สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การประยุกต์การใช้งานรหัสแถบกับระบบควบคุมการเข้าออก

APPLICATION OF BARCODE FOR ACCESS CONTROL SYSTEM

T054953

โดย

นายกล้าณรงค์ ตังวงศ์ประเสริฐ

นายธงชัย วิจิตรพรชัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาตรีวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เลขหมู่.....
เลขทะเบียน 54953
วัน,เดือน,ปี - 1 มิ.ย. 2548
b.....
i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์การใช้งานรหัสแถบกับระบบควบคุมการเข้าออก

APPLICATION OF BARCODE FOR ACCESS CONTROL SYSTEM

โดย

นายกล้าณรงค์ ตังวงศ์ประเสริฐ 43010017

นายธงชัย วิจิตรพรชัย 43010155

อาจารย์ที่ปรึกษา

รศ.ดร.ปราโมทย์ วาดเขียน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาตรีวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2546

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **การประยุกต์ใช้งานรหัสแถบกับระบบควบคุมการเข้าออก**

APPLICATION OF BARCODE FOR ACCESS CONTROL SYSTEM

ผู้จัดทำ

1. นายกล้าณรงค์ ตังวงศ์ประเสริฐ 43010017

2. นายธงชัย วิจิตรพรชัย 43010155

.....ปราโมทย์..... อาจารย์ที่ปรึกษา

(รศ.ดร.ปราโมทย์ วาดเขียน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ใช้งานรหัสแถบกับระบบควบคุมการเข้าออก
APPLICATION OF BARCODE FOR ACCESS CONTROL SYSTEM

โดย นายกล้าณรงค์ ตังวงศ์ประเสริฐ รหัส 43010017
นายธงชัย วิจิตรพรชัย รหัส 43010155

อาจารย์ที่ปรึกษา รศ.ดร.ปราโมทย์ วาดเขียน

บทคัดย่อ

โครงงานนี้เป็นการสร้างระบบควบคุมการเข้าออกประตูโดยใช้รหัสแถบ โดยการอ่านค่ารหัสแถบจะใช้เครื่องอ่านแล้วส่งค่าที่อ่านได้ไปประมวลผลที่คอมพิวเตอร์ ผ่านทางพอร์ตอนุกรม RS232 เพื่อทำการเปรียบเทียบค่าที่อ่านได้กับฐานข้อมูล แล้วนำค่าที่ประมวลผลได้ส่งให้ไมโครคอนโทรลเลอร์ ทำการสั่งให้สเตปปิ้งมอเตอร์ ควบคุมให้ประตูเปิด-ปิด พร้อมทั้งแสดงสถานะการเข้าใช้ผ่านทางจอ แอลซีดี พร้อมทั้งบันทึกเวลาเข้าใช้ลงในฐานข้อมูล

Abstract

Access control system using Barcode technology is realized in this project. Barcode reader reads the code from card and sends code to process in computer through a serial port (RS232). The code from barcode reader is compared with the code from database in computer. Then computer sends a message to microcontroller in order to control stepping motor for locking or unlocking the door. In addition, the status of the door is displayed on a LCD and the access time is recorded to the database.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	5
2.1 ระบบมาตรฐานรหัสแถบ	5
2.1.1 หลักการรหัสแถบ	5
2.1.2 การอ่านรหัสแถบ	5
2.1.3 ชนิดของรหัสแถบ	6
2.2 หลักการอ่านและประมวลผลรหัส 3 ใน 9	7
2.3 ชนิดของหัวอ่านรหัสแถบ	9
2.3.1 หัวอ่านชนิดสัมผัสโดยตรง (Contact Scanner)	9
2.3.2 หัวอ่านชนิดไม่สัมผัสโดยตรง (Non-contact Scanner)	10
2.4 โครงสร้างของ MCS-51	11
2.4.1 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51	11
2.4.2 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51	12
2.4.3 โครงสร้างและการทำงานของพอร์ต	14
2.5 พื้นฐานการติดต่อสื่อสารข้อมูลแบบอนุกรม	16
2.5.1 การอินเตอร์เฟสผ่านพอร์ตอนุกรม	16
2.5.2 จังหวะเวลาของการสื่อสารข้อมูลอนุกรม	16
2.5.3 รูปแบบของข้อมูลอนุกรม	16
2.5.4 การจัดการข้อมูลอนุกรมของ MCS-51	17
2.5.5 การอินเตอร์รัพต์ของการสื่อสารอนุกรม	18
2.5.6 กระบวนการรับและส่งข้อมูลอนุกรมของ MCS-51	18
2.5.7 โหมดการทำงานของการสื่อสารผ่านพอร์ตอนุกรม	19
2.5.8 การเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์	22
2.6 มาตรฐานพอร์ตอนุกรมแบบ RS232	23
2.7 สเตปปิ้งมอเตอร์	25
2.7.1 หลักการทำงานเบื้องต้นของสเตปปิ้งมอเตอร์	25
2.7.2 รูปแบบการขับสเตปปิ้งมอเตอร์	27
2.8 แอลซีดีโมดูล (LCD Module)	28
2.8.1 โครงสร้างภายในตัวควบคุมแอลซีดีโมดูล	29
2.8.2 ขาสัญญาณของแอลซีดีโมดูล	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง	หน้า
2.8.3 การต่อเข้ากับระบบไมโครคอนโทรลเลอร์	30
2.9 วงจรตัวส่ง และตัวรับสัญญาณอินฟราเรด	31
2.10 คีย์แพด (Keypad)	32
2.11 ระบบบัส I^2C	32
2.11.1 คุณสมบัติโดยทั่วไปของระบบบัส I^2C	33
2.11.2 หลักการของบัส I^2C	34
2.11.3 สภาวะที่เกิดขึ้นบนบัส I^2C	35
2.11.4 การทำงานบนบัส I^2C	36
2.11.5 อุปกรณ์ที่ใช้การเชื่อมต่อแบบบัส I^2C	37
2.11.6 การต่ออุปกรณ์ระบบบัส I^2C กับไมโครคอนโทรลเลอร์ MCS-51	38
2.11.7 การเขียนโปรแกรมติดต่อบัส I^2C	38
2.12 แนะนำ DS1307 ไอซีสร้างฐานเวลาจริงหรือรีลไทม์คล็อก (RTC)	39
2.12.1 รายละเอียดขาต่อใช้งานของ DS1307	40
2.12.2 การทำงานของ DS1307	40
2.12.3 การจัดสรรหน่วยความจำใน DS1307	41
2.12.4 รีจีสเตอร์ควบคุม	42
2.12.5 โหมดการทำงานของ DS1307	42
2.12.6 การเชื่อมต่อ DS1307 กับไมโครคอนโทรลเลอร์ MCS-51	44
2.13 รอม (ROM: Read Only Memory)	44
2.14 แนะนำไอซี 24LC256 CMOS Serial EEPROM	45
2.14.1 รายละเอียดของขาที่ต่อใช้งาน	45
2.14.2 การระบุแอดเดรสของอุปกรณ์	45
2.14.3 กระบวนการเขียนข้อมูล	46
2.14.4 กระบวนการอ่านข้อมูล	47
บทที่ 3 การคำนวณและการออกแบบ	49
3.1 การรับค่ารหัสจากเครื่องอ่านบาร์โค้ด	49
3.2 Stepping Motor	50
3.3 วงจร Sensor	52
3.4 ไอซีสร้างฐานเวลาจริงหรือรีลไทม์คล็อก (RTC)	54
3.5 การติดต่อรอม	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
3.6 คีย์แพด	57
3.7 การส่งข้อมูลไปยังเครื่องคอมพิวเตอร์ผ่านพอร์ตอนุกรม	58
3.8 การรับข้อมูลจากไมโครคอนโทรลเลอร์ ผ่านทางพอร์ตอนุกรม	59
3.9 การติดต่อกับฐานข้อมูลในคอมพิวเตอร์	61
บทที่ 4 การทดลองและผลการทดลอง	62
4.1 การทดลองวัดสัญญาณจากเครื่องอ่านบาร์โค้ด	62
4.2 การทดลองขับ Stepping motor	68
4.3 การทดลองวงจรรับส่งอินฟราเรด	70
4.3.1 วงจรทางด้านส่ง	70
4.3.2 วงจรทางด้านรับ	71
4.4 การทดสอบส่วนตัวเครื่อง	72
4.4.1 การทดสอบการเข้าใช้งานของ USER	72
4.4.2 การทดสอบส่วนของผู้ดูแลระบบ	73
4.5 การทดสอบโปรแกรมรับข้อมูลบนเครื่องคอมพิวเตอร์	76
บทที่ 5 บทวิจารณ์และบทสรุป	83
ภาคผนวก	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่ 1.1 แสดงบล็อกไดอะแกรมของระบบ 2

รูปที่ 1.2 แสดงแผนภาพการทำงานของระบบ 3

รูปที่ 1.3 บล็อกไดอะแกรมการทำงานในการเชื่อมต่อกับคอมพิวเตอร์ 4

รูปที่ 1.4 แผนภาพการทำงานขณะเชื่อมต่อกับคอมพิวเตอร์ 4

รูปที่ 2.1 แสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89CXX 11

รูปที่ 2.2 แสดงรายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชของ Atmel 13

รูปที่ 2.3 แสดงวงจรภายในของพอร์ตในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 14

รูปที่ 2.4 แสดงแผนภาพเวลาต่างๆในโหมด 0 เมื่อมีการรับและส่งข้อมูล 1 ไบต์ 19

รูปที่ 2.5 แสดงรูปแบบของสัญญาณข้อมูลอนุกรมในโหมด 1 20

รูปที่ 2.6 แสดงรูปแบบของสัญญาณข้อมูลอนุกรมในโหมด 2 และ โหมด3 21

รูปที่ 2.7 แสดงรายละเอียดเบื้องต้นของไอซีแปลงสัญญาณเพื่อเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์ 23

รูปที่ 2.8 แสดงการจัดขาของคอนเนคเตอร์พอร์ตอนุกรมตามมาตรฐาน RS232 24

รูปที่ 2.9 แสดงส่วนประกอบแบบง่ายๆของสเตปปิ้งมอเตอร์ที่มีมุมโรเตอร์เคลื่อนที่ 60 องศา 26

รูปที่ 2.10 แสดงหลักการควบคุมสเตปปิ้งมอเตอร์ 26

รูปที่ 2.11 แสดงวงจรสร้างพัลส์เพื่อใช้ในการขับหลอดแอลอีดีในวงจรอินฟราเรดทางด้านส่ง 31

รูปที่ 2.12 วงจรเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51 32

รูปที่ 2.13 แสดงแผนผังการเชื่อมต่ออุปกรณ์ต่างๆบนระบบบัส I^2C 33

รูปที่ 2.14 วงจรเอาท์พุตของอุปกรณ์ที่ต่ออยู่กับระบบบัส I^2C 33

รูปที่ 2.15 การต่อตัวต้านทานพูลอัปบนสายสัญญาณในระบบ I^2C 34

รูปที่ 2.16 ไดอะแกรมเวลาแสดงสถานะต่างๆในบัส I^2C 35

รูปที่ 2.17 รูปแบบข้อมูลกำหนดแอดเดรสที่ใช้ในการอ้างอิงถึงแบบ 7 บิท 36

รูปที่ 2.18 รูปแบบของข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส I^2C เมื่อใช้การอ้างถึงแบบ 7 บิท 37

รูปที่ 2.19 รูปแบบของข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส I^2C เมื่อใช้การอ้างถึงแบบ 7 บิท 37

รูปที่ 2.20 วงจรตัวอย่างการต่อไมโครคอนโทรลเลอร์ MCS-51 กับอุปกรณ์ระบบบัส I^2C 38

รูปที่ 2.21 แสดงการจัดขาของ DS1307 40

รูปที่ 2.22 โครงสร้างภายในของไอซีรีลไทม์คล็อกเบอร์ DS1307 41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 2.23 (ก) แสดงการจัดสรรพื้นที่ของหน่วยความจำภายในของ DS1307 42
(ข) แสดงรายละเอียดของรีจีสเตอร์ค่าเวลา และรีจีสเตอร์ควบคุม
รูปที่ 2.24 รูปแบบข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการเขียนข้อมูล 43
รูปที่ 2.25 รูปแบบข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการอ่านข้อมูล 43
รูปที่ 2.26 การเชื่อมต่อไมโครคอนโทรลเลอร์ MCS-51 กับไอซีรีลไทม์คล็อก DS1307 44
รูปที่ 2.27 การจัดขาของไอซี 24LC256 45
รูปที่ 2.28 การระบุแอดเดรสของอุปกรณ์ 45
รูปที่ 2.29รูปแบบของคอนโทรลไบต์ 46
รูปที่ 2.30 กระบวนการขียนข้อมูล 1 ไบต์ 46
รูปที่ 2.31กระบวนการเขียนข้อมูลต่อเนื่อง 47
รูปที่ 2.32 การอ่านข้อมูล 1 ไบต์ ที่แอดเดรสปัจจุบัน 47
รูปที่ 2.33 การอ่านข้อมูล 1 ไบต์ โดยระบุแอดเดรส 48
รูปที่ 2.34 การอ่านข้อมูล ต่อเนื่องตั้งแต่แอดเดรสที่ระบุต่อเนื่อง 48
รูปที่ 3.1 แผนภาพการทำงานส่วนที่ทำการรับค่ารหัสจากเครื่องอ่านบาร์โค้ด 50
รูปที่ 3.2 แผนภาพการทำงานของ Stepping Motor 51
รูปที่ 3.3 วงจรภาคส่งสัญญาณอินฟราเรด 52
รูปที่ 3.4 วงจรภาครับสัญญาณอินฟราเรด 53
รูปที่ 3.5 แผนภาพการทำงานของ Sensor สำหรับตรวจสอบสถานะประตู 53
รูปที่ 3.6 การอ่านและเขียนข้อมูลระบบบัส I^2C 54
รูปที่ 3.7 แสดงการติดต่อกับรอมโดยการอ่านข้อมูลและเขียนข้อมูล 1 ไบต์ 56
รูปที่ 3.8 การทำงานของโปรแกรมที่ใช้ตรวจสอบการกดปุ่มคีย์แพด 57
รูปที่ 3.9 แผนผังการส่งข้อมูลไปยังเครื่องคอมพิวเตอร์ผ่านพอร์ตอนุกรม 58
รูปที่ 3.10 การทำงานของการรับข้อมูลจากไมโครคอนโทรลเลอร์ ผ่านทางพอร์ตอนุกรม 60
รูปที่ 3.11 แสดงแผนภาพการทำงานส่วนที่เชื่อมต่อกับ Computer 61
รูปที่ 4.1 แสดงวงจรที่ใช้ทดลองวัดค่าสัญญาณจากเครื่องอ่านบาร์โค้ด 62
รูปที่ 4.2 แสดงสัญญาณเมื่อทำการรูดรหัส 0 63
รูปที่ 4.3 แสดงสัญญาณเมื่อทำการรูดรหัส 1 63
รูปที่ 4.4 แสดงสัญญาณเมื่อทำการรูดรหัส 2 64
รูปที่ 4.5 แสดงสัญญาณเมื่อทำการรูดรหัส 3 64
รูปที่ 4.6 แสดงสัญญาณเมื่อทำการรูดรหัส 4 65
รูปที่ 4.7 แสดงสัญญาณเมื่อทำการรูดรหัส 5 65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 4.8 แสดงสัญญาณเมื่อทำการรูดรหัส 6 66
รูปที่ 4.9 แสดงสัญญาณเมื่อทำการรูดรหัส 7 66
รูปที่ 4.10 แสดงสัญญาณเมื่อทำการรูดรหัส 8 67
รูปที่ 4.11 แสดงสัญญาณเมื่อทำการรูดรหัส 9 67
รูปที่ 4.12 แสดงวงจรที่ใช้ในการทดลองการขับ Stepping motor 68
รูปที่ 4.13 แสดงรูปสัญญาณที่พอร์ตต่างๆ เมื่อทำการกดสวิตซ์ S1 เพื่อให้สเตปมอเตอร์หมุนทางขวา 69
รูปที่ 4.14 แสดงรูปสัญญาณที่พอร์ตต่างๆ เมื่อทำการกดสวิตซ์ S2 เพื่อให้สเตปมอเตอร์หมุนทางซ้าย 69
รูปที่ 4.15 แสดงวงจรทางด้านส่งอินฟราเรด 70
รูปที่ 4.16 แสดงผลการทดลองการวัดสัญญาณที่ขา 3 ของ IC เบอร์ LM555 เป็น pulse ความถี่ 38 kHz 70
รูปที่ 4.17 แสดงวงจรทางด้านรับอินฟราเรด 71
รูปที่ 4.18 แสดงสัญญาณทางด้านรับกรณีที่รับสัญญาณได้ 71
รูปที่ 4.19 แสดงสัญญาณทางด้านรับกรณีที่รับสัญญาณไม่ได้ 72
รูปที่ 4.20 กรณีที่นักศึกษาที่ได้รับอนุญาตให้เข้าใช้งาน ป้อน PASSWORD ถูก 72
รูปที่ 4.21 กรณีที่นักศึกษาที่ได้รับอนุญาตให้เข้าใช้งาน ป้อน PASSWORD ผิด 73
รูปที่ 4.22 กรณีที่นักศึกษาที่ไม่ได้รับอนุญาตให้เข้าใช้งาน 73
รูปที่ 4.23 การเข้าสู่ระบบของ “ADMIN” 74
รูปที่ 4.24 ขั้นตอนการตั้งเวลาให้กับตัวเครื่อง 74
รูปที่ 4.25 การติดต่อกับ ROM ตัวที่ 1 เพื่อเพิ่มรหัสในฐานข้อมูล 75
รูปที่ 4.26 การติดต่อกับ ROM2 เพื่อดูฐานข้อมูลผู้เข้าใช้งาน 75
รูปที่ 4.27 การส่งข้อมูลผู้ใช้งานที่เก็บใน ROM2 ไปยังคอมพิวเตอร์ 76
รูปที่ 4.28 โปรแกรมที่ใช้รับข้อมูล 76
รูปที่ 4.29 หน้าจอขณะรอรับข้อมูลจากไมโครคอนโทรลเลอร์ 77
รูปที่ 4.30 แสดงหน้าจอเมื่อกดปุ่ม “รับข้อมูล” เพื่อรับข้อมูลจากไมโครคอนโทรลเลอร์ 77
รูปที่ 4.31 แสดงข้อมูลในฐานข้อมูลผู้เข้าใช้งาน 78
รูปที่ 4.32 แสดงการค้นหาข้อมูลตามรหัสนักศึกษา 79
รูปที่ 4.33 แสดงการค้นหาข้อมูลตามวันเดือนปีที่เข้าใช้งาน 80
รูปที่ 4.34 แสดงการเรียงข้อมูลตามรหัสนักศึกษา 81
รูปที่ 4.35 แสดงฐานข้อมูลนักศึกษา 82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงเลขฐาน 2 ที่ใช้แทนอักขระทั้ง 44 ตัว	8
ตารางที่ 2.2 แสดงอักขระและรหัสแถบที่นำมาใช้งาน	9
ตารางที่ 2.3 แสดงโหมดการทำงานของพอร์ตอนุกรมของ MCS-51	17
ตารางที่ 2.4 แสดงการกำหนดบิต SM0 และ SM1 ซึ่งอยู่ภายในรีจีสเตอร์ควบคุม และบอกสถานะ SCON	18
ตารางที่ 2.5 แสดงค่าสภาวะที่จะทำให้เกิดการโอนย้ายข้อมูลไปเก็บในรีจีสเตอร์ SBUF	20
ตารางที่ 2.6 แสดงการเลือกอัตราบอดโดยการกำหนดค่าต่างๆของไทเมอร์ 1	22
ตารางที่ 2.7 แสดงการจัดขาของคอนเนคเตอร์พอร์ตอนุกรมตามมาตรฐาน RS232	24
ตารางที่ 2.8 แสดงการจ่ายกระแสแบบการกระตุ้น 1 เฟส	27
ตารางที่ 2.9 แสดงการจ่ายกระแสแบบการกระตุ้น 2 เฟส	27
ตารางที่ 2.10 แสดงการจ่ายกระแสแบบการกระตุ้น 1 เฟสสลับ 2 เฟส	28
ตารางที่ 2.11 แสดงขาสัญญาณของ แอลซีดีโมดูล	30
ตารางที่ 3.1 แสดงค่ารีจิสเตอร์ต่างๆที่ใช้ในการเขียนโปรแกรมรับค่าจากเครื่องอ่านบาร์โค้ด	49
ตารางที่ 3.2 แอดเดรสสำหรับเก็บข้อมูลต่างๆ	55
ตารางที่ 3.3 การกำหนดคอนโทรลไบต์สำหรับติดต่อรอม	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบัน ระบบการสื่อสาร ได้มีการพัฒนาเทคโนโลยีต่างๆ ให้เจริญก้าวหน้าอย่างไม่หยุด รูปแบบของข้อมูลก็เช่นกัน ได้มีการเปลี่ยนรูปแบบของข้อมูลให้อยู่ในรูปแบบของรหัส ซึ่งจะใช้แทนตัวแปรต่างๆ ของข้อมูลขึ้นอยู่กับการใช้งาน

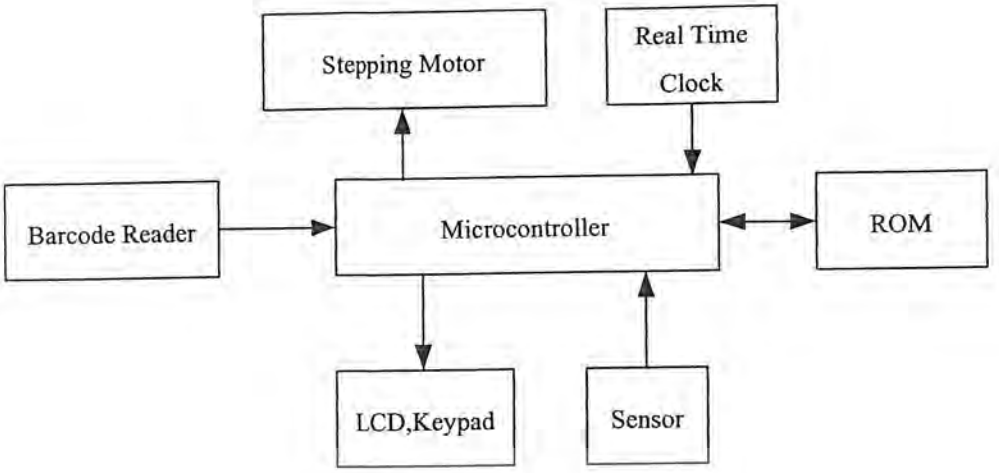
รหัสแถบ (Barcode) ก็เป็นรูปแบบของรหัสที่ใช้แทนข้อมูลชนิดหนึ่ง ซึ่งใช้กันอย่างแพร่หลายในปัจจุบัน ซึ่งจะพบว่าได้มีการนำรหัสแถบมาใช้ในงานด้านต่างๆมากมาย เช่น ใช้แทนรหัสสินค้าต่างๆ, ใช้แทนรหัสสมาชิกร้านวิดีโอ, ร้านเช่าหนังสือ, งานไปรษณีย์ใช้คัดจดหมาย และยังใช้แทนรหัสนักศึกษาบนบัตรนักศึกษาอีกด้วย ซึ่งการแทนข้อมูลจะแทนด้วยแถบสีดำและสีขาว โดยใช้อุปกรณ์ในการอ่านรหัสแถบสีขาวและดำนั้นให้กลายเป็นข้อมูลดังเดิม

รหัสแถบจึงมีรูปแบบการใช้งานที่ง่าย และมีความน่าเชื่อถือ เหมาะกับการใช้งานกับระบบคอมพิวเตอร์ โดยทั่วไปแล้วอุปกรณ์ที่ใช้ในการอ่านรหัสแถบนั้นจะมีอยู่ 2 ชนิด ได้แก่ ชนิดมือถือและชนิดที่ติดตั้งอยู่กับที่ ซึ่งถ้าเทียบการใช้งานระหว่างการใช้รหัสแถบกับการป้อนข้อมูลเอง พบว่าการใช้รหัสแถบจะมีความน่าเชื่อถือ เพราะใช้อุปกรณ์ในการรหัสแถบ และแปลเป็นข้อมูลซึ่งมีความถูกต้องสูง ต่างจากการป้อนข้อมูลเอง ซึ่งอาจเกิดความผิดพลาดในขณะทำการป้อนข้อมูลได้ง่าย ยิ่งข้อมูลจำนวนมากๆ ด้วยแล้ว จะพบว่ารหัสแถบจะมีความได้เปรียบทั้งในด้านความถูกต้องของข้อมูล และความสะดวกรวดเร็วในการทำงาน รหัสแถบจึงเป็นที่นิยมในการใช้งานในปัจจุบัน

รหัสแถบมีรูปแบบต่างๆมากมาย เช่น รหัส EAN, Code39, UPC, Code25, Codebar เป็นต้น โดยความแตกต่าง จะอยู่ที่การเข้ารหัสแถบสีขาวและดำ แทนค่าข้อมูลอักขระต่างๆ ทั้งตัวเลข และตัวอักษร ซึ่งการถอดรหัสก็ต้องใช้อุปกรณ์ที่อ่านรหัสแถบชนิดนั้นๆ จึงจะสามารถแปลงรหัสเป็นข้อมูลดังเดิมได้ โดยผู้จัดทำได้เลือกรหัสชนิด Code39 มาใช้ เนื่องจากเป็นชนิดเดียวกับที่ใช้ในบัตรนักศึกษาของสถาบัน สำหรับวัตถุประสงค์ในการทำโครงงานนี้คือ เพื่อศึกษาความรู้เกี่ยวกับรหัสแถบ และนำมาประยุกต์ใช้ในการรักษาความปลอดภัยและเก็บข้อมูลนักศึกษาที่เข้าใช้งานห้อง โดยสามารถตรวจสอบข้อมูลการเข้าใช้งานจากฐานข้อมูลในคอมพิวเตอร์ได้

สำหรับภาพรวมของโครงงานนี้ จะเป็นระบบควบคุมการเข้าออกห้อง โดยใช้บัตรรหัสแถบซึ่งผู้ที่สามารถเข้าใช้งานห้องได้นั้น ต้องเป็นผู้ที่มีรหัสอยู่ในฐานข้อมูลเท่านั้น โดยจะมีอุปกรณ์ 2 ส่วน คือ ส่วนที่ติดตั้งภายนอกจะมีเครื่องอ่านรหัสแถบ สำหรับไว้รับค่ารหัสจากบัตรนักศึกษา มีจอแอลซีดี สำหรับแสดงผล และมีส่วนของแป้นกดสำหรับใช้ป้อนรหัสผ่าน ส่วนที่ติดตั้งภายในจะมีส่วนของสเตปมอเตอร์ไว้เป็นกลไกควบคุมการเปิด-ปิด ประตูห้อง ซึ่งสามารถแสดงแผนภาพจำลองระบบได้ดังรูปที่ 1.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 แสดงบล็อกไดอะแกรมของระบบ

โดย ระบบจะประกอบไปด้วยส่วนที่สำคัญ ได้แก่

1. เครื่องอ่านบาร์โค้ด (Barcode Reader)
2. ไมโครคอนโทรลเลอร์ (Microcontroller)
3. หน่วยความจำ (ROM)
4. สเตปปิ้งมอเตอร์ (Stepping Motor)
5. จอแอลซีดี (LCD)
6. คีย์แพด (Keypad)
7. วงจรเซ็นเซอร์ (Sensor)

ซึ่งการทำงานของระบบคือ เครื่องอ่านบาร์โค้ดจะรับค่ารหัสนักศึกษาจากบัตรนักศึกษา ซึ่งรหัสนี้จะถูกส่งมาเปรียบเทียบ กับฐานข้อมูลของผู้ที่ได้รับอนุญาตให้เข้าใช้งาน ที่เก็บอยู่ในหน่วยความจำของไมโครคอนโทรลเลอร์ ซึ่งถ้าไม่มีรหัสในฐานข้อมูล ก็จะแสดงข้อความที่จอแอลซีดีว่าไม่อนุญาตให้เข้า แต่ถ้ามีรหัสในฐานข้อมูล ก็จะถามรหัสผ่านเพื่อเพิ่มความปลอดภัย ถ้ารหัสผ่านถูกต้อง ก็จะอนุญาตให้เข้าใช้ห้อง ซึ่งไมโครคอลโทรลเลอร์ จะสั่งให้สเตปปิ้งมอเตอร์หมุนเพื่อคลายล็อคประตูห้อง หลังจากที่คลายล็อคประตูแล้ว วงจรเซ็นเซอร์จะตรวจสอบสภาวะการปิดประตู เมื่อมีการปิดประตูก็จะสั่งให้สเตปปิ้งมอเตอร์หมุนเพื่อทำการล็อคประตูเข้าสู่สภาวะเริ่มต้นดังเดิมพร้อมทั้งมีการบันทึกรหัสผู้เข้าใช้งานลงในหน่วยความจำเพื่อที่จะสามารถตรวจสอบได้ในภายหลังว่ามีผู้ใดเข้ามาใช้งานบ้าง

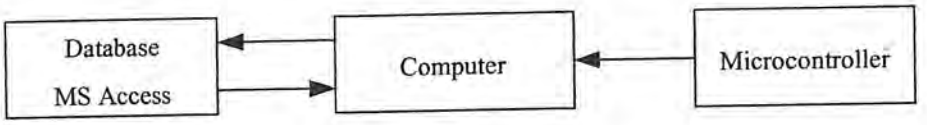
ซึ่งแผนภาพการทำงานของระบบสามารถแสดงได้ดังรูปที่ 1.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 1.2 แสดงแผนภาพการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในกรณีที่หน่วยความจำภายนอก เก็บรหัสผู้ใช้งานเต็มแล้ว สามารถนำตัวเครื่องมาเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์เพื่อทำการถ่ายข้อมูลรหัสของผู้ใช้งานลงในฐานข้อมูลที่อยู่บนคอมพิวเตอร์ได้ โดยมีบล็อกไดอะแกรม การทำงานดังรูปที่ 1.3



รูปที่ 1.3 บล็อกไดอะแกรมการทำงานในการเชื่อมต่อกับคอมพิวเตอร์

โดยการทำงานคือ เมื่อทำการเชื่อมต่อกับคอมพิวเตอร์ ชุดไมโครคอนโทรลเลอร์ จะทำการส่งข้อมูลรหัสนักศึกษา ที่เก็บไว้ในหน่วยความจำภายนอก มาให้คอมพิวเตอร์ ทางด้านคอมพิวเตอร์ จะมีโปรแกรมรองรับการโหลดข้อมูล โดยเมื่อรหัสถูกส่งมา คอมพิวเตอร์ จะนำรหัสมาเปรียบเทียบกับฐานข้อมูลภายในเครื่อง ซึ่งจะนำข้อมูลต่างๆมาแสดงที่หน้าจอ พร้อมบันทึกลงฐานข้อมูลผู้ใช้งาน โดยแผนภาพการทำงานแสดงได้ดังรูปที่ 1.4

รูปที่ 1.4 แผนภาพการทำงานขณะเชื่อมต่อกับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

นับตั้งแต่ที่สหรัฐอเมริกาได้ออกสิทธิบัตรรับรองรหัสแถบในปี ค.ศ. 1949 และ ค.ศ. 1960 ทำให้เกิดรูปแบบของรหัสแถบชนิดต่างๆมากมาย และเริ่มมีการใช้งานเป็นทางการเมื่อปี ค.ศ. 1970 เมื่อคณะกรรมการบริหารด้านห้างสรรพสินค้าของสหรัฐอเมริกา ได้นำรหัสที่ใช้กันมากในสินค้า ออกเผยแพร่ ทำให้มีการใช้งานรหัสแถบอย่างแพร่หลายในสหรัฐอเมริกาและยุโรปในปี ค.ศ. 1973 และ ค.ศ. 1977 ตามลำดับ

การใช้งานรหัสแถบในด้านอื่นๆ เริ่มตั้งแต่ปี ค.ศ. 1980 เมื่อหน่วยป้องกันประเทศของสหรัฐอเมริกาใช้รหัสแถบเพื่อตรวจสอบอุปกรณ์เครื่องมือและยุทโธปกรณ์ ในขณะที่โรงงานอุตสาหกรรมนำไปใช้เพื่องานต่างๆ เช่น ฉลากสินค้า ทำให้คนคุ้นเคยกับรหัสแถบมากขึ้นจากรหัสสินค้าและการชำระเงินด้วยคอมพิวเตอร์ เมื่อใช้รหัสแถบแทนรหัสสินค้าทำให้เกิดความสะดวกมากขึ้นในปี ค.ศ.1981 จึงมีห้างสรรพสินค้ามากกว่า 4000 แห่ง ในแคนาดาและสหรัฐอเมริกานำรหัสแถบมาใช้งาน และนอกจากนี้ยังมีการใช้งานรหัสแถบในด้านอื่นๆ เช่น ห้องสมุด บริการสุขภาพ งานเอกสาร งานการทหาร การวิจัย การผลิตสินค้า ส่วนประเทศไทย ในปัจจุบันได้มีการใช้งานรหัสแถบในงานด้านต่างๆ เช่น งานไปรษณีย์ในการคัดแยกจดหมาย, ใช้แทนรหัสสินค้า บริการห้องสมุด, การเช่า-ยืม หนังสือ วิดีโอ, การชำระค่าบริการต่างๆ, ใช้บนบัตรพนักงานหรือบัตรนักศึกษา เป็นต้น

2.1 ระบบมาตรฐานรหัสแถบ

2.1.1 หลักการรหัสแถบ

รหัสแถบ คือ รูปแบบที่ใช้แทนข้อมูล มีลักษณะเป็นแถบสีดำและแถบสีขาววางเรียงขนานสลับกัน โดยชุดความกว้างของแถบ หรือ จำนวนแถบจะแตกต่างกันไป ขึ้นกับข้อมูลที่ต่างชนิดกัน และชนิดของรหัสที่แตกต่างกันด้วย

การเข้ารหัสของรหัสแถบ (Barcode) แบ่งออกเป็น 2 วิธีการคือ

1. เดลต้าโค้ด (Delta code) จะใช้สีของแถบนำมาเข้ารหัส โดยใช้แถบสีดำแทน "1" และแถบสี ขาว แทน "0"
2. วิดท์โค้ด (Width code) จะใช้ความกว้างของแถบนำมาเข้ารหัส โดยใช้แถบกว้างแทน "1" คำและแถบแคบใช้แทน "0"

2.1.2 การอ่านรหัสแถบ

ในการอ่านรหัสแถบ ใช้หลักการเปลี่ยนรหัสแถบให้กลายเป็นรหัสแอสกี (ASCII code) โดยอาศัยความแตกต่างกันระหว่างแถบเข้ม (แถบสีดำ) และพื้นที่ว่าง (แถบสีขาว) โดยปกติพื้นที่ว่าง จะมีการสะท้อนกลับของแสงได้มากกว่าบริเวณที่เป็นแถบสีเข้ม ตัวอ่านรหัสแถบ (Barcode Reader) จะประกอบด้วยตัวกำเนิดแสง โดยจะถูกส่งผ่านเลนส์ออกมาโดยให้มีจุดรวมแสงเล็กที่สุด กับตัวรับแสงความไวสูงทั้งตัวกำเนิดแสงและตัวรับแสง จะถูกบรรจุไว้ในตัวอ่านเดียวกันซึ่งมีหลายรูปแบบ เช่น รูปแบบปากกา (Wand Type), รูปแบบสล็อต (Slot Type) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอ่านจะสแกนผ่านรหัสแถบ โดยตัวกำเนิดแสงจะกำเนิดแสงส่งผ่านเลนส์ไปกระทบบนรหัสแถบและสะท้อนกลับจากแถบ (แถบสีดำและแถบสีขาว) กลับไปยังตัวรับแสง (Photo sensor) ความไวสูง ที่เกิดค่าความแตกต่างขึ้นตามหลักการสะท้อนกลับในแต่ละแถบ ซึ่งแถบสีขาวจะมีการสะท้อนกลับของแสงมากกว่าแถบสีดำ ซึ่งจะทำให้เกิดสภาวะลอจิก "0" หรือ "1" ซึ่งเมื่อรวมสภาวะลอจิก "0" และ "1" ทั้งหมดตลอดความกว้างของทุกแถบแล้วจะตรงกับรูปแบบที่กำหนดไว้ ก็จะสามารถอ่านออกมาเป็นรหัสตามที่ต้องการได้ ในตัวอ่านรหัสแถบจะใช้ตัวกำเนิดแสงสีแดงหรือสีขาว แต่ส่วนใหญ่นิยมใช้สีแดง เนื่องจากสีขาวต้องการพลังงานและความเข้มของแสงมากกว่าสีแดง

องค์ประกอบที่สำคัญ 2 ประการที่จำเป็นในการอ่านรหัสแถบ

1. พื้นที่ภายในแถบและช่องว่าง จะต้องทำให้เกิดความแตกต่างของการสะท้อนกลับ (Contrast) อย่างมาก เช่นแถบสีดำและสีขาว ซึ่งปกติความแตกต่างนี้ต้องอยู่ในช่วงระหว่าง 80-90 เปอร์เซ็นต์ ขึ้นไป

2. ความกว้างระหว่างแถบกว้างหรือช่องว่างกว้างต่อแถบแคบหรือช่องว่างแคบจะเป็นอัตราส่วน 2:0.5, 2:1 และ 3:1

สำหรับตัวอ่านรหัสแถบ (Barcode Reader) ที่มีใช้งานกัน ชนิดที่นำมาต่อกับคอมพิวเตอร์มีหลายชนิด เช่น แบบที่ต่อกับ RS-422 หรือ RS232 (COM1, COM2) ซึ่งจะมีชุดควบคุมที่สามารถปรับค่าต่างๆได้ตามความต้องการ เช่น ความเร็ว, ชนิดของรหัส เป็นต้น และยังมีชนิดที่ต่อใช้งานแทนคีย์บอร์ด โดยใช้ Keyboard Emulator เป็นตัวควบคุมการทำงาน โดยการทำงานของตัวอ่านรหัสแถบทั้ง 2 ชนิดที่กล่าวมา ให้ผลการทำงานใกล้เคียงกัน สามารถอ่านรหัสได้ 2 ทิศทางโดยไม่ผิดพลาด

2.1.3 ชนิดของรหัสแถบ

ปัจจุบันรหัสแถบที่ใช้งานกันมีอยู่หลายชนิด รหัสแถบที่นิยมใช้แพร่หลายแบ่งได้เป็น

1. รหัส 2 ใน 5 (Code 25)

สำหรับรหัส "2 ใน 5" ซึ่งตามความเป็นมาแล้ว เป็นรหัสชนิดแรกที่ถูกใช้อย่างเป็นกิจจะลักษณะ หนึ่งตัวรหัสจะประกอบด้วยแถบห้าแถบ ซึ่งสองในจำนวนนี้จะมีลักษณะผิดแผกจากที่เหลือ ซึ่งเราจะได้เห็นกันต่อไป รหัสในตระกูลนี้ได้แก่ "2 ใน 5 อุตสาหกรรม", "2 ใน 5 เมตริกซ์" และ "2 ใน 5 สอดแทรก" ทั้งหมดเป็นรหัสแทนตัวเลข

รหัส "2 ใน 5 อุตสาหกรรม" นั้น แถบรหัสหนึ่งจะมีความยาวระหว่าง 1 ถึง 32 ตัว ในรหัสชนิดนี้แถบดำเท่านั้นที่ถือเป็นองค์ประกอบของแถบรหัส โดยแถบดำแคบถือเป็น 0 และแถบดำกว้างถือเป็น 1 รหัส "2 ใน 5 อุตสาหกรรม" นี้ เป็นรหัสที่ง่ายต่อการพิมพ์ แต่ว่าขาดความแน่นอนในการอ่าน ดังนั้นจึงมีการเติมเอาอักษรควบคุมที่ท้ายแถบรหัส รหัสชนิดนี้ใช้กันแพร่หลายในโรงงานอุตสาหกรรมต่าง ๆ บนตั๋วเครื่องบิน และเครื่องแยกจดหมาย

สำหรับรหัส "2 ใน 5 เมตริกซ์" นั้น แถบดำและแถบขาวล้วนถือเป็นองค์ประกอบของรหัส หนึ่งตัวรหัสประกอบด้วยสามแถบดำและสองแถบขาว ระหว่างรหัสแต่ละตัวจะมีช่องไฟคั่น แถบรหัสจะขึ้นต้นและลงท้ายด้วยรหัส 10000 เสมอ การถือเอาแถบขาว ซึ่งก็คือ พื้นที่ที่ใช้ในการพิมพ์รหัสเข้าเป็นส่วนหนึ่งของรหัสทำให้รหัสชนิดนี้กินเนื้อที่น้อยกว่ารหัสชนิดแรก จาก 28 ถึง 33 เปอร์เซ็นต์ ข้อเสียคือความต้านทานต่อความผิดพลาดจะลดต่ำลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัส "2 ใน 5 สอดแทรก" นั้น อาจถือได้ว่าเป็นรหัสที่น่าสนใจที่สุดในรหัสตระกูลนี้ ในรหัสชนิดนี้แถบดำและขาวล้วนถือเป็นองค์ประกอบของรหัสเช่นเดียวกับ "2 ใน 5 เมตริกซ์" แต่จะไม่มีช่องไฟระหว่างรหัส และการใส่รหัสนั้นจะทำในลักษณะ "สอดแทรก" คือ อักษรตัวแรกจะถูกใส่รหัสด้วยรหัส "2 ใน 5 อุตสาหกรรม" โดยใช้แถบดำเป็นตัวประกอบ แต่ตัวอักษรตัวต่อมาจะถูกใส่รหัสด้วย "2 ใน 5 อุตสาหกรรม" ที่ใช้คราวนี้แถบขาวเป็นตัวประกอบ แถบขาวที่ได้มีห้าแถบด้วยกัน คือแบ่งเป็นสองแถบกว้างและสามแถบแคบ ซึ่งจะถูกแทรกเข้าสลับกับแถบดำห้าแถบที่ได้จากการใส่รหัสตัวอักษรแรก แถบรหัสของ "2 ใน 5 สอดแทรก" นี้จะขึ้นต้นด้วยรหัส 0000 และลงท้ายด้วยรหัส 100 เมื่อเทียบกับรหัส "2 ใน 5 อุตสาหกรรม" รหัสชนิดนี้ให้ความหนาแน่นมากกว่าจาก 36 ถึง 42 เปอร์เซ็นต์ และจาก 10 ถึง 12 เปอร์เซ็นต์ เมื่อเทียบกับรหัส "2 ใน 5 เมตริกซ์" มันจึงเป็นที่ใช้กันอย่างแพร่หลายในวงการอุตสาหกรรม

2. รหัส 3 ใน 9 (Code 39)

รหัส 3 ใน 9 เป็นรหัสชนิดแรกที่ใช้แทนตัวอักษรด้วย ปัจจุบันได้มีรหัสซึ่งขยายจากรหัส 3 ใน 9 แล้ว คือ รหัส 128 รหัส 39 นั้น ประกอบด้วยอักขระ 44 ตัว ซึ่งแบ่งเป็นพยัญชนะ 26 ตัว ตัวเลข 10 ตัว และอักษรพิเศษที่เหลือรหัส 39 นี้สามารถถือเป็นรหัส 3 ใน 9 เพราะหนึ่งตัวรหัสประกอบด้วย 9 ตัวประกอบ โดยสามตัวในนั้นจะเป็นแถบกว้าง และอีกสองตัวจะเป็นแถบแคบ หนึ่งแถบรหัสจะมีหนึ่งถึงสามตัวอักษรเท่านั้นซึ่งตามด้วย Check digit ดังนั้นรหัส 3 ใน 9 จึงมีความแน่นอนในการอ่านสูง แต่เปลืองเนื้อที่ รหัสชนิดนี้มีใช้กันมากในอุตสาหกรรมอิเล็กทรอนิกส์โดยใช้ในการแยกชนิดแผงวงจร

3. รหัส Codebar

Codebar เป็นรหัสสำหรับตัวเลขและมีความยาวของแถบรหัสจาก 1 ถึง 32 ตัว เป็นรหัสที่ใช้ในธนาคารเลือดของสหรัฐอเมริกา และในอุตสาหกรรมยาและทางการแพทย์ หนึ่งตัวรหัส ประกอบด้วย 7 บิต ซึ่งแบ่งเป็น 4 แถบดำ และ 3 แถบขาว แถบดำหรือขาวที่แคบแทน 0 และแถบดำหรือขาวกว้างแทน 1

4. รหัส UPC (Universal Product Code), รหัส EAN (European Article Numbering)

รหัส EAN/UPC เป็นรหัสแทนตัวเลขเท่านั้น แถบรหัสหนึ่งประกอบด้วยเลข 8 ตัว หรือ 13 ตัว แต่ขนาด 13 ตัวเป็นแบบที่ใช้กันแพร่หลายมากที่สุด แถบรหัสจะขึ้นต้นและลงท้ายด้วยรหัส 101 เสมอ ตัวเลข 13 หลักนี้จะถูกแบ่งเป็นสามส่วน ส่วนแรกประกอบด้วยเลข 2 ตัว ซึ่งบ่งบอกประเทศ ส่วนที่สองประกอบด้วยเลข 4 ตัว บ่งบอกผู้ผลิตและส่วนสุดท้าย ซึ่งแยกจากส่วนที่สองโดยมีรหัส 01010 เป็นตัวคั่นนั้น จะบ่งบอกรหัสตัวสินค้า รหัสแต่ละตัวจะใช้แถบ 7 แถบ แต่ละแถบมีความกว้างตายตัวเท่ากัน โดยแถบดำคือ 1 และแถบขาวคือ 0 รหัส EAN/UPC นี้เป็นรหัสที่ใช้กับสินค้าอุปโภคบริโภค และเป็นที่ใช้กันแพร่หลายทั่วโลก

ในโครงงานนี้ทางผู้จัดทำเลือกใช้รหัส 3 ใน 9 เนื่องจากเป็นรหัสชนิดเดียวกับบัตรนักศึกษา จึงต้องศึกษาการประมวลผลรหัส 3 ใน 9

2.2 หลักการอ่านและประมวลผลรหัส 3 ใน 9

ตามที่กล่าวมาข้างต้น รหัส 3 ใน 9 เป็นรหัสที่ใช้แทนตัวอักขระทั้งหมด 44 ตัว ประกอบด้วยตัวอักษร 26 ตัว, ตัวเลข 10 ตัว และตัวอักขระพิเศษ 8 ตัว โดยอักขระ 1 ตัวจะแทนด้วยแถบดำ 5 แถบและแถบขาว 4 แถบ รวมเป็น 9 แถบ โดยจะเป็นบิต 1 ทั้งหมด 3 แถบ และเป็นบิต 0 ทั้งหมด 6 แถบ โดยมีส่วนเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Start code) และสิ้นสุด (Stop code) เป็นรหัส * (Asterisk) ดังนั้นรหัสชนิดนี้จะมีการเข้ารหัสตามความกว้าง โดยแถบกว้างจะแทนรหัส "1" และแถบแคบแทนรหัส "0"

โดยเลขฐาน 2 ที่ใช้แทนอักขระทั้ง 44 ตัว แสดงดังตารางที่ 2.1

อักขระ	แถบสีดำ	แถบสีขาว
0	00110	0100
1	10001	0100
2	01001	0100
3	11000	0100
4	00101	0100
5	10100	0100
6	01100	0100
7	00011	0100
8	10010	0100
9	01010	0100
A	10001	0010
B	01001	0010
C	11000	0010
D	00101	0010
E	10100	0010
F	01100	0010
G	00011	0010
H	10010	0010
I	01010	0010
J	00110	0010
K	10001	0001
L	01001	0001

อักขระ	แถบสีดำ	แถบสีขาว
M	11000	0001
N	00101	0001
O	10100	0001
P	01100	0001
Q	00011	0001
R	10010	0001
S	01010	0001
T	00110	0001
U	10001	1000
V	01001	1000
W	11000	1000
X	00101	1000
Y	10100	1000
Z	01100	1000
-	00011	1000
.	10010	1000
SPACE	01010	1000
*	00110	1000
$	00000	1110
/	00000	1101
+	00000	1011
%	00000	0111

ตารางที่ 2.1 แสดงเลขฐาน 2 ที่ใช้แทนอักขระทั้ง 44 ตัว

เนื่องจากรหัสชนิดนี้ไม่มีข้อกำหนดในเรื่องความยาวของข้อมูล ทำให้สามารถใช้แทนข้อมูลที่มีความยาวมากน้อยได้ตามต้องการ และยังสามารถแทนข้อมูลที่มีทั้งตัวอักษรและตัวเลขปนกันได้ จึงเป็นรหัสที่ครอบคลุมการใช้งานได้หลากหลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อักขระที่เราใช้งานจะมีอยู่ 11 อักขระ ได้แก่ 0 ถึง 9 และ * สามารถแสดงตัวอย่างของรหัสแถบได้ตามตารางที่ 2.2

อักขระ	แถบสีดำ	แถบสีขาว	รหัสแถบ 3 ใน 9
0	00110	0100	
1	10001	0100	
2	01001	0100	
3	11000	0100	
4	00101	0100	
5	10100	0100	
6	01100	0100	
7	00011	0100	
8	10010	0100	
9	01010	0100	
*	00110	1000	

ตารางที่ 2.2 แสดงอักขระและรหัสแถบที่นำมาใช้งาน

2.3 ชนิดของหัวอ่านรหัสแถบ

ส่วนประกอบที่สำคัญของเครื่องอ่านรหัสแถบ ที่สำคัญอีกอย่างก็คือ หัวอ่านรหัสแถบซึ่ง ส่วนนี้จะทำหน้าที่ในการอ่านรหัสแถบแล้วแปลงสัญญาณที่ได้เป็นสัญญาณไฟฟ้า เพื่อนำไปแปลงหรือถอดรหัสให้เป็นข้อมูลที่แท้จริง โครงสร้างพื้นฐานของหัวอ่านรหัสแถบแบ่งออกเป็น 2 ชนิด ได้แก่

2.3.1 หัวอ่านชนิดสัมผัสโดยตรง (Contact Scanner)

โดยทั่วไปจะหมายถึง หัวอ่านแบบแวน (Wand Scanner) ซึ่งเป็นหัวอ่านแบบมือถือ มีรูปร่างคล้ายปากกามีขนาดเล็ก พกพาสะดวก สำหรับการอ่านรหัสแถบ หัวอ่านจะต้องสัมผัสกับรหัสแถบโดยตรง โดยทำการรูดหัวอ่านผ่านรหัสแถบ

โครงสร้างของหัวอ่านชนิดนี้ ลักษณะภายนอกคล้ายปากกา ภายในจะประกอบด้วยตัวกำเนิดแสงและตัวรับแสง หลักการทำงานคือตัวกำเนิดแสงจะให้แสงผ่านรูขนาดเล็ก เมื่อแสงตกกระทบรหัสแถบจะเกิดการสะท้อนและดูดกลืนตามคุณสมบัติของแสงที่มีต่อแถบขาวและแถบดำ แสงที่สะท้อนก็จะผ่านกลับมายังตัวรับแสง โดยจะทำการแปลงความเข้มแสงให้เป็นสัญญาณทางไฟฟ้า ซึ่งจะบ่งบอกว่า ขณะนั้นกำลังอ่านรหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แถบส่วนที่เป็นสีขาวหรือสีดำ โดยทั่วไปจะแปลงในรูปลอจิก "0" หรือ "1" สัญญาณส่วนนี้จะถูกส่งไปยังส่วนถอดรหัสเพื่อให้ได้ข้อมูลที่แท้จริงออกมา

สิ่งที่ต้องคำนึงในการใช้หัวอ่านชนิดสัมผัสโดยตรง คือ

1. ระยะห่างระหว่างหัวอ่านกับรหัสแถบ ถ้าระยะห่างมีความคลาดเคลื่อน จะมีผลต่อความถูกต้องแม่นยำในการอ่านรหัสแถบ โดยค่าระยะห่างนี้จะมีค่าเท่ากับระยะโฟกัสของหัวอ่าน

2. รหัสแถบโดยทั่วไป จะมีการเคลือบด้วยพลาสติก เพื่อป้องกันจากสิ่งสกปรกและรอยขีดข่วนซึ่งความหนาของพลาสติกต้องไม่หนามากเกินไป จนทำให้รหัสแถบห่างจากหัวอ่านมากกว่าระยะโฟกัสซึ่งจะทำให้การอ่านผิดพลาด

3. ความคลาดเคลื่อนอันเนื่องมาจาก ความเร็วในการรูดหัวอ่านผ่านรหัสแถบไม่สม่ำเสมอ ดังนั้นในการถอดรหัสต้องมีการชดเชยผลที่เกิดจากความเร็วในการรูดไม่สม่ำเสมอ แหล่งกำเนิดแสงภายในหัวอ่านแบบแวน โดยทั่วไปจะใช้ ไดโอดเปล่งแสง (LED) ใช้ความยาวคลื่นช่วง 630-720 nm (แสงสีแดงถึงแดงเข้ม) หรือในช่วง 720-900 nm (ในย่านอินฟราเรด มองไม่เห็นด้วยตาเปล่า)

2.3.2 หัวอ่านชนิดไม่สัมผัสโดยตรง (Non-contact Scanner)

หัวอ่านชนิดนี้ไม่จำเป็นต้องสัมผัสกับรหัสแถบโดยตรง สามารถอ่านรหัสได้โดยหัวอ่านอยู่ห่างจากรหัสแถบ ยังเป็นการลดความผิดพลาดจากผลของระยะโฟกัสของลำแสง แบ่งออกได้เป็น 2 ชนิด

1. หัวอ่านชนิดไม่สัมผัสโดยตรงแบบแอกทีฟ (Active non-contact Scanner)

เป็นหัวอ่านที่อาศัยหลักการของเลเซอร์มักพบในลักษณะที่เป็นมือถือ และแบบที่ตั้งอยู่กับที่และลักษณะการอ่านของหัวอ่านแบบนี้จะมี 2 ลักษณะคือ แบบที่กวาดลำแสงผ่านรหัสแถบด้วยตัวเอง (Self scanner) และแบบที่ลำแสงอยู่นิ่ง แล้วให้รหัสแถบวิ่งผ่านโดยการกวาดของลำแสงผ่านรหัสแถบ จะมีอัตราการกวาดประมาณ 40-800 ครั้งต่อวินาทีขึ้นอยู่กับลักษณะของหัวอ่าน ถ้าเป็นหัวอ่านแบบที่มือถือ จะมีอัตราการกวาดของลำแสงต่ำเพราะใช้อ่านรหัสแถบที่ติดอยู่กับที่ แต่ถ้าเป็นแบบที่ต้องอ่านรหัสแถบที่มีการเคลื่อนที่ตลอดเวลา เช่นการอ่านรหัสแถบของสินค้าในสายพานการผลิต อัตราการกวาดของลำแสงจะต้องสูง

2. หัวอ่านชนิดไม่สัมผัสโดยตรงแบบพาสสีฟ (Passive non-contact scanner)

การทำงานของหัวอ่านชนิดนี้ จะอาศัยหลักการคล้ายกับการถ่ายภาพคือ ใช้แสงแฟลชฉายลงบนรหัสแถบ โดยคุณสมบัติของแสงที่มีต่อแถบสีขาวและสีดำ จะทำให้เกิดการสะท้อนและไม่สะท้อนแสงกลับไปสู่ส่วนรับแสงที่ถูกจัดเรียงกันในแนวเส้นตรงเป็นอาเรย์ (Array) แสงที่ตกบนอาเรย์แต่ละตัวจะถูกเปลี่ยนให้เป็นสัญญาณทางไฟฟ้า และถูกส่งไปในลักษณะอนุกรม เพื่อนำไปประมวลผลให้ได้ข้อมูลที่แท้จริงต่อไป

การใช้หัวอ่านแบบเฟลช เพื่อเพิ่มระยะห่างในการอ่านรหัสแถบ ให้กว้างมากขึ้น ทำให้สามารถอ่านรหัสแถบที่อยู่ห่างจากหัวอ่านได้ แต่มีข้อจำกัดที่ความกว้างของรหัสแถบ ต้องมีความแน่นอนหรือมีจำนวนข้อมูลที่แน่นอนนั่นเอง แต่มีข้อดีตรงที่จะมีความถูกต้องสูงกว่าแบบแวนที่ต้องใช้มือรูดหัวอ่านเองและดีกว่าแบบเลเซอร์ตรงที่ราคาประหยัดกว่าโดยที่ประสิทธิภาพใกล้เคียงกัน มี 2 ชนิดคือ แบบมือถือและติดตั้งอยู่กับที่ โดยปกติแบบมือถือมีอัตราการกวาดประมาณ 3-5 ครั้งต่อวินาที และแบบติดอยู่กับที่อัตรากวาด 7-10 ครั้งต่อวินาที

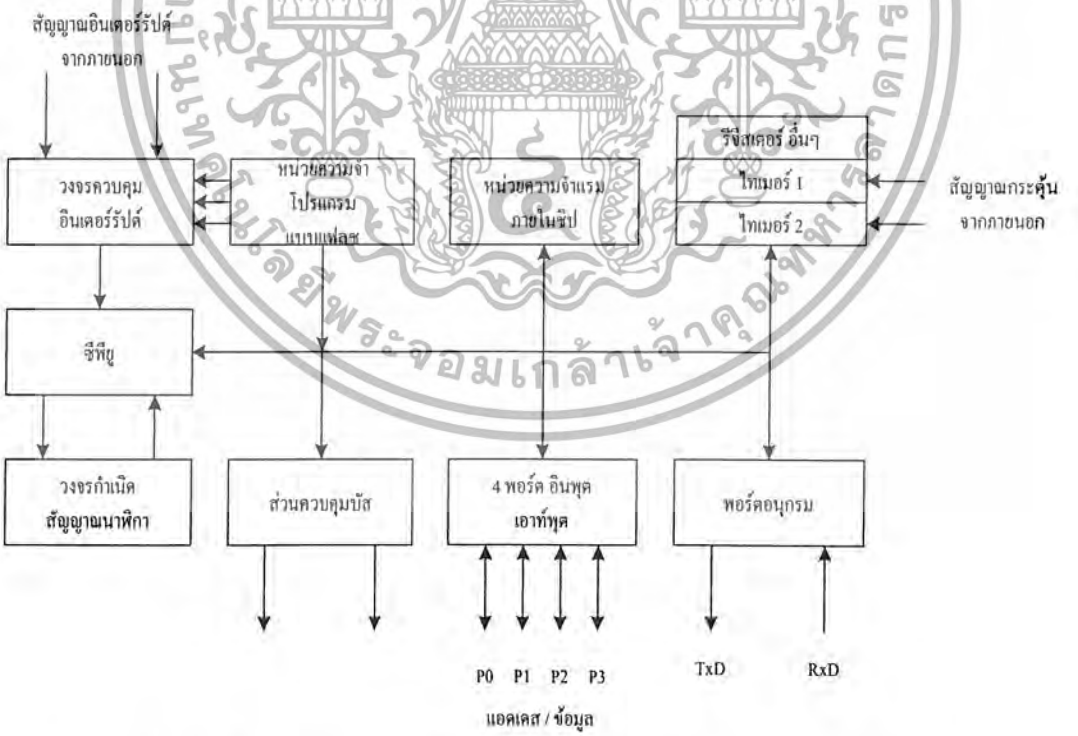
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 โครงสร้างของ MCS-51

2.4.1 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 อนุกรม AT89CXX

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลช สามารถลบ และเขียนใหม่ได้ 1000 ครั้ง
- หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความจำแบบอีอีพรอมเพิ่มเติม
- ขาพอร์ต เป็นแบบ 2 ทิศทาง สามารถใช้เป็นได้ทั้งอินพุต และเอาท์พุต
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
- มีไทเมอร์ / เคาท์เตอร์ ขนาด 16 บิตอย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัพต์ ได้ 16 แหล่ง
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป

ในรูปที่ 2.1 เป็นโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51ในอนุกรม AT89CXX ซึ่งจะเห็นได้ว่าโครงสร้างของ AT89CXX จะเหมือนกับไมโครคอนโทรลเลอร์ MCS-51 ตระกูลพื้นฐาน ซึ่งจะต่างกันเพียง หน่วยความจำโปรแกรมแบบแฟลชที่เพิ่มเติมเข้ามา หากเป็นไมโครคอนโทรลเลอร์ในตระกูล 87XX หน่วยความจำภายในจะเป็นแบบอีพรอม และในบางเบอร์จะสามารถโปรแกรมได้เพียงครั้งเดียว



รูปที่ 2.1 แสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชในอนุกรม AT89CXX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ทุกเบอร์จะมีสถาปัตยกรรม และขาที่ใช้งานพื้นฐานเหมือนกัน ดังรูปที่ 2.2 โดยมีรายละเอียดของขาต่างๆดังนี้

ขา VCC ใช้สำหรับต่อไฟเลี้ยง +5 โวลต์

ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 0 (P0.0-P0.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุต และเอาท์พุต สำหรับใช้งานทั่วไป ถ้าต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ต ที่ต้องการจะติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (Float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนี้ขาพอร์ตนี้ ยังถูกใช้งานในการติดต่อกับขาแอดเดรสต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วยเพื่อสลับการทำงานทำให้สามารถติดต่อได้ทั้งขาแอดเดรส และขาข้อมูล

ขาพอร์ต 1 (P0.0-P0.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุต และเอาท์พุต สำหรับใช้งานทั่วไป ถ้าต้องการกำหนดให้ขาพอร์ต 1 ขาใดขาหนึ่งเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ต ที่ต้องการจะติดต่อด้วย

ขาพอร์ต 2 (P0.0-P0.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุต และเอาท์พุต สำหรับใช้งานทั่วไป ถ้าต้องการกำหนดให้ขาพอร์ต 2 ขาใดขาหนึ่งเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ต ที่ต้องการจะติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (Float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนี้ขาพอร์ตนี้ ยังถูกใช้งานในการติดต่อกับขาแอดเดรสสูงของหน่วยความจำภายนอก (A8-A15)

ขาพอร์ต 3 (P0.0-P0.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุต และเอาท์พุต สำหรับใช้งานทั่วไป ถ้าต้องการกำหนดให้ขาพอร์ต 3 ขาใดขาหนึ่งเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ต ที่ต้องการจะติดต่อด้วยด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (Float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังมีหน้าที่การใช้งานพิเศษอีกดังต่อไปนี้

ขา P3.0 ใช้เป็นขาอินพุตสำหรับรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือ ขา RxD

ขา P3.1 ใช้เป็นขาเอาท์พุตสำหรับรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือ ขา TxD

ขา P3.2 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัพต์จากภายนอกช่อง 0 หรือ ขา INT0

ขา P3.3 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัพต์จากภายนอกช่อง 1 หรือ ขา INT1

ขา P3.4 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัพต์จากไทเมอร์ 0 หรือ ขา T0

ขา P3.5 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัพต์จากไทเมอร์ 1 หรือ ขา T1

ขา P3.6 ใช้เป็นขาสัญญาณ WR ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

ขา P3.7 ใช้เป็นขาสัญญาณ RD ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

ขารีเซ็ต (Reset) ใช้ในการรีเซ็ตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อทำการรีเซ็ตสถานะ ที่ขานี้ต้องอยู่ในระดับรีเซ็ตอย่างน้อย 2 แมชชีนไซเคิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องตามปกติต่อไป

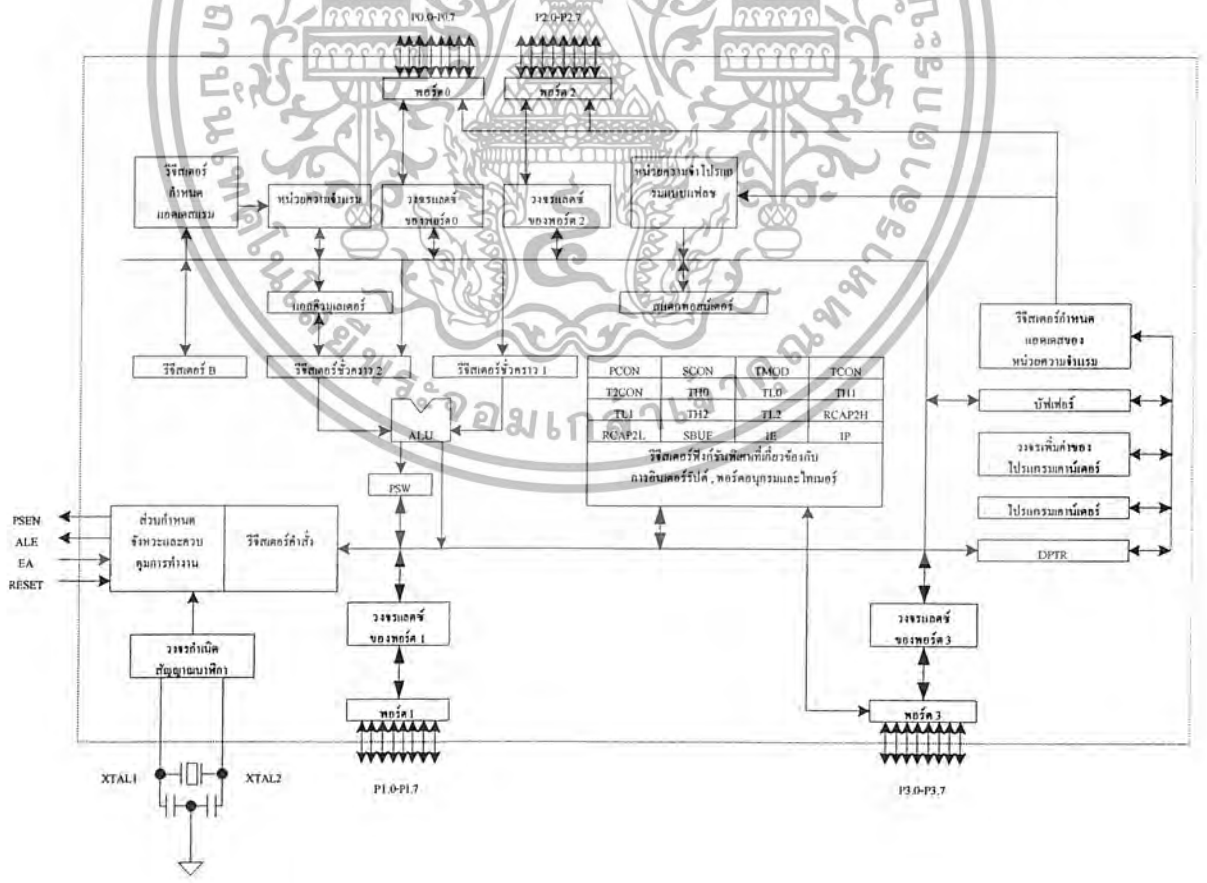
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา ALE/PROG (Address Latch Enable/Program Pulse Input) เป็นขาที่ใช้ในการควบคุมการแลตช์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ ยังใช้เป็น ขาสำหรับพัลส์ ของโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีพรอม

ขา PSEN (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำภายนอก ตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละแมชชีนไซเคิล แต่ถ้าหากติดต่อกับหน่วยความจำภายนอก ขานี้จะไม่มีการส่งสัญญาณใดๆออกมา

ขา EA/VPP (External Access Enable/Programmable Voltage Input) ใช้สำหรับการติดต่อหน่วยความจำภายนอก หรือหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น "0"จะเป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อหน่วยความจำภายนอก แต่ถ้าหากขานี้เป็น "1" จะเป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อหน่วยความจำภายใน นอกจากนี้ ที่ขานี้ยังใช้เป็นอินพุต สำหรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม +12 โวลต์

ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตอลออสซิลเลเตอร์ เพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะในการทำงานของไมโครคอนโทรลเลอร์



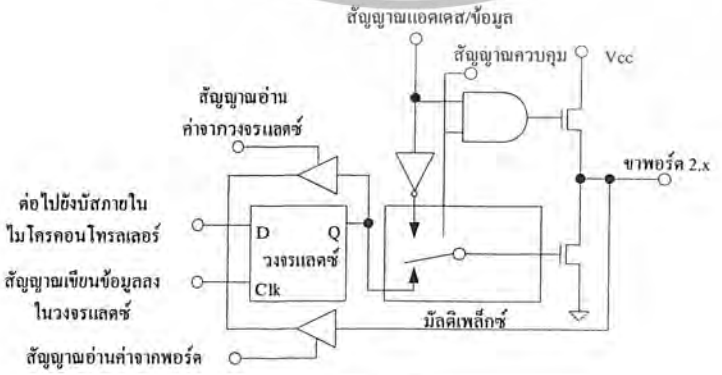
รูปที่ 2.2 แสดงรายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชของ Atmel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 โครงสร้างและการทำงานของพอร์ต

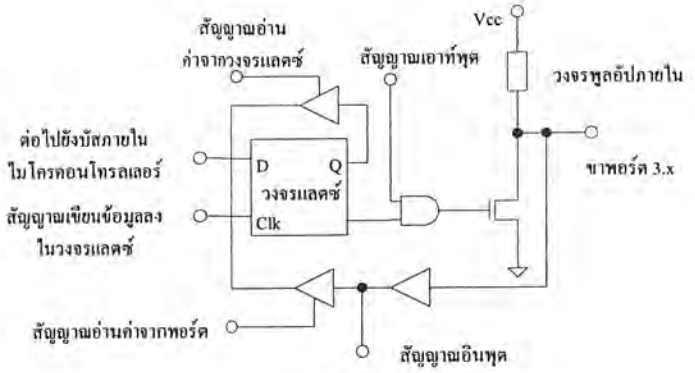
ไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชมีพอร์ตให้ใช้งานทั้งสิ้น 4 พอร์ต คือ พอร์ต 0, พอร์ต 1, พอร์ต 2, พอร์ต 3 แต่ละพอร์ตมีขนาด 8 บิต เป็นพอร์ตแบบ 2 ทิศทางคือสามารถเป็นได้ทั้งอินพุตสำหรับรับสัญญาณข้อมูลเข้า และเอาท์พุตสำหรับส่งสัญญาณข้อมูลออกไปทุกพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชจะมีวงจรแลตช์ และวงจรขับ ตลอดจนบัฟเฟอร์อินพุต ดังแสดงได้ใน รูปที่ 2.3

ที่พอร์ต 0 และพอร์ต 2 จะใช้งานเป็นพอร์ตอินพุต และเอาท์พุตสำหรับใช้งานทั่วไป และใช้ในการติดต่อกับหน่วยความจำภายนอก สำหรับพอร์ต 1 และพอร์ต 3 บางขานอกจากจะใช้เป็นขาอินพุต และเอาท์พุตตามปกติแล้ว ยังสามารถใช้งานในหน้าที่พิเศษได้อีก ขึ้นอยู่กับว่าเป็นไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชเบอร์ใด



(3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงวงจรภายในของพอร์ตในไมโครคอนโทรลเลอร์ MCS-51แบบแฟลช

โดยในรูปที่ 2.3.1 เป็นวงจรของพอร์ต0 วงจรแลตช์ของแต่ละบิตในแต่ละพอร์ตก็คือวงจรดีฟลิบฟลอบนั่นเอง การอ่านค่าสถานะของพอร์ต และสถานะของวงจรแลตช์สามารถกระทำได้อย่างอิสระด้วยสัญญาณที่แยกจากกัน คือสัญญาณอ่านข้อมูลจากขาพอร์ต และสัญญาณอ่านข้อมูลจากวงจรแลตช์ ส่วนการเขียนข้อมูลมายังพอร์ตต้องส่งสัญญาณมายังขา CLK ของดีฟลิฟลอบในขณะที่ข้อมูลจะผ่านมาทางขาบัสข้อมูลภายในเข้าสู่ขา D ของดีฟลิบฟลอบ ที่พอร์ตนี้มีวงจรมัลติเพล็กซ์สำหรับกำหนดลักษณะการทำงานของพอร์ต ว่าต้องการใช้งานเป็นขาพอร์ตอินพุตเอาท์พุตตามปกติ หรือ ต้องการใช้ในการติดต่อกับหน่วยความจำภายนอก ไมโครคอนโทรเลอร์

เนื่องจากที่ขาพอร์ต 0 ไม่มีวงจรพูลอัปภายใน หากมีการนำพอร์ต 0 ไปใช้งานเป็นพอร์ตอินพุตจะต้องต่อตัวต้านทานพูลอัปภายนอกที่ขาพอร์ต 0 ทุกขาด้วย ส่วนในรูปที่ 2.3.2 เป็นวงจรของพอร์ต 1 จะมีลักษณะทั่วไปเหมือนพอร์ต0 แต่จะไม่มีวงจรมัลติเพล็กซ์ เนื่องจากพอร์ตไม่ใช้ในการติดต่อกับหน่วยความจำภายนอก แต่จะมีวงจรพูลอัปภายในที่แต่ละบิตแทน

ส่วนใน รูปที่ 2.3.3 เป็นวงจรของพอร์ต 2 จะเหมือนกับพอร์ต 0 มาก ต่างกันที่มีวงจรพูลอัปเพิ่มเติมเข้ามา และในรูปที่ 2.3.4 เป็นวงจรของพอร์ต 3 จะเหมือนกับพอร์ต 0 มาก ต่างกันที่มีวงจรบัฟเฟอร์ และวงจรอินพุตเอาท์พุตเมื่อทำงานในฟังก์ชันพิเศษเข้ามา เนื่องจากพอร์ต 3 สามารถนำไปใช้งานในหน้าที่พิเศษทุกขา

การใช้งานเป็นพอร์ตอินพุต

ในการกำหนดพอร์ตให้เป็นอินพุตพอร์ตต้องเริ่มต้นด้วยการเขียนข้อมูล "1"มาที่แต่ละบิตของพอร์ตที่ใช้งานเป็นอินพุตพอร์ต เมื่อหยุดการทำงานของเฟตที่ใช้ในการขับสัญญาณเอาท์พุตของบิตนั้นๆ ทำให้ขาสัญญาณของพอร์ตเชื่อมต่อเข้ากับวงจรพูลอัปภายในโดยตรง ส่งผลให้ขาพอร์ตนั้นๆมีลอจิกเป็น "1" สามารถรับสัญญาณลอจิก "0" จากอุปกรณ์ภายนอกได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาแล้วเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ต แล้วรอให้ซีพียูมาอ่านค่าไป เมื่อเป็นเช่นนี้แล้วสัญญาณข้อมูลจากอุปกรณ์ภายนอกที่ติดต่อกับไมโครคอนโทรลเลอร์ MCS–51ควรจะทำงานในสภาวะ ลอจิก "0" จึงจะดีและสะดวกที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานเป็นพอร์ตเอาท์พุต

โดยปกติแล้ว ขาพอร์ตจะถูกกำหนดให้เป็นเอาท์พุตพอร์ตอยู่แล้ว จึงสามารถส่งข้อมูลออกไปได้เลย คือ เมื่อต้องการส่งข้อมูล "0" ก็เขียนข้อมูล "0"ไปยังวงจรแลตช์ ซึ่งจะทำการส่งต่อไปยังวงจรขับเฟต ทำให้เฟตทำงาน ที่ขาพอร์ตที่กำหนดก็จะมีค่าลอจิกเป็น "0" และถ้าต้องการส่งข้อมูล "1" ออกไปก็เขียนข้อมูล "1" ออกไปยังวงจรแลตช์ ซึ่งจะทำการส่งต่อไปยังวงจรขับเฟต ทำให้เฟตทำงาน ที่ขาพอร์ตที่กำหนดก็จะมีค่าลอจิกเป็น "1" ซึ่งเหมือนกับการกำหนดให้เป็นพอร์ตอินพุต เพียงแต่ไม่มีการการอ่านข้อมูลที่บัฟเฟอร์ ยกเว้นกรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกทางเอาท์พุตพอร์ต

เมื่อใช้งานเป็นเอาท์พุตพอร์ต แต่ละขา (หรือแต่ละบิต) ของแต่ละพอร์ตมีความสามารถในการจ่ายกระแส หรือที่เรียกว่า แหล่งจ่ายกระแส (Current Source) ได้สูงสุด 10 มิลลิแอมป์ และทุกขารวมกันในแต่ละพอร์ตสูงสุด 26 มิลลิแอมป์ สำหรับพอร์ต 0และ15 มิลลิแอมป์ สำหรับพอร์ต 1 ถึงพอร์ต 3 และในกรณีที่ใช้ทุกพอร์ตเป็นเอาท์พุตพอร์ต จะสามารถจ่ายกระแสได้รวมกันสูงสุด 71 มิลลิแอมป์ ดังนั้นในการใช้งานเป็นเอาท์พุตพอร์ตต้องคำนึงถึงความสามารถในการจ่ายกระแสด้วย โดยอาจจะต่อวงจรบัฟเฟอร์ไว้ที่เอาท์พุตเพื่อช่วยในการขับกระแสอีกทางหนึ่ง

2.5 พื้นฐานการติดต่อสื่อสารข้อมูลแบบอนุกรม

2.5.1 การอินเตอร์เฟสผ่านพอร์ตอนุกรม

การสื่อสารข้อมูลอนุกรมเป็นการรับหรือส่งข้อมูลในลักษณะของบิตหรือกลุ่มบิต คราวละ 1 บิต เป็นลำดับเรียงกันไปเรื่อยๆ จนสิ้นสุดข้อมูลโดยการสื่อสารข้อมูลอนุกรมจะใช้สายสัญญาณเพียง 2 หรือ 3 เส้นเท่านั้น ซึ่งจะแตกต่างจากการสื่อสารแบบขนานที่มีการถ่ายโอนข้อมูลมาพร้อมๆกันจึงต้องใช้สายสัญญาณจำนวนมาก ทำให้ไม่เหมาะกับการสื่อสารเพื่อรับส่งข้อมูลกับอุปกรณ์ภายนอกที่อยู่ในระยะทางไกลๆ

2.5.2 จังหวะเวลาของการสื่อสารข้อมูลอนุกรม

เนื่องจากการสื่อสารข้อมูลอนุกรมเป็นการรับหรือส่งข้อมูลในลักษณะของบิตหรือกลุ่มบิต เราจึงจะพิจารณาถึงเรื่องอัตราเร็วในการรับส่งข้อมูลเหล่านี้เป็นอันดับแรก โดยทั่วไปจะระบุเป็นหน่วยของจำนวนบิตข้อมูลภายในเวลา 1 วินาที เรียกว่า อัตราบอด (Baud Rate) ตามค่ามาตรฐานจะมีอัตราบอดดังนี้ 110, 150, 300, 1200, 2400, 4800 และ 9600 บิตต่อวินาที

2.5.3 รูปแบบของข้อมูลอนุกรม

รูปแบบของการส่อสารข้อมูลอนุกรม จะมีการเพิ่มเติมบิตข้อมูลบางอย่าง ร่วมไปกับการส่งข้อมูลจริง เพื่อความถูกต้องในการสื่อสารให้มีความถูกต้องมากขึ้นโดยการส่งข้อมูลจะประกอบด้วย บิตเริ่มต้น, ข้อมูลที่จะส่งไป, บิตแสดงภาวะความเป็นเลขคู่หรือเลขคี่ , บิตสุดท้าย โดยบิตแสดงภาวะความเป็นเลขคู่หรือเลขคี่อาจจะมีหรือไม่ก็ได้ขึ้นอยู่กับรูปแบบในการรับส่งข้อมูลที่ใช้

1. บิตเริ่มต้น (Start Bit) ทำหน้าที่บอกให้วงจรทางฮาร์ดแวร์ทางด้านรับทราบว่าถึงตำแหน่งจุดเริ่มต้นของข้อมูลที่ส่งมาใหม่ เพื่อที่จะปรับจังหวะของสัญญาณรับข้อมูลให้ตรงกัน ดังนั้นบิตเริ่มต้นนี้จะถูกเพิ่มเข้าไปก่อนที่จะมีการส่งข้อมูลจริง ตามปกติมักจะเป็นระดับลอจิกที่ตรงกันข้ามกับสถานะของลอจิกของสายส่งสัญญาณขณะที่ไม่มีการส่งข้อมูล (Idel State)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2. บิตแสดงภาวะความเป็นเลขคู่หรือเลขคี่ (Parity Bit) ทำหน้าที่ตรวจสอบความถูกต้องในการส่งข้อมูล โดนทั่วไปจะเรียกว่า บิตพาริตี และจะนำไปแทรกไว้ต่อท้ายบิตสุดท้ายของข้อมูลจริง ค่าของบิตนี้จะขึ้นอยู่กับจำนวนของบิตของข้อมูลที่เป็น 1 ซึ่งจะเป็นไปได้ 2 ลักษณะ คือ พาริตีคู่ (Even Parity) หรือพาริตีคี่ (Odd Parity) เช่น ถ้าระบบที่มีการติดต่อโดยระบุว่าจะใช้พาริตีคู่ในการส่ง ทางด้านส่งจะนำข้อมูลที่จะส่งมาตรวจสอบหากจำนวนบิตที่เป็น 1 มีจำนวนคู่อยู่แล้ว ค่าของบิตพาริตีจะมีค่าเป็น 0 และถ้าหากจำนวนบิตที่เป็น 1 มีจำนวนคี่ ค่าของบิตพาริตีจะมีค่าเป็น 1 และในทางด้านรับก็จะทำการตรวจสอบจำนวนบิตที่มีค่าเป็น 1 ของข้อมูลทั้งหมดรวมทั้งบิตพาริตีด้วย ถ้ามีจำนวนบิตที่มีค่าเป็น 1 เป็นจำนวนคู่ แสดงว่าการรับส่งข้อมูลเข้ามาถูกต้อง แต่ถ้าไม่เป็นจำนวนคู่ แสดงว่าการรับส่งข้อมูลเข้ามามีความผิดพลาดของการส่งขึ้น

3. บิตสุดท้าย (Stop bit) ทำหน้าที่บอกให้วงจรทางฮาร์ดแวร์ทางด้านรับทราบว่าถึงตำแหน่งจุดสิ้นสุดของข้อมูลที่ส่งมา บิตสุดท้ายนี้อาจจะมีจำนวนมากกว่า 1 บิตก็ได้ โดยจะแทรกเข้ามาที่ท้ายสุดของข้อมูลที่จะส่งไปทางด้านรับ

2.5.4 การจัดการข้อมูลอนุกรมของ MCS–51

การพอร์ตอนุกรมของ MCS-51มีโครงสร้างการทำงานที่เรียกว่า ฟูลดูเพล็กซ์ (Full Duplex) หมายถึงความสามารถในการรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน โดยวงจรทางด้านส่ง (Transmission) ประกอบด้วยรีจีสเตอร์ SBUF ที่ทำหน้าที่ในการเก็บข้อมูลที่จะส่งออก การใช้คำสั่งเขียน หรือ ถ่ายโอนข้อมูลมายังรีจีสเตอร์นี้จะเป็นการส่งข้อมูลนั้นออกไปยังพอร์ตอนุกรมทางขา Tx (พอร์ต 3.1) โดยอัตโนมัติ ส่วนทางด้านรับ (Receiver) ประกอบด้วยรีจีสเตอร์ SBUF เช่นเดียวกัน แต่ทำหน้าที่นำข้อมูลที่นำมาจากส่วนของวงจรเลื่อนบิตหรือชิพรีจีสเตอร์ (Shift Register) ของวงจรจัดการข้อมูลอนุกรมภายใน สัญญาณข้อมูลอนุกรมที่รับเข้ามาจะผ่านทางขาสัญญาณ Rx (พอร์ต 3.0)

การพอร์ตอนุกรมของ MCS-51สามารถโปรแกรมให้ทำหน้าที่ในรูปแบบต่างๆได้ 4 แบบ หรือเรียกว่า 4 โหมดการทำงานซึ่งจะสามารถแสดงได้ตามตารางที่ 2.3 โดยการกำหนดบิต SM0 และ SM1 ซึ่งอยู่ภายในรีจีสเตอร์ควบคุมและบอกสถานะ SCON ดังแสดงได้ในตารางที่ 2.4

โหมดการทำงาน	คำอธิบาย
Mode 0	เป็นการขยายพอร์ตอินพุต เอาท์พุต โดยทำงานร่วมกับ ไอซีชิพรีจีสเตอร์ภายนอก ประเภททีทีแอล และซีมอส
Mode 1	ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART (Universal Asynchronous Receiver / Transmission) โดยใช้กลุ่มข้อมูลแบบ 10 บิต และสามารถเปลี่ยนแปลงความเร็วในการส่งข้อมูลได้
Mode 2	ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART โดยใช้กลุ่มข้อมูลแบบ 11 บิต และกำหนดความเร็วในการส่งข้อมูลคงที่
Mode 3	ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART โดยใช้กลุ่มข้อมูลแบบ 11 บิต และสามารถเปลี่ยนแปลงความเร็วในการส่งข้อมูลได้

ตารางที่ 2.3 แสดงโหมดการทำงานของพอร์ตอนุกรมของ MCS – 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
54953

ชื่อบิต : SCON ตำแหน่ง : 98H ค่าเริ่มต้น : 0000 0000H

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

ชื่อบิต	ตำแหน่ง	ความหมาย
SM0	SCON.7	บิตเลือกโหมดการทำงาน
SM0	SCON.6	บิตเลือกโหมดการทำงาน
SM0	SCON.5	แฟล็กกำหนดการทำงานแบบมัลติโปรเซสเซอร์
REN	SCON.4	แฟล็กยอมให้มีการรับข้อมูล
TB8	SCON.3	ค่าของบิตที่ 9 สำหรับการส่งข้อมูลออก
RB8	SCON.2	ค่าของบิตที่ 9 สำหรับการรับข้อมูลเข้า
TI	SCON.1	แฟล็กแสดงการอินเตอร์รัพต์ภายหลังการส่งข้อมูล
RI	SCON.0	แฟล็กแสดงการอินเตอร์รัพต์ภายเมื่อมรการรับข้อมูลเข้า

ตารางที่ 2.4 แสดงการกำหนดบิต SM0 และ SM1 ซึ่งอยู่ภายในรีจีสเตอร์ควบคุมและบอกสถานะ SCON

2.5.5 การอินเตอร์รัพต์ของการสื่อสารอนุกรม

เนื่องจากการรับส่งข้อมูลอนุกรมในการส่งข้อมูลไบต์หนึ่งๆ จะใช้เวลานานหลายมิลลิวินาที ดังนั้นเพื่อประสิทธิภาพจึงได้มีการกำหนดบิตหรือแฟล็กของสถานะที่เกี่ยวข้องทั้งหมดจัดรวมอยู่ในรีจีสเตอร์ SCON เท่านั้น เช่น แฟล็ก RI ที่มีค่าเป็น 1 จะมีผลแจ้งให้ทราบว่าได้รับข้อมูลเข้ามาทางพอร์ตอนุกรม เมื่อแฟล็กตัวใดมีค่าเป็น 1 ก็จะมีผลทำให้เกิดการอินเตอร์รัพต์ขึ้น ดังนั้นภายในโปรแกรมจะต้องทำการตรวจสอบจากสภาวะของแฟล็กเหล่านี้เองว่าเกิดการอินเตอร์รัพต์ขึ้นจากสาเหตุใด จากนั้นค่อยกำหนดค่า 0 ให้กับแฟล็กนั้น ซึ่งจะแตกต่างจากการการอินเตอร์รัพต์จากสัญญาณอื่นๆ เช่น วงจรนับ/จับเวลา ที่มีการกำหนดค่า 0 ให้กับแฟล็กสถานะที่เกี่ยวข้องโดยอัตโนมัติ ภายหลังจากที่ได้เข้าไปทำงานในส่วนของโปรแกรมย่อยบริการอินเตอร์รัพต์

2.5.6 กระบวนการรับ และส่งข้อมูลอนุกรมของ MCS–51

การส่งข้อมูลออกทางพอร์ตอนุกรมของ MCS-51จะเริ่มต้นภายหลังการเขียนข้อมูลลงในรีจีสเตอร์ SBUF ข้อมูลนี้จะถูกจัดการด้วยวิธีทางฮาร์ดแวร์ในการเลื่อนบิต และส่งสัญญาณออกไปภายนอกโดยอัตโนมัติ เมื่อข้อมูลเหล่านี้ถูกส่งออกไปจนครบถ้วนแล้ว จึงทำการกำหนดค่า 1 ให้กับแฟล็ก TI เพื่อแจ้งให้ทราบว่าขณะนี้รีจีสเตอร์ว่าง พร้อมที่จะส่งข้อมูลไบต์ต่อไปสำหรับการรับข้อมูลจากพอร์ตอนุกรมจะต้องเริ่มต้นโดยการกำหนดค่าบิต REN (Receiver Enable) ให้มีค่าเป็น 1 ก่อน หลังจากนั้นเมื่อมีการรับข้อมูลเข้ามาจากภายนอก ระบบฮาร์ดแวร์ก็จะทำการเลื่อนบิตข้อมูลเข้ามาโดยอัตโนมัติ และเมื่อบิตสุดท้ายถูกเลื่อนเข้ามาเรียบร้อยแล้ว ข้อมูลนั้นก็จะถูกย้ายมาเก็บไว้ที่รีจีสเตอร์ SBUF และทำการกำหนดค่า 1 ให้กับแฟล็ก RI มีผลให้เกิดการอินเตอร์รัพต์โปรแกรมขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

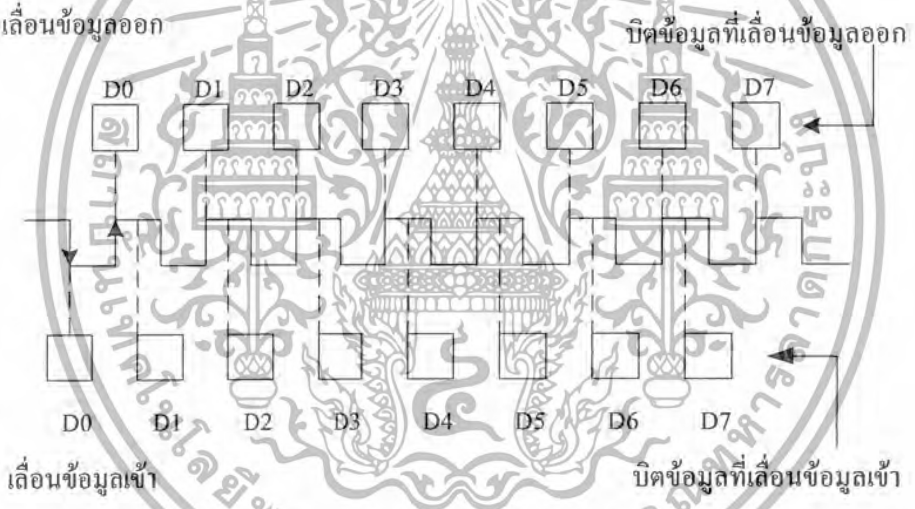
2.5.7 โหมดการทำงานของการสื่อสารผ่านพอร์ตอนุกรม

โหมด 0

การสื่อสารผ่านพอร์ตอนุกรมในโหมด 0 เป็นการขยายพอร์ตอินพุต เอาท์พุต ให้มีจำนวนมากขึ้น โดยจะทำการสร้างสัญญาณนาฬิกาขึ้นเพื่อให้เข้าจังหวะของการทำงานที่พร้อมกัน (Synchronizing) สำหรับการเลื่อนบิตเข้าหรือออกจากไอซีรีจิสเตอร์ภายนอก เมื่อมีการโอนย้ายข้อมูลมายังรีจิสเตอร์ในแต่ละครั้งก็จะมีผลทำให้เกิดการส่งบิตข้อมูลทั้ง 8 บิตออกมา แม้ว่าแฟล็กสถานะ TI จะยังมีค่าเป็น 1 อยู่ก็ตาม นอกจากนี้เมื่อค่าของแฟล็กสถานะ RI มีค่าเป็น 1 ก็ควรจะย้ายข้อมูลออกไปยังรีจิสเตอร์ SBUF เสียก่อนที่จะมีการกำหนดค่าแฟล็ก RI เป็น 0 เพื่อรับข้อมูลถัดไป

การทำงานของพอร์ตอนุกรมในโหมด 0 เป็นการรับส่งข้อมูลอนุกรม 8 บิตโดยใช้ขาสัญญาณ RxD เท่านั้น ส่วนขา TxD จะนำไปใช้เป็นขาสัญญาณนาฬิกาในการบอกให้จังหวะการเลื่อนข้อมูลกับการเลื่อนบิตภายนอก อัตราการเลื่อนบิตหรืออัตราบอดถูกกำหนดไว้เป็นค่าคงที่คือ 1/12 ของค่าความถี่ออสซิลเลเตอร์

โดยจากรูปที่ 2.4 เป็นการแสดงแผนภาพเวลาต่างๆในโหมด 0 เมื่อมีการรับและส่งข้อมูล 1 ไบต์ โดยสัญญาณนาฬิกาในการเลื่อนบิตจะสร้างขึ้นภายในตัว MCS-51 เพื่อนำไปใช้สำหรับวงจรชิพรีจิสเตอร์ภายนอก



รูปที่ 2.4 แสดงแผนภาพเวลาต่างๆในโหมด 0 เมื่อมีการรับและส่งข้อมูล 1 ไบต์

สัญญาณนาฬิกาที่สร้างขึ้นทางขาสัญญาณ Tx นี้จะสลับค่าไปมาระหว่างลอจิกสูงไปต่ำ ในช่วงเวลาที่ใกล้เคียงกับขอบขาลงของสัญญาณ ALE ซึ่งอยู่ในคาบเวลาของออสซิลเลเตอร์ที่ 15 ภายหลังจากที่ได้ทำคำสั่งการโอนย้ายข้อมูลมายังรีจิสเตอร์ SBUF หรือทำคำสั่งที่ทำให้ค่าแฟล็กสถานะ RI ทีค่าเป็น 0 หลังจากนั้นสัญญาณนาฬิกาจะเปลี่ยนแปลงอีกครั้งในช่วงเวลาที่ใกล้เคียงกับขอบขาลงของสัญญาณ ALE ซึ่งอยู่ในคาบเวลาของออสซิลเลเตอร์หลังจากนั้นอีก 6 คาบ และจะดำเนินต่อไปเช่นนี้จนข้อมูลทั้ง 8 ไบต์ได้ถูกส่งออกไปเรียบร้อยแล้ว เมื่อสัญญาณขอบขาขึ้นของสัญญาณนาฬิกาเกิดขึ้นจนครบ 8 ครั้งแล้วจะมีผลให้แฟล็กสถานะ TI หรือ RI มีค่าเป็น 1 และสถานะของขาสัญญาณ TxD ก็จะเป็นระดับลอจิกสูงไปตลอด

ข้อมูลที่จะส่งออกไปภายนอกจะถูกเลื่อนบิตนัยสำคัญต่ำออกไปก่อนเป็นลำดับแรก โดยจะเริ่มขึ้นในเวลาเริ่มต้นของคาบเวลาของออสซิลเลเตอร์ ภายหลังจากที่ได้ทำคำสั่งการโอนย้ายข้อมูลมายังรีจิสเตอร์ SBUF สำหรับบิตแรกของข้อมูลที่รับมานั้นจะถูกแลตช์ไว้ด้วยขอบขาขึ้นของสัญญาณนาฬิกาในคาบเวลาที่ 24 ภาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่มีการกำหนดให้แฟล็กสถานะ RI เป็นค่า 0 หลังจากนั้นในคาบเวลาของ ออสซิเลเตอร์อีก 12 คาบถัดมาก็จะรับบิตต่อไปได้ ซึ่งจะดำเนินไปเรื่อยๆจนได้รับข้อมูลครบ 8 บิต

โหมด 1

การสื่อสารผ่านพอร์ตอนุกรมในโหมด 1 ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART (Universal Asynchronous Receiver/Transmission) โดยใช้กลุ่มข้อมูลแบบ 10 บิต ประกอบด้วยบิตเริ่มต้นจำนวน 1 บิต, ข้อมูลที่จะส่งไป จำนวน 8 บิต และบิตสุดท้าย จำนวน 1 บิต ดังรูปที่ 2.5 โดยข้อมูลจะถูกส่งออกทางขาสัญญาณ TxD และรับเข้ามาทางขาสัญญาณ RxD ในส่วนของข้อมูล 8 บิตที่ได้รับหรือทำการส่งออกจะเป็นบิตนัยสำคัญต่ำก่อนเป็นอันดับแรก และบิตท้ายสุดของข้อมูลที่รับมาจะเก็บไว้ในบิต RB8 ภายในรีจีสเตอร์ SCON โดยสามารถเลือกเปลี่ยนแปลงความเร็วในการส่งข้อมูลได้

การส่งข้อมูลจะเกิดขึ้นภายหลังจากการเขียนหรือโอนถ่ายข้อมูลเข้าไปยังรีจีสเตอร์ SBUFโดยสามารถโปรแกรมเพื่อทำการตรวจสอบค่าสถานะแฟล็ก TI ภายในรีจีสเตอร์ SCON ได้ ซึ่งจะมีค่าเป็น 1 ภายหลังที่ข้อมูลได้เลื่อนออกไปภายนอกจนเสร็จสิ้นแล้ว สำหรับการรับข้อมูล จะเริ่มต้นเมื่อได้มีการกำหนดค่าในบิต REN เป็น 1 และมีการเปลี่ยนแปลงระดับสัญญาณที่ขาสัญญาณ RxD เกิดขึ้น การสุ่มอ่านค่าข้อมูลเข้ามาจะใช้อัตราเดียวกับอัตราบอดที่ได้กำหนดไว้ในช่วงกลางของคาบเวลาของบิตหลังจากที่ได้รับข้อมูลครบจำนวน 10 บิตแล้ว และหากมีสภาวะดังในตารางที่ 2.5 ก็จะทำให้เกิดการโอนย้ายข้อมูลไปเก็บในรีจีสเตอร์ SBUF

1	ค่าแฟล็ก RI มีค่าเป็น 0 (แสดงว่ามีการอ่านไบต์ข้อมูลเข้ามาแล้ว และพร้อมที่จะรับข้อมูลถัดไป) และบิต SM2 มีค่าเป็น 0 เช่นเดียวกัน
2	ค่าของบิตสุดท้ายมีค่าเป็น 1 (แสดงว่าข้อมูลที่รับมาถูกต้อง จึงได้โอนย้ายไปเก็บในรีจีสเตอร์ SBUF โดยไม่สนใจค่าของ SM2)

ตารางที่ 2.5 แสดงค่าสภาวะที่จะทำให้เกิดการโอนย้ายข้อมูลไปเก็บในรีจีสเตอร์ SBUF

รูปที่ 2.5 แสดงรูปแบบของสัญญาณข้อมูลนุกรมในโหมด 1

ดังนั้นเมื่อข้อมูลที่เข้ามาทั้ง 10 บิต บิตเริ่มต้นจะไม่ถูกนำไปใช้งาน , บิตข้อมูล 8 บิตจะถูกย้ายไปเก็บไว้ที่รีจีสเตอร์ SBUF และบิตสุดท้ายจะถูกเก็บไว้ในตำแหน่งของบิต RB8 ภายในรีจีสเตอร์ SCON นอกจากนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยังมีแฟล็กสถานะ RI ซึ่งจะเป็นค่า 1 เพื่อบอกว่าได้มีการรับข้อมูลใหม่เข้ามาแล้ว ในกรณีที่โปรแกรมอ่านข้อมูลจากรีจีสเตอร์ SBUF แล้วไม่ได้กำหนดค่าบิต RI ให้เป็น 0 อีกครั้ง ข้อมูลที่รับมาใหม่จะสูญหายไป

อัตราการส่งข้อมูลอนุกรมโหมด 1

อัตราการส่งข้อมูลอนุกรมโหมด 1 สามารถเปลี่ยนแปลงได้โดยใช้ Timer 1 หรือ Timer 2 ทำหน้าที่เป็นตัวกำเนิดอัตราการส่งข้อมูล และใช้แฟล็กที่เกิดขึ้นจากการโอเวอร์โฟลว์

กรณีใช้ Timer 1 ทำงานในโหมด 2 (8 – Bit Automatic Reload)

$$Baud\ rate = \frac{2^{SMOD}}{32} = \frac{oscillating\ frequency}{12 \times [256 - TH1]} \qquad (2.1)$$

โดย SMOD เป็นค่าของบิตภายในรีจีสเตอร์ PCON (มีค่าเป็น 0 หรือ 1)

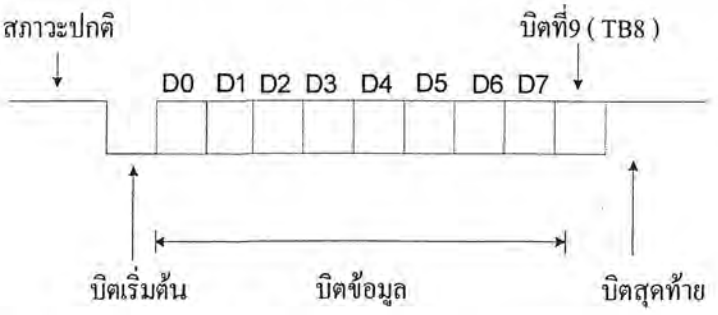
TH1 เป็นค่าภายในรีจีสเตอร์ TH1 ซึ่งใช้เป็นค่าสำหรับReload

กรณีใช้ Timer 1 ทำงานในโหมดอื่นๆที่ไม่ใช่โหมด 2

$$Baud\ rate = \frac{2^{SMOD}}{32} \times (Overflow\ rate\ Timer1) \qquad (2.2)$$

โหมด 2 และโหมด3

การทำงานโหมด 2 หรือ โหมด 3 ของพอร์ตอนุกรม จะทำหน้าที่รับส่งข้อมูลจำนวน 11 บิต เหมือนกัน ซึ่งประกอบด้วยบิตเริ่มต้นจำนวน 1 บิต , ข้อมูลที่จะส่งไป จำนวน 8 บิต , บิตแสดงภาวะความเป็นเลขคู่หรือเลขคี่ (ข้อมูลบิตที่ 9) จำนวน 1 บิต และบิตสุดท้าย จำนวน 1 บิต ดังแสดงในรูปที่ 2.6 โดยจะต่างกันที่อัตราบอดที่ใช้ในการรับส่งข้อมูลอนุกรมคือ ในโหมด 2 อัตราส่งข้อมูลจะถูกกำหนดไว้คงที่ ส่วนในโหมด 3 จะสามารถเปลี่ยนแปลงอัตราส่งข้อมูลได้



รูปที่ 2.6 แสดงรูปแบบของสัญญาณข้อมูลนุกรมในโหมด 2 และ โหมด3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราการส่งข้อมูลของโหมดที่ 2

$$Baud\ rate = \frac{2^{SMOD}}{32} \times (Oscillating\ frequency) \qquad (2.3)$$

อัตราการส่งข้อมูลของโหมดที่ 3

อัตราการส่งข้อมูลอนุกรมโหมด 3 สามารถเปลี่ยนแปลงได้โดยใช้ Timer 1 หรือ Timer 2 ทำหน้าที่เป็นตัวกำเนิดอัตราการส่งข้อมูล และใช้แฟล็กที่เกิดขึ้นจากการโอเวอร์โฟลว์เหมือนในโหมด 1

การส่งข้อมูลในโหมด 2 และโหมด 3 จะต้องนำค่าข้อมูลนั้นนำไปเก็บไว้ในรีจีสเตอร์ SBUF สำหรับบิตที่ 9 ที่เพิ่มขึ้นมาจะถูกนำมาจากบิต TB8 ภายในรีจีสเตอร์ SCON ซึ่งจะต้องได้รับการกำหนดจากผู้ใช้ เมื่อข้อมูลถูกเลื่อนบิตส่งออกไปภายนอกเรียบร้อยแล้ว แฟล็กสถานะ TI จึงจะมีค่าเป็น 1 เช่นเดียวกับโหมดที่ผ่านๆ มา และจะต้องเปลี่ยนกลับให้เป็นค่า 0 ตามเดิม สำหรับการรับข้อมูล ข้อมูลจะถูกนำมาเก็บไว้ที่รีจีสเตอร์ SBUF เช่นเดียวกัน และค่าของบิตที่ 9 จะนำไปเก็บไว้ยังบิตRB8 ภายในรีจีสเตอร์ SCON

อัตราบอด	ความถี่สัญญาณ	SMOD	ไทเมอร์1		
(บิตต่อวินาที)	นาฬิกา		C/ T	โหมด	ค่ารีโหลด
โหมด 0 : สูงสุด 1 MHz	12 MHz	X	X	X	X
โหมด 2 : สูงสุด 375 kbps	12 MHz	1	X	X	X
โหมด1 ,3 : สูงสุด 62.5 kbps	12 MHz	1	0	2	FFH
19.2 kbps (19,200)	11.0592 MHz	1	0	2	FDH
9.6 kbps (9,600)	11.0592 MHz	0	0	2	FDH
4.8 kbps (4,800)	11.0592 MHz	0	0	2	FAH
2.4 kbps (2,400)	11.0592 MHz	0	0	2	F4H
1.2 kbps (1,200)	11.0592 MHz	0	0	2	E8H
137.5 bps	11.0592 MHz	0	0	2	1DH
110 bps	6 MHz	0	0	2	72H
110 bps	12 MHz	0	0	1	FEEH

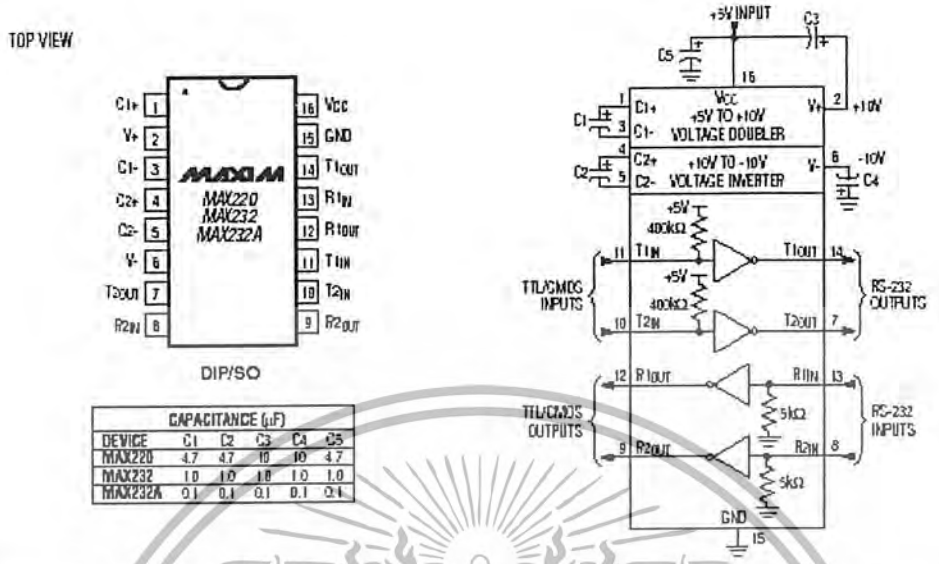
ตารางที่ 2.6 แสดงการเลือกอัตราบอดโดยการกำหนดค่าต่างๆของไทเมอร์ 1

2.5.8 การเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์

การใช้งานพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 มักนิยมใช้กับการเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมมาตรฐาน RS232 เป็นส่วนใหญ่ เนื่องจากระดับสัญญาณของพอร์ตอนุกรม RS232มีระดับ +3 ถึง +12 โวลต์ ในขณะที่ระดับสัญญาณของไมโครคอนโทรลเลอร์อยู่ที่ระดับที่ทีแอล จึงไม่สามารถต่อกันได้โดยตรง จึงต้องอาศัยการเชื่อมต่อผ่านทางไอซีที่ทำหน้าที่ในการแปลงสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้สามารถติดต่อกันได้ โดยไอซีดังกล่าวมีหลายเบอร์ด้วยกัน เช่น MAX232 จาก MAXIM หรือ ICL232 จาก HARRIS เป็นต้น



รูปที่ 2.7 แสดงรายละเอียดเบื้องต้นของไอซีแปลงสัญญาณเพื่อเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์

2.6 มาตรฐานพอร์ตอนุกรมแบบ RS232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้สื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดที่อยู่ห่างไกลกัน โดยคณะกรรมการที่เรียกว่าสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association : EIA) ได้วางมาตรฐานที่มีชื่อเรียกว่า EIA RS232 มาตรฐานนี้ในช่วงแรกจะใช้คอนเนคเตอร์เป็นแบบ DB - 25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3 V จนถึง -12 V แสดงว่ามีข้อมูล (Mark) และ +3 V จนถึง +12 V แสดงว่าเป็นช่องว่าง (Space)

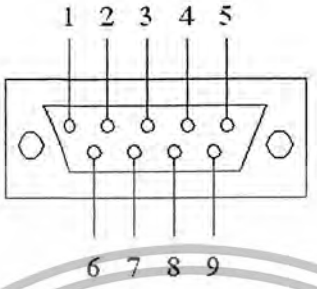
มาตรฐาน RS232 ได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment : DTE) กับข้อมูลปลายทาง (Data Circuit Terminating : DCE)ไว้ว่าอุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอมพิวเตอร์หรือไมโครคอนโทรลเลอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS232

ข้อแตกต่างของอุปกรณ์ DTE กับ DCE อย่างหนึ่งที่เห็นได้ชัดคือ คอนเนคเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเนคเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ทอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบDTEส่วนคอนเนคเตอร์ที่อยู่ที่โมเด็มจะเป็นแบบ DCE สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS232 มักถูกใช้เชื่อมต่อกับโมเด็มหรือเมาส์ โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุดถึง 20 เมตร

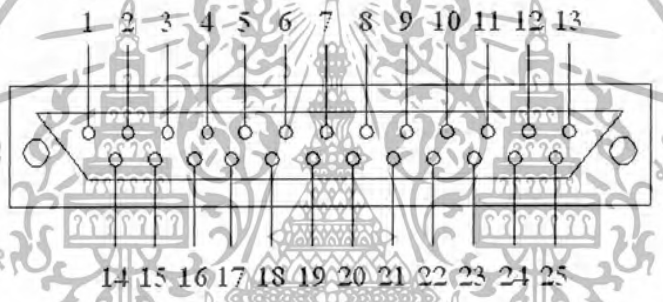
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนเนคเตอร์สำหรับพอร์ท RS232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS232 จะใช้คอนเนคเตอร์แบบ DB-25 ตัวผู้ หรือ DB-9 ตัวผู้ซึ่งคอนเนคเตอร์แบบ DB-25 จะมีขาอื่นๆ ที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนักจึงถูกยกเลิกไป โดยแสดงได้ดังรูปที่ 2.8



(ก) คอนเนคเตอร์อนุกรม 9 ขา หรือ แบบ DB-9 (มองจากด้านหลังคอมพิวเตอร์)



(ข) คอนเนคเตอร์อนุกรม 25 ขา หรือ แบบ DB-25 (มองจากด้านหลังคอมพิวเตอร์)

รูปที่ 2.8 แสดงการจัดขาของคอนเนคเตอร์พอร์ทอนุกรมตามมาตรฐาน RS232

คอนเนคเตอร์ DB-9	คอนเนคเตอร์ DB-25	ชื่อสายสัญญาณ	ชนิดสายสัญญาณ
1	8	Data Carrier Detect	อินพุท
2	3	Received Data	อินพุท
3	2	Transmitted Data	เอาท์พุท
4	20	Data Terminal Ready	เอาท์พุท
5	7	Signal Ground	-
6	6	Data Set Ready	อินพุท
7	4	Request To Send	เอาท์พุท
8	5	Clear To Send	อินพุท
9	22	Ring Indicator	อินพุท

ตารางที่ 2.7 แสดงการจัดขาของคอนเนคเตอร์พอร์ตอนุกรมตามมาตรฐาน RS232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 สเตปปิ้งมอเตอร์

สเตปปิ้งมอเตอร์แบ่งตามพื้นฐานได้ 3 ชนิดคือ วาริเอเบิลรีลักแตนซ์ (Variable Reluctance: VR), เพอร์มาเนนต์แมกเน็ต (Permanent Magnet: PM) และแบบไฮบริด (Hybrid)

- สเตปปิ้งมอเตอร์ชนิดวาริเอเบิลรีลักแตนซ์ มีโครงสร้างของโรเตอร์แบบมัลติทูธ (Multi-Tooth) ทำจากเหล็กอ่อน เราจะทราบได้ว่าเป็นมอเตอร์ชนิดวาริเอเบิลรีลักแตนซ์ได้ โดยการทดสอบง่ายๆคือการใช้นิ้วหมุนเพลาของมอเตอร์ และสังเกตได้ว่ามอเตอร์ชนิดนี้จะไม่เกิดปรากฏการณ์ทางแม่เหล็ก (Magnetism) มอเตอร์จึงสามารถหมุนได้ตลอดเวลาโดยไม่ติดขัด ซึ่งจะแตกต่างจากมอเตอร์ชนิดเพอร์มาเนนต์แมกเน็ต และมอเตอร์แบบไฮบริด ที่มีสนามแม่เหล็กที่โรเตอร์ โดยเมื่อทำการหมุนเพลาของมอเตอร์จะรู้สึกขัดๆเหมือนเป็นฟันเฟือง โดยมอเตอร์ชนิดวาริเอเบิลรีลักแตนซ์จะมีข้อด้อยในเรื่องความถูกต้องของตำแหน่ง และจะทำงานได้ไม่ดีนักเมื่อมีสเตปการหมุนสูง

- สเตปปิ้งมอเตอร์ชนิดเพอร์มาเนนต์แมกเน็ต มีโครงสร้างของโรเตอร์แบบเรียบ ไม่มีซี่ขั้วแม่เหล็กและบนโรเตอร์จะเป็นแบบแม่เหล็กถาวร การควบคุมสเตปปิ้งมอเตอร์ชนิดเพอร์มาเนนต์ ทำได้โดยการป้อนกระแสกระตุ้นที่ขดลวดบนสเตเตอร์ เช่น ถ้าเป็นสเตเตอร์แบบ 4 เฟส จะมีขั้วแม่เหล็กอยู่ 4 ขั้ว ซึ่งมีคอยล์พันอยู่แยกจากกัน ขั้วแม่เหล็กถาวรบนโรเตอร์จะถูกแรงดึงดูดจากขั้วแม่เหล็กบนสเตเตอร์เมื่อป้อนกระแสเข้าสู่ขดลวด และโรเตอร์จะคงอยู่กับที่ที่ขั้วแม่เหล็กบนสเตเตอร์นั้นถึงแม้ว่าจะไม่มีการป้อนกระแสไฟฟ้าอีกต่อไป ทำให้เกิดแรงยึดหน่วงขึ้น สเตปปิ้งมอเตอร์ชนิดเพอร์มาเนนต์นี้มีข้อดีในเรื่องความถูกต้องของตำแหน่ง และความเร็วมากขึ้นเมื่อเปรียบเทียบกับสเตปปิ้งมอเตอร์ชนิดอื่นๆ

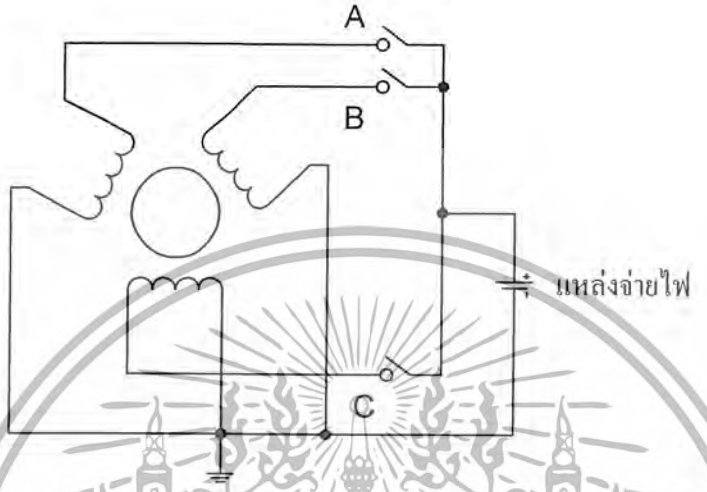
- สเตปปิ้งมอเตอร์ชนิดไฮบริด เป็นชนิดที่นิยมใช้งานกันมากที่สุด โดยเฉพาะการนำมาใช้งานร่วมกับอุปกรณ์ที่ใช้ในคอมพิวเตอร์ สเตปปิ้งมอเตอร์ชนิดไฮบริดมีโครงสร้างภายในที่รวมเอาโครงสร้างของสเตปปิ้งมอเตอร์ชนิดวาริเอเบิลรีลักแตนซ์ และสเตปปิ้งมอเตอร์ชนิดเพอร์มาเนนต์แมกเน็ตมาประกอบเข้าด้วยกันทำให้เป็นมอเตอร์ชนิดที่มีแรงยึดเหนี่ยวสูง , มีแรงบิดดึงและพลักได้ดี ซึ่งมีความคงที่ และทำงานได้ดีถึงแม้ว่าจะมีสเตปต่อรอบในการหมุนสูง

2.7.1 หลักการทำงานเบื้องต้นของสเตปปิ้งมอเตอร์

ส่วนประกอบแบบง่ายๆของสเตปปิ้งมอเตอร์สามารถแลดงได้ดังรูปที่ 2.9 ประกอบด้วยขั้วโพลสเตอร์ 3 ขั้ว และโรเตอร์ 2 ขั้ว ที่ทำจากเหล็กอ่อน ขดลวดสเตเตอร์จะถูกต่อเข้ากับแหล่งจ่ายไฟฟ้าโดยผ่านสวิตช์ 3 ตัว คือ A, B, C เมื่อสวิตช์เปิดวงจร มอเตอร์จะอยู่ตำแหน่งใดก็ได้ แต่เมื่อสวิตช์ A ปิดจะทำให้เกิดสนามแม่เหล็กที่สร้างโดยโพล 1 จะดูดโรเตอร์ให้อยู่ในแนวเส้นตรงเดียวกับขั้วโพลเหมือนที่แสดงในรูป แต่เมื่อเราเปิดสวิตช์ A และปิดสวิตช์ B โรเตอร์ ก็จะอยู่ในแนวเดียวกับขั้วโพล 2 ดังนั้นมอเตอร์จะหมุนในทิศทางทวนเข็มนาฬิกา (CC : Counter colckwise) 60 องศา เมื่อเปิดสวิตช์ B และปิดสวิตช์ C จะทำให้โรเตอร์อยู่ในแนวเดียวกับขั้วโพล 3 และส่งผลให้มอเตอร์หมุนในทิศทวนเข็มนาฬิกา ดังนั้นเราสามารถให้โรเตอร์ของสเตปมอเตอร์หมุนในทิศทวนเข็มนาฬิกาได้ 60 องศาต่อ 1 สเตปได้โดยการเปิดและปิดสวิตช์ไปตามลำดับจาก A, B, C, A, B, C, ... และในทางกลับกันเราสามารถให้โรเตอร์ของสเตปมอเตอร์หมุนในทิศตามเข็มนาฬิกาได้ 60 องศาต่อ 1 สเตปได้โดยการเปิดและปิดสวิตช์ไปตามลำดับจาก A, C, B, A, C, B, ...

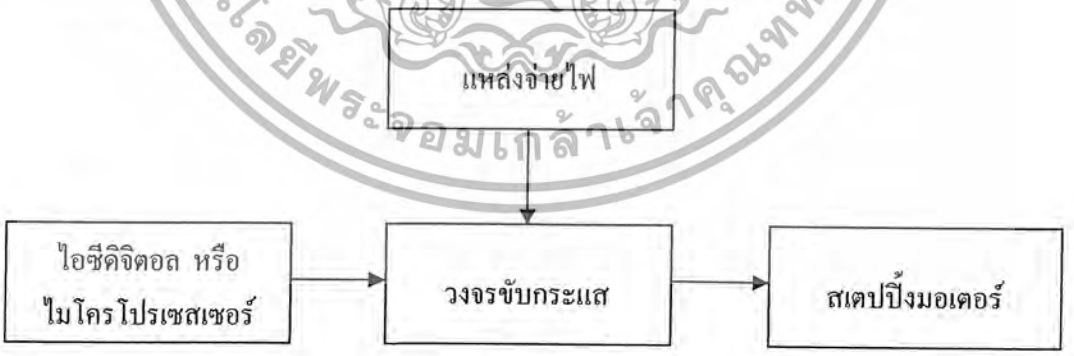
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยถ้าเราคงการปิดของสวิตช์สุดท้ายไว้ จะได้ตำแหน่งของโรเตอร์ยังคงอยู่ในตำแหน่งสุดท้าย เพื่อเป็นการป้องกันการกระทบกระเทือน จากแรงภายนอก ในสถานะคงตัวนี้มอเตอร์จะยังถูกล็อกอยู่ได้ ถ้ามีแรงภายนอกมากระทำด้วยแรงไม่มากเกินค่า โฮลดิ้งทอร์ก (Holing torque) ของมอเตอร์ ในการเคลื่อนที่จากตำแหน่งหนึ่งไปยังอีกตำแหน่งต่อไป การเคลื่อนของโรเตอร์จะถูกกระทบกระเทือนโดยอินเนอร์เชียร์ (Inertai) และแรงเสียดทานภายนอกได้



รูปที่ 2.9 แสดงส่วนประกอบแบบง่ายๆของสเตปปิ้งมอเตอร์ที่มีมุมโรเตอร์เคลื่อนที่ 60 องศา

ปัจจุบันได้มีการใช้งานสเตปปิ้งมอเตอร์ร่วมกับอุปกรณ์ทางอิเล็กทรอนิกส์มากขึ้น ทำให้การใช้งานสเตปปิ้งมอเตอร์สามารถทำๆ ได้ง่ายและสะดวกมากขึ้น โดยจะมีการนำอุปกรณ์ทางอิเล็กทรอนิกส์มาใช้เป็นอิเล็กทรอนิกส์สวิตช์ สำหรับควบคุมการจ่ายกระแสให้กับสเตปปิ้งมอเตอร์ และอุปกรณ์ทางดิจิตอล ไอซี หรือไมโครโปรเซสเซอร์ในการผลิตสัญญาณพัลส์เพื่อความสะดวกต่อการควบคุม โดยมีหลักการควบคุมสเตปปิ้งมอเตอร์ดังที่แสดงในรูปที่ 2.10



รูปที่ 2.10 แสดงหลักการควบคุมสเตปปิ้งมอเตอร์

สาเหตุที่มีการใช้งานสเตปปิ้งมอเตอร์มีการใช้งานอย่างแพร่หลาย เนื่องจากมีข้อดีดังต่อไปนี้

1. ไม่ต้องการส่วนป้อนกลับของสัญญาณเพื่อควบคุมตำแหน่งหรือความเร็วของมอเตอร์จึงสามารถควบคุม แบบวงเปิดได้ ทำให้วงจรการควบคุมทำได้ง่าย และต้นทุนต่ำ
2. ไม่มีการสะสม หรือเพิ่มขึ้นของความคลาดเคลื่อนของตำแหน่งการเคลื่อนที่
3. สามารถทำงานร่วมกับอุปกรณ์ดิจิตอล และคอมพิวเตอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 รูปแบบการขับสเตปปิ้งมอเตอร์

รูปแบบการขับ หรือการกระตุ้นสเตปปิ้งมอเตอร์ (Mode of excitation) คือ การจัดลำดับสัญญาณการกระตุ้นวงจรขับขดลวดเพื่อควบคุมทิศทางการหมุน และปรับปรุงประสิทธิภาพของแรงบิดมีอยู่ 3 แบบดังนี้

1. การขับแบบกระตุ้น 1 เฟส (One-Phase Excitation)

เป็นรูปแบบการขับสเตปปิ้งมอเตอร์แบบจ่ายกระแสให้กับขดลวดของสเตปปิ้งมอเตอร์ครั้งละ 1 เฟส เปลี่ยนไปตามลำดับ จะได้การเคลื่อนที่แบบเต็มสเตป จุดสมดุลของการหมุนจะอยู่ที่เฟสใดเฟสหนึ่ง วิธีนี้จะมีแรงบิดน้อย การจ่ายกระแสให้แต่ละเฟสจะแสดงได้ตามตารางที่ 2.8 และสามารถสั่งให้มอเตอร์หมุนได้ทั้งทิศทวนเข็มนาฬิกา และทิศตามเข็มนาฬิกา

Step	เฟสที่ถูกจ่ายกระแส			
	ϕ_1	ϕ_2	ϕ_3	ϕ_4
1	0	0	0	1
2	0	0	1	0
3	0	1	0	0
4	1	0	0	0
ทิศตามเข็มนาฬิกา	←			
ทิศทวนเข็มนาฬิกา	→			

หมายเหตุ : เลข 1 คือ มีการจ่ายกระแสให้กับขดลวดเฟสมอเตอร์
เลข 0 คือ ไม่มีการจ่ายกระแสให้กับขดลวดเฟสมอเตอร์

ตารางที่ 2.8 แสดงการจ่ายกระแสแบบการกระตุ้น 1 เฟส

2. การขับแบบกระตุ้น 2 เฟส (Two-Phase Excitation)

เป็นรูปแบบการขับสเตปปิ้งมอเตอร์แบบจ่ายกระแสให้กับขดลวดของสเตปปิ้งมอเตอร์แบบเต็ม สเตป จุดสมดุลของการหมุนจะไม่อยู่ที่เฟสใดเฟสหนึ่ง แต่จะอยู่ที่จุดกึ่งกลางระหว่างเฟสทั้งสอง แรงบิดของมอเตอร์จะมากกว่าแบบการกระตุ้นที่ละเฟส มีลักษณะการจ่ายกระแสให้แต่ละเฟสตามตารางที่ 2.9

Step	เฟสที่ถูกจ่ายกระแส			
	ϕ_1	ϕ_2	ϕ_3	ϕ_4
1	0	0	1	1
2	0	1	1	0
3	1	1	0	0
4	1	0	0	1
ทิศตามเข็มนาฬิกา	←			
ทิศทวนเข็มนาฬิกา	→			

ตารางที่ 2.9 แสดงการจ่ายกระแสแบบการกระตุ้น 2 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การขับแบบกระตุ้น 1 เฟสสลับ 2 เฟส

เป็นการขับสเตปมอเตอร์แบบจ่ายกระแสให้กับเฟสของมอเตอร์แบบ 1 เฟส สลับกับการจ่ายกระแสแบบ 2 เฟส วิธีนี้จะเป็นการเคลื่อนที่แบบครึ่งสเตป โดยมีจุดสมดุลของแรงบิดอยู่ที่เฟสใดเฟสหนึ่ง และที่จุดกึ่งกลางระหว่างเฟสทั้งสอง วิธีนี้จะมีจำนวนสเตปเพิ่มขึ้นเป็น 2 เท่าของ 2 แบบแรก แต่แรงบิดเฉลี่ยจะน้อยกว่าแบบที่ 2 ซึ่งมีลักษณะการจ่ายกระแสให้แต่ละเฟสดังตารางที่ 2.10

Step	เฟสที่ถูกจ่ายกระแส			
	ϕ_1	ϕ_2	ϕ_3	ϕ_4
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	0	0
6	1	1	0	0
7	1	0	0	0
8	1	0	0	1
ทิศตามเข็มนาฬิกา	←			
ทิศทวนเข็มนาฬิกา				→

ตารางที่ 2.10 แสดงการจ่ายกระแสแบบการกระตุ้น 1 เฟสสลับ 2 เฟส

2.8 แอลซีดีโมดูล (LCD Module)

แอลซีดีโมดูลมีอยู่หลายรุ่น และคุณสมบัติแตกต่างกันไป ซึ่งแบ่งได้เป็น 2 กลุ่มหลักคือแบบดอทเมตริก (Dot Matrix) และกราฟฟิก (Graphic) โดยดอทเมตริกจะแสดงผลเป็นตัวอักษรขนาด 5×8 ดอท และมีจำนวนอักษรและบรรทัดแตกต่างออกไปในแต่ละรุ่น ส่วนแบบกราฟฟิกจะสามารถแสดงผลในแบบบิตแม็ป (Bit Map) คือจะสร้างเป็นภาพใดๆก็ได้ตามต้องการ แนวทางในการใช้งานของทั้งสองแบบจะมีคุณลักษณะที่ใกล้เคียงกัน การใช้งานโดยทั่วไปจะใช้แบบดอทเมตริกมากกว่าเนื่องจากราคาถูกกว่า และเพียงพอต่อการใช้งานส่วนใหญ่ คุณสมบัติของดอทเมตริก แอลซีดีโมดูลสรุปเป็นข้อๆได้ดังนี้

1. มีให้เลือกหลายรุ่นตามความต้องการใช้งานโดยมีจำนวนตัวอักษรและบรรทัดแตกต่างกันไป
2. ตัวอักษรแสดงด้วยดอทเมตริกขนาด 5×8 ดอท
3. สามารถต่อเข้ากับระบบไมโครคอนโทรลเลอร์ได้ 2 ลักษณะ คือ แบบเมมโมรีแม็ป และแบบผ่าน 8255 พอร์ต ซึ่งจะใช้ขาสัญญาณทั้งหมด 12 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การใช้งานง่ายและสะดวกระบบไมโครคอนโทรลเลอร์เพียงแค่ส่งข้อมูลให้กับแอลซีดีโมดูล เท่านั้นข้อความก็จะปรากฏบนแผงแสดงผลและจะค้างค่าไว้ตลอดทำให้ไม่ต้องเสียเวลาของระบบ
5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากมาย เช่น เคลียร์ (Clear) , ดิสเพลย์ (Display),โฮม (Home) , เคอร์เซอร์ออนอ็อฟ (Cursor On Off) และอื่นๆอีก
6. สามารถแสดงผลเป็นตัวอักษรภาษาอังกฤษ และตัวเลขได้ 160 ตัวอักษร และสัญลักษณ์พิเศษอีก 32 ตัว รวมทั้งสามารถกำหนดตังอักษรที่ออกแบบเองได้อีก 8 ตัว
7. กินกระแสน้อย มีน้ำหนักเบา ทำงานได้ด้วยไฟเลี้ยงระดับ 5 โวลต์ เท่านั้น

2.8.1 โครงสร้างภายในตัวควบคุมแอลซีดีโมดูล

โครงสร้างภายในตัวควบคุมของแอลซีดีโมดูลแบบอักขระประกอบด้วยส่วนประกอบหลักๆดังนี้

บัฟเฟอร์อินพุต เอาท์พุต เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

รีจีสเตอร์คำสั่ง (Instruction Register: IR) เป็นรีจีสเตอร์ใช้สำหรับรับข้อมูลคำสั่งจากอุปกรณ์ภายนอก เพื่อนำไปควบคุมการแสดงผล

รีจีสเตอร์ข้อมูล (Data Register: DR) เป็นรีจีสเตอร์ ใช้สำหรับรับข้อมูลจากอุปกรณ์ภายนอกเพื่อถ่ายทอดต่อไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผลหรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร

แรมเก็บข้อมูลแสดงผล (Display Data RAM: DDRAM) เป็นหน่วยความจำรอมที่ใช้เก็บข้อมูลที่มาจากรีจีสเตอร์ (DR) ตัวควบคุมจะนำข้อมูลในแรมเก็บข้อมูลแสดงผลนี้ไปเปิดตาราง (Look up Table) ของตัวอักษรที่เก็บไว้ในหน่วยความจำรอม และแรมเก็บตัวอักษร เพื่อนำไปแสดงที่ตัวแสดงผล

รอมเก็บตัวอักษร (Character Generator ROM: CGROM) เป็นหน่วยความจำรอมที่ใช้เก็บข้อมูลอักษรหรือสัญลักษณ์ที่สามารถอ่านออกไปแสดงตัวที่ตัวแสดงผลได้ มีขนาด 7200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลในแรมเก็บข้อมูลแสดงผล (DDRAM)

แรมเก็บตัวอักษร (Character Generator RAM: CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บข้อมูลอักษรหรือสัญลักษณ์ที่มีการสร้างเพิ่มเติมขึ้นมาใหม่ ในกรณีที่ตัวอักษรในรอมเก็บตัวอักษร (CGROM) ไม่เพียงพอ มีขนาด 512 บิต การเขียนและการอ่านค่าไปใช้นั้นทำได้เดียวกับรอมเก็บตัวอักษร (CGROM) คือเขียนข้อมูลลงในแรมเก็บข้อมูลแสดงผล (DDRAM) แล้วตัวควบคุมจะมาอ่านค่าจากแรมเก็บตัวอักษร (CGRAM) เอง

แฟล็กบิซี่ (Busy Flag) เป็นส่วนที่ทำหน้าที่แจ้งสถานะของการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่า ตัวควบคุมพร้อมที่จะรับข้อมูล หรือ คำสั่งต่อไปหรือไม่ ดังนั้นก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุมจะต้องตรวจสอบสถานะของแฟลกบิซี่นี้เสียก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2 ขาสัญญาณของแอลซีดีโมดูล

ขา	สัญลักษณ์	ระดับ	หน้าที่
1	VSS	-	0 โวลต์ กราวด์
2	VCC	-	+ 5 โวลต์ พาวเวอร์ซัพพลาย
3	VEE	-	+ 5 โวลต์ ลิขวิด คริสตอล ไดรฟ์ (Liquid Crystal Drive)
4	RS	H / L	เลือกรีจีสเตอร์ (Register Select) H : Data Input L : Instruction Input
5	R / W	H / L	H : Data Read L : Data Write
6	E	H	เอนนาเบิ้ล ซิกแนล (L → H)
7	DB0	H / L	Data Bus Bit 0
8	DB1	H / L	Data Bus Bit 1
9	DB2	H / L	Data Bus Bit 2
10	DB3	H / L	Data Bus Bit 3
11	DB4	H / L	Data Bus Bit 4
12	DB5	H / L	Data Bus Bit 5
13	DB6	H / L	Data Bus Bit 6
14	DB7	H / L	Data Bus Bit 7

ตารางที่ 2.11 แสดงขาสัญญาณของ แอลซีดีโมดูล

2.8.3 การต่อเข้ากับระบบไมโครคอนโทรลเลอร์

จอแอลซีดี จะต่อเข้ากับระบบไมโครคอนโทรลเลอร์ได้ 2 ลักษณะตามที่กล่าวมาแล้ว ซึ่งทั้งสองแบบมีข้อดี และข้อเสียแตกต่างกันไป โดยแต่ละแบบมีหลักการดังนี้

1. การต่อแบบเมมโมรี่แม็ป

1.1 สามารถต่อเข้ากับชิปเบอร์ทั่วๆไป เช่น 8051 หรือ Z80 โดยจะทำให้ระบบไมโครคอนโทรลเลอร์มองเห็นแอลซีดีโมดูลในลักษณะของเมมโมรี่ได้ทันที

1.2 ผู้ใช้สามารถเขียนและอ่านข้อมูลจากแอลซีดีได้ทำให้มองเสมือนว่าเป็นเมมโมรี่บัฟเฟอร์ไปในตัว

1.3 เนื่องจากสามารถอ่านข้อมูลกลับได้ จึงสามารถตรวจสอบแฟล็กความพร้อม ในขณะที่แอลซีดีโมดูลกำลังทำงาน

1.4 ใช้ได้กับบอร์ดที่มีแอลซีดีโมดูลบัสมาให้พร้อม ทำให้กินพื้นที่ของหน่วยความจำไป ส่วนหนึ่งและต้องมีการดีโค้ดที่ละเอียดพอสมควร

1.5 การจัดขาสัญญาณจะต้องเป็นไปตามแบบของชิป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การต่อแบบ อินพุต / เอาท์พุตพอร์ต

2.1 สามารถต่อเข้ากับอินพุต / เอาท์พุตพอร์ตใดๆก็ได้ โดยใช้สายสัญญาณจำนวน 11 เส้น และใช้โปรแกรมเป็นตัวสร้างสัญญาณขึ้นมา ให้ตรงกับข้อกำหนดของแอลซีดีโมดูล

2.2 ผู้ใช้จะเขียนข้อมูลให้แอลซีดีโมดูลได้อย่างเดียว ซึ่งผู้ใช้ควรกำหนดเมมโมรี่ส่วนหนึ่งให้เสมือนบัฟเฟอร์ให้กับแอลซีดีโมดูล

2.3 เนื่องจากไม่สามารถอ่านข้อมูลกลับได้ จึงต้องใช้การหน่วงเวลาของระบบเอง เพื่อรอให้แอลซีดีโมดูลกระทำกระบวนการต่างๆ

2.4 ใช้กับบอร์ดทั่วๆไปที่มีพอร์ต ทำให้ไม่เปลืองส่วนของเมมโมรี่ในการใช้งาน

2.5 การจัดขาสัญญาณกระทำได้อย่างอิสระ

2.9 วงจรตัวส่ง และตัวรับสัญญาณอินฟราเรด

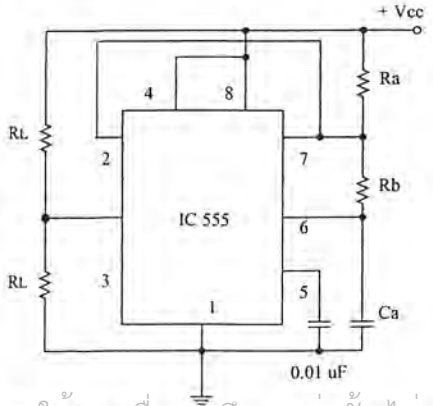
วงจรอินฟราเรดนี้ จะใช้เป็นอุปกรณ์ที่ใช้ในการตรวจสอบสถานะของประตู โดยจะติดตั้งเครื่องส่งสัญญาณอินฟราเรด และเครื่องรับสัญญาณไว้คนละด้านของประตูเพื่อตรวจสอบสัญญาณอินฟราเรดที่ส่งมาจากด้านส่ง ซึ่งในสภาวะปกติจะรับสัญญาณได้ โดยถ้าเมื่อด้านรับไม่สามารถรับสัญญาณได้แสดงว่าประตูได้ถูกเปิดออก เราจึงแจ้งสถานะของประตูเพื่อไปควบคุมส่วนของไมโครคอนโทรลเลอร์ได้

ในส่วนของการออกแบบวงจรโดยใช้ IC555 สร้างสัญญาณพัลส์โดยการเก็บประจุเข้าไปในตัวเก็บประจุ Ca ผ่าน Ra และ Rb เมื่อแรงดันคร่อม Ca มีค่าเป็น 2/3 ของ Vcc จะทำให้วงจรเปรียบเทียบแรงดันชุดบนทำงาน ซึ่งจะไปกระตุ้นวงจรควบคุมฟลิปฟลอปทำงานอีกต่อ ส่งผลทำให้ Ca คายประจุผ่าน Rb และขา 7 ลงกราวด์ ช่วงนี้เอาท์พุตที่ขา 3 จะมีแรงดันต่ำ และเมื่อแรงดันที่ Ca ลดลงมาถึง 1/3 ของ Vcc ก็จะไปกระตุ้นให้วงจรเปรียบเทียบชุดล่างทำงานกระตุ้นวงจรฟลิปฟลอป ทำให้แรงดันเอาท์พุตสูงขึ้น และ Ca ไม่สามารถคายประจุได้อีกเพราะทรานซิสเตอร์ภายในที่ทำหน้าที่เป็นสวิตช์อยู่ในสภาวะไม่ทำงาน ดังนั้น Ca จะทำการเก็บประจุอีกครั้ง ซึ่งมีการต่อวงจรตามรูปที่ 2.11 จากลักษณะการทำงานของวงจร จะเห็นได้ว่าช่วงความกว้างพัลส์ที่ได้ทางเอาท์พุตจะขึ้นอยู่กับการเก็บประจุ และการคายประจุของ Ca ซึ่งจาก Datasheet จะได้ว่า

ช่วงการเก็บประจุ : $t1 = 0.693*(Ra+Rb)*Ca$

ช่วงการคายประจุ : $t2 = 0.693*Ra*Ca$

ความถี่ของสัญญาณเอาท์พุต : $f = 1/(t1+t2)$



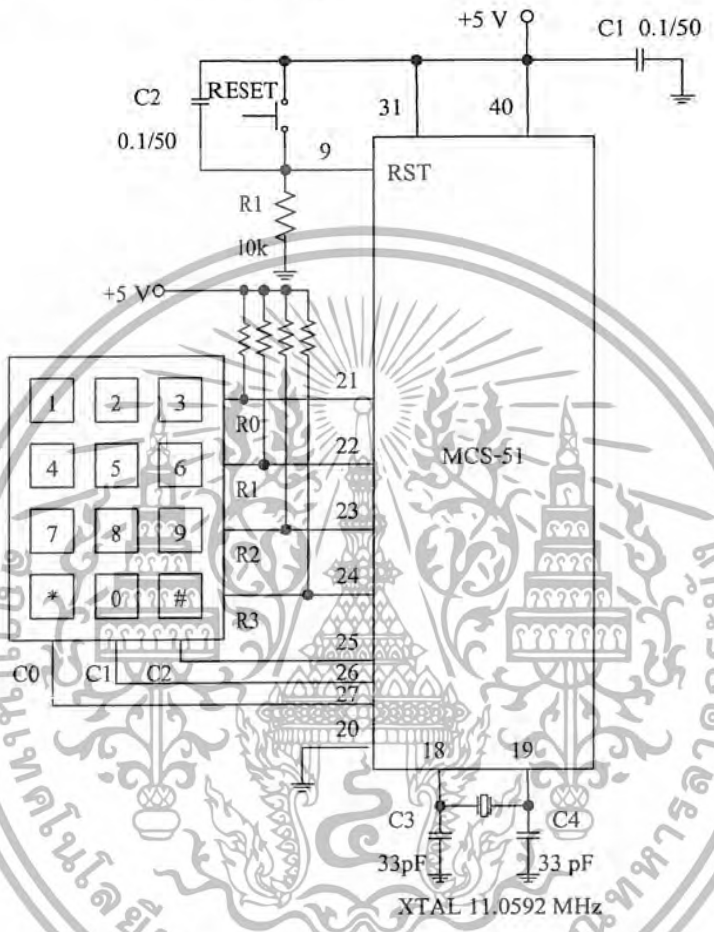
รูปที่ 2.11 แสดงวงจรสร้างพัลส์เพื่อใช้ในการขับหลอดแอลอีดีในวงจรอินฟราเรดทางด้านส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 คีย์แพด (Keypad)

เป็นวงจรของสวิตช์ลักษณะหนึ่ง โดยที่สวิตช์จะถูกต่อกันในแนวแกนตั้งและแกนนอน จะเรียกแนวตั้งว่าหลักหรือคอลัมน์ (Column) ในขณะที่แนวนอนจะเรียกว่าแถวหรือโรว์ (Row) ดังนั้นค่าของสวิตช์จึงประกอบด้วย ตำแหน่งในแนวหลักและแถว

การเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51



รูปที่ 2.12 วงจรเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51

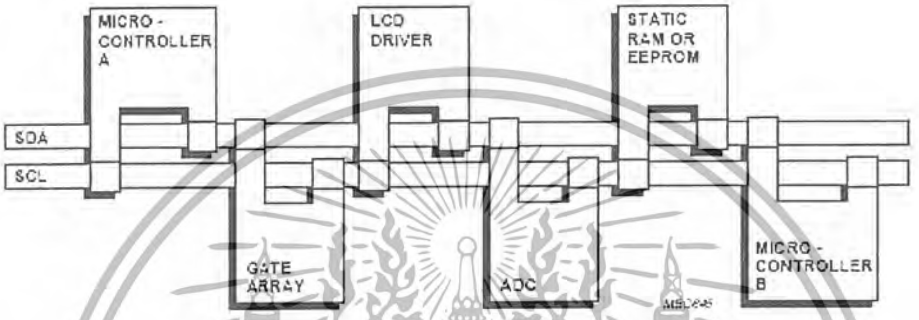
2.11 ระบบบัส I^2C

คำว่า I^2C ย่อมาจากคำว่า Inter-IC communication หมายถึง การติดต่อสื่อสารระหว่างไอซี โดยระบบบัส I^2C ได้รับการพัฒนาขึ้นโดยฟิลิปส์ (Philips) ด้วยจุดมุ่งหมายหลักคือ ต้องการให้ไอซี หรือโมดูลสามารถติดต่อ สั่งงาน และควบคุมภายใต้สายสัญญาณเพียง 2 เส้น เส้นที่หนึ่งคือ สายข้อมูล และอีกเส้นคือ สายสัญญาณนาฬิกาที่ใช้ในการควบคุมจังหวะการทำงาน การต่ออุปกรณ์ร่วมกันของอุปกรณ์บนระบบบัส I^2C ทำได้โดยการเชื่อมต่อสายข้อมูล และสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัวขนาน หรือพ่วงกันไป ส่วนการกำหนดแอดเดส หรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัว จะใช้รหัสข้อมูล และการกำหนดสภาวะลอจิกที่ขาแอดเดสของอุปกรณ์แต่ละตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สายข้อมูลบนระบบบัส I^2C มีชื่อเรียกอย่างเป็นทางการว่า สายข้อมูลอนุกรม หรือ SDA (Serial Data Line) ส่วนสายสัญญาณนาฬิกามีชื่อเรียกว่า สายสัญญาณนาฬิกาอนุกรม หรือ SCL (Serial Clock Line) โดยในการอธิบายต่อไปหลังจากนี้จะใช้คำว่า สาย SDA และ สาย SCL แทนสายสัญญาณทั้งสอง

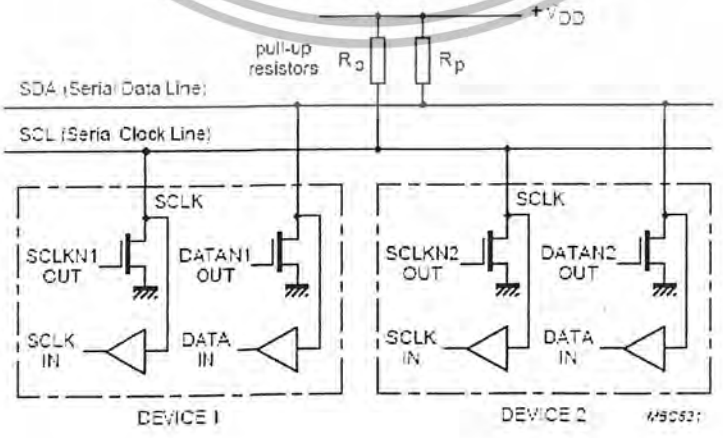
ในรูปที่ 2.13 แสดงแผนผังการเชื่อมต่ออุปกรณ์ต่างๆบนระบบบัส I^2C ซึ่งสามารถทำการต่ออุปกรณ์ได้หลากหลาย เช่น ไอซีขยายพอร์ตอินพุต เอาท์พุต (I/O Expander), ไอซีแปลงสัญญาณอนาลอกเป็นดิจิตอล (ADC) และแปลงสัญญาณดิจิตอลเป็นอนาลอก (DAC), ไอซีรีลไทม์คล็อก (RTC), ไอซีขับโมดูล LCD, หน่วยความจำอีอีพรอม และไมโครคอนโทรลเลอร์



รูปที่ 2.13 แสดงแผนผังการเชื่อมต่ออุปกรณ์ต่างๆบนระบบบัส I^2C

2.11.1 คุณสมบัติโดยทั่วไปของระบบบัส I^2C

สาย SDA และ SCL เป็นสายสัญญาณสองทิศทาง (Bi-Directional Line) ต้องการมีตัวต้านทานพูลอัปกับแรงดัน +5 โวลต์ ไว้ตลอดเวลา เพื่อให้สถานะของสายมีสถานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งาน ทั้งยังช่วยในการป้องกันสัญญาณรบกวนที่อาจมีเข้ามายังสายทั้งสอง วงจรเอาท์พุตของอุปกรณ์ที่ต่ออยู่กับระบบบัส I^2C ต้องมีลักษณะเป็นวงจรเดรนเปิด (Open-Drain) หรือคอลเล็กเตอร์เปิด (Open-Collector) ดังแสดงได้ดังรูปที่ 2.14

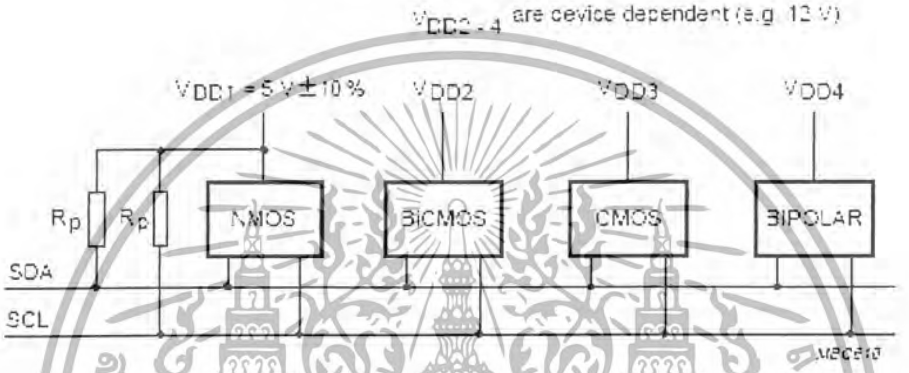


รูปที่ 2.14 วงจรเอาท์พุตของอุปกรณ์ที่ต่ออยู่กับระบบบัส I^2C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราการถ่ายทอดข้อมูลบนระบบบัส I^2C สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติ (Standard Mode) และสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง (Fast Mode) อุปกรณ์ที่ต่อร่วมอยู่บนระบบบัส I^2C จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400 pF การเข้าถึงอุปกรณ์บนระบบบัส I^2C ใช้ข้อมูลสำหรับการเข้าถึง 2 ค่าคือ 7 บิต (7-bit addressing) หรือ 10 บิต (10-bit addressing)

ข้อเด่นอีกประการของระบบบัส I^2C ก็คือสามารถเชื่อมต่ออุปกรณ์ที่ใช้ไฟเลี้ยงไม่เท่ากันให้สามารถติดต่อสื่อสารกันได้ โดยการต่อร่วมกันบนระบบบัส I^2C สามารถกระทำได้ในลักษณะเดียวกับกรณีที่อุปกรณ์ทั้งสองตัวใช้ไฟเลี้ยงเท่ากัน คือ ต่อสาย SDA และ SCL ของแต่ละอุปกรณ์เข้าด้วยกัน และต้องต่อตัวต้านทานพูลอัป (R_p) เข้ากับแรงดัน +5 โวลต์ไวเสมอดังรูปที่ 2.15



รูปที่ 2.15 การต่อตัวต้านทานพูลอัปบนสายสัญญาณในระบบ I^2C

2.11.2 หลักการของบัส I^2C

บัส I^2C ประกอบด้วยสายสัญญาณ 2 เส้น ดังที่ได้กล่าวมาแล้วคือ SDA และ SCL อุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้นจึงต้องมีการกำหนดรูปแบบของการติดต่อบนบัส หรือเรียกว่า โปรโตคอล (Protocol) เพื่อให้ผู้ใช้งานทราบว่า ขณะนี้อุปกรณ์ใดติดต่อกันอยู่ และอุปกรณ์ตัวใดเป็นตัวรับหรือตัวส่ง ต่อไปนี้จะขออธิบายลักษณะ หน้าที่ และนิยามของอุปกรณ์ที่ต่ออยู่บนบัส I^2C เพื่อเป็นข้อตกลงพื้นฐานก่อนที่จะอธิบายการทำงานของบัส I^2C ต่อไป

อุปกรณ์ที่เป็นผู้สร้างข้อมูลหรือส่งข้อมูลเรียกว่า ตัวส่ง (Transmitter)

อุปกรณ์ที่เป็นผู้รับข้อมูล เรียกว่า ตัวรับ (Receiver) ในอุปกรณ์บนบัส I^2C สามารถเป็นได้ทั้งตัวรับและตัวส่ง บางอุปกรณ์ทำหน้าที่เป็นตัวรับเพียงอย่างเดียว จะไม่มีอุปกรณ์ใดบนบัส I^2C ที่ทำหน้าที่เป็นตัวส่งเพียงอย่างเดียว อุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการติดต่อบนบัส I^2C เรียกว่า มาสเตอร์ (Master) อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ต่อพ่วงเข้าไปบนบัส I^2C เรียกว่า สเลฟ (slave)

ข้อกำหนด 2 ประการสำคัญของการติดต่อบนบัส I^2C คือ

(1) การถ่ายทอดข้อมูลจะเกิดขึ้นได้เมื่อบัสว่างเท่านั้น

(2) ในระหว่างการถ่ายทอดข้อมูล เมื่อใดก็ตามที่สาย SCL มีสถานะเป็นลอจิกสูง สายข้อมูลต้องรักษาข้อมูลไว้ อย่าให้เกิดการเปลี่ยนแปลงขึ้นเด็ดขาด มิฉะนั้น สัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาณควบคุมแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.3 สภาวะที่เกิดขึ้นบนบัส I^2C

มีด้วยกัน 5 สภาวะดังนี้

(1) บัสว่าง (Bus not busy) สภาวะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL เป็นลอจิกสูงทั้งคู่ นั้นหมายความว่า การถ่ายทอดข้อมูลสามารถเริ่มต้นได้

(2) เริ่มต้นการถ่ายทอดข้อมูล (start data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะเริ่มต้น (Start)

(3) หยุดการถ่ายทอดข้อมูล (stop data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะหยุด (Stop)

(4) ข้อมูลดำรงอยู่บนบัส (Data valid) สภาวะนี้เกิดขึ้นถัดจาก สภาวะเริ่มต้น โดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA ก็คือข้อมูลที่ทำการถ่ายทอด เมื่อสาย SCL เป็นลอจิกสูง สถานะที่สาย SDA ต้องคงที่เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่า เป็น "0" หรือ "1" ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่ สาย SCL เป็นลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอดข้อมูลอย่างสมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ตลอดช่วงเวลาที่สาย SCL เป็นลอจิกสูง หากเกิดการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL เป็นลอจิกสูงอยู่นั้น อุปกรณ์มาสเตอร์ที่ทำการควบคุมการถ่ายทอดข้อมูลจะแปลความหมายเป็นสภาวะหยุดหรือสภาวะเริ่มต้นก็ได้ ทำให้ข้อมูลที่ทำการถ่ายทอดนั้นเกิดความผิดพลาดขึ้น

(5) รับรู้ข้อมูล (acknowledge) เกิดขึ้นหลังจากที่การถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้นอย่างสมบูรณ์ โดย ตัวส่งจะทำการส่งข้อมูลมา 1 บิต เรียกว่า บิตรับรู้ (Acknowledge bit) มีสถานะเป็นลอจิกสูง หลังจากส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์จะทำการส่งสัญญาณรับรู้พิเศษซึ่งสัมพันธ์กับสัญญาณนาฬิกา เพื่อตอบสนองบิตรับรู้ที่ส่งมาจากตัวส่ง ทางด้านตัวรับจะส่งบิตรับรู้ ที่มีสถานะลอจิกต่ำลงบนบัส อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังติดต่ออยู่ในขณะนั้นก็จะกำเนิดบิตรับรู้เพื่อตอบสนองให้ทราบว่าได้รับข้อมูลในแต่ละไบต์เรียบร้อยแล้ว

ในรูปที่ 2.16 เป็นไดอะแกรมเวลาที่แสดงถึงการเกิดสภาวะต่างๆบนบัส I^2C ไม่ว่าจะเป็นสภาวะบัสว่าง, เริ่มต้น, ถ่ายทอดข้อมูล, รับรู้ และหยุดการถ่ายทอดข้อมูล

รูปที่ 2.16 ไดอะแกรมเวลาแสดงสถานะต่างๆในบัส I^2C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.4 การทำงานบนบัส I^2C

ก่อนที่จะเริ่มต้นการถ่ายทอดข้อมูลระหว่างอุปกรณ์ต่างๆที่ต่ออยู่บนบัส ต้องมีการอ้างถึงเสียก่อน โดยการอ้างถึงอุปกรณ์บนบัส I^2C นั้นจะใช้การอ้างถึงแบบ 7 บิต หรือ 10 บิต ในกรณีที่มีอุปกรณ์ต่ออยู่บนบัสไม่มาก ใช้การอ้างถึงแบบ 7 บิตก็เพียงพอ แต่ถ้ามีอุปกรณ์ต่ออยู่บนบัสมากกว่า 127 แอดเดรส จำเป็นต้องใช้การอ้างถึงแบบ 10 บิต หลังจากที่ติดต่ออุปกรณ์แต่ละตัวได้เรียบร้อยแล้ว ก็จะเริ่มต้นการถ่ายทอดข้อมูลกันต่อไป

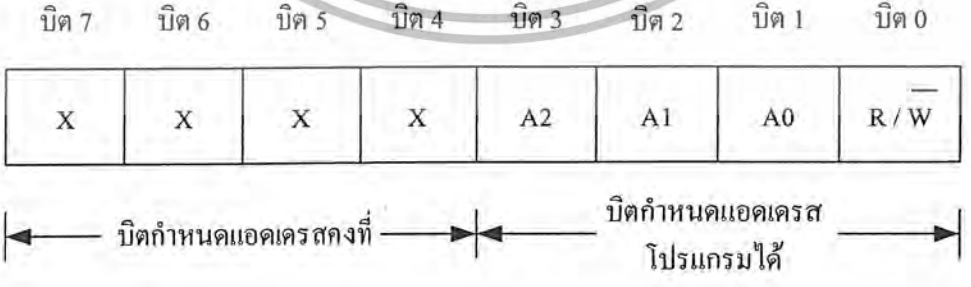
ดังนั้นหัวใจสำคัญในอันดับแรกของการทำงานบนบัส I^2C คือการอ้างถึงอุปกรณ์แต่ละตัว ซึ่งในที่นี้จะอธิบายรายละเอียดของการอ้างถึงทั้ง 2 รูปแบบ

1) การอ้างถึงแบบ 7 บิต (7-bit addressing)

ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสภาวะเริ่มต้นคือ ข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ที่ต้องการติดต่อหรือข้อมูลกำหนดแอดเดรส โดยมีรูปแบบแสดงในรูปที่ 2.17 ใน 7 บิตบนรวมทั้งบิต MSB ด้วย จะเป็นข้อมูลแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อ โดยแบ่งเป็น บิตกำหนดแอดเดรสคงที่ (Fixed address bit) จำนวน 4 บิต ซึ่งข้อมูลนี้อุปกรณ์แต่ละตัวจะถูกกำหนดมาจากผู้ผลิต ไม่สามารถเปลี่ยนแปลงแก้ไขได้ ถัดมาอีก 3 บิต เป็น บิตกำหนดแอดเดรสที่สามารถโปรแกรมได้ (Programmable address bit) โดยผู้ใช้งานต้องกำหนดสถานะลอจิกให้แก่ขา A0-A2 ของอุปกรณ์ที่มีการเชื่อมต่อแบบบัส I^2C ส่วนในบิต LSB เป็นบิตที่ใช้กำหนดการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวนั้นๆ หากบิต LSB เป็น “0” หมายถึงต้องการเขียนข้อมูลไปยังอุปกรณ์นั้น ถ้าเป็น “1” จะเป็นอ่านข้อมูลจากอุปกรณ์สเลฟ

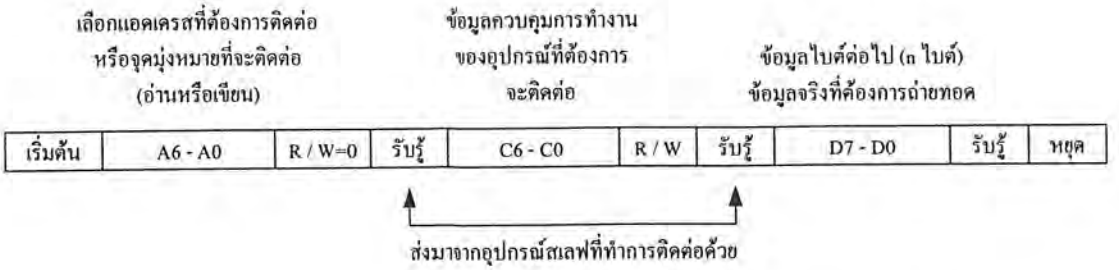
ข้อมูลในไบต์ต่อมาคือ ข้อมูลควบคุม (Control byte) ในอุปกรณ์แต่ละตัวมีการกำหนดข้อมูลควบคุมที่แตกต่างกันไป ยกตัวอย่าง ไอซีขยายพอร์ตมีข้อมูลควบคุมที่ใช้กำหนดว่า บิตใดเป็นอินพุต บิตใดเป็นเอาท์พุต ในขณะที่ไอซี ADC/DAC ต้องการข้อมูลควบคุมเพื่อกำหนดให้ทำงานเป็นวงจร ADC หรือ DAC เป็นต้น

ข้อมูลในไบต์ต่อมาคือ ข้อมูลที่ทำการถ่ายทอดจริง (Data) หลังจากที่มีการถ่ายทอดข้อมูลในแต่ละไบต์ อุปกรณ์สเลฟที่ได้รับการติดต่อต้องส่งสัญญาณรับรู้ตอบกลับมาด้วยทุกครั้ง เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้ ในรูปที่ 2.18 แสดงรูปแบบข้อมูลอนุกรมที่ได้เกิดขึ้นในการติดต่อบนบัส I^2C ของการอ้างถึงแบบ 7 บิต



รูปที่ 2.17 รูปแบบข้อมูลกำหนดแอดเดรสที่ใช้ในการอ้างอิงถึงแบบ 7 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

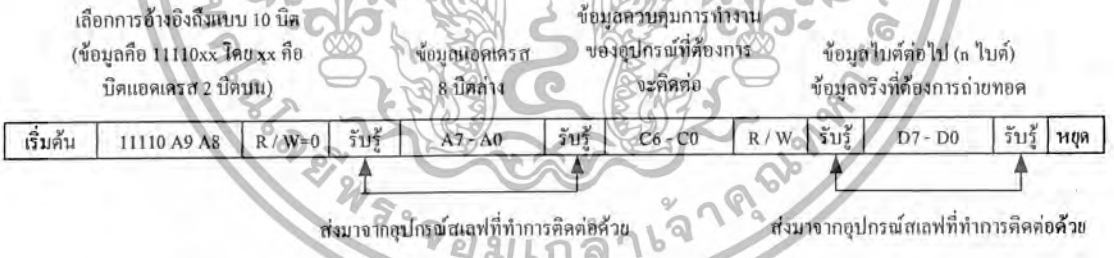


รูปที่ 2.18 รูปแบบของข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส I^2C เมื่อใช้การอ้างถึงแบบ 7 บิต

2) การอ้างถึงแบบ 10 บิต

ในการอ้างถึงแบบนี้ยังคงใช้รูปแบบข้อมูลอนุกรมที่เหมือนกับแบบ 7 บิต หากแต่จะมีข้อมูลเพิ่มเติมขึ้นมาเล็กน้อย โดยในข้อมูลไบต์แรกหลังจากเกิดสภาวะเริ่มต้น ต้องกำหนดให้ 5 บิตบนข้อมูลเป็น 11110 อีก 2 บิตถัดมาเป็นบิตแอดเดรสของอุปกรณ์ที่ต้องการติดต่อ ในบิต LSB ของข้อมูลไบต์แรกยังคงเป็นการกำหนดว่า ต้องการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวที่ต้องการติดต่อด้วย ข้อมูลไบต์ต่อมาเป็นข้อมูลแอดเดรสในไบต์ที่ 2 ของอุปกรณ์ที่ต้องการติดต่อด้วย ข้อมูลไบต์ถัดไปจึงเป็นข้อมูลควบคุม ข้อมูลหลังจากนั้นก็จะเป็นข้อมูลจริงที่ใช้ในการติดต่อ

เช่นเดียวกับการอ้างถึงแบบ 7 บิต หลังจากถ่ายทอดข้อมูลครบทุกไบต์ ต้องมีสภาวะรับรู้เกิดขึ้นเพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้ ในรูปที่ 2.19 แสดงรูปแบบข้อมูลอนุกรมของการอ้างถึงแบบ 10 บิต



รูปที่ 2.19 รูปแบบของข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส I^2C เมื่อใช้การอ้างถึงแบบ 7 บิต

2.11.5 อุปกรณ์ที่ใช้การเชื่อมต่อแบบบัส I^2C

ในปัจจุบัน บัส I^2C ได้รับความนิยมเพิ่มมากขึ้นเรื่อยๆ ด้วยข้อดีที่ชัดเจนคือ ใช้สายสัญญาณเพียง 2 เส้นเท่านั้น และการขยายระบบไมโครคอนโทรลเลอร์ที่มีจำนวนอินพุต เอาต์พุต และหน่วยความจำจำกัด สามารถทำได้ง่ายขึ้นด้วยระบบบัส I^2C เมื่อเป็นเช่นนี้จึงมีอุปกรณ์เพอริเฟอรัลที่ใช้การเชื่อมต่อแบบบัส I^2C มากมายจากหลายผู้ผลิตออกมาให้ใช้งานกัน ดังมีตัวอย่างดังต่อไปนี้

ไอซีขยายพอร์ตอินพุตเอาต์พุต (I/O expander) PCF8574, PCF8582, PCF8584

ไอซีหน่วยความจำอีอีพรอมอนุกรม (Serial EEPROM): 24Cxx, PCF8570, PCF72/73, PCF8582

ไอซี ADC/DAC : PCF8591

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอซีรีลไทม์คล็อก (Real-time clock: RTC): PCF8583, PCF8593, PCF8598, 41T56C

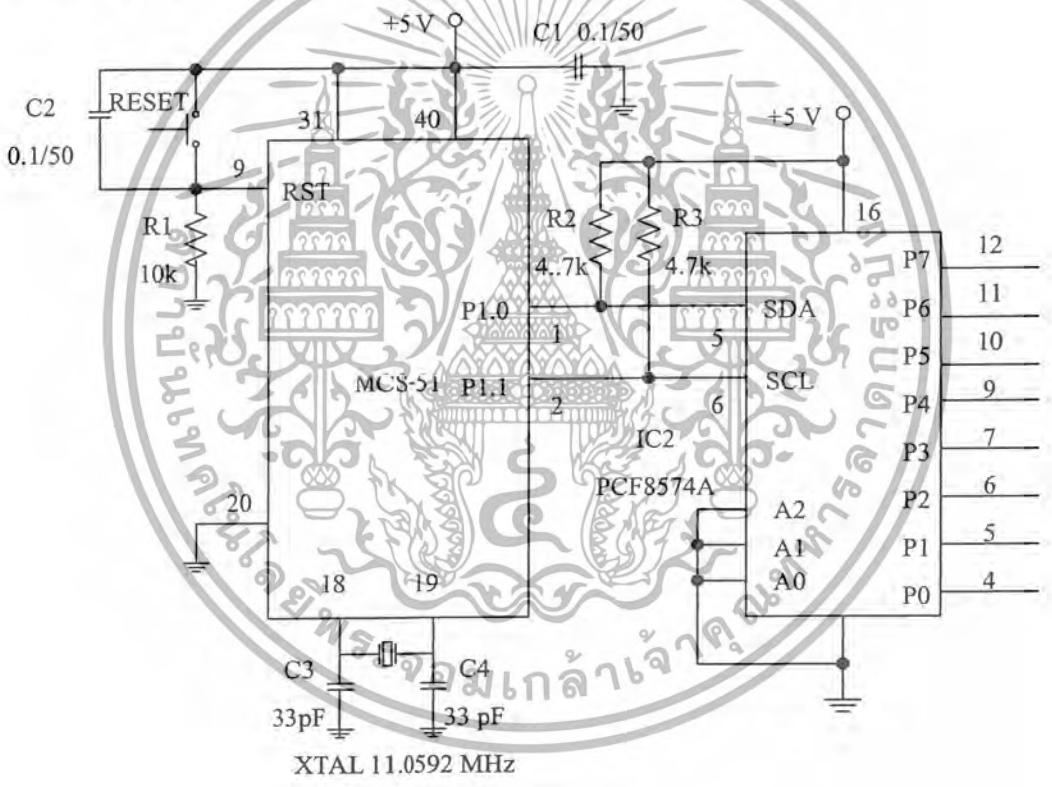
ไอซีขับ LCD โมดูล (LCD driver): PCF8466 , PCF8576 , PCF8577/78 , PCF8579 , SAA1064

ไอซีกำเนิดสัญญาณ DTMF (DTMF generator): PCD3311/12

2.11.6 การต่ออุปกรณ์ระบบบัส I^2C กับไมโครคอนโทรลเลอร์ MCS-51

สามารถทำได้ง่ายมาก เพียงใช้ขาพอร์ต 2 ขา โดยกำหนดให้ขาหนึ่งเป็น SDA อีกขาหนึ่งเป็น SCL และต่อตัวต้านทานค่าประมาณ 4.7 k พูลอัปที่ขาพอร์ตทั้งสองข้าง เพียงเท่านี้ก็สามารถติดต่อกับอุปกรณ์ระบบบัส I^2C ได้แล้ว

ในรูปที่ 2.20 เป็นวงจรตัวอย่างการต่อไมโครคอนโทรลเลอร์ MCS-51 เข้ากับระบบบัส I^2C จากวงจรจะใช้ขาพอร์ต P1.0 เป็นขา SDA และ P1.1 เป็นขา SCL อุปกรณ์ที่ทำการติดต่อด้วยคือไอซีขยายพอร์ตอินพุตเอาต์พุตเบอร์ PCF8574



รูปที่ 2.20 วงจรตัวอย่างการต่อไมโครคอนโทรลเลอร์ MCS-51 กับอุปกรณ์ระบบบัส I^2C

2.11.7 การเขียนโปรแกรมติดต่อบัส I^2C

เริ่มต้นด้วยการสรางสภาวะมาตรฐานของบัส I^2C อันประกอบด้วย สภาวะเริ่มต้น, สภาวะสิ้นสุดการส่งข้อมูล, สภาวะหยุด และสัญญาณนาฬิกาบนขา SCL

การสร้างสภาวะเริ่มต้น

1. เมื่อต้องการติดต่อกับบัส I^2C สิ่งแรกที่ต้องทำสำหรับโมโครคอนโทรลเลอร์ซึ่งถือว่าเป็นอุปกรณ์มาสเตอร์คือ การทำให้บัสว่างด้วยการกำหนดให้ขา SCL และ SDA มีลอจิกเป็น "1" ทั้งคู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จากนั้นทำให้ขา SDA มีลอจิก "0" โดยที่ขา SCL ยังคงเป็นลอจิก "1" อยู่
3. กำหนดให้ขา SCL มีลอจิกเป็น "0" ถึงตอนนี้ทั้ง SCL และ SDA จะมีลอจิกเป็น "0" ทั้งคู่ พร้อมที่จะติดต่อได้แล้ว

การสร้างสภาวะหยุด

1. เมื่อต้องการหยุดส่งข้อมูลจะต้องส่งสภาวะหยุดออไป โดยในตอนแรกต้องกำหนดให้ขา SCL และ SDA เป็นลอจิก "0" ทั้งคู่ก่อน
2. กำหนดให้ขา SCL มีลอจิกเป็น "1" โดย SDA ยังคงมีลอจิกเป็น "0"
3. จากนั้นทำให้ขา SDA มีลอจิกเป็น "1" ซึ่งจะทำให้ระบบบัสกลับเข้าสู่บัสว่างอีกครั้ง พร้อมที่จะรับหรือส่งข้อมูลต่อไป

การส่งข้อมูลลอจิก "0" และลอจิก "1"

หลังจากที่ทำการส่งบิตเริ่มต้นแล้ว ลำดับต่อไปคือ จะต้องส่งข้อมูลควบคุมซึ่งจะเป็นขบวนของลอจิก "0" และลอจิก "1" สำหรับการส่งข้อมูลลอจิก "0" ต้องดำเนินตามขั้นตอนดังนี้

1. ทำให้ขา SDA เป็น "0" สำหรับการส่งข้อมูลลอจิก "0"
2. ทำให้ขา SCL เป็น "1" สำหรับการป้อนสัญญาณนาฬิกา ในขณะที่ขา SDA ยังคงเป็น "0" อยู่
3. จากนั้นทำให้ขา SCL กลับมามีสถานะเป็นลอจิก "0" เหมือนเดิม

ในขณะที่การส่งข้อมูลลอจิก "1" มีขั้นตอนดังนี้

1. ทำให้ขา SDA เป็น "1" สำหรับการส่งข้อมูลลอจิก "1"
2. ทำให้ขา SCL เป็น "1" สำหรับการป้อนสัญญาณนาฬิกา ในขณะที่ขา SDA ยังคงเป็น "1" อยู่
3. จากนั้นทำให้ขา SCL กลับมามีสถานะเป็นลอจิก "0" เหมือนเดิม

2.12 แนะนำ DS1307 ไอซีสร้างฐานเวลาจริงหรือรีลไทม์คล็อก (RTC)

ผู้ผลิตคือ ดัลลัสเซมิคอนดักเตอร์ (Dalllas semiconductor) มีหน้าที่สร้างฐานเวลาจริงให้แก่ระบบไมโครคอนโทรลเลอร์ โดย DS1307 จะให้ข้อมูลเกี่ยวกับเวลาทั้งหมด ไม่ว่าจะเป็นค่าของเวลาที่ละเอียดถึงหลักวินาที, นาที, ชั่วโมง, วันที่ (date), วันในสัปดาห์ (day), เดือน และปี โดยสามารถปรับวันเดือนปีให้ตรงตามปฏิทินได้อย่างถูกต้อง รวมถึงการกำหนดวันในปีอธิกสุรทินด้วย คุณสมบัติทางเทคนิคที่สำคัญมีดังนี้

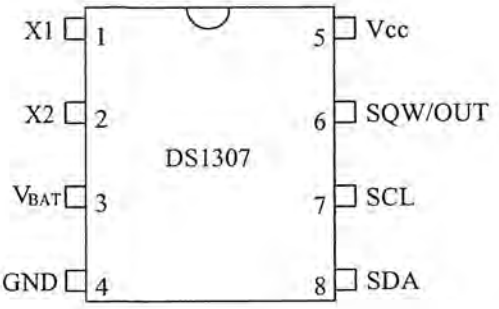
- เป็นไอซีรีลไทม์คล็อกให้ข้อมูลตั้งแต่วินาทีจนถึงปี รวมถึงการปรับวันในปีอธิกสุรทินด้วย สามารถให้ข้อมูลเวลาได้อย่างเที่ยงตรงถึงปีคริสตศักราช 2100
- มีหน่วยความจำนอนโวลาไทล์แรม 56 ไบต์อยู่ภายใน สามารถใช้เก็บข้อมูลทั่วไปได้

ใช้การเชื่อมต่อแบบระบบบัส I^2C

- มีวงจรตรวจจับไฟเลี้ยงต่ำหรือหายไปอย่างอัตโนมัติ และสามารถรักษาข้อมูลเวลาไว้ได้แม้ไม่มีไฟเลี้ยงไอซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12.1 รายละเอียดขาต่อใช้งานของ DS1307



รูปที่ 2.21 แสดงการจัดขาของ DS1307

แต่ละขามีหน้าที่และการใช้งานดังนี้

V_{CC}, GND (ขา 8, 4) ต่อกับไฟเลี้ยง +5V

V_{BAT} (ขา 3) ใช้ต่อกับแบตเตอรี่ 3 V เพื่อรักษาการทำงานของวงจรสร้างฐานเวลาของ DS1307 ให้คงอยู่ต่อไป แม้ว่าไม่มีไฟเลี้ยงจ่ายให้แก่ DS1307 ชนิดของแบตเตอรี่ที่เหมาะสมคือ แบตเตอรี่แบบลิเธียม ซึ่งมีความจุ 40 mAhr หรือมากกว่า จะสามารถรักษาข้อมูลได้นาน 10 ปี ที่อุณหภูมิ 25 องศาเซลเซียส

SDA, SCL (ขา 5 และ 6) เป็นขาสำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์บนระบบบัส I^2C

SQW/OUT (ขา 7) ที่ขานี้ จะมีสัญญาณรูปสี่เหลี่ยมส่งออกมา โดยสามารถเลือกความถี่ได้ 1Hz, 4.096 kHz, 8.192 kHz และ 32 kHz ในการใช้งานต้องต่อตัวต้านทาน 1 k พูลอัปที่ขานี้ด้วย

X1, X2 (ขา 1 และ 2) ใช้ต่อกับคริสตอลความถี่มาตรฐาน 32.768 kHz เพื่อใช้เป็นฐานเวลาในการสร้างค่าเวลาจริง ในการใช้งานต้องต่อคริสตอลข้ากับขาทั้งสองข้างนี้และที่แต่ละขาต้องต่อตัวเก็บประจุค่าต่ำๆ ประมาณ 15 pF คร่อมกับขากราวน์ด้วย

2.12.2 การทำงานของ DS1307

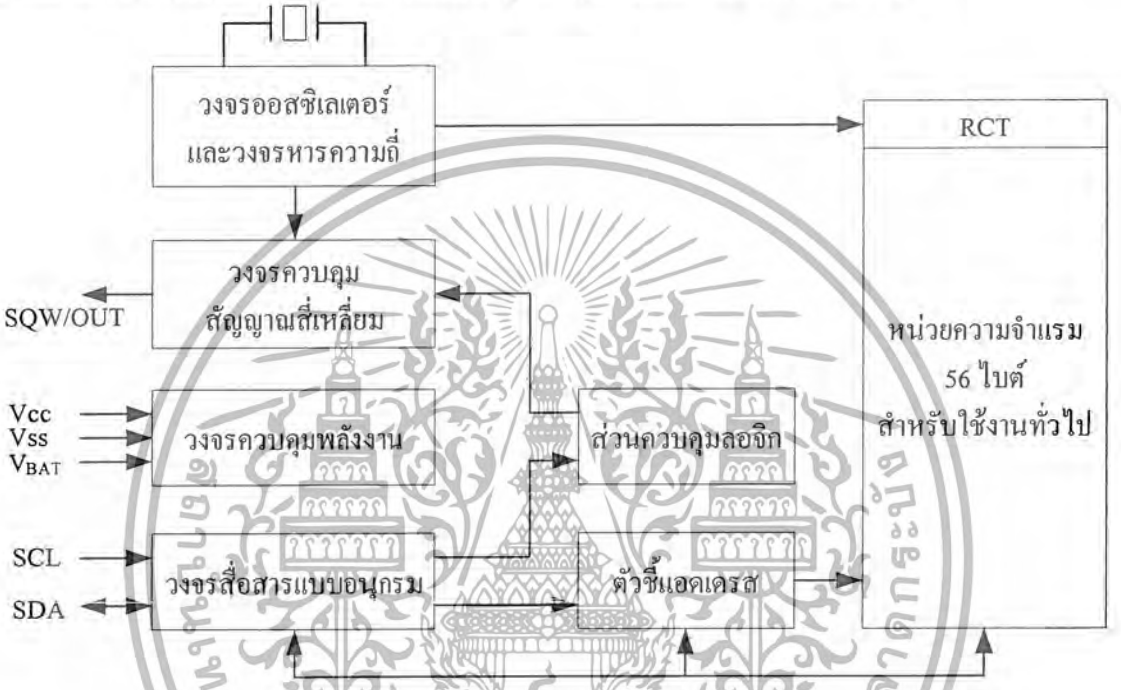
ไอซี DS1307 จัดการเชื่อมต่อในแบบบัส I^2C โดยจะทำงานเป็นอุปกรณ์สเลฟเสมอ ดังนั้นการติดต่อเพื่อใช้งานจึงต้องกำหนดรูปแบบตามที่กำหนดไว้ในการติดต่อแบบ ในรูปที่ 2.22 แสดงส่วนประกอบหลักที่สำคัญและไดอะแกรมการทำงานของ DS1307 วงจรออสซิลเลเตอร์ถือเป็นหัวใจหลักของไอซี เนื่องจากเป็นจุดเริ่มต้นของการสร้างข้อมูลเวลาจริง ในขณะที่ DS1307 ทำงานที่ขา SQW/OUT จะมีสัญญาณพัลส์สี่เหลี่ยมส่งออกมาตลอดเวลาในกรณีที่มีการอีนาเบิลวงจรกำเนิดสัญญาณพัลส์ที่รีจิสเตอร์ควบคุม ค่าความถี่ของสัญญาณนี้สามารถเลือกได้ 4 ค่าคือ 1 Hz , 4.096 kHz , 8.192 kHz และ 32 kHz พร้อมกันนั้นก็จะมีการเก็บค่าของเวลาไว้ในหน่วยความจำนอนโวลาไทล์แรม ซึ่งมีขนาดรวม 64 ไบต์ แต่จัดสรรให้ใช้เก็บข้อมูลเวลา 8 ไบต์ และเป็นหน่วยความจำสำหรับเก็บข้อมูลทั่วไปสำหรับผู้ใช้งานอีก 56 ไบต์

วงจรควบคุมพลังงานไฟฟ้าจะคอยตรวจสอบสภาวะของไฟเลี้ยงไอซี หากไฟเลี้ยงต่ำกว่า 1.25(V_{BAT}) ก็จะควบคุมให้ DS1307 หยุดการทำงาน รีเซตค่าตัวนับแอดเดสภายในทำให้ไม่สามารถติดต่อกับ DS1307 ได้ ดังนั้นในการใช้งาน DS1307 ต้องระมัดระวังอย่าให้ไฟเลี้ยงตกต่ำกว่า 1.25(V_{BAT}) หรือประมาณ 3.75 โวลต์ ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ใช้ V_{BAT} เท่ากับ 3 โวลต์ หากไฟเลี้ยงมีค่าต่ำกว่าV_{BAT} ไอซี DS1307 จะเข้าสู่โหมดสำรองข้อมูลกระแสต่ำทันที จะไม่มีการส่งสัญญาณพัลส์ออกมาที่ขา SQW/OUT แต่วงจรสร้างฐานเวลายังคงทำงานต่อไป เพื่อให้ค่าของเวลาเดินไปอย่างไม่ผิดพลาด เมื่อมีไฟเลี้ยงปรากฏขึ้นอีกครั้ง DS1307 ก็จะสามารถให้ค่าเวลาที่เป็นจริงแก่ผู้ใช้งานได้ต่อไป

วงจรสื่อสารอนุกรมภายใน DS1307 ได้รับการกำหนดให้ทำงานตามรูปแบบของบัส I^2C เป็นช่องทางการสื่อสารระหว่าง DS1307 กับอุปกรณ์มาสเตอร์ ผู้ใช้งานสามารถเข้าถึงหน่วยความจำที่ใช้เก็บค่าเวลาและหน่วยความจำใช้งานทั่วไปได้โดยการเขียนข้อมูลตามรูปแบบที่กำหนดในระบบบัส I^2C



รูปที่ 2.22 โครงสร้างภายในของไอซีรีลไทม์คล็อกเบอร์ DS1307

2.12.3 การจัดสรรหน่วยความจำใน DS1307

ในรูปที่ 2.23 (ก) แสดงการจัดสรรพื้นที่ของหน่วยความจำภายในของ DS1307 พื้นที่ 7 ไบต์แรก ตั้งแต่แอดเดส 00H-06H เป็นพื้นที่ของรีจีสเตอร์ค่าเวลาที่ใช้เก็บข้อมูลเกี่ยวกับเวลา ไบต์ต่อมาที่แอดเดส 07H เป็นพื้นที่ของรีจีสเตอร์ควบคุมการทำงานของ DS1307 ในรูปที่ 2.23 (ข) แสดงรายละเอียดของรีจีสเตอร์ค่าเวลา และรีจีสเตอร์ควบคุม

ด้วยการจัดสรรพื้นที่แบบนี้ ทำให้ผู้ใช้งานสามารถเรียกข้อมูลเวลาออกมาได้ตามต้องการ โดยไม่จำเป็นต้องอ่านออกมาทั้งหมดก็ได้ ค่าของเวลาทั้งหมดจะอยู่ในรูปของเลขฐาน 16 สำหรับการแสดงค่าเวลาในรูปของชั่วโมงสามารถเลือกได้ว่าต้องการแบบ 12 หรือ 24 ชั่วโมง โดยการกำหนดที่บิต 6 ของแอดเดส 02H และเมื่อเลือกแบบ 12 ชั่วโมง ที่บิตที่ 5 ในแอดเดสเดียวกันจะใช้ในการแสดงค่า AM/PM โดยบิตนี้มีค่าเป็น "1" หมายถึง ค่าชั่วโมงในขณะนั้นเป็นช่วงเวลาหลังเที่ยงวัน ในกรณีที่เป็นแบบ 24 ชั่วโมง บิตนี้จะใช้ในการแสดงค่า 2 ของหลักสิบในหน่วยชั่วโมง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0	ค่าของข้อมูล
00H	วินาที	CH	ข้อมูลวินาที (หลักสิบ)			ข้อมูลวินาที (หลักหน่วย)				00-59
	นาที	x	ข้อมูลนาที (หลักสิบ)			ข้อมูลวินาที (หลักหน่วย)				00-59
	ชั่วโมง	x	12 ชั่วโมง / 24 ชั่วโมง	ชั่วโมง(หลักสิบ) AM/PM	ข้อมูลชั่วโมง (หลักสิบ)	ข้อมูลชั่วโมง (หลักหน่วย)				01-12 / 00-23
	วัน	x	x	x	x	x	ข้อมูลวันในสัปดาห์			1-7
	วันที่	x	x	ข้อมูลวันที่ (หลักสิบ)		ข้อมูลวันที่ (หลักหน่วย)				01-28/29 / 01-30 / 01-31
	เดือน	x	x	x	ข้อมูลเดือน (หลักสิบ)	ข้อมูลเดือน (หลักหน่วย)				01-12
	ปี	ข้อมูลปี (หลักสิบ)				ข้อมูลปี (หลักหน่วย)				00-99
07H	รีจิสเตอร์ควบคุม	OUT	x	x	SQWE	x	x	RS1	RS0	
08H	แรม 56 ไบต์									
3FH										

(ก) (ข)

รูปที่ 2.23 (ก) แสดงการจัดสรรพื้นที่ของหน่วยความจำภายในของ DS1307
(ข) แสดงรายละเอียดของรีจิสเตอร์ค่าเวลา และรีจิสเตอร์ควบคุม

2.12.4 รีจิสเตอร์ควบคุม

มีแอดเดสอยู่ที่ 07H มีรายละเอียดของแต่ละบิตดังนี้

OUT (Output control): ใช้ในการควบคุมระดับลอจิกที่ขา SQW/OUT ในกรณีที่ดิสเอเบิลการกำเนิดสัญญาณสี่เหลี่ยม โดยถ้าบิตนี้มีค่าเป็น "1" ที่ขา SQW/OUT ก็จะเป็นค่า "1" และถ้าบิตนี้มีค่าเป็น "0" ที่ขา SQW/OUT ก็จะเป็นค่า "0"

SQWE (Square Wave Enable): ใช้ในการอีนาเบิลวงจรกำเนิดสัญญาณสี่เหลี่ยมที่ขา SQW/OUT ถ้าต้องการให้มีสัญญาณสี่เหลี่ยมออก ให้กำหนดบิตนี้เป็น "1"

RS1, RS0 (Rate Select): ใช้ในการเลือกความถี่ของสัญญาณสี่เหลี่ยมที่ออกจากขา SQW/OUT มีรายละเอียดต่อไปนี้

RS1	RS0	ค่าความถี่ของสัญญาณสี่เหลี่ยม
0	0	1 Hz
0	1	4.096 kHz
1	0	8.192 kHz
1	1	32.768 kHz

2.12.5 โหมดการทำงานของ DS1307

มีรูปแบบการทำงานอยู่ 2 โหมดคือโหมดเขียนข้อมูล และโหมดอ่านข้อมูล ในการใช้งาน DS1307 ตามปกติจะใช้งานเฉพาะโหมดอ่านข้อมูลเท่านั้น เนื่องจากไมโครคอนโทรลเลอร์จะติดต่อกับ DS1307 เพื่ออ่านข้อมูลของเวลาไปใช้งาน โหมดการเขียนข้อมูลจะถูกใช้งานก็ต่อเมื่อต้องการตั้งค่าเวลาใหม่ และต้องการเขียนข้อมูลลงในหน่วยความจำใช้งานทั่วไป โดยเมื่อเริ่มต้นการติดต่อกับ DS1307 ก็ต้องใช้งานโหมดการเขียนข้อมูลเพื่อกำหนดแอดเดสที่ต้องการอ่านข้อมูล จากนั้นจึงเปลี่ยนไปเป็นโหมดอ่านข้อมูลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

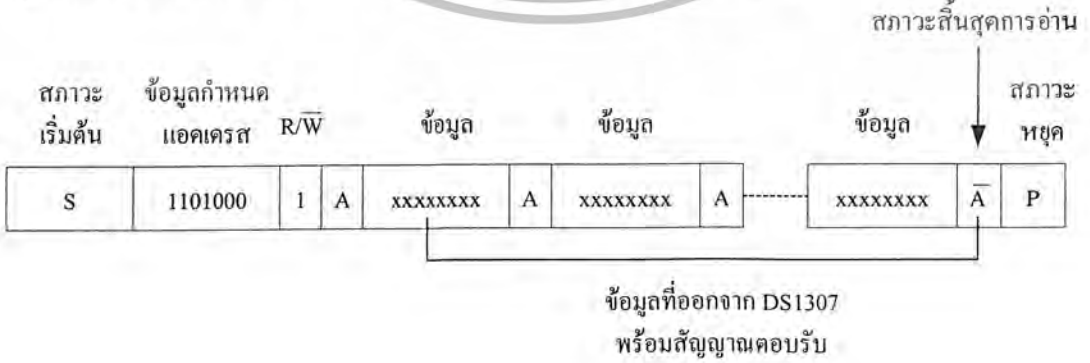
1) โหมดการเขียนข้อมูล

มีรูปแบบดังรูปที่ 2.24 เริ่มต้นเมื่อไมโครคอนโทรลเลอร์ทำการกำหนดสภาวะเริ่มต้น (Start: S) จากนั้นส่งข้อมูลกำหนดแอดเดส 1101000 ตามด้วยข้อมูลเลือกการเขียน นั่นคือค่า "0" จากนั้นจะรอรับการตอบรับจาก DS1307 ขั้นตอนถัดมาก็คือ ส่งข้อมูลเพื่อเลือกแอดเดสที่ต้องการเขียน จากนั้นจะรอการตอบรับจาก DS1307 เมื่อมีการตอบรับเรียบร้อยแล้ว ก็จะเริ่มทยอยเขียนข้อมูลลงไปครั้งละแอดเดส หลังจากเขียนข้อมูลลงในแต่ละแอดเดสแล้วจะต้องรอการตอบรับจาก DS1307 ทุกครั้ง จึงสามารถเขียนข้อมูลถัดไปได้ เมื่อเขียนเรียบร้อยแล้ว ให้ส่งสภาวะหยุด (Stop: P) เป็นอันเสร็จสิ้นกระบวนการเขียนข้อมูล

รูปที่ 2.24 รูปแบบข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการเขียนข้อมูล

2) โหมดการอ่านข้อมูล

มีรูปแบบดังรูปที่ 2.25 เริ่มต้นการทำงานเหมือนกับโหมดการเขียนข้อมูลคือไมโครคอนโทรลเลอร์กำหนดสภาวะเริ่มต้น แล้วส่งข้อมูลกำหนดแอดเดส แล้วตามด้วยข้อมูลเลือกการอ่าน ซึ่งเท่ากับ "1" จากนั้นรอรับการตอบรับจาก DS1307 เมื่อตอบรับเรียบร้อยแล้ว DS1307 จะทยอยส่งข้อมูลออกมาให้ไมโครคอนโทรลเลอร์ครางละ 1 แอดเดส หรือ 1 ไบต์ โดยแอดเดสที่เลือกอ่านข้อมูลจะต้องมีการกำหนดมาก่อนล่วงหน้าด้วยโหมดการเขียนข้อมูล วิธีการง่ายๆคือ เข้าสู่โหมดการเขียนข้อมูลก่อน เมื่อถึงจังหวะที่ต้องเขียนข้อมูลให้ทำการสร้างสภาวะเริ่มต้น และส่งข้อมูลแอดเดสใหม่อีกครั้ง ตามด้วยโหมดการอ่านข้อมูล ข้อมูลที่ออกมาจาก DS1307 ก็จะเป็นข้อมูลจากแอดเดสที่กำหนดไว้ก่อนหน้านี้

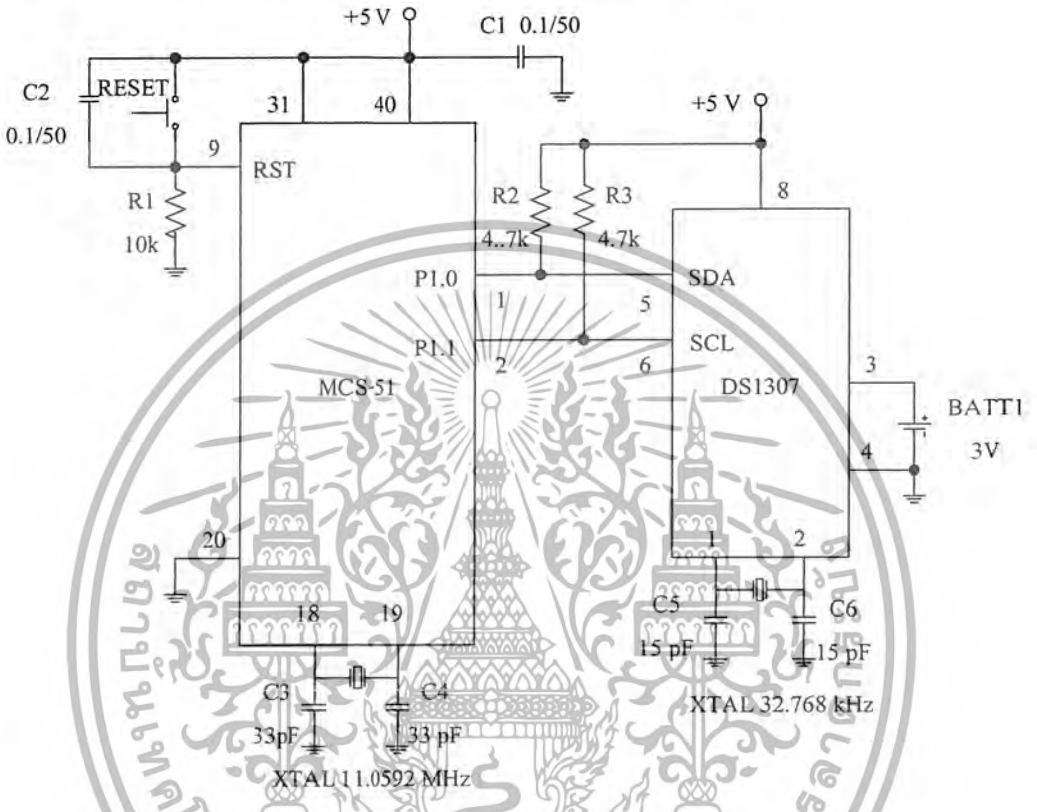


รูปที่ 2.25 รูปแบบข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการอ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12.6 การเชื่อมต่อ DS1307 กับไมโครคอนโทรลเลอร์ MCS-51

การต่อมีลักษณะการต่อเหมือนอุปกรณ์บนระบบบัส I^2C ตัวอื่นๆทุกประการ และสามารถที่จะต่อไอซีทั้งหมดร่วมกับบนสาย SDA และ SCL ได้ โดยจะทำการต่อกันดังรูป 2.26 ไอซี DS1307 จำเป็นต้องต่อแบตเตอรี่ไว้ตลอดเวลา ไม่ว่าจะใช้งานหรือไม่ ทั้งนี้เพื่อรักษาการทำงานของวงจรภายใน DS1307 ให้ยังคงทำงานต่อเนื่องไปเมื่อใดที่ไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลก็จะได้ข้อมูลเวลาที่เป็นจริงตลอด



รูปที่ 2.26 การเชื่อมต่อไมโครคอนโทรลเลอร์ MCS-51 กับไอซีรีลไทม์คล็อก DS1307

2.13 รอม (ROM: Read Only Memory)

เป็นหน่วยความจำแบบถาวร ต่างจาก RAM (Random Access Memory) ตรงที่ เมื่อขาดไฟเลี้ยงข้อมูลที่เก็บอยู่ภายในจะไม่สูญหาย เป็นหน่วยความจำชนิดที่อ่านได้อย่างเดียว แต่มีรอมบางชนิด ที่สามารถเขียนหรือลบข้อมูลได้ ชนิดของรอมที่มีเทคโนโลยีการเขียนและการลบ มีดังต่อไปนี้

1. PROM (Programmable Read Only Memory) เป็นหน่วยความจำชนิดที่สามารถโปรแกรมข้อมูลลงไปได้ ข้อมูลที่โปรแกรมลงไปจะอยู่ถาวรไม่สามารถลบข้อมูลได้

2. EPROM (Erasable Programmable Read Only Memory) เป็นหน่วยความจำที่สามารถลบข้อมูลได้และโปรแกรมข้อมูลลงไปใหม่ได้ โดยการลบข้อมูล สามารถทำโดยใช้รังสี UV-B เมื่อทำการลบข้อมูลแล้วจะสามารถโปรแกรมข้อมูลใหม่ได้

3. EEPROM (Electrically Erasable Programmable Read Only Memory) เป็นหน่วยความจำที่สามารถลบข้อมูลและโปรแกรมข้อมูลได้โดยการใช้ไฟฟ้า

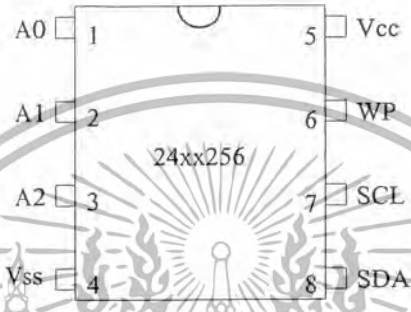
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานนนี้จะนำเทคโนโลยีของ EEPROM มาใช้ในการเก็บรหัสของผู้เข้าใช้งาน ในกรณีที่ระบบไฟฟ้าขัดข้อง ข้อมูลผู้เข้าใช้งานจะได้ไม่สูญหาย

2.14 แนะนำไอซี 24LC256 CMOS Serial EEPROM

เป็นหน่วยความจำแบบ EEPROM ที่มีระบบบัสแบบ 2-wire serial สนับสนุนการทำงานระบบบัส I^2C สามารถลบและโปรแกรมข้อมูลได้มากถึง 1 ล้านครั้ง สามารถเก็บข้อมูลได้ 256 กิโลบิต

2.14.1 รายละเอียดของขาที่ต่อใช้งาน



รูปที่ 2.27 การจัดขาของไอซี 24LC256

หน้าที่การทำงานของแต่ละขา มีดังต่อไปนี้

A0, A1, A2 ใช้ระบุว่าจะเลือกใช้หน่วยความจำตัวใด กรณีที่มีการต่อหน่วยความจำหลายตัว

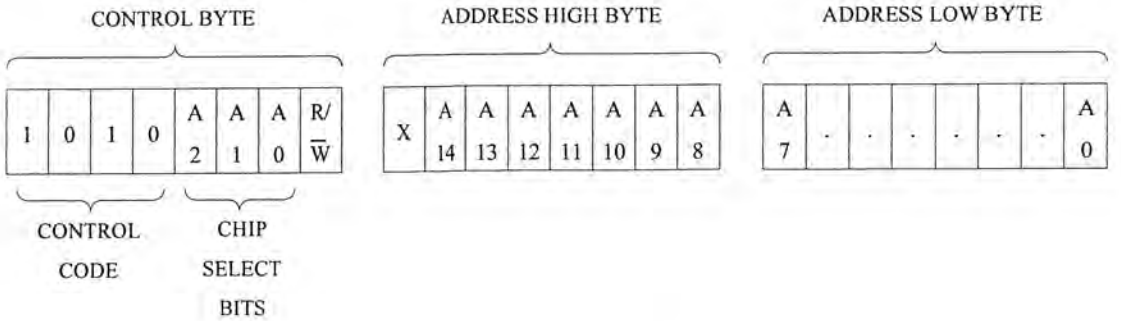
SDA (Serial Data) ทำหน้าที่ส่งแอดเดรสและข้อมูลเข้าและออกอุปกรณ์

SCL (Serial Clock) ทำหน้าที่รักษาจังหวะเวลาในการส่งข้อมูลเข้าออกอุปกรณ์

WP (Write Protect) จะป้องกันการเขียนข้อมูลถ้าต่อกับ Vcc แต่ยังสามารถอ่านข้อมูลได้

2.14.2 การระบุแอดเดรสของอุปกรณ์

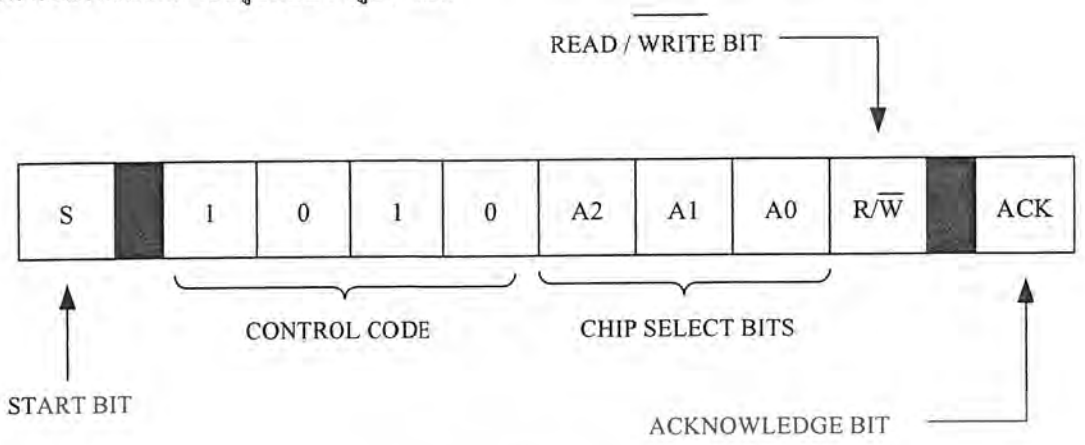
การส่งข้อมูลแอดเดรสจะต้องประกอบด้วย คอนโทรลไบต์ ตามด้วย แอดเดรสไบต์ ซึ่งสามารถแสดงรูปแบบของการส่งข้อมูลแอดเดรสของอุปกรณ์ ดังรูป 2.28



รูปที่ 2.28 การระบุแอดเดรสของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ คอนโทรลไบต์ จะมีรูปแบบ ดังรูปที่ 2.29



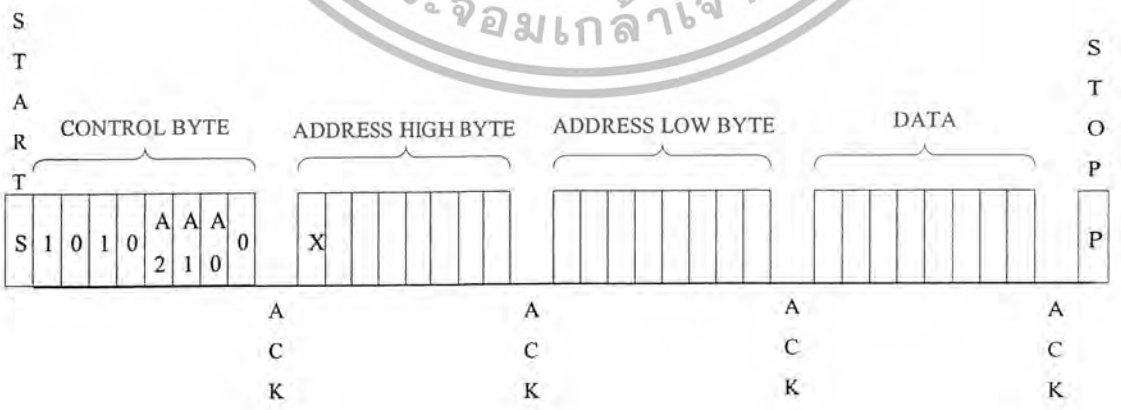
รูปที่ 2.29 รูปแบบของคอนโทรลไบต์

มีรายละเอียดดังนี้

1. Control Code เป็นรหัสเฉพาะของไอซีแต่ละรุ่น ในตระกูล 24XX256 ใช้ 1010 เป็น Control Code
2. Chip select bits ใช้ระบุอุปกรณ์ที่ต้องการติดต่อ
3. R/$\overline{W}$ ใช้ระบุว่าเป็นกระบวนการอ่านข้อมูลหรือเขียนข้อมูล

2.14.3 กระบวนการเขียนข้อมูล

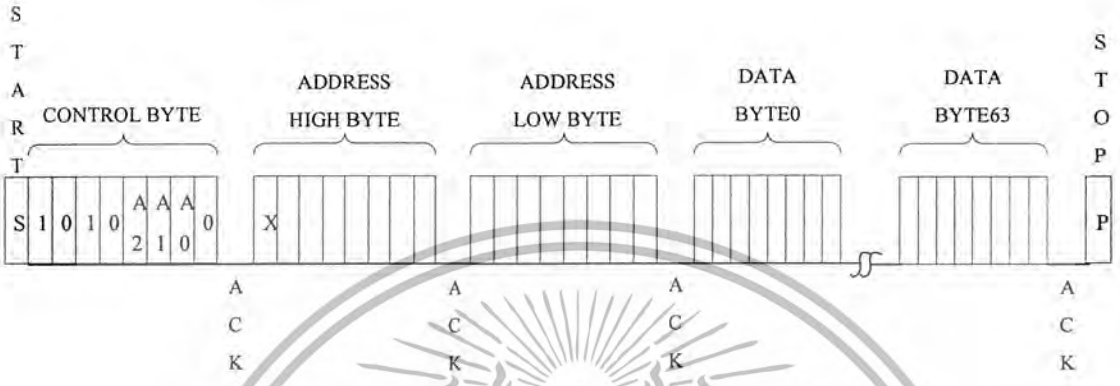
กระบวนการเขียนข้อมูล ทำโดยการส่งคอนโทรลไบต์ โดยการระบุ Control code (1010 สำหรับตระกูล 24XX256) ระบุอุปกรณ์ และระบุว่าเป็นกระบวนการเขียนโดย R/$\overline{W}$ = 0 ต่อจาก คอนโทรลไบต์จะตามด้วยแอดเดรสที่จะทำการเขียนข้อมูล และตามด้วยข้อมูลที่จะเขียน และปิดท้ายด้วย Stop Condition เพื่อบอกว่าสิ้นสุดการเขียนข้อมูล โดยสามารถแสดงกระบวนการเขียนข้อมูลได้ดังรูปที่ 2.30



รูปที่ 2.30 กระบวนการเขียนข้อมูล 1 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่ต้องการเขียนข้อมูลต่อเนื่องกันไป โดยเริ่มตั้งแต่ แอดเดรสแรกเรียงกันไป สามารถเขียนได้ติดต่อกัน 64 ไบต์ โดยในกระบวนการเขียนจะคล้ายกับกระบวนการเขียนข้อมูล 1 ไบต์ ต่างกันที่เมื่อสิ้นสุดข้อมูลไบต์แรก ก็ให้ส่งข้อมูลที่ต้องการเขียนต่อไป เรื่อยๆ จนครบแล้วจึงใส่ Stop Condition โดยกระบวนการเขียนข้อมูลต่อเนื่องแสดงได้ดังรูปที่ 2.31



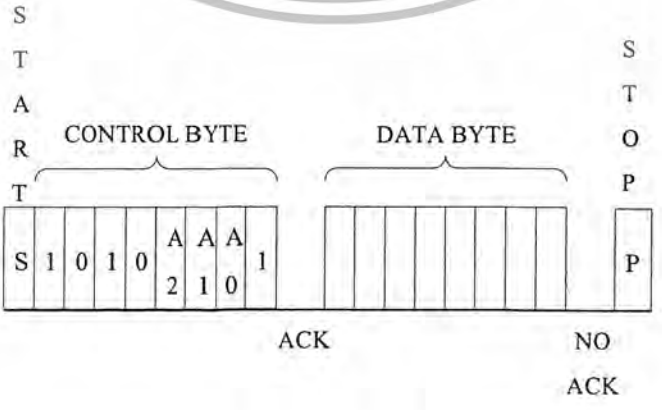
รูปที่ 2.31 กระบวนการเขียนข้อมูลต่อเนื่อง

2.14.4 กระบวนการอ่านข้อมูล

กระบวนการอ่านข้อมูล ทำโดยการส่งคอนโทรลไบต์ โดยการระบุ Control code (1010 สำหรับตระกูล 24XX256) ระบุอุปกรณ์ และระบุว่าเป็นกระบวนการอ่านโดย R/$\overline{W}$ = 1 โดยกระบวนการอ่านข้อมูลจะแบ่งออกเป็น 3 ลักษณะ ดังนี้

1. การอ่านข้อมูล 1 ไบต์ ที่แอดเดรสปัจจุบัน

ทำโดยส่งข้อมูล Control code โดย ระบุว่าเป็นกระบวนการอ่านโดย R/$\overline{W}$ = 1 เมื่อ 24XX256 ได้รับข้อมูลคอนโทรลไบต์ ก็จะสร้างสัญญาณ Acknowledge แล้วตามด้วยข้อมูลที่ต้องการอ่านออกมา แสดงดังรูปที่ 2.32

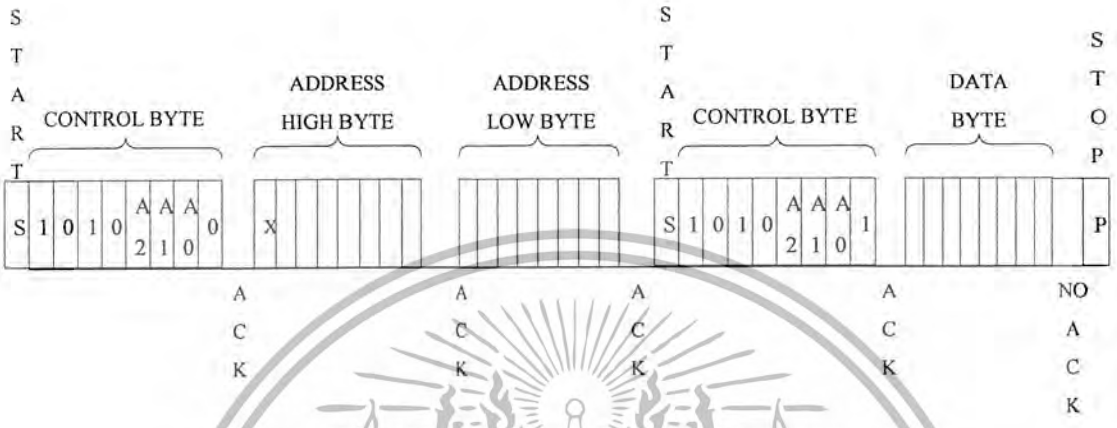


รูปที่ 2.32 การอ่านข้อมูล 1 ไบต์ ที่แอดเดรสปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การอ่านข้อมูล 1 ไบต์ โดยระบุแอดเดรส

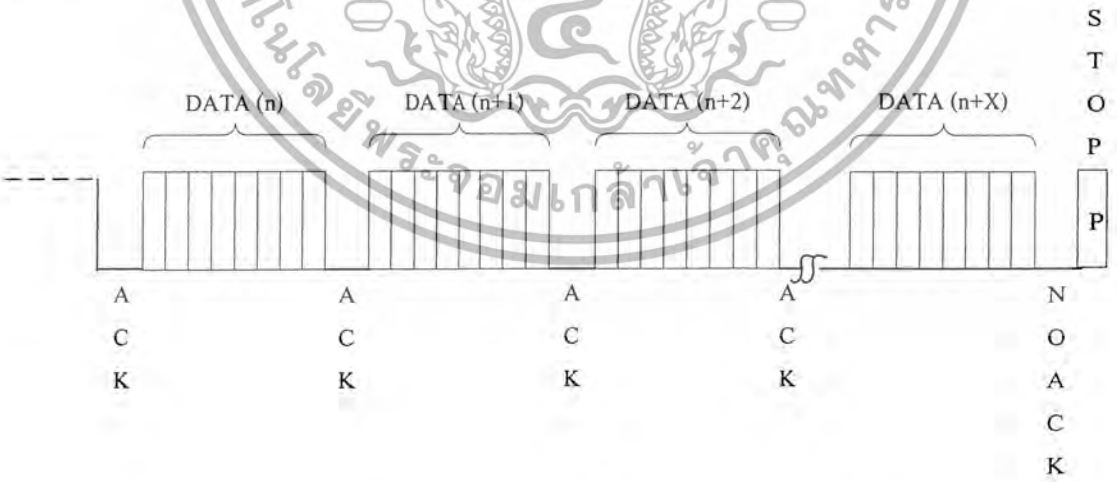
ทำโดยส่งข้อมูล Control code โดย ระบุว่าเป็นกระบวนการเขียนโดย R/$\overline{W}$ = 0 ตามด้วยแอดเดรสที่ต้องการจะอ่านข้อมูล เมื่อได้ 24XX256 รับข้อมูลก็จะส่ง Control byte ระบุว่าเป็นการอ่าน และตามด้วยข้อมูลที่อ่านได้ 1 ไบต์ แสดงดังรูปที่ 2.33



รูปที่ 2.33 การอ่านข้อมูล 1 ไบต์ โดยระบุแอดเดรส

3. การอ่านข้อมูล ต่อเนื่องตั้งแต่แอดเดรสที่ระบุต่อเนื่องตามลำดับ

ทำโดยส่งข้อมูล Control code โดย ระบุว่าเป็นกระบวนการเขียนโดย R/$\overline{W}$ = 0 ตามด้วยแอดเดรสเริ่มต้นที่ต้องการจะอ่านข้อมูล เมื่อได้ 24XX256 รับข้อมูลก็จะส่ง Control byte ระบุว่าเป็นการอ่าน และตามด้วยข้อมูลที่อ่านได้ตั้งแต่แอดเดรสที่ระบุไปสิ้นสุดที่แอดเดรสสุดท้าย และจึงสิ้นสุดด้วย Stop condition แสดงดังรูปที่ 2.34



รูปที่ 2.34 การอ่านข้อมูล ต่อเนื่องตั้งแต่แอดเดรสที่ระบุต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและออกแบบ

โครงงานนี้ จะแบ่งออกเป็น 2 ส่วน ได้แก่ ส่วนที่ติดต่อกับไมโครคอนโทรลเลอร์ และส่วนที่ติดต่อกับคอมพิวเตอร์ โดยแต่ละส่วนจะมีการออกแบบดังต่อไปนี้

ส่วนที่ติดต่อกับไมโครคอนโทรลเลอร์

3.1 การรับค่ารหัสจากเครื่องอ่านบาร์โค้ด

ผู้จัดทำได้เลือกใช้เครื่องอ่านบาร์โค้ด Arextek รุ่น 103DR โดยจะให้สัญญาณ Output ออกมาเป็นสัญญาณอนุกรม รูปแบบของสัญญาณเป็นรหัส ASCII ที่ได้จากการแปลงรหัสแถบจากบัตร โดยสามารถรองรับมาตรฐานของรหัสแถบได้หลายชนิด โดยรายละเอียดของเครื่องอ่านบาร์โค้ดรุ่นนี้ สามารถดูได้ที่ภาคผนวก โดยเราสามารถกำหนดค่า Parameter ต่างๆของตัวเครื่องได้ การที่เราจะออกแบบส่วนที่จะทำการรับค่ารหัสจากเครื่องอ่านบาร์โค้ด เราต้องทราบเกี่ยวกับมาตรฐานของตัวเครื่อง และรูปแบบของข้อมูลที่ออกมาจากเครื่อง

เนื่องจากค่ามาตรฐานของเครื่องอ่านบาร์โค้ดมีดังนี้

1) อัตราการส่งข้อมูล 9600 บิตต่อวินาที
2) Data Bit 8 bit มี 1 Stop Bit
3) ไม่มีการตรวจสอบพาริตี้ (Non-parity Checking)

เราจึงต้องทำการเขียนโปรแกรม ไมโครคอนโทรลเลอร์ เพื่อรับค่าจากเครื่องอ่านบาร์โค้ด โดยกำหนดค่าที่ Register ต่างๆดังนี้

รีจิสเตอร์	ค่าที่ใช้	เหตุผล
TMOD	20H	เป็นการกำหนดค่า Data Bit 8 bit
SCON	50H	เป็นการกำหนดโหมดการรับส่งข้อมูล
TH1	0FDH	เป็นการกำหนดอัตราการส่งข้อมูล 9600 บิตต่อวินาที

ตารางที่ 3.1 แสดงค่ารีจิสเตอร์ต่างๆที่ใช้ในการเขียนโปรแกรมรับค่าจากเครื่องอ่านบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากเครื่องอ่านบาร์โค้ดสามารถอ่านค่ารหัสนักศึกษาแล้วแปลงเป็นรหัส ASCII 8 ไบต์ ดังนั้นในการเขียนโปรแกรมรับค่าจากเครื่องอ่านบาร์โค้ด จะใช้การวนลูปรับค่าจำนวน 8 ไบต์ มาเก็บในบัฟเฟอร์ตั้งแต่แอดเดรส 051H ถึง 058H โดยแผนภาพการทำงานแสดงดังรูปที่ 3.1

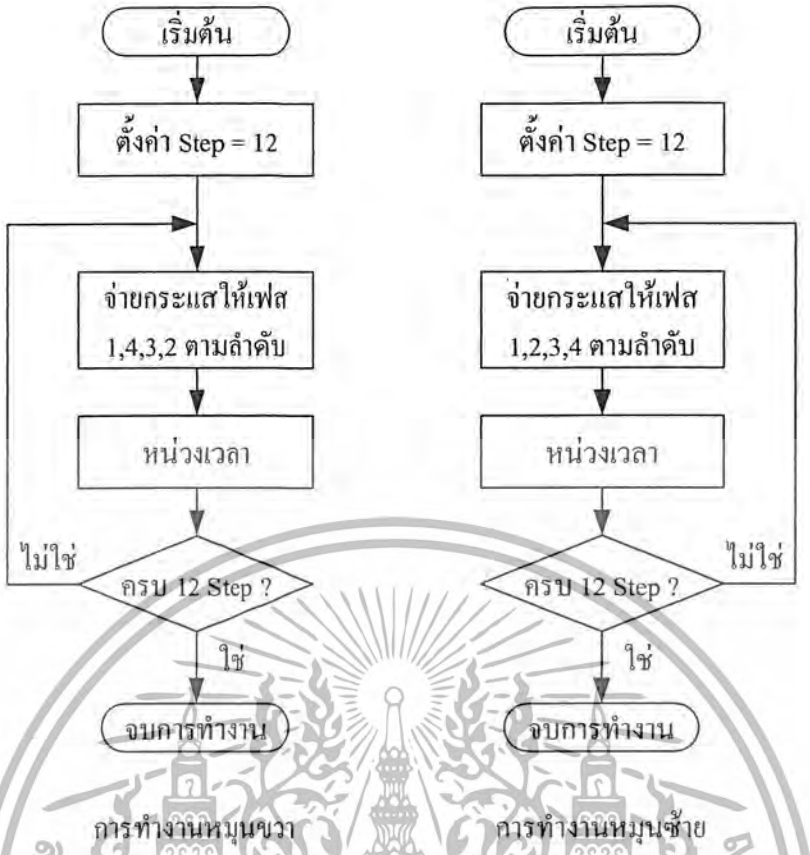
รูปที่ 3.1 แผนภาพการทำงานส่วนที่ทำการรับค่ารหัสจากเครื่องอ่านบาร์โค้ด

การทำงานของโปรแกรมคือ เริ่มต้นจะทำการรอรับค่าจากเครื่องอ่านบาร์โค้ด ถ้ายังไม่มีการรูดบัตรนักศึกษาผ่าน ก็ยังไม่ทำงาน แต่เมื่อมีบัตรนักศึกษาผ่านเครื่องอ่านบาร์โค้ด เครื่องจะถอดรหัสในบัตรให้เป็นรหัส ASCII 8 ไบต์ ไมโครคอนโทรลเลอร์ก็จะรับค่าทีละ 1 ไบต์เก็บลงในบัฟเฟอร์ และตรวจสอบว่าข้อมูลที่รับมาครบ 8 ไบต์รึยัง ถ้ายังก็จะกลับไปรับข้อมูลไบต์ต่อไปจนครบทั้ง 8 ไบต์ ก็จะจบการทำงาน

3.2 Stepping Motor

ส่วนนี้จะเป็นส่วนที่ควบคุมการขับกลอน เพื่อให้ประตูมีสภาวะล็อค หรือ ไม่ล็อค โดยลักษณะการขับกลอนจะขับเป็นมุม 90 องศาโดนการสั่งให้มอเตอร์หมุนขวา 90 องศา เพื่อให้ประตูอยู่ในสภาวะล็อค และสั่งให้มอเตอร์หมุนทางซ้าย 90 องศา เพื่อให้ประตูอยู่ในสภาวะไม่ล็อค โดยในการออกแบบนั้น มอเตอร์ที่นำมาใช้งานมีมุมสเตป สเตปละ 7.5 องศา ดังนั้นต้องสั่งให้มอเตอร์หมุน 12 สเตป เพื่อให้ได้มุม 90 องศา โดยมีแผนภาพการทำงานดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แผนภาพการทำงานของ Stepping Motor

การทำงานของโปรแกรมนี้คือ เริ่มต้นโปรแกรมจะตั้งค่าในการหมุนสเตปไว้ที่ 12 สเตปเพื่อให้ได้มุมในการหมุน 90 องศา ต่อมาก็จะตรวจสอบการสั่งงานจากไมโครคอนโทรลเลอร์ ในกรณีที่มีการสั่งงานมา ก็จะทำการตรวจสอบว่าต้องการหมุนซ้ายหรือหมุนขวา

การทำงานหมุนซ้าย ใช้เมื่อระบบต้องการสั่งให้คลายล็อกประตู หรือเปิดประตู โปรแกรมจะสั่งให้มีการจ่ายกระแสให้แต่ละเฟสของมอเตอร์ โดยเรียงจากเฟส 1, 2, 3 และ 4 ตามลำดับ เมื่อครบทั้ง 4 เฟส ก็จะตรวจสอบจำนวนครั้งของสเตปว่าครบ 12 สเตปหรือไม่ ถ้าไม่ครบก็จะกลับไปสั่งให้จ่ายกระแสให้เฟสต่างๆ ของมอเตอร์ จนค่าสเตปครบ 12 สเตป

การทำงานหมุนขวา ใช้เมื่อระบบต้องการจะสั่งให้ล็อกประตูภายหลังการปิด โปรแกรมจะสั่งให้มีการจ่ายกระแสให้แต่ละเฟสของมอเตอร์ โดยเรียงจากเฟส 1, 4, 3 และ 2 ตามลำดับ เมื่อครบทั้ง 4 เฟส ก็จะตรวจสอบจำนวนครั้งของสเตปว่าครบ 12 สเตปหรือไม่ ถ้าไม่ครบก็จะกลับไปสั่งให้จ่ายกระแสให้เฟสต่างๆ ของมอเตอร์ จนค่าสเตปครบ 12 สเตป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 วงจร Sensor

เป็นส่วนที่ทำหน้าที่ ในการตรวจสถานะของประตู ภายหลังจากผู้ที่ได้รับอนุญาตให้เข้าใช้งานห้องเข้าไปในห้อง ส่วนนี้จะทำการตรวจการปิดประตู โดยถ้ามีการปิดประตู Sensor จะส่งข้อมูลไปบอกไมโครคอนโทรลเลอร์ เพื่อที่ไมโครคอนโทรลเลอร์จะได้สั่งให้ล็อคประตู

ภาคส่งสัญญาณ

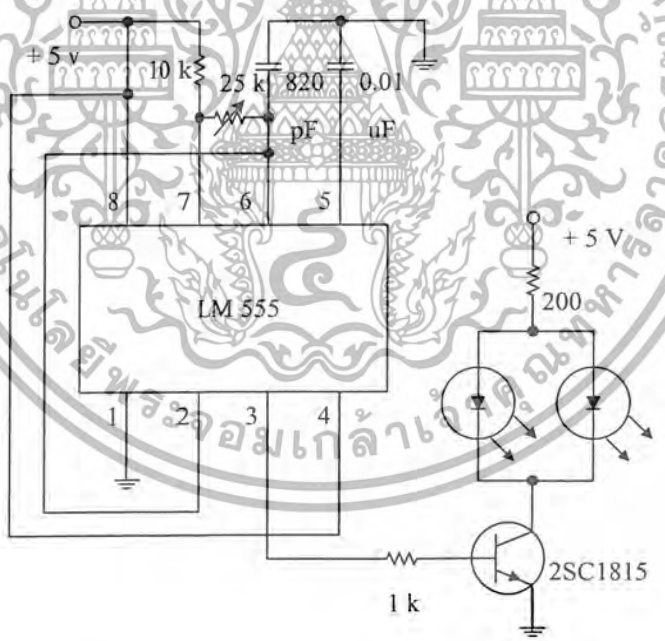
จากลักษณะการทำงานของวงจรภาคส่งสัญญาณอินฟราเรด พบว่า ความกว้างของพัลส์ที่ได้ทางด้านเอาท์พุต จะขึ้นกับการเก็บประจุและคายประจุของ C_a จาก Datasheet จะได้ว่า

ช่วงการเก็บประจุ : $t_1 = 0.693C_a(R_a + R_b)$

ช่วงการคายประจุ : $t_2 = 0.693C_aR_a$

ความถี่ของสัญญาณเอาท์พุต : $f = 1/(t_1 + t_2)$

วงจรอินฟราเรดทางด้านส่ง จะสร้างพัลส์โดยใช้ไอซี LM555 โดยความถี่ที่สร้างกำหนดได้จากค่า R_a, R_b และ C_a โดยต้องการสร้างพัลส์ความถี่ 38 kHz สัญญาณพัลส์ที่สร้างได้ถูกส่งไปยังวงจรไดรว์ โดยใช้เพาเวอร์ทรานซิสเตอร์เบอร์ 2SC1815

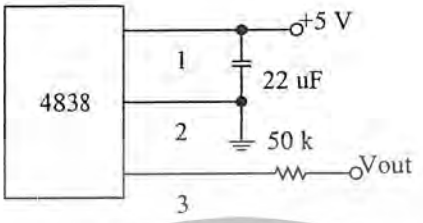


รูปที่ 3.3 วงจรภาคส่งสัญญาณอินฟราเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาครับสัญญาณ

วงจรทางด้านภาครับสัญญาณอินฟราเรด จะใช้ Photo Diode เบอร์ 4838 ซึ่งสามารถรับสัญญาณอินฟราเรดที่ความถี่ 38 kHz ซึ่งภายใน Photo Diode เบอร์ 4838 จะประกอบด้วยส่วนรับสัญญาณอินฟราเรดและภาคขยายสัญญาณที่เอาท์พุต



รูปที่ 3.4 วงจรภาครับสัญญาณอินฟราเรด

เมื่อนำส่วนของวงจร Sensor สำหรับตรวจสอบสถานะประตูไปใช้ในไมโครคอนโทรลเลอร์ โดยการต่อขา Vout เข้าที่ P2.7 โดยสถานะประตูถ้าเปิดอยู่ วงจรรับสัญญาณจะให้เอาท์พุตเป็น 5 โวลต์ แต่ถ้าสถานะประตูปิดอยู่ จะให้สัญญาณเอาท์พุตเป็น 0 โวลต์ จึงใช้การเปลี่ยนลอจิกของขา P2.7 เป็นตัวตัดสินสถานะของประตู ตามแผนภาพ รูปที่ 3.5

รูปที่ 3.5 แผนภาพการทำงานของ Sensor สำหรับตรวจสอบสถานะประตู

การทำงานคือ เมื่อผู้เข้าใช้ระบบได้รับอนุญาตให้เข้าใช้งานสถานะประตูจะเป็นสถานะเปิดทำให้ P2.7 เป็นลอจิก "1" โปรแกรมก็จะตรวจสอบสถานะจนกระทั่ง P2.7 เปลี่ยนเป็นลอจิก "0" แสดงว่าประตูอยู่ในสถานะปิดแล้ว ก็จะเรียกโปรแกรมหมุนมอเตอร์ทางขวา 90 องศา เพื่อล็อกประตู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ไอซีสร้างฐานเวลาจริงหรือรีลไทม์คล็อก (RTC)

ทำหน้าที่ให้ค่าวันเดือนปี และค่าเวลาปัจจุบัน เพื่อนำค่าเวลานี้ไปบันทึกในกรณีที่มีผู้เข้าใช้งานโดย การทำงานของไอซีสร้างฐานเวลาจริง จะใช้หลักการของ บัส I^2C โดยระบบบัส I^2C จะมีโหมดการทำงาน 2 โหมด คือ โหมดเขียนข้อมูล และโหมดอ่านข้อมูล มีการทำงานตามแผนภาพในรูปที่ 3.6

รูปที่ 3.6 การอ่านและเขียนข้อมูลระบบบัส I^2C

โดยการใช้กับ RTC นั้น โหมดการเขียนข้อมูล จะใช้เมื่อต้องการตั้งค่าวันและเวลาปัจจุบันให้กับ RTC ส่วนโหมดการอ่านข้อมูลนั้น ใช้สำหรับการดึงค่าเวลาจาก RTC มาบันทึกในกรณีที่มีการเข้าใช้งานต้องมีการกำหนดคอนโทรลไบต์เพื่อบอกว่าเป็นการเขียนหรือการอ่านข้อมูลมีการกำหนดดังนี้

การอ่านข้อมูล : คอนโทรลไบต์ คือ 11010001B

การเขียนข้อมูล : คอนโทรลไบต์ คือ 11010000B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RTC จะแอดเดรสที่เก็บข้อมูลต่างๆ ที่แน่นอน และเราจะทำการกำหนดแอดเดรสสำหรับเก็บข้อมูลต่างๆ ซึ่งจะแสดงได้ตามตารางที่ 3.2

ข้อมูล	แอสเดรสของ RTC (DS1307)	แอดเดรสที่ใช้เก็บข้อมูล
วินาที	00H	41H
นาที	01H	42H
ชั่วโมง	02H	43H
วัน	03H	44H
วันที่	04H	45H
เดือน	05H	46H
ปี	06H	47H

ตารางที่ 3.2 แอดเดรสสำหรับเก็บข้อมูลต่างๆ

3.5 การติดต่อรอม

รอมที่นำมาใช้งาน จะใช้รอม 2 ตัว เป็นชนิด EEPROM เพื่อให้สามารถเพิ่มหรือลบข้อมูลได้โดย รอมตัวที่ 1 จะไว้ใช้สำหรับเก็บค่ารหัสของผู้ที่สามารถเข้าใช้งานได้ และรหัสผ่านของแต่ละคน ส่วนรอมตัวที่ 2 จะไว้ใช้สำหรับเก็บค่ารหัส และวันเวลาเข้าใช้งานของผู้เข้าใช้งาน โดยการติดต่อกับรอม จะแบ่งออกเป็น 2 โหมด คือโหมดการอ่านข้อมูล และโหมดการเขียนข้อมูล การติดต่อจะใช้บัส I^2C

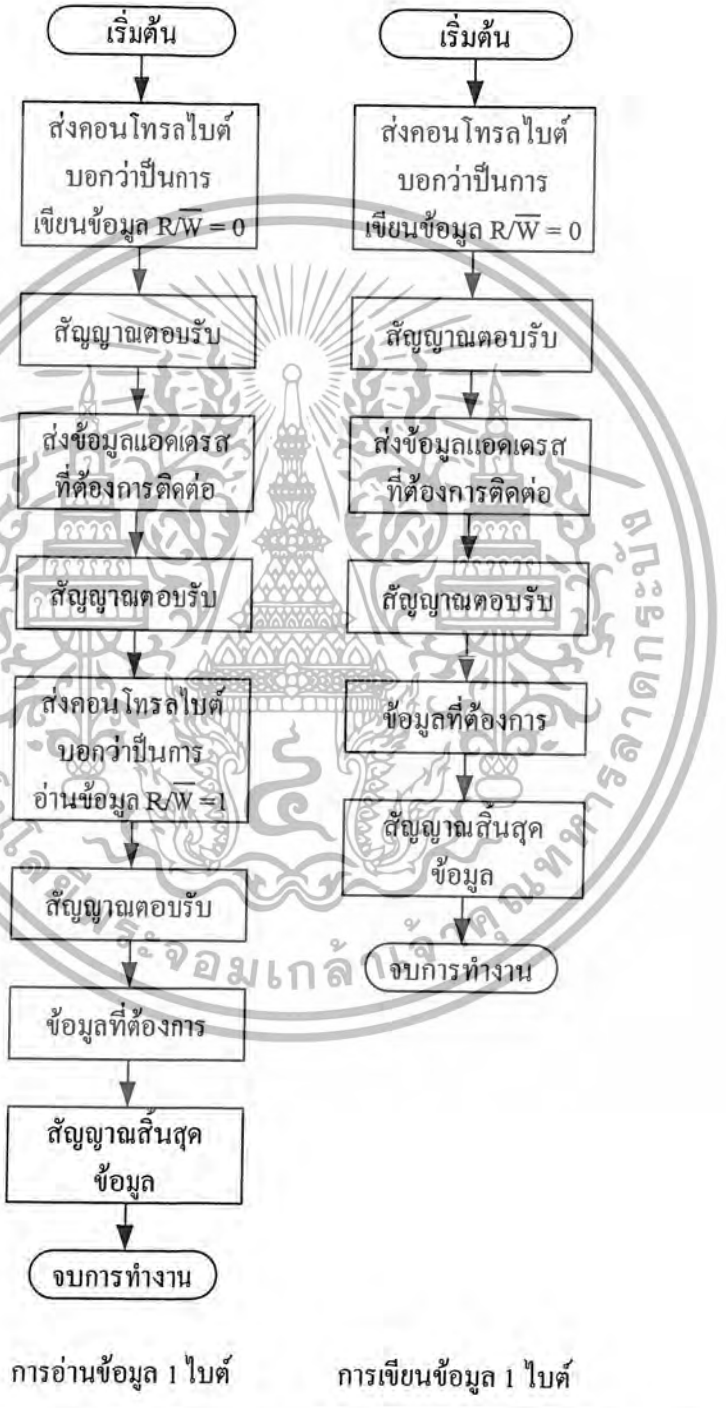
การติดต่อรอมทั้ง 2 ตัว จะใช้คอนโทรลไบต์เป็นตัวระบุว่าจะติดต่อรอมตัวใด และเป็นการติดต่อเพื่อเขียนหรืออ่านข้อมูล โดยกำหนดคอนโทรลไบต์ในการติดต่อตามตารางที่ 3.3

รอมที่ต้องการติดต่อ	กระบวนการติดต่อ	คอนโทรลไบต์
ตัวที่ 1	อ่านข้อมูล	10100001B
ตัวที่ 1	เขียนข้อมูล	10100000B
ตัวที่ 2	อ่านข้อมูล	10100011B
ตัวที่ 2	เขียนข้อมูล	10100010B

ตารางที่ 3.3 การกำหนดคอนโทรลไบต์สำหรับติดต่อรอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับโหมดการเขียนข้อมูลของรอมนั้นจะเป็นไปตามแผนภาพการเขียนข้อมูลในรูปที่ 3.7 โดยการเขียนจะเหมือนกระบวนการเขียนระบบ I^2C แต่การอ่านข้อมูลจะต่างกัน ที่ การอ่านข้อมูล 1ไบต์ จะต้องทำการส่งคอนโทรลไบต์บอกว่าจะเขียนข้อมูลและรอ ACK จากนั้นส่งแอดเดรสที่ต้องการอ่านและรอ ACK แล้วจึงส่งคอนโทรลไบต์เพื่อบอกว่าจะอ่านและรอ ACK กลับมาพร้อมกับรับข้อมูลที่ต้องการอ่านและสัญญาณสิ้นสุดข้อมูลจะถูกส่งมาหลังจากข้อมูล

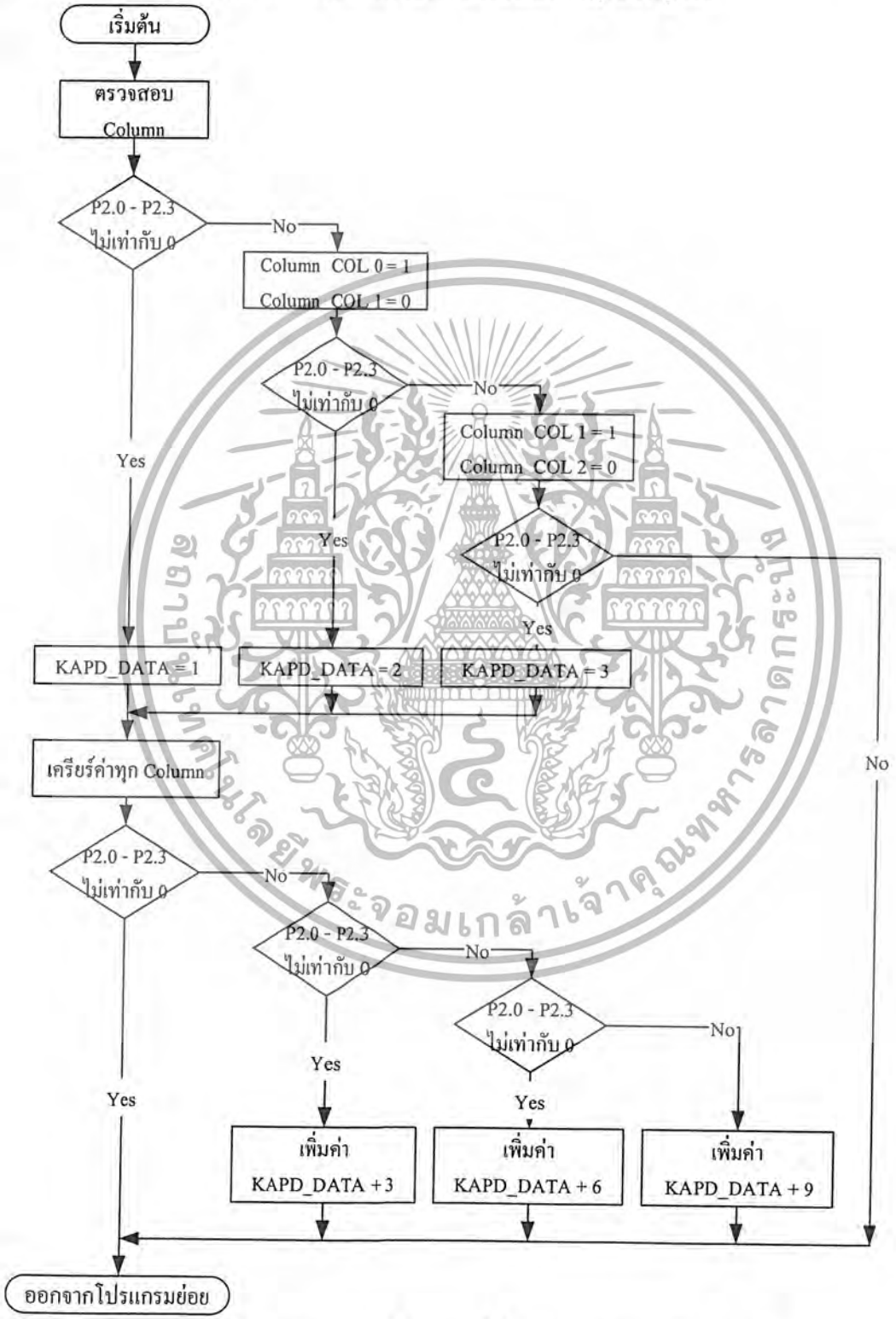


รูปที่ 3.7 แสดงการติดต่อกับรอมโดยการอ่านข้อมูลและเขียนข้อมูล 1 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 คีย์แพด

ใช้สำหรับป้อนค่ารหัสผ่าน เพื่อเข้าใช้งานห้อง หรือใช้สำหรับเพิ่มฐานข้อมูล ของผู้เข้าใช้งานโดยคีย์แพดจะมีลักษณะเป็นสวิตช์แบบเมตริก แบ่งเป็นแถวและหลัก โดยจะต่อตามภาพที่ 2.12 ซึ่งการทำงานจะแสดงได้ตามแผนภาพในรูปที่ 3.8 กำหนดแอดเดรส 037H สำหรับเก็บข้อมูลคีย์ที่ถูกกด



รูปที่ 3.8 การทำงานของโปรแกรมที่ใช้ตรวจสอบการกดปุ่มคีย์แพด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานคือ จะมีการตรวจสอบที่หลักแรก ก่อน ถ้ามีการกด ค่าจะเป็น 0 ก็จะไปตรวจสอบแถวเพื่อดูว่าเป็นสวิตช์เลขใดโดยดูจากแถวที่มีค่าเป็น 0 ก็จะสามารถบอกได้ว่าปุ่มใดถูกกด แต่ถ้าตรวจสอบแล้วหลักแรกเป็น 1 ก็จะทำการตรวจหลักต่อๆไปได้แก่ หลักที่ 2 และ หลักที่ 3 เพื่อดูว่ามีการกดหรือไม่ เมื่อได้หลักก็นำไปหาแถวต่อเพื่อที่จะทราบว่าปุ่มใดถูกกด

3.7 การส่งข้อมูลไปยังเครื่องคอมพิวเตอร์ผ่านพอร์ตอนุกรม

ทำหน้าส่งค่ารหัสและวันเวลาเข้าใช้งานของแต่ละ USER ที่เก็บอยู่ในรอมไปเก็บไว้ที่คอมพิวเตอร์โดยผ่านทางพอร์ตอนุกรม โดยจะทำการอ่านค่าจากรอมขึ้นมาเก็บลงในบัฟเฟอร์ที่ได้กำหนดไว้ จากนั้นก็ย้ายค่าข้อมูลที่ต้องการส่งไปไว้ที่ SBUF (เป็น BUFFER ของการส่งข้อมูลอนุกรม) แล้วทำการส่งค่าออกไป วนลูปจนกว่าจะส่งข้อมูลครบทั้งหมด (ประกอบด้วย รหัสนักศึกษา 8 ไบต์ วันเดือนปี 6 ไบต์ และเวลา 6 ไบต์)

โครงสร้างทางด้านฮาร์ดแวร์ของการเชื่อมต่อ ไมโครคอนโทรลเลอร์ กับ Computer ผ่านทางพอร์ตอนุกรมจะต้องทำการเชื่อมต่อขาสัญญาณ Tx และ Rx ของ ไมโครคอนโทรลเลอร์ เข้ากับ IC MAX232 เพื่อที่จะแปลงสัญญาณจาก ไมโครคอนโทรลเลอร์ ให้กลายเป็นมาตรฐานของ RS232 จึงจะสามารถเชื่อมต่อใช้งานกันได้ในทางกลับกัน MAX232 ยังทำหน้าที่แปลงระดับสัญญาณที่ส่งจาก คอมพิวเตอร์ มาตรฐานของ RS232 เป็นสัญญาณที่ใช้ในไมโครคอนโทรลเลอร์ โดยการทำงานเป็นดังรูปที่ 3.9

รูปที่ 3.9 แผนผังการส่งข้อมูลไปยังเครื่องคอมพิวเตอร์ผ่านพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานคือ ข้อมูลที่เราจะส่ง 1 ครั้ง จะประกอบด้วย รหัสนักศึกษา 8 ไบต์ วันเดือนปี 6 ไบต์ และเวลา 6 ไบต์ ทั้งหมดคือ 20 ไบต์ ดังนั้น เราจะตั้งค่าให้มีการวนลูปเพื่อส่งข้อมูลทั้งหมด 20 ครั้งโดยเริ่มจากการดึงข้อมูลไบต์แรกจากรอมขึ้นมาจากบัฟเฟอร์ จากนั้นก็ส่งไปยังเครื่องคอมพิวเตอร์ และทำการตรวจสอบว่าข้อมูลส่งไปครบ 20 ไบต์รึยัง ถ้ายังก็จะไปดึงข้อมูลไบต์ต่อไปจากรอมเพื่อทำการส่ง แต่ถ้าครบ 20 ไบต์ก็จะสิ้นสุดการทำงาน สำหรับการส่งข้อมูล 1 ชุด

ส่วนที่ติดต่อกับคอมพิวเตอร์

3.8 การรับข้อมูลจากไมโครคอนโทรลเลอร์ ผ่านทางพอร์ตอนุกรม

ทำหน้าที่รับข้อมูลรหัสนักศึกษาและวันเวลาเข้าใช้งาน จากไมโครคอนโทรลเลอร์ มาเพื่อทำการประมวลผล ซึ่งการออกแบบส่วนการรับข้อมูลนี้ จำเป็นต้องทราบเกี่ยวกับ รูปแบบของข้อมูลและลักษณะของข้อมูลซึ่งจากการออกแบบทางด้านไมโครคอนโทรลเลอร์ ทำให้ทราบรูปแบบและลักษณะของข้อมูล ดังนี้

1) อัตราการส่งข้อมูล 9600 บิตต่อวินาที
2) Data Bit 8 bit มี 1 Stop Bit
3) ไม่มีการตรวจสอบพาริตี้ (Non-parity Checking)
4) ข้อมูลจะเป็นรหัส ASCII จำนวน 10 อักขระ

การติดต่อกับ ไมโครคอนโทรลเลอร์ เราจะใช้การเขียนโปรแกรมรับค่าโดยใช้ Microsoft Visual Basic โดยใน Visual Basic จะมีฟังก์ชันที่ใช้ติดต่อกับ Serial Port อยู่แล้ว ชื่อ MSComm โดยเขียนโปรแกรมเพื่อกำหนดค่าต่างๆตามเงื่อนไขดังนี้

1) กำหนดว่าจะต่ออุปกรณ์เข้ากับพอร์ตไหนของคอมพิวเตอร์ โดยกำหนดดังนี้
   MSComm1.CommPort = 1 (หมายถึง กำหนดว่าเชื่อมต่อกับ COM Port 1)
2) กำหนดค่าการรับส่งข้อมูล ในที่นี้คือ อัตราเร็ว 9600 bps, เป็น Data 8 Bit, ไม่มีการตรวจสอบพาริตี้ , มี 1 Stop Bit สามารถเขียนได้เป็น
   MSComm1.Settings = "9600,N,8,1"
3) กำหนดชนิดของข้อมูลที่จะรับเข้ามา เราทราบว่าเป็นรหัส ASCII ดังนั้นจะกำหนดค่าดังนี้
   MSComm1.InputMode = ComInputModeText
4) กำหนดให้มีการเปิดพอร์ตใช้งาน กำหนดดังนี้
   MSComm1.PortOpen = True
5) การรับค่าข้อมูลทำได้โดยการใช้คำสั่งรับค่ามาเก็บในบัฟเฟอร์ โดยกำหนดดังนี้
   Buffer (เป็นตัวแปรที่กำหนดเอง) = MSComm1.Input
6) กำหนดขนาดของข้อมูลที่จะดึงขึ้นมาใช้งานจากในบัฟเฟอร์ ในที่นี้ขนาดข้อมูลจะประกอบด้วยคือ ค่ารหัส 8 ไบต์, ค่าวันเดือนปี 6 ไบต์ และ ค่าของเวลาอีก 6 ไบต์ ดังนั้น
   ขนาดของข้อมูลต้องมีขนาด 20 ไบต์
   MSComm1.InputLen = 20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยแผนผังการทำงานของการรับข้อมูลจากไมโครคอนโทรลเลอร์ ผ่านทางพอร์ตอนุกรม แสดงได้ดังรูปที่ 3.10

รูปที่ 3.10 การทำงานของการรับข้อมูลจากไมโครคอนโทรลเลอร์ ผ่านทางพอร์ตอนุกรม

การทำงานคือ จะทำการตรวจสอบว่า มีข้อมูลเข้ามาทางพอร์ตอนุกรมหรือไม่ ถ้าไม่มีก็จะตรวจสอบใหม่ จนกระทั่งมีข้อมูลเข้ามาทางพอร์ตอนุกรม โดยจะส่งจากไมโครคอนโทรลเลอร์ เป็นข้อมูลชุดละ 20 ไบต์ จะถูกเก็บไว้ในบัฟเฟอร์รับข้อมูล จากนั้นการนำข้อมูลที่รับมาใช้งาน จะต้องแบ่งข้อมูลออกเป็นชุดๆ โดยจะทำงานทีละชุดข้อมูล โดยกำหนดข้อมูลความยาว 20 ไบต์ จะทำให้ได้ข้อมูลเป็นชุดย่อยๆ และข้อมูล 20 ไบต์นี้ จะนำมาแยกเป็นข้อมูล 3 อย่าง ได้แก่ รหัสนักศึกษา, วันเดือนปี และเวลาที่เข้าใช้งาน หลังจากนั้นจะตรวจสอบข้อมูลในบัฟเฟอร์ว่าหมดรึยัง ถ้ายังไม่หมดจะไปตัดข้อมูล 20 ไบต์ ถัดไป เพื่อแบ่งข้อมูลตามขั้นตอนไป จนกระทั่งข้อมูลในบัฟเฟอร์หมด ก็จะจบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 การติดต่อกับฐานข้อมูลในคอมพิวเตอร์

ส่วนที่ติดต่อกับฐานข้อมูลเราต้องทำการสร้างฐานข้อมูลโดยใช้ Microsoft Access ในการสร้างโดยในโครงงานนี้จะมีฐานข้อมูล 2 ชุดด้วยกัน คือ

1) ฐานข้อมูลที่เก็บข้อมูลนักศึกษาที่ได้รับอนุญาตให้เข้าใช้ห้อง ประกอบด้วย ชื่อ-นามสกุล ของนักศึกษา, รหัสประจำตัว, ภาควิชา
2) ฐานข้อมูลที่เก็บข้อมูลนักศึกษาที่เข้าใช้ห้อง ประกอบไปด้วย ชื่อ-นามสกุล ของนักศึกษา, รหัสประจำตัว, ภาควิชา, เวลา และวันที่ที่เข้าใช้

โดยการทำงานในส่วนนี้แสดงดังแผนภาพในรูปที่ 3.11

รูปที่ 3.11 แสดงแผนภาพการทำงานส่วนที่เชื่อมต่อกับ Computer

การออกแบบโปรแกรมในส่วนนี้ ให้พิจารณาขั้นตอนการทำงาน ตามแผนภาพในรูปที่ 3.9 คือ เมื่อรับข้อมูลที่ส่งมาจากไมโครคอนโทรลเลอร์มาแล้ว ก็จะนำค่าที่รับได้มาแยกเป็น 3 ส่วนคือ ส่วนของวันเดือนปี ส่วนของเวลา และส่วนของรหัสนักศึกษา โดยเราจะนำส่วนรหัสนักศึกษา ไปเปรียบเทียบกับฐานข้อมูลนักศึกษาที่ได้รับอนุญาตให้เข้าใช้ห้องโดยจะทำการค้นหาจากรหัสนักศึกษาว่ามีในฐานข้อมูลหรือไม่ ถ้าไม่มีในฐานข้อมูลก็จะไม่ทำงานต่อ แต่ถ้ามีอยู่ในฐานข้อมูลก็จะดึงข้อมูลนักศึกษาออกมาแสดงที่หน้าจอ ในส่วนของฐานข้อมูลนักศึกษา ที่ได้รับอนุญาตให้เข้าใช้ห้องต้องออกแบบให้ สามารถเพิ่มข้อมูลนักศึกษาเข้าไปภายหลังได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

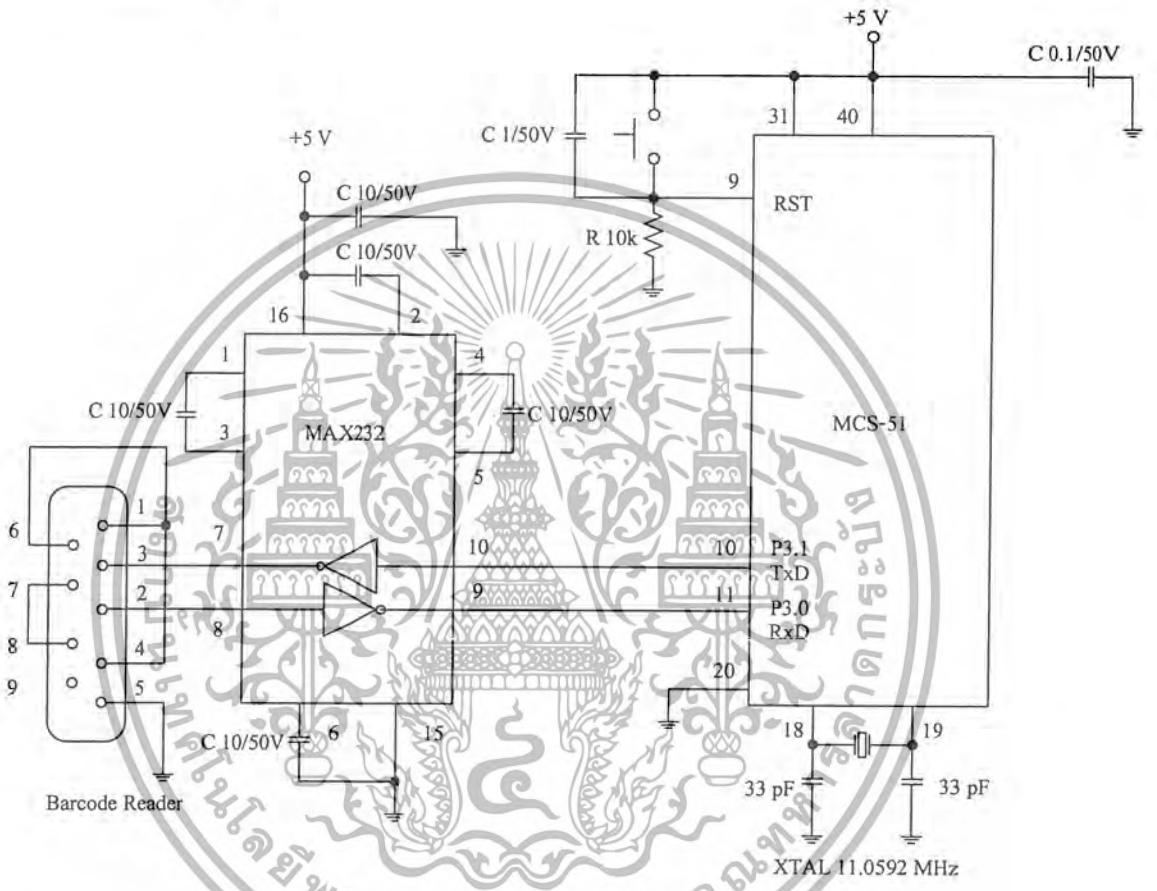
บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองวัดสัญญาณจากเครื่องอ่านบาร์โค้ด

ขั้นตอนการทดลอง

1. ต่อวงจรตามรูปที่ 4.1



รูปที่ 4.1 แสดงวงจรที่ใช้ทดลองวัดค่าสัญญาณจากเครื่องอ่านบาร์โค้ด

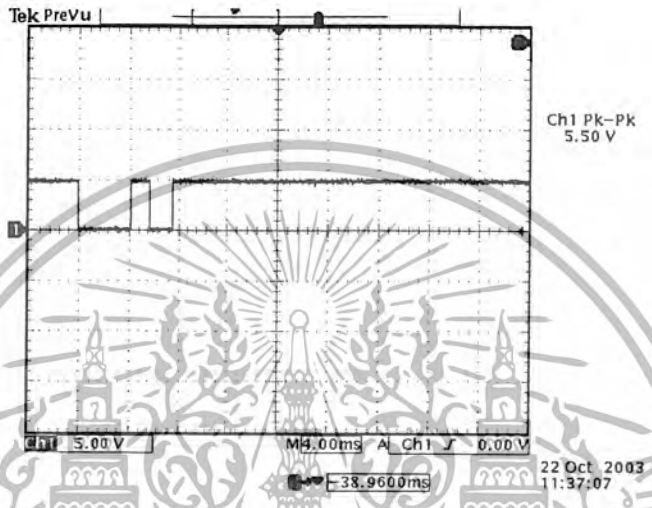
2. ทำการทดลองโดยการรูดบัตรรหัส 0 – 9 เพื่อตรวจสอบดูสัญญาณจากเครื่องอ่านบาร์โค้ด ว่าถูกต้องตามที่ระบุของเครื่องอ่านบาร์โค้ดหรือไม่ โดยทำการวัดสัญญาณที่ขา 11 ของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

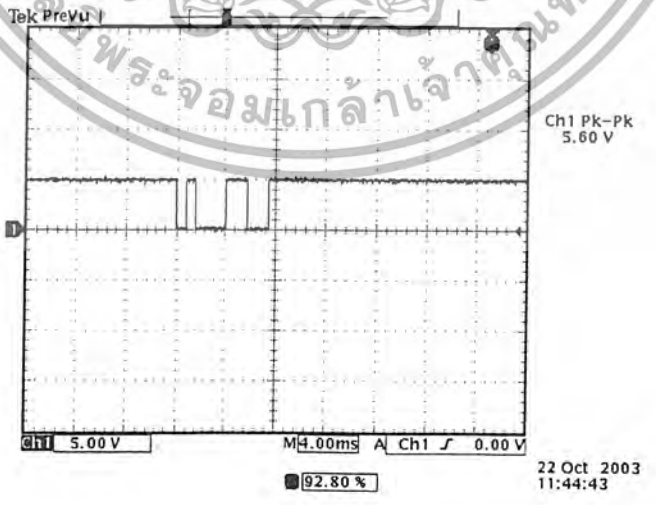
เมื่อทำการรูดรหัสเลข 0 – 9 แล้วทำการวัดสัญญาณพบว่า สัญญาณที่วัดได้จากขา 11 ของไมโครคอนโทรลเลอร์ ได้ผลการทดลองดังนี้

สัญญาณ 0 : รูปสัญญาณจะเป็นสัญญาณข้อมูล 8 บิต คือ 00001100B อ่านรหัสจากหลัง จะได้เป็น 30H ซึ่งตรงกับรหัส ASCII ของเลข 0



รูปที่ 4.2 แสดงสัญญาณเมื่อทำการรูดรหัส 0

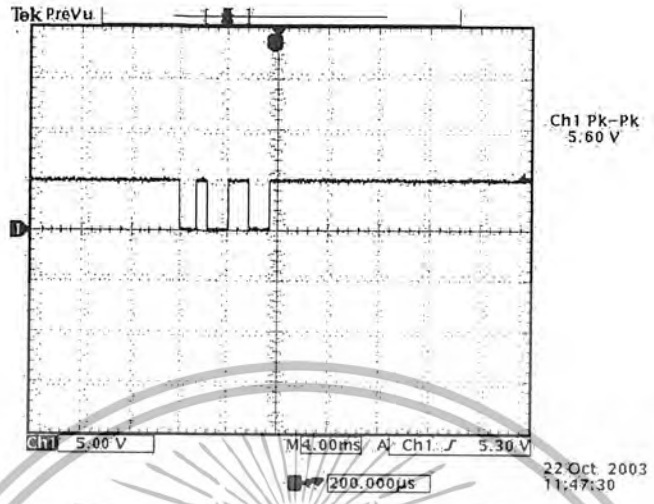
สัญญาณ 1 : รูปสัญญาณจะเป็นสัญญาณข้อมูล 8 บิต คือ 10001100B อ่านรหัสจากหลัง จะได้เป็น 31H ซึ่งตรงกับรหัส ASCII ของเลข 1



รูปที่ 4.3 แสดงสัญญาณเมื่อทำการรูดรหัส 1

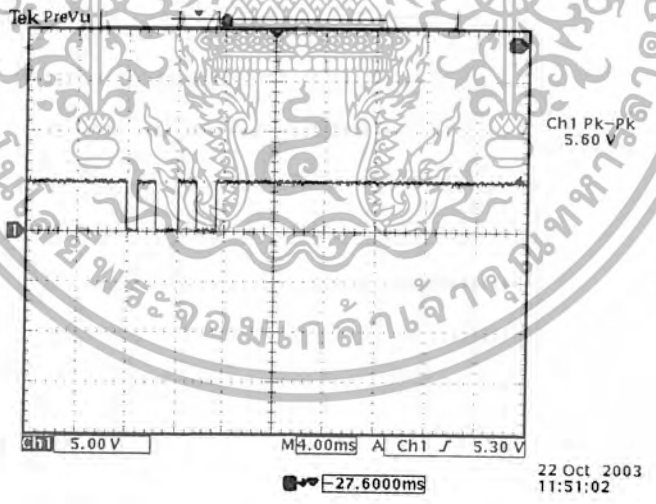
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ 2 : รูปสัญญาณจะเป็นสัญญาณข้อมูล 8 บิต คือ 01001100B อ่านรหัสจากหลัง จะได้เป็น 32H ซึ่งตรงกับรหัส ASCII ของเลข 2



รูปที่ 4.4 แสดงสัญญาณเมื่อทำการรูดรหัส 2

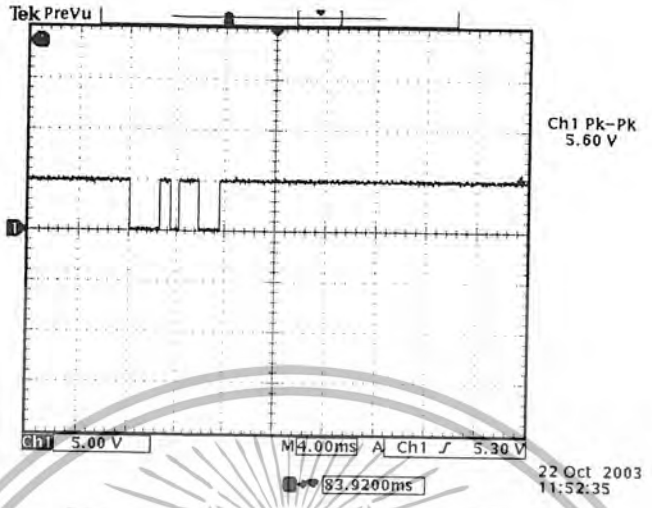
สัญญาณ 3 : รูปสัญญาณจะเป็นสัญญาณข้อมูล 8 บิต คือ 11001100B อ่านรหัสจากหลัง จะได้เป็น 33H ซึ่งตรงกับรหัส ASCII ของเลข 3



รูปที่ 4.5 แสดงสัญญาณเมื่อทำการรูดรหัส 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ 4 : รูปสัญญาณจะเป็นสัญญาณข้อมูล 8 บิต คือ 00101100B อ่านรหัสจากหลัง จะได้เป็น 34H ซึ่งตรงกับรหัส ASCII ของเลข 4



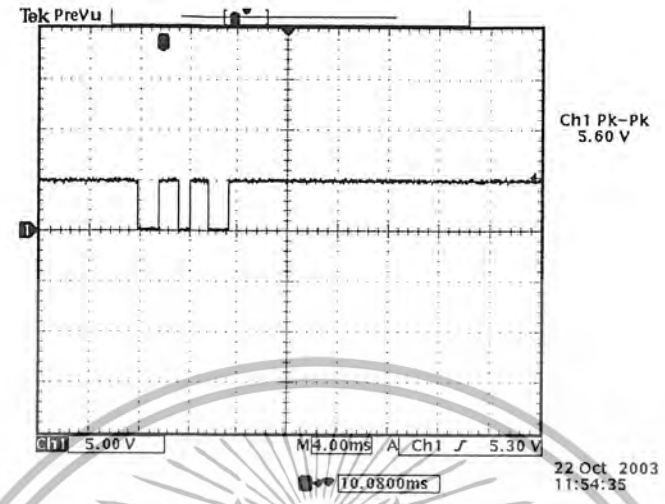
รูปที่ 4.6 แสดงสัญญาณเมื่อทำการรูดรหัส 4

สัญญาณ 5 : รูปสัญญาณจะเป็นสัญญาณข้อมูล 8 บิต คือ 10101100B อ่านรหัสจากหลัง จะได้เป็น 35H ซึ่งตรงกับรหัส ASCII ของเลข 5

รูปที่ 4.7 แสดงสัญญาณเมื่อทำการรูดรหัส 5

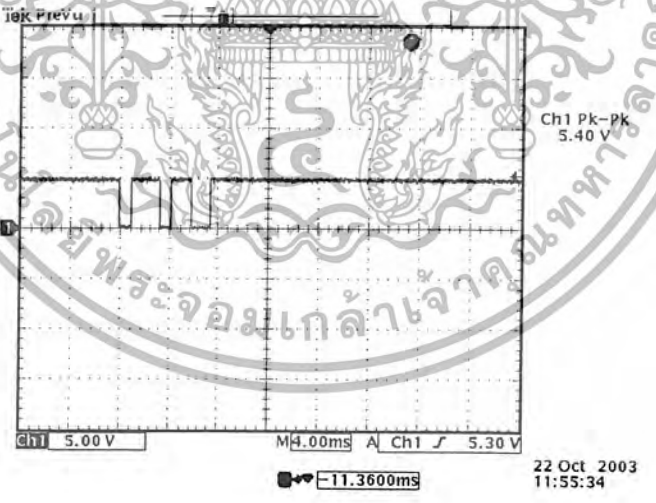
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ 6 : รูปสัญญาณจะเป็นสัญญาณข้อมูล 8 บิต คือ 01101100B อ่านรหัสจากหลัง จะได้เป็น 36H ซึ่งตรงกับรหัส ASCII ของเลข 6



รูปที่ 4.8 แสดงสัญญาณเมื่อทำการรูดรหัส 6

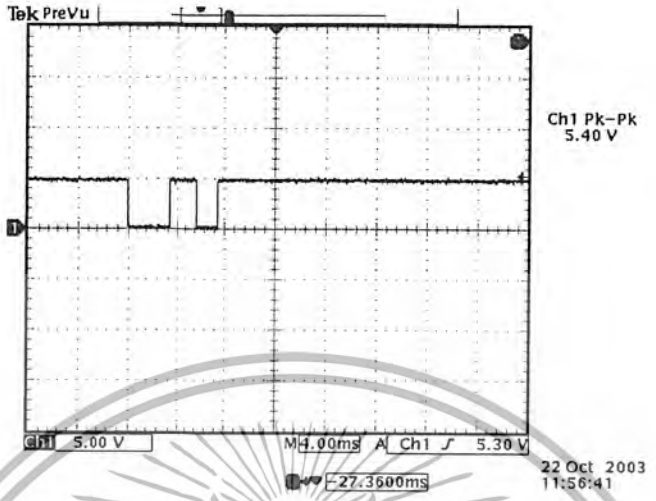
สัญญาณ 7 : รูปสัญญาณจะเป็นสัญญาณข้อมูล 8 บิต คือ 11101100B อ่านรหัสจากหลัง จะได้เป็น 37H ซึ่งตรงกับรหัส ASCII ของเลข 7



รูปที่ 4.9 แสดงสัญญาณเมื่อทำการรูดรหัส 7

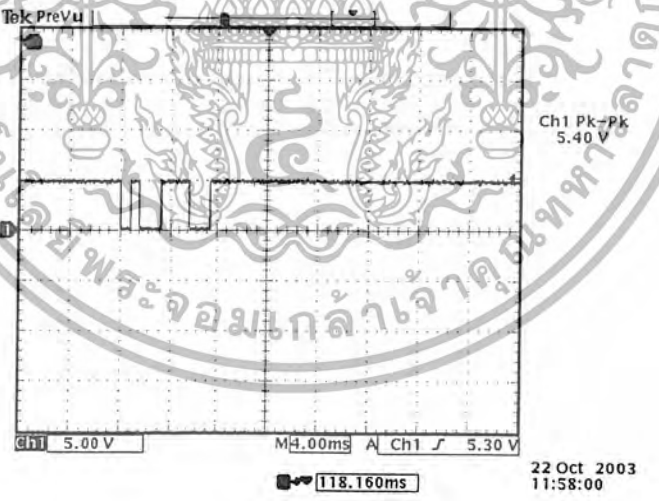
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ 8 : รูปสัญญาณจะเป็นสัญญาณข้อมูล 8 บิต คือ 00011100B อ่านรหัสจากหลัง จะได้เป็น 38H ซึ่งตรงกับรหัส ASCII ของเลข 8



รูปที่ 4.10 แสดงสัญญาณเมื่อทำการรูดรหัส 8

สัญญาณ 9 : รูปสัญญาณจะเป็นสัญญาณข้อมูล 8 บิต คือ 10011100B อ่านรหัสจากหลัง จะได้เป็น 39H ซึ่งตรงกับรหัส ASCII ของเลข 9



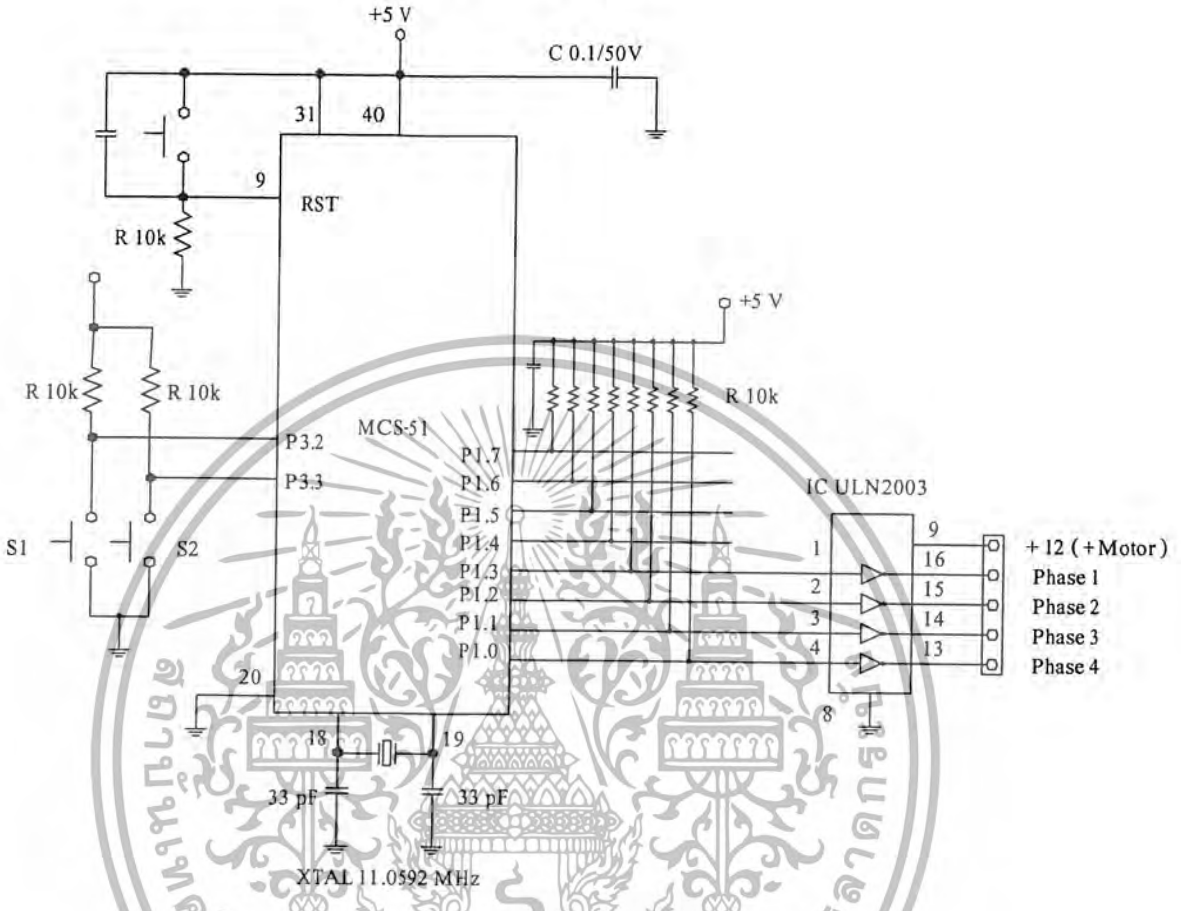
รูปที่ 4.11 แสดงสัญญาณเมื่อทำการรูดรหัส 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองขับ Stepping motor

ขั้นตอนการทดลอง

1. ต่อวงจรที่ใช้ทดลองการขับ Stepping motor ตามรูปที่ 4.12



รูปที่ 4.12 แสดงวงจรที่ใช้ในการทดลองการขับ Stepping motor

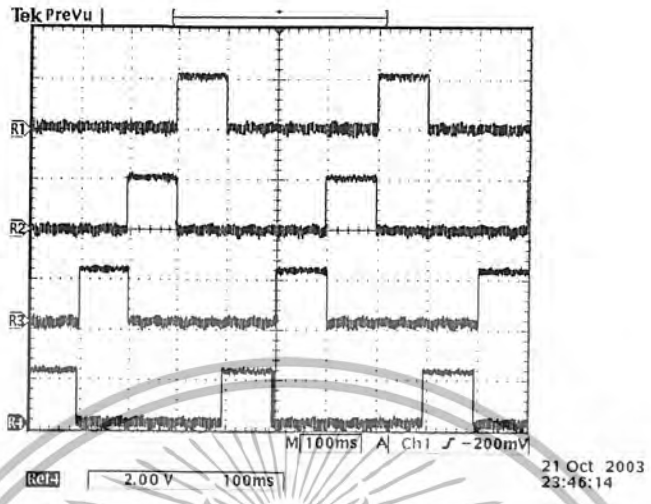
2. ทำการกดสวิตช์ S1 เพื่อ Stepping motor หมุนทางขวา พร้อมทำการวัดรูปสัญญาณที่ขั้วที่ 1 , 2 , 3 และ 4 โดยทำการเก็บรูปสัญญาณเปรียบเทียบกันทั้ง 4 ขั้ว

3. จากนั้นทำการกดสวิตช์ S2 เพื่อให้ Stepping motor หมุนทางซ้าย พร้อมทำการวัดรูปสัญญาณที่ขั้วที่ 1, 2, 3 และ 4 โดยทำการเก็บรูปสัญญาณเปรียบเทียบกันทั้ง 4 ขั้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

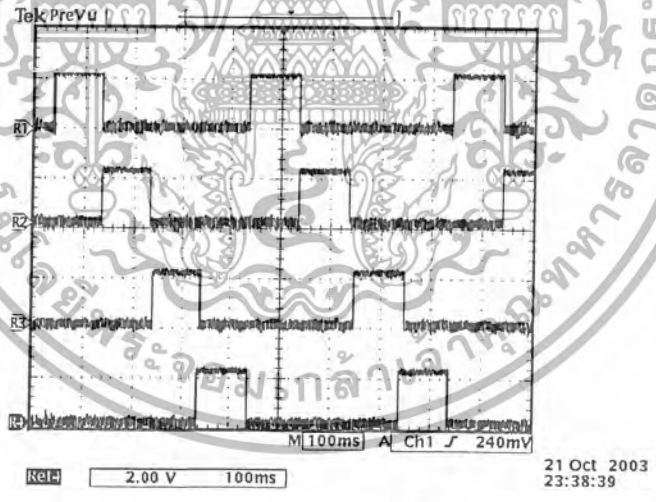
ผลการทดลอง

1. ผลการทดลองเมื่อทำการกดสวิตซ์ S1 เพื่อให้สเตปมอเตอร์หมุนทางขวา



รูปที่ 4.13 แสดงรูปสัญญาณที่พอร์ตต่างๆ เมื่อทำการกดสวิตซ์ S1 เพื่อให้สเตปมอเตอร์หมุนทางขวา

2. ผลการทดลองเมื่อทำการกดสวิตซ์ S2 เพื่อให้สเตปมอเตอร์หมุนทางซ้าย



รูปที่ 4.14 แสดงรูปสัญญาณที่พอร์ตต่างๆ เมื่อทำการกดสวิตซ์ S2 เพื่อให้สเตปมอเตอร์หมุนทางซ้าย

โดย สัญญาณ R1 คือสัญญาณที่วัดได้จากสเตปมอเตอร์ที่ขั้วที่ 1
สัญญาณ R2 คือสัญญาณที่วัดได้จากสเตปมอเตอร์ที่ขั้วที่ 2
สัญญาณ R3 คือสัญญาณที่วัดได้จากสเตปมอเตอร์ที่ขั้วที่ 3
สัญญาณ R4 คือสัญญาณที่วัดได้จากสเตปมอเตอร์ที่ขั้วที่ 4

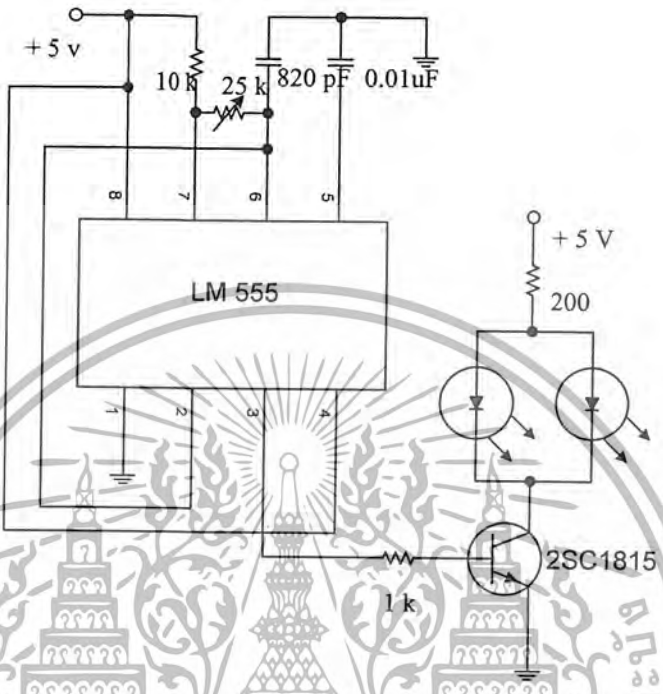
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองวงจรรับส่งอินฟราเรด

4.3.1 วงจรทางด้านส่ง

ขั้นตอนการทดลอง

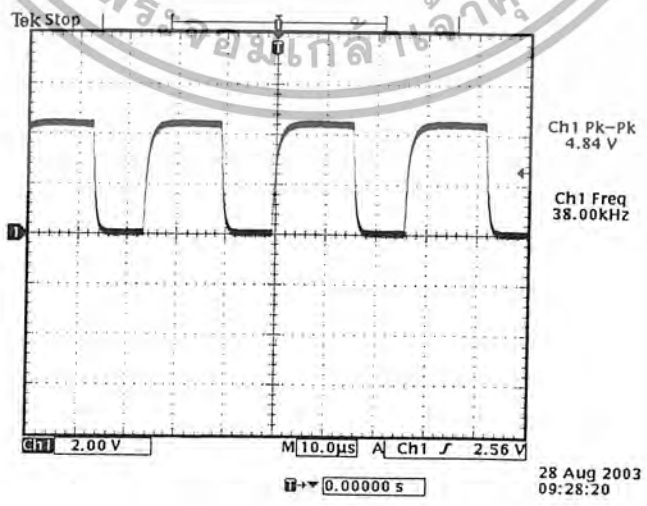
1. ต่อวงจรทางด้านส่งของอินฟราเรดตามรูปที่ 4.15



รูปที่ 4.15 แสดงวงจรทางด้านส่งอินฟราเรด

2. ทำวัดสัญญาณ Pulse ที่ขา3 ของ IC เบอร์ LM 555 บันทึกผลการทดลอง

ผลการทดลองวงจรทางด้านส่งอินฟราเรด



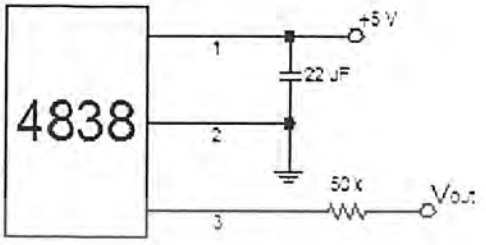
รูปที่ 4.16 แสดงผลการทดลองการวัดสัญญาณที่ขา 3 ของ IC เบอร์ LM555 เป็น pulse ความถี่ 38 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 วงจรทางด้านรับ

ขั้นตอนการทดลอง

1. ต่อวงจรทางด้านรับอินฟราเรดดังรูปที่ 4.17

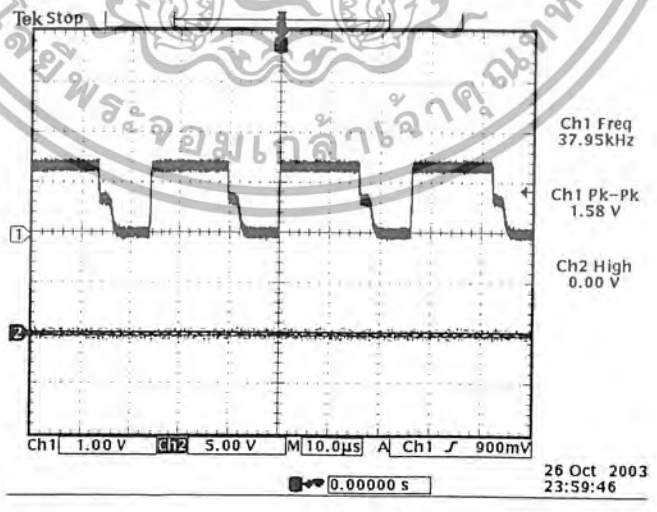


รูปที่ 4.17 แสดงวงจรทางด้านรับอินฟราเรด

2. ทดสอบการทำงานโดยการรับสัญญาณจากวงจรทางด้านส่งอินฟราเรดแล้วทำการวัดสัญญาณที่ขา Vout ของวงจร โดยวัดสัญญาณในกรณีที่รับสัญญาณได้ และ กรณีที่รับสัญญาณไม่ได้

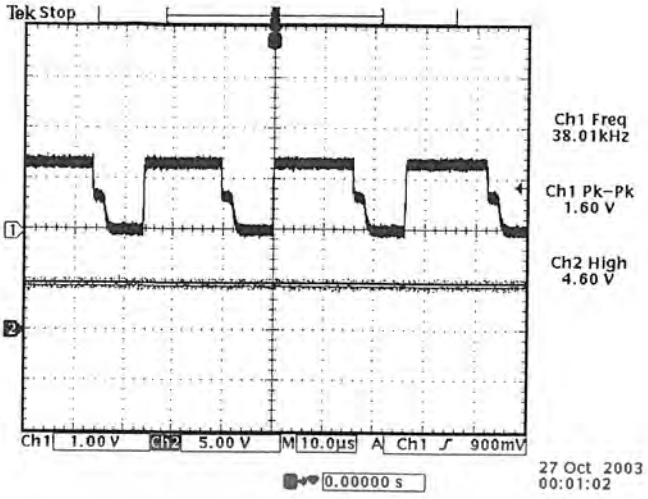
ผลการทดลองทางด้านรับอินฟราเรด

เนื่องจากการทดลองนี้ใช้ Photo module เป็นตัวรับแสง ในกรณีที่สามารถรับสัญญาณจากทางด้านส่งได้ จะให้สัญญาณ Output เป็นสัญญาณไฟ DC ขนาด 0 โวลต์ และกรณีที่รับสัญญาณจากทางด้านส่งไม่ได้ จะให้สัญญาณ Output เป็นไฟ DC ขนาด 5 โวลต์



รูปที่ 4.18 แสดงสัญญาณทางด้านรับกรณีที่รับสัญญาณได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 แสดงสัญญาณทางด้านรับกรณีที่รับสัญญาณไม่ได้

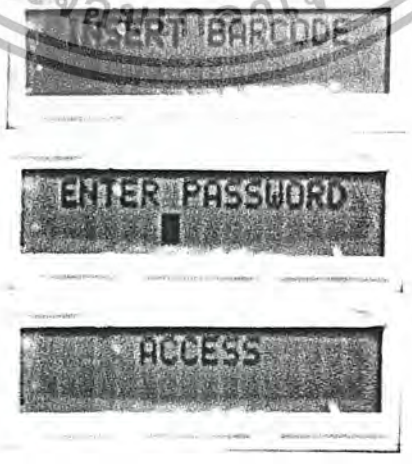
ผลการทดสอบระบบรวม

4.4 การทดสอบส่วนตัวเครื่อง

จะแบ่งการทดสอบออกเป็น 2 ส่วน ได้แก่ ส่วนของผู้ใช้งาน (USER) และ ส่วนของผู้ดูแลระบบ (ADMIN) โดยมีรายละเอียดการทดสอบดังนี้

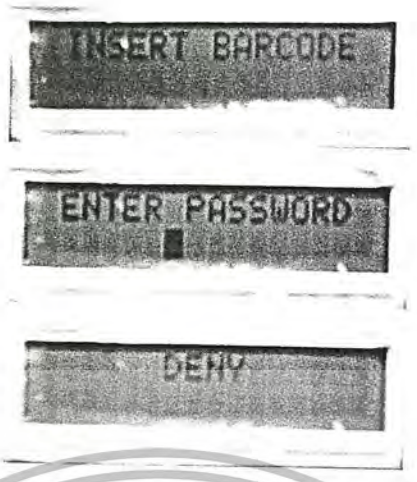
4.4.1 การทดสอบการเข้าใช้งานของ USER

การทดสอบทำโดยการกดปุ่มเพื่อร้องขอ เข้าใช้งานห้องจะแสดงหน้าจอ "INSERT BARCODE" จากนั้น ก็ทำการรูดบัตรนักศึกษาโดยถ้าเป็นนักศึกษาที่ได้รับอนุญาตให้เข้าใช้งานห้องได้ จะแสดงข้อความ "ENTER PASSWORD" ถ้าป้อน PASSWORD ถูก จะแสดงข้อความ "ACCESS" พร้อมทั้งสั่งให้ขับสเตปมอเตอร์ เพื่อคลายล็อกประตู และบันทึกข้อมูลลงใน ROM ถ้าผิดจะแสดงข้อความ "DENY" แต่ถ้าเป็นนักศึกษาที่ไม่ได้รับอนุญาต จะแสดงข้อความ "DENY" แสดงดังภาพตามกรณีต่างๆดังนี้



รูปที่ 4.20 กรณีที่นักศึกษาที่ได้รับอนุญาตให้เข้าใช้งาน ป้อน PASSWORD ถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 กรณีที่นักศึกษาที่ได้รับอนุญาตให้เข้าใช้งาน ป้อน PASSWORD ผิด

รูปที่ 4.22 กรณีที่นักศึกษาที่ไม่ได้รับอนุญาตให้เข้าใช้งาน

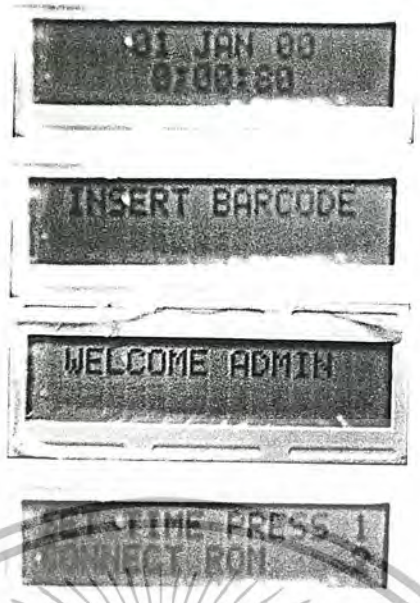
4.4.2 การทดสอบส่วนของผู้ดูแลระบบ

ผู้ดูแลระบบสามารถควบคุมการทำงานของตัวเครื่องได้ โดยมีรายละเอียดต่างๆดังนี้

1) การเข้าระบบของ ADMIN

การตั้งเวลา สามารถทำโดยการกดปุ่มเพื่อร้องขอการเข้าใช้งาน จะขึ้นหน้าจอ “INSERT BARCODE” จากนั้นให้ใช้บัตร “ADMIN” รูดผ่านเครื่องอ่านบาร์โค้ด จะขึ้นหน้าจอ “WELCOME ADMIN” จะมีรายการขึ้นมา 2 รายการ ให้เลือกที่ “SET TIME” และ “CONNECT ROM” โดยจะแสดงการเข้าระบบของ ADMIN ดังรูปที่ 4.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.23 การเข้าสู่ระบบของ "ADMIN"

2) การตั้งเวลาให้กับตัวเครื่อง

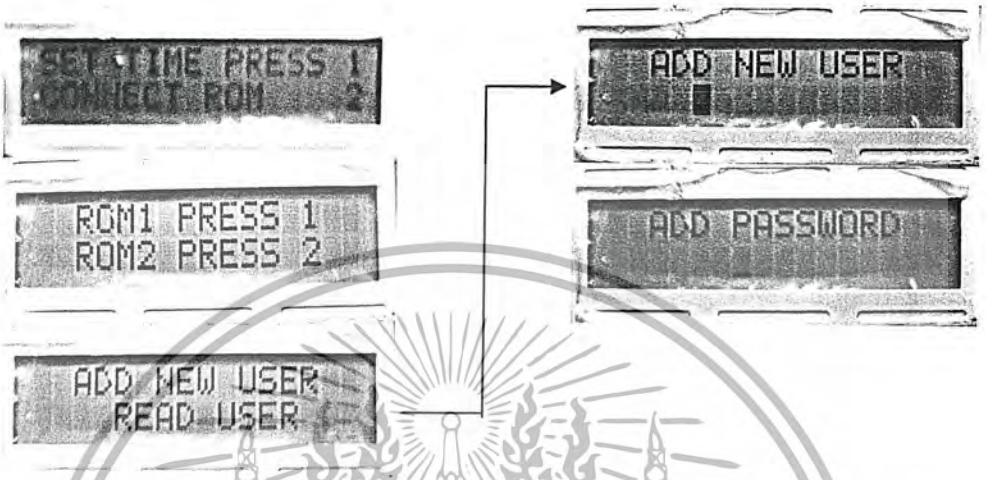
ทำโดยการเลือกรายการที่ 1 คือ "SET TIME" จะขึ้นหน้าจอให้ป้อนวันเดือนปี ในการทดสอบป้อนค่าเป็น 26 / 03 / 04 เมื่อป้อนค่าวัน, เดือน, ปีเรียบร้อย จะขึ้นหน้าจอให้ป้อนเวลา ในการทดสอบป้อนค่าเป็น 18:30:33 ซึ่งเมื่อตั้งค่าเสร็จเรียบร้อย ก็จะกลับไปหน้าจอหลักพร้อมทั้งแสดงค่าเวลาใหม่ตามที่ได้ตั้งไว้

รูปที่ 4.24 ขั้นตอนการตั้งเวลาให้กับตัวเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) การติดต่อกับ ROM ตัวที่ 1 เพื่อเพิ่มรหัสในฐานข้อมูล

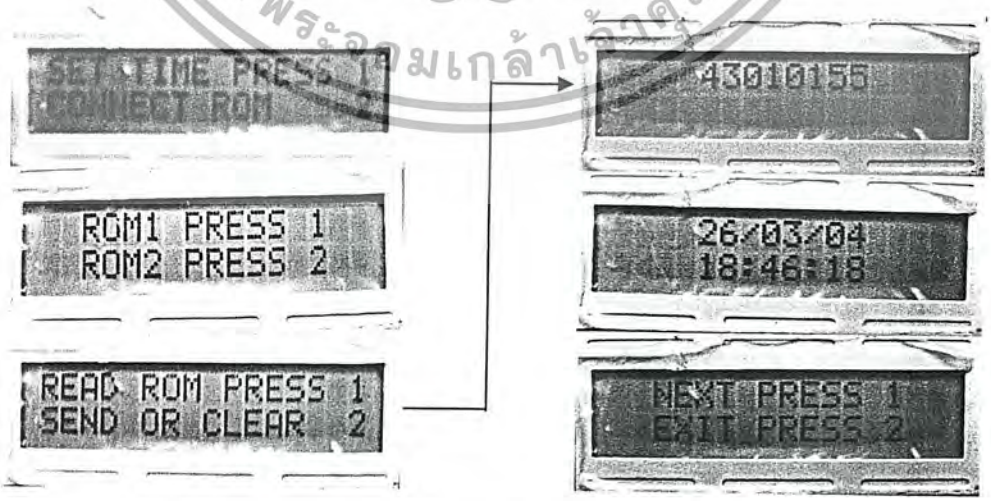
การเพิ่มรหัสทำโดยการเลือกรายการที่ 2 คือ "CONNECT ROM" เลือก "ADD NEW USER" จะขึ้นหน้าจอ สำหรับป้อนค่ารหัสนักศึกษา 8 หลัก เมื่อป้อนครบ จะขึ้นหน้าจอ "ADD PASSWORD" เพื่อป้อนรหัสผ่านเป็นตัวเลขจำนวน 4 หลัก แสดงการทำงานดังรูปที่ 4.25



รูปที่ 4.25 การติดต่อกับ ROM ตัวที่ 1 เพื่อเพิ่มรหัสในฐานข้อมูล

4) การติดต่อกับ ROM2 เพื่อดูฐานข้อมูลผู้เข้าใช้งาน

ข้อมูลของผู้ที่เข้าใช้งานห้องนั้น จะถูกเก็บไว้ใน ROM ตัวที่ 2 โดยการเลือก รายการ ROM 2 และเลือกรายการ "READ ROM" ก็จะขึ้นข้อมูลเป็นหน้าจอรหัสของผู้ที่เข้าใช้งาน และหน้าจอแสดงวันเวลาที่เข้าใช้งาน และจะขึ้นหน้าจอเพื่อเลือกว่าจะดูข้อมูลต่อไปหรือจะยกเลิกการดูข้อมูล ถ้าจะดูข้อมูลต่อเลือก "NEXT" ถ้าจะยกเลิกการดูข้อมูล เลือก "EXIT" แสดงการทำงานดังรูปที่ 4.26

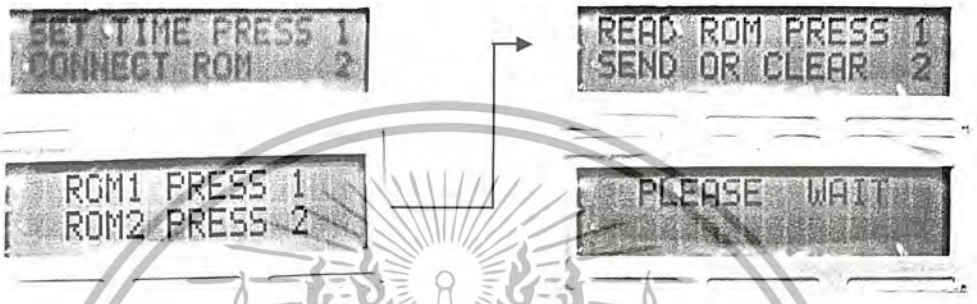


รูปที่ 4.26 การติดต่อกับ ROM2 เพื่อดูฐานข้อมูลผู้เข้าใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) การส่งข้อมูลผู้ใช้งานที่เก็บใน ROM2 ไปยังคอมพิวเตอร์

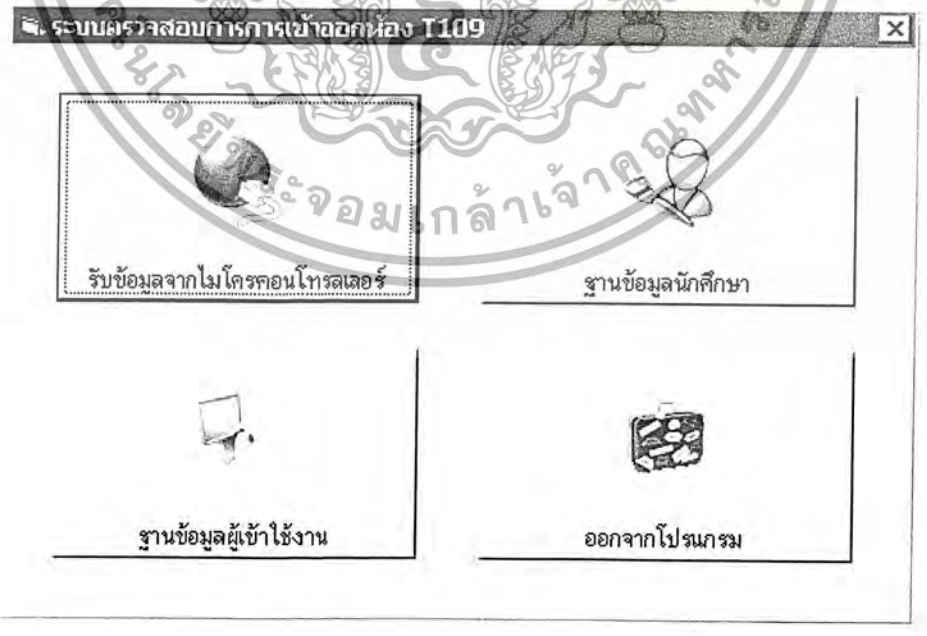
ข้อมูลที่จะส่งไปยังคอมพิวเตอร์จะประกอบไปด้วย รหัสนักศึกษา, วันเดือนปี และเวลาที่เข้าใช้งานโดย 1 ชุดข้อมูล จะมี 20 ไบต์ โดยการส่งจะเข้ารายการ ROM 2 เลือกรายการ "SEND OR CLEAR" และเลือก "SEND" จะทำการส่งข้อมูล และ ขึ้นข้อความ "PLEASE WAIT" จนกระทั่งส่งข้อมูลเสร็จ ก็จะกลับสู่หน้าจอแสดงเวลา แสดงดังรูปที่ 4.27 และเมื่อข้อมูลเต็มพื้นที่ใน ROM ก็สามารถ CLEAR ข้อมูลโดยใช้รายการ "CLEAR"



รูปที่ 4.27 การส่งข้อมูลผู้ใช้งานที่เก็บใน ROM2 ไปยังคอมพิวเตอร์

4.5 การทดสอบโปรแกรมรับข้อมูลบนเครื่องคอมพิวเตอร์

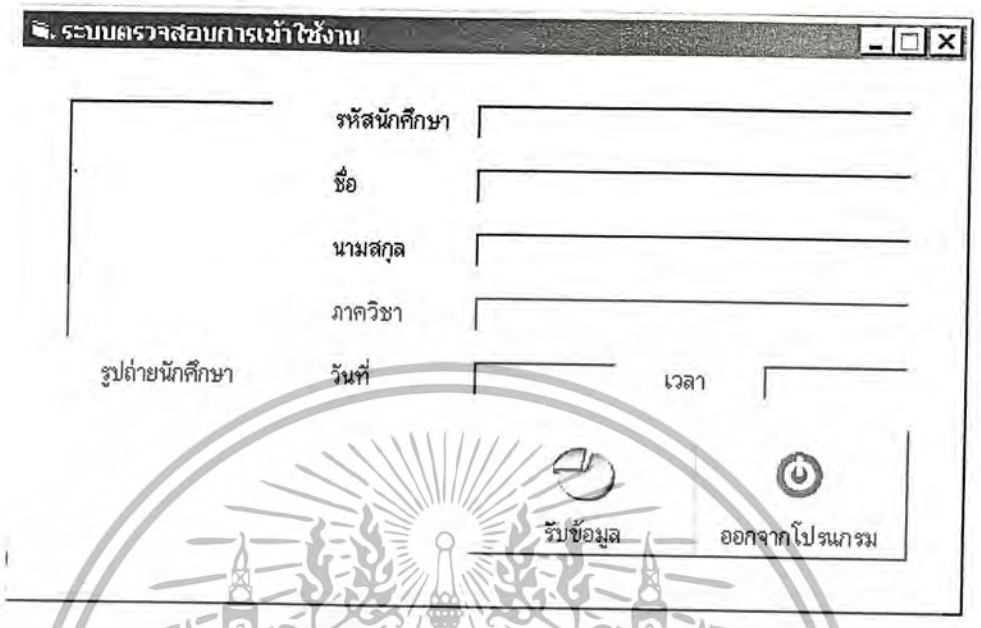
โปรแกรมรับข้อมูลบนคอมพิวเตอร์ จะประกอบไปด้วย โปรแกรมสำหรับรับข้อมูล, ฐานข้อมูลของผู้เข้าใช้งาน และ ฐานข้อมูลของนักศึกษาที่ได้รับอนุญาตให้เข้าใช้งาน



รูปที่ 4.28 โปรแกรมที่ใช้รับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดลองโดยการส่งค่าข้อมูลใน ROM2 ของไมโครคอนโทรลเลอร์ แล้วทำการเรียกโปรแกรมรับข้อมูลโดยการกดรับข้อมูลจากไมโครคอนโทรลเลอร์ จะปรากฏหน้าต่างโปรแกรมดังรูปที่ 4.29



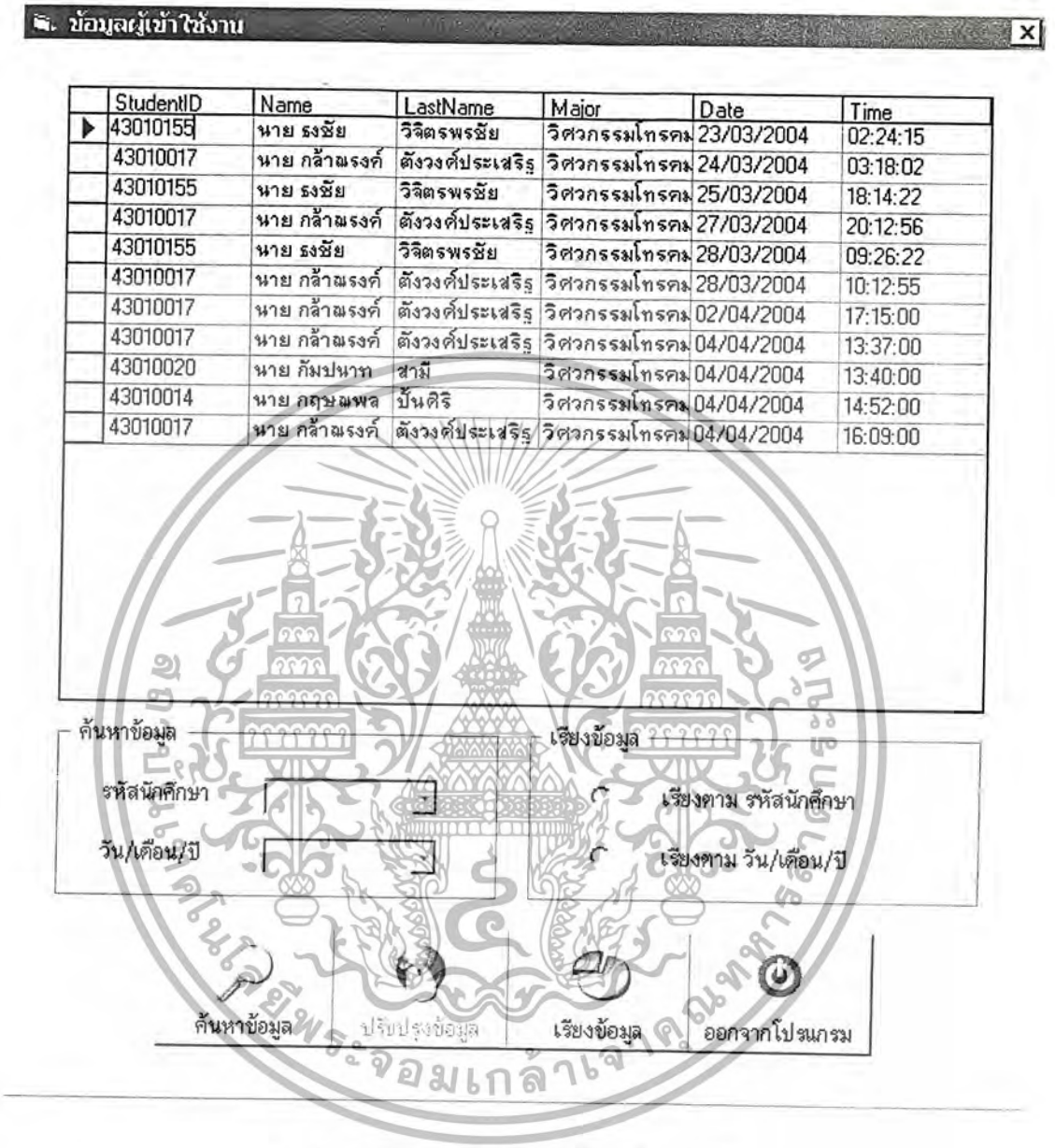
รูปที่ 4.29 หน้าจอขณะรอรับข้อมูลจากไมโครคอนโทรลเลอร์

เมื่อไมโครคอนโทรลเลอร์ส่งค่ารหัสมาที่คอมพิวเตอร์ ก็ให้กดปุ่ม "รับข้อมูล" เพื่อรับข้อมูลจะได้ผลตามรูปที่ 4.30 ในกรณีที่มีข้อมูลมากกว่า 1 ชุดข้อมูลก็ให้กด "รับข้อมูล" เพื่อรับข้อมูลชุดต่อๆไป

รูปที่ 4.30 แสดงหน้าจอเมื่อกดปุ่ม "รับข้อมูล" เพื่อรับข้อมูลจากไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยข้อมูลทั้งหมดจะถูกเก็บในฐานข้อมูลผู้เข้าใช้งาน สามารถตรวจสอบโดยการกดปุ่ม "ฐานข้อมูลผู้เข้าใช้งาน" เพื่อดูข้อมูล



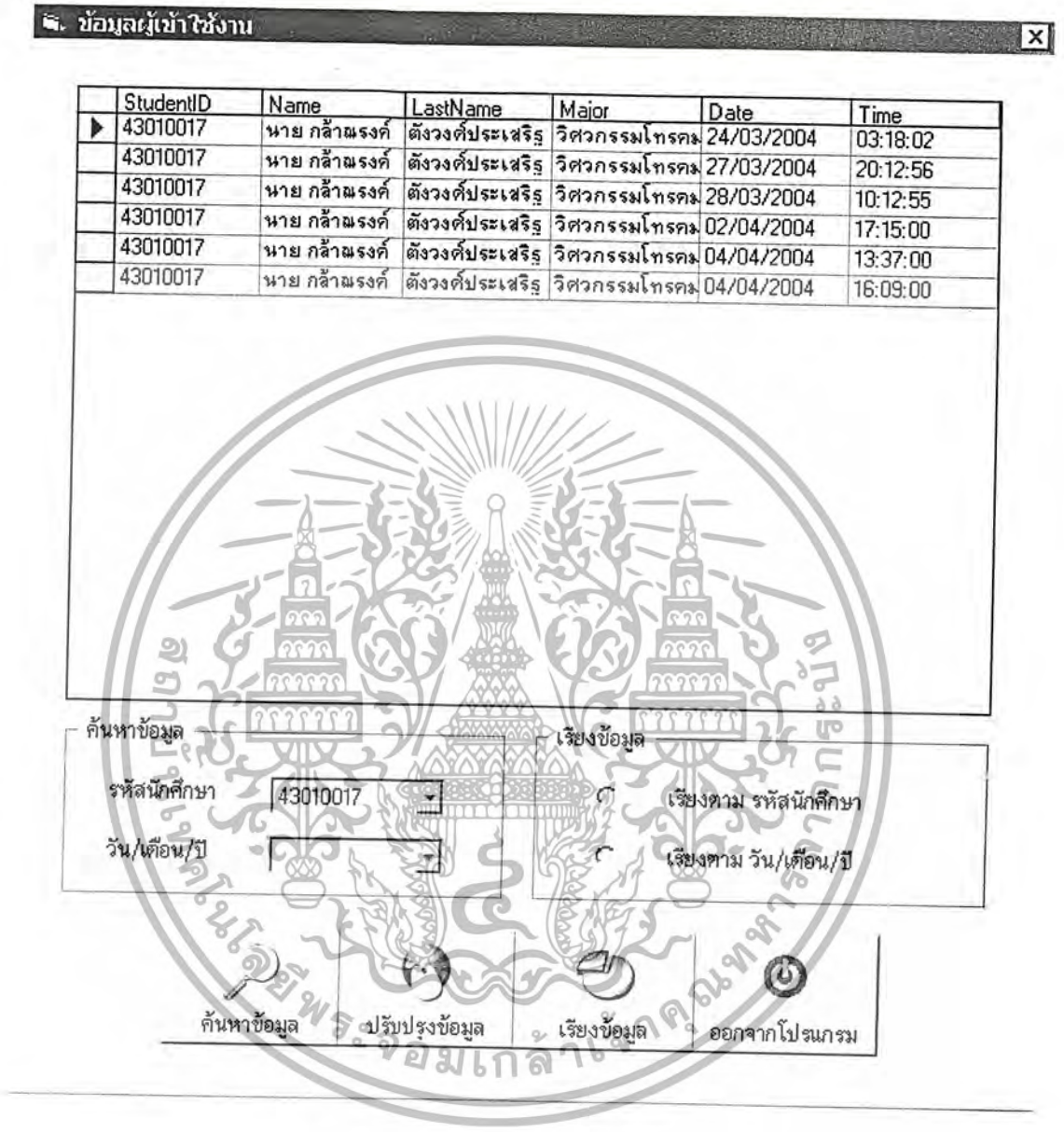
รูปที่ 4.31 แสดงข้อมูลในฐานข้อมูลผู้เข้าใช้งาน

โดยฐานข้อมูลผู้เข้าใช้งาน สามารถค้นหาข้อมูลโดยค้นหาตามรหัสนักศึกษา ค้นหาตามวันที่ หรือค้นหาทั้ง 2 อย่าง หรือเลือกเรียงข้อมูลโดยเรียงตามรหัส หรือ เรียงตามวันที่

การค้นหาข้อมูล ทำโดยการกดปุ่มค้นหาข้อมูลและเลือกเงื่อนไขในการค้นหา ว่าจะค้นหาตามรหัสนักศึกษา หรือ วันเดือนปีที่เข้าใช้งาน หรือทั้ง 2 เงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

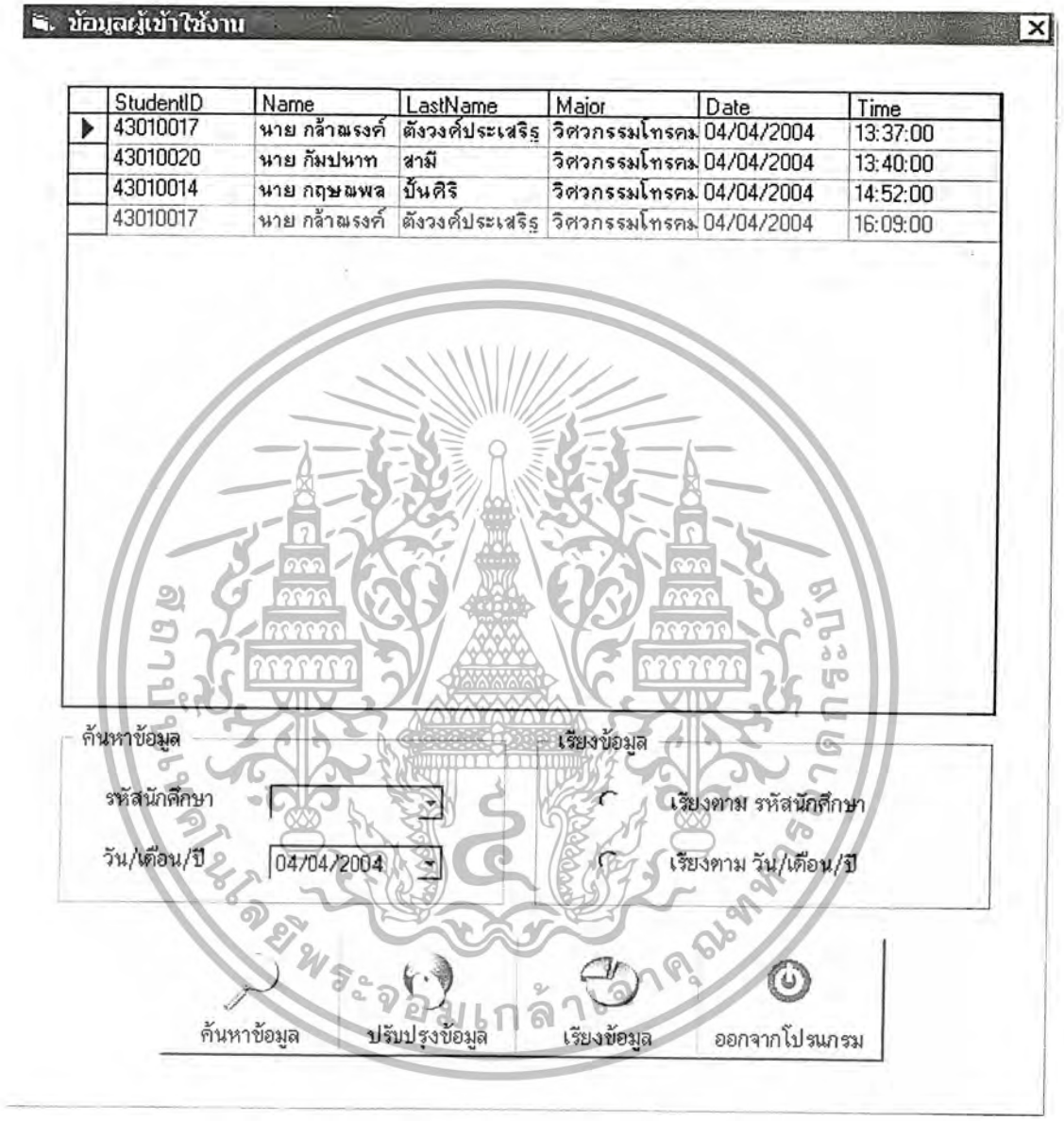
การค้นหาตามรหัสนักศึกษา ทำโดยเลือกรหัสนักศึกษา และ กดปุ่มปรับปรุงข้อมูล ข้อมูลรหัสนักศึกษาที่ต้องการก็จะปรากฏขึ้นมาที่หน้าจอดังรูปที่ 4.32



รูปที่ 4.32 แสดงการค้นหาข้อมูลตามรหัสนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

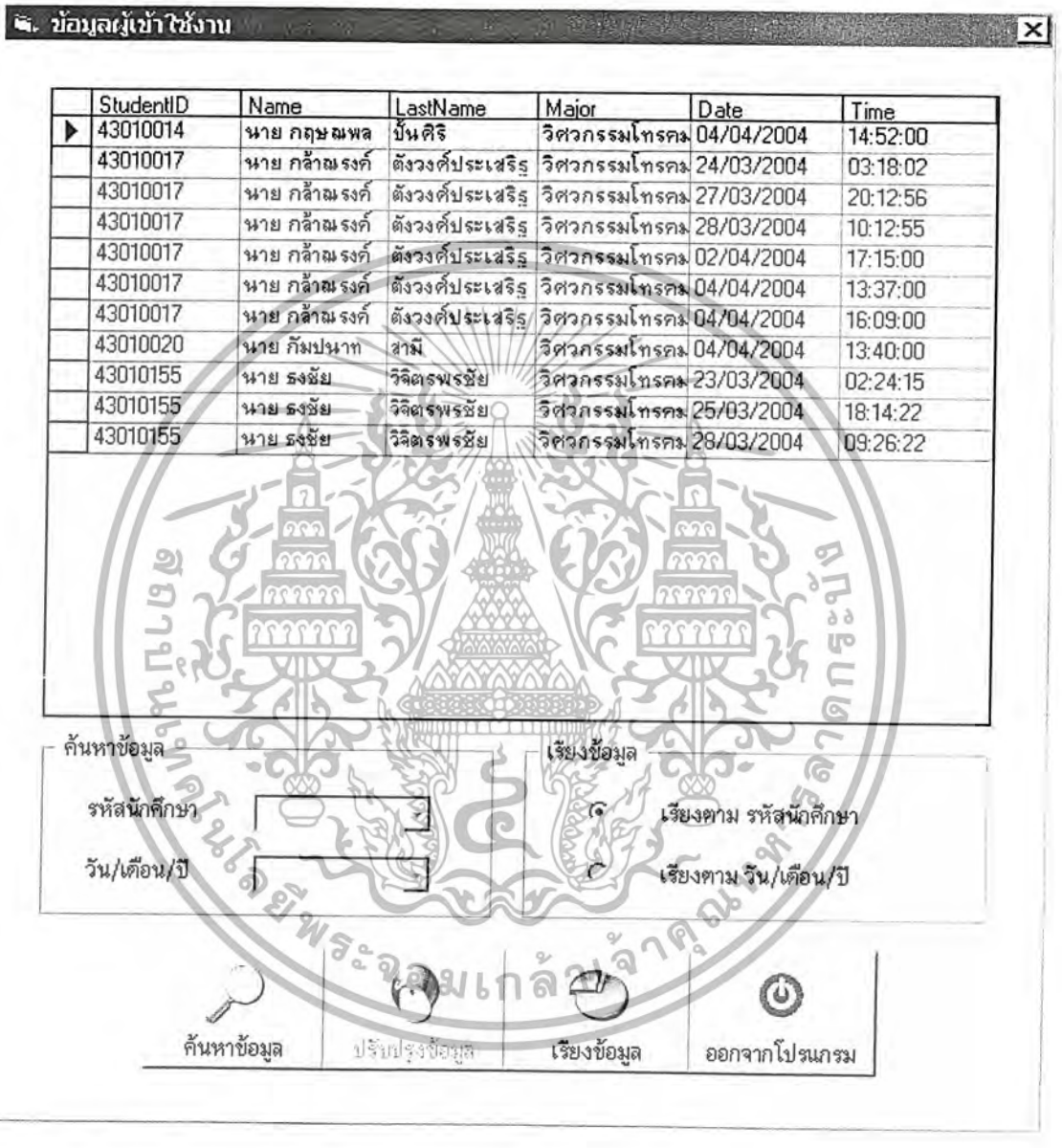
การค้นหาตามวันเดือนปีที่เข้าใช้งาน ทำโดยการเลือกวันเดือนปีที่ต้องการค้นหา แล้วกดปุ่มปรับปรุงข้อมูล ข้อมูลของรหัสนักศึกษาที่เข้าใช้งานใน วันเดือนปี ที่เลือก ก็จะปรากฏที่หน้าจอ โดยผลการค้นหาแสดงดังรูปที่ 4.33



รูปที่ 4.33 แสดงการค้นหาข้อมูลตามวันเดือนปีที่เข้าใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเรียงข้อมูล ใช้ในกรณีที่ต้องการให้แสดงข้อมูลเป็นแบบเรียงตามรหัสนักศึกษา เพื่อความสะดวกในการดูข้อมูลการเข้าใช้งาน โดยการเรียงข้อมูลนี้ สามารถทำโดยการกดปุ่ม "เรียงข้อมูล" และเลือกลักษณะการเรียงว่าจะเรียงตามรหัสนักศึกษาหรือเรียงตามวันเดือนปีที่เข้าใช้งาน โดยตัวอย่างการเรียงข้อมูลตามรหัสนักศึกษาแสดงดังรูปที่ 4.34



รูปที่ 4.34 แสดงการเรียงข้อมูลตามรหัสนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลนักศึกษา เป็นส่วนที่เก็บข้อมูลของนักศึกษาเพื่อใช้ในการเปรียบเทียบข้อมูลรหัสที่รับจากไมโครคอนโทรลเลอร์เพื่อนำชื่อนามสกุล มาแสดงผลที่หน้าจอ โดยฐานข้อมูลนี้สามารถเรียกใช้งานโดยการกดปุ่ม "ฐานข้อมูลนักศึกษา" โดยสามารถเพิ่ม, ลบ หรือ แก้ไขฐานข้อมูลได้ ฐานข้อมูลนักศึกษา แสดงดังรูปที่ 4.35

รูปที่ 4.35 แสดงฐานข้อมูลนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และบทสรุป

ในโครงงานนี้จะแบ่งเป็น 2 ส่วนใหญ่ๆคือ ส่วนของไมโครคอนโทรลเลอร์เพื่อควบคุมการทำงานของอุปกรณ์ต่างๆในระบบ และส่วนที่ทำการเชื่อมต่อกับคอมพิวเตอร์เพื่อทำการรับข้อมูลที่เก็บข้อมูลการใช้งานของผู้ที่มาใช้งานเพื่อไปบันทึกลงในฐานข้อมูล

ในการทดลองส่วนของไมโครคอนโทรลเลอร์มีการทำงานคือ จะทำการรับค่าจากเครื่องอ่านบัตรรหัสแถบ จากนั้นจะทำการตรวจสอบค่าว่าเป็นค่าที่ได้จากบัตรของผู้ดูแลระบบหรือไม่ ถ้าเป็นค่าที่ได้จากบัตรของผู้ดูแลระบบก็จะให้เลือกว่าจะทำการตั้งค่าเวลาใหม่ หรือทำการติดต่อกับฐานข้อมูลที่เก็บในรอม โดยแบ่งเป็นรอมตัวที่ 1 ที่เก็บค่ารหัสของผู้ที่สามารถเข้าใช้งานระบบได้ โดยจะสามารถอ่านค่ารหัส และรหัสผ่านของแต่ละคนได้ หรือจะทำการเพิ่มค่ารหัสที่สามารถเข้าใช้งานระบบได้เข้าไปในฐานข้อมูลในรอมได้ ส่วนรอมตัวที่ 2 จะทำการเก็บค่าข้อมูลการเข้าใช้งานระบบของแต่ละคนว่ามีใครเข้ามาใช้งานบ้างและเข้ามาตอนเวลาเท่าใด ซึ่งจะสามารถเลือกที่จะทำการอ่านค่าข้อมูลที่บันทึกไว้อยู่ หรือทำการส่งค่าที่บันทึกไว้ไปยังคอมพิวเตอร์เพื่อไปบันทึกในฐานข้อมูลที่เก็บไว้บนคอมพิวเตอร์ หรือจะทำการลบข้อมูลที่บันทึกไว้เพื่อทำการเริ่มต้นเก็บข้อมูลใหม่

แต่ถ้าค่าที่รับจากเครื่องอ่านบัตรรหัสแถบไม่ใช่ค่าที่ได้จากบัตรผู้ดูแลระบบก็จะทำการตรวจสอบว่าเป็นค่าที่มีอยู่ในฐานข้อมูลที่เก็บไว้ในรอมตัวที่ทำการเก็บรหัสของผู้ที่สามารถเข้าใช้งานระบบได้หรือไม่ ถ้าไม่ใช่ค่าที่มีอยู่ในรอม ก็จะไม่อนุญาตให้เข้าใช้งานระบบ แต่ถ้าเป็นค่าที่มีอยู่ในรอมก็จะทำการรับค่ารหัสผ่าน 4 หลัก เพื่อนำไปตรวจสอบกับค่าที่เก็บไว้ในรอม ถ้าค่ารหัสผ่านไม่ถูกต้องก็จะไม่อนุญาตให้เข้าใช้งานระบบ แต่ถ้าค่าของรหัสผ่านถูกต้องก็จะสั่งให้สเตปมอเตอร์หมุนเพื่อปลดล็อคกลอนประตู จากนั้นจะทำการตรวจสอบการปิดประตูจากส่วนของ วงจรตัวส่ง และตัวรับสัญญาณอินฟราเรด เมื่อตรวจพบการปิดประตูแล้วก็จะสั่งให้สเตปมอเตอร์ทำการหมุนเพื่อล็อคประตู เป็นการจบการทำงานของส่วนไมโครคอนโทรลเลอร์ ซึ่งในการทดลองระบบในส่วนของไมโครคอนโทรลเลอร์นั้น สามารถให้ผลออกมาตามที่ออกแบบไว้ได้

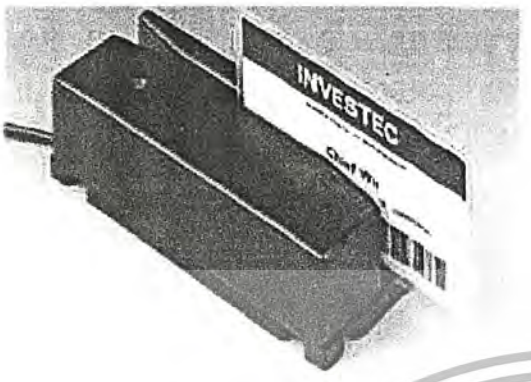
ในส่วนที่ทำการเชื่อมต่อกับคอมพิวเตอร์ จะทำการรับค่าจากไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรมและสามารถนำค่าที่ได้มาประมวลผลเพื่อตรวจสอบกับฐานข้อมูลที่บันทึกไว้ในคอมพิวเตอร์ ซึ่งสามารถทำการตรวจสอบกับฐานข้อมูลในคอมพิวเตอร์และสามารถบันทึกลงในฐานข้อมูลและสามารถเรียกฐานข้อมูลที่บันทึกขึ้นมาดู และแก้ไขได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

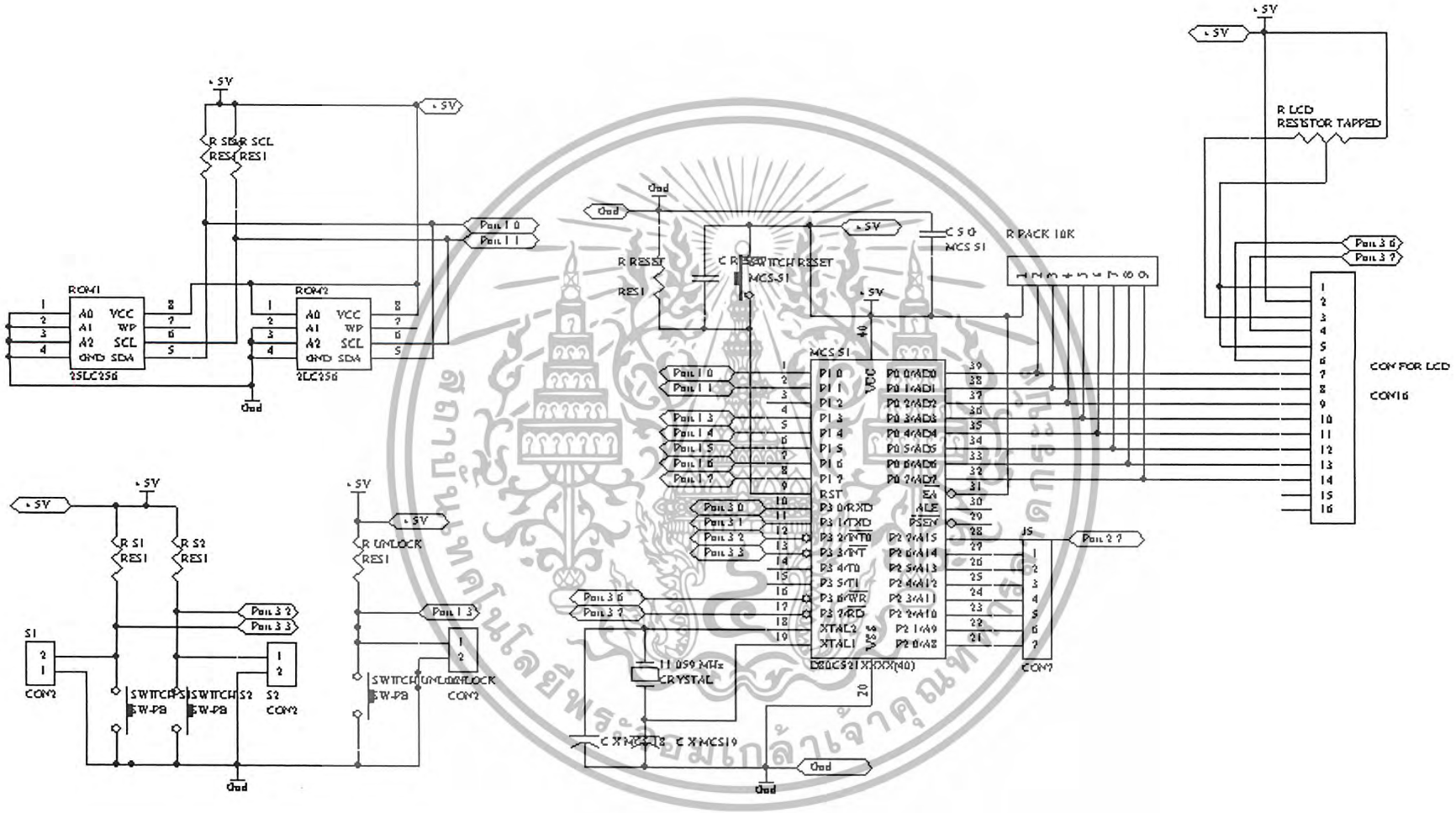
เครื่องอ่านบาร์โค้ด ArexTek รุ่น 103DR



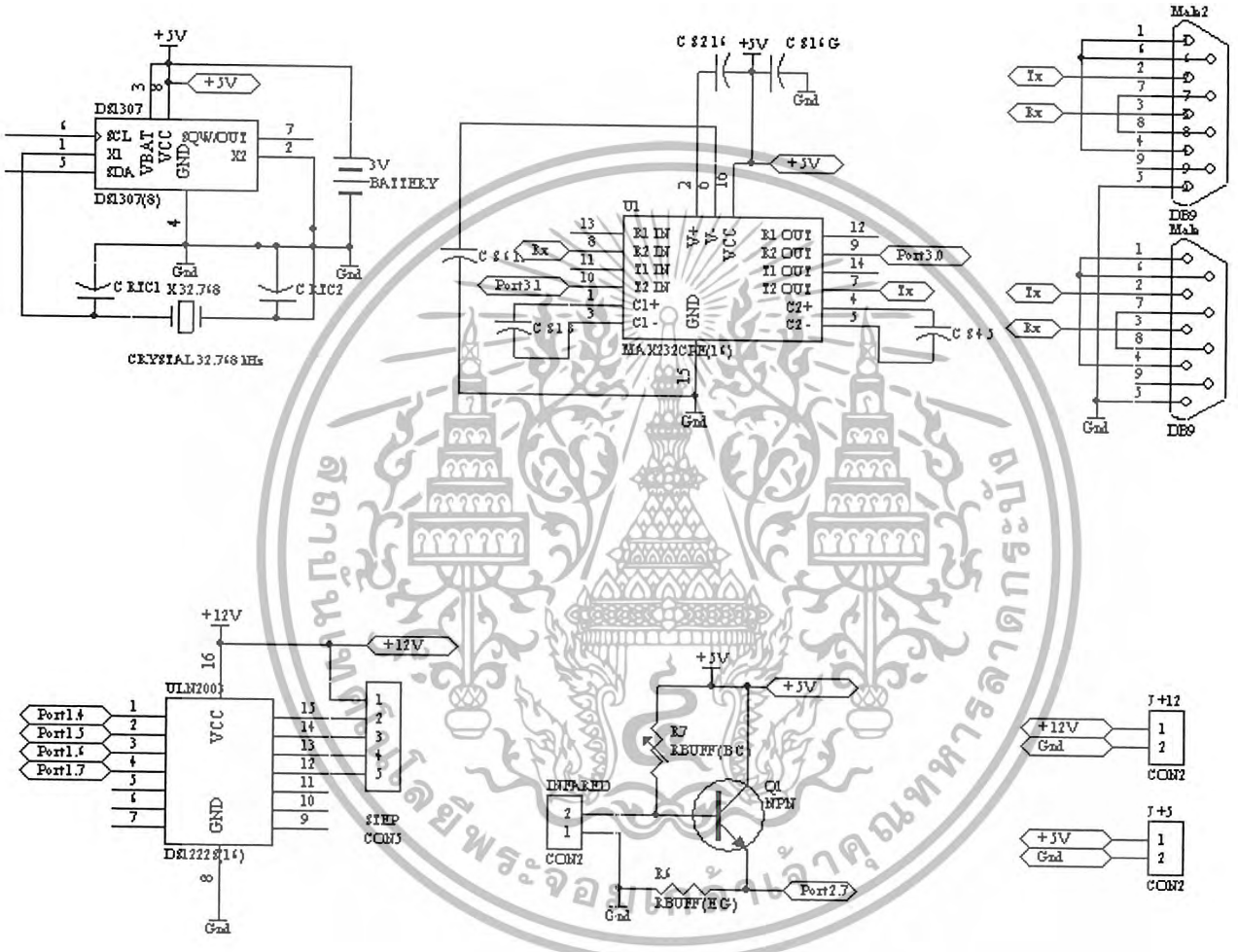
Hardware Specification

Output Interface Signal	:	RS232 (Standard DB9)
Light Source	:	Infrared 940 nm
Sensor	:	Hi-density photo image sensor
Power Consumption	:	90 mA /5 VDC
Card Thickness	:	0 – 1.5 mm
Reading Speed	:	2-2800 mm/Sec at 0.19 mm
Resolution	:	0.116 mm (4.7 mil)
Programmable Multifunction	:	RS232 Format, Barcode Format, Inter-character Delay
Barcode Response	:	UPC-A, UPC-E, EAN/JAN, EAN/JAN-8, EAN/UPC SUP,
	:	ISBN/ISSN, Code39, Code39EXT, Code25, Code128,
	:	Code93, Codebar/NW7 and Code11
Indicator	:	LED Power,Buzzer Beep for any function
Body Meterial	:	Metal die-casting
Body Dimension	:	114.5 mm x 44 mm x 36.7 mm
Weight	:	440 g (with Cable)

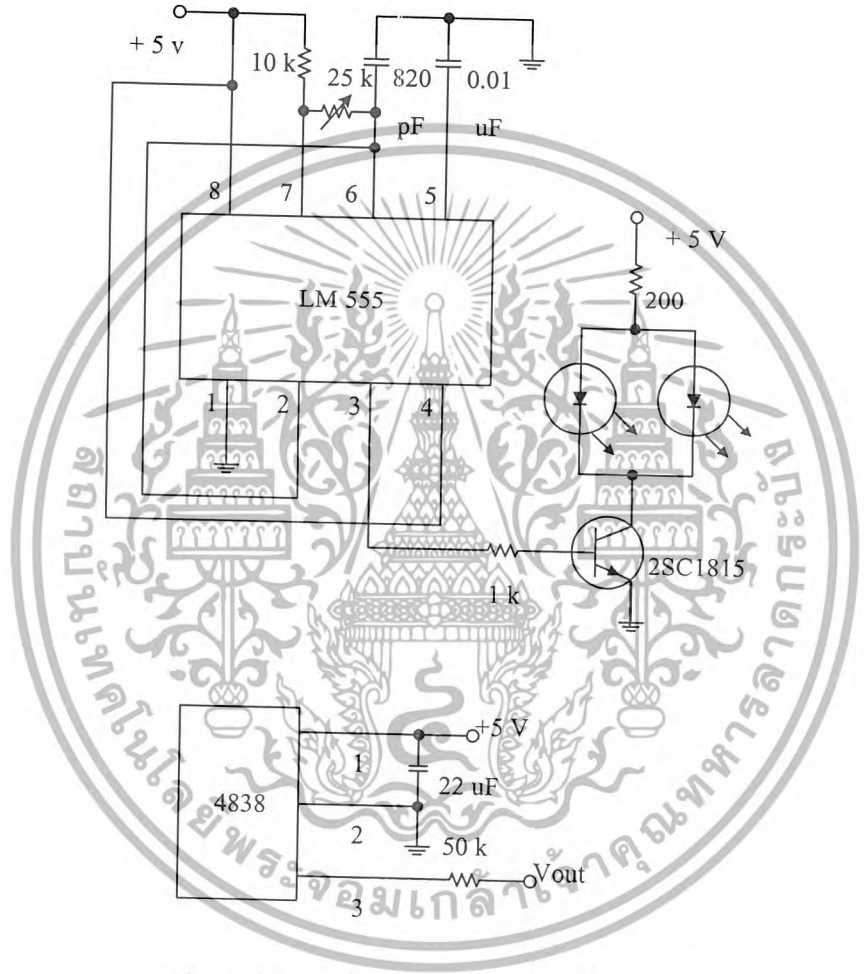
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปวงจรส่วนไมโครคอนโทรลเลอร์ที่ใช้ในโครงงาน



รูปวงจรส่วนไมโครคอนโทรลเลอร์ที่ใช้ในโครงงาน (ต่อ)



รูปวงจรส่วนภาคส่ง และรับสัญญาณอินฟราเรด

เอกสารอ้างอิง

1. นายกิตติคุณ พร้อมพลากร, นายไพรัตน์ วิบูลศรี, นายไววิทย์ ศิริตา , " ระบบรักษาความปลอดภัย โดยใช้ รหัสแถบ (A simple security system using Barcode) " , ปริญญานิพนธ์ หลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาอิเล็กทรอนิกส์ ปีการศึกษา 2543
2. นายกวิน ยะมาลี, นายชาลี วรกุลพิพัฒน์ , " เครื่องบันทึกเวลาพนักงานโดยใช้รหัสแถบ (Barcode time recording machine) " , ปริญญานิพนธ์ หลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาอิเล็กทรอนิกส์ ปีการศึกษา 2539
3. ชัยวัฒน์ ลิ้มพรจิตรวิไล , วงพจน์ กรแก้ววัฒนกุล , "เรียนรู้ และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 " , บริษัท อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด , กรุงเทพฯ , 2541
4. นายไพบูลย์ ศิริพัฒน์, " ระบบการเก็บข้อมูลรหัสแถบ (Barcode data collecting system) " , วิทยานิพนธ์หลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟ ฟ้า พ.ศ. 2536
5. Michael Halvorson , นายอนิรุต ลิ่วหาทอง , " คู่มือการเรียนรู้ด้วยตนเอง Step By Step Microsoft Visual Basic " , บริษัท ซีเอ็ดยูเคชั่น จำกัด , กรุงเทพฯ , 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้