สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบจัดการอัจฉริยะเพื่อประหยัดการใช้พลังงานไฟฟ้า

EMBEDDED SYSTEM FOR ELECTRICAL ENERGY MANAGEMENT

โดย

นายเดชสวัสดิ์ ว่องไวพานิช

นายทรงธรรม หมู่หมื่นศรี

นายทศพล เดชทวี

วศ.
ว 8372
2549

เลขหมู่.....
เลขทะเบียน 72725
วัน,เดือน,ปี 22 ส.ค. 2550

b 11771945
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EMBEDDED SYSTEM FOR ELECTRICAL ENERGY MANAGEMENT

BY

MR. DEJSAWAT WONGWAIPANICH

MR. SONGTHAM MOOMUENSRI

MR. TOSAPOL DETTAWEE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2006

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบจัดการอัจฉริยะเพื่อประหยัดการใช้พลังงานไฟฟ้า		
ชื่อนักศึกษา	นายเดชสวัสดิ์	ว่องไวพานิช	รหัสประจำตัว 46010242
	นายทรงธรรม	หมู่หมื่นศรี	รหัสประจำตัว 46010249
	นายทศพล	เดชทวี	รหัสประจำตัว 46010257
อาจารย์ที่ปรึกษา	อ.สถาพร พรหมวงศ์		
อาจารย์ที่ปรึกษาร่วม	ผศ.มนต์ชัย แช่มช้อย		
ระดับการศึกษา	ปริญญาตรีวิศวกรรมศาสตร์บัณฑิต		
	สาขาวิศวกรรมสารสนเทศ		
ภาควิชา	วิศวกรรมสารสนเทศ		
ปีการศึกษา	2549		

ปริญญานิพนธ์นี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเป็นที่เรียบร้อยแล้ว

.....

(อ.สถาพร พรหมวงศ์)

อาจารย์ที่ปรึกษา

.....

(ผศ.มนต์ชัย แช่มช้อย)

อาจารย์ที่ปรึกษาร่วม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบจัดการอัจฉริยะเพื่อประหยัดการใช้พลังงานไฟฟ้า
ชื่อนักศึกษา	นายเดชสวัสดิ์ ว่องไวพานิช รหัสประจำตัว 46010242
	นายทรงธรรม หมู่หมื่นศรี รหัสประจำตัว 46010249
	นายทศพล เดชทวี รหัสประจำตัว 46010257
อาจารย์ที่ปรึกษา	อ.สถาพร พรหมวงศ์
อาจารย์ที่ปรึกษาร่วม	ผศ.มนต์ชัย แช่มช้อย
ระดับการศึกษา	ปริญญาตรีวิศวกรรมศาสตร์บัณฑิต
	สาขาวิศวกรรมสารสนเทศ
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2549

บทคัดย่อ

โครงงานนี้เป็นการสร้างระบบจัดการให้กับการเปิด-ปิดไฟตามทางเดินภายในอาคาร เพื่อลดปริมาณการใช้ไฟฟ้าในกรณีที่เปิดไฟเกินความจำเป็น โดยจะใช้เซ็นเซอร์เพื่อตรวจจับว่ามีบุคคลอยู่ในพื้นที่ที่ระบบตรวจจับหรือไม่ หากตรวจจับได้ก็จะส่งสัญญาณเพื่อให้หลอดไฟบริเวณนั้นเปิดทำงาน รวมทั้งยังนำเทคโนโลยีระบบฝังตัวมาประยุกต์ใช้ร่วมกัน เพื่อประมวลผลและสามารถเข้าถึงฐานข้อมูลได้ผ่านทางเครือข่ายอินเตอร์เน็ต และยังสามารถปรับเปลี่ยนการทำงานของระบบนี้โดยผ่านระบบเครือข่ายอินเตอร์เน็ตดังกล่าวได้เช่นกัน ด้วยเหตุนี้เราสามารถควบคุมและจัดการกับการใช้ไฟฟ้าของหลอดไฟภายในอาคารได้อย่างมีประสิทธิภาพมากขึ้น ซึ่งเป็นประโยชน์โดยตรงในการลดค่าไฟฟ้าได้เป็นอย่างดี และที่สำคัญที่สุดคือช่วยลดปริมาณการใช้พลังงานไฟฟ้าของประเทศให้น้อยลงยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Embedded system for electrical energy management	
Student	Mr.Dejsawat Wongwaipanich	ID. 46010242
	Mr.Songtham Moomuensri	ID. 46010249
	Mr.Tosapol Dettawee	ID. 46010257
Advisor	Mr.Sathaporn Promwong	
Co-advisor	Asst.Prof.Monchai Chamchoy	
Graduate Level	Bachelor Degree of Information Engineering	
Department	Information Engineering	
Academic Year	2006	

Abstract

This project is the production of the interior switched electricity management system of the hallway to reduce the over electricity usage. By using PIR sensor, if there are any persons in a detection area, the sensor will signal to switch on the light in the detection area. Technology of the embedded system is applied to compile the results, and enable to access to the database by internet network, and also enable to adjust this system through the internet network. Consequently, we would be able to control, and manage the interior electricity usage more effectively. These would be benefit to reduce the cost of using electricity as well. The most important thing is it can help to reduce the using of domestic electricity.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้มีอาจสำเร็จลุล่วงไปได้หากขาดความช่วยเหลือจากบุคคลต่างๆ ในหลายๆ ฝ่าย ดังนั้นทางกลุ่มผู้จัดทำจึงขอขอบพระคุณบุคคลต่างๆ ดังต่อไปนี้

ขอขอบพระคุณ อาจารย์สถาพร พรหมวงศ์ และ ผศ.มนต์ชัย แช่มช้อย ที่ค่อยให้คำปรึกษาความช่วยเหลือทั้งทางด้านสถานที่ และเครื่องมือที่ใช้ในการทดลอง รวมถึงรุ่นพี่ปริญญาโททุกท่านที่ให้การช่วยเหลือทางด้านอุปกรณ์และให้คำแนะนำต่างๆ มาโดยตลอด

นายเดชสวัสดิ์ ว่องไวพานิช
นายทรงธรรม หมู่หมื่นศรี
นายทศพล เดชทวี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	ซ
สารบัญตาราง	ญ
บทที่ 1 บทนำ	1
1.1 บทนำ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงงาน	2
บทที่ 2 ทฤษฎีและหลักการทำงาน	3
2.1 ระบบฝังตัว (Embedded System)	3
2.1.1 ลักษณะเฉพาะของเทคโนโลยีสมองกลฝังตัว	3
2.1.2 การเทคโนโลยีสมองกลฝังตัวอย่างมีประสิทธิภาพ	4
2.1.3 ภาพรวมส่วนประกอบซอฟต์แวร์ของระบบสมองกลฝังตัว	6
2.2 เทคโนโลยีทางซอฟต์แวร์	7
2.2.1 ลักษณะเฉพาะของซอฟต์แวร์สมองกลฝังตัว(Characteristics of Embedded Software)	8
2.2.2 การประมวลผลแบบเรียลไทม์ (Real-time processing)	8
2.2.3 ส่วนประกอบของซอฟต์แวร์ (Software component)	11
2.3 เทคโนโลยีฮาร์ดแวร์	12
2.3.1 สถาปัตยกรรมของระบบสมองกลฝังตัว	12
2.3.2 อินพุตเอาต์พุตพื้นฐาน (Basic I/O)	20
2.3.3 อินพุตเอาต์พุตแบบอนุกรม (Serial I/O)	22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
2.4 ไมโครคอนโทรลเลอร์ MCS-51	23
2.4.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51	24
2.4.2 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ MCS-51	24
2.4.3 การจัดการของไมโครคอนโทรลเลอร์ตระกูล MCS-51(เบอร์ 89C52)	25
2.5 การใช้งานพอร์ตต่างๆ	26
2.5.1 พอร์ต 0 (P0.0-P0.7)	26
2.5.2 พอร์ต1 (P1.0-P1.7)	27
2.5.3 พอร์ต2 (P2.0-P2.7)	27
2.5.4 พอร์ต 3 (P3.0-P3.7)	27
2.6 ไทม์เมอร์ / เคาเตอร์ (Timer/Counter)	28
2.6.1 การทำงานในโหมดไทม์เมอร์ (Timer Mode) ของไทม์เมอร์/เคาเตอร์ 0 และ 1	29
2.6.2 การทำงานในโหมดเคาเตอร์ (Counter Mode) ของไทม์เมอร์/เคาเตอร์ 0 และ 1	29
2.6.3 รีจิสเตอร์ของไทม์เมอร์/ เคาเตอร์ 0 และ 1 (Timer / Counter0,1Register)	30
2.6.4 รีจิสเตอร์ควบคุมการทำงานของไทม์เมอร์/เคาเตอร์ 0 และ1 (TCON : Timer/Counter0,1 Control Register)	30
2.6.5 รีจิสเตอร์เลือกโหมดการทำงานของไทม์เมอร์/เคาเตอร์ 0 และ 1(TMOD : Timer/ Counter 0,1Mode Control Register)	32
2.7 กระบวนการอินเตอร์รัปต์	33
2.7.1 รีจิสเตอร์ที่เกี่ยวข้องกับกระบวนการอินเตอร์รัปต์	33
2.7.2 การอินเตอร์รัปต์จากภายนอก	39
2.7.3 การอินเตอร์รัปต์จากภายใน	40
2.8 รีเลย์ (Relay)	42
2.8.1 โครงสร้างของรีเลย์	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
2.8.2 หลักการทำงานของรีเลย์	43
2.9 ตัวต้านทานไวแสง (LDR)	43
2.9.1 โครงสร้างของ LDR	44
2.9.2 สมบัติทางแสง	44
2.10 PIR เซ็นเซอร์	45
บทที่ 3 การออกแบบโครงงาน	47
3.1 โครงสร้างการทำงานของระบบที่ออกแบบ	47
3.2 ส่วนประกอบของระบบ	48
3.2.1 ส่วนควบคุมการเปิด-ปิดหลอดไฟ	48
3.2.2 ตัวแปลงสัญญาณ RS232/RS485	50
3.2.3 ระบบฝังตัว	50
3.2.4 การติดต่อกับผู้ใช้	51
3.3 แผนผังโปรแกรม	52
บทที่ 4 ผลการทดลอง	54
4.1 การทดสอบเซ็นเซอร์	54
4.2 การทดสอบการควบคุมการเปิด-ปิดหลอดไฟ	54
4.2.1 การควบคุมของไมโครคอนโทรลเลอร์	54
4.2.2 การควบคุมจากระบบฝังตัว	57
บทที่ 5 สรุปผลโครงงาน	60
5.1 สรุปผลโครงงาน	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
5.2 ปัญหาที่พบในโครงงาน	61
5.2.1 ส่วนของโปรแกรม	61
5.2.2 ส่วนไมโครคอนโทรลเลอร์	61
5.2.3 ส่วนของอุปกรณ์ต่างๆ	62
บรรณานุกรม	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1.2 ภาพรวมส่วนประกอบฮาร์ดแวร์ของระบบสมองกลฝังตัว	5
รูปที่ 2.1.3 โครงสร้างโดยสังเขปของระบบสมองกลฝังตัว	5
รูปที่ 2.2.1 การร้องขอให้ตอบสนองภายในเวลาที่กำหนด	9
รูปที่ 2.2.2 วิธีการพรีเอมชันตามเหตุการณ์	10
รูปที่ 2.2.3 แผนภาพส่วนประกอบของซอฟต์แวร์	11
รูปที่ 2.3.1 สถาปัตยกรรมพื้นฐานของระบบสมองกลฝังตัว	13
รูปที่ 2.3.3 การกำหนดพื้นฐานของ MPU	15
รูปที่ 2.3.4 การประมวลผลแบบ pipeline	17
รูปที่ 2.3.5 ประเภทของไอซีที่ใช้เป็นหน่วยความจำ	19
รูปที่ 2.3.6 การเชื่อมต่ออินพุตเอาต์พุต	21
รูปที่ 2.6 โครงสร้างโดยรวมของไทม์เมอร์/เคาเตอร์	29
รูปที่ 2.6.4 บิตต่างๆ ในรีจิสเตอร์ TCON	30
รูปที่ 2.6.5 บิตต่างๆ ในรีจิสเตอร์ TMOD	32
รูปที่ 2.7.1 บิตต่างๆ ของรีจิสเตอร์ IE	34
รูปที่ 2.7.2 บิตต่างๆ ของรีจิสเตอร์ IP	35
รูปที่ 2.7.3 บิตต่างๆ ของรีจิสเตอร์ TCON	37
รูปที่ 2.7.4 บิตต่างๆ ของรีจิสเตอร์ T2CON	39
รูปที่ 2.8.1 รีเลย์ที่ใช้ในงานอิเล็กทรอนิกส์	42
รูปที่ 2.8.2 สัญลักษณ์ของรีเลย์	42
รูปที่ 2.8.3 สัญลักษณ์ของรีเลย์แทนโครงสร้างรีเลย์	42
รูปที่ 2.8.4 ตัวอย่างตำแหน่งขาของรีเลย์แบบ 5 ขา	43
รูปที่ 2.9 ตัวต้านทานไวแสง	43
รูปที่ 2.10 PIR Sensor	45
รูปที่ 3.1 โครงสร้างโดยรวมของระบบ	47
รูปที่ 3.2.1 ส่วนของเซ็นเซอร์	48
รูปที่ 3.2.2 ส่วนของรีเลย์	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

	หน้า
รูปที่ 3.2.3 ส่วนของแอลดีอาร์	49
รูปที่ 3.2.4 ส่วนของบอร์ดไมโครคอนโทรลเลอร์แต่ละชั้น	50
รูปที่ 3.2.5 ตัวระบบฝังตัว รุ่น Mity-mite module	51
รูปที่ 3.2.6 การควบคุมระบบผ่านทางหน้าเว็บบราวน์เซอร์	52
รูปที่ 3.3 แผนผังโปรแกรมส่วนควบคุมหลอดไฟฟลูออเรสเซนต์	53
รูปที่ 4.1.1 สัญญาณในขณะที่ไม่มีมนุษย์เคลื่อนไหว	54
รูปที่ 4.1.2 สัญญาณในขณะที่มีมนุษย์เคลื่อนไหว	54
รูปที่ 4.2.3 การควบคุมการเปิด-ปิดหลอดไฟเมื่อแอลดีอาร์ไม่ได้ทำงาน	55
รูปที่ 4.2.4 การควบคุมเมื่อแอลดีอาร์และเซ็นเซอร์ทำงาน	56
รูปที่ 4.2.5 การควบคุมเมื่อแอลดีอาร์ทำงานแต่เซ็นเซอร์ไม่ทำงาน	56
รูปที่ 4.2.6 หน้าเว็บในการเปลี่ยนค่าพารามิเตอร์	57
รูปที่ 4.2.7 หน้าเว็บแสดงสถานะการของแอลดีอาร์และหลอดไฟ	58
รูปที่ 4.2.8 หน้าเว็บในการตั้งเวลาเปิดของระบบ	58
รูปที่ 4.2.9 หน้าเว็บในการตั้งเวลาปิดของระบบ	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.5.2 หน้าที่พิเศษของพอร์ต 1	27
ตารางที่ 2.5.4 หน้าที่พิเศษของพอร์ต 3	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 บทนำ

ในปัจจุบันโลกของเรามีการใช้พลังงานจากทรัพยากรธรรมชาติมากขึ้น ซึ่งการใช้พลังงานจากทรัพยากรธรรมชาติมีแนวโน้มสูงขึ้นเรื่อยๆ โดยเฉพาะพลังงานไฟฟ้าเป็นพลังงานที่มีความสำคัญต่อการดำรงชีวิตในปัจจุบัน ทำให้ทรัพยากรธรรมชาติเหลือน้อยลง ดังนั้นทุกคนควรคำนึงถึงการใช้ประโยชน์จากพลังงานให้คุ้มค่ามากที่สุด

โครงงานนี้จะนำความสามารถของไมโครคอนโทรลเลอร์มาใช้ควบคุมการปิด-เปิดหลอดฟลูออเรสเซนต์ เพื่อช่วยประหยัดทั้งค่าใช้จ่ายและการใช้พลังงานไฟฟ้าที่มากเกินความจำเป็น โดยที่พัฒนาในระดับสูงสุด คือ สามารถติดต่อกับผู้ควบคุมระบบผ่านทางเครือข่ายอินเตอร์เน็ตได้ ซึ่งเป็นการประยุกต์ใช้ระบบฝังตัวสมองกลให้ทำงานเป็นเว็บเซิร์ฟเวอร์ (Web Server) ทำให้สะดวกในการปรับเปลี่ยนการใช้งานของระบบและง่ายต่อการบำรุงรักษา

สำหรับสถานที่ต่างๆ ที่มีการใช้ไฟฟ้าอยู่ตลอดเวลา เช่น โรงแรม โรงพยาบาล โรงพัก เป็นต้น อาจมีการใช้พลังงานไฟฟ้าเกินความจำเป็น โครงงานนี้มีความเหมาะสมที่จะนำไปประยุกต์ใช้ในสถานที่ดังกล่าวได้ ทำให้สามารถลดการใช้พลังงานไฟฟ้าและช่วยลดค่าใช้จ่ายได้

1.2 วัตถุประสงค์

1.2.1 ศึกษาการทำงานของ PIR เซ็นเซอร์ เพื่อนำมาใช้ในการตรวจจับมนุษย์ว่าอยู่ในบริเวณที่พิจารณาหรือไม่

1.2.2 ศึกษาการเขียนโปรแกรมของภาษาซี สำหรับการควบคุมและการติดต่อเชื่อมโยงข้อมูลของไมโครคอนโทรลเลอร์ MCS-51

1.2.3 ศึกษาการทำงานของระบบฝังตัว เพื่อใช้ควบคุมการทำงานต่างๆ ได้

1.2.4 เพื่อสร้างระบบควบคุมการปิด-เปิดหลอดไฟฟลูออเรสเซนต์บริเวณทางเดินภายในอาคารเรียน

1.2.5 สามารถลดการใช้พลังงานไฟฟ้าภายในอาคารเรียนได้

1.2.6 สามารถประหยัดค่าใช้จ่ายสำหรับค่าไฟฟ้าประจำเดือนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงงาน

1.3.1 สามารถเขียนโปรแกรมภาษาซี เพื่อนำมาใช้ควบคุมและติดต่อเชื่อมโยงข้อมูลของไมโครคอนโทรลเลอร์ MCS-51 ได้

1.3.2 สามารถเข้าใจการทำงานของ PIR เซ็นเซอร์

1.3.3 สามารถเข้าใจการทำงานของรีเลย์

1.3.4 สามารถเข้าใจการทำงานของไมโครคอนโทรลเลอร์ MCS-51 เพื่อใช้ในการควบคุมการปิด-เปิดหลอดไฟฟลูออเรสเซนต์

1.3.5 สามารถเขียนโปรแกรมให้ระบบฝังตัวทำงานเป็นเว็บเซิร์ฟเวอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการทำงาน

2.1 ระบบฝังตัว (Embedded System)

เทคโนโลยีสมองกลฝังตัวเกี่ยวข้องกับเทคโนโลยีหลายด้าน เช่น เทคโนโลยีซอฟแวร์ เทคโนโลยีฮาร์ดแวร์ เทคโนโลยีเครือข่ายเน็ตเวิร์ก เทคโนโลยีการสื่อสารและเทคโนโลยีกครื่องกล ดังนั้นจึงเป็นการยากที่จะนิยามว่าเทคโนโลยีสมองกลฝังตัวคืออะไร และความหลากหลายนี้ทำให้ยากต่อการเรียนรู้เทคโนโลยีสมองกลฝังตัวอย่างเป็นระบบ รวมทั้งยากที่จะสร้างผู้เชี่ยวชาญทางด้านนี้ ในบทนี้จะขออธิบายว่าระบบสมองกลฝังตัวมีลักษณะอย่างไร เทคโนโลยีสมองกลฝังตัวถูกใช้งานที่ไหนบ้าง และโครงสร้างของเทคโนโลยีสมองกลฝังตัวเป็นอย่างไร โดยจะอธิบายตามหัวข้อต่อไปนี้

2.1.1 ลักษณะเฉพาะของเทคโนโลยีสมองกลฝังตัว

สิ่งที่เรียกว่าระบบสมองกลฝังตัวนั้นมีความหลากหลายมาก ดังนั้นเทคโนโลยีสมองกลฝังตัวที่ใช้ในการออกแบบระบบสมองกลฝังตัวจึงยากต่อการจัดการให้เป็นระบบ ตัวอย่างเช่น เมื่อเปรียบเทียบระบบนำทางในรถยนต์และหม้อหุงข้าว อุปกรณ์ทั้งส่วนจะมีส่วนประกอบและฟังก์ชันที่ต่างกัน ดังนั้นความรู้และเทคโนโลยีที่ใช้ก็แตกต่างกันออกไป แต่อุปกรณ์ทั้งสองก็ใช้เทคโนโลยีสมองกลฝังตัวในการออกแบบวงจรดรรกะภายในและส่วนที่เกี่ยวกับไทมิงเหมือนกัน ถึงแม้ว่าจะเป็นอุปกรณ์ที่มีขนาดไม่เท่ากันนอกจากนี้เทคโนโลยีสมองกลฝังตัวยังใช้ในโทรศัพท์มือถือ กล้องดิจิทัล และสายการผลิตในโรงงานด้วย กล่าวคือ เทคโนโลยีนี้เริ่มกลายเป็นเทคโนโลยีพื้นฐานที่ถูกนำไปประยุกต์ใช้ในสาขาต่างๆ มากมาย

ทำไมเทคโนโลยีสมองกลฝังตัวจึงกลายเป็นเทคโนโลยีกลางที่ถูกใช้ร่วมกันในสาขาที่แตกต่างกันได้ สามารถอธิบายได้ในเชิงเทคนิคด้วยลักษณะเฉพาะ 2 อย่าง ดังนี้

(1) ฟังก์ชันงานในสาขาเหล่านั้นสามารถอธิบายได้ด้วยตรรกศาสตร์และไทมิงเหมือนกัน

(2) เพื่อไม่ให้เทคโนโลยีที่ใช้ในการพัฒนามีความหลากหลายมากเกินไป จึงนำเอาฟังก์ชันคอมพิวเตอร์มาใช้เพื่อสร้างพื้นฐานร่วมกันในการพัฒนา ด้วยวิธีการนี้จะทำให้เราสามารถลดงานด้านการออกแบบฮาร์ดแวร์ต่างๆ สำหรับงานหลากหลายประเภทได้ ส่งผลให้การพัฒนาซอฟต์แวร์กลายเป็นสิ่งที่สำคัญมากขึ้นในการพัฒนาระบบสมองกล

ด้วยลักษณะเฉพาะเช่นนี้จึงทำให้เทคโนโลยีสมองกลฝังตัวถูกนำมาใช้ โดยจะทำการแปลงฟังก์ชันเชิงตรรกและฟังก์ชันไทมิงเป็นซอฟต์แวร์ ทำให้เกิดผลตามมา 2 ประการด้วยกันคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. สัดส่วนของชิ้นงานส่วนฮาร์ดแวร์ในระบบเก่าซึ่งมีค่าสูงมากนั้นถูกทำให้ต่ำลงด้วยเทคโนโลยีสมองกลฝังตัว และองค์ประกอบเชิงฮาร์ดแวร์ที่เหลือนั้นยังถูกทำให้ไปอยู่บนชิปที่เรียกว่า "ระบบบนชิป หรือ system-on-chip" ซึ่งทำให้เราสามารถสร้างระบบที่เล็กลงไปได้อีก และทำให้จำนวนชิ้นส่วนสมองกลฝังตัวที่เป็นฮาร์ดแวร์มีจำนวนน้อยลง ส่งผลให้เกิดผลดีตามในหลายๆ ด้าน เช่น ความเสี่ยงในการซื้อชิ้นส่วนลดลงกระบวนการออกแบบและพัฒนามีความยืดหยุ่นมากขึ้น ต้นทุนการผลิตลดลง โรงงานมีขนาดเล็กลงคุณภาพของงานดีขึ้น เป็นต้น

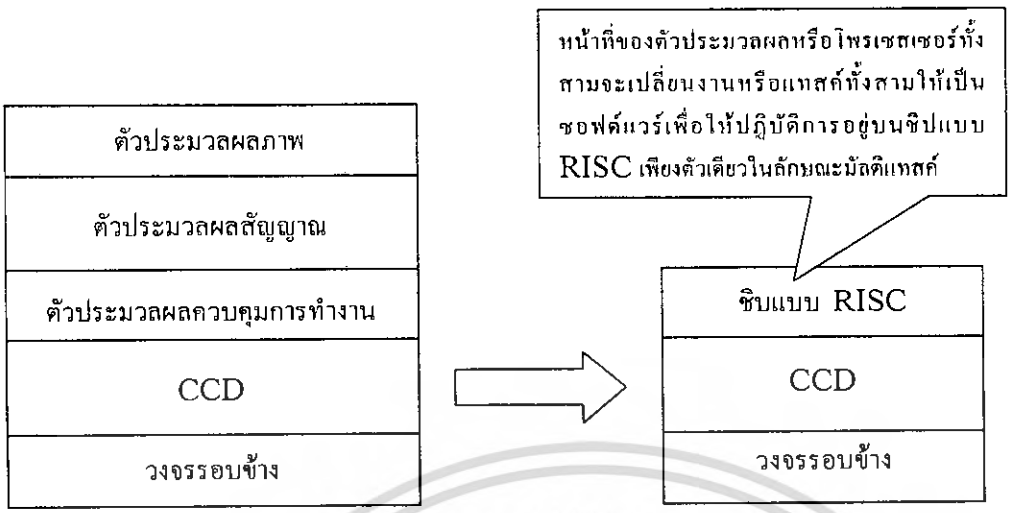
2. ทำให้เราสามารถสร้างฟังก์ชันงานที่ยากหรือซับซ้อนได้ รวมทั้งทำให้เราสามารถใส่ฟังก์ชันที่หลากหลายลงไปในอุปกรณ์ได้ และทำให้อุปกรณ์มีขนาดเล็กลง ส่งผลให้มีการพัฒนาเทคโนโลยีใหม่ๆ เพื่อพัฒนาผลิตภัณฑ์ใหม่ๆ และเกิดความเข้มแข็งเชิงการแข่งขัน อันจะนำไปสู่การพัฒนาทรัพยากรบุคคลในยุคถัดไป

2.1.2 การเทคโนโลยีสมองกลฝังตัวอย่างมีประสิทธิภาพ

ตามที่ได้กล่าวไว้แล้วข้างต้น การนิยามว่าระบบสมองกลฝังตัวคืออะไรนั้นทำได้ยาก แต่กล่าวโดยคร่าวๆ จะหมายถึง ระบบที่มีไมโครชิปกับโปรแกรมเพื่อควบคุมไมโครชิปนั้นฝังตัวอยู่ หรือระบบที่มีไอทีฝังตัวอยู่ ตัวอย่างระบบฝังตัวสมองกลที่ฝังตัวอยู่ ได้แก่ เครื่องใช้ไฟฟ้าภายในบ้าน อุปกรณ์สำนักงาน อุปกรณ์สื่อสาร อุปกรณ์ควบคุม อุปกรณ์ขนส่ง เป็นต้น ซึ่งมีจำนวนนับไม่ครบถ้วน เทคโนโลยีหลักที่ประกอบเป็นเทคโนโลยีสมองกลฝังตัวนั้น ได้แก่ เทคโนโลยีคอมพิวเตอร์ เทคโนโลยีซอฟต์แวร์ เทคโนโลยีสื่อสาร เทคโนโลยีเน็ตเวิร์ก เทคโนโลยีการวัดและการควบคุม เป็นต้น คุณลักษณะพิเศษของระบบสมองกลฝังตัวคือ การมีชุดของซอฟต์แวร์คู่กับฮาร์ดแวร์ และส่วนใหญ่จะตอบสนองแบบเรียลไทม์

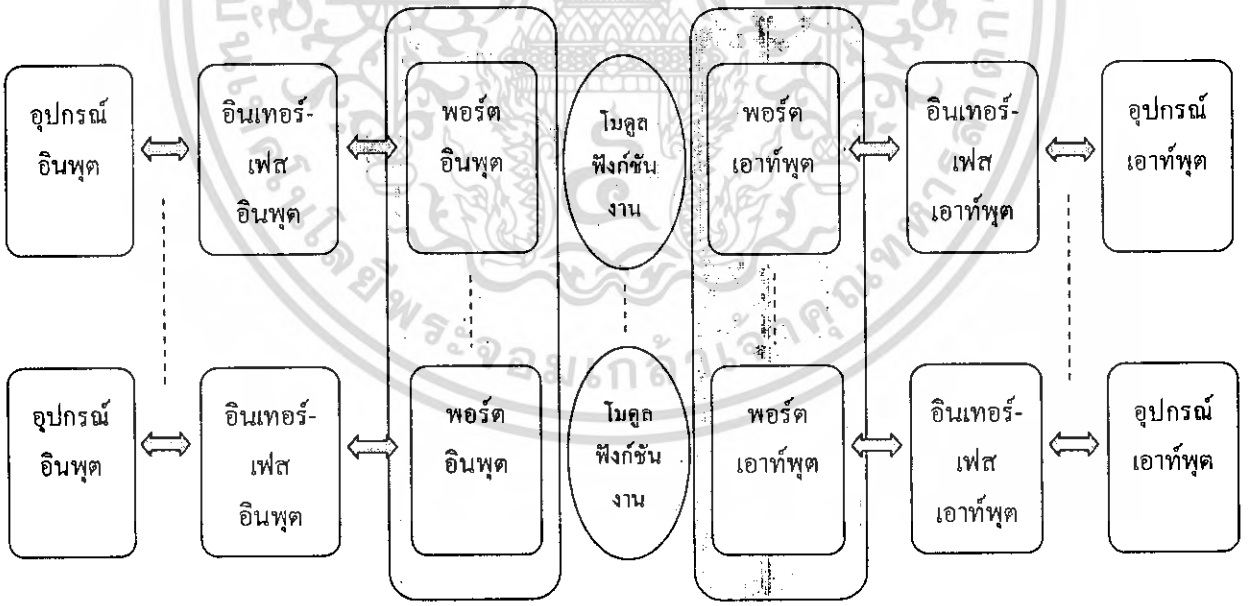
เทคโนโลยีสมองกลฝังตัวนั้นเป็นเทคโนโลยีที่ใช้องค์ประกอบดังกล่าวข้างต้นมาแปลงความสัมพันธ์เชิงกายภาพระหว่างอินพุตกับเอาท์พุต (ความสัมพันธ์โดยตรง) ให้เป็นความสัมพันธ์เชิงสารสนเทศ (ความสัมพันธ์โดยอ้อม) นอกจากนี้แม้จะมีไมโครชิปที่เป็นไมโครคอมพิวเตอร์ฝังตัวอยู่ แต่ผู้ใช้จะไม่รู้สึกถึงไมโครคอมพิวเตอร์ หรือโปรแกรมเหล่านั้น เนื่องจากฟังก์ชันที่เดิมทำโดยชิ้นส่วนวงจรหรือชิ้นส่วนประกอบนั้นจะถูกทำให้กลายเป็นซอฟต์แวร์ด้วยเทคโนโลยีสมองกลฝังตัว เพียงแต่จำนวนชิ้นส่วนน้อยลงอย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1.2 ภาพรวมส่วนประกอบฮาร์ดแวร์ของระบบสมองกลฝังตัว [1]

รูปที่ 2.1.2 แสดงส่วนประกอบด้านฮาร์ดแวร์ของระบบสมองกลฝังตัว ในกรณีที่ผู้ใช้ปลายทางใช้ระบบสมองกลฝังตัว ส่วนใหญ่จะใช้ผ่านทางอุปกรณ์อินพุตและเอาต์พุตโดยไม่จำเป็นต้องสัมผัสส่วนอื่นของระบบ



รูปที่ 2.1.3 โครงสร้างโดยสังเขปของระบบสมองกลฝังตัว [1]

ข้อมูลจากสิ่งแวดล้อมภายนอก เช่น อุณหภูมิและความดันจะถูกนำเข้าระบบผ่านทางพอร์ตอินพุตหรืออินเทอร์เฟสแบบอินพุตของอุปกรณ์อินพุตมาสู่โมดูลการทำงานเพื่อจัดเก็บหรือประมวลผล ตัวอย่างของอุปกรณ์อินพุต ได้แก่ เซ็นเซอร์ความร้อนและเซ็นเซอร์ความดัน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลเหล่านี้ส่วนใหญ่อยู่ในรูปของสัญญาณแบบแอนะล็อก แต่โดยมากโมดูลฟังก์ชันต่างๆ มักจะจัดการประมวลผลแบบดิจิทัล ดังนั้นจึงมีความจำเป็นที่จะต้องแปลงสัญญาณแอนะล็อกไปเป็นข้อมูลดิจิทัลด้วยเครื่องแปลงสัญญาณแอนะล็อก/ ดิจิทัล

นอกจากนี้โมดูลฟังก์ชันงานจะนำเอาข้อมูลที่รับเข้ามาจากสิ่งแวดล้อมภายนอก เช่น ข้อมูลความร้อนหรือความดันมาใช้ประมวลผลเพื่อเป็นสารสนเทศในการควบคุมสิ่งแวดล้อมภายนอกนั้น สารสนเทศเหล่านี้จะถูกส่งผ่านทางพอร์ตเอาต์พุต หรืออินเทอร์เฟสแบบเอาท์พุตของอุปกรณ์เอาต์พุต

การแบ่งงานระหว่างอุปกรณ์อินพุตกับอินเทอร์เฟสอินพุต และอุปกรณ์เอาต์พุตกับอินเทอร์เฟสเอาต์พุตจะเป็นไปตามข้อกำหนดเฉพาะของการออกแบบระบบสมองกลฝังตัวนั้นๆ เนื่องจากจัดการในส่วนนี้จะมีระบบที่เป็นมาตรฐานแล้วกับระบบที่ยังไม่เป็นมาตรฐาน ดังนั้นจึงต้องระมัดระวังเวลาออกแบบเพื่อให้เหมาะสมกับความต้องการ

ลักษณะพิเศษเชิงโครงสร้างเฉพาะของระบบสมองกลฝังตัว คือ การที่อุปกรณ์ที่เป็นองค์ประกอบในระบบสมองกลฝังตัว เช่น อุปกรณ์อินเทอร์เฟส หรือโมดูลฟังก์ชันงานนั้นแต่ละส่วนก็เป็นระบบสมองกลฝังตัวด้วย และระบบสมองกลฝังตัวยังสามารถเชื่อมโยงกันเป็นเน็ตเวิร์กกลายเป็นระบบสมองกลฝังตัวระดับสูงขึ้นอีกได้ การที่ระบบสมองกลฝังตัวมีทั้งที่เป็นระบบอย่างง่ายๆ จนถึงระบบที่ซับซ้อนมากนั้น ทำให้ยากที่จะนิยามว่าระบบสมองกลฝังตัวนั้นคืออะไร มีขอบเขตเท่าใด

2.1.3 ภาพรวมส่วนประกอบซอฟต์แวร์ของระบบสมองกลฝังตัว

การพัฒนาฮาร์ดแวร์ใหม่หรือปรับปรุงฮาร์ดแวร์เดิมนั้นจะเกิดจากการพัฒนาซอฟต์แวร์หรือนำซอฟต์แวร์เหล่านั้นมาประกอบกันเป็นระบบสมองกลฝังตัวขึ้น ปัญหาและความต้องการในงานดังกล่าวมีดังต่อไปนี้

1.โดยมีฟังก์ชันของระบบสมองกลฝังตัวนั้นจะขึ้นกับซอฟต์แวร์เป็นส่วนใหญ่ ดังนั้นจึงจำเป็นจะต้องให้ความสำคัญกับความถูกต้องและความชัดเจนของข้อกำหนดเฉพาะด้านความต้องการของซอฟต์แวร์ที่จะสร้าง

2. ในการกำจัดบักหรือข้อผิดพลาดที่เกิดในซอฟต์แวร์เพื่อให้สามารถใช้งานจริงได้ ถ้าเป็นไปได้เราอาจเลือกระบบที่เคยใช้ในสนามจริง และมีผลการทำงานกลับมาใช้ใหม่ ถ้าระบบนั้นมีความใกล้เคียงกับฟังก์ชันที่เราต้องการ

3. ไม่มีการรับประกันว่าฮาร์ดแวร์ที่รันซอฟต์แวร์หรือควบคุมด้วยซอฟต์แวร์นั้นจะทำงานถูกต้อง 100%

4. ขณะที่กำลังเขียนซอฟต์แวร์ ฮาร์ดแวร์ที่รันซอฟต์แวร์หรือถูกควบคุมด้วยซอฟต์แวร์นั้นอาจยังไม่ถูกสร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เนื่องจากเวลาในการพัฒนานั้นค่อนข้างจำกัด ดังนั้นจึงจำเป็นต้องย่นระยะเวลาในการเรียนรู้ฮาร์ดแวร์ใหม่ นอกจากนี้ยังต้องการลดขนาดของซอฟต์แวร์ที่กำลังพัฒนาด้วย

ข้อ 1 และ 2 เกี่ยวข้องกับกระบวนการพัฒนาระบบสมองกลฝังตัว ซึ่งมีการเสนอวิธีการแก้ปัญหาต่างๆ มากมาย เช่น การสร้างแบบจำลองโดยใช้เทคนิค UML ซึ่งข้อ 2 นั้นมีความสำคัญมาก เพื่อให้สามารถสร้างระบบสมองกลฝังตัวที่ตรงตามจุดประสงค์ และสามารถนำไปใช้ในสนามใช้งานจริงให้เร็วที่สุด

ส่วนข้อ 3-5 นั้นจะเกิดขึ้นพร้อมกับข้อ 2 จึงทำให้เราจำเป็นต้องหาวิธีการหรือเทคนิคในการพัฒนาที่ไม่ขึ้นกับฮาร์ดแวร์ โดยเฉพาะปัจจุบันมีแนวโน้มที่ชิ้นส่วนที่จะฝังตัวเข้าสู่ระบบสมองกลฝังตัวนั้นมีความเร็วสูงขึ้นและมีการทำให้ระบบอยู่บนชิป หรือแผ่นวงจรมีความหนาแน่นของวงจรสูงขึ้นมาก จึงยิ่งทำให้มีความต้องการระบบที่สนับสนุนการจำลองหรือซิมมูเลชัน เพื่อจะได้นำเอาซอฟต์แวร์ที่เคยพัฒนาแล้วมาใช้ใหม่ และตอบสนองต่อความต้องการข้อ 3 และ 4 ได้

สำหรับข้อ 5 นั้นเราสามารถแก้ไขได้โดยการใช้ระบบปฏิบัติการหรือโอเอส (Operating System) เข้าช่วย โดยให้โปรแกรมหรือซอฟต์แวร์ประยุกต์ทำงานบนระบบปฏิบัติการอีกที ดังนั้นถ้าเราเตรียมระบบปฏิบัติการสำหรับฮาร์ดแวร์แต่ละตัวแล้ว ผู้พัฒนาซอฟต์แวร์ประยุกต์ก็ไม่จำเป็นต้องรู้เกี่ยวกับเรื่องการอินเทอร์รัพต์ หรือการจัดการพื้นที่ในหน่วยความจำ ซึ่งเป็นความรู้ในระดับฮาร์ดแวร์และไม่จำเป็นต้องรู้ลำดับการควบคุมฮาร์ดแวร์นั้นๆ ด้วย นอกจากนี้ถ้าใช้ระบบปฏิบัติการแบบเรียลไทม์ แล้วเราจะสามารถพัฒนาซอฟต์แวร์ได้ง่าย ไม่ว่าจะเป็นระบบที่ต้องจัดการกับจังหวะหรือไทมิงในการทำงาน หรือการซิงโครไนเซชัน (synchronization) หรือการจัดการที่ขึ้นกับลักษณะพิเศษของฮาร์ดแวร์นั้นๆ การพัฒนาซอฟท์แวร์ของระบบสมองกลฝังตัวส่วนใหญ่จะมีความต้องการหลากหลาย แต่ระยะเวลาในการพัฒนามักจะถูกจำกัด ดังนั้นระบบปฏิบัติการจึงเป็นสิ่งจำเป็นมาก สิ่งสำคัญในการเลือกใช้ระบบปฏิบัติการในระบบสมองกลฝังตัวนั้นคือ ต้องคำนึงถึงความแพร่หลายของซอฟต์แวร์ประยุกต์ ความพร้อมของดีไวซ์ไดร์เวอร์ (device driver) ที่ควบคุมดูแลฮาร์ดแวร์ และสิ่งแวดล้อมเครื่องมืออุปกรณ์ต่างๆ ที่ใช้ในการพัฒนาซอฟต์แวร์บนระบบปฏิบัติการนั้น

2.2 เทคโนโลยีทางซอฟต์แวร์

เรามักจะได้ยินบ่อยๆ ว่า การพัฒนาซอฟต์แวร์สมองกลฝังตัวนั้นทำได้ยากเหตุผลหนึ่งก็คือการพัฒนาซอฟต์แวร์สมองกลฝังตัวส่วนใหญ่เราไม่สามารถใช้ประโยชน์จากเครื่องมือทันสมัยที่ใช้ในการพัฒนาซอฟแวร์ใช้งานทั่วไปได้ แต่ผู้พัฒนาจำเป็นต้องมีความรู้ขั้นสูงในเรื่องสถาปัตยกรรมคอมพิวเตอร์ ซึ่งบ่อยครั้งที่จำเป็นด้งมีความรู้ในเรื่องฮาร์ดแวร์ด้วย

สาเหตุที่สำคัญมากที่สุดอีกอันหนึ่งก็คือ ในการสร้างอุปกรณ์สมองกลฝังตัวหรือเรียกว่าระบบสมองกลฝังตัวนั้น จำเป็นต้องมีการพัฒนาและออกแบบที่ค่อนข้างเฉพาะ เพื่อทำให้ฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และซอฟต์แวร์เป็นอันหนึ่งอันเดียวกัน ดังนั้นจึงไม่เหมาะกับแพลตฟอร์มฮาร์ดแวร์และซอฟต์แวร์เป็นอันหนึ่งอันเดียวกัน ดังนั้นจังไม่เหมาะสมกับอิสระจากลักษณะเฉพาะหรือข้อกำหนดใดๆ ของระบบสมองกลฝังตัวแต่ละประเภทนอกจากนี้ในปัจจุบันยังมีหนังสือเอกสารไม่มากนักที่อธิบายและให้ความรู้เกี่ยวกับการพัฒนาซอฟต์แวร์สมองกลฝังตัวอย่างเป็นระบบ

2.2.1 ลักษณะเฉพาะของซอฟต์แวร์สมองกลฝังตัว (Characteristics of Embedded Software)

จุดประสงค์ของซอฟต์แวร์สมองกลฝังตัวก็คือ การเพิ่มขีดความสามารถหรือฟังก์ชันการทำงานของอุปกรณ์สมองกลฝังตัว(หรืออุปกรณ์ที่ใส่เข้าไปในระบบที่ใช้เทคโนโลยีสมองกลฝังตัว) โดยปรกติอุปกรณ์สมองกลฝังตัวจะมีหลากหลายประเภทและรูปแบบ ตั้งแต่อุปกรณ์ในดาวเทียม เครื่องบิน หรือผลิตภัณฑ์ของผู้บริโภค เช่น กล้องดิจิตอล ระบบนำทางในรถยนต์ เป็นต้น ไปถึงการควบคุมเครื่องใช้ไฟฟ้าต่างๆ เช่น หม้อหุงข้าว ตู้เย็น เป็นต้น สิ่งเหล่านี้จะมีอุปกรณ์สมองกลฝังตัวอยู่ ในปัจจุบันมีเครื่องมือเครื่องใช้ที่ใช้ระบบสมองกลตัวนั้นมีอยู่นับไม่ถ้วน

ยกเว้นอุปกรณ์สมองกลฝังตัวที่มีฟังก์ชันที่น้อยมากแล้วอุปกรณ์สมองกลฝังตัวส่วนใหญ่จะเป็นระบบตอบสนองทันที หรือที่รู้จักกันในชื่อของ ระบบเรียลไทม์ (Real-time system) ซึ่งระบบจะรับสัญญาณจากภายนอกและจะตอบสนองภายในเวลาที่กำหนด ระบบสมองกลฝังตัวส่วนใหญ่จะปืนระบบเรียลไทม์ ซึ่งกลายเป็นลักษณะพิเศษอย่างหนึ่งของซอฟต์แวร์สมองกลฝังตัว

โดยทั่วไปอุปกรณ์ที่ต่างกันมักต้องมีองค์ประกอบด้านฮาร์ดแวร์ของระบบสมองกลฝังตัวที่ต่างกันไปเพราะอุปกรณ์เหล่านั้นจะมีรูปแบบและฟังก์ชันการใช้งานที่แตกต่างกันไป ดังนั้นซอฟแวร์สมองกลฝังตัวจึงถูกสร้างขึ้นบนพื้นฐานของฮาร์ดแวร์แพลตฟอร์มที่หลากหลายแตกต่างกันไปตามระบบ

แต่ถ้าต้องเตรียมสิ่งแวดล้อมเพื่อการพัฒนาซอฟต์แวร์ที่แตกต่างกัน รวมถึงฮาร์ดแวร์แพลตฟอร์มที่ต่างกันก็จะเสียเวลาและเสียค่าใช้จ่ายมาก เพื่อการแก้ไขปัญหานี้ การพัฒนาซอฟต์แวร์สำหรับระบบสมองกลฝังตัวจึงมักจะไม่ได้พัฒนาบนฮาร์ดแวร์แพลตฟอร์มที่ซอฟแวร์นั้นจะไปทำงานอยู่ แต่จะพัฒนาบนอาร์ดแวร์แพลตฟอร์มทเป็นกลาง ซึ่งสามารถใช้งานร่วมกันได้หลายๆ งาน สิ่งแวดล้อมเพื่อใช้ในการพัฒนาซอฟต์แวร์แบบนี้เรียกว่าสิ่งแวดล้อมการพัฒนาแบบไขว้ (Cross development environment) การใช้สิ่งแวดล้อมการพัฒนาแบบไขว้นี้ถือเป็นลักษณะเฉพาะของการพัฒนาซอฟต์แวร์สมองกลฝังตัว

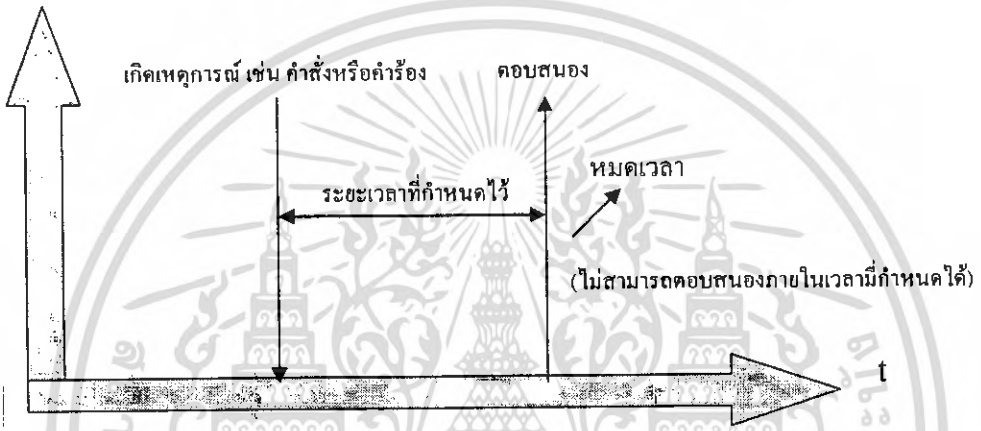
2.2.2 การประมวลผลแบบเรียลไทม์ (Real-time processing)

โดยทั่วไประบบสมองกลฝังตัวจะรับคำร้องขอประมวลผลต่างๆ จากภายนอก ตัวอย่างของคำร้องขอประมวลผลได้แก่ การเริ่มทำงานของอุปกรณ์ตามค่าที่อ่านได้จากเซ็นเซอร์ (Sensor) การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงผลบนหน้าจอตามคำสั่งประมวลผลของผู้บังคับควบคุมเครื่องมือ การตอบสนองต่อข้อมูลที่กำลังสื่อสาร คำร้องขอประมวลผลต่างๆ ซึ่งส่วนใหญ่ต้องการความเป็นเรียลไทม์ หมายความว่าการประมวลผลต้องสำเร็จเสร็จสิ้นภายในเวลาที่กำหนด โดยต้องตอบสนองอะไรบางอย่าง ดังนั้นฟังก์ชันในการสื่อสารระหว่างอุปกรณ์เหล่านี้จำเป็นต้องตอบสนองภายในเวลาที่จำกัด

การได้รับคำสั่งต่างๆ จากภายนอกโดยต้องการการตอบสนองจากระบบ เราเรียกสภาวะการตอบสนองว่า เหตุการณ์ (Event) ในระบบเรียลไทม์ เมื่อเกิดเหตุการณ์อะไรบางอย่างขึ้นมาจะต้องตอบสนองต่อเหตุการณ์นั้นภายในเวลาที่กำหนด ดังรูปที่ 2.2.1



รูปที่ 2.2.1 การร้องขอให้ตอบสนองภายในเวลาที่กำหนด [1]

ระบบเรียลไทม์สามารถแบ่งออกเป็น 2 ประเภท คือ 1 ประเภทที่มีข้อจำกัดด้านเวลาแบบหลวมๆ เช่น การควบคุมเครื่องมืออุปกรณ์มนุษย์ และ 2 ประเภทที่มีข้อจำกัดด้านเวลาที่เข้มงวดมาก เช่น การสื่อสารระหว่างอุปกรณ์ โดยแบบแรกจะเรียกว่า ระบบเรียลไทม์แบบอ่อน (soft real time system) และแบบหลังจะเรียกว่า ระบบเรียลไทม์แบบแข็ง (hard real-time system)

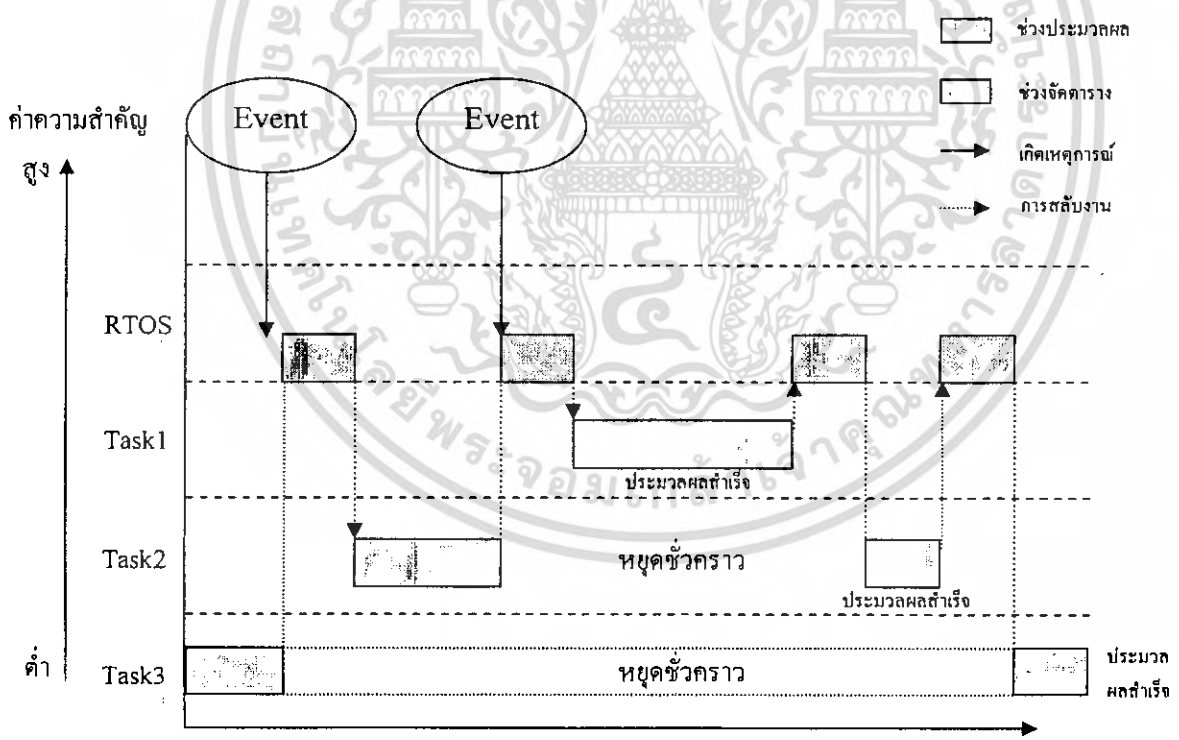
ในกรณีที่เกิดเหตุการณ์พร้อมๆ กัน ระบบจะต้องสามารถตอบสนองต่อเหตุการณ์จากเหตุการณ์ภายในเวลาที่กำหนด แต่โดยปรกติแล้วหน่วยประมวลผลไมโครคอนโทรลเลอร์จะมีเพียงชิ้นเดียว จึงทำให้ไม่สามารถตอบสนองต่อเหตุการณ์ที่เข้ามา เพื่อควบคุมให้เหตุการณ์ไหนถูกประมวลผลก่อน เหตุการณ์ไหนถูกประมวลผลหลัง โพรเซสการควบคุมแบบนี้เรียกว่า การจัดการตารางเวลางาน · (Task scheduling) ตัวชี้วัดว่าโพรเซสไหนทำก่อน โพรเซสไหนทำหลังนั้นเรียกว่า ค่าความสำคัญหรือไพร์ออริตี (priority) เหตุการณ์ที่มีไพร์ออริตีสูงนั้นจะถูกประมวลผลก่อนการให้ค่าความสำคัญหรือไพร์ออริตีเพื่อจัดลำดับให้ประมวลผลก่อนหรือหลังนี้ อาจทำให้เกิดเหตุการณ์ที่มีไพร์ออริตีต่ำนั้นไม่ถูกตอบสนองในเวลาที่กำหนด สถานการณ์เช่นนี้เรียกว่า หมดเวลา หรือไทม์เอาต์ (time out) หรือเลยเวลา หรือโอเวอร์ไทม์ (over time)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจุดประสงค์หลักของระบบเรียลไทม์ก็คือ การตอบสนองเหตุการณ์หลายๆ เหตุการณ์ภายในเวลาที่กำหนด โดยสิ่งที่ให้กรอบการจัดตารางเวลางาน (Scheduling framework) เพื่อทำให้เกิดระบบเรียลไทม์ก็คือ ระบบปฏิบัติการเรียลไทม์ หรือ เรียลไทม์โอเอส (real-time OS) เรียกย่อๆ ว่า RTOS

โดยระบบปฏิบัติการเรียลไทม์จะให้ค่าความสำคัญหรือไพร์ออริตีแก่การประมวลผลเหตุการณ์ (งานหรือแทสค์) ด้วยวิธีนี้งานที่มีความสำคัญน้อยจะถูกหยุดไว้ก่อน และระบบจะไปทำงานที่มีค่าความสำคัญมากกว่าทำให้ระบบสามารถตอบสนองต่อเหตุการณ์เร่งด่วนได้ และจะเป็นการลดความเสี่ยงของการตอบสนองไม่ทันเวลาที่กำหนดได้

การจัดตารางเวลาทำงานนี้เรียกว่า การพรีเอมชันตามเหตุการณ์ (event-driven preemption) ดังแสดงในรูปที่ 2.2.2 (ข) ค่าความสำคัญของงานนั้นจำเป็นต้องเรียงตามความเร่งด่วนของงาน ซึ่งเราจะไม่ยอมให้เคอร์เนลมาทำการเปลี่ยนลำดับค่าความสำคัญตามใจชอบได้ รูปแบบที่เราไม่ยอมให้เคอร์เนลมาแก้ไขค่าความสำคัญได้นั้น เรียกว่า วิธีจัดการแบบค่าความสำคัญตายตัว (fixed priority scheduling method)



รูปที่ 2.2.2 วิธีการพรีเอมชันตามเหตุการณ์ [1]

ข้อควรระวังอันหนึ่งเกี่ยวกับระบบปฏิบัติการเรียลไทม์ก็คือการกำหนดกรอบการทำงานของระบบปฏิบัติการเรียลไทม์ ซึ่งไม่ได้หมายความว่า กรอบการทำงานนี้จะรับประกันการ

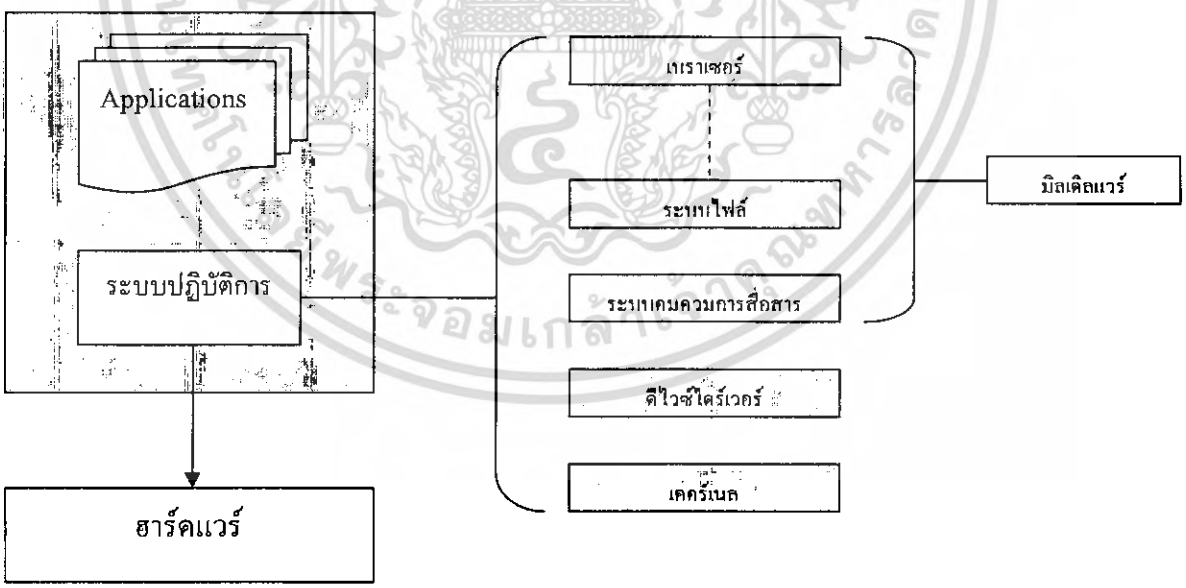
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอบสนองภายในเวลาที่กำหนดความเร็วของการประมวลผลก็จะแตกต่างกันไปตาม เนื้อหาการประมวลผลและความสามารถของฮาร์ดแวร์ เช่น MPU เป็นต้น การทำให้การทำงานของระบบเป็นเรียลไทม์จริงๆ จำเป็นต้องพึ่งความสามารถของฮาร์ดแวร์ที่จะนำมาประกอบการออกแบบ และสร้างระบบเรียลไทม์ภายใต้กรอบการทำงานของระบบปฏิบัติการเรียลไทม์

ตัวอย่างเช่น การใช้งานกรอบการทำงานของระบบปฏิบัติการเรียลไทม์นั้น เมื่อเลยเวลาที่กำหนดจะเกิดเหตุการณ์ที่เรียกว่า เลยเวลาหรือไทม์โอเวอร์ และ หมดเวลาหรือไทม์เอาต์ และการจัดการเมื่อเกิดเหตุการณ์เหล่านี้ หรือการป้องกันไม่ให้เกิดไทม์โอเวอร์หรือไทม์เอาต์นั้นเป็นหน้าที่ของผู้พัฒนาระบบที่ใช้ระบบปฏิบัติการเรียลไทม์

2.2.3 ส่วนประกอบของซอฟต์แวร์ (Software component)

รูปที่ 2.2.3 แสดงส่วนประกอบโดยสังเขปของซอฟต์แวร์สอมงกลฝังตัว โดยซอฟต์แวร์สมองกลฝังตัวก็เหมือนกับระบบทั่วไปที่สามารถแบ่งออกได้ 2 ประเภทใหญ่ๆ คือระบบปฏิบัติการและโปรแกรมประยุกต์ โดยระบบปฏิบัติการนั้นจะเป็นซอฟต์แวร์ที่ใช้ร่วมกันในระบบต่างๆ และมีหน้าที่ให้กรอบการทำงานแก่ซอฟต์แวร์อื่นๆ ในขณะที่มันทำงานอยู่ ส่วนโปรแกรมประยุกต์หรือแอพพลิเคชันเป็นโปรแกรมที่ใช้กรอบการทำงานที่ระบบปฏิบัติการเสนอให้ เพื่อให้ประมวลผลแก้ปัญหาใดปัญหาหนึ่ง



รูปที่ 2.2.3 แผนภาพส่วนประกอบของซอฟต์แวร์ [1]

ระบบปฏิบัติการประกอบด้วยเคอร์เนล และมิดเดิลแวร์ต่างๆ ดังแสดงในรูปที่ 2.2.3 เคอร์เนลจะทำหน้าที่ควบคุม ดูแล และรันแอพพลิเคชันต่างๆ อย่างที่ได้อธิบายไปแล้วว่า เคอร์เนลของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสมองกลฝังตัวนั้นมีหน้าที่ให้กรอบการทำงานในการสร้างระบบเรียลไทม์จึงมักถูกเรียกว่า เรียลไทม์เคอร์เนล (real-time kernel) ด้วย และการปฏิบัติการที่ใช้เคอร์เนลชนิดนี้คือ ระบบปฏิบัติการเรียลไทม์

ส่วนดีไวซ์ไดร์เวอร์นั้นจะใช้เคอร์เนลเพื่อควบคุมอุปกรณ์รับเข้าและส่งออกข้อมูล เพื่อทำให้เกิดฟังชันในการนำเข้าและส่งออกข้อมูลแก่แอพพลิเคชัน ดีไวซ์ไดร์เวอร์หรือฟังชันบางส่วน บางครั้งเราจะเรียกว่า ระบบควบคุมอินพุตเอาต์พุตหรือระบบอินพุตเอาต์พุตพื้นฐานเป็นต้นหลังจากเขียนหรือสร้างระบบควบคุมอินพุตเอาต์พุตแล้ว ถ้าไม่มีการเปลี่ยนแปลงฮาร์ดแวร์ ก็สามารถใช้งานระบบควบคุมอินพุตเอาต์พุตนี้กับแอพพลิเคชันต่างๆ ได้

ระบบปฏิบัติการของระบบทั่วไปนั้น ส่วนใหญ่ดีไวซ์ไดร์เวอร์และเคอร์เนลเพื่อทำให้แอพพลิเคชันต่างๆ สามารถใช้ฟังก์ชันได้สะดวกสบายมากขึ้นเรียกว่า มิดเดิลแวร์ (Middleware) ฟังก์ชันที่มิดเดิลแวร์มีให้นั้นนอกจากฟังก์ชันทั่วไป เช่น ระบบไฟล์ (file system) อินเทอร์เฟสผู้ใช้กราฟิก ฟังชันสื่อสารเป็นต้น ยังรวมไปถึงฟังก์ชันเอ็มเพ็กMPAG ซึ่งทำการอัดบีบหรือขยายคืนภาพเคลื่อนไหวหรือฟังก์ชันเจเพ็ก JPAG ซึ่งทำหน้าที่ในการอัดบีบหรือขยายคืนภาพนิ่ง หรือโปรแกรมบราวเซอร์ที่ใช้แสดงเอกสาร HTML

สำหรับระบบสมองกลฝังตัว เคอร์เนลดีไวซ์ไดร์เวอร์และมิดเดิลแวร์ จะถูกขายเป็นแพ็กเกจแยกตามฟังก์ชันของมันหรือเรียกว่า ชิ้นส่วนซอฟต์แวร์ (Software component) ในการพัฒนาระบบสมองกลฝังตัวนั้น ส่วนใหญ่ผู้ใช้แต่ละคนจะทำการประกอบชิ้นส่วนซอฟต์แวร์ต่างๆ เหล่านี้เพื่อสร้างเป็นระบบปฏิบัติการเฉพาะสำหรับผู้ใช้คนนั้น แน่นอนว่ามีบ่อยครั้งที่ผู้ใช้นั้นต้องสร้างหรือเขียนชิ้นส่วนซอฟต์แวร์บางอย่างขึ้นมาเอง

2.3 เทคโนโลยีฮาร์ดแวร์

ปัจจุบันผลิตภัณฑ์ที่เป็นระบบสมองกลฝังตัวนับวันจะมีอายุการใช้งานที่สั้นลง เนื่องจากมีของใหม่มาแทนที่ ดังนั้นการย่นระยะเวลาในการพัฒนาผลิตภัณฑ์ใหม่ได้จึงเป็นสิ่งสำคัญ จะทำเช่นนั้นได้ ผู้พัฒนาระบบสมองกลฝังตัวจะต้องมีความเร็วในส่วนต่างๆ ของฮาร์ดแวร์ที่เกี่ยวข้อง เช่น ข้อกำหนดและโครงสร้างของระบบฮาร์ดแวร์ที่จะใช้ในระบบสมองกลฝังตัว โดยสามารถกำหนดรายละเอียดได้อย่างถูกต้องแม่นยำ นอกจากนี้ผู้ที่เกี่ยวข้องกับการสั่งซื้อฮาร์ดแวร์จะต้องสามารถบอกข้อกำหนดของสิ่งที่ตนเองต้องการได้อย่างถูกต้องด้วย

2.3.1 สถาปัตยกรรมของระบบสมองกลฝังตัว

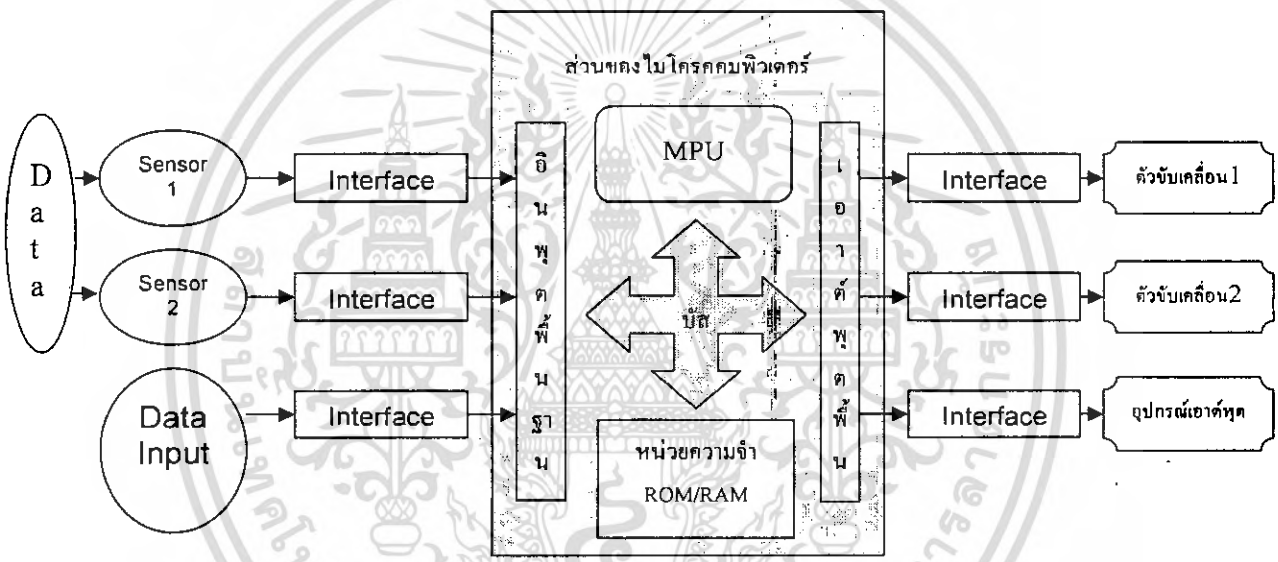
ระบบสมองกลฝังตัวเป็นระบบคอมพิวเตอร์ที่ออกแบบมาเพื่อควบคุมการทำงานของอุปกรณ์หรือเครื่องจักรต่างๆ โดยจะต้องเชื่อมโยงหรือฝังตัวอยู่กับอุปกรณ์เหล่านั้น ซึ่งจะต้องมีการกำหนดคุณลักษณะทางด้านฮาร์ดแวร์และซอฟแวร์อย่างเหมาะสม ดังนั้น สำหรับอุปกรณ์หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องจักรที่แตกต่างกัน ก็จะทำให้ข้อกำหนดคุณลักษณะของระบบสมองกลฝังตัวนั้นแตกต่างกันออกไปด้วย ในส่วนของฮาร์แวร์ ผู้ออกแบบจะต้องวางโครงสร้างให้รองรับโครงสร้างของซอฟแวร์ได้ และควรจะมองเผื่อไปถึงการขยายขีดความสามารถของระบบต่อไปในอนาคตด้วย โดยควรจะวางโครงสร้างที่ง่ายต่อการขยายระบบในอนาคต

สถาปัตยกรรมพื้นฐาน (Basic architecture)

ความต้องการของสถาปัตยกรรมของระบบสมองกลฝังตัว (โครงสร้างของฮาร์ดแวร์) โดยปรกติจะแตกต่างกันไปตามลักษณะของการประยุกต์ใช้ ซึ่งสามารถสรุปโครงสร้างพื้นฐานของฮาร์ดแวร์ได้ดังรูปที่ 2.3.1



รูปที่ 2.3.1 สถาปัตยกรรมพื้นฐานของระบบสมองกลฝังตัว [1]

ส่วนแรกและเป็นหัวใจสำคัญของระบบ คือ ไมโครคอมพิวเตอร์ (microcomputer) เป็นส่วนที่มีหน่วยประมวลผลและหน่วยความจำซึ่งเป็นส่วนสำคัญของระบบ ประกอบด้วย MPU หน่วยความจำรอมและแรม (ROM,RAM) อินพุตเอาต์พุตพื้นฐาน (basic I/O) ส่วนของ MPU เป็นส่วนที่จะกำหนดความเร็วในการประมวลผลและกำหนดความสามารถของระบบ ส่วน ROM เป็นหน่วยความจำสำหรับเก็บข้อมูลที่ไม่มีความจำเป็นต้องมีการเปลี่ยนแปลง เช่น โค้ดโปรแกรม (Program code) และข้อมูลตัวแปรแบบกำหนดตายตัว ส่วน RAM เป็นหน่วยความจำสำหรับ MPU ให้ทำงานตามโปรแกรม เช่น สแตก เป็นต้น โดยจะทำการเก็บหรือเขียนทับค่าตัวแปรที่ใช้งานระหว่างการทำงาน (หรือเป็นบัฟเฟอร์สำหรับตัวแปรหรือข้อมูลต่างๆ) โดยจะต้องมีขนาดไม่เล็กไปกว่าขนาดที่จำเป็นสำหรับการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนอินพุตพื้นฐาน (รวมถึงการเชื่อมต่อเซ็นเซอร์ และอุปกรณ์อินพุตต่างๆ)เป็นส่วนรับข้อมูลและสัญญาณต่างๆ จากไมโครคอมพิวเตอร์ โดยจะส่งข้อมูลเหล่านี้ในรูปของข้อมูลดิจิตอลให้กับ MPU เพื่อนำไปประมวลผลต่อ โดยทั่วไปเรียกว่าอินพุตพอร์ต โดยทั่วไประยะที่ MPU ทำการอ่านข้อมูลจากอินพุตพอร์ตจะสั้นมาก ในกรณีที่มีสัญญาณต่างๆ ที่จะต้องผ่านอินพุตพอร์ตจำนวนมาก จะต้องมีสวิตซ์ระหว่างการอ่านอินพุตพอร์ตแต่ละช่องสัญญาณ ซึ่งการปรับจังหวะให้เข้ากับจังหวะการทำงานของ MPU รวมทั้งการสวิตซ์ช่องสัญญาณต่างๆ จะเป็นคุณสมบัติพื้นฐานของอินพุตพอร์ต

สำหรับเซ็นเซอร์เป็นอุปกรณ์ที่แปลงการเปลี่ยนแปลงเชิงกายภาพให้เป็นการเปลี่ยนแปลงเชิงสัญญาณไฟฟ้าอย่างไรก็ตาม ข้อมูลที่ส่งออกมาจากเซ็นเซอร์อาจเป็นค่าการเปลี่ยนแปลงของความต้านทาน หรือเป็นการเปลี่ยนแปลงของแรงดันไฟฟ้าในระดับอ่อนๆ ซึ่งอินพุตพื้นฐานของไมโครคอมพิวเตอร์ไม่สามารถที่จะรับข้อมูลเหล่านี้ได้ จึงจำเป็นต้องมีอุปกรณ์ที่ทำหน้าที่แปลงสัญญาณจากรูปแบบสัญญาณออกของเซ็นเซอร์ให้เป็นส่วนสัญญาณที่ส่วนอินพุตของไมโครคอมพิวเตอร์สามารถรับได้ วงจรที่ทำหน้าที่นี้เรียกว่า วงจรอินเทอร์เฟส (Interfacing circuit) ซึ่งมีการพัฒนาขึ้นใช้งานแพร่หลาย

ส่วนของเอาร์พุตพื้นฐาน (รวมถึงการเชื่อมต่อตัวขับเคลื่อนและอุปกรณ์เอาต์พุตต่างๆ)เป็นช่องทางออกของสัญญาณที่เกิดจากผลการคำนวณของ MPU โดยมีการเปลี่ยนแปลงข้อมูลแบบดิจิตอลที่ได้นี้ ให้เป็นสัญญาณกายภาพที่เหมาะสม เรียกว่า เอาต์พุตพอร์ต ในกรณีนี้ก็เช่นกัน ระยะเวลาที่ MPU ส่งข้อมูลออกไปยังเอาต์พุตนั้นเป็นช่วงเวลาสั้นๆ โดยอุปกรณ์เอาต์พุตจะต้องทำหน้าที่ในการเก็บข้อมูลนั้นเพื่อแปลงเป็นสัญญาณออกที่ต่อเนื่องและมีเสถียรภาพ นอกจากนี้ MPU เองยังต้องทำงานติดต่อกับเอาต์พุตพอร์ตหลายช่องสัญญาณ ดังนั้นส่วนของเอาต์พุตพื้นฐานจึงต้องมีคุณสมบัติในการปรับจังหวะเวลาให้เหมาะสมกับการทำงานของ MPU และการสวิตซ์ระหว่างเอาต์พุตพอร์ตหลายๆ ช่องสัญญาณเช่นเดียวกันกับในกรณีของอินพุตพื้นฐาน

ตัวขับเคลื่อนเป็นส่วนที่ทำหน้าที่แปลงการเปลี่ยนแปลงของสัญญาณไฟฟ้าให้เป็นการทำงานด้านกายภาพซึ่งเป็นส่วนสำคัญในการเชื่อมผลการคำนวณของ MPU กับโลกภายนอก ในการเชื่อมโยงสัญญาณออกของเอาต์พุตพอร์ตกับตัวขับเคลื่อน ให้สามารถทำงานได้ตามที่ต้องการนั้นจำเป็นต้องมีวงจรเชื่อมต่อ ซึ่งในกรณีนี้วงจรเชื่อมต่อจะทำหน้าที่แปลงสัญญาณดิจิตทัลที่ออกจากเอาต์พุตพอร์ตให้เป็นกำลังไฟฟ้าสำหรับจ่ายให้กับตัวขับเคลื่อน

การวางโครงสร้างของระบบสมองกลฝังตัวสามารถทำได้หลายแบบ องค์ประกอบที่สำคัญคือ การใช้ชิปไอซีโดยผู้ออกแบบอาจทำเป็นแผ่นลายวงจรพิมพ์

ในกรณีที่ต้องการไมโครคอมพิวเตอร์ที่มีความความสามารถสูง (เช่น MPU ที่มีความเร็วสูงและฟังก์ชั่นการทำงานที่ซับซ้อนจะต้องการหน่วยความจำขนาดใหญ่ หรือต้องการอินพุตเอาต์พุตแบบพิเศษ) ก็อาจใช้โครงสร้างในผลิตภัณฑ์ที่ผลิตจำนวนมากๆ ก็อาจเลือกใช้

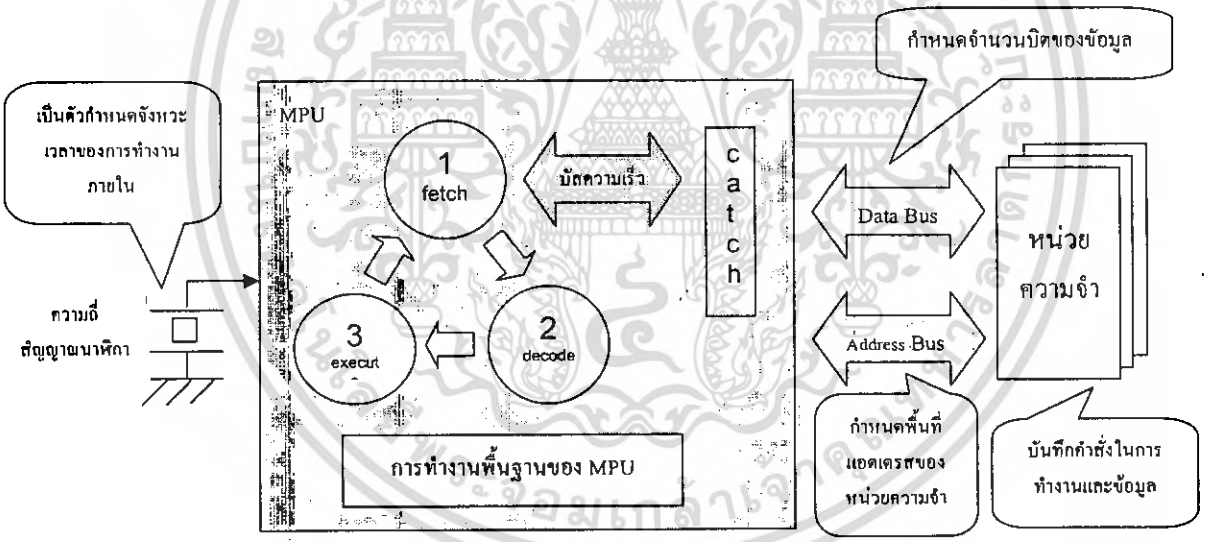
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอมพิวเตอร์แบบชิปเดี่ยว ซึ่งในกรณีนี้เราสามารถเลือกไมโครคอมพิวเตอร์จากรุ่นต่างๆ ที่มีอยู่เพื่อให้เหมาะสมกับการใช้งานของเราได้ โดยสามารถเลือกได้ตั้งแต่ประเภทของ MPU ที่อยู่ภายในขนาดของหน่วยความจำ และประเภทของอินพุตเอาต์พุตพื้นฐานที่อยู่ภายในชิป เพื่อให้เหมาะกับลักษณะงานได้

โครงสร้างของระบบในรูปแบบต่างๆ ที่กล่าวมานั้น การที่จะเลือกเอารูปแบบใดรูปแบบหนึ่งนั้น นอกจากจะพิจารณาจากความสามารถในการประมวลผลของระบบสมองกลฝังตัวที่จะพัฒนาแล้ว ยังต้องคำนึงของระยะเวลาในการพัฒนา ต้นทุนการพัฒนา รวมถึงการผลิต บุคลากรที่จะเป็นผู้รับผิดชอบ และอุปกรณ์ต่างที่ต้องใช้ในการพัฒนาระบบด้วย

2.3.1.1 เอ็มพียู (MPU)

MPU ย่อมาจาก Micro processing Unit เป็นส่วนที่เป็นหัวใจสำคัญของไมโครคอมพิวเตอร์ ในการกำหนดสเปกของระบบฝังตัวนั้น การเลือก MPU เป็นปัจจัยสำคัญที่จะส่งผลให้ระบบมีความสามารถพอเพียงสำหรับการประมวลผลการใช้งาน



รูปที่ 2.3.2 การกำหนดพื้นฐานของ MPU [1]

สำหรับคุณลักษณะสมบัติของ MPU ที่จะส่งผลกับความสามารถในการประมวลผล ได้แก่ จำนวนบิตของข้อมูลบนบัสข้อมูล (ที่สามารถส่งไปได้บนบัสข้อมูลต่อคำสั่ง 1 คำสั่งของ MPU เช่น 8, 16 , หรือ 32 บิต เป็นต้น) ขนาดพื้นที่แอดเดรสของหน่วยความจำ (memory address space) ซึ่งเป็นตัวกำหนดว่า MPU สามารถเข้าถึงข้อมูลในหน่วยความจำเป็นปริมาณเท่าไร และความถี่สัญญาณนาฬิกา (เป็นมาตรฐานในการวัดความเร็วในการประมวลผล) เป็นต้น ปัจจุบัน MPU ส่วนมากมีการสังเคราห์ความถี่สัญญาณนาฬิกาภายในตัวเอง ทำให้สามารถเพิ่มความถี่ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประมวลผลให้สูงขึ้นได้มาก ผลจากการทำงานที่ความถี่สูงมากนี้จะทำให้อุณหภูมิภายในชิปเพิ่มสูงขึ้น จึงจำเป็นจะต้องมีการระบายความร้อนให้เหมาะสม

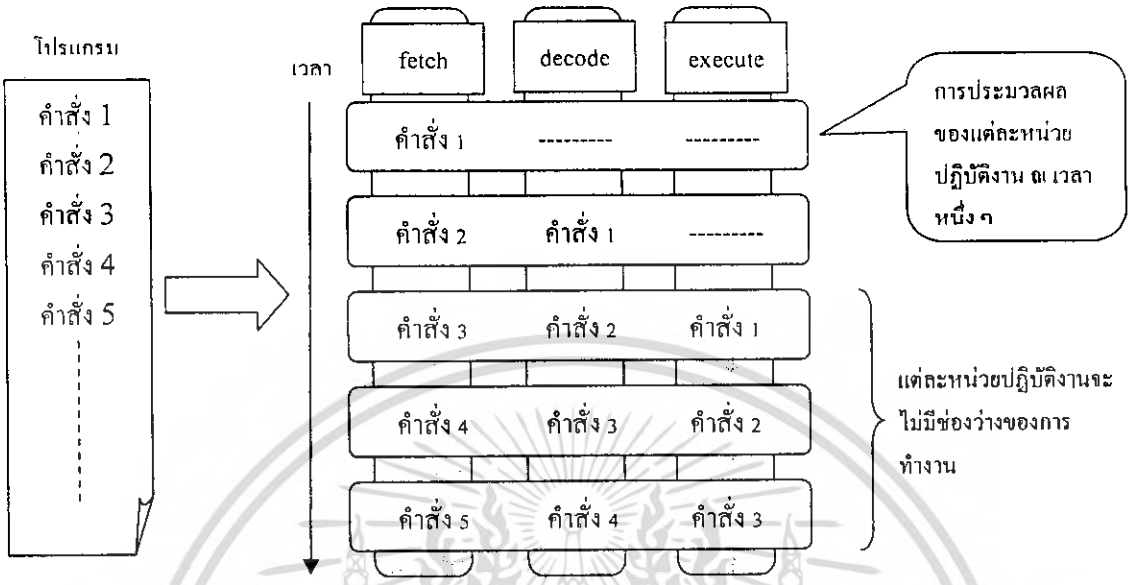
ในส่วนความกว้างบิตบัสของข้อมูล หากนำไปใช้ในการควบคุมโดยทำการ ON/OFF แต่ละบิตแต่ MPU ที่เป็น 8 บิต ก็เพียงพอแล้ว แต่หากต้องการนำไปใช้ในการประมวลผลทางคณิตศาสตร์ที่ต้องการความละเอียดสูง ก็อาจใช้ MPU ที่มีความกว้างบัสข้อมูลเป็น 32 บิต ส่วนในการนำไปใช้ในการประมวลผลภาพนั้น นอกจากจะต้องมีพื้นที่หน่วยความจำขนาดใหญ่เพื่อเป็นที่เก็บข้อมูลแล้ว ความเร็วในการประมวลผลที่สูงก็เป็นองค์ประกอบที่จำเป็นด้วย ถ้าเป็นงานควบคุมตัวแปรที่มีการเปลี่ยนแปลงช้า เช่น อุณหภูมิ รวมถึงถ้าไม่จำเป็นต้องมีการบันทึกผลมากมายนัก ก็อาจใช้ MPU ที่ช้าลงได้ จากการพิจารณาเช่นนี้ การเลือก MPU ให้เหมาะสมกับงานเพื่อใช้ในระบบสมองกลฝังตัวจะต้องสามารถตอบสนองต่อความต้องการ โดยอย่างน้อย MPU จะต้องมีสเปกเพียงพอที่จะสามารถทำงานนั้นได้ เป็นต้น

การตอบสนองต่อการต้องการความสามารถในการประมวลผลที่สูงนั้น นอกจากจะเพิ่มความถี่สัญญาณนาฬิกาแล้ว การปรับเปลี่ยนวิธีการประมวลผลภายในของ MPU ก็มีผลกับความเร็วเช่นกัน โดยปรกติแล้วการทำงานภายใน MPU จะเริ่มด้วยการอ่านคำสั่ง (fetch เป็นการอ่านคำสั่งจากหน่วยความจำเข้ามา) แล้วทำการแปลความหมาย (decode เป็นการตีคำสั่ง) แล้วจึงทำตามคำสั่ง (execute เป็นการปฏิบัติตามคำสั่ง) ขั้นตอนทั้งสามนี้จะทำซ้ำกันไปเรื่อยๆ เป็นผลให้เกิดการประมวลผลของโปรแกรมทั้งชุด

การทำงานแบบ pipeline จะมีส่วนช่วยให้การทำงานของยูนิตทั้งสามไม่ต้องมีการว่างการทำงานต่อเนื่องจากต้องรอผลของส่วนอื่น โดยทำการประมวลผลของแต่ละส่วนไปพร้อมๆ กันดังรูปที่ 3.1.4 วิธีนี้ทำให้มีความเร็วในการประมวลผลสูงขึ้น นอกจากนี้การเพิ่มความเร็วในการเคลื่อนย้ายข้อมูลระหว่าง MPU และหน่วยความจำก็สามารถช่วยให้ความเร็วในการประมวลผลสูงขึ้นเช่นกัน โดยนำหน่วยความจำที่มีความเร็วสูงที่เรียกว่า หน่วยความจำแคช วางระหว่าง MPU และหน่วยความจำหลัก เพื่อเพิ่มความเร็วในการอ่านคำสั่ง ซึ่งวิธีนี้เป็นวิธีที่ใช้กันแพร่หลาย อย่างไรก็ตาม วิธีนี้สามารถรองรับการทำงานของโปรแกรมที่มีการกระโดดหรือมีการทำงานแยกตามเงื่อนไขต่างๆ รวมทั้งมีจุดด้อยในเรื่องการประมาณเวลาในการทำงานของ MPU นั้นทำได้ยากเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.3.3 การประมวลผลแบบ pipeline [1]

นอกจากนี้ยังสามารถแบ่ง MPU ออกตามโครงสร้างภายในได้เป็น 2 ประเภทหลักๆ คือ CISC และRISC โดยโครงสร้างแบบ CISC (Complex Instruction Set Complex) เป็น MPU ที่ออกแบบมาให้มีคำสั่งในการทำงานที่ซับซ้อนจำนวนมาก เพื่อให้สามารถตอบสนองกับความต้องการในการโปรแกรมด้วยฟังก์ชันต่างๆ (เพื่อให้ใช้จำนวนคำสั่งน้อยลง) ในการทำงาน เป็นผลให้โครงสร้างภายในของ MPU มีความซับซ้อนสูง และใช้ความถี่สัญญาณนาฬิกาจำนวนหลายลูกในการปฏิบัติตามคำสั่ง นอกจากนี้ยังสามารถเข้าถึงหน่วยความจำเพื่ออ่านหรือเขียนค่าประมวลผลได้โดยตรงไม่ต้องผ่านรีจิสเตอร์

ส่วน MPU ที่มีโครงสร้างแบบ RISC (Reduced Instruction Set Computer) ถูกออกแบบมาให้มีคำสั่งทำงานจำนวนน้อย และมีเพียงคำสั่งที่จำเป็นเท่านั้น ทำให้โครงสร้างภายใน MPU ไม่ซับซ้อน และสามารถประมวลผลได้อย่างมีประสิทธิภาพมากขึ้น จากการที่คำสั่งในการทำงานเป็นคำสั่งง่ายๆ ทำให้สามารถทำงานได้อย่างวรวดเร็วเพียงแค่ไม่กี่สัญญาณนาฬิกา และสามารถใช้กับสัญญาณนาฬิกาที่ความถี่สูงมากๆ ได้ นอกจากนี้การทำงานยังถูกออกแบบมาให้ใช้รีจิสเตอร์ภายใน จึงต้องมีการถ่ายเทข้อมูลจากหน่วยความจำไปไว้ในรีจิสเตอร์ จึงจะประมวลผลได้ทำให้ทำงานได้เร็วมาก

ในการสร้างโปรแกรมให้สามารถทำงานได้อย่างที่ต้องการจากคำสั่งพื้นฐานนั้น จำเป็นจะต้องมีหน่วยประมวลผลทางภาษาที่มีสมรรถนะสูง เช่น ตัวแปลภาษาหรือคอมไพเลอร์ และการพัฒนาของคอมไพเรอร์นี้ก็จะเป็นประโยชน์ต่อการใช้ MPU แบบ RISC เป็นอย่างมาก ในช่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้นๆ กระแสนิยมส่วนใหญ่จะไปทาง MPU ที่เป็นแบบ CISC ทั้งนี้เนื่องจากสมัยก่อนเวลาที่ใช้ในการเข้าถึงหน่วยความจำที่ช้าอยู่ ทำให้มีแนวคิดที่จะลดจำนวนครั้งของการอ่านข้อมูลจากหน่วยความจำ โดยการทำให้คำสั่งในการทำงานซับซ้อนขึ้น ต่อมาภายหลังได้มีการพัฒนาหน่วยความจำให้สามารถเข้าถึงได้เร็วขึ้น โครงสร้างแบบ RISC จึงเริ่มเป็นที่นิยมกันมากขึ้น

2.3.1.2 บัส (BUS)

บัส (BUS) สามารถแบ่งตามตำแหน่งที่ใช้งานได้เป็น 2 ประเภทหลักๆ ได้แก่บัสภายใน (internal bus) และบัสภายนอก (external bus) บัสภายในเป็นกลุ่มของสายสัญญาณภายในไมโครคอนโทรลเลอร์ที่ใช้ร่วมกันสำหรับส่งผ่านข้อมูลระหว่างMPU หน่วยความจำ และอินพุตเอาต์พุตพื้นฐาน บัสภายนอกเป็นกลุ่มของสายสัญญาณเพื่อส่งผ่านข้อมูลระหว่างไมโครคอมพิวเตอร์และอุปกรณ์ภายนอกต่างๆ และมีการแยกประเภทการใช้งานตามประเภทอุปกรณ์ภายนอกและวัตถุประสงค์การใช้การ

สำหรับมาตรฐานของบัส นอกจากที่ ISO และ IEEE กำหนดไว้แล้ว ยังมีที่ถูกกำหนดโดยผู้ผลิตรายต่างๆ ที่นำมาใช้กับผลิตภัณฑ์ของตนเองจนกลายเป็นมาตรฐานในตลาด (de facto standard) มาตรฐานของบัสมีการพัฒนาการเปลี่ยนแปลงของ MPU ที่เร็วขึ้นมีการประมวลผลที่หลากหลายขึ้น หน่วยความจำมีขนาดใหญ่ขึ้น และมีการประยุกต์ใช้การที่หลากหลายขึ้น

สำหรับกลุ่มของสายสัญญาณที่เป็นบัสภายในสามารถแบ่งออกได้เป็น 3 ประเภทหลักๆ คือ แอดเดรสบัส (address bus) บัสข้อมูล (data bus) และบัสควบคุม (control bus) ในส่วนของแอดเดรสบัสจะใช้ในการระบุดีไวซ์หรืออุปกรณ์ และข้อมูลจะถูกแลกเปลี่ยนผ่านบัสข้อมูล เพื่อทำให้ให้เกิดการระบุและแลกเปลี่ยนข้อมูล ทั้งหมดนี้ถูกกำหนดผ่านบัสควบคุม

จำนวนบิตของแอดเดรสบัสจะเป็นตัวกำหนดขนาดพื้นที่แอดเดรสที่บัสนั้นสามารถอ้างอิงถึงได้ ซึ่งรวมไปถึงขนาดของโปรแกรมที่เก็บอยู่ในหน่วยความจำรอมเพื่อการประมวลผล ขนาดของแอดเดรสที่ใช้ระหว่างการทำงานในลักษณะของสแต็ก พื้นที่ที่เป็นตัวแปร และบัฟเฟอร์ข้อมูลที่อ้างถึงได้

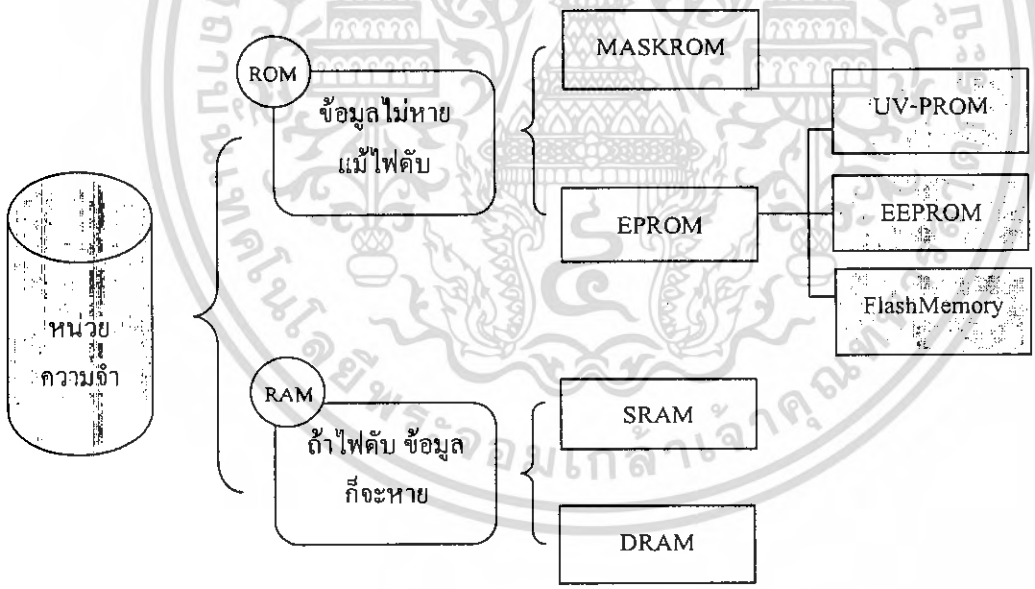
ในส่วนของบัสข้อมูล ถ้าจำนวนบิดของบัสข้อมูลมาก ก็หมายถึงข้อมูลที่สามารถส่งได้ในแต่ละครั้งมีขนาดใหญ่ขึ้น ซึ่งหากจำนวนบิตมากขึ้น ก็หมายความว่าสามารถส่งข้อมูลได้มากขึ้นและเร็วขึ้นนั่นเอง นอกจากนี้บาง MPU อาจมีจำนวนบิตที่ใช้ในการประมวลผลจริงกับขนาดของบัสข้อมูลที่แตกต่างกัน เนื่องจากข้อจำกัดเกี่ยวกับขาของชิปไอซี เช่น บาง MPU มีการประมวลผลภายในเป็นหน่วยของ 16 บิต แต่มีบัสข้อมูลขนาด 8 บิตเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.3 หน่วยความจำ (Memory)

หน่วยความจำเป็นที่สำหรับเก็บรักษาข้อมูลข่าวสาร ได้แก่ โปรแกรมและข้อมูล ที่อยู่ภายในไมโครคอมพิวเตอร์ สำหรับระบบสมองกลฝังตัวซึ่งประกอบด้วยส่วนที่เป็นฮาร์ดแวร์และส่วนที่เป็นซอฟแวร์นั้น ที่เก็บซอฟแวร์ก็คือหน่วยความจำนั่นเอง

อุปกรณ์หน่วยความจำที่ใช้มีเดียแบบแม่เหล็กไฟฟ้า มักจะเรียกว่าหน่วยความจำภายนอกบ้าง หรือหน่วยความจำสำรองบ้าง ซึ่งโดยปรกติแล้วจะใช้เป็นที่เก็บซอฟแวร์ ส่วนหน่วยความจำที่เป็นแบบสารกึ่งตัวนำนั้นจะใช้เป็นที่สำหรับรันโปรแกรม หรือเป็นที่สำหรับให้ซอฟแวร์ทำงาน ซึ่งต้องมีการอ่านโปรแกรมไปยังหน่วยความจำแบบสารกึ่งตัวนำก่อนเริ่มต้นทำงาน หน่วยความจำที่เป็นสารกึ่งตัวนำแบ่งได้เป็น 2 ประเภทใหญ่ๆ ได้แก่ หน่วยความจำที่เก็บข้อมูลแบบไม่สามารถเปลี่ยนแปลงแก้ไขได้และหน่วยความจำที่สามารถเปลี่ยนแปลงแก้ไขได้อย่างอิสระ รวมถึงการที่ข้อมูลอาจหายไปเนื่องจากไฟดับ ในการใช้งานเรามักจะใช้ ROM เพื่อจัดเก็บโค้ดโปรแกรมที่ไม่มีการเปลี่ยนแปลง และใช้ RAM เป็นหน่วยความจำที่เก็บค่าตัวแปรต่างๆ หรือสแตกที่มักมีการเปลี่ยนแปลงไปตามจังหวะการทำงานของระบบ โดยหน่วยความจำทั้งสองส่วนนี้ล้วนมีความจำเป็นสำหรับการทำงานของระบบสมองกลฝังตัวด้วยกันทั้งนั้น



รูปที่ 2.3.4 ประเภทของไอซีที่ใช้เป็นหน่วยความจำ [1]

ROM (Read Only Memory) เป็นหน่วยความจำที่ใช้สำหรับอ่านข้อมูลเมื่อมีการระบุแอดเดรสใดๆ โดยมีคุณสมบัติพิเศษคือ ข้อมูลจะไม่หายไปไหน แม้ว่าจะงดจ่ายไฟให้ก็ตาม (non-volatile) ดังนั้น ROM จึงมักใช้เป็นที่เก็บโปรแกรมเริ่มต้นที่จะถูก MPU อ่านเข้าไปเพื่อตีคำสั่งขณะเริ่มต้นการทำงาน หลังจากที่มีการจ่ายไฟเข้าระบบ ซึ่งโปรแกรมนี้จะถูกเรียกทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนเพื่อโหลดโปรแกรม (เช่น ระบบปฏิบัติการ)จากอุปกรณ์ที่เป็นหน่วยความจำภายนอก รวมถึงเป็นที่เก็บค่าตัวแปรที่ไม่มีการเปลี่ยนค่าระหว่างการทำงานด้วย โดยทั่วไปแล้ว ROM สามารถแบ่งได้เป็น 2 ประเภทหลักๆ ได้แก่ Mask ROM คือ ROM ที่ถูกเขียนค่าไว้ตั้งแต่ขั้นตอนการผลิต และอีกประเภทหนึ่งเป็น ROM ที่จะเขียนข้อมูลเข้าภายหลังหรือสามารถลบได้ เรียกว่า EPROM

EPROM (Erasable Programmable ROM) ที่นิยมใช้กันสมัยก่อน ได้แก่ UV-PROM สามารถลบข้อมูลได้ด้วยการฉายแสงอัตราไวโอเลต เข้าไปยังหน้าต่างของชิปหน่วยความจำ (ดังรูป2.3.4) ซึ่งปัจจุบันหาได้ยากขึ้น นอกจากนี้ในการโปรแกรมลงใน PROM หรือเขียนข้อมูลลงไปก็ต้องใช้อุปกรณ์ที่เรียกว่า ตัวเขียนรอม (ROM writer) ซึ่งจะมีการจ่ายแรงดันไฟฟ้าที่สูงกว่าแรงดันไฟเลี้ยงไอซี เพื่อทำการเขียนข้อมูลลงไป

RAM (Random Access Memory) เป็นหน่วยความจำที่เมื่อระบุแอดเดรสแล้วสามารถถูกเขียนหรืออ่านได้ มี 2 ประเภท ได้แก่ SRAM และ DRAM โดยที่ SRAM (static RAM) เป็นหน่วยความจำที่ใช้ทรานซิสเตอร์ 4 ตัว หรือ 6 ตัวประกอบขึ้นเป็นวงจรฟลิปฟลอป สำหรับ 1 เซลในหน่วยความจำเพื่อเก็บข้อมูล 1 บิต เนื่องจาก SRAM สามารถทำงานได้ด้วยความเร็วสูง จึงนิยมนำมาใช้เป็นหน่วยความจำแบบแคช นอกจากนี้ SRAM ยังมีข้อดีอีก คือ หากให้แรงดันประมาณ 1 โวทต์กับหน่วยความจำนี้ ถึงแม้แหล่งจ่ายไฟของระบบฝังตัวจะดับ ข้อมูลในหน่วยความจำก็ยังคงอยู่ ดังนั้นในการใช้งานบางประเภทจะมีการป้อนไฟจากแบตเตอรี่เพื่อจ่ายให้กับหน่วยความจำแบบ SRAM บางส่วน เพื่อให้สามารถเก็บข้อมูลไว้ได้ (เรียกว่า battery back up) ระบบก็จะสามารถเก็บข้อมูลระหว่างประมวลผลหรือผลการคำนวณต่างๆ ไว้ได้โดยไม่จำเป็นต้องมีอุปกรณ์หน่วยความจำภายนอกประเภทที่ทำงานด้วยแม่เหล็กไฟฟ้า อุปกรณ์หน่วยความจำภายนอกโดยมากจะมีส่วนที่เป็นจักรกลหรือส่วนที่มีการเคลื่อนที่ จึงไม่เหมาะสมกับการใช้งานในระบบสมองกลฝังตัว ซึ่งกรณีนี้การใช้หน่วยความจำแบบที่มีแบตเตอรี่เพื่อรักษาข้อมูลจะเหมาะสมมากกว่า

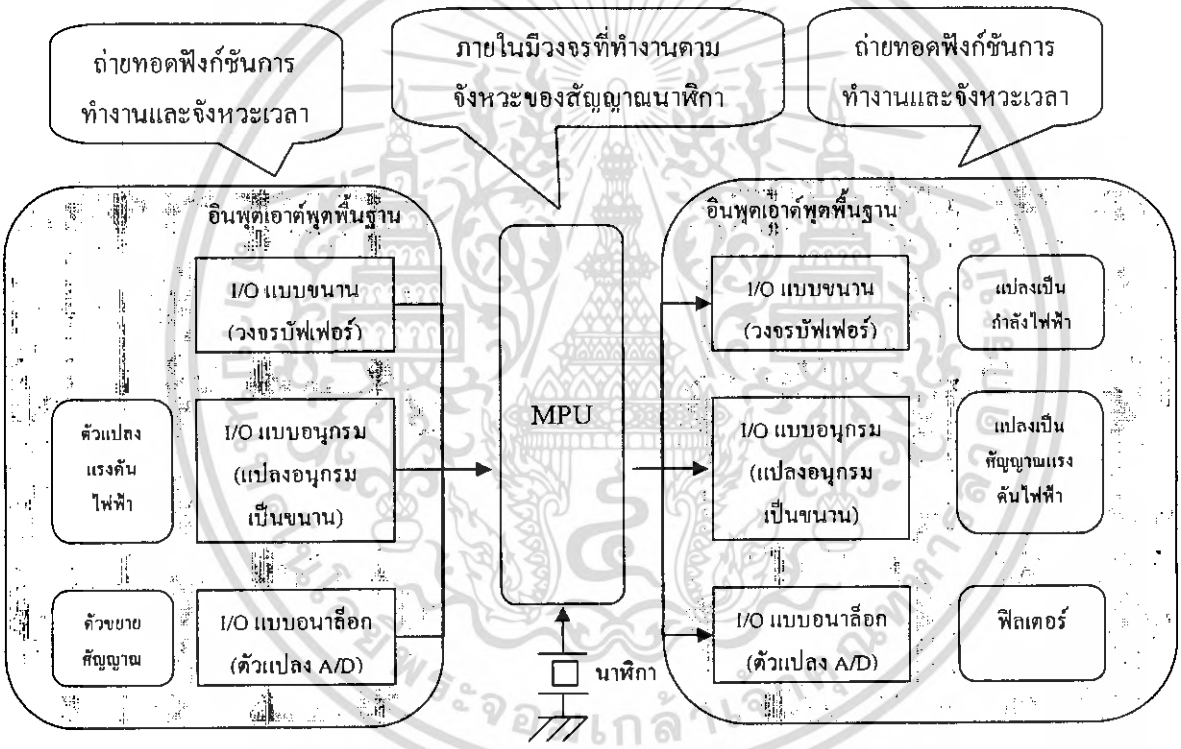
2.3.2 อินพุตเอาต์พุตพื้นฐาน (Basic I/O)

ในระบบสมองกลฝังตัวนั้น MPU จะรับข้อมูลข่าวสารจากภายนอกเข้ามา และทำการคำนวณและประมวลผลก่อนที่จะส่งผลนั้นออกไปภายนอกอีกครั้ง ซึ่งจะต้องมีการแลกเปลี่ยนข้อมูลระหว่าง MPU กับภายนอกเพื่อให้สามารถทำงานได้ตามที่กำหนด การแลกเปลี่ยนข้อมูลระหว่าง MPU กับภายนอกจำเป็นต้องมีส่วนของการเชื่อมต่ออินพุตเอาต์พุต (I/O interface) เข้ามาเกี่ยวข้องในการปรับระดับแรงดันของสัญญาณไฟฟ้า หรือจังหวะเวลาในการแลกเปลี่ยนข้อมูลระหว่างการเชื่อมต่อกันนี้ ส่วนของการเชื่อมต่ออินพุตเอาต์พุตนี้จะประกอบด้วยส่วนของซอฟต์แวร์และส่วนของฮาร์ดแวร์รวมกันขึ้นเป็นโครงสร้างที่มีลักษณะต่างๆ มากมาย ดังนั้นผู้พัฒนาจำเป็นจะต้องมีความรู้ครอบคลุมทั้งสองส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่ออินพุตเอาต์พุต

ถึงแม้การประมวลผลอินพุตเอาต์พุตของระบบสมองกลฝังตัวจะถูกควบคุมโดยโปรแกรมก็ตาม แต่ก็มีความสัมพันธ์อย่างใกล้ชิดกับฮาร์ดแวร์ที่เป็นส่วนเชื่อมต่อสัญญาณระหว่าง MPU กับภายนอก ฮาร์ดแวร์ในส่วนนี้ ได้แก่ ส่วนของอินพุตเอาต์พุตพื้นฐาน ซึ่งมีความจำเป็นในการรับส่งข้อมูลข่าวสารระหว่างภายในกับภายนอกระบบ โดยจะต้องสอดคล้องกับจังหวะเวลาของสัญญาณนาฬิกาของ MPU และส่วนของวงจรแปลงสัญญาณต่างๆ ตามจุดประสงค์ที่ต้องการ ทั้งสองส่วนนี้รวมกันเรียกว่า ส่วนเชื่อมต่ออินพุตเอาต์พุต หรืออินพุตเอาต์พุตอินเทอร์เฟส เราเรียกว่าดึงข้อมูลจากภายนอกเข้ามายัง MPU ว่า "อินพุต" และเรียกการส่งข้อมูล MPU ออกไปยังภายนอกว่า "เอาต์พุต"



รูปที่ 2.3.5 การเชื่อมต่ออินพุตเอาต์พุต [1]

อินพุตเอาต์พุตพื้นฐานจะใช้ระดับสัญญาณแรงดันไฟฟ้าสำหรับข้อมูลดิจิทัลเหมือนกับ MPU ส่วนอินพุตเอาต์พุตแบบขนาน จะทำการรับส่งข้อมูลแบบดิจิทัลทีละหลายๆ บิตพร้อมกัน ส่วนอินพุตเอาต์พุตแบบอนุกรม จะใช้สายสัญญาณหนึ่งในการรับส่งข้อมูลแบบดิจิทัล โดยจะทำการรับหรือส่งทีละบิต โดยมีการแปลงข้อมูล (ขนานเป็นอนุกรม อนุกรมเป็นขนาน)

นอกจากนี้ยังมีอุปกรณ์อินพุตเอาต์พุตพื้นฐานที่ทำหน้าที่นับจำนวนครั้งการเปลี่ยนแปลงของสัญญาณดิจิทัลจากภายนอกที่เป็นอินพุตเข้ามา เรียกว่า ตัวนับหรือเคาน์เตอร์ (Counter) และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ที่ทำการจับเวลา (ตามที่กำหนดไว้) แล้วส่งพัลส์ออกไปเมื่อถึงเวลาที่กำหนด เรียกว่า ตัวจับเวลาหรือไทม์เมอร์ (timer) และยังมีอุปกรณ์สำหรับทำการแปลง A/D. D/A สำหรับแปลงสัญญาณแอนะล็อกที่อินพุตเข้ามาให้เป็นสัญญาณดิจิทัลและแปลงสัญญาณดิจิทัลให้กลายเป็นสัญญาณแอนะล็อกส่งออกไปได้

อินพุตเอาต์พุตอินเทอร์เฟสโดยมากจำเป็นจะต้องมีวงจรแปลงสัญญาณ เพื่อแปลงระดับแรงดันไฟฟ้าหรือแปลงพลังงานที่อยู่ก่อนหน้าหรือหลังส่วนที่เป็นอินพุตเอาต์พุตพื้นฐาน แต่อย่างไรก็ตาม วงจรเหล่านี้จำเป็นจะต้องถูกออกแบบให้ตรงกับลักษณะการใช้งาน ซึ่งในที่นี้จะขอกล่าวถึงเพียงแค่อินพุตเอาต์พุตพื้นฐาน

2.3.3 อินพุตเอาต์พุตแบบอนุกรม (Serial I/O)

อินพุตเอาต์พุตแบบอนุกรมเป็นการเชื่อมต่อโดยใช้สายสัญญาณเส้นเดียวในการรับส่งข้อมูล ซึ่งใช้สายสัญญาณจำนวนน้อย จึงเป็นที่นิยมใช้กัน โดยจะเรียกช่วงต่อสำหรับการเชื่อมต่อสายสัญญาณในการใช้งานอินพุตเอาต์พุตแบบอนุกรมนี้ว่า "พอร์ตอนุกรม (serial port)"

ข้อดีของการเชื่อมต่ออินพุตเอาต์พุตแบบอนุกรมนี้คือ การใช้สายสัญญาณน้อย ทำให้ต้นทุนของระบบลดลง แต่มีข้อเสียคือ หากต้องการเพิ่มความเร็วในการรับส่งข้อมูล ก็จะต้องเพิ่มความถี่ของสัญญาณนาฬิกาขึ้น ซึ่งปัจจุบันความถี่ของระบบที่ใช้กันสูงถึงหลายร้อย Mbps (Megabit per second: ล้านบิตต่อวินาที) ถือเป็นความถี่ที่สูงมาก การรับส่งสัญญาณความถี่สูงจะต้องมีการออกแบบบอร์ดและการวางตำแหน่งสายสัญญาณที่ดี เพื่อไม่ให้รูปทรงของสัญญาณเสียไป

วงจรเชื่อมต่อสัญญาณออกของอินพุตเอาต์พุตแบบอนุกรมเป็นวงจรที่แปลงข้อมูลแบบขนานที่ส่งออกท่จาก MPU ให้กลายเป็นข้อมูลแบบอนุกรม โดยมีการแปลงทีละบิต เริ่มจากบิตล่างสุดหรือ LSB (Least Significant Bit) หรือบิตบนสุด MSB (Most Significant Bit) ให้เป็นข้อมูลบนสายสัญญาณเพียงหนึ่งเส้น การประมวลลักษณะนี้เรียกว่า การแปลงข้อมูลขนานเป็นอนุกรม (parallel-sequential conversion) วงจรสำหรับการเชื่อมต่อสัญญาณเข้าจะทำการแปลงข้อมูลทีละบิตที่ถูกส่งมายังอินพุตให้กลายเป็นข้อมูลแบบขนาน (เรียงกันให้ครบจำนวนบิตของข้อมูลแบบขนาน) เรียกการประมวลผลในลักษณะนี้ว่า การแปลงข้อมูลอนุกรมเป็นข้อมูลขนาน (sequential-parallel conversion)

โดยทั่วไปแล้วการรับส่งข้อมูลแบบอนุกรมจะใช้สายสัญญาณสำหรับส่งและรับส่งข้อมูลคนละเส้นแยกจากกัน และส่งข้อมูลทีละบิต หากใช้กราวด์ร่วมกัน ก็จะใช้สายสัญญาณทั้งหมดเพียง 3 เส้นสำหรับการรับส่งข้อมูล นอกจากนี้อาจมีการใช้สายสัญญาณเพิ่มเพื่อทำการตรวจสอบสถานะของฝั่งส่งและฝั่งรับ รวมไปถึงอาจมีการกำหนดรูปแบบการรับส่งข้อมูลว่าเป็นแบบซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(synchronous) หรืออะซิงโครนัส (asynchronous) เป็นแบบ SDLC (synchronous data link control) หรือ HDLC (high-level data link control) ต่าง ๆ เป็นต้น

การสื่อสารแบบอนุกรมอย่างซิงโครนัส (Synchronous serial communication) จะส่งข้อมูลทีละบิต โดยอาจเริ่มจาก LSB (Least Significant Bit) หรือ MSB (Most Significant Bit) และส่งจนครบทั้งหมด วิธีในการซิงโครไนซ์สัญญาณจะต้องมีการตกลงกันระหว่างฝั่งรับและฝั่งส่งตั้งแต่แรก โดยวิธีพื้นฐานที่นิยมในการส่งข้อมูลแบบซิงโครนัส ได้แก่ การใช้สัญญาณนาฬิกาเป็นตัวช่วยในการซิงโครไนซ์สัญญาณ โดยวิธีนี้จะต้องใช้สายสัญญาณ 2 เส้นที่อิสระต่อกัน เส้นหนึ่งเป็นสัญญาณนาฬิกา และอีกเส้นหนึ่งเป็นข้อมูล

การสื่อสารข้อมูลอย่างอะซิงโครนัส (Asynchronous serial communication) ฝั่งส่งและฝั่งรับจะสามารถแบ่งแยกข้อมูลออกจากสัญญาณที่ติดต่อสื่อสารกันได้อย่างถูกต้องโดยไม่จำเป็นต้องมีสายสัญญาณนาฬิกาอีกเส้นหนึ่งต่างหาก แต่จะต้องมีสัญญาณในส่วนที่จะบอกการเริ่มต้นของข้อมูล คือ บิตเริ่มต้น (start bit) และสัญญาณในส่วนที่จะบอกการสิ้นสุดของข้อมูล คือ บิตสิ้นสุด (stop bit)

รูปแบบของการอิมพลีเมนต์ช่องสัญญาณการสื่อสารที่ใช้ส่งข้อมูลอนุกรมอาจแตกต่างกันไปตามลักษณะเฉพาะเชิงไฟฟ้า หรือรูปร่างเชิงกล ปัจจุบันรูปแบบที่เป็นมาตรฐาน ได้แก่ มาตรฐาน RS-232C, RS-422, RS-485 และเคอร์เรนต์ลูปแบบ 20 mA (20 mA current loop) เป็นต้น นอกจากนี้การส่งข้อมูลแบบไร้สายผ่านการเชื่อมต่อสัญญาณอินฟราเรด (IrDA) และมาตรฐานการส่งข้อมูลแบบ USB ก็เป็นที่นิยมกันในปัจจุบัน พอร์ตที่ใช้ในการสื่อสารแบบอนุกรมนิยมเรียกว่า พอร์ตคอม (COM port)

2.4 ไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์หรืออาจเรียกว่าชิงเกิลชิพไมโครคอนโทรลเลอร์ (Single Chip Microcontroller) เป็นอุปกรณ์ที่นำเอาไมโครโพรเซสเซอร์มารวมกับหน่วยความจำและอุปกรณ์ไอโอต่างๆ เอาไว้ในตัวเดียวกัน สามารถทำงานได้ทันทีเมื่อป้อนไฟเลี้ยงและสัญญาณนาฬิกาให้ตัวไมโครคอนโทรลเลอร์ จึงทำให้การออกแบบวงจรง่ายขึ้นและมีขนาดเล็กลงเป็นอย่างมาก ในที่นี้จะขอยกตัวอย่างไมโครคอนโทรลเลอร์ตระกูล MCS-51

การพัฒนาขีดความสามารถของไมโครคอนโทรลเลอร์ได้เป็นมาอย่างต่อเนื่อง ซึ่งมีการพัฒนาทั้งในด้านฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ (Software) ในการพัฒนาด้านฮาร์ดแวร์นั้น สิ่งที่เห็นได้ชัดเจนคือการพัฒนาเกี่ยวกับหน่วยความจำ ซึ่งได้เริ่มมีการพัฒนามาจาก PROM EPROM EEPROM และในปัจจุบัน ได้พัฒนามาจนถึงหน่วยความจำที่เรียกว่า FLASH MEMORY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของซอฟต์แวร์นั้นได้พัฒนาการเขียนโปรแกรมมาจาก การเขียนโปรแกรมในระดับบิต(เลขฐาน2), ระดับไบต์ (เลขฐาน 16), แอสแซมบลีและภาษาในระดับสูง เช่น ภาษาเบสิกและภาษาซี เป็นต้น

2.4.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ใช้เทคโนโลยีในการผลิตแบบ NMOS และ CMOS ซึ่งภายในได้รวมวงจรต่างๆ ไว้อย่างครบถ้วนพร้อมที่จะทำงานได้เมื่อจ่ายไฟเลี้ยงและสัญญาณนาฬิกาไมโครคอนโทรลเลอร์ตระกูล MSC-51 ได้ถูกผลิตออกมามากมายหลายเบอร์โดยบริษัทต่างๆ เช่น บริษัท Atmel, Philips, Dallas, Infinion และบริษัทอื่นๆ ซึ่งไม่ว่าจะเป็นเบอร์อะไรก็ตามถ้าเป็นไมโครคอนโทรลเลอร์ตระกูล MSC-51 และจะมีโครงสร้างต่างๆ ที่คล้ายกันจะแตกต่างกันออกไปในส่วนของความสามารพิเศษของแต่ละเบอร์ ยกตัวอย่างเช่น เบอร์ AT89C51 มีไทม์เมอร์ 2 ตัว ในขณะที่เบอร์ AT89C52 มีไทม์เมอร์ 3 ตัว เป็นต้น

แต่อย่างไรก็ตามในหนังสือเล่มนี้จะใช้เบอร์ 89C52 เป็นตัวอ้างอิงในการอธิบายการทำงานต่างๆ ซึ่งสามารถนำไปเป็นพื้นฐานในการเรียนรู้การทำงานของไมโครคอนโทรลเลอร์เบอร์อื่นๆ ต่อไป

2.4.2 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ MCS-51

เนื่องจากคุณสมบัติของไมโครคอนโทรลเลอร์แต่ละเบอร์นั้นจะมีแตกต่างกันในรายละเอียดปลีกย่อยดังได้กล่าวแล้วข้างต้น ดังนั้นในที่นี้จะขออ้างอิงถึงเบอร์ AT89C52 ของบริษัท Atmel ซึ่งเป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต ซึ่งจะใช้อ้างอิงในหนังสือเล่นนี้ตลอดทั้งเล่ม และสามารถใช้อ้างอิงร่วมกับชิพเบอร์อื่นๆ ในตระกูลเดียวกันได้ เช่น AT89C51, AT89S51/52/53 และ AT89S8252 เป็นต้น สำหรับในการทดลองจะเน้นไปที่เบอร์ AT89S5252 ซึ่งมีคุณสมบัติดังนี้

* มีหน่วยความจำโปรมแกรมแบบแฟรช (Flash Memory) ขนาด 8 กิโลไบต์
  - โปรแกรมผ่านพอร์ตอนุกรมมาตรฐาน SPI (SPI Serial Interface)
  - สามารถโปรแกรมและลบซ้ำได้นับ 1,000 ครั้ง
* มีหน่วยความจำแบบ EEPROM ขนาด 2 กิโลไบต์
  - สามารถโปรแกรมและลบซ้ำได้นับ 100,000 ครั้ง
* ใช้แหล่งจ่ายไฟฟ้ากระแสตรงขนาด 5 โวลท์ (ทำงานได้ในช่วง 4-6 โวลท์)
* ทำงานได้ด้วยสัญญาณนาฬิกาตั้งแต่ 0-24 MHz
* สามารถป้องกันหน่วยความจำได้ 3 ระดับ
* มีหน่วยความจำข้อมูล (RAM) ขนาด 256 ไบต์
* มีพอร์ต 32 พอร์ต อิสระสามารถเข้าถึงในระดับบิตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

* มีไทม์เมอร์/เคาเตอร์ขนาด 16 bit ทั้งหมด 3 ตัว

* รองรับการอินเตอร์รัปต์ได้ 8 แหล่ง

* สามารถสื่อสารข้อมูลแบบอนุกรมได้ด้วย UART Channel

* มีโหมดการทำงานแบบ Low Power Idle และPower Down สำหรับการประหยัดพลังงาน

* มี Watchdog Timer เพื่อเพิ่มเสถียรภาพการทำงานของระบบ

2.4.3 การจัดการของไมโครคอนโทรลเลอร์ตระกูล MCS-51 (เบอร์ 89C52)

VCC ต่อไฟเลี้ยง (supply voltage)

GND ต่อกราวด์ (ground)

Port0 (P0.0-P2.7) เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต สามารถทำงานได้สองหน้าที่คือเป็นพอร์ตอินพุตเอาต์พุตทั่วไป และใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอก คือ รับ/ส่งข้อมูลและกำหนดแอดเดรสไบต์ต่ำ

Port1 (P1.0-P1.7) เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อความต้านทานพูลอัพ (pull-up resistor) ไว้ภายในทำหน้าที่เป็นพอร์ตอินพุตเอาท์พุตทั่วไป นอกจากนี้ยังใช้งานเป็นขาอินพุตเอาต์พุตของไทม์เมอร์ 2

Port2 (P2.0-P2.7) เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อความต้านทานพูลอัพ (pull-up resistor) ไว้ภายในสามารถทำงานได้สองหน้าที่คือเป็นพอร์ตอินพุตเอาต์พุตทั่วไป และใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอกคือกำหนดแอดเดรสไบต์สูง

Port3 เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อความต้านทานพูลอัพ (pull-up resistor) ไว้ภายในทำหน้าที่เป็นพอร์ตอินพุตเอาท์พุตทั่วไป นอกจากนี้ยังใช้งานเป็นขาสัญญาณควบคุมการติดต่อกับหน่วยความจำการอินเตอร์รัปต์ และอื่นๆ

RST เป็นขาอินพุตที่ใช้รับสัญญาณสำหรับรีเซตซีพียู ซีพียูจะถูกรีเซตเมื่อขานี้เป็นลอจิก "1" นาน 2 แมชชินไซเคิล หรือ 24 ไซเคิลของสัญญาณนาฬิกา

ALE/PROG ทำหน้าที่เป็นขาเอาต์พุตเมื่อซีพียูต้องการติดต่อกับหน่วยความจำภายนอก คือจะทำการส่งสัญญาณพัลซ์ออกมาที่ขานี้เพื่อทำการแลตแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก และขานี้จะเป็นอินพุตเมื่ออยู่ในระหว่างโปรแกรมแฟรช

PSEN เป็นขาเอาต์พุตใช้ในการติดต่อกับหน่วยความจำโปรแกรมภายนอก คือเมื่อซีพียูทำการประมวลผลกับหน่วยความจำโปรแกรมภายนอกขานี้จะแอดทีฟสองครั้งในแต่ละแมชชินไซเคิล

EA/VPP เป็นขาอินพุตและต้องการลอจิก "0" เพื่อยอมให้ซีพียูสามารถเข้าถึงหน่วยความจำโปรแกรมภายนอกได้ ซึ่งอยู่ที่ตำแหน่ง 0000H ถึง FFFFH นอกจากนี้แล้วขานี้ยังใช้รับไฟ 12 โวลท์เพื่อใช้ในระหว่างที่ทำการโปรแกรมแฟรช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XTAL1 เป็นขาอินพุตของวงจรออสซิลเลเตอร์แอมปลิไฟเออร์ และยังเป็นอินพุตของวงจรกำเนิดสัญญาณนาฬิกาภายใน

XTAL2 เป็นขาเอาต์พุตของวงจรออสซิลเลเลอร์แอมปลิไฟเออร์

2.5 การใช้งานพอร์ตต่างๆ

การนำไมโครคอนโทรลเลอร์ไปใช้งานในลักษณะต่างๆ นั้น ไม่สามารถหลีกเลี่ยงการใช้พอร์ตได้ เนื่องจากพอร์ตเป็นเส้นทางในการรับและส่งสัญญาณระหว่างอุปกรณ์ต่างๆ จากภายนอก ไม่ว่าจะเป็นอุปกรณ์เอาต์พุต (Output) หรืออุปกรณ์อินพุต (Input) โดยทั่วไปพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 มีทั้งหมด 4 พอร์ต คือ พอร์ต0, พอร์ต1, พอร์ต2 และพอร์ต 3 (ไมโครคอนโทรลเลอร์บางเบอร์อาจมีพอร์ตมากหรือน้อยกว่า 4 พอร์ต) การใช้งานพอร์ตต่างๆ ต้องเลือกใช้งานให้เหมาะสมกับงานและความสามารถของพอร์ตนั้น

2.5.1 พอร์ต 0 (P0.0-P0.7)

พอร์ต 0 เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต พอร์ต 0 สามารถทำงานได้ 2 หน้าที่คือเป็นพอร์ตอินพุตเอาต์พุตสำหรับใช้งานทั่วไปแต่ในกรณีที่ใช้งานเป็นพอร์ตอินพุต เอาต์พุตพอร์ต 0 จะไม่สามารถจ่ายแรงดันและกระแสได้ แต่สามารถรับกระแสได้ตามปกติดังนั้นพอร์ตนี้ไม่สามารถใช้ขับอุปกรณ์ที่รับกระแสได้ เช่นหากนำLEDไปต่อโดยเอาขาA ต่อที่ขาพอร์ต และขาK ต่อกราวด์ LED จะไม่สว่างเพราะพอร์ตนี้ไม่สามารถจ่ายไฟให้LEDได้ แต่อย่างไรก็ตามสามารถดัดแปลงให้พอร์ต0 จ่ายกระแสได้ โดยการต่อตัวต้านทานพูลอัพ (pull-up resistor) ซึ่งกระแสที่จ่ายออกมานั้นขึ้นอยู่กับตัวต้านทานพูลอัพ

ในกรณีที่มีการต่อหน่วยความจำภายนอก พอร์ต0 จะทำหน้าที่เป็นพอร์ตซึ่งกำหนดตำแหน่ง (address) ไบต์ต่ำ และเป็นพอร์ตสำหรับรับส่งข้อมูลกับหน่วยความจำภายนอก ซึ่งจะเรียกการทำงานของพอร์ตลักษณะนี้ว่า "multiplexed low-order address/data bus"

วงจรภายในของพอร์ต0 มีอยู่ 2 ส่วน ส่วนแรกคือ วงจรมัลติเพล็ก (Multiplex) ทำหน้าที่เลือกการทำงานของพอร์ตว่าจะให้เป็นพอร์ตอินพุตเอาต์พุตทั่วไปหรือจะให้เป็นพอร์ตเพื่อใช้ติดต่อกันกับหน่วยความจำภายนอก ส่วนที่สองคือวงจร แลตซ์ (Latch) ซึ่งเป็น ดี-ฟลิปฟลอบ (D-Flip Flop) ทำหน้าที่คงสภาวะข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 พอร์ต1 (P1.0-P1.7)

พอร์ต 1 เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อความต้านทานชนิดพูลอัพ (pull-up resister) ไว้ภายใน ทำหน้าที่เป็นพอร์ตอินพุตเอาท์พุตทั่วไป นอกจากนี้ยังใช้งานเป็นขาอินพุต/เอาต์พุตของไทม์เมอร์2 ซึ่งรายละเอียดแสดงในตารางที่3.1 ในรูปที่3.2 จะเห็นได้ชัดว่าพอร์ต1 จะไม่มีส่วนของวงจรมัลติเพล็ก (Multiplex) เพราะพอร์ต1 ไม่ได้ใช้ในการติดต่อกับหน่วยความจำภายนอก แต่มีวงจรแลตช์ (Latch) ซึ่งเป็น ดี-ฟลิบฟลอบ (D-Flip Flop) ทำหน้าที่คงสภาวะข้อมูล

Port Pin	Alternate Functions
P1.0	T2 (external count input Timer/Counter), clock-out
P2.0	T2EX (Timer/Counter 2 capture/reload trigger and direction control)

ตาราง 2.5.2 หน้าที่พิเศษของพอร์ต 1

2.5.3 พอร์ต2 (P2.0-P2.7)

พอร์ต 2 เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อความต้านทานพูลอัพ (pull- up resister) ไว้ภายใน สามารถทำงานได้สองหน้าที่คือเป็นพอร์ตอินพุต / เอาท์พุตทั่วไป และใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอกคือกำหนดแอดเตรสไบต์สูง

วงจรมัลติเพล็ก และวงจรแลตช์จะเหมือนกันกับพอร์ต0 คือ วงจรมัลติเพล็กทำหน้าที่เลือกการทำงานของพอร์ตว่าจะให้เป็นพอร์ตอินพุต / เอาท์พุตทั่วไปหรือจะให้เป็นพอร์ตเพื่อใช้ติดต่อกับหน่วยความจำภายนอก และวงจรแลตช์ (Latch) ซึ่งเป็น ดี-ฟลิบฟลอบ (D-Flip Flop) ทำหน้าที่คงสภาวะข้อมูล

2.5.4 พอร์ต 3 (P3.0-P3.7)

พอร์ต 3 เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อความต้านทานพูลอัพ (pull-up resister) ไว้ภายใน ทำหน้าที่เป็นพอร์ตอินพุตเอาท์พุตทั่วไป นอกจากนี้พอร์ต 3 ยังทำหน้าที่พิเศษต่างๆ เช่นการสื่อสารข้อมูลแบบอนุกรม, การอินเตอร์รัปต์จากภายนอก, รับ/ส่งสัญญาณสำหรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไทม์เมอร์1 และสัญญาณควบคุมการอ่านและเขียนหน่วยความจำ ซึ่งรายละเอียดต่างๆ ของพอร์ต 3 แสดงในตารางที่ 2.2.4

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0/(external interrupt 0)
P3.3	INT1/(external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	WR/(external data memory write strobe)
P3.7	RD/(external data memory read strobe)

ตารางที่ 2.5.4 หน้าที่พิเศษของพอร์ต 3

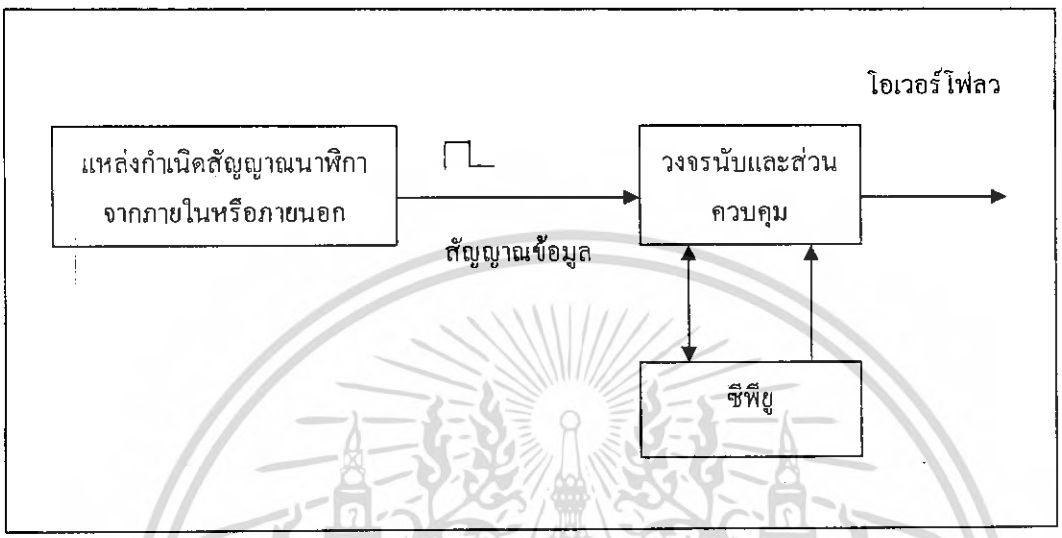
2.6 ไทม์เมอร์ / เคาเตอร์ (Timer/Counter)

ไทม์เมอร์/เคาเตอร์ เป็นวงจรภายในที่ช่วยเพิ่มความสามารถให้ไมโครคอนโทรเลอร์ เนื่องจากไทม์เมอร์/ เคาเตอร์จะมีลักษณะการทำงานเป็นตัวนับเวลาหรือนับสัญญาณนาฬิกา ซึ่งสามารถทำงานเกี่ยวกับการสร้างฐานเวลา, การสร้างสัญญาณพัลซ์, การเปรียบเทียบค่าของเวลาหรือค่าการนับสัญญาณพัลซ์ที่รับเข้ามาจากภายนอก และรวมไปถึงการควบคุมอัตราการรับส่งข้อมูลของพอร์ตอนุกรมด้วย ไทม์เมอร์/เคาเตอร์เป็นวงจรที่สามารถทำงานได้เองโดยเป็นอิสระจากซีพียู แต่ซีพียูสามารถควบคุมการทำงาน, การอ่านและเขียนข้อมูลต่าง ๆ ได้

ไมโครคอนโทรลเลอร์แต่ละเบอร์อาจจะมีไทม์เมอร์ / เคาเตอร์จำนวนต่างกัน หรือมีความสามารถที่แตกต่างกัน เช่น เบอร์AT89C51 มีไทม์เมอร์ / เคาเตอร์จำนวน 2 ตัว(T0 และ T1) ในขณะที่เบอร์ AT89C52 หรือเบอร์ AT89S8252 มีไทม์เมอร์ / เคาเตอร์จำนวน 3 ตัว(T1 และ T2) เป็นต้น ไทม์เมอร์ / เคาเตอร์แต่ละตัวอาจจะมีความสามารถที่แตกต่างกันในรายละเอียดทางฮาร์ดแวร์และการทำงาน แต่อย่างไรก็ตามจะมีหลักการทำงานที่เหมือนกันคือ ถูกควบคุมจากซีพียู และใช้สัญญาณนาฬิกามาเป็นสัญญาณอินพุตในกรณีใช้งานเป็นไทม์เมอร์หรือรับสัญญาณอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภายนอกในกรณีใช้งานเป็นเคาเตอร์นอกจากนี้แล้วไทม์เมอร์/เคาเตอร์ทุกตัวยังมีการทำงานที่เป็นอิสระต่อกันอีกด้วย ในที่นี้จะขอกล่าวเพียงสองตัวเท่านั้น คือไทม์เมอร์/เคาเตอร์ 0 และไทม์เมอร์/เคาเตอร์ 1



รูปที่ 2.6 โครงสร้างโดยรวมของไทม์เมอร์/เคาเตอร์

ไทม์เมอร์/เคาเตอร์ 0 และไทม์เมอร์/เคาเตอร์ 1

2.6.1 การทำงานในโหมดไทม์เมอร์ (Timer Mode) ของไทม์เมอร์/เคาเตอร์ 0 และ 1

เมื่อกำหนดให้มีการทำงานเป็นโหมดไทม์เมอร์หรือตัวตั้งเวลา วงจรนับหรือเคาเตอร์จะทำการนับขึ้นโดยใช้สัญญาณนาฬิกาภายในซึ่งสัญญาณนาฬิกาตัวนี้จะเกิดขึ้นทุกๆ 1 แมชชินไซเคิล (Machine cycle) นั่นหมายความว่าจะเกิดการนับขึ้น 1 ค่าในแต่ละแมชชีนไซเคิลนั่นเอง โดยในไมโครคอนโทรลเลอร์โดยทั่วๆ ไป 1 แมชชีนไซเคิลจะใช้เวลา 12 คาบของสัญญาณนาฬิกาที่ป้อนให้กับตัวไมโครคอนโทรลเลอร์ นั่นหมายความว่าจะเกิดการนับขึ้นทุกๆ 1/12 ของความถี่สัญญาณนาฬิกา(บางเบอร์อาจใช้แค่ 4 คาบ เช่น เบอร์ DS80C320 หรือ ใช้ 6 คาบ เช่นเบอร์ P89V51RD2 เป็นต้น)

2.6.2 การทำงานในโหมดเคาเตอร์ (Counter Mode) ของไทม์เมอร์/เคาเตอร์ 0 และ 1

เมื่อกำหนดให้มีการทำงานในโหมดเคาเตอร์หรือตัวนับ วงจรนับหรือเคาเตอร์ภายในจะทำการนับขึ้นโดยใช้สัญญาณขอบขาลง (เปลี่ยน 1 เป็น 0) จากภายนอกที่เข้ามาทางขา T0 (P3.4), T1 (P3.5) กระบวนการอ่านสัญญาณอินพุตเหล่านี้จะใช้เวลาทั้งสิ้น 2 แมชชีนไซเคิล นั่นหมายความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว่าอัตราการนับจะมีค่าเท่ากับ 1/24 (กรณี 1 แมชชินไซเคิลใช้เวลา 12 คาบ) ของความถี่สัญญาณนาฬิกา ดังนั้นสามารถใช้การทำงานในโหมดเคาเตอร์เพื่อนับสัญญาณนาฬิกาได้สูงสุดเท่ากับความถี่สัญญาณนาฬิกา/24 (กรณี 1 แมชชินไซเคิลใช้เวลา 12 คาบ) เช่นใช้สัญญาณนาฬิกา 12 MHz จะสามารถรับสัญญาณอินพุตที่มีความถี่สูงได้ 12 MHz/24 = 500 kHz (กรณี 1 แมชชินไซเคิลใช้เวลา 12 คาบ)

2.6.3 รีจิสเตอร์ของไทม์เมอร์/ เคาเตอร์ 0 และ 1 (Timer / Counter0, 1 Register)

การกำหนดให้ไทม์เมอร์ / เคาเตอร์ ทำงานตามความต้องการนั้นจะต้องมีการกำหนดค่าต่างๆ ให้กับรีจิสเตอร์ที่เกี่ยวข้องกับการทำงานอย่างเหมาะสม ซึ่งรีจิสเตอร์เหล่านี้เป็นรีจิสเตอร์ฟังก์ชันพิเศษ (SFR) ซึ่งมีรายละเอียดดังนี้

TL0 เป็นรีจิสเตอร์ไบต์ต่ำของไทม์เมอร์ 0 มีแอดเดรสอยู่ที่ 8AH

TH0 เป็นรีจิสเตอร์ไบต์สูงของไทม์เมอร์ 0 มีแอดเดรสอยู่ที่ 8BH

TL1 เป็นรีจิสเตอร์ไบต์ต่ำของไทม์เมอร์ 1 มีแอดเดรสอยู่ที่ 8CH

TH1 เป็นรีจิสเตอร์ไบต์สูงของไทม์เมอร์ 1 มีแอดเดรสอยู่ที่ 8DH

รีจิสเตอร์ของไทม์เมอร์ทั้ง 4 ตัวนี้เป็นรีจิสเตอร์ขนาด 8 บิต ดังนั้นเมื่อนำ TL0 มารวมกับ TH0 จะทำให้ได้รีจิสเตอร์ของไทม์เมอร์ 0 ขนาด 16 บิต ในทำนองเดียวกันเมื่อนำ TL1 มารวมกับ TH1 จะทำให้ได้รีจิสเตอร์ของไทม์เมอร์ 1 ขนาด 16 บิต ซึ่งสามารถเก็บค่าได้สูงสุดถึง 65,536 ค่า (0-65,535 หรือ 0000H ถึง FFFFH)

2.6.4 รีจิสเตอร์ควบคุมการทำงานของไทม์เมอร์/เคาเตอร์ 0 และ1 (TCON: Timer/Counter0, 1 Control Register)

TCON เป็นรีจิสเตอร์ที่ทำงานของไทม์เมอร์/เคาเตอร์ 0 และ1 มีขนาด 8 บิตและมีแอดเดรสอยู่ที่ 88H และสามารถเข้าถึงในระดับบิตได้ ในแต่ละบิตมีหน้าที่แตกต่างกันออกไปดังนี้

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

รูปที่ 2.6.4 บิตต่างๆ ในรีจีสเตอร์ TCON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

***TF 1 (Timer 1overflow flag)** จะเกิดการเซตด้วยขบวนการฮาร์ดแวร์ คือเมื่อการนับเพิ่มขึ้นจนเกิดการโอเวอร์โฟลว (เปลี่ยนจาก FFH ไปเป็น 00H ในกรณีที่เป็น 8บิต และเปลี่ยนจาก FFFFH ไปเป็น 0000H กรณีที่เป็น 16บิต) บิตนี้จะถูกเคลียร์โดยกระบวนการทางฮาร์ดแวร์เช่นกัน โดยจะเคลียร์เมื่อ ซีพียูกระโดดปทำงานในโปรแกรมบริการอินเตอร์รัปต์ของไทม์เมอร์ 1 โอเวอร์โฟลว์

***TR 1 (Timer 1 run control bit)** ใช้ควบคุมการเปิด/ปิด (Enable/Disable) การทำงานของไทม์เมอร์ 1การเซตและการเคลียร์สามารถทำได้โดยกระบวนการทางซอฟแวร์

"0" ปิดการทำงานของไทม์เมอร์ 0 (disable)

"1" เปิดการทำงานของไทม์เมอร์ 0 (enable)

***TF 0 (Timer 0 overflow flag)** จะเกิดการเซตด้วยกระบวนการทางฮาร์ดแวร์ คือเมื่อการนับเพิ่มขึ้นจนถึงโอเวอร์โฟลว (เปลี่ยนจาก FFH ไปเป็น 00H ในกรณีที่เป็นแบบ 8 บิต และเปลี่ยนจาก FFFFH ไปเป็น 0000H กรณีที่เป็น 16บิต) บิตนี้จะถูกเคลียร์โดยกระบวนการทางฮาร์ดแวร์เช่นกันโดยจะเคลียร์เมื่อ ซีพียูกระโดดปทำงานในโปรแกรมบริการการอินเตอร์รัปต์ของไทม์เมอร์ 0โอเวอร์โฟลว

***TR 0 (Timer 0 run control bit)** ใช้ควบคุมการเปิด/ปิด (Enable/Disable) การทำงานของไทม์เมอร์ 1การเซตและการเคลียร์สามารถทำได้โดยกระบวนการทางซอฟแวร์

"0" ปิดการทำงานของไทม์เมอร์ 0 (disable)

"1" เปิดการทำงานของไทม์เมอร์ 0 (enable)

***IE 1 (External Interrupt 1 edge flag)** ใช้ในขบวนการอินเตอร์รัปต์ คือจะเกิดการเซตโดยกระบวนการทางฮาร์ดแวร์ นั่นคือเมื่อตรวจพบว่ามีสัญญาณอินเตอร์รัปต์เข้ามาที่ขาอินพุตของอินเตอร์รัปต์ 1 (INT1: P3.3) บิตนี้จะเซต และจะเคลียร์เมื่อ ซีพียูจะกระโดดไปทำงานในโปรแกรมบริการอินเตอร์รัปต์ของ external interrupt 1

***IT1 (Interrupt 1 type control bit)** ใช้ในกระบวนการอินเตอร์รัปต์ของเอ็กเทอร์นอลอินเตอร์รัปต์ 1 โดยเป็นตัวกำหนดว่าจะให้อินเตอร์รัปต์เกิดขึ้นเมื่อตรวจพบสัญญาณที่เข้ามาทางขาอินพุตของอินเตอร์รัปต์ 1 (INT1: P3.3) เป็นการตรวจสอบสัญญาณขอบขาลงหรือระดับลอจิก" 0 "การเซตและเคลียร์บิตนี้สามารถทำได้โดยกระบวนการทางซอฟแวร์

***IE 0 (External Interrupt 0 edge flag)** ใช้ในขบวนการอินเดอร์รัปต์ คือจะเกิดการเซตโดยกระบวนการทางฮาร์ดแวร์ นั่นคือเมื่อตรวจพบว่ามีสัญญาณอินเตอร์รัปต์เข้ามาที่ขาอินพุตของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเตอร์รัปต์ (INT1: P3.2) บิตนี้จะเซต และจะเคลียร์เมื่อ ซีพียูจะกระโดดไปทำงานในโปรแกรมบริการอินเตอร์รัปต์ของ external interrupt 0

***IT0 (Interrupt 0 type control bit)** ใช้ในกระบวนการอินเตอร์รัปต์ของเอกเทอร์นอลอินเตอร์รัปต์ 0โดยเป็นตัวกำหนดว่าจะให้อินเตอร์รัปต์เกิดขึ้นเมื่อตรวจพบสัญญาณที่เข้ามาทางขาอินพุตของอินเตอร์รัปต์ 0(INT1: P3.2 เป็นการตรวจสอบสัญญาณขอบขาลงหรือระดับลอจิก "0" การเซตและเคลียร์บิตนี้สามารถทำได้โดยกระบวนการทางซอฟต์แวร์

"0" จะเกิดการอินเตอร์รัปต์เมื่อตรวจพบสัญญาณขอบขาลง (falling edge)

"1" จะเกิดการอินเตอร์รัปต์เมื่อตรวจพบสัญญาณเป็นลอจิก" 0" (low Level triggered)

2.6.5 รีจิสเตอร์เลือกโหมดการทำงานของไทม์เมอร์/เคาเตอร์ 0 และ 1 (TMOD: Timer/ Counter 0,1Mode Control Register)

TMOD รีจิสเตอร์ที่ทำงานของไทม์เมอร์/เคาเตอร์ 0 และ1 มีขนาด 8 บิตและมีแอดเดรสอยู่ที่ 89H ไม่สามารถเข้าถึงในระดับบิตได้ รีจิสเตอร์ตัวนี้จะแยกออกเป็นสองส่วน คือ 4 บิตบนจะใช้เลือกโหมดของไทม์เมอร์/เคาเตอร์1 และ 4 บิตล่างจะใช้เลือกโหมดของไทม์เมอร์/เคาเตอร์ 0 โดยในแต่ละบิตจะมีหน้าที่แตกต่างกันออกไปดังนี้

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Gate	C/T'	M1	M0	GATE	C/T'	M1	M0

รูปที่ 2.6.5 บิตต่างๆ ในรีจีสเตอร์ TMOD

***GATE** ทำหน้าที่เป็นตัวเลือกรูปแบบการทำงานของไทม์เมอร์/เคาเตอร์ ว่าจะให้มีการควบคุมการทำงานแบบซอฟแวร์หรือฮาร์ดแวร์ดังนี้

"0"เลือกรูปแบบการควบคุมการทำงานเป็นแบบซอฟแวร์ คือ ไทม์เมอร์/เคาเตอร์ x จะทำงานเมื่อ TRx ในรีจิสเตอร์ TCON เป็น "1"

"1" เลือกรูปแบบการควบคุมการทำงานเป็นแบบฮาร์ดแวร์ คือ ไทม์เมอร์/เคาเตอร์ x จะทำงานเมื่อ TRx ในรีจิสเตอร์ TCON เป็น "1" และที่ขาอินพุตของอินเตอร์รัปต์ INTx เป็นสถานะลอจิก "1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

***M1. M0** (Mode Selector Bit) เป็นบิตที่ใช้เลือกโหมดการทำงานของไทเตอร์ซึ่งมีดังนี้

"00" เลือกโหมดการทำงานของไทม์เมอร์/เคาเตอร์เป็นแบบ 13 บิต

"01" เลือกโหมดการทำงานของไทม์เมอร์/เคาเตอร์เป็นแบบ 16 บิต

"10" เลือกโหมดการทำงานของไทม์เมอร์/เคาเตอร์เป็นแบบ 8 บิต รีโหลดค่าอัตโนมัติ

"11"เลือกโหมดการทำงานของไทม์เมอร์/เคาเตอร์0 (ไทม์เมอร์/เคาเตอร์ 1 ไม่ทำงาน) ให้ทำงานแบบแยกส่วน โดยแยกออกเป็นไทม์เมอร์/เคาเตอร์สองตัว โดยตัวแรกจะใช้รีจิสเตอร์ TL0 จะถูกควบคุมการทำงานโดย TR0 ส่วนตัวที่สองจะใช้รีจิสเตอร์ TH0 จะถูกควบคุมการทำงานโดย TR1 (TR0 และTR1 อยู่ในรีจิสเตอร์ TCON)

2.7 กระบวนการอินเตอร์รัปต์

การอินเตอร์รัปต์คือการขัดจังหวะการทำงานของ CPU เพื่อ CPU หยุดการประมวลผลในปัจจุบันเอาไว้ชั่วคราวก่อน จากนั้นจะกระโดดไปทำงานในฟังชันก์หรือชุดคำสั่งของการอินเตอร์รัปต์ฟังก์ชันหรือชุดคำสั่งที่ CPU กระโดดไปประมวลผลเมื่อเกิดการอินเตอร์รัปต์ จะเรียกว่าโปรแกรมสำหรับบริการอินเตอร์รัปต์ (Interrupt Service Routine : ISR) เมื่อ CPU ประมวลผมชุดคำสั่งในโปรแกรมการสำหรับบริการอินเตอร์รัปต์เสร็จสิ้น CPU จะกลับไปประมวลผลโปรแกรมที่ได้หยุดไว้ก่อนหน้าที่จะมีการอินเตอร์รัปต์ การเกิดอินเตอร์รัปต์ของไมโครคอลโทรลเลอร์ตระกูล MCS-51 ที่นิยมใช้กันทั่วไป (40 ขา) จะเกิดขึ้นได้จาก 6 แหล่งกำเนิดสัญญาณอินเตอร์รัปต์ แต่มีเวกเตอร์ (Vector) ของการอินเตอร์รัปต์เพียง 5 เวกเตอร์ เนื่องจากการอินเตอร์ที่เกิดจากพอร์ตอนุกรม (Serial port) ทั้งการรับข้อมูลและการส่งข้อมูลจะใช้เวกเตอร์ร่วมกัน การอินเตอร์รัปต์เกิดขึ้นได้ 2 ลักษณะ คือการอินเตอร์รัปต์จากภายนอกและการอินเตอร์รัปต์จากภายในรายละเอียดของการอินเตอร์รัปต์ต่างๆ มีดังต่อไปนี้

2.7.1 รีจิสเตอร์ที่เกี่ยวข้องกับกระบวนการอินเตอร์รัปต์

กระบวนการอินเตอร์รัปต์มีรีจิสเตอร์ที่คอยควบคุมการทำงานอยู่ทั้งหมด 4 ตัว แต่ละตัวมีหน้าที่การทำงานที่แตกต่างกันออกไป ดังนี้

(ก) รีจีสเตอร์ควบคุมการปิด/เปิดการอินเตอร์รัปต์ (Interrupt Enable register; IE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EA	-	ET2	ES	ET1	EX1	ET0	EX0

รูปที่ 2.7.1 บิตต่างๆ ของรีจิสเตอร์ IE

รีจิสเตอร์ IE เป็นรีจิสเตอร์ฟังก์ชันพิเศษมีขนาด 8 บิตอยู่ที่แอดเดรส 0×A8 สามารถเข้าถึงในระดับบิตได้ ใช้ในการควบคุมการปิด/เปิด หรือการตอบสนองของการอินเตอร์รัปต์ต่างๆ รายละเอียดของแต่ละบิตมีดังต่อไปนี้

EA (Global enable/disable interrupt): ใช้ในการปิด/เปิดการอินเตอร์รัปต์ทั้งหมด บิตนี้เป็นบิตที่มีลำดับความสำคัญสูงสุด คือการอินเตอร์รัปต์ต่างๆ จะไม่สามารถเกิดขึ้นได้เลย ถ้ามีการปิดการอินเตอร์รัปต์ที่บิตนี้เอาไว้ บิต EA สามารถเข้าถึงเพื่อเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น

"0" ปิดการอินเตอร์รัปต์เพื่อให้การอินเตอร์รัปต์ทั้งหมด*ไม่*สามารถเกิดขึ้นได้

"1" เปิดการอินเตอร์รัปต์เพื่อให้การอินเตอร์รัปต์ทั้งหมดสามารถเกิดขึ้นได้

EA2 (Timer2 interrupt enable): ใช้ในการปิด/เปิดการอินเตอร์รัปต์จากไทม์เมอร์ 2 อันเนื่องจากไทม์เมอร์ 2 มีการนับขึ้นจนเกิดการโอเวอร์โฟลว์ หรือมีการแคบเจอร์เกิดขึ้น บิตนี้สามารถเข้าถึงเพื่อเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น

"0" ปิดการอินเตอร์รัปต์จากไทม์เมอร์ 2

"1" เปิดการอินเตอร์รัปต์จากไทม์เมอร์ 2

ES (Serial port interrupt enable): ใช้ในการปิด/เปิดการอินเตอร์รัปต์จากพอร์ตอนุกรมอันเนื่องมาจากกระบวนการรับหรือส่งข้อมูล บิตนี้สามารถเข้าถึงเพื่อเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น

"0" ปิดการอินเตอร์รัปต์จากพอร์ตอนุกรม

"1" เปิดการอินเตอร์รัปต์จากพอร์ตอนุกรม

ET1 (Timer1 interrupt enable): ใช้ในการปิด/เปิดการอินเตอร์รัปต์จากไทม์เมอร์ 1 อันเนื่องจากการเกิดโอเวอร์โฟลว์ของไทม์เมอร์ 1 บิตนี้สามารถเข้าถึงเพื่อเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น

"0" ปิดการอินเตอร์รัปต์จากไทม์เมอร์ 1

"1" เปิดการอินเตอร์รัปต์จากไทม์เมอร์ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EX1 (External interrupt1 enable): ใช้ในการปิด /เปิดการอินเตอร์รัปต์จากจากภายนอกหมายเลข 1 (INT1; P3.3) อันเนื่องมาจากมีสัญญาณลอจิก "0" หรือสัญญาณขอบขาลงเข้ามาที่ขา INT1 บิตนี้สามารถเข้าถึงเพื่อเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น

"0" ปิดการอินเดอร์รัปต์จากภายนอกหมายเลข 1

"1" เปิดการอินเตอร์รัปต์จากภายนอกหมายเลข 1

ET0 (Timer0 interrupt enable): ใช้ในการปิด/เปิดการอินเตอร์รัปต์จากไทม์เมอร์ 0 อันเนื่องมาจากการเกิดโอเวอร์โฟลว์ของไทม์เมอร์ 0 บิดนี้สามารถเข้าถึงเพื่อเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น

"0" ปิดการอินเตอร์รัปต์จากไทม์เมอร์ 0

"1" เปิดการอินเตอร์รัปต์จากไทม์เมอร์ 0

EX0 (External interrupt0 enable): ใช้ในการปิด/เปิดการอินเตอร์รัปต์จากจากภายนอกหมายเลข 0 (INT0; P3.2) อันเนื่องมาจากมีสัญญาณลอจิก "0" หรือสัญญาณขอบขาลงเข้ามาที่ขา INT0 บิตนี้สามารถเข้าถึงเพื่อเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น

"0" ปิดการอินเตอร์รัปต์จากภายนอกหมายเลข 0

"1" เปิดการอินเตอร์รัปต์จากภายนอกหมายเลข 0

(ข) รีจิสเตอร์จัดลำดับความสำคัญของการอินเตอร์รัปต์ (Interrupt Priority register; IP)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	PT2	PS	PT1	PX1	PT0	PX0

รูปที่ 2.7.2 บิดต่างๆ ของรีจิสเตอร์ IP

รีจิสเตอร์ IP เป็นรีจิสเตอร์ฟังก์ชันพิเศษ มีขนาด 8 บิตอยู่ที่แอดเดรส 0×88 สามารถเข้าถึงในระดับบิตได้ ใช้ในการควบคุมลำดับความสำคัญของการอินเตอร์รัปต์ต่างๆ เนื่องจากในบางครั้งการอินเตอร์รัปต์จะเกิดขึ้นพร้อมกันหรือซ้อนกัน ดังนั้นจึงต้องมีการจัดลำดับความสำคัญของการอินเตอร์รัปต์เพราะเมื่อมีการอินเตอร์รัปต์เกิดขึ้นซ้อนกัน CPU จะไปประมวลผลในโปรแกรมบริการ การอินเตอร์รัปต์ที่มีความสำคัญสูงกว่าก่อน หากต้องการให้การอินเตอร์รัปต์จากแหล่งกำเนิดไหนสูงกว่าสามารถทำได้โดยการให้บิตประจำตำแหน่งของการอินเตอร์รัปต์เป็น "1" รายละเอียดของแต่ละบิตมีดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PT2 (Timer2 interrupt priority): ใช้ในการกำหนดลำดับความสำคัญของการอินเตอร์รัปต์ที่เกิดขึ้นจากไทม์เมอร์ 2 (ไทม์เมอร์ 2 เกิดการนับขึ้นจึงเกิดการโอเวอร์โฟลว์ หรือมีการแคปเจอร์เกิดขึ้น) บิตนี้สามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟแวร์เท่านั้น

"0" ลำดับความสำคัญของการอินเตอร์รัปต์ที่เกิดขึ้นจากไทม์เมอร์ 2 มีความสำคัญต่ำ

"1" ลำดับความสำคัญของการอินเตอร์รัปต์ที่เกิดขึ้นจากไทม์เมอร์ 2 มีความสำคัญสูง

PS (Serial port interrupt priority): ใช้ในการกำหนดลำดับความสำคัญของการอินเตอร์รัปต์ที่เกิดขึ้นจากพอร์ตอนุกรม (การรับหรือส่งข้อมูลเสร็จสิ้น) บิตนี้สามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น

"0" ลำดับความสำคัญของการอินเตอร์รัปต์ที่เกิดขึ้นจากพอร์ตอนุกรม มีความสำคัญต่ำ

"1" ลำดับความสำคัญของการอินเตอร์รัปต์ที่เกิดขึ้นจากพอร์ตอนุกรม มีความสำคัญสูง

PT1 (Timer1 interrupt priority): ใช้ในการกำหนดลำดับความสำคัญของการอินเตอร์รัปต์ที่เกิดขึ้นจากไทม์เมอร์ 1 (ไทม์เมอร์ 1 เกิดการนับขึ้นจนเกิดการโอเวอร์โฟลว์) บิตนี้สามารถเซตหรือเคลียร์ได้โดยการกระบวนการทางซอฟต์แวร์เท่านั้น

"0" ลำดับความสำคัญของการอินเตอร์รัปต์ที่เกิดขึ้นจากไทม์เมอร์ 1 มีความสำคัญต่ำ

"1" ลำดับความสำคัญของการอินเตอร์รัปต์ที่เกิดขึ้นจากไทม์เมอร์ 1 มีความสำคัญสูง

PX1 (External interrupt1 priority): ใช้ในการกำหนดลำดับความสำคัญของการอินเตอร์รัปต์จากภายนอกหมายเลข 1 (มีสัญญาณที่เป็นลอจิก "0" หรือสัญญาณขอบขาลงเข้ามาที่ขา INT1) บิตนี้สามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น

"0" ลำดับความสำคัญของการอินเตอร์รัปต์จากภายนอกหมายเลข 1 มีความสำคัญต่ำ

"1" ลำดับความสำคัญของการอินเตอร์รัปต์จากภายนอกหมายเลข 1 มีความสำคัญสูง

PT0 (Timer0 interrupt priority): ใช้ในการลำดับความสำคัญของการอินเตอร์รัปต์ที่เกิดขึ้นจากไทม์เมอร์ 0 (ไทม์เมอร์ 0 เกิดการนับขึ้นจนเกิดการโอเวอร์โฟลว์) บิตนี้สามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"0" ลำดับความสำคัญของการอินเตอร์รัปต์ที่เกิดขึ้นจากไทม์เมอร์ 0 มีความสำคัญต่ำ

"1" ลำดับความสำคัญของการอินเตอร์รัปต์ที่เกิดขึ้นจากไทม์เมอร์ 0 มีความสำคัญสูง

PX0 (External interrupt0 priority): ใช้ในการกำหนดลำดับความสำคัญของการอินเตอร์รัปต์จากภายนอกหมายเลข 0 (มีสัญญาณที่เป็นลอจิก "0" หรือสัญญาณขอบขาลงเข้ามาที่ขา INT0) บิตนี้สามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น

"0" ลำดับความสำคัญของการอินเตอร์รัปต์จากภายนอกหมายเลข 0 มีความสำคัญต่ำ

"1" ลำดับความสำคัญของการอินเตอร์รัปต์จากภายนอกหมายเลข 0 มีความสำคัญสูง

(ค) รีจิสเตอร์ควบคุมการทำงานของไทม์เมอร์ 0 และ 1 (Timer/Counter control register: TCON)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

รูปที่ 2.7.3 บิดต่างๆ ของรีจิสเตอร์ TCON

รีจิสเตอร์ TCON เป็นรีจิสเตอร์ฟังก์ชันพิเศษ มีขนาด 8 บิตมีแอดเดรส 0×88 สามารถเข้าถึงในระดับบิตได้ บิต 6 หรือ บิต 8 ใช้ในการควบคุมการปิด/เปิดการทำงานของไทม์เมอร์ 0 และไทม์เมอร์ 0 ส่วนบิตที่เหลือใช้สำหรับกระบวนการอินเตอร์รัปต์ รายละเอียดของแต่ละบิตดังต่อไปนี้

TF1 (Timer1 overflow flag): ใช้ในการบอกสถานะของการอินเตอร์รัปต์ที่เกิดขึ้นจากไทม์เมอร์ 1 บิตนี้จะเซตเป็น "1" เมื่อไทม์เมอร์ 1 นับขึ้นจนเกิดการโอเวอร์โฟลว์ นอกจากนี้ยังสามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์

"0" ไทม์เมอร์ 1 ยังไม่เกิดการโอเวอร์โฟลว์

"1" ไทม์เมอร์ 1 เกิดการโอเวอร์โฟลว์ ถ้ามีการอีนาเบิ้ลการอินเตอร์รัปต์ของไทม์เมอร์ 1 เอาไว้จะมีการอินเตอร์รัปต์จากไทม์เมอร์ 1 เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TF0 (Timer0 overflow flag): ใช้ในการบอกสถานะของการอินเตอร์รัปต์ที่เกิดขึ้นจากไทม์เมอร์ 0 บิตนี้จะเซตเป็น "1" เมื่อไทม์เมอร์ 0 นับขึ้นจนเกิดการโอเวอร์โฟลว์ นอกจากนี้ยังสามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์

"0" ไทม์เมอร์ 0 ยังไม่เกิดการโอเวอร์โฟลว์

"1" ไทม์เมอร์ 0 เกิดการโอเวอร์โฟลว์ ถ้ามีการอีนาเบิ้ลการอินเตอร์รัปต์ของไทม์เมอร์1 เอาไว้จะมีการอินเตอร์รัปต์จากไทม์เมอร์ 0 เกิดขึ้น

IE1 (External interrupt1 edge flag): ใช้ในการบอกสถานะของการอินเตอร์รัปต์ที่เกิดขึ้นจากการอินเตอร์รัปต์จากภายนอกหมายเลข 1 บิตนี้จะเซตเป็น "1" เมื่อมีสัญญาณลอจิก "0" หรือสัญญาณขอบขาลงเข้ามาที่ขา INT1 (P3.3) นอกจากนี้ยังสามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์

"0" ยังไม่มีสัญญาณอินเตอร์รัปต์เข้ามาที่ขา INT1

"1" มีสัญญาณอินเตอร์รับเข้ามาที่ขา INT1 ถ้ามีการอีนาเบิ้ลการอินเตอร์รัปต์เอาไว้จะมีการอินเตอร์รัปต์จากภายนอกหมายเลข 1 เกิดขึ้น

IT1 (Intertup1 type control): ใช้ในการกำหนดรูปแบบของการอินเตอร์รัปต์จากภายนอกหมายเลข 1 ว่าจะให้มีการอินเตอร์รัปต์เกิดขึ้นเมื่อมีสัญญาณลอจิก "0" หรือมีสัญญาณขอบขาลงเข้ามาที่ขา INT0 (P3.3) บิตนี้สามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น

"0" ให้การอินเตอร์รัปต์เกิดขึ้นเมื่อมีสัญญาณลอจิก "0" เข้ามาที่ขา INT1 (P3.3)

"1" ให้การอินเตอร์รัปต์เกิดขึ้นเมื่อมีสัญญาณขอบขาลงเข้ามาที่ขา INT1 (P3.3)

IE0 (External interrupt0 edge flag): ใช้ในการบอกสถานะของการอินเตอร์รัปต์ที่เกิดขึ้นจากการอินเตอร์รัปต์จากภายนอกหมายเลข 0 บิตนี้จะเซตเป็น "1" เมื่อมีสัญญาณลอจิก "0" หรือสัญญาณขอบขาลงเข้ามาที่ขา INT1 (P3.2) นอกจากนี้ยังสามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์

"0" ยังไม่มีสัญญาณอินเตอร์รัปต์เข้ามาที่ขา INT0

"1" มีสัญญาณอินเตอร์รับเข้ามาที่ขา INT0 ถ้ามีการอินาเบิ้ลการอินเตอร์รัปต์เอาไว้จะมีการอินเตอร์รัปต์จากภายนอกหมายเลข 0 เกิดขึ้น

IT0 (Intertup0 type control): ใช้ในการกำหนดรูปแบบของการอินเตอร์รัปต์จากภายนอกหมายเลข 0 ว่าจะให้มีการอินเตอร์รัปต์เกิดขึ้นเมื่อมีสัญญาณลอจิก "0" หรือมีสัญญาณขอบขาลงเข้ามาที่ขา INT0 (P3.2) บิตนี้สามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น

"0" ให้การอินเตอร์รัปต์เกิดขึ้นเมื่อมีสัญญาณลอจิก "0" เข้ามาที่ขา INT1 (P3.2)

"1" ให้การอินเตอร์รัปต์เกิดขึ้นเมื่อมีสัญญาณขอบขาลงเข้ามาที่ขา INT1 (P3.2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ง) รีจิสเตอร์ควบคุมการทำงานของไทม์เมอร์ 2 (Timer/Counter 2 control register: T2CON)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2'	CP/RL2'

รูปที่ 2.7.4 บิตต่างๆ ของรีจิสเตอร์ T2CON

รีจิสเตอร์ T2CON เป็นรีจิสเตอร์ฟังก์ชันพิเศษ มีขนาด 8 บิตอยู่ที่แอดเดรส 0×CB สามารถเข้าถึงในระดับบิตได้ บิตที่ใช้สำหรับกระบวนการอินเตอร์รัปต์มีบิตเดียวคือบิต 7 รายละเอียดมีดังต่อไปนี้

TF2 (Timer2 overflow flag): ใช้ในการบอกสถานะของการอินเตอร์รัปต์ที่เกิดขึ้นจากไทม์เมอร์ 2 บิตนี้จะเซตเป็น "1" เมื่อไทม์เมอร์ 2 นับขึ้นจนเกิดการโอเวอร์โฟลว์ หรือมีการแคปเจอร์เกิดขึ้นนอกจากนี้ยังสามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์

"0" ไทม์เมอร์ 2 ยังไม่มีการเกิดการโอเวอร์โฟลว์ และไม่มีการแคปเจอร์เกิดขึ้น

"1" ไทม์เมอร์ 2 มีการเกิดการโอเวอร์โฟลว์ หรือมีการแคปเจอร์เกิดขึ้น ถ้ามีการอีนาเบิ้ลการอินเตอร์รัปต์ของไทม์เมอร์ 2 เอาไว้จะมีการอินเตอร์รัปต์จากไทม์เมอร์ 2 เกิดขึ้น

2.7.2 การอินเตอร์รัปต์จากภายนอก

การอินเตอร์รัปต์จากภายนอกจะเกิดขึ้นเมื่อได้รับจากสัญญาณอินเตอร์รัปต์ที่ส่งเข้ามาจากภายนอกผ่านทางขารับสัญญาณอินเตอร์รัปต์ การอินเตอร์รัปต์จากถายนอกสามารถเกิดขึ้นได้ 2 แหล่งกำเนิด คืออินเตอร์รัปต์จากภายนอกหมายเลข 0 (External Interrupt 0: INT0) และอินเตอร์รัปต์จากภายนอกหมายเลข 1 (External Interrupt 1: INT1) รายละเอียดของการอินเตอร์รัปต์จากภายนอกทั้งสองแหล่งมีดังนี้

การเกิดอินเตอรัปต์จากภายนอกหมายเลข 0 (External Interrupt 0: INT0)

การอินเตอร์รัปต์จากภายนอกหมายเลข 0 จะเกิดขึ้นเมื่อมีสัญญาณเข้ามาที่ขา INT0 (P3.2) การอินเดอร์รัปต์ตัวนี้จะเกิดขึ้นได้ 2 ลักษณะคือ เมื่อขา INT0 ได้รับสัญญาณลอจิก "0" และเมื่อขา INT0 ได้รับสัญญาณขอบขาลง อินเตอร์รัปต์จากภายนอกหมายเลข 0 มีเว็กเตอร์ของการอินเตอร์รัปต์ต่ออยู่ที่หมายเลข 0 (interrupt 0)

การเกิดอินเตอรัปต์จากภายนอกหมายเลข 1 (External Interrupt 1: INT0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอินเตอร์รัปต์จากภายนอกหมายเลข 1 จะเกิดขึ้นเมื่อมีสัญญาณเข้ามาที่ขา INT0 (P3.3) การอินเตอร์รัปต์ตัวนี้จะเกิดขึ้นได้ 2 ลักษณะคือ เมื่อขา INT1 ได้รับสัญญาณลอจิก "0" และเมื่อขา INT1 ได้รับสัญญาณขอบขาลง อินเตอร์รัปต์จากภายนอกหมายเลข 1 มีเว็กเตอร์ของการอินเตอร์รัปต์ต่ออยู่ที่หมายเลข 2 (Interrupt 2)

การเลือกรูปแบบของการเกิดอินเตอร์รัปต์จากภายนอก (INTO & INT1)

ดังที่ได้กล่าวมาแล้วจากข้างต้นมา การอินเตอร์รัปต์จากภายนอกสามารถเกิดขึ้นได้ 2 รูปแบบ คือ จะเกิดการอินเตอร์รัปต์เกิดขึ้นเมื่อมีสัญญาณลอจิก "0" เข้ามาที่ขา INTx และมีสัญญาณขอบขาลงเข้ามาที่ขา INTx การเลือกรูปแบบของการอินเตอร์รัปต์ดังกล่าวสามารถทำได้โดยการกำหนดที่ ITx ดังนี้

***IT0 = "0"** ให้การอินเตอร์รัปต์หมายเลข 0 เกิดขึ้นเมื่อมีสัญญาณลอจิก "0" เข้ามาที่ขา INT0

***IT0 = "1"** ให้การอินเตอร์รัปต์หมายเลข 0 เกิดขึ้นเมื่อมีสัญญาณขอบขาลงเข้ามาที่ขา INT0

***IT1 = "0"** ให้การอินเตอร์รัปต์หมายเลข 1 เกิดขึ้นเมื่อมีสัญญาณลอจิก "0" เข้ามาที่ขา INT1

***IT1 = "1"** ให้การอินเตอร์รัปต์หมายเลข 1 เกิดขึ้นเมื่อมีสัญญาณขอบขาลงเข้ามาที่ขา INT1

2.7.3 การอินเตอร์รัปต์จากภายใน

การอินเตอร์รัปต์จากภายในจะเกิดขึ้นจากสัญญาณอินเตอร์รัปต์ของวงจรภายในที่มีอยู่ในตัวไมโครคอนโทลเลอร์ เช่น ไทม์เมอร์ และพอร์ตอนุกรม เป็นต้น

การอินเตอร์รัปต์จากไทม์เมอร์ 0

ไทม์เมอร์ 0 สามารถสร้างสัญญาณอินเตอร์รัปต์ให้เกิดขึ้นได้เมื่อมีการนับขึ้นจนเกิดการโอเวอร์โฟลว์ เมื่อไทม์เมอร์ 0 นับขึ้นจนเกิดการโอเวอร์โฟลว์ จะมีผลทำให้บิต TF0 ซึ่งเป็นบิตที่ 5 ของรีจิสเตอร์ TCON เกิดการเซตเป็น "1" และในเวลาเดียวกันนี้เองหากมีการอีนาเบิ้ลการอินเตอร์รัปต์ (EA=1; ET0=1;) CPU จะกระโดดไปประมวลผลยังโปรแกรมบริการ การอินเตอร์รัปต์ของไทม์เมอร์ 0 โดยการอินเตอร์รัปต์จากไทม์เมอร์ 0 มีเว็กเตอร์ของการอินเตอร์รัปต์อยู่ที่หมายเลข 1 (interrupt 1) *การเกิดโอเวอร์โฟลว์ของไทม์เมอร์ จะเกิดขึ้นในจังหวะของการเปลี่ยนแปลงจากค่าสูงสุดไปยังค่า 0 หรือเกิดขึ้นในจังหวะของการเริ่มนับในรอบใหม่นั้นเอง*

การอินเตอร์รัปต์จากไทม์เมอร์ 1

การอินเตอร์รัปต์จากไทม์เมอร์ 1 ไม่ได้มีความแตกต่างไปจากการอินเตอร์รัปต์จากไทม์เมอร์ 0 นั้น คือ ไทม์เมอร์ 1 สามารถสร้างสัญญาณอินเตอร์รัปต์ให้เกิดขึ้นได้เมื่อมีการนับขึ้นจนเกิดการโอเวอร์โฟลว์ เมื่อไทม์เมอร์ 1 นับขึ้นจนเกิดการโอเวอร์โฟลว์ จะมีผลทำให้บิต TF1 ซึ่งเป็นบิตที่ 7 ของรีจิสเตอร์ TCON เกิดการเซตเป็น "1" และในเวลาเดียวกันนี้เองหากมีการอีนาเบิ้ลการอินเตอร์รัปต์ไว้ (EA=1; ET=1;) CPU จะกระโดดไปประมวลผลยังโปรแกรมบริการ การอินเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รัปต์ของไทม์เมอร์ 1 โดยการอินเตอร์รัปต์จากไทม์เมอร์ 1 มีเว็กเตอร์ของการอินเตอร์รัปต์อยู่ที่หมายเลข 3 (Interrupt 3)

การอินเตอร์รัปต์จากไทม์เมอร์ 2

ไทม์เมอร์ 2 เป็นไทม์เมอร์ที่มีความสามารถมากกว่า 0 และ 1 อย่างเห็นได้ชัด เช่น สามารถทำงานในโหมด 16 บิต รีโหลดค่าอัตโนมัติ (16 bit auto-reload) และโหมดแคปเจอร์ (Capture) ได้ ด้วยความสามารถดังกล่าวจำทำให้ไทม์เมอร์ 2 สามารถสร้างสัญญาณอินเตอร์รัปต์ของไทม์เมอร์ได้ 2 แหล่งกำเนิดนั้น คือการอินเตอร์รัปต์อันเนื่องมาจากเกิดการโอเวอร์โฟลว์ และการอินเดอร์รัปต์อันเนื่องมาจากการแคปเจอร์การเกิดการอินเตอร์รัปต์ทั้ง 2 กรณีสามารถแยกอธิบายได้ดังต่อไปนี้

การอินเตอร์รัปต์จากไทม์เมอร์ 2 เมื่อเกิดการโอเวอร์โฟลว์

การอินเตอร์รัปต์จากการโอเวอร์โฟลว์ของเมื่อไมเทอร์ 2 ไม่ได้มีความแตกต่างจากการอินเตอร์รัปต์จากไทม์เมอร์ 0 และ 1 คือเมื่อไมเทอร์ 2 นับขึ้นจนเกิดการโอเวอร์โฟลว์จะทำให้บิต TF2 ซึ่งเป็นบิตที่ 7 ในรีจิสเตอร์ T2CON เซตเป็น "1" และในเวลาเดียวกันนี้เองหากมีการอีนาเบิ้ลการอินเตอร์รัปต์ไว้ (EA=1; ET=1;) CPU จะกระโดดไปประมวลผลยังโปรแกรมบริการ การอินเตอร์รัปต์ของไทม์เมอร์ 2

การอินเตอร์รัปต์จากไทม์เมอร์ 2 เมื่อมีการแคปเจอร์

การอินเตอร์รัปต์จากไทม์เมอร์ 2 เมื่อมีการแคปเจอร์ เป็นอินเตอร์รัปต์ที่เกิดขึ้นอันเนื่องมาจากไมโครคอนโทรลเลอร์ตรวจจับสัญญาณขอบขาลง (เปลี่ยนจาก "1" เป็น "0") ที่ขา T2EX (P1.1) ได้ การตรวจจับสัญญาณนี้ได้จะมีผลทำให้บิต EXF2 ซึ่งเป็นบิตที่ 6 ในรีจิสเตอร์ T2CON เซตเป็น "1" และในเวลาเดียวกันเองหากมีการอีนาเบิ้ลการอินเตอร์รัปต์ไว้ (EA=1; ET=1; EXEN2=1;) CPU จะกระโดดไปประมวลผลยังโปรแกรมบริการ การอินเตอร์รัปต์ของไทม์เมอร์ 2 โดยการอินเตอร์รัปต์จากไทม์เมอร์ 2 มีเว็กเตอร์ของการอินเตอร์รัปต์อยู่ที่หมายเลข 5 (interrupt 5)

การอินเตอร์รัปต์ที่เกิดจากไทม์เมอร์ 2 ทั้ง 2 กรณีจะเว็กเตอร์ในการอินเตอร์รัปต์ตัวเดียวกันคือเว็กเตอร์อินเตอร์รัปต์หมายเลขที่ 5 (Interrupt 5) ดังนั้นในการใช้งานจะต้องใช้บิต EXEN2 เป็นตัวเลือกว่าจะให้เกิดการอินเตอร์รัปต์จากกรณีใด ดังนี้

EXEN2 = "0" ตอบสนองการอินเตอร์รัปต์จากไทม์เมอร์ 2 เมื่อมีการนับขึ้นจนเกิดโอเวอร์โฟลว์

EXEN2 = "1" ตอบสนองการอินเตอร์รัปต์จากไทม์เมอร์ 2 เมื่อมีสัญญาณของขาลงเข้ามาที่ขา T2EX (P1.1) หรือมีการแคปเจอร์เกิดขึ้นนั้นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

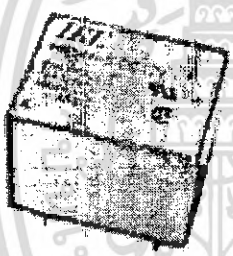
2.8 รีเลย์ (Relay)

เป็นอุปกรณ์ทำหน้าที่เป็นตัดต่อวงจรแบบเดียวกับสวิตช์ มีหลักการทำงานคล้ายกับ ขดลวดแม่เหล็กไฟฟ้าหรือโซลินอยด์ (Solenoid) รีเลย์ใช้ในการควบคุมวงจรไฟฟ้าได้อย่างหลากหลาย รีเลย์เป็นสวิตช์ควบคุมที่ทำงานด้วยไฟฟ้า แบ่งออกตามลักษณะการใช้งานได้เป็น 2 ประเภทคือ

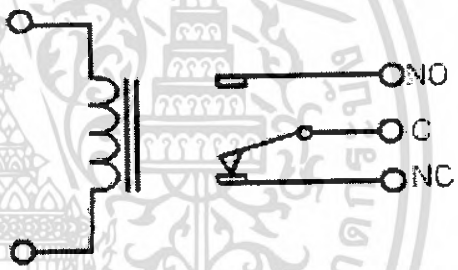
1. รีเลย์กำลัง (Power relay) หรือมักเรียกกันว่าคอนแทกเตอร์ (Contactor or Magnetic contactor) ใช้ในการควบคุมไฟฟ้ากำลัง มีขนาดใหญ่กว่ารีเลย์ธรรมดา

2. รีเลย์ควบคุม (Control Relay) มีขนาดเล็กกำลังไฟฟ้าต่ำ ใช้ในวงจรควบคุมทั่วไปที่มีกำลังไฟฟ้าไม่มากนัก หรือเพื่อการควบคุมรีเลย์หรือคอนแทกเตอร์ขนาดใหญ่ รีเลย์ควบคุมบางทีเรียกกันง่าย ๆ ว่า "รีเลย์"

สำหรับโครงงานนี้เลือกใช้รีเลย์ควบคุมเท่านั้นเพราะใช้ในการควบคุมกำลังไฟฟ้าต่ำ เราสามารถพบเห็นการใช้งานรีเลย์ในการตัดต่อวงจรทั่วไป โดยที่ตัวมันนั้นจะถูกควบคุมด้วยไฟฟ้าและหน้าสัมผัสของรีเลย์เองยังสามารถเลือก การทำงานได้หลายรูปแบบ



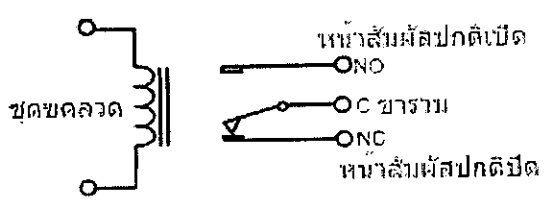
รูปที่ 2.8.1 รีเลย์ที่ใช้ในงานอิเล็กทรอนิกส์ [7]



รูปที่ 2.8.2 สัญลักษณ์ของรีเลย์ [7]

2.8.1 โครงสร้างของรีเลย์

ภายในโครงสร้างของ รีเลย์ จะประกอบไปด้วยขดลวด 1 ชุด และ หน้าสัมผัสซึ่งในหน้าสัมผัส 1 ชุด ซึ่งจะประกอบไปด้วย หน้าสัมผัสแบบปกติปิด (Normally Close หรือ NC.) ซึ่งในสภาวะปกติ ขานี้จะต่ออยู่กับขาร่วม (C) และ หน้าสัมผัสแบบปกติเปิด (Normally Open หรือ NO.) ขานี้จะต่อเข้ากับขาร่วม (C) เมื่อขดลวดมีแรงดันดกคร่อม หรือกระแสไหลผ่าน (ในปริมาณที่เพียงพอ) ใน รีเลย์ 1 ตัว อาจมีหน้าสัมผัสมากกว่า 1 ชุด ซึ่งขึ้นอยู่กับผู้ผลิต

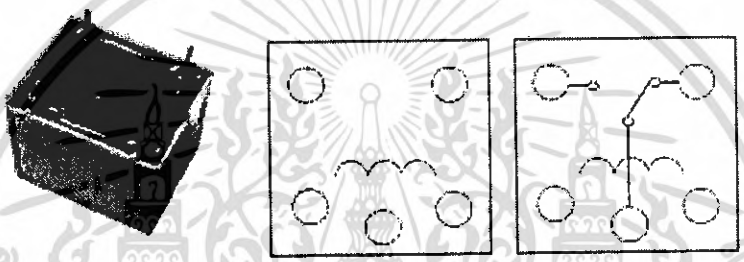


รูปที่ 2.8.3 สัญลักษณ์ของรีเลย์แทนโครงสร้างรีเลย์ [7]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2 หลักการทำงานของรีเลย์

รีเลย์จะทำงานตามหลักการแม่เหล็กไฟฟ้าเมื่อเรานำเอาขดลวดพันรอบแกนเหล็กหลายรอบแล้วป้อนกระแสไฟฟ้าเข้าในขดลวดนั้น แกนเหล็กจะกลายเป็นแม่เหล็ก(แต่จะเป็นแบบชั่วคราวเท่านั้น)และเมื่อเรานำไฟฟ้าออกแกนเหล็กจะกลายเป็นแกนเหล็กธรรมดา เมื่อรีเลย์อยู่ในสภาวะปกติยังไม่มีการจ่ายกระแสให้รีเลย์ หน้าสัมผัส NC กับ C จะต่อถึงกัน ทำให้กระแสไฟฟ้าไหลผ่านไปได้ และเมื่อจ่ายกระแสไฟฟ้าให้รีเลย์ ทำให้ชุดขดลวดเกิดเป็นแม่เหล็ก อำนาจแม่เหล็กจะดึงหน้าสัมผัส C มาต่อกับหน้าสัมผัส NO ทำให้กระแสไฟฟ้าไหลจาก NO ไปยัง C ได้ และ เมื่อเราเอากระแสไฟฟ้าออกจากรีเลย์ หน้าสัมผัส C จะถูกสปริงดึงไปให้ติดกับหน้าสัมผัส NC ดังเดิม



รูปที่ 2.8.4 ตัวอย่างตำแหน่งขาของรีเลย์แบบ 5 ขา [7]

ตำแหน่งของขาของรีเลย์นั้นอาจแตกต่างกันไปบ้างตามบริษัทผู้ผลิต และที่ตำแหน่งขาจะมีสัญลักษณ์บอกตำแหน่งขานั้นด้วยเช่นกัน

2.9 ตัวต้านทานไวแสง (LDR)

รูปที่ 2.9 ตัวต้านทานไวแสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในอุปกรณ์อิเล็กทรอนิกส์ประเภทไวต่อแสง หรือ เปลี่ยนแปลงการทำงานของตัวมันเอง ตามปริมาณของแสงมีอยู่หลายอย่าง ตั้งแต่ LDR (light dependent resistor) โฟโตโวลตาอิกเซล (photovoltaic cell) ซึ่งจ่ายแรงดันออกมาได้เมื่อได้รับแสง, โฟโต้ไดโอด (photodiode) โฟโต้ทรานซิสเตอร์ (phototransistor) ไปจนถึงเอสซีอาร์ ที่ทำงานด้วยแสง (LASCR - light activated silicon controlled rectifier) ซึ่งใช้หลักการของสารกึ่งตัวนำทั้งนั้น อุปกรณ์ประเภทนี้ที่มีโครงสร้างและลักษณะการทำงานง่ายที่สุดก็เห็นจะได้แก่ LDR เพราะไม่ได้ใช้หลักการของรอยต่อ พี - เอ็น เหมือนกันแบบอื่นๆ ที่ได้กล่าวมา

2.9.1 โครงสร้างของ LDR

ตัว LDR มีเรียกกันอีกหลายชื่อ เช่น โฟโต้คอนดักตีฟเซล (photoconductive cell) หรือ ตัวต้านทานไวแสง (LSR - light sensitive resistor) ส่วนใหญ่จะทำด้วยสารแคดเมียมซัลไฟต์ (CdS) หรือไม่ก็แคดเมียมซีนิไนด์ (CdSe) ซึ่งทั้งสองตัวนี้ก็เป็นสารประเภทกึ่งตัวนำเอามาฉาบลงบนแผ่นเซรามิกที่ใช้เป็นฐานรองแล้วต่อขาจากสารที่ฉาบ ไว้ออกมา

โครงสร้างของ LDR ส่วนที่ขดเป็นแนวเล็กๆ สีดำจะทำหน้าที่เป็นตัวต้านทานไวแสงและแนวสีดำนั้นจะแบ่งพื้นที่ของตัวมันออกเป็น 2 ข้าง ซึ่งถ้าดูของจริงจะเห็นว่าออกสีทองนั้นเป็นตัวนำไฟฟ้า ทำหน้าที่สัมผัสกับตัวต้านทานไวแสงเป็นที่สำหรับต่อขาออกมาภายนอกหรือเรียกว่าอิเล็กโทรด ที่เหลือก็จะเป็นฐานเซรามิกและอุปกรณ์สำหรับห่อหุ้มมัน ซึ่งมีได้หลายแบบ

2.9.2 สมบัติทางแสง

LDR เป็นสารประเภทกึ่งตัวนำ ดังนั้นจะทำงานเมื่อมีแสงมาตกกระทบลง LDR แสงจะถ่ายทอดพลังงานให้กับสารแคดเมียมซัลไฟด์ที่ฉาบบริเวณทางเข้าแสง ทำให้เกิดโฮลกับอิเล็กตรอนวิ่งชนกัน การที่มีโฮลกับอิเล็กตรอนอิสระนี้มากก็เท่ากับความต้านทานลดลงนั่นเอง ยิ่งความเข้มของแสงที่ตกกระทบมากเท่าไร ความต้านทานก็ยิ่งลดลงมากเท่านั้น

ในส่วนที่ว่าแสงตกกระทบนั้นมิใช่ว่าจะเป็นแสงอะไรก็ได้ เฉพาะแสงในช่วงความยาวคลื่นประมาณ 4,000 อังสตรอม (1 อังสตรอม เท่ากับ 10^{-10} เมตร) ถึงประมาณ 10,000 อังสตรอม เท่านั้นที่จะใช้ได้ (สายตาคนจะเห็นได้ ในช่วงประมาณ 4,000 อังสตรอม ถึง 7,000 อังสตรอม) ซึ่งคิดแล้วก็เป็นช่วงคลื่นเพียงแคบ ๆ เมื่อเทียบกับการทำงานของอุปกรณ์ไวแสง ประเภทอื่น ๆ แต่ถึงอย่างไรแสงในช่วงคลื่นนี้ ก็มีอยู่ในแสงอาทิตย์ แสงจากหลอดไฟแบบไส้และแสงจากหลอดฟลูออเรสเซนต์ด้วย หรือถ้าจะคิดถึงความยาวคลื่นที่ LDR จะตอบสนองไวที่สุด ก็มีอยู่หลายความยาวคลื่น โดยทั่วไป LDR ที่ทำจากแคดเมียมซัลไฟด์ จะไวต่อแสงที่มีความยาวคลื่นในช่วง 5,000 กว่าอังสตรอม. ซึ่งเราจะเห็นเป็นสีเขียว ไปจนถึงสีเหลือง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 PIR เซ็นเซอร์

เซ็นเซอร์เป็นอุปกรณ์แปลงสัญญาณสำหรับเครื่องมือวัดหนึ่ง ๆ ทำหน้าที่เปลี่ยนปริมาณทางฟิสิกส์ เช่น การสั่นสะเทือน หรือปริมาณทางเคมี เช่น ปฏิกิริยาเคมีต่าง ๆ ให้เป็นปริมาณทางไฟฟ้า ซึ่งสามารถแบ่งออกได้หลายชนิด เช่น เซ็นเซอร์ทางความร้อน เซ็นเซอร์ทางกลศาสตร์ เซ็นเซอร์ทางเสียง เป็นต้น

โครงงานนี้ได้ศึกษาเซ็นเซอร์ทางการเคลื่อนไหว (Motion Sensor) คือ PIR เซ็นเซอร์ (Passive Infra Red) สามารถตรวจจับการเคลื่อนไหวของมนุษย์ได้ สามารถนำไปประยุกต์ใช้งานได้หลายรูปแบบ เช่น สามารถนำไปดักจับขโมยหรือผู้บุกรุกที่เข้ามาในบริเวณบ้าน หรือเขตหวงห้าม โดยที่ไม่ต้องมีใครเฝ้าอยู่ตลอด หรือจะนำไปใช้เป็นตัวตรวจจับคนเดินเข้าออกผ่านประตู เพื่อให้ประตูเปิด-ปิดอัตโนมัติ โดยต่อเอาต์พุตเข้ากับตัวรีเลย์เพื่อไปขับให้กลไกที่ขับเคลื่อนประตูให้ทำงานและสามารถนำไปประยุกต์ใช้กับหุ่นยนต์

รูปที่ 2.10 PIR Sensor

หลักการทำงานของเซ็นเซอร์

PIR เป็นอุปกรณ์สำหรับตรวจจับความเคลื่อนไหวที่มีชื่อเต็มว่า ไพโรอิเล็กตริก เป็นอุปกรณ์จำพวก พาสซีฟอินฟราเรดดีเท็กเตอร์ (Passive Infrared detector – PIR) หรือตัวตรวจจับรังสีอินฟราเรดแบบหนึ่ง โดยตัวมันจะทำงาน เมื่อมันตรวจจับพบความเปลี่ยนแปลงของรังสี อินฟราเรดที่แผ่ออกมาดัวคนหรือสัตว์ ในขณะที่มีการเคลื่อนไหว ในตัวคนหรือสัตว์จะมีรังสีความร้อนแผ่ออกมารอบๆ ตัวในปริมาณที่แน่นอนอยู่จำนวนหนึ่ง เมื่อเกิดการเคลื่อนไหวหรือเคลื่อนที่ก็จะทำให้อุณหภูมิในบริเวณนั้นเกิดการเปลี่ยนแปลง ส่งผลให้คลื่นรังสีความร้อนที่ว่านี้แผ่กระจายออกมา มีความยาวคลื่นประมาณ 0.74-300 ไมโครเมตร อันเป็นแถบความถี่ในย่านอินฟราเรดพอดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายใน PIR ประกอบด้วยเลนส์ที่เรียกว่า ฟรีสเนลเลนส์ (fresnel lenses) ซึ่งเป็นเลนส์ที่มีขนาดเล็กจำนวน มากเพื่อสร้างแพตเทิร์นการแทรกสอด (Interfered) ของแสงย่านอินฟราเรด ขณะที่ยังไม่มีใครเข้ามาในรัศมีรูปแบบการแทรกสอดของ แสงนั้นจะมีแพตเทิร์นหยุดนิ่งคงที่ แต่เมื่อวัตถุนั้นมีการเคลื่อนไหวเกิดขึ้น แพตเทิร์นการแทรกสอดของคลื่นแสงที่ปรากฏบนตัวเซ็นเซอร์ PIR จะเปลี่ยนไปเป็นสัญญาณไฟฟ้า ตามการเคลื่อนไหวนั้นออกมาทางขาเอาต์พุต

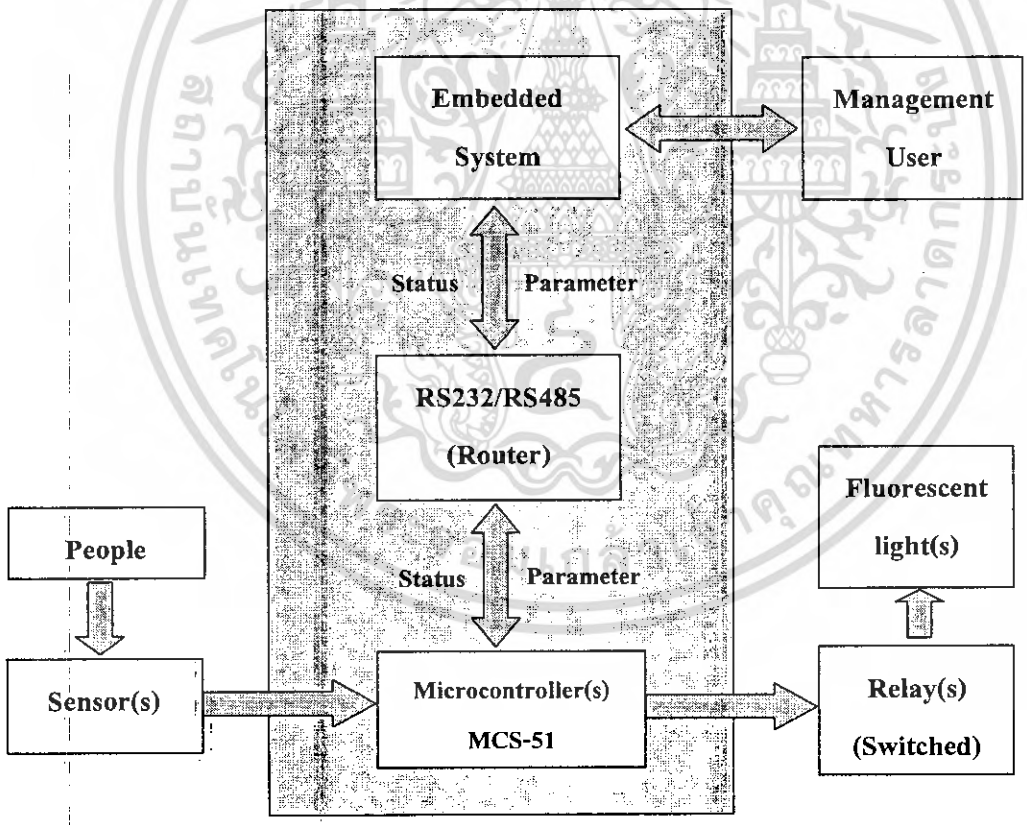
PIR Sensor ทำงานได้ในทั้งที่มืดและที่สว่าง สามารถหน่วงเวลาในการทำงานได้ถึง 30 วินาที แต่ไม่สามารถเจาะจงระยะเวลาการหน่วงเวลาได้ ซึ่งถือเป็นข้อเสียอย่างหนึ่ง ทำงานได้อย่างมีประสิทธิภาพในระยะ 3-4 เมตร ซึ่งเป็นข้อจำกัดของการตรวจจับความเคลื่อนไหว ข้อเสียคือไม่สามารถทำงานได้ทันทีเพราะจะต้องมีการ warm-up ก่อนประมาณ 30 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบโครงงาน

3.1 โครงสร้างการทำงานของระบบที่ออกแบบ

โครงสร้างของระบบนี้จะถูกประกอบขึ้นโดยจะมีส่วนของระบบจัดการอัจฉริยะหรือระบบฝังตัวเพื่อการประมวลผลในระดับสูงสุดและสามารถติดต่อกับผู้ใช้ได้โดยผ่านเครือข่ายอินเตอร์เน็ต ส่วนไมโครคอนโทรลเลอร์นั้นทำหน้าที่ควบคุมการปิด-เปิดหลอดไฟฟลูออเรสเซนต์ เมื่อได้รับคำสั่งจากการทำงานของเซ็นเซอร์ซึ่งคอยตรวจจับมนุษย์ในบริเวณที่ติดตั้งแล้ว ไมโครคอนโทรลเลอร์ยังสามารถเปลี่ยนแปลงระบบการควบคุมการทำงานจากผู้ใช้ที่ส่งจากระบบฝังตัวผ่านตัวแปลง RS232 เป็น RS485 ซึ่งทำหน้าที่เสมือนเป็นเราท์เตอร์เพื่อกระจายข้อมูลให้กับไมโครคอนโทรลเลอร์ตัวอื่นๆ โครงสร้างโดยรวมแสดงดังรูปที่ 3.1



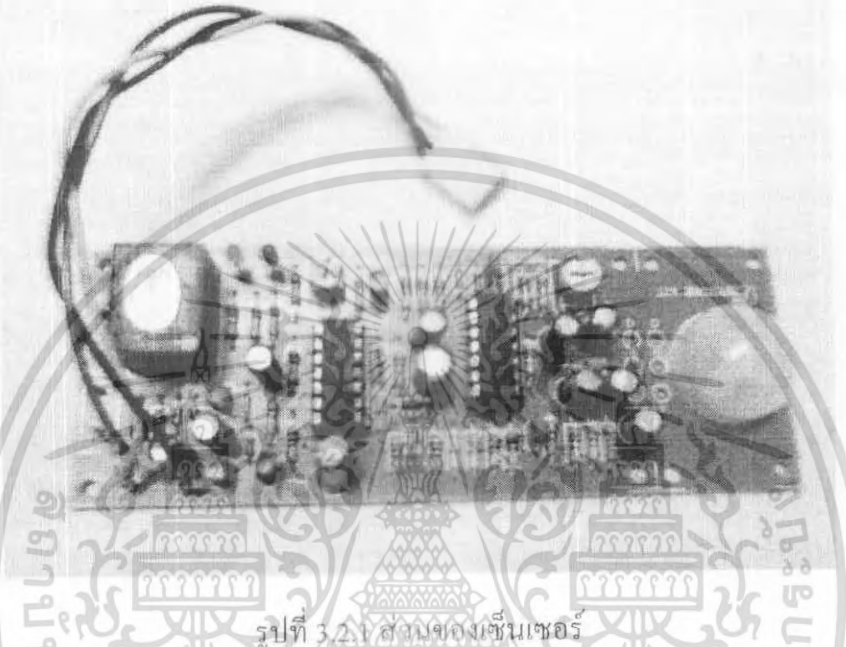
รูปที่ 3.1 โครงสร้างโดยรวมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

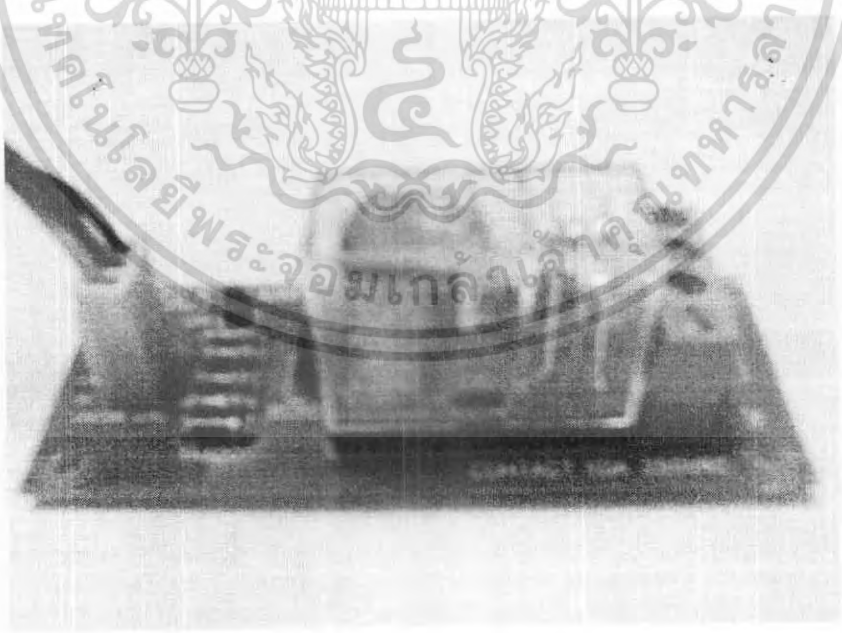
3.2 ส่วนประกอบของระบบ

3.2.1 ส่วนควบคุมการเปิด-ปิดหลอดไฟ

ไมโครคอนโทรลเลอร์เป็นส่วนที่ทำการควบคุมการปิด-เปิดหลอดไฟฟลูออเรสเซนต์ จะติดต่อกับการทำงาน 3 ส่วนด้วยกัน คือ ส่วนของเซ็นเซอร์ ส่วนของรีเลย์และส่วนของแอลดีอาร์

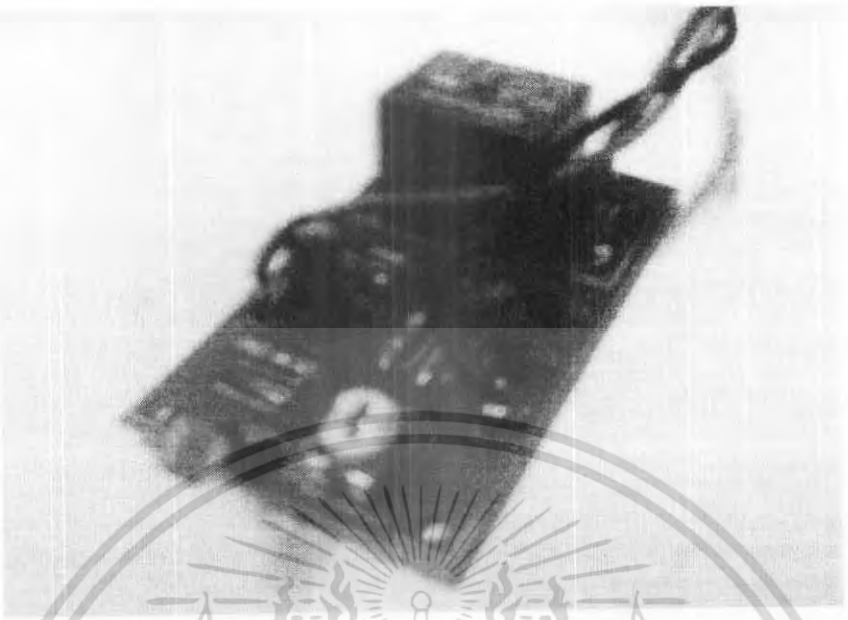


รูปที่ 3.2.1 ส่วนของเซ็นเซอร์



รูปที่ 3.2.2 ส่วนของรีเลย์

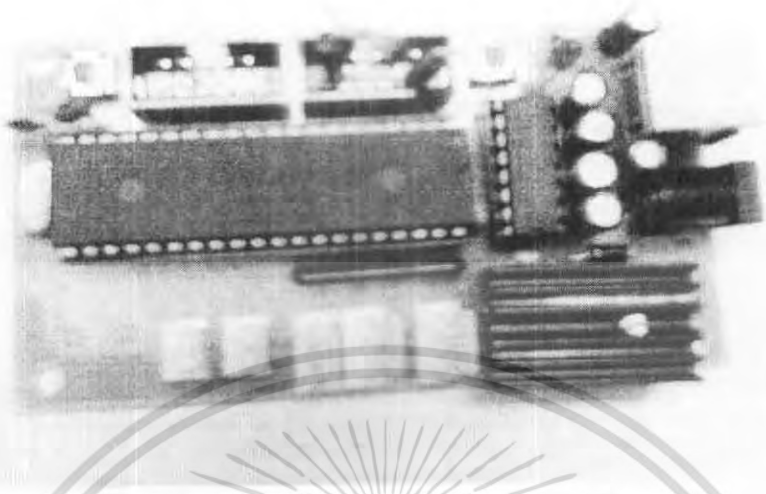
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2.3 ส่วนของแอลดีอาร์

ส่วนของเซ็นเซอร์ค่อยตรวจสอบว่ามีคนเข้ามาในพื้นที่ที่เซ็นเซอร์ตรวจจับอยู่หรือไม่ ส่วนของรีเลย์ทำงานก็ต่อเมื่อได้รับคำสั่งจากไมโครคอนโทรลเลอร์ให้เปิดไฟ และส่วนของแอลดีอาร์ทำหน้าที่ตรวจจับแสง ถ้าแอลดีอาร์ได้รับแสงที่มีความสว่างน้อยกว่าที่ตั้งไว้ในส่วนของวงจร รีเลย์ก็จะถูกสั่งให้เปิดการทำงานในทางตรงกันข้ามถ้าแอลดีอาร์ได้รับแสงที่มีความสว่างมากกว่าที่ตั้งไว้ รีเลย์ก็ไม่ทำงาน และโปรแกรมภาษาซีแล้วทำการเบิร์นลงบนไมโครคอนโทรลเลอร์เพื่อควบคุมการปิด-เปิดหลอดไฟ ในส่วนของวงจรนั้นไมโครคอนโทรลเลอร์ ต้องต่อขากับอิเลเมนต์ตามคุณสมบัติของไมโครคอนโทรลเลอร์ที่ใช้ให้ถูกต้อง มิฉะนั้นอาจเกิดความเสียหายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2.4 ส่วนของบอร์ดไมโครคอนโทรลเลอร์แต่ละชั้น

3.2.2 ตัวแปลงสัญญาณ RS232/RS485

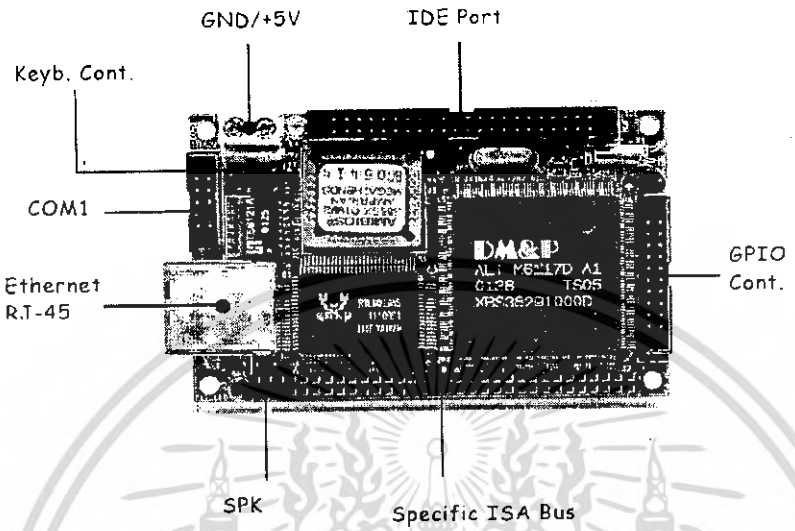
มีคุณสมบัติใช้เปลี่ยนสัญญาณจาก RS232 หรือสัญญาณจาก Serial Port เป็น RS485 คือ แปลงสัญญาณที่ติดต่อกับระบบฝังตัวให้สามารถสื่อสารกับไมโครคอนโทรลเลอร์ได้ มีอัตราการรับส่งข้อมูล 115.2 Kbps ความยาวของสายเชื่อมต่อได้สูงสุดถึง 1.2 กิโลเมตร ในที่นี้ใช้โหมดการทำงานแบบ Half Duplex การทำงานคือให้อุปกรณ์นี้ทำหน้าที่เป็นตัวแม่ (Master) สำหรับเป็นตัวกลางในการรับส่งข้อมูลเองทั้งหมด หรือเป็นเพียงการจัดลำดับการรับส่งให้กับไมโครคอนโทรลเลอร์ตัวอื่นๆ ส่วนตัวลูกหรือไมโครคอนโทรลเลอร์นั้น ในครั้งแรกต้องกำหนดให้เป็นการรับข้อมูลไว้ตลอดเวลา ตัวแม่จะทำหน้าที่สแกนตัวลูกทีละตัว เพื่อตรวจสอบว่ามีข้อมูลที่ต้องการส่งหรือไม่ ถ้ามีตัวแม่จะเปลี่ยนทิศทางมาเป็นรับข้อมูล และส่งข้อมูลกันต่อไป เมื่อส่งเสร็จเรียบร้อยแล้วตัวลูกต้องส่งคำสั่งบอกตัวแม่ให้ทราบว่า เสร็จแล้ว เพื่อที่ตัวแม่จะได้สแกนตัวอื่นต่อไป

3.2.3 ระบบฝังตัว

ระบบฝังตัวเป็นระบบที่เราสามารถเขียนโปรแกรมควบคุมการทำงานต่างๆ ได้ ดังเช่นโครงงานนี้เป็นระบบที่ใช้ในการควบคุมการทำงานของไมโครคอนโทรลเลอร์ในระดับสูงสุดและประมวลผลข้อมูลต่าง ๆ ความสามารถที่ระบบนี้ทำได้คือ ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ สามารถเปลี่ยนค่าพารามิเตอร์ของไมโครคอนโทรลเลอร์และตรวจสอบสภาวะของเซ็นเซอร์ผ่านทางเครือข่ายอินเตอร์เน็ตได้อีกด้วย ข้อดีของระบบนี้คือ ระบบมีการทำงานเสมือนคอมพิวเตอร์ขนาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เล็ก สามารถต่อเชื่อมกับพอร์ตต่างๆ ได้มากมาย นิยมใช้ในงานที่มีการประมวลผลไม่ใหญ่มากนัก ทำให้สะดวกในการต่อใช้งาน



รูปที่ 3.2.5 ระบบฝังตัว รุ่น Mity-mite module

3.2.4 การติดต่อกับผู้ใช้

ผู้ใช้สามารถควมคุมการทำงานของระบบได้โดยผ่านเครือข่ายอินเตอร์เน็ต เนื่องจากตัวระบบฝังตัวได้ถูกโปรแกรมให้เป็นเว็บเซิร์ฟเวอร์ ซึ่งสามารถเปลี่ยนแปลงการทำงานได้ดังนี้

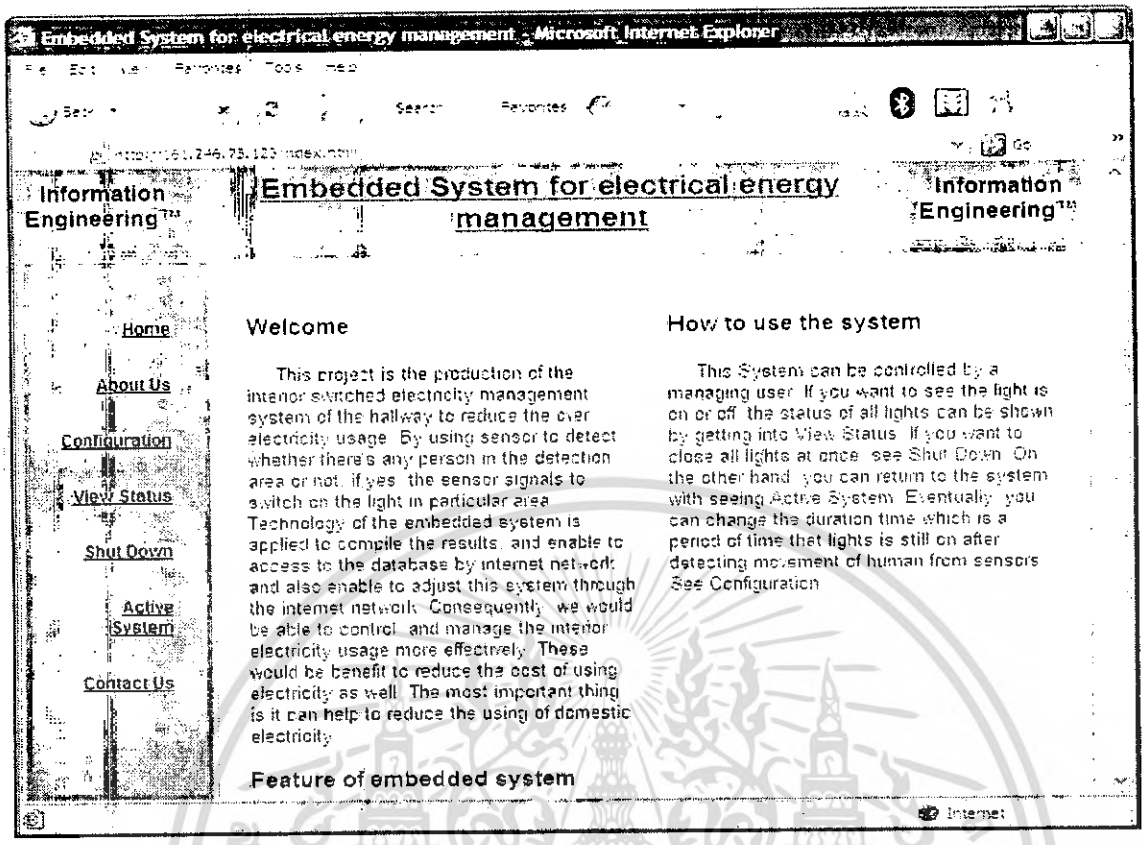
1. สามารถเปลี่ยนแปลงช่วงเวลาให้หลอดไฟติดได้นานตามที่ต้องการ หลังจากที่เซ็นเซอร์ตรวจจับได้ว่ามีมนุษย์เคลื่อนไหวอยู่ภายในพื้นที่ที่ตรวจจับได้

2. สามารถแสดงสถานะการทำงานของหลอดไฟและแอลดีอาร์ได้

3. สามารถตั้งเวลาการเปิด-ปิดของระบบได้

นอกจากนี้เว็บไซต์ยังบอกรายละเอียดของโปรเจค จุดเด่นและวิธีใช้ในส่วนต่างๆ ไว้อีกด้วย หรือถ้าผู้ใช้มีข้อสงสัยประการใดสามารถติดต่อกับเราได้ตามรายละเอียดที่ระบุไว้ในเว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

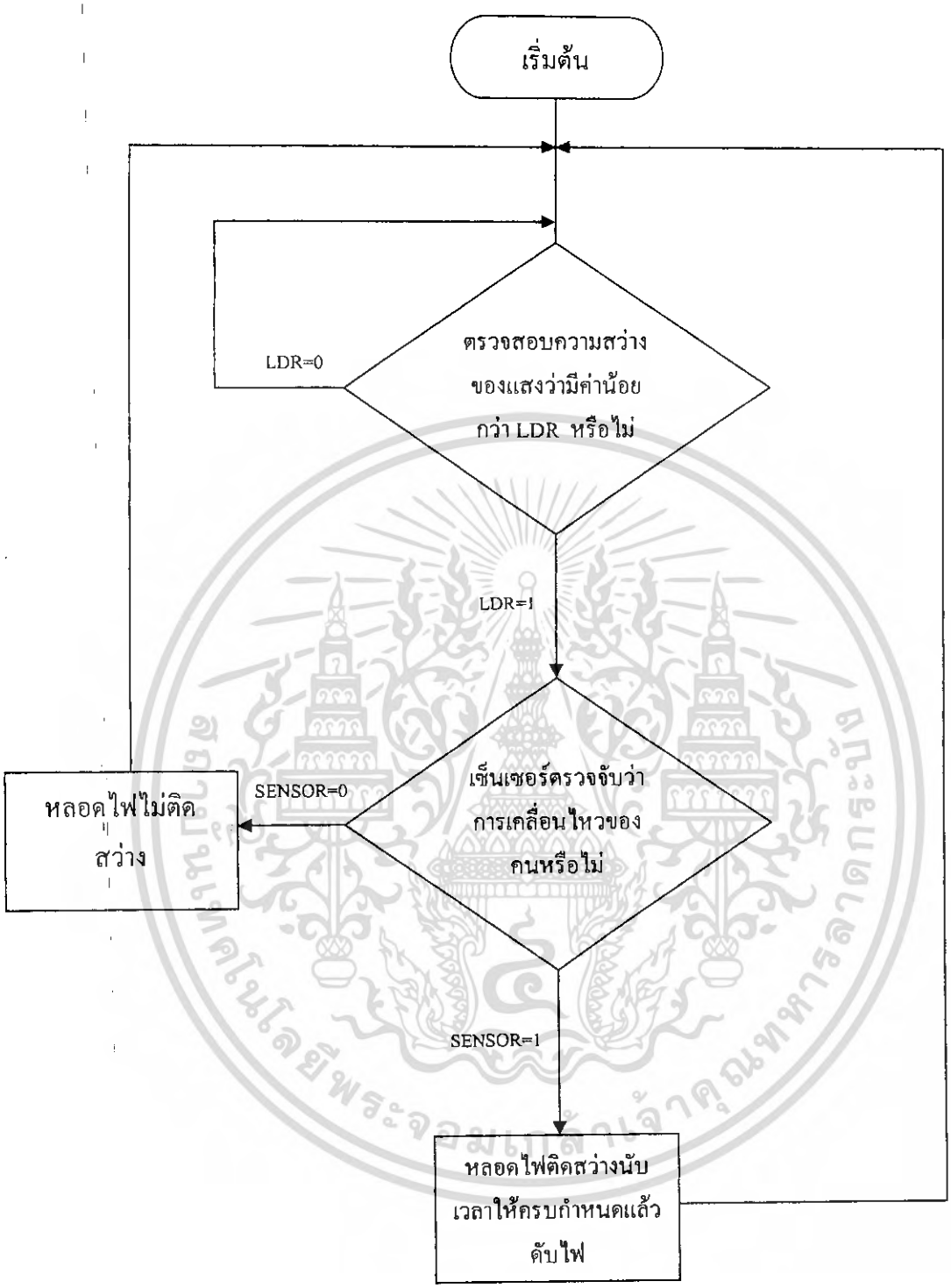


รูปที่ 3.2.6 แสดงการควบคุมระบบผ่านทางหน้าเว็บบราวน์เซอร์

3.3 แผนผังโปรแกรม

แผนผังโปรแกรม (Flowchart) ของโครงงานนี้จะเป็นแผนผังที่มีการทำงานที่วนลูปการทำงานอยู่ตลอดเวลา เพราะว่าการทำงานของโครงงานนี้จะเป็นการตรวจจับมนุษย์ เช็คเวลาและความสว่างของสภาวะภายนอกเพื่อเริ่มการทำงานของระบบ แต่จะมีการทำงานแบบอินเทอร์รัพท์จากภายนอก ซึ่งได้มีการออกแบบไว้ในเบื้องต้นสำหรับการรับข้อมูลจากภายนอกเพื่อนำมาใช้ในการประมวลผลภายในโปรแกรมของการเปิด-ปิด หลอดฟลูออเรสเซนต์ แผนผังโปรแกรมจะเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แผนผังโปรแกรมส่วนควบคุมหลอดไฟฟลูออเรสเซนต์

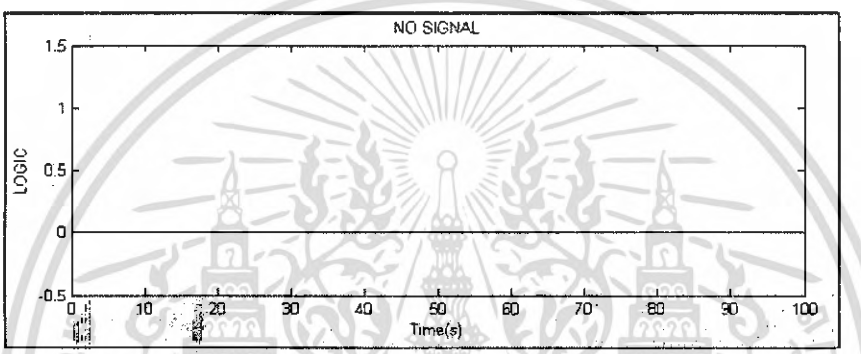
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

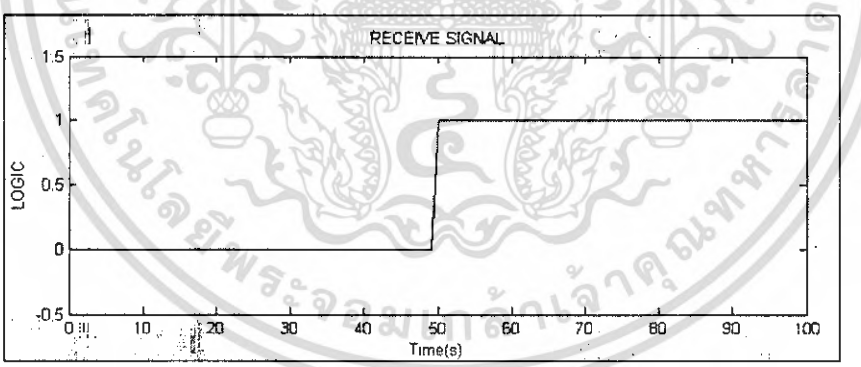
ผลการทดลอง

4.1 การทดสอบเซ็นเซอร์

การทดสอบเซ็นเซอร์เป็นการทดสอบสัญญาณในขณะที่เซ็นเซอร์กำลังทำงานอยู่ว่า มีสภาพใช้งานได้หรือไม่ ซึ่งแบ่งได้ 2 กรณี คือ มีมนุษย์เคลื่อนไหวและไม่มีมนุษย์เคลื่อนไหวหรือมีมนุษย์แต่ไม่เคลื่อนไหวในบริเวณที่เซ็นเซอร์ครอบคลุมถึง ผลการทดลองเป็นดังนี้



รูปที่ 4.1.1 สัญญาณในขณะที่ไม่มีมนุษย์เคลื่อนไหว



รูปที่ 4.1.2 สัญญาณในขณะที่มีมนุษย์เคลื่อนไหว

4.2 การทดสอบการควบคุมการเปิด-ปิดหลอดไฟ

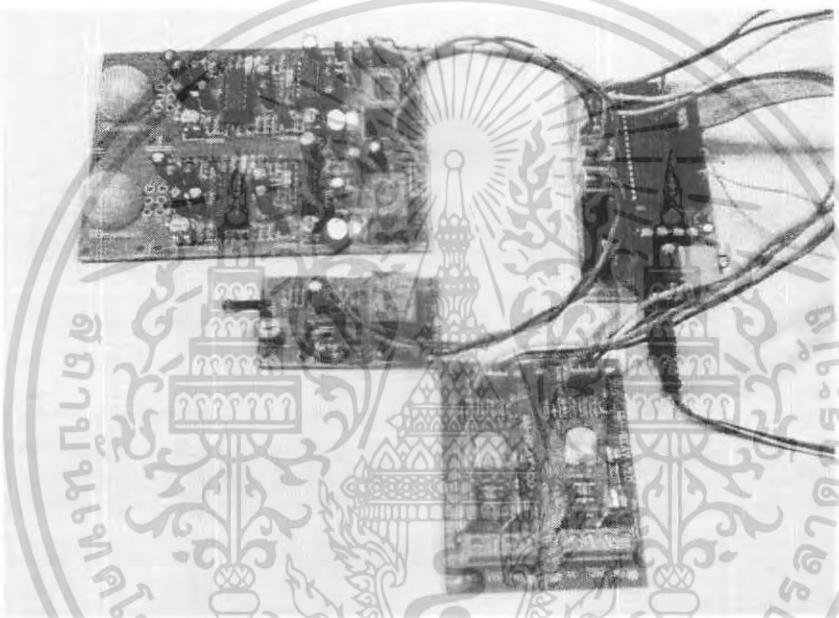
4.2.1 การควบคุมของไมโครคอนโทรลเลอร์

ไมโครคอนโทรเลอร์หนึ่งตัวจะใช้แทนการทำงานของหลอดไฟทั้งหมดภายในหนึ่งชั้น ในโปรเจคนี้เราจะแสดงการควบคุมหลอดไฟผ่านเซ็นเซอร์เพียงสองหลอดต่อหนึ่งชั้น ในขณะที่สามารถควบคุมหลอดไฟได้กว่า 20 หลอดต่อหนึ่งชั้นขึ้นกับจำนวนพอร์ทของไมโครคอนโทรลเลอร์ และแสดงการเชื่อมต่อชั้นต่อชั้นผ่าน RS485 เพียงสองชั้นแทนที่จะสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อได้มากถึงกว่า 100 ชั้น ขึ้นกับความยาวของสายเชื่อมชั้นต่อชั้นซึ่งมีความยาวได้มากถึง 1200 เมตร

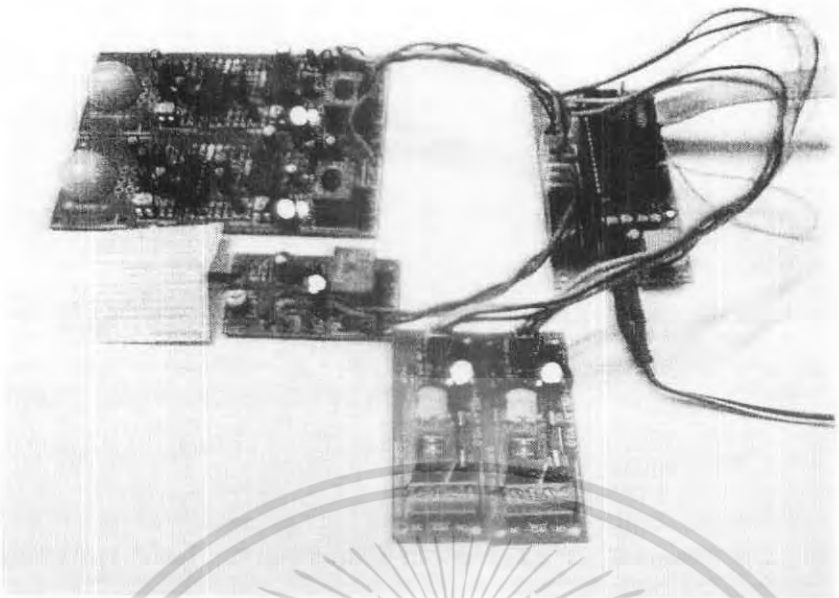
ในการทำงานของไมโครคอนโทรลเลอร์นั้นมีเงื่อนไข คือ เช็คสภาพความสว่างจากแอลดีอาร์ ก่อนจึงจะสามารถทำให้หลอดไฟติดได้ ส่วนเซ็นเซอร์นั้นถ้าตรวจจับได้ว่ามีการเคลื่อนไหวของมนุษย์จึงส่งสัญญาณไปให้ไมโครคอนโทรลเลอร์เพื่อสั่งให้รีเลย์ทำงานหลอดไฟจึงติด เพราะฉะนั้นหลอดไฟจะติดได้ก็ต่อเมื่อเซ็นเซอร์จับได้ว่า มนุษย์มีการเคลื่อนไหวและแอลดีอาร์ก็ต้องทำงานด้วย แม้เซ็นเซอร์จะตรวจจับได้ว่ามีการเคลื่อนไหวของมนุษย์ แต่ถ้าแอลดีอาร์ไม่ได้ทำงานอยู่ ไมโครคอนโทรลเลอร์ก็จะไม่สั่งให้รีเลย์ทำงาน หลอดไฟจึงไม่ติดสว่างได้



รูปที่ 4.2.1 การควบคุมการเปิด-ปิดหลอดไฟเมื่อแอลดีอาร์ไม่ได้ทำงาน

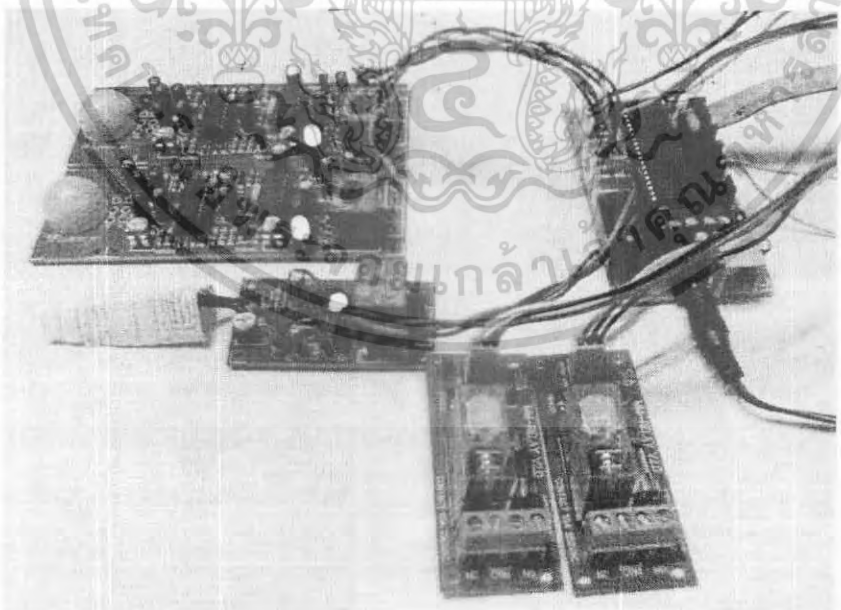
จากรูปที่ 4.2.1 จะเห็นว่าเซ็นเซอร์ตรวจจับได้ว่ามีการเคลื่อนไหวเกิดขึ้น (LED สีเขียวด้านขวาที่บอร์ดเซ็นเซอร์) ทั้งสองหลอด แต่ LED ที่รีเลย์ก็ยังไม่ทำงาน แสดงว่าหลอดไฟจะยังไม่ติด เนื่องจากยังมีแสงสว่างเพียงพออยู่ทำให้แอลดีอาร์ยังไม่ทำงาน เพราะฉะนั้นหลอดไฟไม่มีทางติดเด็ดขาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2.2 การควบคุมเมื่อแอลดีอาร์และเซ็นเซอร์ทำงาน

จากรูปที่ 4.2.2 แอลดีดาร์ถูกปิดด้วยกระดาษทำให้แสงไม่เพียงพอ LED จึงติดไฟ แสดงหลอดไฟสามารถติดสว่างได้แล้ว ที่บอร์ดของเซ็นเซอร์ดีตรวจจับได้ว่ามีการเคลื่อนไหวอยู่ทั้งสองหลอด เพราะฉนั้นรีเลย์จึงถูกสั่งให้เปิดหลอดไฟได้ทั้งสองหลอด สังเกตได้จากไฟสีแดงของ LED ที่บอร์ดรีเลย์



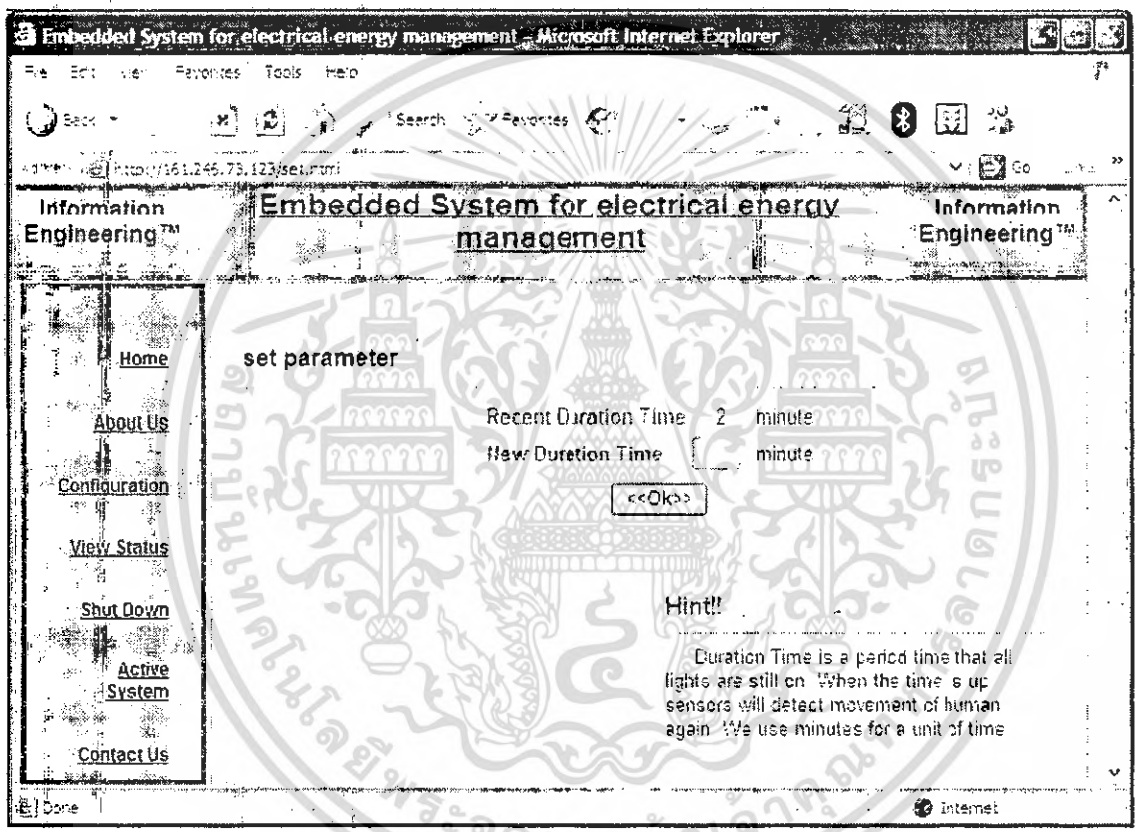
รูปที่ 4.2.3 การควบคุมเมื่อแอลดีอาร์ทำงานแต่เซ็นเซอร์ไม่ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.2.3 แอลดีอาร์ทำงานได้แล้ว แต่ถ้าเซ็นเซอร์ไม่ทำงาน หลอดไฟก็ไม่ติดด้วย เนื่องจากรีเลย์ไม่ได้ถูกสั่งให้ทำงาน

4.2.2 การควบคุมจากระบบฝังตัว

ระบบฝังตัวถูกออกแบบให้ควบคุมการเปิด-ปิดหลอดไฟในระดับสูงสุด ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ ให้ผู้ใช้สามารถติดต่อได้สะดวกผ่านเครือข่ายอินเตอร์เน็ต ความสามารถในการทำงานมีดังนี้

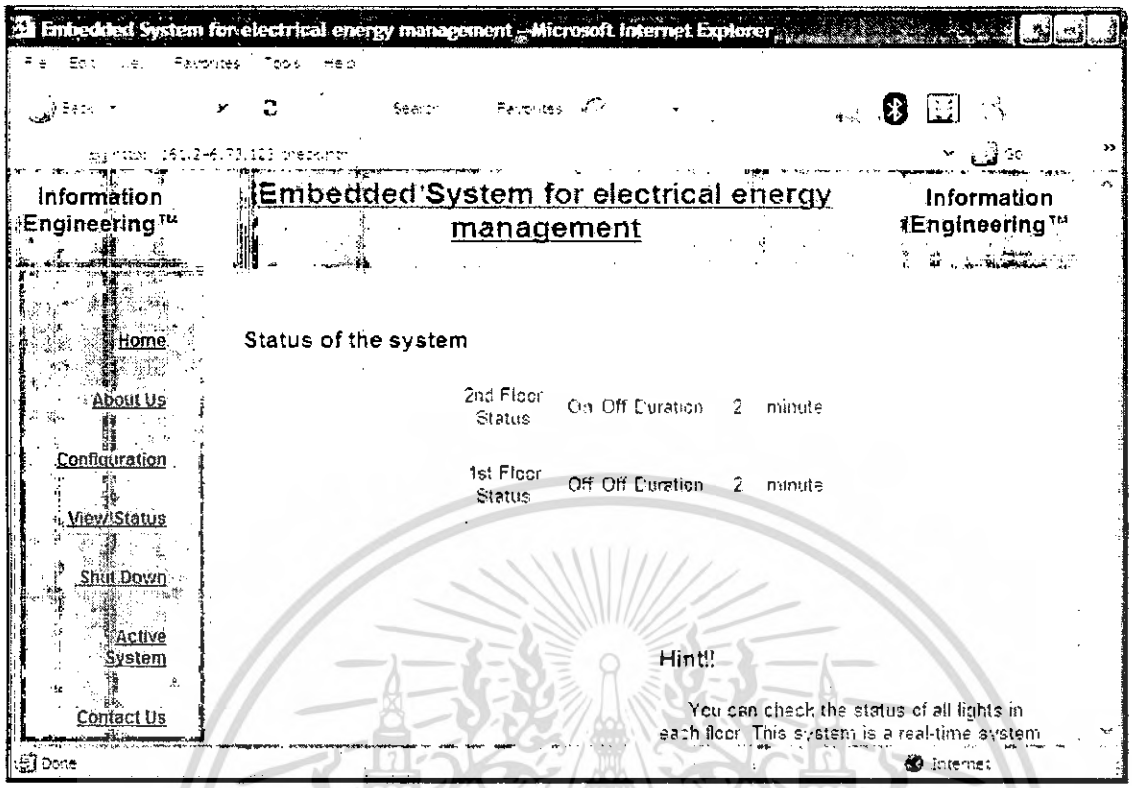


รูปที่ 4.2.4 หน้าเว็บในการเปลี่ยนค่าพารามิเตอร์

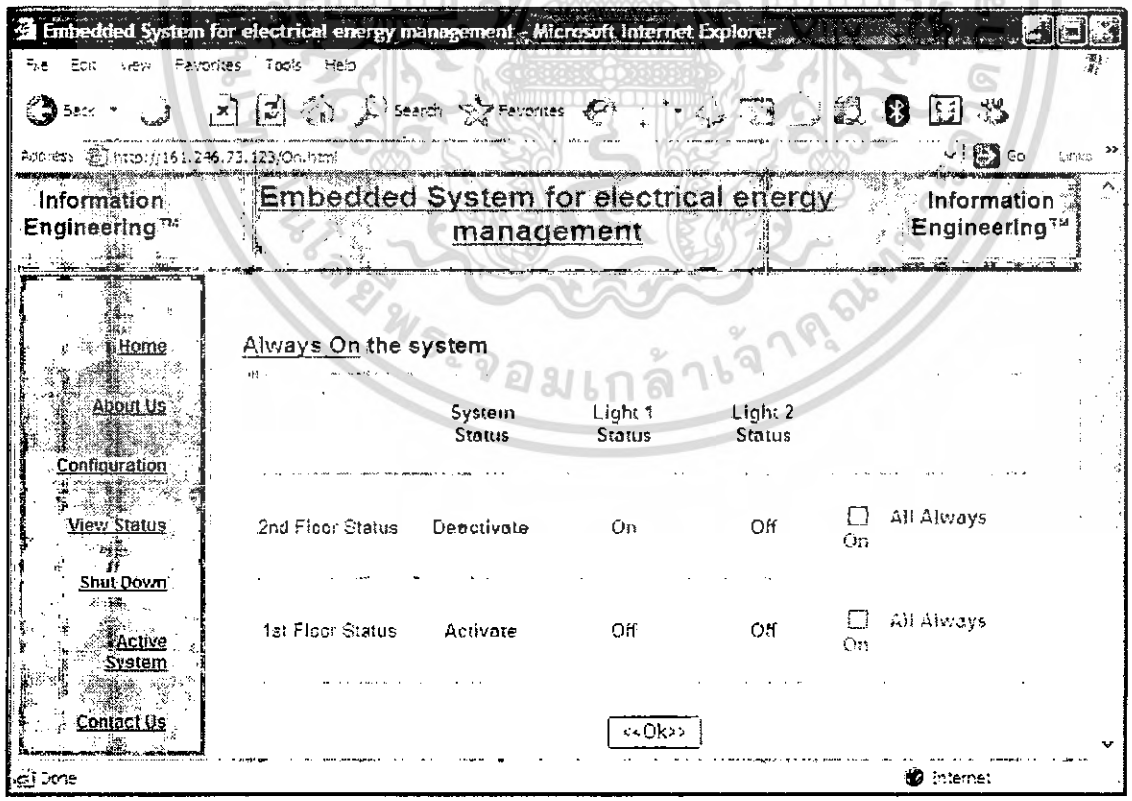
จากรูปที่ 4.2.4 จะเห็นแถบสีเทาด้านขวาเมื่อต้องการเปลี่ยนค่าพารามิเตอร์ให้คลิ๊กที่คำว่า Configuration จะเห็นหน้า set parameter ซึ่งสามารถให้ใส่ค่า Duration Time ใหม่ได้ในหน่วยนาที ซึ่งเป็นค่าที่บอกให้หลอดไฟติดค้างไว้ จนหมดเวลาแล้วจึงตรวจสอบเซ็นเซอร์อีกครั้ง

จากรูปที่ 4.2.5 เมื่อต้องการดูสถานะของหลอดไฟทั้งหมดว่า เปิดหรือปิดอยู่ ให้คลิ๊กเข้าไปที่ View Status จะเห็นหน้าของ Status of the system ซึ่งแสดงสถานะของหลอดไฟทั้งสองหลอดของแต่ละชั้น รวมทั้ง Duration Time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

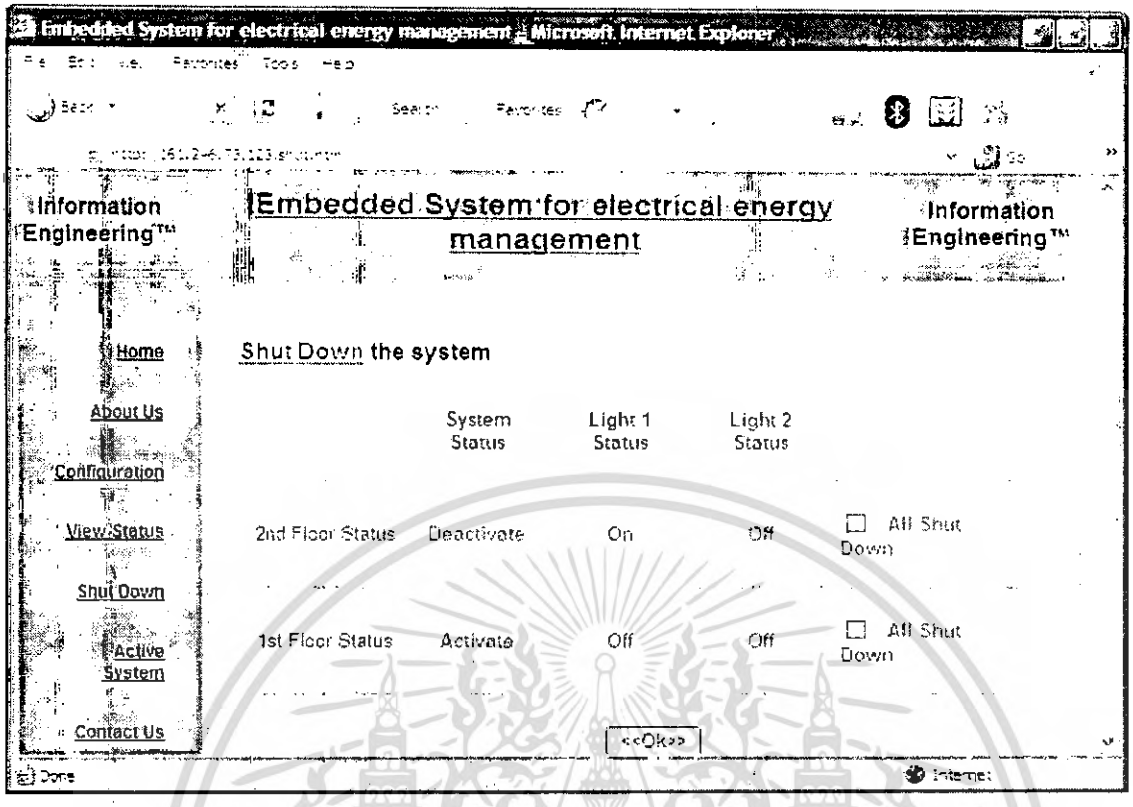


รูปที่ 4.2.5 หน้าเว็บแสดงสถานะการของแอลดีอาร์และหลอดไฟ



รูปที่ 4.2.6 หน้าเว็บในการตั้งเวลาเปิดของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2.7 หน้าเว็บในการตั้งเวลาปิดของระบบ

จากรูปที่ 4.2.6 และรูปที่ 4.2.7 เป็นหน้าต่างการรับคำสั่งให้เปิดและปิดตามลำดับ ถ้าเป็นการสั่งให้เปิดก็ให้คลิ๊กที่คำว่า Always On และคลิ๊กที่ช่อง All Always On ระบบจะสั่งให้เปิดหลอดไฟทั้งหมด ซึ่งสามารถสั่งให้เปิดได้ในแต่ละชั้น ส่วนการสั่งให้ปิดให้คลิ๊กที่คำว่า Shut Down และคลิ๊กที่ช่อง All Shut Down ระบบจะสั่งให้ชั้นนั้นๆ ปิดหลอดไฟทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลโครงงาน

5.1 สรุปผลโครงงาน

โครงงานนี้เป็นโครงงานที่สร้างระบบจัดการการปิด-เปิดหลอดไฟฟลูออเรสเซนต์ภายในอาคาร โดยนำเอาเทคโนโลยีระบบฝังตัวมาควบคุมในระดับสูงสุด และให้ไมโครคอนโทรลเลอร์ควบคุมการปิด-เปิดหลอดไฟ โดยมีเงื่อนไขจากแอลดีอาร์เป็นตัวกำหนดให้เซ็นเซอร์เริ่มทำงานหรือไม่ ซึ่งใช้ความเข้มแสงเป็นตัวกำหนด ถ้าความเข้มแสงน้อยกว่าค่าที่ตั้งไว้เซ็นเซอร์จะเริ่มทำงาน ค่าที่เซ็นเซอร์ตรวจจับได้ส่งเข้าไมโครคอนโทรลเลอร์ ถ้ามีมนุษย์เคลื่อนไหวอยู่ในบริเวณดังกล่าว เซ็นเซอร์ส่งลอจิก 1 ไมโครคอนโทรลเลอร์ประมวลผลส่งค่าลอจิก 1 ให้กับรีเลย์ เมื่อมีแรงดันให้กับรีเลย์ รีเลย์ทำงานเหนี่ยวนำให้ขารวมแตะที่ขา NO กระแสไฟฟ้าไหลผ่านได้หลอดไฟจึงติด เมื่อหมดช่วงเวลาที่กำหนด (ช่วงเวลาที่กำหนดให้หลอดไฟติดสว่างค้างไว้ในหน่วยนาที) ระบบจะทำการตรวจสอบค่าที่ได้รับเซ็นเซอร์อีกครั้งว่าในบริเวณนั้นๆ มีมนุษย์เคลื่อนไหวอยู่หรือไม่ ถ้ามีก็ให้หลอดไฟทำงานต่อไปโดยส่งลอจิก 1 ไปที่รีเลย์ ถ้าไม่มีให้ส่งลอจิก 0 ไปที่รีเลย์แทน เมื่อรีเลย์ได้รับลอจิก 0 ทำให้แรงดันไม่มี ขารวมจึงมาแตะที่ขา NC ดังเดิม กระแสไฟฟ้าจึงไม่สามารถไหลผ่านไปได้หลอดไฟจึงดับ

ส่วนการเปลี่ยนแปลงค่าพารามิเตอร์ต่างนั้นๆ สามารถทำได้ผ่านเครือข่ายอินเตอร์เน็ต เนื่องจากระบบฝังตัวถูกโปรแกรมให้เป็นเว็บเซิร์ฟเวอร์ จึงสามารถเข้าถึงฐานข้อมูลได้ผ่านทางเครือข่ายอินเตอร์เน็ต เปลี่ยนแปลงช่วงเวลาหรือเปิด-ปิดหลอดไฟได้ในขณะที่เซ็นเซอร์ทำงานอยู่ได้ สามารถแสดงสถานะของหลอดไฟแต่ละหลอด และแอลดีอาร์แต่ละตัวได้ว่าทำงานอยู่หรือไม่ และสามารถตั้งเวลาการทำงานของระบบนี้ได้อีกด้วย

5.2 ปัญหาที่พบในโครงงาน

5.2.1 ส่วนของโปรแกรม

การเขียนโปรแกรมที่มีประสิทธิภาพดีต้องมีการจัดการกับหน่วยความจำให้คุ้มค่ามากที่สุด มิฉะนั้นโปรแกรมจะทำงานได้ช้ามาก และปัญหาสำคัญที่เกิดขึ้นอีกก็คือ เกิดการวนลูปไม่รู้จบของการโปรแกรมที่เกี่ยวเนื่องกับการอินเตอร์รัพท์ เพราะเราไม่สามารถรู้ได้เลยว่าจะออกจากลูปได้เมื่อไหร่ การเขียนโปรแกรมจึงต้องมีการทดสอบให้ถูกต้องเสียก่อน

5.2.2 ส่วนไมโครคอนโทรลเลอร์

การจัดการการพอร์ทต่างๆ ในไมโครคอนโทรลเลอร์ พอร์ทบางพอร์ทมีหน้าที่พิเศษแตกต่างกัน เพราะฉะนั้นการกำหนดจึงต้องศึกษาการทำงานของพอร์ทแต่ละพอร์ทให้ดีเสียก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ควรกำหนดหน้าที่ของพอร์ทซ้ำซ้อนกัน จะทำให้ไมโครคอนโทรลเลอร์ไม่สามารถทำงานได้อย่างมีประสิทธิภาพ

เราไม่สามารถใช้ไมโครคอนโทรลเลอร์เป็นตัวกลางกระจายข้อมูลได้ เนื่องจากมีข้อจำกัดของจำนวนพอร์ทที่ใช้งาน และระยะทางในการส่งข้อมูล จึงเลือกใช้ RS422/RS485 ซึ่งสามารถใช้ได้ในระยะทางมากที่สุดถึง 1.2 กิโลเมตร และไม่จำกัดจำนวนชั้นที่จะกระจายข้อมูลไปถึง

5.2.3 ส่วนของอุปกรณ์ต่างๆ

เซ็นเซอร์เป็นอุปกรณ์สำเร็จรูป ไม่ต้องผ่านการขยายสัญญาณเพิ่มเติมอีกเนื่องจากมีวงจรขยายสัญญาณอยู่ภายในเรียบร้อยแล้ว จึงเป็นอุปกรณ์ที่มีราคาแพงพอสมควร

อุปกรณ์ชิ้นเล็กอย่างอื่นเช่น ตัวต้านทาน ตัวเก็บประจุ หลอด LED เป็นต้น แม้จะมีราคาถูกแต่ก็ได้รับความเสียหายง่ายเช่นกัน จึงควรระมัดระวังการใช้อุปกรณ์เหล่านี้ให้เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] ดร.ธนารักษ์ ธีระมั่นคง, 2549, เทคโนโลยีสมองกลฝังตัว Embedded Technology, กรุงเทพ, สำนักพิมพ์ สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น)

[2] อุดม รานอก, 2548, ภาษา C สำหรับงานควบคุมไมโครคอนโทรลเลอร์ MCS-51, นนทบุรี, บริษัท ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์ จำกัด

[3] สันดิ นุราช และ อุกฤษฎ์ ตันทสุทธานนท์, 2548, เรียนรู้ไมโครคอนโทรลเลอร์ MCS-51 ฉบับภาษา C, ปทุมธานี, บริษัท ไมโคร รีเสริช เทคโนโลยี จำกัด

[4] http://electronics.se-ed.com/contents.html

[5] http://en.wikipedia.org/wiki/Sensor

[6] http://www.geocities.com/kitti_hirun2001/pir.html

[7] http://www.smartlearningweb.com/knowledge/relay/relay.html

[8] http://www.st.kmutt.ac.th/~s5400211/module4/contactor1.html

[9] http://www.silaresearch.com/data/mp10m.pdf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการควบคุมการปิด-เปิดหลอดไฟ

```
#include "reg52.h"
#include "absacc.h"
#include "Function.c"

sbit     Dir  = P3^2;
sbit   Lamp1 = P2^4;
sbit   Lamp2 = P2^6;


sbit   Sw1  = P1^0;
sbit   Sw2  = P1^1;
sbit   Sw3  = P1^2;
sbit   Sw4  = P1^3;

sbit   LED1 = P1^4;
sbit   LED2 = P1^5;

sbit   Sen1 = P0^4;
sbit   Sen2 = P0^6;
sbit   LDR  = P0^0;

// ---------------------------------- //
// -------- Main Program ------------ //
// ---------------------------------- //

unsigned char  RxBuf[6];
unsigned char  Slave,Level,Comm,Power;
unsigned int   Times1,Times2,STime1,STime2,STime;
bit            TFlag;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void main(void)
{
        Init_Serial(96);
        Serial_IntOn();
        Init_Timer0(15);
        Slave ='1';
        Dir   = 0;
        TFlag = 0;
        STime1= 0;
        STime2= 0;
        STime = 1;
//      Power = '1';
while(1)
  {

        if((Power=='0')||(LDR==1))
        {Lamp1 = 1;Lamp2 = 1;
         STime2 = 0;
         STime1 = 0;
         }else
         { if(Sen1 == 0){STime1 = STime;LED1 = 1;Times1=0;}
    if(Sen2 == 0){STime2        = STime;LED2= 1;Times2=0;}
         if(STime1 > 0){ Lamp1 = 0; }
         if(STime2 > 0){ Lamp2 = 0; }
         if(STime1 ==0){Lamp1 = 1;}
         if(STime2 ==0){Lamp2 = 1;}
         }
    if(TFlag == 1)
         { if(Comm=='1')        // Acknowledge  Read Command
           {Delay(100);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Dir = 1; Delay(50);
      Tx_Byte('S');
      Tx_Byte(Slave);
                  Tx_Byte('1');  // Command  Read
                  if(LDR ==0){Tx_Byte('1');}else{Tx_Byte('0');}
                  if(STime1>0){Tx_Byte('1');}else{Tx_Byte('0');}
                  if(STime2>0){Tx_Byte('1');}else{Tx_Byte('0');}
                  Delay(50);
                  Dir = 0;
                  TFlag = 0;
                  }
            }
           if(Sw1 == 0) { Slave = '1';}else
           if(Sw2 == 0) { Slave = '2';}else
           if(Sw3 == 0) { Slave = '3';}else
     if(Sw4 == 0) { Slave = '4';}
   }
}

void  Serial_Int(void)  interrupt 4
{ unsigned char Ch;
 if (RI==1)                  /* Check  interrupt from  Received  */
           {
     Ch = Rx_Byte();
           if(Level == 0)
           { if(Ch == 'M'){Level = 1;}else{Level = 0;}
           }else
           if(Level == 1)
           { if(Ch == Slave){Level = 2;}else{Level = 0;}
           }else
           if(Level == 2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        { if(Ch == '1'){Level = 3;Comm = '1';}else
          if(Ch == '0'){Level = 3;Comm = '0';}else
          {Level = 0;}
        }else
       if(Level == 3)
        {
         Power   = Ch;
              RxBuf[1] = Rx_Byte();
              RxBuf[0] = Rx_Byte();
              STime   = ((RxBuf[1]-0x30)*10)+(RxBuf[0]-0x30);
              STime   = STime * 60;
              Level = 0;
              TFlag = 1;
        }else { Level = 0;}
       }
}
void Timer0_Int(void) interrupt 1
{
 TH0 = _TH0;
 TL0 = _TL0;
 Times1++;
 Times2++;
 if(Times1==15)
 {Times1 = 0;
  if(STime1 > 0){STime1--;}
  }
 if(Times2==15)
  {Times2 = 0;
  if(STime2 > 0){STime2--;}
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้