **หมายเลข 6 คือ** ขั้วต่อ GPIO(P0[4..7],P1[20..23]) สำหรับ Keyboard Matrix 4x4 หรือ GPIO

• **หมายเลข 7 คือ** ขั้วต่อ UART-0(RS232) สำหรับใช้งาน และ Download Hex File ให้ CPU

• **หมายเลข 8 คือ** ขั้วต่อ UART-2(RS232) สำหรับใช้งาน

• **หมายเลข 9 คือ** ขั้วต่อ Character LCD โดยใช้กับ LCD แบบ +5V Supply

• **หมายเลข 10 คือ** VR สำหรับปรับค่าความสว่างให้ Character LCD

• **หมายเลข 11 คือ** ขั้วต่อ USB สำหรับเชื่อมต่อกับ USB Hub รุ่น 2.0

• **หมายเลข 12 คือ** LED แสดงค่าสถานะของการทำงานและการเชื่อมต่อของ USB

• **หมายเลข 13 คือ** Jumper สำหรับเลือกโหมดการทำงานของ USB

• **หมายเลข 14 และ 15 คือ** ขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ดใช้ได้กับไฟ 7-12V AC/DC

• **หมายเลข 16 คือ** LED แสดงสถานะของ Power +VDD(+3V3)

• **หมายเลข 17 คือ** SW1 เป็น ISP LOAD หรือ P2.10/EINT0

• **หมายเลข 18 คือ** SW2 หรือ สวิตช์ RESET

• **หมายเลข 19 และ 20 คือ** SW3 และ SW4 ใช้ทดสอบ Logic Input ของ P4.28 และ P4.29

• **หมายเลข 21 และ 22 คือ** LED ใช้ทดสอบ Logic Output ของ P3.25 และ P3.26

• **หมายเลข 23 คือ** VR สำหรับปรับค่าแรงดัน 0-3V3 สำหรับทดสอบ A/D(P0.23/AD0.0)

• **หมายเลข 24 คือ** Mini Speaker สำหรับใช้กำเนิดเสียงความถี่ต่างๆ

• **หมายเลข 25 คือ** Jumper สำหรับเลือกแหล่งจ่ายไฟให้กับการ์ดหน่วยความจำ SD/MMC

•**หมายเลข 26 คือ** LED แสดงสถานะของ แหล่งจ่ายไฟของการ์ดหน่วยความจำ SD/MMC

• **หมายเลข 27 คือ** ช่องเสียบการ์ดหน่วยความจำสามารถใช้ได้กับ SD Card และ MMC Card

• **หมายเลข 28 และ 29 คือ** Jumper สำหรับเลือกกำหนดการทำงานของ RS422/485

• **หมายเลข 30 คือ** IC Line Driver ของ RS422 Receive ใช้ได้กับ 75176 หรือ MAX3088

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

• **หมายเลข 31 คือ** IC Line Driver ของ RS422 Transmit และ RS485 Transceiver สามารถใช้ได้กับเบอร์ 75176 หรือ MAX3088

• **หมายเลข 32 และ 33** คือ Jumper สำหรับเลือก Enable/Disable Fail-Save Resistor และ Terminate Resistance ของ RS422 Receive,RS422 Transmit และ RS485 Transceiver

• **หมายเลข 34 คือ** ขั้วต่อสัญญาณ RS422/485

• **หมายเลข 35 คือ** LED แสดงสถานะของ Ethernet LAN

• **หมายเลข 36 คือ** ขั้วต่อสัญญาณ Ethernet LAN แบบ RJ45

• **หมายเลข 37 คือ** IC Physical Ethernet Driver เบอร์ DP83848

• **หมายเลข 38 คือ** ขั้วต่อ GPIO(P2[0..7] สำหรับ Full-Duplex UART1 หรือ GPIO

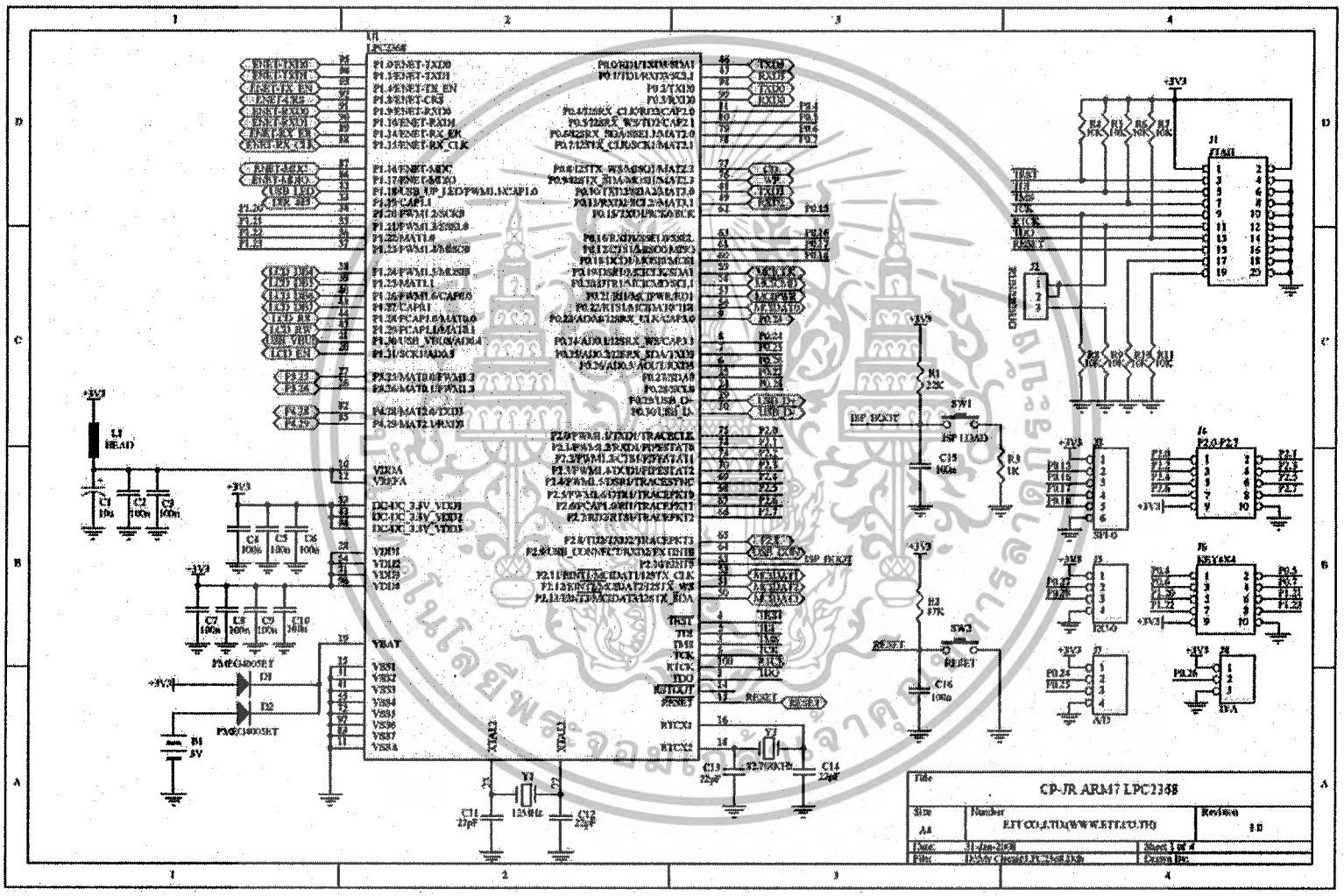
• **หมายเลข 39 คือ** ขั้วต่อ SPI0-Bus หรือ GPIO P0[15..18]

• **หมายเลข 40 คือ** ขั้วต่อ I2C0-Bus หรือ GPIO P0[27..28]

• **หมายเลข 41 คือ** ขั้วต่อ A/D หรือ GPIO P0[24..25]

• **หมายเลข 42 คือ** ขั้วต่อ D/A หรือ GPIO P0.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงวงจรไมโครคอนโทรลเลอร์

3.4 ไอซีตรวจสอบอุณหภูมิ (Temperature Sensors IC. :DS-1820)

การตรวจสอบอุณหภูมิของสภาพแวดล้อมเราสามารถใช้อุปกรณ์ตรวจจับอุณหภูมิได้หลายชนิด เช่น การใช้งานเทอร์โมมิเตอร์แบบปรอท เทอร์โมคัปเปอร์(เทอร์โมพลาย), เทอร์มิสเตอร์, ไอซีตรวจสอบอุณหภูมิกึ่งสำเร็จรูป (LM-335) , หรือไอซีตรวจสอบอุณหภูมิแบบสำเร็จรูป (DS-1820) สำหรับโครงงานนี้ผู้จัดทำเลือกใช้ไอซีตรวจสอบอุณหภูมิ (DS-1820) มาทำหน้าที่เป็นอุปกรณ์ตรวจสอบอุณหภูมิ ทั้งนี้เนื่องมาจากคุณสมบัติที่โดดเด่นหลายประการซึ่งจะได้กล่าวถึงในหัวข้อ "คุณสมบัติของไอซีตรวจสอบอุณหภูมิ DS-1820" ต่อไป

3.4.1 ระบบการสื่อสารข้อมูลอนุกรมแบบหนึ่งสาย (1-Wire™ Serial Bus)

DS-1820 เป็นไอซีที่ทำหน้าที่ในการตรวจสอบอุณหภูมิ ซึ่งผลิตขึ้นโดยบริษัทดัลลัสเซมิคอนดัคเตอร์ DS-1820 จัดเป็นอุปกรณ์หนึ่งที่อยู่ในระบบการสื่อสารข้อมูลอนุกรมแบบหนึ่งสาย (1-Wire™ Serial Bus) ซึ่งเป็นระบบที่ถูกคิดค้นและพัฒนาขึ้นโดยบริษัทดัลลัสเซมิคอนดัคเตอร์ เช่นกัน ในบางครั้งเราเรียกระบบการสื่อสารข้อมูลดัลลัสหนึ่งสายระบบการสื่อสารข้อมูลรูปแบบนี้เป็นระบบที่มีความชาญฉลาดใช้สายสัญญาณในการถ่ายทอดข้อมูลเพียง 1 เส้นเท่านั้น โดยที่ไม่ต้องมีสายสัญญาณนาฬิกามาทำการควบคุมจังหวะการทำงานดังเช่นกับระบบสื่อสารข้อมูลอนุกรมในรูปแบบอื่นๆ เนื่องจากสายข้อมูลเพียง 1 เส้นดังกล่าวสามารถทำหน้าที่เสมือนหนึ่งเป็นสายสัญญาณนาฬิกาไปในตัว ส่วนค่าของข้อมูลจะพิจารณาจากลักษณะของรูปสัญญาณที่ปรากฏบนสายสัญญาณในแต่ละช่องของเวลาหรือต่อไปนี้เราจะเรียกกันว่า "ไทม์สล็อต" (Time-Slot) โดยคาบเวลาสูงสุดและต่ำสุดของสถานะต่างๆ ในการสื่อสารข้อมูลในแต่ละไทม์สล็อตมีการกำหนดขอบเขตไว้อย่างชัดเจน การถ่ายทอดข้อมูลจะเกิดขึ้นได้ในแต่ละไทม์สล็อตนั้นรูปแบบการถ่ายทอดข้อมูลจะเป็นแบบอะซิงโครไนซ์ในระดับบิต ไม่มีการกำหนดความยาวของข้อมูลเป็นระดับไบต์

3.4.2 คุณสมบัติทางเทคนิคของระบบบัสหนึ่งสาย

สายสัญญาณบนระบบแบบบัสหนึ่งสายนี้จะเป็นสายสัญญาณแบบสองทิศทาง แต่ข้อมูลจะสามารถเดินทางได้ในทิศทางเดียวภายในช่วงเวลาหนึ่งๆ นั่นก็คือมีลักษณะคล้ายกับการสื่อสารแบบฮาล์ฟดูเพล็กซ์ (Half-Duplex) อุปกรณ์บนระบบบัสแบบหนึ่งสายนี้จะต้องมีการระบุอย่างชัดเจนว่าอุปกรณ์ตัวใดเป็นอุปกรณ์มาสเตอร์ และอุปกรณ์ตัวใดที่เป็นอุปกรณ์สเลฟ โดยส่วนใหญ่แล้วอุปกรณ์มาสเตอร์มักจะเป็นไมโครคอนโทรลเลอร์ ส่วนอุปกรณ์สเลฟได้แก่ไอซีตรวจสอบอุณหภูมิหน่วยความจำแรม เป็นต้น อุปกรณ์มาสเตอร์จะทำหน้าที่เป็นตัวจัดเตรียมความพร้อมของสายสัญญาณและควบคุมการถ่ายทอดข้อมูลลงบนสายสัญญาณนั้น ข้อมูลทั้งหมดไม่ว่าจะเป็นข้อมูลของคำสั่งควบคุมต่างๆ หรือข้อมูลใช้งานทั่วๆไปจะถูกส่งลงบนสายสัญญาณที่มีเพียงเส้นเดียวนี้ทั้งหมด ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างการทำงานของอุปกรณ์มาสเตอร์หรืออุปกรณ์สเลฟสามารถเป็นได้ทั้งตัวส่งและตัวรับทั้งนี้ขึ้นอยู่กับเงื่อนไขการทำงานในขณะนั้น ตัวอย่างเช่น หากมีการเขียนข้อมูลจากอุปกรณ์มาสเตอร์ไปยังอุปกรณ์สเลฟ ตัวที่ทำหน้าที่ในการส่งข้อมูลก็คืออุปกรณ์มาสเตอร์ และ ตัวรับข้อมูลก็คืออุปกรณ์สเลฟนั่นเอง แต่ถ้าหากเป็นการอ่านข้อมูลจากอุปกรณ์มาสเตอร์ตัวที่ทำหน้าที่ส่งข้อมูลก็คืออุปกรณ์สเลฟและตัวที่ทำหน้าที่เป็นตัวรับข้อมูลก็คืออุปกรณ์มาสเตอร์นั่นเอง ในระบบการสื่อสารข้อมูลแบบหนึ่งสายนั้นจะต้องมีอุปกรณ์ที่ทำหน้าที่เป็นอุปกรณ์มาสเตอร์เพียงตัวเดียวเท่านั้น

3.4.3 รูปแบบการสื่อสารข้อมูลแบบหนึ่งสาย (1-Wire™ Communication Protocal)

การติดต่อสื่อสารข้อมูลในระบบบัสหนึ่งอุปกรณ์มาสเตอร์จะสามารถทำการติดต่ออุปกรณ์สเลฟได้เพียงครั้งละ 1 ตัวเท่านั้น ดังนั้นอุปกรณ์สเลฟแต่ละตัวจะต้องมีข้อมูลกำหนดแอดเดรสเฉพาะตัวโดยจะเก็บไว้ในหน่วยความจำรอมภายในอุปกรณ์สเลฟนั้นๆ โดยปกติอุปกรณ์บนบัสหนึ่งสายของบริษัทดัลลัสจะมีหน่วยความจำขนาด 64 บิต หรือ 8 ไบต์ สำหรับเก็บข้อมูลต่างๆ ที่สำคัญของตัวอุปกรณ์แต่ละตัว ซึ่งประกอบด้วย

1. รหัสประจำตระกูล จำนวน 8 บิต เช่น DS-1820 มีรหัสประจำตระกูล 10H
2. เลขหมายประจำตัว (Serial Number) จำนวน 48 บิต
3. รหัสตรวจสอบความผิดพลาด (CRC : Cyclical Redundancy Check) 8 บิต

เราสามารถอ่านข้อมูลประจำตัวอุปกรณ์สเลฟได้ด้วยการใช้คำสั่งอ่านหน่วยความจำรอม (Read Rom) ในกรณีที่บนสายสัญญาณมีอุปกรณ์สเลฟเพียงตัวเดียวก็ไม่จำเป็นจะต้องอ้างแอดเดรสในการติดต่อ

รูปแบบการติดต่อบนระบบบัสหนึ่งสายจะเริ่มต้นขึ้น เมื่ออุปกรณ์มาสเตอร์ทำการรีเซตและกำหนดแอดเดรสของอุปกรณ์ที่ทำการติดต่อ ถ้าหากมีอุปกรณ์สเลฟเพียงตัวเดียวสามารถข้ามขั้นตอนการติดต่อหน่วยความจำรอมในอุปกรณ์สเลฟได้ โดยเราเรียกวิธีการดังกล่าวนี้ว่า “การไม่ติดต่อหน่วยความจำรอม หรือ สคริปรอม” จากนั้นก็จะรอการตอบรับจากอุปกรณ์สเลฟ เมื่อการตอบรับสมบูรณ์ก็จะสามารถเริ่มต้นขั้นตอนการอ่านหรือเขียนข้อมูลต่อไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.4 คุณสมบัติของไทม์สล็อต

อุปกรณ์มาสเตอร์จะเป็นเพียงอุปกรณ์ตัวเดียวบนระบบบัสหนึ่งสายที่สามารถทำการอินนิเชียลสายสัญญาณได้ โดยอุปกรณ์มาสเตอร์จะกำเนิดจุดเริ่มต้นของไทม์สล็อตด้วยการทำให้สายสัญญาณเป็นลอจิกต่ำในช่วงเวลาหนึ่ง จากนั้นจะทำให้กลับเป็นลอจิกสูง ถ้าหากอุปกรณ์สเลฟต้องการทำการส่งข้อมูลมายังอุปกรณ์มาสเตอร์ อุปกรณ์สเลฟก็จะเป็นตัวควบคุมสถานะของสายสัญญาณต่อไปจนเสร็จสิ้นกระบวนการ แต่ถ้าหากอุปกรณ์มาสเตอร์ต้องการทำการส่งข้อมูลก็จะสามารถดำเนินการต่อไปได้เลย

ฟังก์ชั่นของไทม์สล็อตที่กำหนดโดยอุปกรณ์มาสเตอร์มีด้วยกัน 4 ฟังก์ชั่นคือไทม์สล็อตของการรีเซต (Reset) การอ่านข้อมูล (Read Data) การเขียนข้อมูล “1” (Write One) และการเขียนข้อมูล “0” (Write Zero) ไทม์สล็อตของการรีเซตใช้ในการเริ่มต้นของการติดต่อกับอุปกรณ์สเลฟ ส่วนไทม์สล็อตของการเขียนข้อมูล “1” และ “0” ใช้สำหรับการเขียนข้อมูลไปยังอุปกรณ์สเลฟผ่านทางสายสัญญาณของระบบ

ทางด้านอุปกรณ์สเลฟมีฟังก์ชั่นของไทม์สล็อตอยู่ทั้งสิ้น 3 ฟังก์ชั่นคือ ไทม์สล็อตของการตอบสนอง (Presence) การเขียนข้อมูล “1” (Write One) และการเขียนข้อมูล “0” (Write Zero) ไทม์สล็อตของการตอบสนองการติดต่อจากอุปกรณ์มาสเตอร์ โดยอุปกรณ์สเลฟตัวที่ถูกเลือกจากอุปกรณ์มาสเตอร์จะต้องส่งสัญญาณการตอบสนองลงบนสายสัญญาณเพื่อแจ้งอุปกรณ์มาสเตอร์รับทราบว่าขณะนี้สามารถติดต่อกันได้แล้ว ส่วนไทม์สล็อตการเขียนข้อมูล “1” และ “0” ใช้สำหรับส่งข้อมูลไปยังอุปกรณ์มาสเตอร์ผ่านทางสัญญาณของระบบ ซึ่งจะสัมพันธ์กับไทม์สล็อตของการอ่านข้อมูลของอุปกรณ์มาสเตอร์

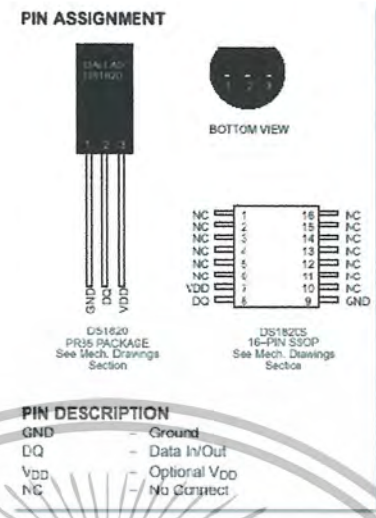
การแยกแยะแต่ละฟังก์ชั่นของไทม์สล็อตจะใช้ความยาวของคาบเวลาและลักษณะของรูปสัญญาณเป็นตัวกำหนด และทุกครั้งที่ทำการเปลี่ยนแปลงฟังก์ชั่นต้องทำให้สายสัญญาณอยู่ในสภาวะว่างเสมอ ซึ่งก็คือการทำให้สายสัญญาณเป็นลอจิกสูงเป็นเวลาอย่างน้อย 1 ไมโครวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5 คุณสมบัติของไอซีตรวจสอบอุณหภูมิ DS-1820

- สามารถวัดอุณหภูมิสภาพแวดล้อมได้ตั้งแต่ -55 ถึง +125 องศาเซลเซียส ความละเอียด 0.5 องศาเซลเซียส หรือวัดอุณหภูมิตั้งแต่ -67 ถึง +257 องศาฟาเรนไฮน์ ความละเอียด 0.9 องศาฟาเรนไฮน์
- สามารถอ่านข้อมูลของอุณหภูมิในรูปแบบของข้อมูลดิจิตอล ขนาด 9 บิต
- ใช้เวลาในการแปลงอุณหภูมิเป็นดิจิตอลเพียง 200 มิลลิวินาที
- ต้องการสายสัญญาณหรือช่องทางการถ่ายทอดข้อมูลเพียง 1 ช่องสัญญาณ (1บิต) เท่านั้น ทำให้ระบบโดยรวมมีขนาดเล็ก
- สามารถต่อ DS-1820 และอุปกรณ์ชนิดอื่นมากว่าหนึ่งตัวบนช่องสัญญาณเดียวกันได้ โดยการทำงานจะเป็นอิสระไม่รบกวนซึ่งกันและกัน ทั้งนี้เนื่องจาก DS-1820 แต่ละตัวจะมีรหัสประจำตัว (Unique silicon serial number) ที่ใช้ในการสื่อสารในระบบสื่อสารแบบหนึ่งสายฝังไว้ภายในตัวชิพแต่ละตัวนั่นเอง
- ไม่ต้องการอุปกรณ์ภายนอกเพิ่มเติม (นอกจากการต่อตัวต้านทานเพื่อพูลอัพที่ขาข้อมูล) จึงทำให้ความคลาดเคลื่อนของระบบเนื่องจากอุปกรณ์ภายนอกและการต่อวงจรลดลงอย่างมาก
- ใช้พลังงานในโหมดสแตนซ์บายต่ำมาก
- DS-1820 เป็นอุปกรณ์ที่ถูกบรรจุในรูปแบบของไอซี จึงมีผลความผิดพลาดอันเนื่องจากสิ่งรบกวนภายนอกน้อยมาก
- ต้นทุนโดยรวมของระบบต่ำกว่าเมื่อนำไปเปรียบเทียบกับระบบที่ใช้อุปกรณ์ตรวจสอบอุณหภูมิชนิดอื่นต่อร่วมกับอุปกรณ์ภายนอก
- มีขนาดที่เล็กกะทัดรัด ทำให้ระบบโดยรวมเล็กลง สะดวกต่อการเคลื่อนย้าย
- เมื่อระบบมีปัญหาสามารถตรวจสอบหาปัญหาที่เกิดขึ้นได้อย่างง่ายและรวดเร็ว

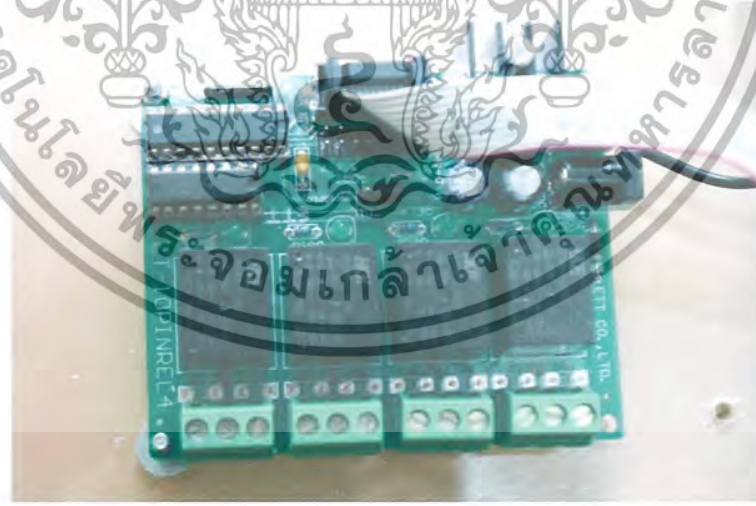
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงลักษณะไอซีตรวจสอบอุณหภูมิ DS-1820

3.5 การออกแบบภาคเอาต์พุต LED

ภาคเอาต์พุตเป็นภาคที่ใช้ควบคุมอุปกรณ์ LED ที่ใช้แทน ระบบอาหารและน้ำผ่านในฟาร์มไก่ โดย LED ใช้ไฟ 5 V จากวงจรกรองแรงดันไฟ โดยผ่านการสั่งงานจากภาคประมวลผลหลักมาที่รีเลย์ โดยหลักจะประกอบไปด้วย รีเลย์ ซึ่งวงจรนี้ทำหน้าที่เหมือนสวิตช์อิเล็กทรอนิกส์ใช้ควบคุมการทำงานเปิดปิดอุปกรณ์ไฟฟ้า



รูปที่ 3.6 วงจรรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ภาคตรวจสอบอุณหภูมิ

ภาคนี้จะเป็นวงจรควบคุมอุณหภูมิ โดยจะรับค่าแรงดันไฟฟ้าจากไอซี ds 1820 ที่ส่งมาเป็นดิจิตัล ผ่าน mcs-51 เบอร์ P89V51RD2 เพื่อประมวลผลว่า ตอนนี้อุณหภูมิเท่าไรแล้วจะสั่งให้พัดลมระบายความร้อน หรือหลอดไฟทำความร้อนทำงาน โดยผ่านวงจรรีเลย์สวิตช์ซึ่งจะเป็นสวิตช์ของการทำงานของพัดลมและหลอดไฟระบายความร้อน เมื่ออุณหภูมิสูงกว่าหรือต่ำกว่าที่เรากำหนดเพราะผ่านในฟาร์มไก่จะต้องรักษาอุณหภูมิให้คงที่

รูปที่ 3.7 วงจรประมวลผลอุณหภูมิ

รูปที่ 3.8 วงจรรีเลย์สวิตช์

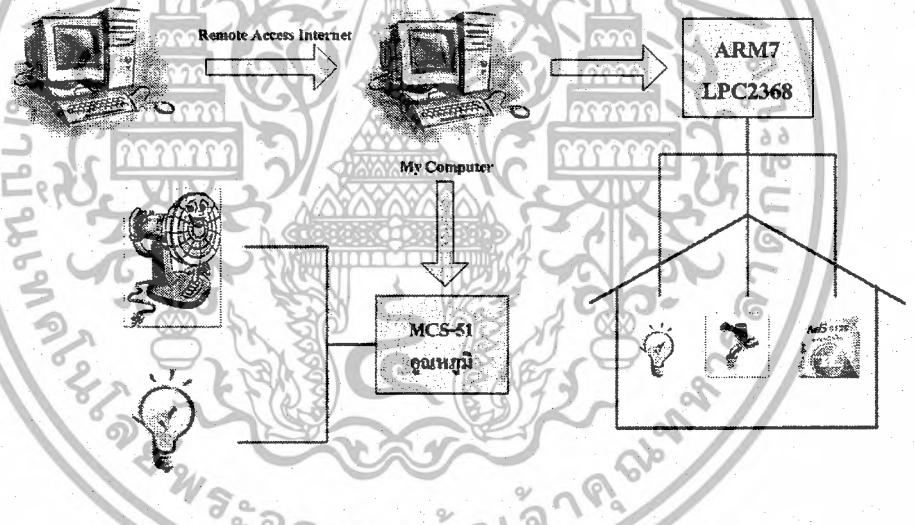
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบ

4.1 ภาพโดยรวมของระบบ

ระบบนี้เป็นการเอาความสามารถของไมโครคอนโทรลเลอร์ ARM7 เบอร์ LPC2368 มาประยุกต์ใช้กับระบบควบคุมภายในฟาร์มไก่ ทำให้สามารถควบคุมเรื่องอาหาร, ไฟฟ้า, น้ำ, และอุณหภูมิ โดยไม่ต้องใช่คนเข้าไปในฟาร์มไก่เพื่อหลีกเหลี่ยงการติดเชื้อโรคระหว่างคนกับไก่ และทำให้ไก่ไม่ติดโรค และเพื่อลดจำนวนคนในการดูแลฟาร์มไก่ให้มีจำนวนคนที่น้อยลง และยังสามารถควบคุมผ่านคอมพิวเตอร์ ทำให้สะดวกในการใช้งาน ภาพโดยรวมของระบบแสดงได้ดังในรูปข้างล่าง



รูปที่ 4.1 Diagram แสดงภาพโดยรวมของโครงงาน

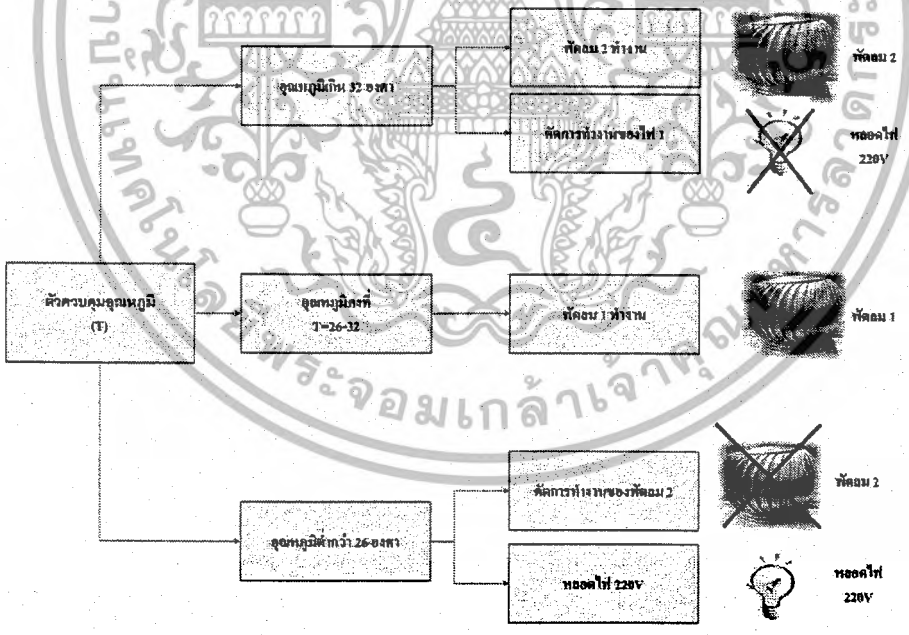
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การออกแบบ

ในปริญญานิพนธ์ฉบับนี้เป็นการออกแบบระบบควบคุมฟาร์มไก่ โดยใช้คอมพิวเตอร์ผ่านไมโครคอนโทรลเลอร์ ARM7 เบอร์ LPC2368 เป็นหลักในส่วนเพิ่มเติมคือการควบคุมผ่านอินเตอร์เน็ต ซึ่งเป็นการสั่งงานจากระยะไกลโดยใช้ remote access แล้วก็จะทำการส่งคำสั่งในรูปแบบข้อมูลในการควบคุมอาหาร, ไฟฟ้า, น้ำ มาให้แก่โปรแกรมควบคุมที่เขียนด้วย Visual Basic ประมวลผลไปสั่งงานส่วนฮาร์ดแวร์ซึ่งมีตัวไมโครคอนโทรลเลอร์ ARM7 เบอร์ LPC2368 โดยในส่วนโปรแกรม Visual Basic จะบอกถึงสถานะต่างๆของฟาร์มแบบ Real time และสามารถควบคุมผ่านคอมพิวเตอร์ได้เลย ร่วมถึงสามารถตั้งเวลาเปิด-ปิดอัตโนมัติได้โดยไม่ต้องนั่งเฝ้า

4.3 การออกแบบการทำงานของตัวควบคุมอุณหภูมิ

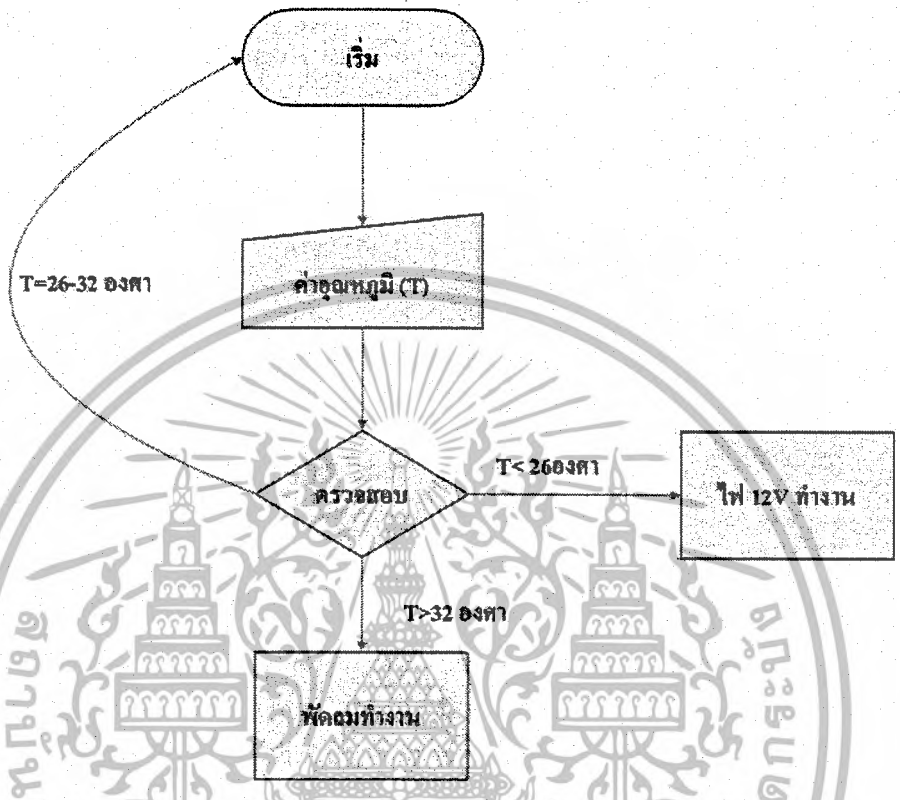
ในการออกแบบตัวควบคุมอุณหภูมิ เราจะวัดอุณหภูมิตอนนั้นแล้วตั้งค่าอุณหภูมิภายในโรงเลี้ยงไก่ให้พัดลมระบายอากาศทำงานเมื่ออุณหภูมิร้อนจนถึงอุณหภูมิที่กำหนดไว้ก็จะเปิดการทำงานของพัดลมตัวที่ 2 หรือเมื่ออุณหภูมิต่ำกว่าที่กำหนดให้ตัดการทำงานของพัดลมระบายอากาศตัวที่ 2 และเปิดการทำงานของหลอดไฟทำความร้อน ภาพของระบบตัวควบคุมอุณหภูมิได้ดังรูปข้างล่าง



รูปที่ 4.2 Diagram ของระบบควบคุมอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 Flow chart การทำงานของตัวควบคุมอุณหภูมิ



รูปที่ 4.3 Flow Chart แสดงการทำงานของตัวควบคุมอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการทดลอง

5.1 การทดลองตรวจจับอุณหภูมิ

ในการทดลองในตรวจจับอุณหภูมิภายในฟาร์มโดยใช้เทอร์โมมิเตอร์ โดยการทดลองนี้ตั้งอุณหภูมิที่ 26 องศาเซลเซียสถึง 32 องศาเซลเซียส ดังนั้นการทดลองนี้จะทำการทดลองอุณหภูมิตรวจจับที่ได้รับการออกแบบให้มีการแสดงผลเทียบค่ากับเทอร์โมมิเตอร์

รูปที่ 5.1 รูปแสดงแบบจำลองฟาร์มเลี้ยงไก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุณหภูมิ		
เทอร์โมมิเตอร์ (˚C)	ค่าที่วัดได้ (˚C)	% ความผิดพลาด
26.0	26.5	1.92
27.0	26.5	1.85
29.0	27.0	3.57
29.0	27.5	5.17
30.0	29.0	3.33
31.0	30.5	1.62
32.0	31.5	1.56

ตารางที่ 5.1 การตรวจวัดอุณหภูมิ

ในการทดลองตัวตรวจจับอุณหภูมิมีค่าความผิดพลาดน้อยที่ช่วงอุณหภูมิที่ทำการวัด เพราะเกิดจาก DS-1820 ที่บริษัทดัลลัสเซมิคอนดัคเตอร์ ได้คิดค้นและพัฒนาระบบ เพื่อใช้ในการวัดอุณหภูมิโดยตรง และมีความเที่ยงตรงในการตรวจจับอุณหภูมิ ทำให้เกิดความผิดพลาดในการทดลองน้อยมาก % ความผิดพลาดสามารถคำนวณได้จาก

$$\%\ \text{ความผิดพลาด} = \frac{(\text{ค่าจริง} - \text{ค่าที่วัดได้}) * 100}{\text{ค่าจริง}}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.2 แสดงการทำงานของไฟเมื่ออุณหภูมิต่ำกว่า 26 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การทดลองด้านโปรแกรม (Software)

รูปที่ 5.3 การทดลองทางด้านโปรแกรม

เป็นการเขียนโปรแกรมขึ้นมาเพื่อควบคุมแบบจำลองการเลี้ยงไก่ในฟาร์ม โดยโปรแกรมนี้จะทำตั้งเวลาการควบคุมระบบ น้ำ อาหาร ไฟ ภายในฟาร์ม ระบบก็จะดำเนินการได้โดยอัตโนมัติ ซึ่งมีปุ่มควบคุมการทำงานดังนี้

SW 1 ใช้ควบคุมการเปิด-ปิด สถานะของอาหาร แสดงผลที่หลอดไฟ LED สีเหลือง

SW 2 ใช้ควบคุมการเปิด-ปิด สถานะของน้ำ แสดงผลที่หลอดไฟ LED สีแดง

SW 3 ใช้ควบคุมการเปิด-ปิด สถานะของไฟในฟาร์ม แสดงผลที่หลอดไฟ 220V

SW 4 ใช้ควบคุมการเปิด-ปิด สถานะของไฟ 220V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปุ่มควบคุม	สถานะ		การตั้งเวลา	
	ON	OFF	ON	OFF
SW 1 คุมระบบอาหาร	✓	✓	✓	✓
SW 2 คุมระบบน้ำ	✓	✓	✓	✓
SW 3 คุมระบบไฟในฟาร์ม	✓	✓	✓	✓
SW 4 ไฟ 220V	✓	✓	✓	✓

ตารางที่ 5.2 การทดลองปุ่มต่าง ๆทางด้านโปรแกรม (Software)

5.3 การทดลองการกดสวิตซ์ที่ตัวเครื่อง

รูปที่ 5.4 การทดลองการกดสวิตซ์ที่ตัวเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สวิตซ์ควบคุม	ON	OFF
สวิตซ์ที่ 1 ควบคุมอาหาร	✓	✓
สวิตซ์ที่ 2 ควบคุมน้ำ	✓	✓
สวิตซ์ที่ 3 ควบคุมไฟในฟาร์ม	✓	✓
สวิตซ์ที่ 4 ควบคุมไฟ 220V	✓	✓

ตารางที่ 5.3 แสดงสวิตซ์ควบคุมที่ตัวเครื่องไปยังหลอดไฟ LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
บทสรุปและวิจารณ์

6.1 สรุปผลการดำเนินงาน

โครงงานนี้เป็นโครงงานที่เกี่ยวกับแบบจำลองโรงเลี้ยงไก่ผ่านคอมพิวเตอร์ ที่สามารถควบคุมและแสดงผลได้ทั้งในตัวเครื่องและที่หน้าจอคอมพิวเตอร์ โดยที่ตัวเครื่องจะแสดงสถานะของระบบน้ำและอาหารด้วยหลอด LED ในส่วนของอุณหภูมิ จะทำงานผ่านไอซี DS-1820 ซึ่งการทำงานของตัวไอซีนี้ จะควบคุมไปถึงพัดลมระบายอากาศและหลอดไฟทำความร้อน เมื่ออุณหภูมิในขณะนั้นมีค่าเปลี่ยนแปลงไปจากที่กำหนดไว้จะสั่งการทำงานของพัดลมระบายระบายอากาศและหลอดไฟทำความร้อน ตามอุณหภูมิที่ตั้งไว้

6.2 ปัญหาที่เกิดขึ้นและแนวทางแก้ไข

ปัญหา

1. หลอดไฟทำความร้อนกินกระแสมากเกิน เมื่อเปิดไฟทำความร้อนนานๆ ทำให้กระแสกรองไฟดร็อบลง
2. ระบบอินเตอร์เน็ตที่ต้องผ่านคอมพิวเตอร์ในการควบคุมฟาร์ม ได้เพียงเครื่องเดียวเท่านั้น และถ้าคอมพิวเตอร์ที่ควบคุมเกิดดับและเสีย ก็จะไม่สามารถควบคุมระบบผ่านอินเตอร์เน็ตได้

แนวทางแก้ไข

1. ออกแบบวงจรกรองไฟฟ้าใหม่หรือเปลี่ยนหม้อแปลงที่ให้กระแสสูงกว่า
2. สร้าง SERVER ขึ้นมาเพื่อรองรับระบบ

6.3 แนวทางพัฒนาในอนาคต

1. สามารถนำไปใช้งานจริงที่โรงเลี้ยงไก่ได้
2. สามารถควบคุมผ่านเว็บไซต์ โดยมี SERVER ได้
3. สนับสนุนให้มีการใช้กันอย่างแพร่หลาย โดยเป็นวิทยาทานแก่ผู้ทำฟาร์มไก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] โอภาส ศิริครรชิตถาวร, 2549, **"เรียนรู้และพัฒนาไมโครคอนโทรลเลอร์ ARM7 LPC2148 ด้วยภาษาซี"**, พิมพ์ครั้งที่ 1, กรุงเทพมหานคร, โรงพิมพ์วชิรินทร์สาส์น รัชดา

[2] นคร ภักดีชาติ, อรรถพล บุญยะโภคา, โอภาส ศิริครรชิตถาวรและชัยวัฒน์ ลิ้มพรจิตรวิไล , **"คู่มือทดลองไมโครคอนโทรลเลอร์ 32 บิต ตระกูล ARM7 เบื้องต้นฉบับ LPC2148"**, พิมพ์ครั้งที่ 1, กรุงเทพมหานคร, บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด

[3] อ.ขจร อนุดิตย์, 2550, **"การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ MCS-51 ด้วยภาษา C"**, บริษัท เอ-บุ๊ค ดิสทริบิวชั่น จำกัด

[4] สัจจะ จรัสรุ่งรวีวร, 2550, **"คู่มือเขียนโปรแกรม Visual Basic 2008"** , บริษัท เอ-บุ๊ค ดิสทริบิวชั่น จำกัด

[5] อ.วัลลภ คงเพิ่มพูน, 2545," **การเลี้ยงไก่เนื้อ"**,พิมพ์ครั้งที่ 4,สำนักพิมพ์ เกษตรสาส์น

[6] สุธรรม ดิสวัสดิ์ ,**"คู่มือการเลี้ยงไก่เนื้อเชิงธุรกิจ"**,สำนักพิมพ์ฐานเกษตรกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code โปรแกรม VB 2008

```
Option Strict On
Option Explicit On

Imports Time.DeviceManagement
Imports Time.FileIo
Imports Time.Hid
Imports Microsoft.Win32.SafeHandles
Imports System.Runtime.InteropServices

Friend Class FrmMain

    Inherits System.Windows.Forms.Form

#Region "Windows Form Designer generated code "
    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()
    End Sub
    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal Disposing As Boolean)
        If Disposing Then
            If Not components Is Nothing Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(Disposing)
    End Sub
    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer
    Public ToolTip1 As System.Windows.Forms.ToolTip
    Public timeinout(7, 1) As String
    Public WithEvents tmrContinuousDataCollect As
System.Windows.Forms.Timer
    Public WithEvents cboByte0 As System.Windows.Forms.ComboBox
    Friend WithEvents Timer1 As System.Windows.Forms.Timer
    Friend WithEvents showtime As System.Windows.Forms.Label
    Friend WithEvents timeBox As System.Windows.Forms.GroupBox
    Friend WithEvents LED1 As System.Windows.Forms.PictureBox
    Friend WithEvents s_h As System.Windows.Forms.ComboBox
    Friend WithEvents s_m As System.Windows.Forms.ComboBox
    Friend WithEvents time_l1 As System.Windows.Forms.TextBox
    Friend WithEvents s_l1 As System.Windows.Forms.CheckBox
    Friend WithEvents s_l2 As System.Windows.Forms.CheckBox
    Friend WithEvents time_l2 As System.Windows.Forms.TextBox
    Friend WithEvents LED2 As System.Windows.Forms.PictureBox
    Friend WithEvents s_l3 As System.Windows.Forms.CheckBox
    Friend WithEvents time_l3 As System.Windows.Forms.TextBox
    Friend WithEvents LED3 As System.Windows.Forms.PictureBox
    Friend WithEvents s_l4 As System.Windows.Forms.CheckBox
    Friend WithEvents time_l4 As System.Windows.Forms.TextBox
    Friend WithEvents LED4 As System.Windows.Forms.PictureBox
    Friend WithEvents e_m As System.Windows.Forms.ComboBox
    Friend WithEvents e_h As System.Windows.Forms.ComboBox
    Friend WithEvents s_am As System.Windows.Forms.ComboBox
    Friend WithEvents e_am As System.Windows.Forms.ComboBox
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents add_t As System.Windows.Forms.Button
    Friend WithEvents GroupBox1 As System.Windows.Forms.GroupBox
    Friend WithEvents picOutput As System.Windows.Forms.PictureBox
    Friend WithEvents s_web As System.Windows.Forms.Button
    Friend WithEvents name1 As System.Windows.Forms.TextBox
    Friend WithEvents name2 As System.Windows.Forms.TextBox
    Friend WithEvents name3 As System.Windows.Forms.TextBox
    Friend WithEvents name4 As System.Windows.Forms.TextBox
    Friend WithEvents Label2 As System.Windows.Forms.Label
    Friend WithEvents Label3 As System.Windows.Forms.Label
    Friend WithEvents Label4 As System.Windows.Forms.Label
    Friend WithEvents Label5 As System.Windows.Forms.Label
    Friend WithEvents Label6 As System.Windows.Forms.Label
    Friend WithEvents Label7 As System.Windows.Forms.Label
    Friend WithEvents Label8 As System.Windows.Forms.Label
    Friend WithEvents Label9 As System.Windows.Forms.Label
    Friend WithEvents Label10 As System.Windows.Forms.Label
    Friend WithEvents Label11 As System.Windows.Forms.Label
    Public WithEvents tmrDelay As System.Windows.Forms.Timer
    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()
        Me.components = New System.ComponentModel.Container
        Me.ToolTip1 = New System.Windows.Forms.ToolTip(Me.components)
        Me.tmrContinuousDataCollect = New
System.Windows.Forms.Timer(Me.components)
        Me.cboByte0 = New System.Windows.Forms.ComboBox
        Me.tmrDelay = New System.Windows.Forms.Timer(Me.components)
        Me.Timer1 = New System.Windows.Forms.Timer(Me.components)
        Me.showtime = New System.Windows.Forms.Label
        Me.timeBox = New System.Windows.Forms.GroupBox
        Me.LED1 = New System.Windows.Forms.PictureBox
        Me.s_h = New System.Windows.Forms.ComboBox
        Me.s_m = New System.Windows.Forms.ComboBox
        Me.time_l1 = New System.Windows.Forms.TextBox
        Me.s_l1 = New System.Windows.Forms.CheckBox
        Me.s_l2 = New System.Windows.Forms.CheckBox
        Me.time_l2 = New System.Windows.Forms.TextBox
        Me.LED2 = New System.Windows.Forms.PictureBox
        Me.s_l3 = New System.Windows.Forms.CheckBox
        Me.time_l3 = New System.Windows.Forms.TextBox
        Me.LED3 = New System.Windows.Forms.PictureBox
        Me.s_l4 = New System.Windows.Forms.CheckBox
        Me.time_l4 = New System.Windows.Forms.TextBox
        Me.LED4 = New System.Windows.Forms.PictureBox
        Me.e_m = New System.Windows.Forms.ComboBox
        Me.e_h = New System.Windows.Forms.ComboBox
        Me.s_am = New System.Windows.Forms.ComboBox
        Me.e_am = New System.Windows.Forms.ComboBox
        Me.Label1 = New System.Windows.Forms.Label
        Me.add_t = New System.Windows.Forms.Button
        Me.GroupBox1 = New System.Windows.Forms.GroupBox
        Me.picOutput = New System.Windows.Forms.PictureBox
        Me.s_web = New System.Windows.Forms.Button
        Me.name1 = New System.Windows.Forms.TextBox
        Me.name2 = New System.Windows.Forms.TextBox
        Me.name3 = New System.Windows.Forms.TextBox
        Me.name4 = New System.Windows.Forms.TextBox
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Me.Label2 = New System.Windows.Forms.Label
        Me.Label3 = New System.Windows.Forms.Label
        Me.Label4 = New System.Windows.Forms.Label
        Me.Label5 = New System.Windows.Forms.Label
        Me.Label6 = New System.Windows.Forms.Label
        Me.Label7 = New System.Windows.Forms.Label
        Me.Label8 = New System.Windows.Forms.Label
        Me.Label9 = New System.Windows.Forms.Label
        Me.Label10 = New System.Windows.Forms.Label
        Me.Label11 = New System.Windows.Forms.Label
        Me.timeBox.SuspendLayout()
        CType(Me.LED1,
System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.LED2,
System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.LED3,
System.ComponentModel.ISupportInitialize).BeginInit()
        CType(Me.LED4,
System.ComponentModel.ISupportInitialize).BeginInit()
        Me.GroupBox1.SuspendLayout()
        CType(Me.picOutput,
System.ComponentModel.ISupportInitialize).BeginInit()
        Me.SuspendLayout()
        '
        'tmrContinuousDataCollect
        '
        Me.tmrContinuousDataCollect.Interval = 1
        '
        'cboByte0
        '
        Me.cboByte0.BackColor = System.Drawing.SystemColors.Window
        Me.cboByte0.Cursor = System.Windows.Forms.Cursors.Default
        Me.cboByte0.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList
        Me.cboByte0.Font = New System.Drawing.Font("Arial", 8.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
        Me.cboByte0.ForeColor = System.Drawing.SystemColors.WindowText
        Me.cboByte0.Location = New System.Drawing.Point(297, 33)
        Me.cboByte0.Name = "cboByte0"
        Me.cboByte0.RightToLeft = System.Windows.Forms.RightToLeft.No
        Me.cboByte0.Size = New System.Drawing.Size(101, 22)
        Me.cboByte0.TabIndex = 2
        Me.cboByte0.Visible = False
        '
        'tmrDelay
        '
        Me.tmrDelay.Interval = 1
        '
        'Timer1
        '
        Me.Timer1.Enabled = True
        Me.Timer1.Interval = 1000
        '
        'showtime
        '
        Me.showtime.AutoSize = True
        Me.showtime.Font = New System.Drawing.Font("Arial", 9.75!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
        Me.showtime.Location = New System.Drawing.Point(14, 16)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Me.showtime.Name = "showtime"
        Me.showtime.Size = New System.Drawing.Size(72, 16)
        Me.showtime.TabIndex = 8
        Me.showtime.Text = "                "
        '
        'timeBox
        '
        Me.timeBox.Controls.Add(Me.showtime)
        Me.timeBox.Location = New System.Drawing.Point(188, 15)
        Me.timeBox.Name = "timeBox"
        Me.timeBox.Size = New System.Drawing.Size(101, 42)
        Me.timeBox.TabIndex = 9
        Me.timeBox.TabStop = False
        Me.timeBox.Text = "Time"
        '
        'LED1
        '
        Me.LED1.BackColor = System.Drawing.Color.Black
        Me.LED1.Location = New System.Drawing.Point(98, 169)
        Me.LED1.Name = "LED1"
        Me.LED1.Size = New System.Drawing.Size(23, 25)
        Me.LED1.TabIndex = 10
        Me.LED1.TabStop = False
        '
        's_h
        '
        Me.s_h.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList
        Me.s_h.FormattingEnabled = True
        Me.s_h.Items.AddRange(New Object() {"1", "2", "3", "4", "5", "6",
"7", "8", "9", "10", "11", "12"})
        Me.s_h.Location = New System.Drawing.Point(16, 22)
        Me.s_h.MaxDropDownItems = 12
        Me.s_h.Name = "s_h"
        Me.s_h.Size = New System.Drawing.Size(40, 22)
        Me.s_h.TabIndex = 18
        '
        's_m
        '
        Me.s_m.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList
        Me.s_m.FormattingEnabled = True
        Me.s_m.Items.AddRange(New Object() {"00", "01", "02", "03", "04",
"05", "06", "07", "08", "09", "10", "11", "12", "13", "14", "15", "16",
"17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28",
"29", "30", "31", "32", "33", "34", "35", "36", "37", "38", "39", "40",
"41", "42", "43", "44", "45", "46", "47", "48", "49", "50", "51", "52",
"53", "54", "55", "56", "57", "58", "59"})
        Me.s_m.Location = New System.Drawing.Point(62, 22)
        Me.s_m.MaxDropDownItems = 15
        Me.s_m.Name = "s_m"
        Me.s_m.Size = New System.Drawing.Size(40, 22)
        Me.s_m.TabIndex = 26
        '
        'time_l1
        '
        Me.time_l1.Location = New System.Drawing.Point(169, 174)
        Me.time_l1.Name = "time_l1"
        Me.time_l1.ReadOnly = True
        Me.time_l1.Size = New System.Drawing.Size(120, 20)
        Me.time_l1.TabIndex = 27
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Me.time_l1.Text = "5:00 PM - 7:00 AM"
        Me.time_l1.TextAlign =
System.Windows.Forms.HorizontalAlignment.Center
        '
        's_l1
        '
        Me.s_l1.AutoSize = True
        Me.s_l1.Location = New System.Drawing.Point(47, 178)
        Me.s_l1.Name = "s_l1"
        Me.s_l1.Size = New System.Drawing.Size(15, 14)
        Me.s_l1.TabIndex = 35
        Me.s_l1.UseVisualStyleBackColor = True
        '
        's_l2
        '
        Me.s_l2.AutoSize = True
        Me.s_l2.Location = New System.Drawing.Point(47, 225)
        Me.s_l2.Name = "s_l2"
        Me.s_l2.Size = New System.Drawing.Size(15, 14)
        Me.s_l2.TabIndex = 38
        Me.s_l2.UseVisualStyleBackColor = True
        '
        'time_l2
        '
        Me.time_l2.Location = New System.Drawing.Point(169, 222)
        Me.time_l2.Name = "time_l2"
        Me.time_l2.ReadOnly = True
        Me.time_l2.Size = New System.Drawing.Size(120, 20)
        Me.time_l2.TabIndex = 37
        Me.time_l2.Text = "5:00 PM - 7:00 AM"
        Me.time_l2.TextAlign =
System.Windows.Forms.HorizontalAlignment.Center
        '
        'LED2
        '
        Me.LED2.BackColor = System.Drawing.Color.Black
        Me.LED2.Location = New System.Drawing.Point(98, 219)
        Me.LED2.Name = "LED2"
        Me.LED2.Size = New System.Drawing.Size(23, 26)
        Me.LED2.TabIndex = 36
        Me.LED2.TabStop = False
        '
        's_l3
        '
        Me.s_l3.AutoSize = True
        Me.s_l3.Location = New System.Drawing.Point(47, 277)
        Me.s_l3.Name = "s_l3"
        Me.s_l3.Size = New System.Drawing.Size(15, 14)
        Me.s_l3.TabIndex = 41
        Me.s_l3.UseVisualStyleBackColor = True
        '
        'time_l3
        '
        Me.time_l3.Location = New System.Drawing.Point(169, 274)
        Me.time_l3.Name = "time_l3"
        Me.time_l3.ReadOnly = True
        Me.time_l3.Size = New System.Drawing.Size(120, 20)
        Me.time_l3.TabIndex = 40
        Me.time_l3.Text = "5:00 PM - 7:00 AM"
        Me.time_l3.TextAlign =
System.Windows.Forms.HorizontalAlignment.Center
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        '
        'LED3
        '
        Me.LED3.BackColor = System.Drawing.Color.Black
        Me.LED3.Location = New System.Drawing.Point(98, 266)
        Me.LED3.Name = "LED3"
        Me.LED3.Size = New System.Drawing.Size(23, 28)
        Me.LED3.TabIndex = 39
        Me.LED3.TabStop = False
        '
        's_14
        '
        Me.s_14.AutoSize = True
        Me.s_14.Location = New System.Drawing.Point(47, 324)
        Me.s_14.Name = "s_14"
        Me.s_14.Size = New System.Drawing.Size(15, 14)
        Me.s_14.TabIndex = 44
        Me.s_14.UseVisualStyleBackColor = True
        '
        'time_14
        '
        Me.time_14.Location = New System.Drawing.Point(170, 324)
        Me.time_14.Name = "time_14"
        Me.time_14.ReadOnly = True
        Me.time_14.Size = New System.Drawing.Size(120, 20)
        Me.time_14.TabIndex = 43
        Me.time_14.Text = "5:00 PM - 7:00 AM"
        Me.time_14.TextAlign =
System.Windows.Forms.HorizontalAlignment.Center
        '
        'LED4
        '
        Me.LED4.BackColor = System.Drawing.Color.Black
        Me.LED4.Location = New System.Drawing.Point(95, 315)
        Me.LED4.Name = "LED4"
        Me.LED4.Size = New System.Drawing.Size(23, 29)
        Me.LED4.TabIndex = 42
        Me.LED4.TabStop = False
        '
        'e_m
        '
        Me.e_m.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList
        Me.e_m.FormattingEnabled = True
        Me.e_m.Items.AddRange(New Object() {"00", "01", "02", "03", "04",
"05", "06", "07", "08", "09", "10", "11", "12", "13", "14", "15", "16",
"17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28",
"29", "30", "31", "32", "33", "34", "35", "36", "37", "38", "39", "40",
"41", "42", "43", "44", "45", "46", "47", "48", "49", "50", "51", "52",
"53", "54", "55", "56", "57", "58", "59"})
        Me.e_m.Location = New System.Drawing.Point(231, 22)
        Me.e_m.MaxDropDownItems = 15
        Me.e_m.Name = "e_m"
        Me.e_m.Size = New System.Drawing.Size(40, 22)
        Me.e_m.TabIndex = 58
        '
        'e_h
        '
        Me.e_h.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList
        Me.e_h.FormattingEnabled = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Me.e_h.Items.AddRange(New Object() {"1", "2", "3", "4", "5", "6",
"7", "8", "9", "10", "11", "12"})
        Me.e_h.Location = New System.Drawing.Point(185, 22)
        Me.e_h.MaxDropDownItems = 12
        Me.e_h.Name = "e_h"
        Me.e_h.Size = New System.Drawing.Size(40, 22)
        Me.e_h.TabIndex = 57
        '
        's_am
        '
        Me.s_am.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList
        Me.s_am.FormattingEnabled = True
        Me.s_am.Items.AddRange(New Object() {"AM", "PM"})
        Me.s_am.Location = New System.Drawing.Point(108, 22)
        Me.s_am.Name = "s_am"
        Me.s_am.Size = New System.Drawing.Size(40, 22)
        Me.s_am.TabIndex = 59
        '
        'e_am
        '
        Me.e_am.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList
        Me.e_am.FormattingEnabled = True
        Me.e_am.Items.AddRange(New Object() {"AM", "PM"})
        Me.e_am.Location = New System.Drawing.Point(277, 22)
        Me.e_am.Name = "e_am"
        Me.e_am.Size = New System.Drawing.Size(40, 22)
        Me.e_am.TabIndex = 60
        '
        'Label1
        '
        Me.Label1.AutoSize = True
        Me.Label1.Location = New System.Drawing.Point(158, 25)
        Me.Label1.Name = "Label1"
        Me.Label1.Size = New System.Drawing.Size(19, 14)
        Me.Label1.TabIndex = 61
        Me.Label1.Text = "To"
        '
        'add_t
        '
        Me.add_t.Location = New System.Drawing.Point(329, 21)
        Me.add_t.Name = "add_t"
        Me.add_t.Size = New System.Drawing.Size(75, 24)
        Me.add_t.TabIndex = 62
        Me.add_t.Text = "Edit"
        Me.add_t.UseVisualStyleBackColor = True
        '
        'GroupBox1
        '
        Me.GroupBox1.Controls.Add(Me.e_h)
        Me.GroupBox1.Controls.Add(Me.s_h)
        Me.GroupBox1.Controls.Add(Me.add_t)
        Me.GroupBox1.Controls.Add(Me.s_m)
        Me.GroupBox1.Controls.Add(Me.Label1)
        Me.GroupBox1.Controls.Add(Me.e_m)
        Me.GroupBox1.Controls.Add(Me.e_am)
        Me.GroupBox1.Controls.Add(Me.s_am)
        Me.GroupBox1.Location = New System.Drawing.Point(19, 69)
        Me.GroupBox1.Name = "GroupBox1"
        Me.GroupBox1.Size = New System.Drawing.Size(422, 61)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Me.GroupBox1.TabIndex = 63
Me.GroupBox1.TabStop = False
Me.GroupBox1.Text = "Time On/Off"
'
'picOutput
'
Me.picOutput.BackColor = System.Drawing.Color.Black
Me.picOutput.Location = New System.Drawing.Point(481, 30)
Me.picOutput.Name = "picOutput"
Me.picOutput.Size = New System.Drawing.Size(320, 240)
Me.picOutput.TabIndex = 64
Me.picOutput.TabStop = False
'
's_web
'
Me.s_web.Location = New System.Drawing.Point(604, 315)
Me.s_web.Name = "s_web"
Me.s_web.Size = New System.Drawing.Size(75, 23)
Me.s_web.TabIndex = 66
Me.s_web.Text = "Start"
Me.s_web.UseVisualStyleBackColor = True
'
'name1
'
Me.name1.Location = New System.Drawing.Point(321, 174)
Me.name1.Name = "name1"
Me.name1.Size = New System.Drawing.Size(120, 20)
Me.name1.TabIndex = 67
Me.name1.Text = "อาหาร"
'
'name2
'
Me.name2.Location = New System.Drawing.Point(321, 225)
Me.name2.Name = "name2"
Me.name2.Size = New System.Drawing.Size(120, 20)
Me.name2.TabIndex = 68
Me.name2.Text = "น้ำ"
'
'name3
'
Me.name3.Location = New System.Drawing.Point(321, 277)
Me.name3.Name = "name3"
Me.name3.Size = New System.Drawing.Size(120, 20)
Me.name3.TabIndex = 69
Me.name3.Text = "ไฟใน Fram"
'
'name4
'
Me.name4.Location = New System.Drawing.Point(321, 324)
Me.name4.Name = "name4"
Me.name4.Size = New System.Drawing.Size(120, 20)
Me.name4.TabIndex = 70
Me.name4.Text = "ไฟ 220 V"
'
'Label2
'
Me.Label2.AutoSize = True
Me.Label2.Location = New System.Drawing.Point(78, 142)
Me.Label2.Name = "Label2"
Me.Label2.Size = New System.Drawing.Size(80, 14)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Me.Label2.TabIndex = 71
Me.Label2.Text = "สถานะ เปิด/ปิด "
'
'Label3
'
Me.Label3.AutoSize = True
Me.Label3.Location = New System.Drawing.Point(35, 142)
Me.Label3.Name = "Label3"
Me.Label3.Size = New System.Drawing.Size(31, 14)
Me.Label3.TabIndex = 72
Me.Label3.Text = "เลือก"
'
'Label4
'
Me.Label4.AutoSize = True
Me.Label4.Location = New System.Drawing.Point(177, 142)
Me.Label4.Name = "Label4"
Me.Label4.Size = New System.Drawing.Size(95, 14)
Me.Label4.TabIndex = 73
Me.Label4.Text = "ช่วงเวลาการทำงาน"
'
'Label5
'
Me.Label5.AutoSize = True
Me.Label5.Location = New System.Drawing.Point(345, 142)
Me.Label5.Name = "Label5"
Me.Label5.Size = New System.Drawing.Size(42, 14)
Me.Label5.TabIndex = 74
Me.Label5.Text = "อุปกรณ์"
'
'Label6
'
Me.Label6.AutoSize = True
Me.Label6.Location = New System.Drawing.Point(15, 52)
Me.Label6.Name = "Label6"
Me.Label6.Size = New System.Drawing.Size(89, 14)
Me.Label6.TabIndex = 75
Me.Label6.Text = "ตั้งเวลาการทำงาน"
'
'Label7
'
Me.Label7.AutoSize = True
Me.Label7.Location = New System.Drawing.Point(588, 289)
Me.Label7.Name = "Label7"
Me.Label7.Size = New System.Drawing.Size(102, 14)
Me.Label7.TabIndex = 76
Me.Label7.Text = "ดูฟาร์มผ่านwebcam"
'
'Label8
'
Me.Label8.AutoSize = True
Me.Label8.Location = New System.Drawing.Point(8, 178)
Me.Label8.Name = "Label8"
Me.Label8.Size = New System.Drawing.Size(33, 14)
Me.Label8.TabIndex = 77
Me.Label8.Text = "SW 1"
'
'Label9
'
Me.Label9.AutoSize = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Me.Label9.Location = New System.Drawing.Point(11, 225)
        Me.Label9.Name = "Label9"
        Me.Label9.Size = New System.Drawing.Size(33, 14)
        Me.Label9.TabIndex = 78
        Me.Label9.Text = "SW 2"
        '
        'Label10
        '
        Me.Label10.AutoSize = True
        Me.Label10.Location = New System.Drawing.Point(11, 277)
        Me.Label10.Name = "Label10"
        Me.Label10.Size = New System.Drawing.Size(33, 14)
        Me.Label10.TabIndex = 79
        Me.Label10.Text = "SW 3"
        '
        'Label11
        '
        Me.Label11.AutoSize = True
        Me.Label11.Location = New System.Drawing.Point(11, 324)
        Me.Label11.Name = "Label11"
        Me.Label11.Size = New System.Drawing.Size(33, 14)
        Me.Label11.TabIndex = 80
        Me.Label11.Text = "SW 4"
        '
        'FrmMain
        '
        Me.ClientSize = New System.Drawing.Size(831, 373)
        Me.Controls.Add(Me.Label11)
        Me.Controls.Add(Me.Label10)
        Me.Controls.Add(Me.Label9)
        Me.Controls.Add(Me.Label8)
        Me.Controls.Add(Me.Label7)
        Me.Controls.Add(Me.Label6)
        Me.Controls.Add(Me.Label5)
        Me.Controls.Add(Me.Label4)
        Me.Controls.Add(Me.Label3)
        Me.Controls.Add(Me.Label2)
        Me.Controls.Add(Me.name4)
        Me.Controls.Add(Me.name3)
        Me.Controls.Add(Me.name2)
        Me.Controls.Add(Me.name1)
        Me.Controls.Add(Me.s_web)
        Me.Controls.Add(Me.picOutput)
        Me.Controls.Add(Me.GroupBox1)
        Me.Controls.Add(Me.cboByte0)
        Me.Controls.Add(Me.s_l4)
        Me.Controls.Add(Me.time_l4)
        Me.Controls.Add(Me.LED4)
        Me.Controls.Add(Me.s_l3)
        Me.Controls.Add(Me.time_l3)
        Me.Controls.Add(Me.LED3)
        Me.Controls.Add(Me.s_l2)
        Me.Controls.Add(Me.time_l2)
        Me.Controls.Add(Me.LED2)
        Me.Controls.Add(Me.s_l1)
        Me.Controls.Add(Me.time_l1)
        Me.Controls.Add(Me.LED1)
        Me.Controls.Add(Me.timeBox)
        Me.Font = New System.Drawing.Font("Arial", 8.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Me.Location = New System.Drawing.Point(21, 28)
        Me.Name = "FrmMain"
        Me.StartPosition = System.Windows.Forms.FormStartPosition.Manual
        Me.Text = "Time"
        Me.timeBox.ResumeLayout(False)
        Me.timeBox.PerformLayout()
        CType(Me.LED1, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.LED2, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.LED3, System.ComponentModel.ISupportInitialize).EndInit()
        CType(Me.LED4, System.ComponentModel.ISupportInitialize).EndInit()
        Me.GroupBox1.ResumeLayout(False)
        Me.GroupBox1.PerformLayout()
        CType(Me.picOutput,
System.ComponentModel.ISupportInitialize).EndInit()
        Me.ResumeLayout(False)
        Me.PerformLayout()

    End Sub
#End Region

    Private deviceNotificationHandle As IntPtr
    Private exclusiveAccess As Boolean
    Private hidHandle As SafeFileHandle
    Private hidUsage As String
    Private myDeviceDetected As Boolean
    Private myDevicePathName As String
    Private readHandle As SafeFileHandle
    Private writeHandle As SafeFileHandle
    Private myCam As iCam
    Private MyDebugging As New Debugging() ' For viewing results of API
calls via Debug.Write.
    Private MyDeviceManagement As New DeviceManagement()
    Private MyHid As New Hid()

    Friend FrmMy As FrmMain

    ''' <summary>
    ''' Define a class of delegates that point to the Hid.ReportIn.Read
function.
    ''' The delegate has the same parameters as Hid.ReportIn.Read.
    ''' Used for asynchronous reads from the device.
    ''' </summary>

    Private Delegate Sub ReadInputReportDelegate _
        (ByVal hidHandle As SafeFileHandle, _
        ByVal readHandle As SafeFileHandle, _
        ByVal writeHandle As SafeFileHandle, _
        ByRef myDeviceDetected As Boolean, _
        ByRef readBuffer() As Byte, _
        ByRef success As Boolean)

    ' This delegate has the same parameters as AccessForm.
    ' Used in accessing the application's form from a different thread.

    Private Delegate Sub MarshalToForm _
        (ByVal action As String, _
        ByVal textToAdd As String)

    ''' <summary>
    ''' Called when a WM_DEVICECHANGE message has arrived,
    ''' indicating that a device has been attached or removed.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ''' </summary>
    '''
    ''' <param name="m"> a message with information about the device
</param>

    Friend Sub OnDeviceChange(ByVal m As Message)

        Debug.WriteLine("WM_DEVICECHANGE")

        Try
            If (m.WParam.ToInt32 = DBT_DEVICEARRIVAL) Then

                ' If WParam contains DBT_DEVICEARRIVAL, a device has been
attached.

                Debug.WriteLine("A device has been attached.")

            ElseIf (m.WParam.ToInt32 = DBT_DEVICEREMOVECOMPLETE) Then

                ' If WParam contains DBT_DEVICEREMOVAL, a device has been
removed.

                Debug.WriteLine("A device has been removed.")

                ' Find out if it's the device we're communicating with.

                If MyDeviceManagement.DeviceNameMatch(m, myDevicePathName)
Then


                    ' Set MyDeviceDetected False so on the next data-
transfer attempt,
                    ' FindTheHid() will be called to look for the device
                    ' and get a new handle.

                    FrmMy.myDeviceDetected = False
                End If
            End If


        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try
    End Sub

    ''' <summary>
    ''' Uses a series of API calls to locate a HID-class device
    ''' by its Vendor ID and Product ID.
    ''' </summary>
    '''
    ''' <returns>
    '''  True if the device is detected, False if not detected.
    ''' </returns>

    Private Function FindTheHid() As Boolean

        Dim deviceFound As Boolean
        Dim devicePathName(127) As String
        Dim hidGuid As System.Guid
        Dim memberIndex As Int32
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Dim myProductID As Int16
        Dim myVendorID As Int16
        Dim success As Boolean

        Try
            myDeviceDetected = False

            ' Get the device's Vendor ID and Product ID from the form's
text boxes.

            GetVendorAndProductIDsFromTextBoxes(myVendorID, myProductID)

            ' ***
            ' API function: 'HidD_GetHidGuid

            ' Purpose: Retrieves the interface class GUID for the HID
class.

            ' Accepts: 'A System.Guid object for storing the GUID.
            ' ***

            HidD_GetHidGuid(hidGuid)

            Debug.WriteLine(MyDebugging.ResultOfAPICall("GetHidGuid"))
            Debug.WriteLine("  GUID for system HIDs: " & hidGuid.ToString)

            ' Fill an array with the device path names of all attached
HIDs.

            deviceFound = MyDeviceManagement.FindDeviceFromGuid _
                (hidGuid, _
                devicePathName)

            ' If there is at least one HID, attempt to read the Vendor ID
and Product ID
            ' of each device until there is a match or all devices have
been examined.

            If deviceFound Then

                memberIndex = 0

                Do
                    ' ***
                    ' API function:
                    ' CreateFile

                    ' Purpose:
                    ' Retrieves a handle to a device.

                    ' Accepts:
                    ' A device path name returned by
SetupDiGetDeviceInterfaceDetail
                    ' The type of access requested (read/write).
                    ' FILE_SHARE attributes to allow other processes to
access the device while this handle is open.
                    ' A Security structure or IntPtr.Zero.
                    ' A creation disposition value. Use OPEN_EXISTING for
devices.
                    ' Flags and attributes for files. Not used for devices.
                    ' Handle to a template file. Not used.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    ' Returns: a handle without read or write access.
                    ' This enables obtaining information about all HIDs,
even system
                    ' keyboards and mice.
                    ' Separate handles are used for reading and writing.
                    ' ***

                    hidHandle = CreateFile _
                        (devicePathName(memberIndex), _
                        0, _
                        FILE_SHARE_READ Or FILE_SHARE_WRITE, _
                        IntPtr.Zero, _
                        OPEN_EXISTING, _
                        0, _
                        0)


Debug.WriteLine(MyDebugging.ResultOfAPICall("CreateFile"))
                    Debug.WriteLine("  Returned handle: " &
hidHandle.ToString)

                    If Not (hidHandle.IsInvalid) Then

                        ' The returned handle is valid,
                        ' so find out if this is the device we're looking
for.

                        ' Set the Size property of DeviceAttributes to the
number of bytes in the structure.

                        MyHid.DeviceAttributes.Size =
Marshal.SizeOf(MyHid.DeviceAttributes)

                        ' ***
                        ' API function:
                        ' HidD_GetAttributes

                        ' Purpose:
                        ' Retrieves a HIDD_ATTRIBUTES structure containing
the Vendor ID,
                        ' Product ID, and Product Version Number for a
device.

                        ' Accepts:
                        ' A handle returned by CreateFile.
                        ' A pointer to receive a HIDD_ATTRIBUTES structure.

                        ' Returns:
                        ' True on success, False on failure.
                        ' ***

                        success = HidD_GetAttributes(hidHandle,
MyHid.DeviceAttributes)

                        If success Then

                            Debug.WriteLine("  HIDD_ATTRIBUTES structure
filled without error.")
                            Debug.WriteLine("  Structure size: " &
MyHid.DeviceAttributes.Size)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                            Debug.WriteLine("  Vendor ID: " &
Hex(MyHid.DeviceAttributes.VendorID))
                            Debug.WriteLine("  Product ID: " &
Hex(MyHid.DeviceAttributes.ProductID))
                            Debug.WriteLine("  Version Number: " &
Hex(MyHid.DeviceAttributes.VersionNumber))

                            ' Find out if the device matches the one we're
looking for.

                            If (MyHid.DeviceAttributes.VendorID =
myVendorID) And _
                                (MyHid.DeviceAttributes.ProductID =
myProductID) Then

                                Debug.WriteLine("  My device detected")

                                myDeviceDetected = True

                                ' Save the DevicePathName for
OnDeviceChange().

                                myDevicePathName =
devicePathName(memberIndex)
                            Else

                                ' It's not a match, so close the handle.

                                myDeviceDetected = False

                                hidHandle.Close()

                            End If

                        Else
                            ' There was a problem in retrieving the
information.

                            Debug.WriteLine("  Error in filling
HIDD_ATTRIBUTES structure.")
                            myDeviceDetected = False
                            hidHandle.Close()
                        End If

                    End If

                    ' Keep looking until we find the device or there are no
devices left to examine.

                    memberIndex = memberIndex + 1

                Loop Until (myDeviceDetected Or (memberIndex =
devicePathName.Length))

            End If

            If myDeviceDetected Then

                ' The device was detected.
                ' Register to receive notifications if the device is
removed or attached.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                success = MyDeviceManagement.RegisterForDeviceNotifications
_
                    (myDevicePathName, _
                    FrmMy.Handle, _
                    hidGuid, _
                    deviceNotificationHandle)

                Debug.WriteLine("RegisterForDeviceNotifications = " &
success)

                ' Learn the capabilities of the device.

                MyHid.Capabilities = MyHid.GetDeviceCapabilities(hidHandle)

                If success Then

                    ' Find out if the device is a system mouse or keyboard.

                    hidUsage = MyHid.GetHidUsage(MyHid.Capabilities)

                    ' Get the Input report buffer size.

                    GetInputReportBufferSize()

                    ' Get handles to use in requesting Input and Output
reports.

                    readHandle = CreateFile _
                        (myDevicePathName, _
                        GENERIC_READ, _
                        FILE_SHARE_READ Or FILE_SHARE_WRITE, _
                        IntPtr.Zero, _
                        OPEN_EXISTING, _
                        FILE_FLAG_OVERLAPPED, _
                        0)

Debug.WriteLine(MyDebugging.ResultOfAPICall("CreateFile, ReadHandle"))
                    Debug.WriteLine("  Returned handle: " &
readHandle.ToString)

                    If readHandle.IsInvalid Then

                        exclusiveAccess = True

                    Else
                        writeHandle = CreateFile _
                            (myDevicePathName, _
                            GENERIC_WRITE, _
                            FILE_SHARE_READ Or FILE_SHARE_WRITE, _
                            IntPtr.Zero, _
                            OPEN_EXISTING, _
                            0, _
                            0)

Debug.WriteLine(MyDebugging.ResultOfAPICall("CreateFile, WriteHandle"))
                    Debug.WriteLine("  Returned handle: " &
writeHandle.ToString)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        ' Flush any waiting reports in the input buffer.
(optional)

                        MyHid.FlushQueue(readHandle)

                    End If
                End If
            Else
                ' The device wasn't detected.

                Debug.WriteLine(" Device not found.")

            End If

            Return myDeviceDetected

        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try
    End Function

    ''' <summary>
    ''' In asynchronous ReadFiles, the callback function GetInputReportData
    ''' uses this routine to access the application's form, which runs in
    ''' a different thread.
    ''' The routine performs various application-specific functions that
    ''' involve accessing the application's form.
    ''' </summary>
    '''
    ''' <param name="action"> a string that names the action to perform on
the form</param>
    ''' <param name="formText"> text that the form displays or the code
uses for
    ''' another purpose. Actions that don't use text ignore this parameter.
</param>

    Private Sub AccessForm(ByVal action As String, ByVal formText As
String)

        Try

        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try

    End Sub

    ''' <summary>
    ''' Sends an Output report, then retrieves an Input report.
    ''' Assumes report ID = 0 for both reports.
    ''' </summary>

    Private Sub ExchangeInputAndOutputReports()

        Dim byteValue As String
        Dim count As Int32
        Dim inputReportBuffer() As Byte
        Dim outputReportBuffer() As Byte
        Dim success As Boolean
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Try
            success = False

            ' Don't attempt to exchange reports if valid handles aren't
available
            ' (as for a mouse or keyboard under Windows 2000/XP.)

            If (Not (readHandle.IsInvalid) And (Not writeHandle.IsInvalid))
Then

                ' Don't attempt to send an Output report if the HID has no
Output report.

                If (MyHid.Capabilities.OutputReportByteLength > 0) Then

                    ' Set the upper bound of the Output report buffer.
                    ' Subtract 1 from OutputReportByteLength because the
array begins at index 0.

                    ReDim
outputReportBuffer(MyHid.Capabilities.OutputReportByteLength - 1)

                    ' Store the report ID in the first byte of the buffer:

                    outputReportBuffer(0) = 0

                    ' Store the report data following the report ID.
                    ' Use the data in the combo boxes on the form.

                    outputReportBuffer(1) =
Convert.ToByte(cboByte0.SelectedIndex)

                    ' Write a report.

                    Dim myOutputReport As New
Hid.OutputReportViaControlTransfer
                    success = myOutputReport.Write(outputReportBuffer,
writeHandle)

                    If success Then

                        For count = 1 To UBound(outputReportBuffer)

                            ' Display bytes as 2-character hex strings.

                            byteValue = String.Format("{0:X2} ",
outputReportBuffer(count))

                        Next count

                    End If

                End If

                ' Read an Input report.

                success = False

                ' Don't attempt to send an Input report if the HID has no
Input report.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ' (The HID spec requires all HIDs to have an interrupt IN
endpoint,
                ' which suggests that all HIDs must support Input reports.)

                If (MyHid.Capabilities.InputReportByteLength > 0) Then

                    ' Set the size of the Input report buffer.
                    ' Subtract 1 from the value in the Capabilities
structure because
                    ' the array begins at index 0.

                    ReDim
inputReportBuffer(MyHid.Capabilities.InputReportByteLength - 1)

                    ' Read a report using a control transfer.

                    Dim myInputReport As New
Hid.InputReportViaControlTransfer

                    ' Read the report.

                    myInputReport.Read _
                        (hidHandle, _
                        readHandle, _
                        writeHandle, _
                        myDeviceDetected, _
                        inputReportBuffer, _
                        success)

                    If success Then

                        For count = 1 To UBound(inputReportBuffer)

                            ' Display bytes as 2-character Hex strings.

                            byteValue = String.Format("{0:X2} ",
inputReportBuffer(count))


                        Next count

                    End If

                    ' Enable requesting another transfer.

                    AccessForm("EnableCmdOnce", "")

                End If


            End If

        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try

    End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ''' <summary>
    ''' Perform shutdown operations.
    ''' </summary>

    Private Sub frmMain_Closed(ByVal eventSender As System.Object, ByVal
eventArgs As System.EventArgs) Handles MyBase.Closed

        Try
            Dim FILE_NAME As String = "C:\Namebase.db"
            Dim objWriter As New System.IO.StreamWriter(FILE_NAME)
            objWriter.WriteLine(name1.Text)
            objWriter.WriteLine(name2.Text)
            objWriter.WriteLine(name3.Text)
            objWriter.WriteLine(name4.Text)
            objWriter.Close()
            Shutdown()

        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try
    End Sub

    ''' <summary>
    ''' Perform startup operations.
    ''' </summary>

    Private Sub frmMain_Load(ByVal eventSender As System.Object, ByVal
eventArgs As System.EventArgs) Handles MyBase.Load

        Dim FILE_NAME As String = "C:\Namebase.db"
        If System.IO.File.Exists(FILE_NAME) = True Then
            Dim objReader As New System.IO.StreamReader(FILE_NAME)
            name1.Text = objReader.ReadLine
            name2.Text = objReader.ReadLine
            name3.Text = objReader.ReadLine
            name4.Text = objReader.ReadLine
            objReader.Close()
        End If

        Dim objWriter As New System.IO.StreamWriter(FILE_NAME)
        objWriter.WriteLine(name1.Text)
        objWriter.WriteLine(name2.Text)
        objWriter.WriteLine(name3.Text)
        objWriter.WriteLine(name4.Text)
        objWriter.Close()

        Try
            Me.picOutput.SizeMode = PictureBoxSizeMode.StretchImage
            myCam = New iCam
            FrmMy = Me
            s_h.SelectedIndex = 4
            s_m.SelectedIndex = 0
            s_am.SelectedIndex = 1
            e_h.SelectedIndex = 6
            e_m.SelectedIndex = 0
            e_am.SelectedIndex = 0
            For i As Integer = 0 To 3
                timeinout(i, 0) = "5:00 PM"
                timeinout(i, 1) = "7:00 AM"
            Next i
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Startup()
            If (myDeviceDetected = False) Then myDeviceDetected =
FindTheHid()
            If (myDeviceDetected = True) Then cboByte0.SelectedIndex = 15
            ExchangeInputAndOutputReports()

        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try

    End Sub

    ''' <summary>
    ''' Finds and displays the number of Input buffers
    ''' (the number of Input reports the host will store).
    ''' </summary>

    Private Sub GetInputReportBufferSize()

        Dim numberOfInputBuffers As Int32

        Try
            ' Get the number of input buffers.

            MyHid.GetNumberOfInputBuffers _
                (hidHandle, _
                numberOfInputBuffers)

            ' Display the result in the text box.

        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try

    End Sub

    ''' <summary>
    ''' Retrieves a Vendor ID and Product ID in hexadecimal
    ''' from the form's text boxes and converts the text to Int16s.
    ''' </summary>
    '''
    ''' <param name="myVendorID"> the Vendor ID as a Int16.</param>
    ''' <param name="myProductID"> the Product ID as a Int16. </param>

    Private Sub GetVendorAndProductIDsFromTextBoxes _
        (ByRef myVendorID As Int16, _
        ByRef myProductID As Int16)

        Try
            myVendorID = Convert.ToInt16(Val("&hC251"))
            myProductID = Convert.ToInt16(Val("&h1701"))


        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try

    End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ''' <summary>
    ''' Initialize the elements on the form.
    ''' </summary>
    '''
    Private Sub InitializeDisplay()

        Dim count As Int16
        Dim byteValue As String

        Try
            ' Create a dropdown list box for each byte to send in a report.
            ' Display the values as 2-character hex strings.

            For count = 0 To 255

                byteValue = String.Format("{0:X2} ", count)
                FrmMy.cboByte0.Items.Insert(count, byteValue)

            Next count

            ' Select a default value for each box

            FrmMy.cboByte0.SelectedIndex = 1

            ' Check the autoincrement box to increment the values each time
a report is sent.

            ' Don't allow the user to select an input report buffer size
until there is
            ' a handle to a HID.

        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try

    End Sub

    ''' <summary>
    ''' Perform actions that must execute when the program ends.
    ''' </summary>

    Private Sub Shutdown()

        Try
            ' Close open handles to the device.

            If Not hidHandle Is Nothing Then
                If Not hidHandle.IsInvalid Then
                    hidHandle.Close()
                End If
            End If

            If Not readHandle Is Nothing Then
                If Not readHandle.IsInvalid Then
                    readHandle.Close()
                End If
            End If

            If Not writeHandle Is Nothing Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                If Not writeHandle.IsInvalid Then
                    writeHandle.Close()
                End If
            End If

            ' Stop receiving notifications.


MyDeviceManagement.StopReceivingDeviceNotifications(deviceNotificationHandl
e)

        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try

    End Sub

    ''' <summary>
    ''' Perform actions that must execute when the program starts.
    ''' </summary>

    Private Sub Startup()

        Try
            MyHid = New Hid()
            InitializeDisplay()
            tmrContinuousDataCollect.Enabled = False
            tmrContinuousDataCollect.Interval = 1000

        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try

    End Sub


    ''' <summary>
    ''' Finalize method.
    ''' </summary>

    Protected Overrides Sub Finalize()
        MyBase.Finalize()
    End Sub

    ''' <summary>
    '''  Overrides WndProc to enable checking for and handling
WM_DEVICECHANGE messages.
    ''' </summary>
    '''
    ''' <param name="m"> a Windows Message </param>

    Protected Overrides Sub WndProc(ByRef m As Message)

        Try
            ' The OnDeviceChange routine processes WM_DEVICECHANGE
messages.

            If m.Msg = WM_DEVICECHANGE Then
                OnDeviceChange(m)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            End If

            ' Let the base form process the message.

            MyBase.WndProc(m)

        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try

    End Sub

    ''' <summary>
    ''' Provides a central mechanism for exception handling.
    ''' Displays a message box that describes the exception.
    ''' </summary>
    '''
    ''' <param name="moduleName"> the module where the exception occurred.
</param>
    ''' <param name="e"> the exception </param>

    Shared Sub DisplayException(ByVal moduleName As String, ByVal e As
Exception)

        Dim message As String
        Dim caption As String

        ' Create an error message.

        message = "Exception: " & e.Message & ControlChars.CrLf & _
        "Module: " & moduleName & ControlChars.CrLf & _
        "Method: " & e.TargetSite.Name

        caption = "Unexpected Exception"

        MessageBox.Show(message, caption, MessageBoxButtons.OK)
        Debug.Write(message)

    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        Dim dateTimeInfo As DateTime = DateTime.Now
        showtime.Text = dateTimeInfo.ToString("T")
        If dateTimeInfo.ToString("ss") = "00" Then onoff()
    End Sub

    Private Sub add_t_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles add_t.Click
        Dim opentime As String
        opentime = s_h.Text + ":" + s_m.Text + " " + s_am.Text + " - " +
e_h.Text + ":" + e_m.Text + " " + e_am.Text
        If s_l1.Checked = True Then
            time_l1.Text = opentime
            timeinout(0, 0) = s_h.Text + ":" + s_m.Text + " " + s_am.Text
            timeinout(0, 1) = e_h.Text + ":" + e_m.Text + " " + e_am.Text
        End If

        If s_l2.Checked = True Then
            time_l2.Text = opentime
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            timeinout(1, 0) = s_h.Text + ":" + s_m.Text + " " + s_am.Text
            timeinout(1, 1) = e_h.Text + ":" + e_m.Text + " " + e_am.Text
        End If
        If s_l3.Checked = True Then
            time_l3.Text = opentime
            timeinout(2, 0) = s_h.Text + ":" + s_m.Text + " " + s_am.Text
            timeinout(2, 1) = e_h.Text + ":" + e_m.Text + " " + e_am.Text
        End If
        If s_l4.Checked = True Then
            time_l4.Text = opentime
            timeinout(3, 0) = s_h.Text + ":" + s_m.Text + " " + s_am.Text
            timeinout(3, 1) = e_h.Text + ":" + e_m.Text + " " + e_am.Text
        End If
    End Sub

    Private Sub LED1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles LED1.Click
        Try
            If (myDeviceDetected = False) Then myDeviceDetected =
FindTheHid()

            If LED1.BackColor = Color.Red Then
                LED1.BackColor = Color.Black
                If (myDeviceDetected = True) Then cboByte0.SelectedIndex =
cboByte0.SelectedIndex + 1
            Else
                LED1.BackColor = Color.Red
                If (myDeviceDetected = True) Then cboByte0.SelectedIndex =
cboByte0.SelectedIndex - 1
            End If

            ExchangeInputAndOutputReports()

        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try
    End Sub

    Private Sub LED2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles LED2.Click
        Try
            If (myDeviceDetected = False) Then myDeviceDetected =
FindTheHid()

            If LED2.BackColor = Color.Red Then
                LED2.BackColor = Color.Black
                If (myDeviceDetected = True) Then cboByte0.SelectedIndex =
cboByte0.SelectedIndex + 2
            Else
                LED2.BackColor = Color.Red
                If (myDeviceDetected = True) Then cboByte0.SelectedIndex =
cboByte0.SelectedIndex - 2
            End If

            ExchangeInputAndOutputReports()

        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End Sub

    Private Sub LED3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles LED3.Click
        Try
            If (myDeviceDetected = False) Then myDeviceDetected =
FindTheHid()

            If LED3.BackColor = Color.Red Then
                LED3.BackColor = Color.Black
                If (myDeviceDetected = True) Then cboByte0.SelectedIndex =
cboByte0.SelectedIndex + 4
            Else
                LED3.BackColor = Color.Red
                If (myDeviceDetected = True) Then cboByte0.SelectedIndex =
cboByte0.SelectedIndex - 4
            End If

            ExchangeInputAndOutputReports()

        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try
    End Sub

    Private Sub LED4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles LED4.Click
        Try
            If (myDeviceDetected = False) Then myDeviceDetected =
FindTheHid()

            If LED4.BackColor = Color.Red Then
                LED4.BackColor = Color.Black
                If (myDeviceDetected = True) Then cboByte0.SelectedIndex =
cboByte0.SelectedIndex + 8
            Else
                LED4.BackColor = Color.Red
                If (myDeviceDetected = True) Then cboByte0.SelectedIndex =
cboByte0.SelectedIndex - 8
            End If

            ExchangeInputAndOutputReports()

        Catch ex As Exception
            DisplayException(Me.Name, ex)
            Throw
        End Try
    End Sub

    Private Sub onoff()
        Dim dateTimeInfo As DateTime = DateTime.Now
        Dim tmp_s As String
        Dim tmp_i As Integer
        tmp_s = dateTimeInfo.ToString("T")
        tmp_i = tmp_s.Length
        If tmp_i = 11 Then
            tmp_s = tmp_s(0) + tmp_s(1) + tmp_s(2) + tmp_s(3) + tmp_s(4) +
tmp_s(8) + tmp_s(9) + tmp_s(10)
        Else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            tmp_s = tmp_s(0) + tmp_s(1) + tmp_s(2) + tmp_s(3) + tmp_s(7) +
tmp_s(8) + tmp_s(9)
        End If

        If LED1.BackColor <> Color.Red Then
            If timeinout(0, 0) = tmp_s Then
                If (myDeviceDetected = False) Then myDeviceDetected =
FindTheHid()
                LED1.BackColor = Color.Red
                If (myDeviceDetected = True) Then cboByte0.SelectedIndex =
cboByte0.SelectedIndex - 1
                ExchangeInputAndOutputReports()
            End If
        End If

        If LED1.BackColor <> Color.Black Then
            If timeinout(0, 1) = tmp_s Then
                If (myDeviceDetected = False) Then myDeviceDetected =
FindTheHid()
                LED1.BackColor = Color.Black
                If (myDeviceDetected = True) Then cboByte0.SelectedIndex =
cboByte0.SelectedIndex + 1
                ExchangeInputAndOutputReports()
            End If
        End If

        If LED2.BackColor <> Color.Red Then
            If timeinout(1, 0) = tmp_s Then
                If (myDeviceDetected = False) Then myDeviceDetected =
FindTheHid()
                LED2.BackColor = Color.Red
                If (myDeviceDetected = True) Then cboByte0.SelectedIndex =
cboByte0.SelectedIndex - 2
                ExchangeInputAndOutputReports()
            End If
        End If

        If LED2.BackColor <> Color.Black Then
            If timeinout(1, 1) = tmp_s Then
                If (myDeviceDetected = False) Then myDeviceDetected =
FindTheHid()
                LED2.BackColor = Color.Black
                If (myDeviceDetected = True) Then cboByte0.SelectedIndex =
cboByte0.SelectedIndex + 2
                ExchangeInputAndOutputReports()
            End If
        End If

        If LED3.BackColor <> Color.Red Then
            If timeinout(2, 0) = tmp_s Then
                If (myDeviceDetected = False) Then myDeviceDetected =
FindTheHid()
                LED3.BackColor = Color.Red
                If (myDeviceDetected = True) Then cboByte0.SelectedIndex =
cboByte0.SelectedIndex - 4
                ExchangeInputAndOutputReports()
            End If
        End If

        If LED3.BackColor <> Color.Black Then
            If timeinout(2, 1) = tmp_s Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                If (myDeviceDetected = False) Then myDeviceDetected =
FindTheHid()
                LED3.BackColor = Color.Black
                If (myDeviceDetected = True) Then cboByte0.SelectedIndex =
cboByte0.SelectedIndex + 4
                ExchangeInputAndOutputReports()
            End If
        End If

        If LED4.BackColor <> Color.Red Then
            If timeinout(3, 0) = tmp_s Then
                If (myDeviceDetected = False) Then myDeviceDetected =
FindTheHid()
                LED4.BackColor = Color.Red
                If (myDeviceDetected = True) Then cboByte0.SelectedIndex =
cboByte0.SelectedIndex - 8
                ExchangeInputAndOutputReports()
            End If
        End If

        If LED4.BackColor <> Color.Black Then
            If timeinout(3, 1) = tmp_s Then
                If (myDeviceDetected = False) Then myDeviceDetected =
FindTheHid()
                LED4.BackColor = Color.Black
                If (myDeviceDetected = True) Then cboByte0.SelectedIndex =
cboByte0.SelectedIndex + 8
                ExchangeInputAndOutputReports()
            End If
        End If

    End Sub

    Private Sub s_web_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles s_web.Click
        If s_web.Text = "Start" Then
            myCam.initCam(Me.picOutput.Handle.ToInt32)
            s_web.Text = "Stop"
        Else
            myCam.closeCam()
            s_web.Text = "Start"
        End If
    End Sub

    Private Sub Label7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label7.Click

    End Sub
End Class
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code ของ ARM7

```
#include "type.h"

#include "usb.h"

#include "usbcfg.h"

#include "usbhw.h"

#include "usbcore.h"

#include "usbdesc.h"

#include "usbuser.h"

#if (USB_AUDIO)

#include "audio.h"

#include "adcuser.h"

#endif

#if (USB_HID)

#include "hid.h"

#include "hiduser.h"

#endif

#if (USB_MSC)

#include "msc.h"

#include "mscuser.h"

#endif
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#pragma diag_suppress 111,1441

WORD  USB_DeviceStatus;

BYTE  USB_DeviceAddress;

BYTE  USB_Configuration;

DWORD USB_EndPointMask;

DWORD USB_EndPointHalt;

BYTE  USB_NumInterfaces;

BYTE  USB_AltSetting[USB_IF_NUM];

BYTE  EP0Buf[USB_MAX_PACKET0];

USB_EP_DATA EP0Data;

USB_SETUP_PACKET SetupPacket;

/*

 * Reset USB Core

 *   Parameters:     None

 *   Return Value:   None

 */

void USB_ResetCore (void) {

 USB_DeviceStatus  = USB_POWER;

 USB_DeviceAddress = 0;

 USB_Configuration = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 USB_EndPointMask  = 0x00010001;
 USB_EndPointHalt  = 0x00000000;
}
/*
 *  USB Request - Setup Stage
 *    Parameters:     None (global SetupPacket)
 *    Return Value:   None
 */
void USB_SetupStage (void) {
 USB_ReadEP(0x00, (BYTE *)&SetupPacket);
}
/*
 *  USB Request - Data In Stage
 *    Parameters:     None (global EP0Data)
 *    Return Value:   None
 */
void USB_DataInStage (void) {
 DWORD cnt;
 if (EP0Data.Count > USB_MAX_PACKET0) {
   cnt = USB_MAX_PACKET0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  } else {
    cnt = EP0Data.Count;
  }
  cnt = USB_WriteEP(0x80, EP0Data.pData, cnt);
  EP0Data.pData += cnt;
  EP0Data.Count -= cnt;
}
/*
 *  USB Request - Data Out Stage
 *    Parameters:     None (global EP0Data)
 *    Return Value:   None
 */
void USB_DataOutStage (void) {
  DWORD cnt;
  cnt = USB_ReadEP(0x00, EP0Data.pData);
  EP0Data.pData += cnt;
  EP0Data.Count -= cnt;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * USB Request - Status In Stage
 *   Parameters:     None
 *   Return Value:   None
 */
void USB_StatusInStage (void) {
  USB_WriteEP(0x80, NULL, 0);
}
/*
 * USB Request - Status Out Stage
 *   Parameters:     None
 *   Return Value:   None
 */
void USB_StatusOutStage (void) {
  USB_ReadEP(0x00, EP0Buf);
}
/*
 * Get Status USB Request
 *   Parameters:     None (global SetupPacket)
 *   Return Value:   TRUE - Success, FALSE - Error
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
*/

__inline BOOL USB_GetStatus (void) {

 DWORD n, m;

 switch (SetupPacket.bmRequestType.BM.Recipient) {

  case REQUEST_TO_DEVICE:

   EP0Data.pData = (BYTE *)&USB_DeviceStatus;

   USB_DataInStage();

   break;

  case REQUEST_TO_INTERFACE:

   if ((USB_Configuration != 0) && (SetupPacket.wIndex.WB.L < USB_NumInterfaces)) {

    *((__packed WORD *)EP0Buf) = 0;

    EP0Data.pData = EP0Buf;

    USB_DataInStage();

   } else {

    return (FALSE);

   }

   break;

  case REQUEST_TO_ENDPOINT:

   n = SetupPacket.wIndex.WB.L & 0x8F;

   m = (n & 0x80) ? ((1 << 16) << (n & 0x0F)) : (1 << n);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (((USB_Configuration != 0) || ((n & 0x0F) == 0)) && (USB_EndPointMask & m)) {
      *((__packed WORD *)EP0Buf) = (USB_EndPointHalt & m) ? 1 : 0;
      EP0Data.pData = EP0Buf;
      USB_DataInStage();
    } else {
      return (FALSE);
    }
    break;
  default:
    return (FALSE);
 }
 return (TRUE);
}
/*
 * Set/Clear Feature USB Request
 *   Parameters:     sc:   0 - Clear, 1 - Set
 *                   None (global SetupPacket)
 *   Return Value:   TRUE - Success, FALSE - Error
 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
__inline BOOL USB_SetClrFeature (DWORD sc) {

 DWORD n, m;

 switch (SetupPacket.bmRequestType.BM.Recipient) {

  case REQUEST_TO_DEVICE:

   if (SetupPacket.wValue.W == USB_FEATURE_REMOTE_WAKEUP) {

    if (sc) {

     USB_WakeUpCfg(TRUE);

     USB_DeviceStatus |= USB_GETSTATUS_REMOTE_WAKEUP;

    } else {

     USB_WakeUpCfg(FALSE);

     USB_DeviceStatus &= ~USB_GETSTATUS_REMOTE_WAKEUP;

    }

   } else {

    return (FALSE);

   }

   break;

  case REQUEST_TO_INTERFACE:

   return (FALSE);

  case REQUEST_TO_ENDPOINT:

   n = SetupPacket.wIndex.WB.L & 0x8F;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  m = (n & 0x80) ? ((1 << 16) << (n & 0x0F)) : (1 << n);
  if ((USB_Configuration != 0) && ((n & 0x0F) != 0) && (USB_EndPointMask & m)) {
    if (SetupPacket.wValue.W == USB_FEATURE_ENDPOINT_STALL) {
      if (sc) {
        USB_SetStallEP(n);
        USB_EndPointHalt |= m;
      } else {
        USB_ClrStallEP(n);
        USB_EndPointHalt &= ~m;
      }
    } else {
      return (FALSE);
    }
  } else {
    return (FALSE);
  }
  break;
default:
  return (FALSE);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 return (TRUE);
}
/*
 * Get Descriptor USB Request
 *   Parameters:     None (global SetupPacket)
 *   Return Value:   TRUE - Success, FALSE - Error
 */
__inline BOOL USB_GetDescriptor (void) {
 BYTE  *pD;
 DWORD len, n;
 switch (SetupPacket.bmRequestType.BM.Recipient) {
  case REQUEST_TO_DEVICE:
    switch (SetupPacket.wValue.WB.H) {
      case USB_DEVICE_DESCRIPTOR_TYPE:
        EP0Data.pData = (BYTE *)USB_DeviceDescriptor;
        len = USB_DEVICE_DESC_SIZE;
        break;
      case USB_CONFIGURATION_DESCRIPTOR_TYPE:
        pD = (BYTE *)USB_ConfigDescriptor;
        for (n = 0; n != SetupPacket.wValue.WB.L; n++) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (((USB_CONFIGURATION_DESCRIPTOR *)pD)->bLength != 0) {
      pD += ((USB_CONFIGURATION_DESCRIPTOR *)pD)->wTotalLength;
    }
  }
  if (((USB_CONFIGURATION_DESCRIPTOR *)pD)->bLength == 0) {
    return (FALSE);
  }
  EP0Data.pData = pD;
  len = ((USB_CONFIGURATION_DESCRIPTOR *)pD)->wTotalLength;
  break;
case USB_STRING_DESCRIPTOR_TYPE:
  EP0Data.pData = (BYTE *)USB_StringDescriptor + SetupPacket.wValue.WB.L;
  len = ((USB_STRING_DESCRIPTOR *)EP0Data.pData)->bLength;
  break;
default:
  return (FALSE);
}
break;
case REQUEST_TO_INTERFACE:
 switch (SetupPacket.wValue.WB.H) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#if USB_HID
      case HID_HID_DESCRIPTOR_TYPE:
       if (SetupPacket.wIndex.WB.L != USB_HID_IF_NUM) {
        return (FALSE);    /* Only Single HID Interface is supported */
       }
       EP0Data.pData = (BYTE *)USB_ConfigDescriptor + HID_DESC_OFFSET;
       len = HID_DESC_SIZE;
       break;
      case HID_REPORT_DESCRIPTOR_TYPE:
       if (SetupPacket.wIndex.WB.L != USB_HID_IF_NUM) {
        return (FALSE);    /* Only Single HID Interface is supported */
       }
       EP0Data.pData = (BYTE *)HID_ReportDescriptor;
       len = HID_ReportDescSize;
       break;
      case HID_PHYSICAL_DESCRIPTOR_TYPE:
       return (FALSE);     /* HID Physical Descriptor is not supported */
#endif
      default:
       return (FALSE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    break;
  default:
    return (FALSE);
  }
  if (EP0Data.Count > len) {
   EP0Data.Count = len;
  }
  USB_DataInStage();

  return (TRUE);
}
/*
 *  Set Configuration USB Request
 *    Parameters:     None (global SetupPacket)
 *    Return Value:   TRUE - Success, FALSE - Error
 */
__inline BOOL USB_SetConfiguration (void) {
  USB_COMMON_DESCRIPTOR *pD;
  DWORD                 alt, n, m;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (SetupPacket.wValue.WB.L) {
  pD = (USB_COMMON_DESCRIPTOR *)USB_ConfigDescriptor;
  while (pD->bLength) {
    switch (pD->bDescriptorType) {
      case USB_CONFIGURATION_DESCRIPTOR_TYPE:
        if (((USB_CONFIGURATION_DESCRIPTOR *)pD)->bConfigurationValue ==
SetupPacket.wValue.WB.L) {
          USB_Configuration = SetupPacket.wValue.WB.L;
          USB_NumInterfaces = ((USB_CONFIGURATION_DESCRIPTOR *)pD)->bNumInterfaces;
          for (n = 0; n < USB_IF_NUM; n++) {
            USB_AltSetting[n] = 0;
          }
          for (n = 1; n < 16; n++) {
            if (USB_EndPointMask & (1 << n)) {
              USB_DisableEP(n);
            }
            if (USB_EndPointMask & ((1 << 16) << n)) {
              USB_DisableEP(n | 0x80);
            }
          }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        USB_EndPointMask = 0x00010001;

        USB_EndPointHalt = 0x00000000;

        USB_Configure(TRUE);

        if (((USB_CONFIGURATION_DESCRIPTOR *)pD)->bmAttributes &
USB_CONFIG_SELF_POWERED) {

          USB_DeviceStatus |= USB_GETSTATUS_SELF_POWERED;

        } else {

          USB_DeviceStatus &= ~USB_GETSTATUS_SELF_POWERED;

        }

       } else {

        (BYTE *)pD += ((USB_CONFIGURATION_DESCRIPTOR *)pD)->wTotalLength;

        continue;

       }

       break;

      case USB_INTERFACE_DESCRIPTOR_TYPE:

       alt = ((USB_INTERFACE_DESCRIPTOR *)pD)->bAlternateSetting;

       break;

      case USB_ENDPOINT_DESCRIPTOR_TYPE:

       if (alt == 0) {

        n = ((USB_ENDPOINT_DESCRIPTOR *)pD)->bEndpointAddress & 0x8F;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        m = (n & 0x80) ? ((1 << 16) << (n & 0x0F)) : (1 << n);
        USB_EndPointMask |= m;
        USB_ConfigEP((USB_ENDPOINT_DESCRIPTOR *)pD);
        USB_EnableEP(n);
        USB_ResetEP(n);
      }
      break;
    }
    (BYTE *)pD += pD->bLength;
  }
}
else {
 USB_Configuration = 0;
 for (n = 1; n < 16; n++) {
  if (USB_EndPointMask & (1 << n)) {
   USB_DisableEP(n);
  }
  if (USB_EndPointMask & ((1 << 16) << n)) {
   USB_DisableEP(n | 0x80);
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }

  USB_EndPointMask  = 0x00010001;

  USB_EndPointHalt  = 0x00000000;

  USB_Configure(FALSE);

 }

 if (USB_Configuration == SetupPacket.wValue.WB.L) {

  return (TRUE);

 } else {

  return (FALSE);

 }

}

/*

 *  Set Interface USB Request

 *    Parameters:      None (global SetupPacket)

 *    Return Value:    TRUE - Success, FALSE - Error

 */

__inline BOOL USB_SetInterface (void) {

 USB_COMMON_DESCRIPTOR *pD;

 DWORD              ifn, alt, old, msk, n, m;

 BOOL               set;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (USB_Configuration == 0) return (FALSE);

set = FALSE;

pD  = (USB_COMMON_DESCRIPTOR *)USB_ConfigDescriptor;

while (pD->bLength) {

 switch (pD->bDescriptorType) {

  case USB_CONFIGURATION_DESCRIPTOR_TYPE:

   if (((USB_CONFIGURATION_DESCRIPTOR *)pD)->bConfigurationValue !=
USB_Configuration) {

    (BYTE *)pD += ((USB_CONFIGURATION_DESCRIPTOR *)pD)->wTotalLength;

    continue;

   }

   break;

  case USB_INTERFACE_DESCRIPTOR_TYPE:

   ifn = ((USB_INTERFACE_DESCRIPTOR *)pD)->bInterfaceNumber;

   alt = ((USB_INTERFACE_DESCRIPTOR *)pD)->bAlternateSetting;

   msk = 0;

   if ((ifn == SetupPacket.wIndex.WB.L) && (alt == SetupPacket.wValue.WB.L)) {

    set = TRUE;

    old = USB_AltSetting[ifn];

    USB_AltSetting[ifn] = (BYTE)alt;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
  break;
case USB_ENDPOINT_DESCRIPTOR_TYPE:
  if (ifn == SetupPacket.wIndex.WB.L) {
    n = ((USB_ENDPOINT_DESCRIPTOR *)pD)->bEndpointAddress & 0x8F;
    m = (n & 0x80) ? ((1 << 16) << (n & 0x0F)) : (1 << n);
    if (alt == SetupPacket.wValue.WB.L) {
      USB_EndPointMask |= m;
      USB_EndPointHalt &= ~m;
      USB_ConfigEP((USB_ENDPOINT_DESCRIPTOR *)pD);
      USB_EnableEP(n);
      USB_ResetEP(n);
      msk |= m;
    }
    else if ((alt == old) && ((msk & m) == 0)) {
      USB_EndPointMask &= ~m;
      USB_EndPointHalt &= ~m;
      USB_DisableEP(n);
    }
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      break;
    }
   (BYTE *)pD += pD->bLength;
  }
 return (set);
}
/*
 *  USB Endpoint 0 Event Callback
 *    Parameter:      event
 */
void USB_EndPoint0 (DWORD event) {
 switch (event) {
   case USB_EVT_SETUP:
     USB_SetupStage();
     USB_DirCtrlEP(SetupPacket.bmRequestType.BM.Dir);
     EP0Data.Count = SetupPacket.wLength;
     switch (SetupPacket.bmRequestType.BM.Type) {
       case REQUEST_STANDARD:
         switch (SetupPacket.bRequest) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case USB_REQUEST_GET_STATUS:
          if (!USB_GetStatus()) {
            goto stall_i;
          }
          break;
        case USB_REQUEST_CLEAR_FEATURE:
          if (!USB_SetClrFeature(0)) {
            goto stall_i;
          }
          USB_StatusInStage();
#if USB_FEATURE_EVENT
          USB_Feature_Event();
#endif
          break;
        case USB_REQUEST_SET_FEATURE:
          if (!USB_SetClrFeature(1)) {
            goto stall_i;
          }
          USB_StatusInStage();
#if USB_FEATURE_EVENT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            USB_Feature_Event();
#endif
            break;
          case USB_REQUEST_SET_ADDRESS:
            switch (SetupPacket.bmRequestType.BM.Recipient) {
              case REQUEST_TO_DEVICE:
                USB_DeviceAddress = 0x80 | SetupPacket.wValue.WB.L;
                USB_StatusInStage();
                break;
              default:
                goto stall_i;
            }
            break;
          case USB_REQUEST_GET_DESCRIPTOR:
            if (!USB_GetDescriptor()) {
              goto stall_i;
            }
            break;
          case USB_REQUEST_SET_DESCRIPTOR:
/*stall_o:*/  USB_SetStallEP(0x00);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    EP0Data.Count = 0;
    break;
  case USB_REQUEST_GET_CONFIGURATION:
    switch (SetupPacket.bmRequestType.BM.Recipient) {
      case REQUEST_TO_DEVICE:
        EP0Data.pData = &USB_Configuration;
        USB_DataInStage();
        break;
      default:
        goto stall_i;
    }
    break;
  case USB_REQUEST_SET_CONFIGURATION:
    switch (SetupPacket.bmRequestType.BM.Recipient) {
      case REQUEST_TO_DEVICE:
        if (!USB_SetConfiguration()) {
          goto stall_i;
        }
        USB_StatusInStage();
#if USB_CONFIGURE_EVENT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            USB_Configure_Event();

#endif

            break;

           default:

            goto stall_i;

          }

          break;

        case USB_REQUEST_GET_INTERFACE:

          switch (SetupPacket.bmRequestType.BM.Recipient) {

           case REQUEST_TO_INTERFACE:

            if ((USB_Configuration != 0) &&

               (SetupPacket.wIndex.WB.L < USB_NumInterfaces)) {

             EP0Data.pData = USB_AltSetting + SetupPacket.wIndex.WB.L;

             USB_DataInStage();

            } else {

              goto stall_i;

             }

             break;

           default:

            goto stall_i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        break;
      case USB_REQUEST_SET_INTERFACE:
        switch (SetupPacket.bmRequestType.BM.Recipient) {
          case REQUEST_TO_INTERFACE:
            if (!USB_SetInterface()) {
              goto stall_i;
            }
            USB_StatusInStage();
#if USB_INTERFACE_EVENT
            USB_Interface_Event();
#endif
            break;
          default:
            goto stall_i;
        }
        break;
      default:
        goto stall_i;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;

      case REQUEST_CLASS:

#if USB_CLASS

        switch (SetupPacket.bmRequestType.BM.Recipient) {

         case REQUEST_TO_INTERFACE:

#if USB_HID

           if (SetupPacket.wIndex.WB.L == USB_HID_IF_NUM) {

            switch (SetupPacket.bRequest) {

             case HID_REQUEST_GET_REPORT:

              if (HID_GetReport()) {

               EP0Data.pData = EP0Buf;

               USB_DataInStage();

               goto class_ok;

              }

              break;

             case HID_REQUEST_SET_REPORT:

              EP0Data.pData = EP0Buf;

              goto class_ok;

             case HID_REQUEST_GET_IDLE:

              if (HID_GetIdle()) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    EP0Data.pData = EP0Buf;
    USB_DataInStage();
    goto class_ok;
  }
  break;
case HID_REQUEST_SET_IDLE:
  if (HID_SetIdle()) {
    USB_StatusInStage();
    goto class_ok;
  }
  break;
case HID_REQUEST_GET_PROTOCOL:
  if (HID_GetProtocol()) {
    EP0Data.pData = EP0Buf;
    USB_DataInStage();
    goto class_ok;
  }
  break;
case HID_REQUEST_SET_PROTOCOL:
  if (HID_SetProtocol()) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            USB_StatusInStage();
            goto class_ok;
          }
          break;
        }
      }
#endif  /* USB_HID */
#if USB_MSC
      if (SetupPacket.wIndex.WB.L == USB_MSC_IF_NUM) {
        switch (SetupPacket.bRequest) {
          case MSC_REQUEST_RESET:
            if (MSC_Reset()) {
              USB_StatusInStage();
              goto class_ok;
            }
            break;
          case MSC_REQUEST_GET_MAX_LUN:
            if (MSC_GetMaxLUN()) {
              EP0Data.pData = EP0Buf;
              USB_DataInStage();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            goto class_ok;
          }
          break;
        }
      }
#endif  /* USB_MSC */
#if USB_AUDIO
      if ((SetupPacket.wIndex.WB.L == USB_ADC_CIF_NUM) ||
        (SetupPacket.wIndex.WB.L == USB_ADC_SIF1_NUM) ||
        (SetupPacket.wIndex.WB.L == USB_ADC_SIF2_NUM)) {
       if (SetupPacket.bmRequestType.BM.Dir) {
        if (ADC_IF_GetRequest()) {
          EP0Data.pData = EP0Buf;
          USB_DataInStage();
          goto class_ok;
        }
       } else {
        EP0Data.pData = EP0Buf;
        goto class_ok;
       }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
#endif  /* USB_AUDIO */
        goto stall_i;
#if USB_AUDIO
      case REQUEST_TO_ENDPOINT:
        if (SetupPacket.bmRequestType.BM.Dir) {
          if (ADC_EP_GetRequest()) {
            EP0Data.pData = EP0Buf;
            USB_DataInStage();
            goto class_ok;
          }
        } else {
          EP0Data.pData = EP0Buf;
          goto class_ok;
        }
        goto stall_i;
#endif  /* USB_AUDIO */
      default:
        goto stall_i;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
class_ok: break;
#else
        goto stall_i;
#endif  /* USB_CLASS */
      case REQUEST_VENDOR:
        goto stall_i;
      default:
stall_i:  USB_SetStallEP(0x80);
        EP0Data.Count = 0;
        break;
    }
    break;
  case USB_EVT_OUT:
    if (SetupPacket.bmRequestType.BM.Dir == 0) {
      if (EP0Data.Count) {
        USB_DataOutStage();
        if (EP0Data.Count == 0) {
          switch (SetupPacket.bmRequestType.BM.Type) {
            case REQUEST_STANDARD:
              goto stall_i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#if (USB_CLASS)
        case REQUEST_CLASS:
         switch (SetupPacket.bmRequestType.BM.Recipient) {
          case REQUEST_TO_INTERFACE:
#if USB_HID
            if (SetupPacket.wIndex.WB.L == USB_HID_IF_NUM) {
             if (!HID_SetReport()) {
              goto stall_i;
             }
             break;
            }
#endif
#if USB_AUDIO
            if ((SetupPacket.wIndex.WB.L == USB_ADC_CIF_NUM) ||
              (SetupPacket.wIndex.WB.L == USB_ADC_SIF1_NUM) ||
              (SetupPacket.wIndex.WB.L == USB_ADC_SIF2_NUM)) {
             if (!ADC_IF_SetRequest()) {
              goto stall_i;
             }
             break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
#endif
            goto stall_i;
          case REQUEST_TO_ENDPOINT:
#if USB_AUDIO
            if (ADC_EP_SetRequest()) break;
#endif
            goto stall_i;
          default:
            goto stall_i;
        }
        break;
#endif
      default:
        goto stall_i;
    }
    USB_StatusInStage();
  }
 }
} else {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    USB_StatusOutStage();
  }
  break;
case USB_EVT_IN:
  if (SetupPacket.bmRequestType.BM.Dir == 1) {
    USB_DataInStage();
  } else {
    if (USB_DeviceAddress & 0x80) {
      USB_DeviceAddress &= 0x7F;
      USB_SetAddress(USB_DeviceAddress);
    }
  }
  break;
case USB_EVT_IN_STALL:
  USB_ClrStallEP(0x80);
  break;
case USB_EVT_OUT_STALL:
  USB_ClrStallEP(0x00);
  break;
}}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้