สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ชุดบริการข้อมูลอัตโนมัติ
AUTOMATIC DATA SERVICE

โดย
นายพีรพัฒน์ อริยะกวี
นายเวชยันต์ พันรูปท้าว

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2546

เลขหมู่
เลขทะเบียน 54951
วัน,เดือน,ปี - 1 มิ.ย. 2548
b.....
i.....
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดบริการข้อมูลอัตโนมัติ

AUTOMATIC DATA SERVICE

โดย

นายพีรพัฒน์ อริยะกวี 44015021

นายเวชยันต์ พันรูปท้าว 44015027

อาจารย์ที่ปรึกษา

อาจารย์สุรพล บุญจันทร์

อาจารย์พิสิฐ บุญศรีเมือง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดบริการข้อมูลอัตโนมัติ
AUTOMATIC DATA SERVICE

โดย นายพีรพัฒน์ อริยะกวี 44015021
นายเวชยันต์ พันรูปท้าว 44015027

อาจารย์ที่ปรึกษา อาจารย์สุรพล บุญจันทร์
อาจารย์พิสิฐ บุญศรีเมือง

บทคัดย่อ

โครงงานนี้จะให้บริการข้อมูลในรูปแบบของมัลติมีเดียภายในคอมพิวเตอร์ โดยใช้โปรแกรมวิชวลเบสิกเวอร์ชั่น 6.0 มาช่วยในการแสดงข้อมูล ซึ่งจะแสดงข้อมูลให้แก่ผู้ชมผลงานหรือชิ้นงานได้โดยอัตโนมัติผ่านระบบเซ็นเซอร์ โดยใช้งานร่วมกับชุดควบคุมไมโครคอนโทรลเลอร์เป็นตัวควบคุมการให้บริการข้อมูลและหยุดให้บริการข้อมูลในคอมพิวเตอร์นั้นๆ โดยอัตโนมัติ และสามารถนำไปใช้ในการจัดแสดงนิทรรศการต่างๆที่ต้องการได้ ซึ่งจะเป็นประโยชน์แก่ผู้จัดงานแสดงผลงานและผู้เข้าร่วมงานได้รับความสะดวกสบายมากขึ้น.

ABSTRACT

This project is a Automatic Data Service in multimedia from using Visual Basic version 6.0 according to exhibit the certained data for visitor who interested our project by automatic sensor system. It uses Microcontroller to control start and stop data service in form that computer by automatic. It can also applied that want to present their work incentively. This Automatic Data Service will be benefit for both the exhibition organizer and participants with their most convenient.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2546

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ชุดบริการข้อมูลอัตโนมัติ

AUTOMATIC DATA SERVICE

ผู้จัดทำ

1. นายพีรพัฒน์ อริยะกวี 44015021
2. นายเวชยันต์ พันธุ์ปทาว 44015027

.....อาจารย์ที่ปรึกษา

(อาจารย์สุรพล บุญจันทร์)

.....อาจารย์ที่ปรึกษา

(อาจารย์พิสิฐ บุญศรีเมือง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีที่เกี่ยวข้องและแนวทางการทำโครงงาน	2
2.1 แอลอีดีอินฟราเรดกำลังสูง	2
2.2 ไมโครคอนโทรลเลอร์ MCS-51	3
2.3 พอร์ทอนุกรม	15
2.4 การสื่อสารแบบอนุกรม	16
2.5 การเขียนโปรแกรมวิชวลเบสิกเพื่อใช้งานพอร์ตอนุกรม	22
2.6 การสร้างแอพพลิเคชันฐานข้อมูล	27
2.7 การติดต่อกับฐานข้อมูลด้วย ActiveX Data Object	29
บทที่ 3 การสร้างและแนวความคิดในการออกแบบ	35
3.1 การออกแบบวงจรอินฟราเรดเซนเซอร์	35
3.2 การออกแบบชุดควบคุมไมโครคอนโทรลเลอร์	37
3.3 การออกแบบโปรแกรมคอมพิวเตอร์	42
บทที่ 4 การทดลองและผลการทดลอง	47
4.1 อุปกรณ์ที่ใช้ในการทดลอง	47
4.2 วงจรภาคส่งอินฟราเรดเซนเซอร์	47
4.3 วงจรภาครับอินฟราเรดเซนเซอร์	49
4.4 ชุดควบคุมไมโครคอนโทรลเลอร์	50
4.5 โปรแกรมวิชวลเบสิก เวอร์ชัน 6.0	53
บทที่ 5 บทวิจารณ์และสรุป	60
ภาคผนวก	
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

		หน้า
รูปที่ 1.1	แสดงภาพรวมของชุดบริการข้อมูลอัตโนมัติ	1
รูปที่ 2.1	ก) แสดงวงจรสร้างสัญญาณไบอัสอินฟราเรดธรรมดา	3
รูปที่ 2.1	ข) แสดงวงจรสร้างสัญญาณไบอัสอินฟราเรดเพื่อเพิ่มปริมาณความเข้มของแสง	3
รูปที่ 2.2	แสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ในอนุกรม AT89C51	4
รูปที่ 2.3	แสดงการจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C51	5
รูปที่ 2.4	แสดงโครงสร้างของพอร์ต 0 (บิต)	7
รูปที่ 2.5	แสดงโครงสร้างของพอร์ต 1 (บิต)	7
รูปที่ 2.6	แสดงโครงสร้างของพอร์ต 2 (บิต)	8
รูปที่ 2.7	แสดงโครงสร้างของพอร์ต 3 (บิต)	8
รูปที่ 2.8	แสดงการจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์เบอร์ AT89C51	10
รูปที่ 2.9	แสดงการเชื่อมต่อหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์ แบบแฟลช	10
รูปที่ 2.10	แสดงการเชื่อมต่อหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์ แบบแฟลช	11
รูปที่ 2.11	แสดง การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในไมโครคอนโทรลเลอร์ แบบแฟลช	12
รูปที่ 2.12	แสดงการจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในส่วนล่างของ ไมโครคอนโทรลเลอร์ แบบแฟลช	12
รูปที่ 2.13	แสดงการจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในส่วนบน ไมโครคอนโทรลเลอร์ แบบแฟลช	13
รูปที่ 2.14	แสดงรูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม	16
รูปที่ 2.15	แสดงรูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส	17
รูปที่ 2.16	ก)แสดงการจัดขาของคอนเน็กเตอร์พอร์ตอนุกรม แบบ DB-9	19
รูปที่ 2.16	ข)แสดงการจัดขาของคอนเน็กเตอร์พอร์ตอนุกรม แบบ DB-25	19
รูปที่ 2.17	ก)แสดงการต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์แบบนัลโมเด็ม	20
รูปที่ 2.17	ข)แสดงการต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ โดยใช้สายสัญญาณเพียง 3 เส้น	21
รูปที่ 2.18	แสดงวงจรขับแบบ อาร์เอสสองสามสอง โดยใช้ MAX 232	22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 2.19 แสดงมุมมองของโปรแกรมวิชวลเบสิกกับฐานข้อมูล 28

รูปที่ 2.20 แสดงโครงสร้างออบเจ็กต์ของ ActiveX Data Object 30

รูปที่ 3.1 แสดงวงจรภาคส่งอินฟราเรดเซนเซอร์ 35

รูปที่ 3.2 แสดงวงจรภาครับอินฟราเรดเซนเซอร์ 36

รูปที่ 3. 3 แสดงวงจรไมโครคอนโทรลเลอร์ ตัวที่ 1 37

รูปที่ 3. 4 แสดงวงจรไมโครคอนโทรลเลอร์ ตัวที่ 2,ตัวที่ 3 และ ตัวที่ 4 38

รูปที่ 3.5 แสดงโฟลว์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์ตัวที่ 1 39

รูปที่ 3.6 แสดงโฟลว์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์ตัวที่ 2 40

รูปที่ 3.7 แสดงโฟลว์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์ตัวที่ 3 40

รูปที่ 3.8 แสดงโฟลว์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์ตัวที่ 4 41

รูปที่ 3.9 แสดงโฟลว์ชาร์ตการส่งและรับข้อมูลทางพอร์ตอนุกรม 41

รูปที่ 3.10 แสดงส่วนของหน้าจอควบคุมหลัก 42

รูปที่ 3.11 แสดงส่วนของการจัดเก็บข้อมูลภายในฐานข้อมูล 43

รูปที่ 3.12 แสดงส่วนของการแสดงผลของข้อมูล 44

รูปที่ 3.13 แสดงโฟลว์ชาร์ตการทำงานของการแสดงผลในโหมดต่างๆ 45

รูปที่ 3.14 แสดงโฟลว์ชาร์ตการเลือกไฟล์การทำงาน 46

รูปที่ 4.1 แสดงวงจรภาคส่งอินฟราเรดเซนเซอร์ 47

รูปที่ 4.2 แสดงสัญญาณที่ขาอิมิตเตอร์ของทรานซิสเตอร์ เทียบกับกราวด์ 48

รูปที่ 4.3 แสดงสัญญาณที่ตกคร่อมไดโอดอินฟราเรด 48

รูปที่ 4. 4 แสดงวงจรภาครับอินฟราเรดเซนเซอร์ 49

รูปที่ 4.5 แสดงภาพถ่ายตัวอินฟราเรดเซนเซอร์ 50

รูปที่ 4.6 แสดงภาพถ่ายด้านในภายในตัวอินฟราเรดเซนเซอร์ 50

รูปที่ 4.7 แสดงด้านหน้าของชุดควบคุมไมโครคอนโทรลเลอร์ที่ต่อกับเซนเซอร์ 51

รูปที่ 4.8 แสดงภาพถ่ายชุดควบคุมไมโครคอนโทรลเลอร์ 52

รูปที่ 4.9 แสดงภาพถ่ายภายในของชุดควบคุมไมโครคอนโทรลเลอร์ 52

รูปที่ 4.10 แสดงภาพถ่ายด้านหลังของชุดควบคุมไมโครคอนโทรลเลอร์ 52

รูปที่ 4.11 แสดงผลการกดปุ่มเลือกไฟล์มัลติมีเดีย 53

รูปที่ 4.12 แสดงผลการกดปุ่มบันทึกไฟล์มัลติมีเดีย 53

รูปที่ 4.13 แสดงผลการบันทึกไฟล์มัลติมีเดียเรียบร้อยแล้ว 54

รูปที่ 4.14 แสดงผลในการจัดเก็บข้อมูลภายในฐานข้อมูล 54

รูปที่ 4.15 แสดงผลข้อมูลสถิติของไฟล์มัลติมีเดีย 55

รูปที่ 4.16 แสดงผลการกดปุ่มลบข้อมูลสถิติของไฟล์มัลติมีเดีย 55

รูปที่ 4.17 แสดงผลหลังจากตกลงลบข้อมูลสถิติของไฟล์มัลติมีเดีย 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 4.18 แสดงผลการกดปุ่มลบข้อมูลของไฟล์มัลติมีเดีย 56
รูปที่ 4.19 แสดงผลการกดปุ่มลบข้อมูลของไฟล์มัลติมีเดียทั้งหมด 57
รูปที่ 4.20 แสดงผลหลังจากตกลงลบข้อมูลของไฟล์มัลติมีเดียทั้งหมด 57
รูปที่ 4.21 แสดงผลของการแสดงผลของข้อมูลไฟล์มัลติมีเดีย 58
รูปที่ 4.22 แสดงภาพถ่ายชุดบริการข้อมูลอัตโนมัติ 59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงบิตพาริตี้ของข้อมูล 18

ตารางที่ 2.2 แสดงตำแหน่งและชื่อขาของ DB-9 และ DB-25 20

ตารางที่ 2.3 แสดงรูปแบบของ Connect String ของออบเจ็กต์ Connection 30

ตารางที่ 2.4 แสดงคุณสมบัติที่สำคัญของออบเจ็กต์ connection 31

ตารางที่ 2.5 แสดงคุณสมบัติที่สำคัญของออบเจ็กต์ Recordset 32

ตารางที่ 2.6 แสดงคุณสมบัติที่สำคัญของออบเจ็กต์ Command 33

ตารางที่ 2.7 แสดงคุณสมบัติที่สำคัญของออบเจ็กต์ Field 33

ตารางที่ 2.8 แสดงคุณสมบัติที่สำคัญของออบเจ็กต์ Parameter 34

ตารางที่ 4.1 แสดงผลการทดลองภาครับอินฟราเรด 49

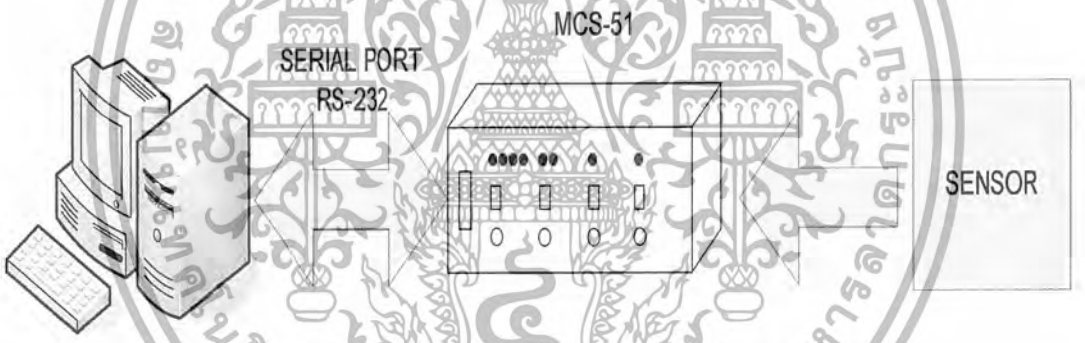
ตารางที่ 4.2 แสดงผลการทดลองชุดควบคุมไมโครคอนโทรลเลอร์ 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

โครงงานนี้เป็นการสร้างระบบที่นำมาใช้ประโยชน์ในการให้บริการข้อมูลโดยอัตโนมัติ โดยให้คอมพิวเตอร์เป็นตัวแสดงผลผ่านทางพอร์ตอนุกรมไปยังชุดไมโครคอนโทรลเลอร์ ที่ใช้เป็นตัวควบคุมในการให้บริการข้อมูล และมีอินฟราเรดเซนเซอร์เป็นเหมือนตัวสวิตซ์ในการเริ่มต้นการทำงาน โดยประกอบการทำงานอยู่ 2 ส่วน คือส่วนของฮาร์ดแวร์ที่ประกอบด้วยอินฟราเรดเซนเซอร์ และชุดไมโครคอนโทรลเลอร์ และส่วนของซอฟแวร์จะประกอบไปด้วยโปรแกรมวิชวลเบสิก เวอร์ชั่น 6.0 ที่ใช้เป็นตัวแสดงผลและใช้ในจัดเก็บข้อมูลการทำงาน และโปรแกรมควบคุมไมโครคอนโทรลเลอร์ ซึ่งเขียนโดยใช้ภาษาแอสเซมบลี

ในโครงงานนี้ได้ออกแบบให้สามารถใช้อินฟราเรดเซนเซอร์ได้ถึง 4 ตัว และสามารถเลือกไฟล์ในการแสดงผลการทำงานได้ถึง 4 ไฟล์ด้วยกัน



รูปที่ 1.1 แสดงภาพรวมของชุดบริการข้อมูลอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้องและแนวทางการทำโครงงาน

2.1 แอลอีดีอินฟราเรดกำลังสูง

เหตุผลที่เลือกใช้แสงในการสื่อสารทางโทรคมนาคมนั้นมีเหตุผลใหญ่ๆ 2 ข้อคือ

1) สามารถป้องกันการสอดแทรกต่างๆทั้งยังทำให้เกิดความปลอดภัยในการสื่อสาร

2) สัญญาณภายนอกต่างๆเช่นจากมอเตอร์ไม่สามารถที่จะรบกวนได้

สำหรับย่านความถี่แสงที่ใช้ส่งนี้จะใช้แสงในย่านอินฟราเรด

2.1.1 แอลอีดีกำลังสูงที่ให้แสงย่านอินฟราเรด

รอยต่อ P-N ของแอลอีดีเป็นแหล่งกำเนิดโฟตอนที่ดีมากแต่ก็ไม่สามารถปล่อยโฟตอนออกมาได้ทั้งหมด เช่น รอยต่อของแกลเลียมอาร์เซไนด์ (GaAs) ซึ่งให้แสงย่านใกล้อินฟราเรดทำให้เกิดโฟตอนประมาณ 88 เปอร์เซนต์เท่านั้น เนื่องจากโฟตอนที่ถูกปล่อยออกมาจากรอยต่อส่วนหนึ่งถูกสกัดกั้นโดยกรอบของชิ้นสาร

2.1.2 คุณสมบัติของแอลอีดีอินฟราเรด

แรงดันตกคร่อมที่รอยต่อ P-N ของไดโอด ต้องมีค่ามากกว่าแรงดันเทรชโฮลด์จึงจะสามารถทำให้ไดโอดนำกระแสได้ สำหรับซิลิกอนไดโอดแรงดันทำงานมีค่าประมาณ 0.6 โวลท์ ส่วนแอลอีดีที่ให้แสงในย่านที่ตามองเห็นได้ถ้าทำจากสาร Gap ซึ่งให้แสงสีเขียว จะมีค่าแรงดันประมาณ 2.1 ถึง 2.8 โวลท์ ถ้าเป็นแอลอีดีที่ทำจาก AlGaAsให้แสงสีแดงมีแรงดันทำงาน 1.75 ถึง 2.5 โวลท์ ส่วนแอลอีดีที่ให้แสงใกล้ย่านอินฟราเรดทำจากสาร GaAs มีแรงดันทำงาน 1.5 โวลท์ โดยให้แสงที่มีความยาวคลื่น 940 นาโนเมตร และถ้าทำจาก AlGaAs จะให้แสงความยาวคลื่น 880 นาโนเมตร ที่แรงดัน 1.75 โวลท์ พลังงานที่ได้จากการเปล่งแสงของแอลอีดีหาได้จากกระแสไบอัสตรงของไดโอด และต้องระวังไม่ให้กระแสส่วนนี้มีค่าสูงจนเกิดความร้อนอันตรายต่อชิ้นไดโอด

ซิลิกอนแอลอีดีที่ทำจาก GaAs ให้กำลังงานประมาณ 5 มิลลิวัตต์ที่กระแสไบอัสตรงค่าเดียวกัน ส่วนข้อดีของแอลอีดีชนิด AlGaAs คือมี Rise time และ Fall time ที่เร็วกว่าคือประมาณ 0.5 ไมโครวินาที ในขณะที่ GaAs ซิลิกอนไดโอดมีค่า 1.5 ไมโครวินาที

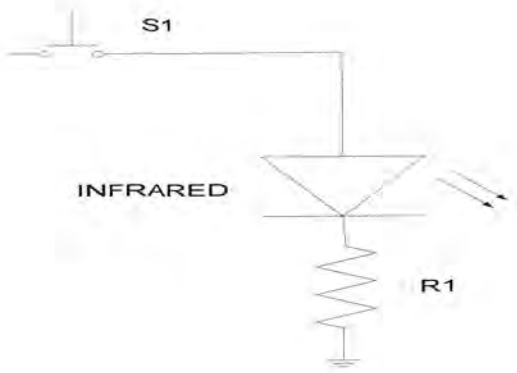
ข้อดีอีกประการหนึ่งคือ การเปล่งแสงของแอลอีดีที่มีความยาวคลื่น 880 นาโนเมตร (AlGaAs) จะใกล้เคียงกับความยาวคลื่นที่ซิลิกอนโฟโตทรานซิสเตอร์มีความไวสูงสุด จึงเป็นการเหมาะสมที่จะใช้แอลอีดีที่มีความยาวคลื่น 880 นาโนเมตรแทนแอลอีดีที่มีความยาวคลื่น 940 นาโนเมตร

นอกจากนั้นแล้วแอลอีดีที่ให้ความยาวคลื่น 940 นาโนเมตร จึงสามารถนำไปใช้ในการตรวจจับไอน้ำในอากาศได้

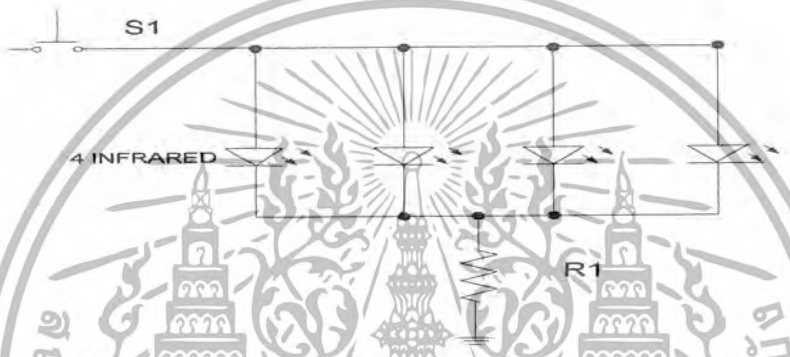
2.1.3 ระบบอินฟราเรด

การใช้แสงอินฟราเรดเป็นสัญญาณควบคุม สามารถตัดปัญหาการรบกวนของแสงภายนอกอื่นๆ ลงไปได้ ยิ่งกว่านั้นวงจรใช้งานของระบบอินฟราเรดยังเป็นวงจรที่ง่ายและไม่ซับซ้อนมีความเชื่อถือได้สูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) การไบอัสอินฟราเรดธรรมดา



ข) แสดงการไบอัสอินฟราเรดเพื่อเพิ่มปริมาณความเข้มของแสง

รูปที่ 2.1 วงจรสร้างสัญญาณอินฟราเรดอย่างง่าย

การส่งสัญญาณแสงอินฟราเรดสามารถกระทำได้ด้วยวงจรง่ายๆดังรูปที่ 2.1 (ก) ซึ่งประกอบด้วยแอลอีดีที่เปล่งแสงในย่านอินฟราเรดต่อเข้ากับแหล่งจ่ายไฟโดยมีความต้านทาน R1 ทำหน้าที่จำกัดกระแสสัญญาณแสงที่ส่งออกมา โดยแอลอีดีเพียงตัวเดียวจะเหมาะกับการใช้งานในระยะใกล้ไม่กี่เมตรเท่านั้น การเพิ่มกำลังส่งของอินฟราเรดให้ไปได้ไกลขึ้นทำได้โดยใช้แอลอีดีหลายตัวต่อขนานกันดังรูป 2.1 (ข) ปัจจุบันมีแอลอีดีย่านอินฟราเรดรุ่นใหม่ที่ให้กำลังส่ง หรือความเข้มแสงสูงพอช่วยให้สัญญาณไปได้ไกลกว่าเดิม

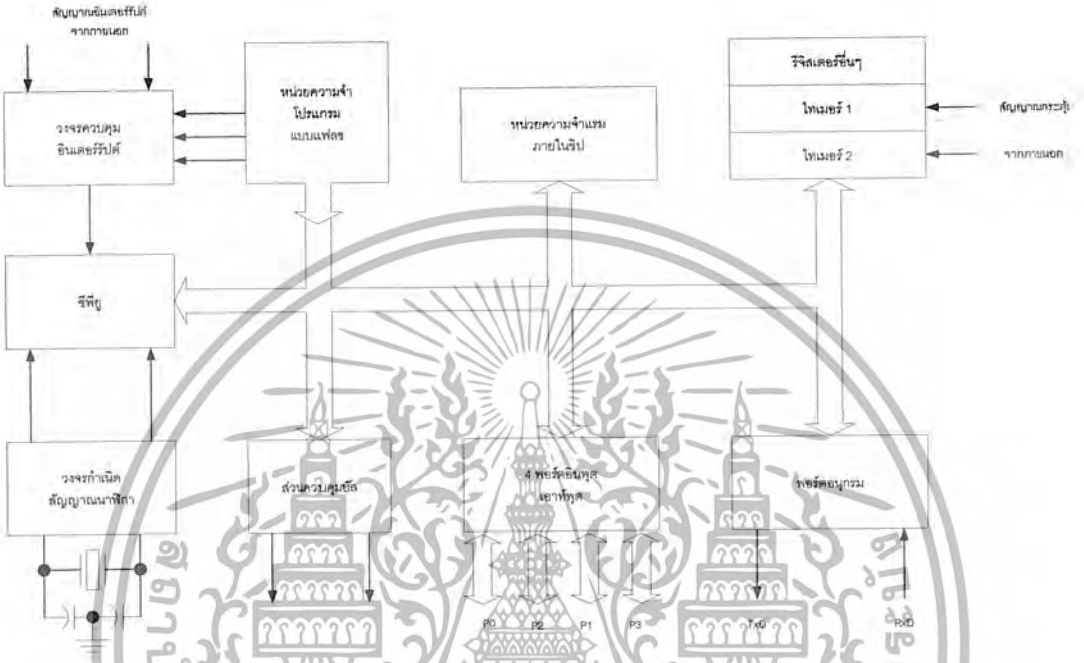
2.2 ไมโครคอนโทรลเลอร์ MCS-51

2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 อนุกรม AT89C51

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง
- หน่วยข้อมูลความจำพื้นฐานเป็นหน่วยความจำแบบแรม
- ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาท์พุต
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไทเมอร์/เคาน์เตอร์ขนาด 16 บิตอย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป



รูปที่ 2.2 แสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89C51

ในรูปที่ 2.2 เป็นโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89C51 โครงสร้างจะเหมือนกับไมโครคอนโทรลเลอร์ตระกูล MCS-51 พื้นฐาน หากแต่แตกต่างกันเฉพาะหน่วยความจำโปรแกรมแบบแฟลชที่เพิ่มเติมเข้ามา

2.2.2 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ทุกเบอร์จะมีสถาปัตยกรรมและขาใช้งานพื้นฐานเหมือนกันโดยมีรายละเอียดขั้นต้น ดังนี้

ขา Vcc (ขา 40)ใช้สำหรับต่อไฟเลี้ยง +5V

ขา GND (ขา 20)เป็นขากราวด์ สำหรับต่อกราวด์ของระบบ

ขาพอร์ต 0 (P0.0-P0.7) คือขา 32-39 มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล"1"ไปยังแต่ละบิตของพอร์ทที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ทนั้นมีสถานะปล่อยลอย(float) จึงมีอินพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วยเพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาพอร์ต 1 (P1.0-P1.7) มี 8 ขา คือขา 1-8 แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย

ขาพอร์ต 2 (P2.0-P2.7) มี 8 ขา คือขา 21-28 แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ทนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

ขาพอร์ต 3 (P3.0-P3.7) มี 8 ขา คือขา 1-8 แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ท 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นต้นต่อไปนี้

P3.0 หรือ ขา RxD ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม

P3.1 หรือ ขา TxD ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม

P3.2 หรือ ขา $\overline{\text{INT0}}$ ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0

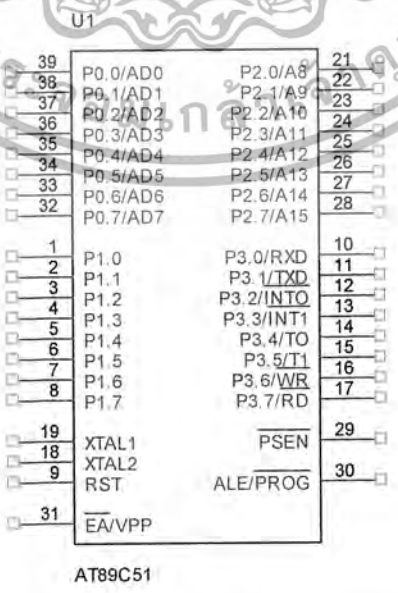
P3.3 หรือ ขา $\overline{\text{INT1}}$ ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1

P3.4 หรือ ขา T0 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0

P3.5 หรือ ขา T1 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1

P3.6 หรือ ขา $\overline{\text{WR}}$ ใช้เป็นขาสัญญาณในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

P3.7 หรือ ขา $\overline{\text{RD}}$ ใช้เป็นขาสัญญาณในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก



รูปที่ 2.3 แสดงการจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51ในอนุกรม AT89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขารีเซต(Reset) หรือขา 9 ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซตสถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชีนไซเกิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างปกติ

ขา $\overline{\text{ALE}}$ / PROG (Address Latch Enable/Program pulse input) หรือขา 30 เป็นขาที่ใช้ในการควบคุมการแลตช์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีพรอม

ขา $\overline{\text{PSEN}}$ (Program Store Enable) หรือ ขา 29 ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้ง ในแต่ละแมชีนไซเคิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใดๆออกมา

ขา $\overline{\text{EA}}$ / Vpp (External Access enable/Programming voltage input) หรือ ขา 31 ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้เป็น "1" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำ ภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ที่ขานี้ยังใช้เป็นขาอินพุต สำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์

ขา XTAL1 หรือ ขา 19 เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

ขา XTAL2 หรือ ขา 18 เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

2.2.3 โครงสร้างและการทำงานของพอร์ต

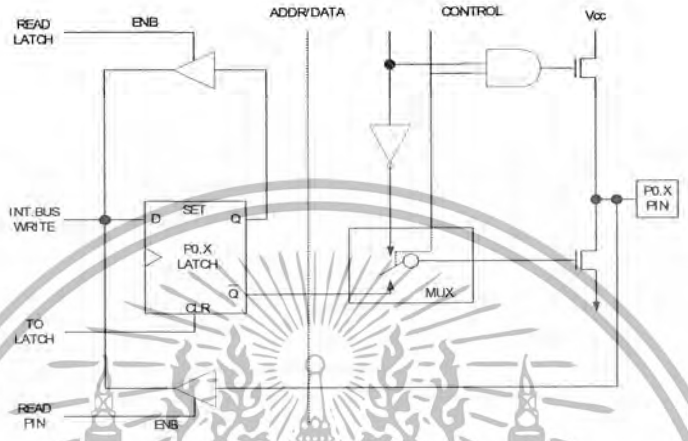
ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีพอร์ตให้ใช้งานทั้งสิ้น 4 พอร์ตคือ พอร์ต 0 ถึง พอร์ต 3 แต่ละพอร์ตมีขนาด 8 บิต เป็นพอร์ตแบบ 2 ทิศทาง กล่าวคือ สามารถเป็นได้ทั้งอินพุตสำหรับรับสัญญาณข้อมูลเข้าและเอาต์พุตสำหรับส่งสัญญาณข้อมูลออก ทุกพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีวงแลตช์และวงจรขับตลอดจนบัฟเฟอร์อินพุต

ที่พอร์ต 0 และพอร์ต 2 จะใช้งานเป็นพอร์ตอินพุตและเอาต์พุตสำหรับงานทั่วไป และใช้ในการติดต่อกับหน่วยความจำภายนอก สำหรับพอร์ต 3 ทั้งพอร์ตและพอร์ต 1 บางขาขอกจากจะใช้เป็นขาพอร์ตอินพุตเอาต์พุตตามปกติแล้ว ยังสามารถใช้งานในหน้าที่พิเศษได้อีก

วงจรภายในของแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช โดยในรูปที่ 2.4 เป็นวงจรของพอร์ต 0 วงจรแลตช์ของแต่ละบิตในแต่ละพอร์ตก็คือวงจรดีฟลิฟลอปนั่นเอง การอ่านค่าสถานะของพอร์ตและสถานะของวงจรแลตช์สามารถกระทำได้อย่างอิสระด้วยสัญญาณที่แยกจากกัน นั่นคือสัญญาณอ่านข้อมูลจากขาพอร์ต และสัญญาณอ่านข้อมูลจากวงจรแลตช์ ส่วนการเขียนข้อมูลมายัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

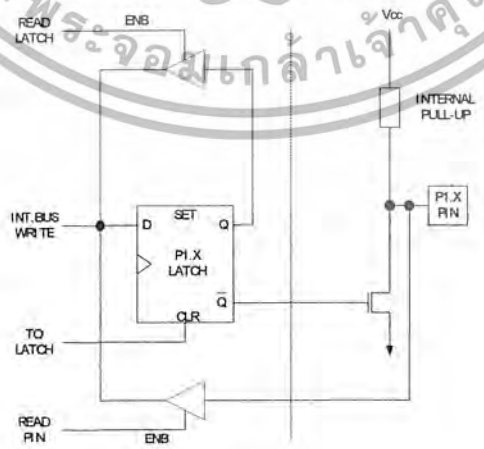
พอร์ตต้องส่งสัญญาณมายังขา CLK ของดีฟลิฟลอปในขณะที่ข้อมูลจะผ่านมาทางขาบัสข้อมูลภายในเข้าสู่ขา D ของดีฟลิฟลอป ที่พอร์ตนี้มีวงจรมัลติเพล็กซ์สำหรับกำหนดลักษณะการทำงานของพอร์ตว่าต้องการใช้งานเป็นขาพอร์ตอินพุตเอาต์พุตปกติ หรือใช้ในการติดต่อกับหน่วยความจำภายนอกไมโครคอนโทรลเลอร์



รูปที่ 2.4 แสดงโครงสร้างของพอร์ต 0 (บิต)

เนื่องจากที่ขาพอร์ต 0 ไม่มีวงจรพูลอัปภายใน หากมีการนำพอร์ต 0 ไปใช้งานเป็นพอร์ตอินพุตจะต้องต่อตัวต้านทานพูลอัปภายนอกเข้าที่ขาพอร์ต 0 ทุกขาด้วย

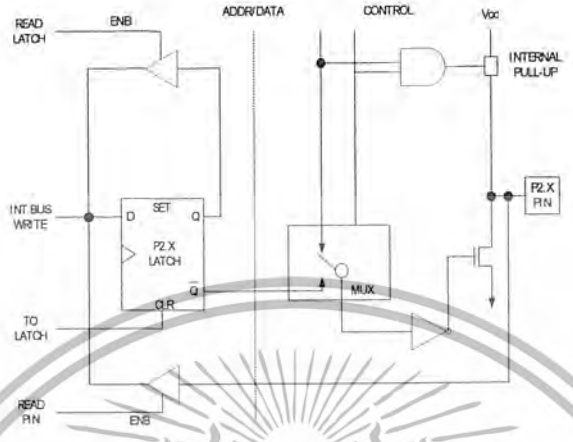
ในรูปที่ 2.5 เป็นวงจรของพอร์ต 1 ซึ่งมีลักษณะโดยทั่วไปคล้ายกับพอร์ต 0 หากแต่ไม่มีวงจรมัลติมัลติเพล็กซ์ เนื่องจากพอร์ตนี้จะไม่ใช้ติดต่อกับหน่วยความจำภายนอก แต่จะมีวงจรพูลอัปภายในที่แต่ละบิตของพอร์ตนี้แทน



รูปที่ 2.5 แสดงโครงสร้างของพอร์ต 1 (บิต)

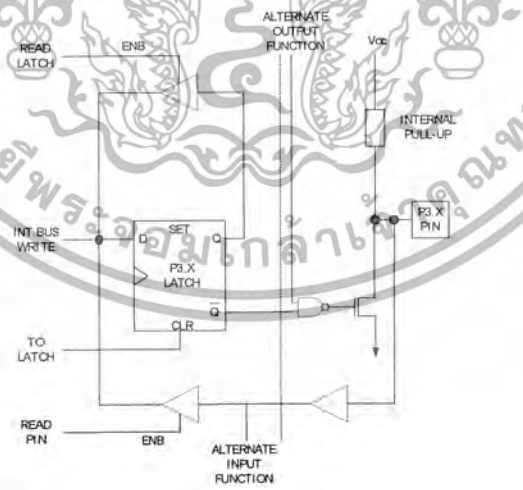
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.6 เป็นวงจรภายในของพอร์ต 2 จะคล้ายกับพอร์ต 0 มาก ต่างกันเพียงมีวงจรพูลอัปเพิ่มเติมเข้ามา



รูปที่ 2.6 แสดงโครงสร้างของพอร์ต 2 (บิต)

ส่วนในรูปที่ 2.7 เป็นวงจรภายในของพอร์ต 3 จะเห็นได้ว่าคล้ายกับพอร์ต 1 มีการเพิ่มเติมวงจรบัฟเฟอร์และวงจรอินพุตเอาท์พุตเมื่อทำงานในฟังก์ชั่นพิเศษเข้ามา เนื่องจากพอร์ต 3 สามารถนำไปใช้งานในหน้าที่พิเศษได้ทุกขา



รูปที่ 2.7 แสดงโครงสร้างของพอร์ต 3 (บิต)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 การใช้งานเป็นพอร์ตอินพุต

เนื่องจากพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชสามารถเป็นได้ทั้งอินพุตและเอาต์พุต ดังนั้นจึงมีความจำเป็นอย่างยิ่งต้องมีความเข้าใจถึง การกำหนดลักษณะการทำงานให้แก่พอร์ตของไมโครคอนโทรลเลอร์ MCS-51แบบแฟลช

ในการกำหนดให้เป็นพอร์ตอินพุต ต้องเริ่มต้นด้วยการเขียนข้อมูล "1" มาที่แต่ละบิตของพอร์ตที่ต้องการใช้งานเป็นอินพุต เพื่อหยุดการทำงานของแฟตที่ใช้ในการขับสัญญาณเอาต์พุตของบิตนั้นๆทำให้ขาสัญญาณของพอร์ตเชื่อมต่อเข้ากับวงจรพูลอัปภายในโดยตรง ส่งผลให้ขาพอร์ตนั้นมีลอจิกเป็น "1" สามารถรับสัญญาณลอจิก "0" จากอุปกรณ์ภายนอกได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาแล้วเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ต แล้วรอให้ซีพียูมาอ่านค่าเข้าไป เมื่อเป็นเช่นนี้อุปกรณ์ภายนอกที่เชื่อมต่อกับพอร์ตอินพุตของไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชควรกำหนดให้ทำงานในสภาวะลอจิก "0"จะดีและสะดวกที่สุด

2.2.5 การใช้งานเป็นพอร์ตเอาท์พุต

โดยปกติแล้วขาพอร์ตจะกำหนดให้มีลักษณะเป็นเอาท์พุตอยู่แล้ว ดังนั้นจึงสามารถส่งข้อมูลออกไปได้อย่างได้ง่ายดายและตรงไปตรงมา กล่าวคือ เมื่อต้องการส่งข้อมูล "0"ออกไปทางเอาท์พุตก็ให้เขียนข้อมูล "0" ไปยังวงจรแลตช์ ซึ่งก็จะส่งต่อไปขับเฟส ทำให้เฟตทำงานที่ขาพอร์ตที่กำหนดให้ทำงานก็จะเกิดลอจิก "0" ขึ้น ในทางตรงกันข้ามหากต้องการส่งข้อมูล "1" ออกไป ก็ให้เขียนข้อมูล "1" ไปยังวงจรแลตช์ วงจรขับก็จะหยุดทำงานทำให้ที่ขาพอร์ตเชื่อมต่อกับวงจรพูลอัปภาย ในเกิดเป็นลอจิก "1" ที่ขาพอร์ตนั้น ซึ่งคล้ายกับการกำหนดให้เป็นขาอินพุตมาก เพียงแต่แตกต่างกันที่กระบวนการในการเคลื่อนย้ายข้อมูล โดยถ้าเป็นอินพุตจะมีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาท์พุตจะไม่มีการอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด เว้นแต่ในกรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาต์พุต

เมื่อใช้งานเป็นพอร์ตเอาท์พุตแต่ละขาของแต่ละพอร์ต มีความสามารถในการจ่ายกระแสได้สูงสุด 10 มิลลิแอมป์ และทุกขารวมกันในแต่ละพอร์ต(ทั้ง 8 บิต) สูงสุด 26 มิลลิแอมป์ สำหรับพอร์ต 0 และ 15 มิลลิแอมป์ สำหรับพอร์ต 1 ถึง พอร์ต 3 ในกรณีที่ใช้งานทุกพอร์ตเอาท์พุตจะสามารถจ่ายกระแสได้รวมกันสูงสุด 71 มิลลิแอมป์ ดังนั้นในการใช้งานเป็นพอร์ตเอาท์พุต เพื่อไม่ให้เกิดปัญหาเกี่ยวกับความสามารถในการจ่ายกระแสจึงต่อวงจรบัฟเฟอร์ทางเอาท์พุตเพื่อช่วยในการขับกระแสอีกทางหนึ่ง

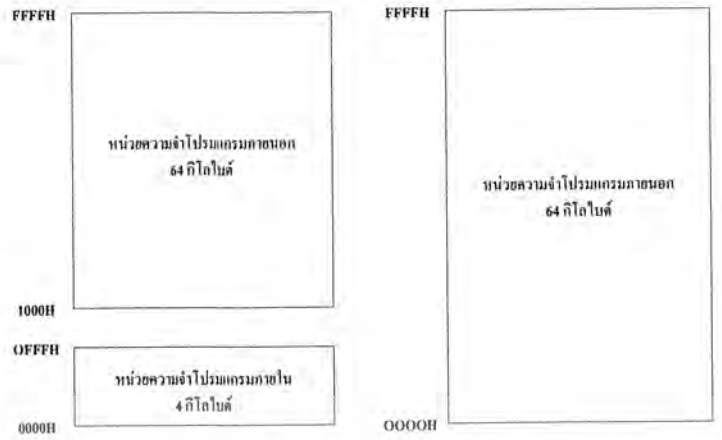
2.2.6 หน่วยความจำของไมโครคอนโทรลเลอร์ แบบแฟลช

ในไมโครคอนโทรลเลอร์ แบบแฟลชมีหน่วยความจำภายในหลักๆอยู่ 2 ส่วนคือ หน่วยความจำโปรแกรมและหน่วยความจำข้อมูล ซึ่งก็มีขนาดและการจัดสรรแตกต่างกันไป

2.2.6.1.หน่วยความจำโปรแกรม (Program memory)

การจัดหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช เบอร์ AT89C51 ที่แสดงดังรูปที่ 2.8 จะเห็นได้ว่าสามารถติดต่อหน่วยความจำโปรมแกรมได้สูงสุด 64 กิโลไบต์ โดยสามารถเลือกใช้หน่วยความจำโปรแกรมภายในอย่างเดียว หรือรวมกับภายนอก หรือเลือกใช้หน่วยความจำภายนอกอย่างเดียวก็ได้ โดยภายใน AT89C51 จะมีหน่วยความจำโปรแกรมภายใน 4 กิโลไบต์ สามารถติดต่อกับหน่วยความจำภายนอกได้ 60 กิโลไบต์

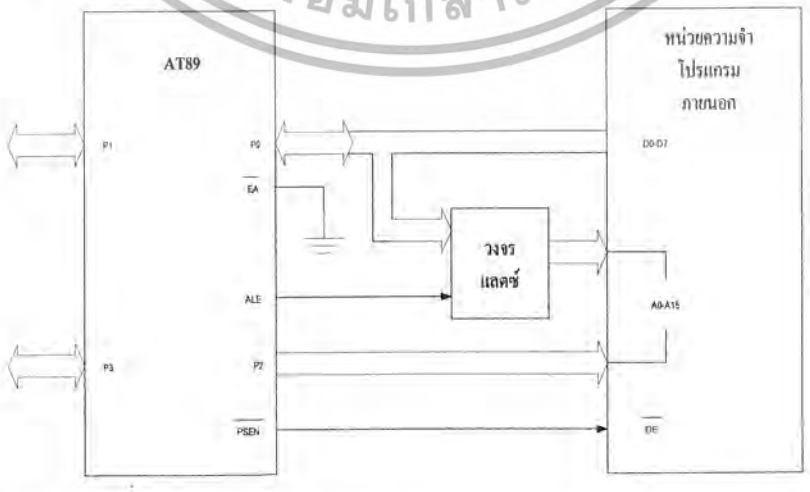
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงการจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์เบอร์ AT89C51

หน่วยความจำโปรแกรม ใช้เก็บข้อมูลของโปรแกรม ควบคุมการทำงานของไมโครคอนโทรลเลอร์หรือที่เรียกว่า โปรแกรมมอนิเตอร์ (monitor program) หากใช้หน่วยความจำภายนอกมักจะบรรจุอยู่ในหน่วยความจำชนิดอีพรอม หรือ EPROM ซึ่งสามารถอ่านได้เพียงอย่างเดียว โดยมีแอดเดรสเริ่มต้นที่ 0000H เมื่อซีพียูได้รับการรีเซตให้เริ่มต้นการทำงาน จะต้องมาเริ่มต้นที่แอดเดรส 0000H นี้เสมอ ในพื้นที่ของหน่วยความจำโปรแกรมไม่ว่าใช้งานจากภายใน หรือภายนอกก็ตามต้องมีการสงวนพื้นที่บางตำแหน่งเอาไว้สำหรับการบริการอินเตอร์รัปต์ 6 ประเภท ประเภทละ 8 ไบต์ ประกอบด้วย

พื้นที่สำหรับบริการอินเตอร์รัปต์ 0 จากภายนอก กำหนดไว้ที่แอดเดรส 0003H
พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 0 กำหนดไว้ที่แอดเดรส 000BH
พื้นที่สำหรับบริการอินเตอร์รัปต์ 1 จากภายนอก กำหนดไว้ที่แอดเดรส 0013H
พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 1 กำหนดไว้ที่แอดเดรส 001BH
พื้นที่สำหรับบริการอินเตอร์รัปต์ของการสื่อสารอนุกรม กำหนดไว้ที่แอดเดรส 0023H
พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 2 กำหนดไว้ที่แอดเดรส 002BH



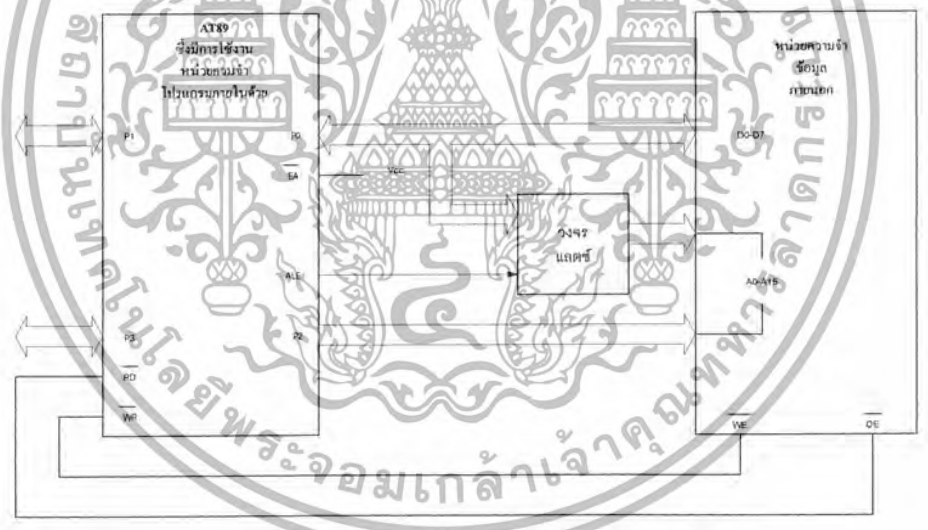
รูปที่ 2.9 แสดงการเชื่อมต่อหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์ แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ใช้ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ที่มีหน่วยความจำโปรแกรมภายใน แต่ต้องการติดต่อกับหน่วยความจำโปรแกรมภายนอกด้วย สามารถทำได้โดยต้องกำหนดแอดเดรสของหน่วยความจำโปรแกรม ให้ต่อจากแอดเดรสสุดท้ายของหน่วยความจำโปรแกรมภายใน ของไมโครคอนโทรลเลอร์ ซึ่งไมโครคอนโทรลเลอร์ AT89C51 มีหน่วยความจำโปรแกรมขนาด 4 กิโลไบต์ มีแอดเดรสอยู่ระหว่าง 0000H - 0FFFH เมื่อต่อหน่วยความจำโปรแกรมภายนอก ต้องกำหนดให้แอดเดรสอยู่ในช่วง 1000H-FFFFH

2.2.6.2. หน่วยความจำข้อมูล (Data memory)

มีด้วยกัน 2 แบบคือ หน่วยความจำข้อมูลภายนอกและภายใน โดยไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช สามารถติดต่อกับหน่วยความจำข้อมูลภายนอกได้สูงสุด 64 กิโลไบต์ โดยการใช้คำสั่ง MOVX ในการติดต่อกับหน่วยความจำข้อมูลภายนอก การติดต่อหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์ แบบแฟลช แสดงดังในรูปที่ 2.10 จะเห็นได้ว่ามีลักษณะคล้ายกับการติดต่อกับหน่วยความจำโปรแกรมนอก แตกต่างกันที่มีสัญญาณที่ใช้สำหรับการอ่านและเขียนหน่วยความจำข้อมูลภายนอก



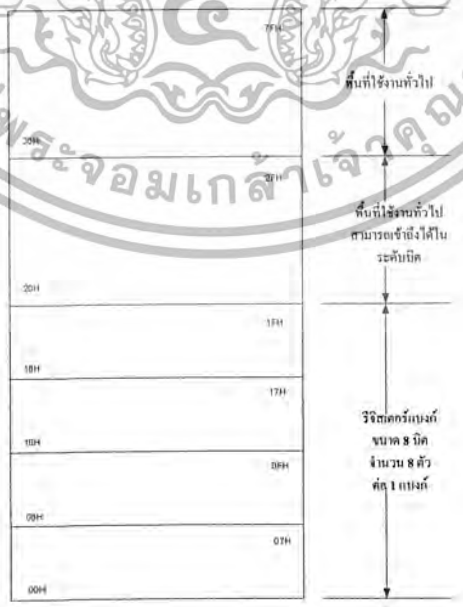
รูปที่ 2.10 แสดงการเชื่อมต่อหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์ แบบแฟลช

ไมโครคอนโทรลเลอร์ แบบแฟลช จะมีหน่วยความจำข้อมูลภายในเป็นแบบแรม โดยในเบอร์ AT89C51 มีหน่วยความจำข้อมูลภายในขนาด 128 ไบต์ สำหรับการจัดสรรหน่วยความจำข้อมูลภายในแบ่งเป็น 3 ส่วนคือ หน่วยความจำข้อมูลส่วนล่าง(lower), ส่วนบน(upper) และรีจิสเตอร์ฟังก์ชั่นพิเศษแต่ละส่วนมีขนาด 128 ไบต์ ดังแสดงการจัดสรรในรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.11 แสดงการจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในไมโครคอนโทรลเลอร์ แบบแฟลช

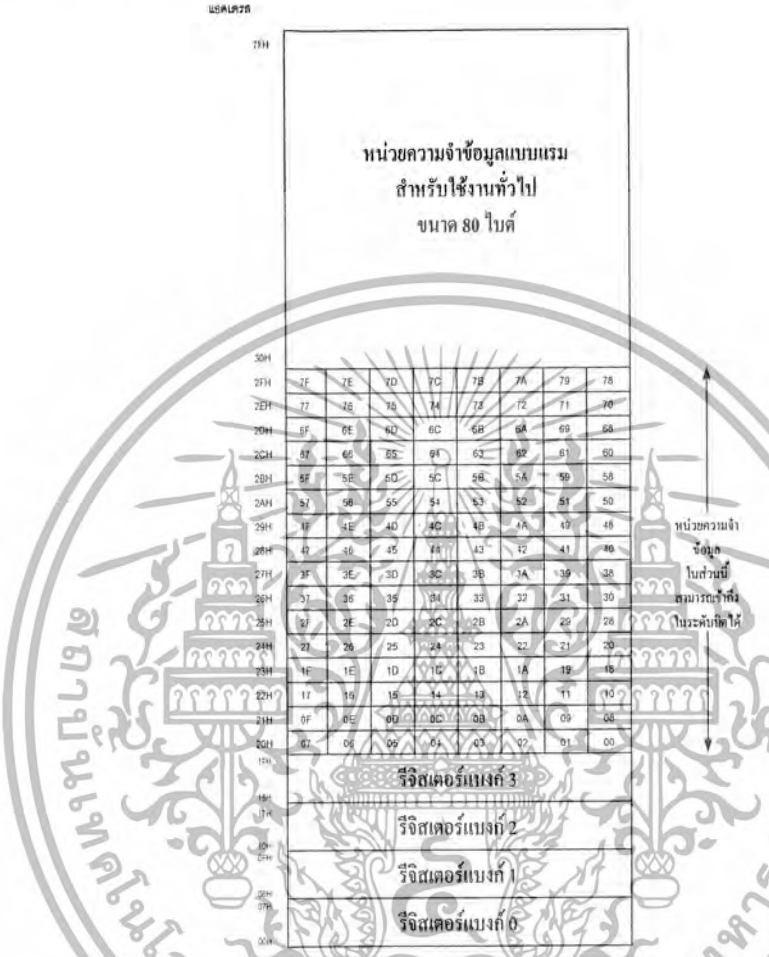
ขนาดของหน่วยความจำของไมโครคอนโทรลเลอร์แบบแฟลชโดยแท้จริงแล้วมีเพียง 256 ไบต์ แต่ด้วยการจัดการเข้าถึงที่แตกต่างกันจึงดูเสมือนว่าจะมีหน่วยความจำข้อมูลภายในสูงถึง 384 ไบต์ โดยในหน่วยความจำข้อมูลส่วนล่างขนาด 128 ไบต์ มีแอดเดรสอยู่ที่ 00H-7FHสามารถเข้าถึงได้โดยตรงและโดยอ้อม สำหรับหน่วยความจำข้อมูลส่วนบนมีขนาด 128 ไบต์เช่นกันมีแอดเดรสอยู่ที่ 80H-FFHสามารถเข้าถึงแบบโดยอ้อมเท่านั้น ในขณะที่รีจิสเตอร์ SFR มีแอดเดรสอยู่ที่ 80H-FFH เช่นเดียวกับหน่วยความจำข้อมูลส่วนบน แต่สำหรับรีจิสเตอร์ SFR ใช้การเข้าถึงแบบโดยตรง การจัดสรรหน่วยความจำข้อมูลส่วนล่างดังรูปที่ 2.12 หน่วยความจำ 32 ไบต์ต่ำสุดที่แอดเดรส 00H-1FH แบ่งเป็น 4 กลุ่ม เรียกว่า 4 แบงก์ (bank) แต่ละแบงก์มีรีจิสเตอร์ 8 ตัวคือ R0-R7 การติดต่อกับหน่วยความจำในแบงก์ใดให้กำหนดที่รีจิสเตอร์ PSW หน่วยความจำข้อมูล 16 ไบต์ถัดมาที่แอดเดรส 20H-2FH เป็นพื้นที่สำหรับใช้งานทั่วไปสามารถเข้าถึงได้ในระดับบิต และหน่วยความจำข้อมูลที่เหลือ 80 ไบต์ จะต้องแบ่งส่วนหนึ่งสำรองไว้เป็นพื้นที่ของสแต็ก(stack:ที่พักข้อมูลชั่วคราวในกรณีที่ซีพียูมีการกระโดดไปทำงานในโปรแกรมย่อย) การเข้าถึงหน่วยความจำในส่วนนี้ต้องใช้การเข้าถึงในระดับไบต์



รูปที่ 2.12 แสดงการจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในส่วนล่างของ
ไมโครคอนโทรลเลอร์ แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับโครงสร้างของหน่วยความจำข้อมูลส่วนบนแสดงดังรูปที่ ซึ่งจะมีลักษณะที่คล้ายกับหน่วยความจำข้อมูลส่วนล่าง หากแต่ใน 80 ไบต์บนไม่จำเป็นต้องสำรองไว้สำหรับสแต็ก และต้องใช้การเข้าถึงในลักษณะโดยอ้อมเท่านั้น



รูปที่ 2.13 แสดงการจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในส่วนบนไมโครคอนโทรลเลอร์ แบบแฟลช

2.2.7 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์ 1 ชุด (วงจรสื่อสารแบบฟูลดูเพล็กซ์ หมายถึง วงจรสื่อสารที่สามารถทำการรับและส่งข้อมูล ในลักษณะ 2 ทิศทางได้ในเวลาเดียวกัน) โดยใช้ขาสัญญาณของพอร์ต 3 คือ ขา P3.0 เป็นขารับข้อมูลเข้าหรือ RxD และขา P3.1 เป็นขาส่งข้อมูลเข้าหรือ TxD โดยวงจรสื่อสารข้อมูลแบบอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเป็นแบบอะซิงโครนัส ปกติแล้วพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 จะใช้ในการติดต่อสื่อสารกับพอร์ตอนุกรมของคอมพิวเตอร์ โดยใช้มาตาฐานอาร์เอสสองสามสอง แต่ในปัจจุบันสามารถติดต่อกันในมาตาฐาน อาร์เอสสี่สองสอง หรือ อาร์เอสสี่แปดห้า ได้แล้ว โดยใช้ไอซีพิเศษที่ทำหน้าที่ในการแปลงสัญญาณการสื่อสารดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.7.1 รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของพอร์ตอนุกรมใน MCS-51

ในการทำงานของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ มีรีจิสเตอร์ที่เกี่ยวข้องอยู่ 2 ตัว ดังนี้

- รีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรมหรือ SBUF (Serial data buffer register)

มีแอดเดรสอยู่ที่ 99H ในพื้นที่ของรีจิสเตอร์ฟังก์ชั่นพิเศษหรือ SFR มีขนาด 8 บิต แบ่งเป็น 2 ส่วนคือ รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (transmit buffer register) และรับข้อมูล(receive buffer register) เมื่อมีการเขียนข้อมูลมายังรีจิสเตอร์ SBUF ข้อมูลนั้นจะถูกส่งต่อไปยังบัฟเฟอร์สำหรับส่งข้อมูลเพื่อส่งออกจากไมโครคอนโทรลเลอร์ผ่านทางขา TxD หรือ ขา P3.1 ในกรณีที่มีการอ่านข้อมูลจากรีจิสเตอร์ SBUF ข้อมูลนั้นจะถูกส่งต่อไปยังบัฟเฟอร์สำหรับส่งข้อมูล เพื่อส่งออกจากไมโครคอนโทรลเลอร์ผ่านทางขา RxD หรือขา P3.0 ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

- รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรมหรือSCON (Serial port Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 98H ในพื้นที่ของรีจิสเตอร์ SFR สามารถเข้าถึงได้ในระดับบิต มีรายละเอียดการทำงานดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
SMO	SM1	SM2	REN	TB8	RB8	TI	RI

SM0-SM1 (Serial port mode bit 0-1) : ใช้ในการเลือกโหมดการทำงานของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 ดังมีรายละเอียดต่อไปนี้

SM2 : ใช้ในการเอ็นเอเบิลการสื่อสารในแบบมัลติโปรเซสเซอร์ (Multiprocessor) ในการทำงานของโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ ถ้าบิตนี้เป็น "1" บิต RI จะไม่แอกตีฟถ้าบิตที่ 9 ที่รับเข้ามาเป็น "0" (ข้อมูลบิตที่ 9 เก็บไว้ที่บิต RB8) ในการทำงานโหมด 1 ถ้าบิตนี้เซตบิต RI จะไม่แอกตีฟถ้ายังไม่ได้รับบิตหยุด ส่วนในโหมด 0 บิตนี้ไม่มีการใช้งาน

REN (Enable serial reception) : ใช้ในการเอ็นเอเบิลการรับข้อมูลของพอร์ตอนุกรม ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ ถ้าต้องการให้มีการรับข้อมูลต้องเซตบิตนี้ให้เป็น "1"

TB8 : ใช้สำหรับเก็บข้อมูลบิตที่ 9 ที่ต้องการส่งออกไปในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ เซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

RB8 : ใช้สำหรับเก็บข้อมูลบิตที่ 9 ที่เข้ามาในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ แต่ถ้าหากพอร์ตอนุกรมทำงานอยู่ในโหมด 1 และบิต SM2 เป็น "0"ข้อมูลที่บิต RB8 คือข้อมูลของบิตหยุด (Stop bit) สำหรับในการทำงานโหมด 0 บิตนี้จะไม่ทำงาน บิต RB8 นี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

TI (Transmit Interrupt flag) : ใช้ในการแสดงการเกิดอินเตอร์รัปต์เมื่อมีการส่งข้อมูลออกจากพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการส่งข้อมูลบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ 8 ไปเรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะเซตเมื่อมีการเริ่มต้นส่งบิตหยุดออกไป การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

RI (Receive Interrupt flag) : ใช้ในการแสดงการเกิดอินเตอร์รัปต์เมื่อมีการรับข้อมูลออกจากพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการส่งข้อมูลบิตที่ 8 ไปเรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะเซตเมื่อมีการเริ่มต้นส่งบิตหยุดของข้อมูลอนุกรมไปได้ครึ่งทางแล้ว ยกเว้นในกรณีที่บิต SM2 มีการเซต บิตนี้จะเซตได้ก็ต่อเมื่อการรับบิตหยุดหรือบิตที่ 9 เกิดขึ้นอย่างสมบูรณ์แล้ว การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

2.2.7.2 การเขียนหรือส่งข้อมูลออกจากพอร์ตอนุกรม : ข้อมูลที่ต้องการส่งออกทุกค่าต้องนำไปเก็บไว้รีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรม ซึ่งก็คือรีจิสเตอร์ SBUF ดังตัวอย่าง

```
MOV SBUF,# 'A'
```

จากคำสั่งข้างต้นเป็นการส่งข้อมูลของตัวอักษร A ออกไปยังพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ อย่างไรก็ตามก่อนทำการส่งข้อมูลทุกครั้ง ต้องแน่ใจว่าบิต TI เคลียร์ หรือมีค่าเป็น "0" และเมื่อทำการส่งข้อมูลเรียบร้อยแล้ว ก็จะเกิดการเซตบิต TI เพื่อแจ้งให้ทราบ ดังตัวอย่างโปรแกรมต่อไปนี้

```
CLR   TI          ; เคลียร์บิต TI เพื่อเตรียมการส่งข้อมูลออก
MOV   SBUF,# 'A'  ; ส่งข้อมูลของตัวอักษร A ไปยังพอร์ตอนุกรม
JNB   TI, $       ; รอการเซตของบิต TI เพื่อแจ้งการส่งข้อมูลที่เสร็จสมบูรณ์
```

2.2.7.3 การอ่านหรือรับข้อมูลออกจากพอร์ตอนุกรม : การรับข้อมูลจากพอร์ตอนุกรมสามารถกระทำได้ง่ายมาก เพียงตรวจสอบว่าบิต RI เกิดการเซตขึ้นหรือไม่ ถ้าพบว่ามีการเซตเกิดขึ้นแล้วให้ทำการอ่านค่าจากรีจิสเตอร์ SBUF โดยต้องทำการโอนย้ายข้อมูลผ่านทางแอกคิวมูเลเตอร์หรือรีจิสเตอร์ A ดังตัวอย่าง

```
CLR RI              ; เคลียร์บิต TI เพื่อเตรียมการส่งข้อมูลออก
JNB  RI , $         ; รอคอยการเซตของบิต RI อันเป็นการแจ้งให้ทราบว่า การรับ
                      ข้อมูลเสร็จ
                    ; สมบูรณ์และมีข้อมูลเกิดขึ้นที่รีจิสเตอร์ SBUF
MOV  A , SBUF       ; อ่านค่าจากรีจิสเตอร์ โดยการโอนย้ายข้อมูลผ่านทางรีจิสเตอร์ A
CLR  RI             ; หลังจากทำการอ่านข้อมูลเรียบร้อยแล้ว ต้องทำการเคลียร์บิต
                      RI เสมอ
```

2.3 พอร์ตอนุกรม

มีทางเลือกอยู่ 2 ทาง ในการที่จะเคลื่อนย้ายข้อมูลจากคอมพิวเตอร์ไปยังอุปกรณ์ต่อพ่วงอื่นๆหรือคอมพิวเตอร์ด้วยกัน นั่นคือการรับส่งข้อมูลแบบขนานและการรับส่งข้อมูลแบบอนุกรม การรับส่งข้อมูลแบบขนานจะเป็นการรับหรือส่งข้อมูลคราวละ 4 บิต หรือ 8 บิต ในเวลาเดียวกัน ซึ่งจะทำให้การรับและ

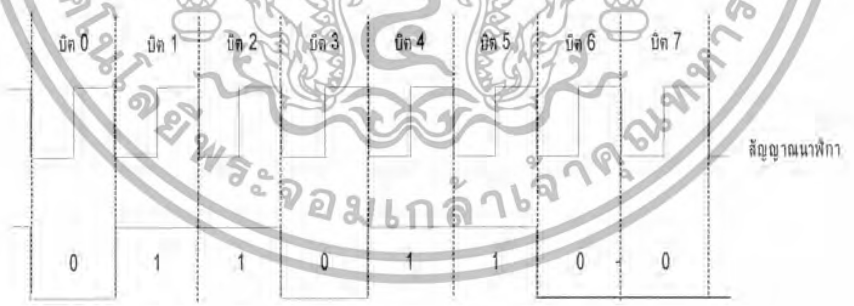
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งข้อมูลทำได้ที่ความเร็วสูง ซึ่งก็หมายถึงว่าจำนวนของสายที่ใช้ในการส่งจะต้องมีมากเท่ากับจำนวนบิตของข้อมูลที่จะส่งด้วย นอกจากนี้ยังจะต้องรวมถึงสายที่ใช้สำหรับการควบคุมและการตรวจสอบการรับส่งข้อมูลด้วย ซึ่งอาจจะต้องใช้สายมากกว่า 2 เท่าของจำนวนบิตข้อมูลที่จะส่งก็ได้ ซึ่งก็เป็นปัญหาในเรื่องราคาของสายที่ใช้ในการเชื่อมต่อแบบขนานมักจะมีราคาแพง

ในขณะที่การรับส่งข้อมูลแบบอนุกรม เป็นการรับส่งข้อมูลครั้งละ 1 บิต แต่ก็สามารถรับส่งข้อมูลได้คราวละหลายๆบิตได้ หากแต่จะต้องมีการตกลงกันระหว่างตัวส่งและตัวรับว่า จะรับส่งข้อมูลคราวละกี่บิต ตัวรับจะต้องรอข้อมูลมาให้ครบทุกบิตเสียก่อนจึงทำการประมวลผล ส่งผลให้การสื่อสารข้อมูลอนุกรมอาจมีความเร็วต่ำกว่าแบบขนาน ในด้านจำนวนสายสัญญาณการรับส่งข้อมูลแบบอนุกรมจะใช้จำนวนสายน้อยกว่ามาก อย่างน้อยที่สุดใช้เพียง 2-3 เส้นเท่านั้น แต่อัตราเร็วในการรับส่งข้อมูลอาจต่ำกว่าแบบขนาน อย่างไรก็ตามการรับส่งข้อมูลแบบอนุกรมสามารถใช้สายสัญญาณที่มีความยาวมากกว่าแบบขนาน ทำให้ระยะทางในการสื่อสารข้อมูลแบบอนุกรมสามารถทำได้มากกว่า

2.4 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นจะแบ่งออกได้เป็น 2 แบบคือการสื่อสารอนุกรมแบบซิงโครนัสและการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คืออีธีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้น คือ สัญญาณนาฬิกา, ข้อมูล และกราวด์ รูปที่ 2.14 แสดงให้เห็นถึงไทมิ่งไดอะแกรมของการส่งข้อมูลแบบซิงโครนัส



รูปที่ 2. 14 แสดงรูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม

2.4.1 การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัส คือการรับข้อมูลและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (baudrate) มีหน่วยเป็น บิตต่อวินาที(bit per second : bps)

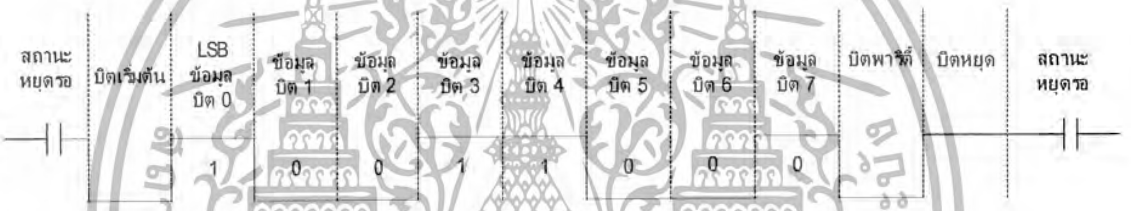
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

1. บิตเริ่มต้น(Start Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5 บิต, 6 บิต, 7 บิต หรือ 8 บิต
3. บิตตรวจสอบพาริตี้(Parity Bit) จะมีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้าย(Stop Bit) จะมีขนาด 1 บิต, 1.5 บิต หรือ 2 บิต

รูปที่ 2.15 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส ซึ่งเมื่อไม่มีข้อมูลที่จะส่ง ขา DATA จะมีสถานะลอจิก "1" ซึ่งจะเรียกสถานะนี้ว่าสถานะหยุดรอ (Waiting stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก "0"ด้วยช่วงระยะเวลา 1 บิต ซึ่งจะเรียกบิตนี้ว่าบิตเริ่มต้นจากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด(LSB) ก่อน ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือบิตปิดท้าย ซึ่งจะให้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่ง คือ บิตปิดท้าย ซึ่งจะให้ขาดาต้ามีสถานะลอจิก 1 อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต, 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว



รูปที่ 2.15 แสดงรูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับและส่งข้อมูลแบบอะซิงโครนัสเรียกว่า Universal Asynchronous Receiver / Transmitter หรือ UART อัตราความเร็วในการรับและส่งข้อมูลของการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสคือค่าบอดเรตซึ่งก็คือ ค่าจำนวนบิตต่อวินาทีที่ใช้ในการรับและส่งข้อมูลบอดเรตมาราฐานที่ใช้สำหรับพอร์ทอนุกรม อาร์เอสสองสามสอง ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ ซึ่งการรับและส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะสามารถกำหนดค่าบอดเรตได้สูงถึง 115200 บิตต่อวินาที เนื่องจากบอดเรต คือ จำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่างข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พาริตี้ความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที

การตรวจสอบพาริตี้สามารถกำหนดให้เป็นแบบคี่ (odd), แบบคู่ (even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้ การตรวจสอบพาริตี้เป็นการตรวจสอบจำนวนรวมของบิตที่เป็นลอจิก "1" ภายในข้อมูลที่ส่งไป 1 ไบต์ว่ามีจำนวนรวมเป็นเลขคู่หรือเลขคี่โดยต้องรวมบิตพาริตี้เข้าไปด้วย ยกตัวอย่างข้อมูลที่จะทำการส่งมีขนาด 8 บิต และมีค่าเท่ากับ 99 ฐานสิบหก หรือ 10011001 ฐานสอง จะเห็นว่าข้อมูลในไบต์นี้มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

54951

จำนวนลอจิก "1" จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตี้เป็นคู่ค่าในบิตพาริตี้ จะต้องมีลอจิกเป็น "0" แต่ถ้าพาริตี้เป็นคี่ ค่าที่บิตพาริตี้จะต้องเป็น "1" เพื่อให้ข้อมูล 1 ไบต์รวมทั้งพาริตี้มีจำนวนบิตที่เป็นลอจิก "1" มีจำนวนรวมกันเป็นเลขคี่ ในตารางที่ 2.1 แสดงตัวอย่างของบิตพาริตี้ในการรับส่งข้อมูลอนุกรม

บิตพาริตี้ถูกสร้างขึ้นจากภาคส่งข้อมูล ของยูเออาร์ที ซึ่งทางภาครับจะต้องทำการกำหนดค่าของคุณสมบัติการตรวจสอบพาริตี้ให้ตรงกันว่าจะตรวจสอบพาริตี้คี่หรือพาริตี้คู่ จากนั้นภาครับของยูเออาร์ทีจะทำการตรวจสอบค่าพาริตี้ที่เกิดขึ้นว่าเป็นคู่หรือเป็นคี่ โดยการนับจำนวนลอจิก "1" ทั้งหมดรวมทั้งบิตพาริตี้ด้วย ถ้ากำหนดพาริตี้ไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้ทราบ นับเป็นการตรวจสอบความผิดพลาดที่เกิดขึ้นในการถ่ายทอดข้อมูลที่ง่ายที่สุด แต่จะเชื่อถือได้เมื่อมีบิตข้อมูลที่ทำการส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ทำการส่งมีบิตที่ผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตี้บิตเป็นนัน (NONE) นั้น ทั้งภาครับและภาคส่ง จะไม่มีการตรวจสอบพาริตี้

ข้อมูล	บิตพาริตี้คู่	บิตพาริตี้คี่
00000000	0	1
00000001	1	0
00000010	1	0
00000011	0	1
00000100	1	0
11111110	0	1
11111111	1	0

ตารางที่ 2.1 แสดงบิตพาริตี้ของข้อมูล

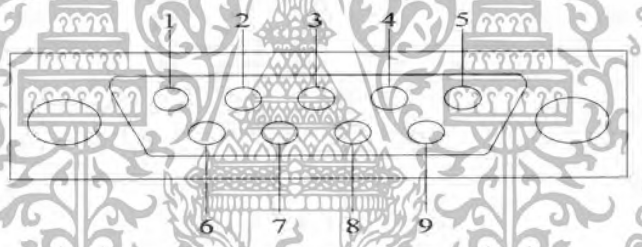
2.4.2 มาตรฐานพอร์ทอนุกรมแบบอาร์เอสสองสามสอง (RS-232)

มาตรฐานการเชื่อมต่อแบบอนุกรม อาร์เอสสองสามสอง เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทางโดยมาตรฐาน อาร์เอสสองสามสอง ในอดีตนั้นถูกออกแบบมา เพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้สื่อสารผ่านสายโทรศัพท์ ไปยังคอมพิวเตอร์อีกชุดซึ่งอยู่ห่างไกลกัน โดยคณะกรรมการที่เรียกว่าสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association : EIA) ได้วางมาตรฐานที่มีชื่อเรียกกันว่า EIA RS-232 มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็กเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3 ถึง -15 โวลท์ แสดงว่ามีข้อมูล (Mark) และ +3 ถึง +15 แสดงว่าเป็นช่องว่าง (Space)

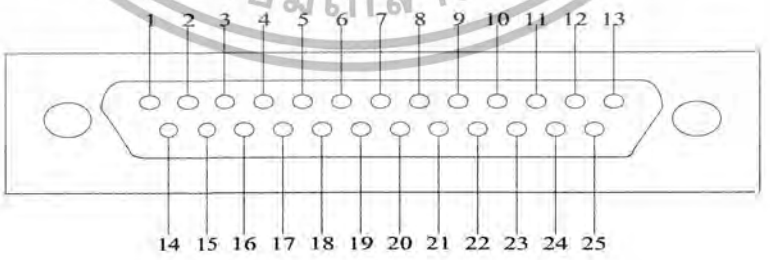
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐาน อาร์เอสสองสามสอง ได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูลไว้ว่าอุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์ หรือ ไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน อาร์เอสสองสามสอง ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE คือ คอนเน็กเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่ที่โมเด็มจะเป็นแบบ DCE

สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม อาร์เอสสองสามสอง มักถูกใช้เชื่อมต่อกับโมเด็ม หรือเมาส์โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุดถึง 20 เมตร คอนเน็กเตอร์สำหรับพอร์ตอนุกรม RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 ตัวผู้ หรือ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้น เช่นเดียวกันคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่นๆที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนักจึงถูกยกเลิกไป โดยแสดงรูปร่างและตำแหน่งขา ดังรูปที่ 2.16



ก) การจัดขาคอนเน็กเตอร์พอร์ตอนุกรม 9 ขา หรือแบบ DB-9



ข) การจัดขาคอนเน็กเตอร์พอร์ตอนุกรม 25 ขา หรือแบบ DB-25

รูปที่ 2.16 แสดงการจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมทั้งแบบ DB-9 และ DB-25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

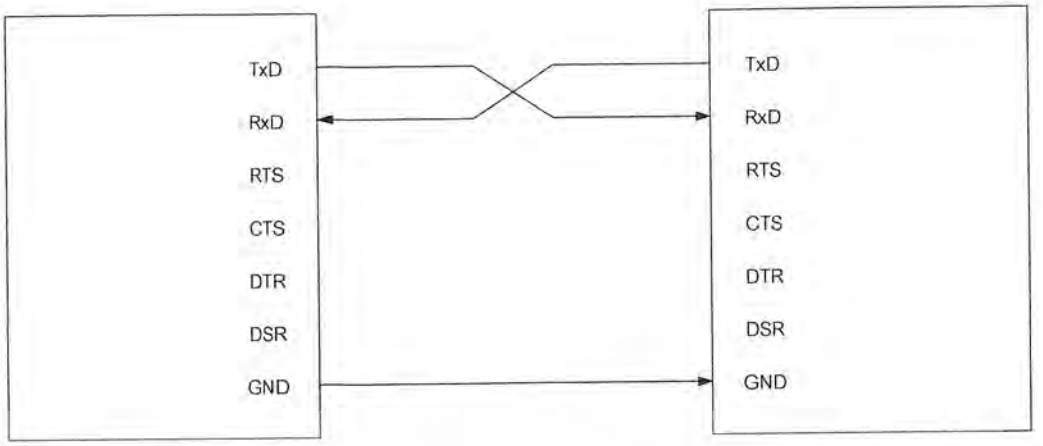
คอนเน็กเตอร์ DB-9	คอนเน็กเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect:DCD	Input
2	3	Received Data : RxD	Input
3	2	Transmitted Data : TxD	Output
4	20	Data Terminal Ready : DTR	Output
5	7	Signal Ground : GND	
6	6	Data Set Ready : DSR	Input
7	4	Request To Send : RTS	Output
8	5	Clear To Send : CTS	Input
9	22	Ring Indicator : RI	Input

ตารางที่ 2.2 แสดงตำแหน่งและชื่อขาของ DB-9 และ DB-25

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกดังรูปที่ 2.17 ลูกศรในรูปแสดงถึงทิศทางของข้อมูล ในรูป 2.17 (ก) เป็นการเชื่อมต่อแบบนัลโมเด็ม (Null modem) หรือการเชื่อมต่อโดยตรงโดยไม่ต้องผ่านโมเด็ม โดยมีการตรวจสอบหรือแฮนด์เช็กเต็มรูปแบบ ส่วนในรูปที่ 2.17 (ข)เป็นการเชื่อมต่อแบบนัลโมเด็ม ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้น

ก) การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์แบบนัลโมเด็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ข) การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ โดยใช้สายสัญญาณเพียง 3 เส้น

รูปที่ 2.17 แสดงการต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์

โดยเส้นหนึ่งสำหรับส่งข้อมูลอีกเส้นสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวด์ สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม อาร์เอสสองสามสอง มีดังนี้

- Data Carrier Detect : DCD หรืออาจเรียกว่า Carrier Detect : CD ขานี้จะแอกทีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติขานี้จะไม่ได้ถูกใช้งานมากนัก
- Received Data : RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์บัฟเฟอร์
- Transmitted Data : TD หรือ TxD ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป
- Data Terminal Ready : DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่า ต้องการติดต่อด้วยโดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อแบบนัลโมเด็มซึ่งใช้สายในการเชื่อมต่อเพียงเส้น 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา DCD ด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์
- Signal Ground : GND ขากราวด์ของระบบ
- Data Set Ready : DSR ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR
- Request To Send : RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบนัลโมเด็ม 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 ระดับแรงดันที่ใช้งานสำหรับพอร์ตอนุกรม อาร์เอสสองสามสอง

มาตรฐานการสื่อสารข้อมูลของพอร์ตอนุกรม ได้ระบุช่วงระดับแรงดันสำหรับการทำงานของพอร์ตอนุกรมไว้ว่า ที่ลอจิก "0" จะมีระดับสัญญาณ +3 ถึง +15 โวลต์ ส่วนลอจิก "1" จะมีระดับสัญญาณ -3 ถึง -15 โวลต์ ระดับสัญญาณนี้ทำให้ไม่สามารถที่จะนำขาเอาท์พุตใดๆ ต่อเข้ากับลอจิกเกตเพื่อใช้งานได้โดยตรง จะต้องผ่านวงจรเพื่อเปลี่ยนระดับแรงดันเสียก่อนนั่นก็คือวงจรขับ อุปกรณ์ขับแบบมาตรฐานอาร์เอสสองสามสอง นั้น ในด้านภาคส่งต้องสามารถเปลี่ยนสัญญาณลอจิกให้เป็นระดับแรงดันตามค่าที่กำหนดไว้ได้ และสำหรับในส่วนของวงจรภาครับเองก็ต้องสามารถตรวจจับระดับแรงดันที่รับเข้ามาแล้วเปลี่ยนกลับให้เป็นสัญญาณลอจิกได้อย่างถูกต้องด้วยเช่นกัน โดยปกติจะใช้ไอซีจำพวก อาร์เอสสองสามสอง Transceiver ที่นิยมมากคือ MAX232 หรือ ICL232 ไอซีในกลุ่มนี้จะทำหน้าที่แปลงระดับแรงดันทีทีแอล ให้เป็นสัญญาณในระดับแรงดันของ อาร์เอสสองสามสอง และยังทำหน้าที่แปลงสัญญาณในระดับแรงดันของ อาร์เอสสองสามสอง ให้กลับมาอยู่ในระดับทีทีแอล โดยลอจิก "0" ซึ่งเดิมมีระดับสัญญาณ +3 ถึง +15 โวลต์ จะถูกแปลงเป็น 0 โวลต์ ส่วนลอจิก "1" ซึ่งมีระดับสัญญาณ -3 ถึง -15 โวลต์ จะแปลงเป็น +5 โวลต์ ทั้งนี้เพื่อสามารถเชื่อมต่อกับอุปกรณ์ดิจิตอลอื่นที่ใช้ระดับแรงดันทีทีแอลได้

รูปที่ 2.18 แสดงวงจรขับแบบ อาร์เอสสองสามสอง โดยใช้ MAX 232

2.5 การเขียนโปรแกรมวิชวลเบสิกเพื่อใช้งานพอร์ตอนุกรม

คอนโทรล MSComm

คอนโทรลที่สำคัญในการทำให้วิชวลเบสิก สามารถสื่อสารผ่านพอร์ตอนุกรมได้นั่นก็ คือ คอนโทรล MSComm ซึ่งจัดเตรียมทางเลือกเอาไว้ 2 ทาง ทางแรกคือการสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (Event-Driven Communications) เป็นรูปแบบการใช้งานที่มีประสิทธิภาพมาก สำหรับการตอบสนองแบบทันทีทันใด เช่น เมื่อตัวอักษรถูกส่งมาที่พอร์ตอนุกรม หรือเกิดการเปลี่ยนแปลงที่ขา Data Carrier Detect (DCD) หรือขา Request To Send (RST) เหตุการณ์ Oncomm ของ MSComm จะสามารถตรวจจับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ทันที ซึ่งจะกล่าวถึงรายละเอียดในหัวข้อคุณสมบัติ CommEvent ต่อไป ส่วนทางเลือกที่สองเป็นการคอยตรวจสอบค่าเหตุการณ์ และความผิดพลาดที่เกิดขึ้นด้วยการดูค่าที่เปลี่ยนแปลงภายในคุณสมบัติ CommEvent หลังจากให้โปรแกรมทำงานในฟังก์ชั่นต่างๆไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้ดีในกรณีที่โปรแกรมมีขนาดเล็ก

คอนโทรล MSComm 1 ตัวสามารถควบคุมการทำงานของพอร์ตอนุกรมได้ 1 พอร์ต ถ้าในโปรแกรมที่ใช้งานต้องการติดต่อกับพอร์ตอนุกรมมากกว่า 1 พอร์ต จะต้องใช้คอนโทรล MSComm มากกว่า 1 ตัว เพื่อควบคุมพอร์ตอนุกรมในแต่ละพอร์ต โดยแอสเดรสของพอร์ตอนุกรมและแอสเดรสของการเกิดอินเทอร์รัปต์สามารถเปลี่ยนแปลงได้จากการแก้ไขค่าที่ Control Panel

ถึงแม้ว่า คอนโทรล MSComm จะมีคุณสมบัติ (property) มากมาย แต่สามารถทำความเข้าใจได้ไม่ยากดังนี้

1. CommPort

ใช้ในการกำหนดและอ่านค่าพอร์ตอนุกรมที่ติดต่ออยู่ (COM1, COM2, COM3, COM4)

รูปแบบการใช้งาน

object.CommPort[=value]

โดย Value เป็นค่าของพอร์ตอนุกรม ชนิดของข้อมูลเป็น Inter ค่า Value สามารถกำหนดได้ในช่วง 1-16 (ค่าเริ่มต้นกำหนดไว้ที่ 1) เมื่อมีการกำหนดค่าแล้วทำการเปิดพอร์ตโดยใช้คุณสมบัติ PortOpen แต่ว่าพอร์ตนั้นไม่มีอยู่ในระบบ MSComm จะสร้างสัญญาณแสดงข้อผิดพลาด Error 68 ขึ้นมา ซึ่งหมายถึงอุปกรณ์ตัวนี้ไม่มีอยู่ในระบบ ดังนั้นการเขียนโปรแกรมจึงต้องกำหนดตำแหน่งของพอร์ตอนุกรมก่อนที่จะใช้คำสั่ง OpenPort

2. Setting

ในการกำหนดและอ่านค่าอัตราบอด, พาริตี้, จำนวนบิตข้อมูล, จำนวนบิตปิดท้าย

รูปแบบการทำงาน

object.Setting[=value]

ค่า Value มีชนิดข้อมูลเป็นแบบ String มีรูปแบบเป็น "BBBB,P,D,S" โดย BBBB เป็นค่าอัตราบอด, P เป็นค่าพาริตี้, D เป็นจำนวนของบิตข้อมูล และ S เป็นจำนวนข้อมูลของบิตปิดท้าย ปกติแล้วค่านี้ถูกกำหนดไว้เป็น "9600,N,8,1"

ค่าบอดเรตมาตรฐานที่ใช้กับ MSComm มีดังนี้

110 บิตต่อวินาที

300 บิตต่อวินาที

600 บิตต่อวินาที

1200 บิตต่อวินาที

2400 บิตต่อวินาที

9600 บิตต่อวินาที (ค่าปกติ)

14400 บิตต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

28800 บิตต่อวินาที

38400 บิตต่อวินาที (สงวน)

56000 บิตต่อวินาที (สงวน)

12800 บิตต่อวินาที (สงวน)

256000 บิตต่อวินาที (สงวน)

สำหรับค่ามาตรฐานในการกำหนดค่าพาริตี้มีดังนี้

สัญลักษณ์	รายละเอียด
E	พาริตี้คู่ (Even)
M	ลอจิก "1" (Mark)
N	ไม่ใช้(ค่าปกติ)
O	พาริตี้คี่ (Odd)
S	ลอจิก "0" (Space)

ค่าที่ใช้ในการกำหนดจำนวนบิตมี 5 ค่าคือ 4,5,6,7 และ 8 (เป็นค่าปกติ)

ค่าที่ระบุจำนวนบิตปิดท้ายมี 3 ค่าคือ 1 (เป็นค่าปกติ) , 1,5 และ 2

ตัวอย่างการใช้งานคำสั่ง Setting โดยจะเป็นการกำหนดค่าบอดเรตเท่ากับ 9600 ไม่มีพาริตี้ จำนวนข้อมูล 8 บิต และบิตปิดท้าย 1 บิต สามารถเขียนโปรแกรมได้ดังนี้

MSComm.Setting = "9600,N,8,1"

หมายเหตุ สาเหตุที่ค่าที่กำหนดจะต้องอยู่ภายในเครื่องหมายคำพูด "" เนื่องจากค่าที่กำหนดนี้อยู่ในรูปตัวแปร String

3. PortOpen

ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อเปิดและปิดพอร์ตอนุกรม

รูปแบบการใช้งาน

object.PortOpen[=value]

ค่า Value มีชนิดข้อมูลเป็นแบบบูลีน คือ True และ False โดย True หมายถึงการเปิดพอร์ตอนุกรม และ False หมายถึงการปิดพอร์ตอนุกรม สำหรับการปิดพอร์ตนั้นจะมีการเคลียร์บัฟเฟอร์รับข้อมูลและบัฟเฟอร์ส่งข้อมูลด้วย คอนโทรล MSComm จะปิดพอร์ตอนุกรมโดยอัตโนมัติเมื่อออกจากโปรแกรม ก่อนที่จะใช้คุณสมบัติ PortOpen ต้องตรวจสอบให้แน่ใจก่อนว่าคุณสมบัติ CommPort นั้นได้ทำการกำหนดตำแหน่งของพอร์ตอนุกรมไว้ถูกต้องหรือไม่ มิเช่นนั้น MSComm จะแสดงข้อผิดพลาด error 68 แจ้งแก่ผู้ใช้งาน หรือถ้าพอร์ตอนุกรมนั้นถูกปิดเอาไว้แล้ว โปรแกรมก็จะแจ้งข้อผิดพลาดออกมาเช่นเดียวกัน

ถ้าคุณสมบัติ DTREnable หรือ RTSEnable ถูกกำหนดให้เป็น True ก่อนที่จะทำการเปิดพอร์ต ค่าคุณสมบัติ DTREnable หรือ RTSEnable จะถูกเซตเป็น False หลังจากเปิดพอร์ต แต่ถ้าเซตเป็น False หลังจากเปิดโปรแกรมแล้วค่าที่กำหนดไว้จะเป็นค่าเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการใช้คำสั่งเปิดพอร์ต เพื่อติดต่อสื่อสารกับพอร์ตอนุกรม COM1 และมีบอดเรต 9600 บิตต่อวินาที ไม่มีพาริตี้จำนวนข้อมูล 8 บิต และบิตปิดท้าย 1 บิต มีดังนี้

```
MSComm1.Settings = "9600,N,8,1"
MSComm1.Commport = 1
MSComm1.PortOpen = True
```

4. Input

อ่านค่าและลบค่าขบวนข้อมูลจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งาน

object.Input

คุณสมบัติ InputLen เป็นตัวกำหนดจำนวนของตัวอักษรที่จะอ่านโดยคุณสมบัติ Input การกำหนดค่าให้ InputLen เท่ากับ 0 เป็นการกำหนดให้คุณสมบัติ Input ทำการอ่านค่าข้อมูลในบัฟเฟอร์รับข้อมูลทั้งหมด

คุณสมบัติ InputMode เป็นตัวกำหนดชนิดของข้อมูลที่คุณสมบัติ Input รับเข้ามาถ้า InputMode ถูกกำหนดเป็น comInputModeText คุณสมบัติ Input จะส่งค่าข้อมูลกลับมาในรูปแบบของข้อความชนิดข้อมูลเป็นแบบ Variant ถ้า InputMode กำหนดเป็น comInputModeBinary คุณสมบัติ Input จะส่งข้อมูลกลับมาในรูปของไบนารี่และชนิดข้อมูลเป็นแบบ Variant

ตัวอย่างโปรแกรมแสดงให้เห็นถึงวิธีในการรับข้อมูลจากบัฟเฟอร์รับข้อมูลทั้งหมด

```
Private Sub Command1_Click( )
Dim Instring as String
    MSComm1.InputLen = 0            'Retrieve all available data.
    If MSComm.InBufferCount Then    'Check for data.
    InString = MSComm1.Input        'Read data.
    End If
    End Sub
```

5. InBufferCount

ส่งค่าจำนวนของตัวอักษรที่อยู่ในบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

object.InbufferCount[= Value]

6. InBufferSize

การกำหนดและคืนค่าขนาดของบัฟเฟอร์ภาครับในหน่วยเป็นไบต์

รูปแบบการใช้งานคำสั่ง

object.InBufferSize [= value]

คำสั่ง InBufferSize ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ โดยค่าเริ่มต้นถูกกำหนดไว้ที่ 1024 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. InputLen

กำหนดค่าและคืนค่าจำนวนของตัวอักษรที่อ่านจากบัฟเฟอร์

รูปแบบการใช้งานคำสั่ง

object.InputLen [= value]

8. InputMode

การกำหนดค่าและคืนค่าชนิดของข้อมูลที่รับโดยคำสั่ง Input

object.InputMode [= value]

คุณสมบัติ InputMode ใช้กำหนดว่าข้อมูลชนิดไหนที่รับเข้ามาผ่านคำสั่ง Input โดยข้อมูลจะเลือกได้ 2 ประเภทคือ

commInputModeText สำหรับข้อมูลที่อยู่ในรูปข้อความตัวอักษรตามมาตรฐาน ASCI โดยจะต้องกำหนดค่าเป็น "0" และค่าเริ่มต้นของการรับค่าข้อมูลก็จะเป็นค่านี้

comInputModeBinary สำหรับข้อมูลอื่นๆ ซึ่งจะเก็บในรูปไบนารี่รวมกันอยู่เป็นไบต์ข้อมูล

9. Output

ใช้ในการส่งขบวนของข้อมูลไปยังบัฟเฟอร์ส่งข้อมูล

รูปแบบการใช้งาน

object.Output [= value]

ค่า Value เป็นค่าของตัวอักษรที่เขียนไปยังบัฟเฟอร์ส่งข้อมูล คุณสมบัติเอาท์พุตสามารถใช้ในการส่งข้อมูลตัวอักษรหรือข้อมูลไบนารี่ก็ได้ โดยการส่งข้อมูลเป็นรูปแบบตัวอักษรจะต้องกำหนดข้อมูลเป็นแบบ Variant และมีข้อมูลภายในเป็นแบบ String สำหรับการส่งข้อมูลไบนารี่จะต้องกำหนดชนิดของข้อมูลเป็นแบบ Variant และมีข้อมูลภายในเป็นแบบไบต์

10. OutBufferCount

คืนค่าจำนวนของข้อมูลตัวอักษรที่เก็บอยู่ในบัฟเฟอร์ภาคส่ง และสามารถใช้คำสั่งนี้เพื่อเคลียร์บัฟเฟอร์ภาคส่งได้ด้วย

รูปแบบการใช้งานคำสั่ง

object.BufferCount [= value]

11. OutBufferSize

กำหนดค่าและคืนค่าขนาดของบัฟเฟอร์ภาคส่ง ชนิดตัวแปรแบบไบต์

รูปแบบการใช้งานคำสั่ง

object.OutBufferSize [= object]

คุณสมบัติ OutBufferSize ใช้สำหรับกำหนดขนาดของบัฟเฟอร์ภาคส่ง โดยค่าปกติที่ใช้งานจะมีค่าเท่ากับ 512 ไบต์

12. ParityReplace

กำหนดและคืนค่าตัวอักษรที่ไปวางแทนในตำแหน่งที่เกิดข้อผิดพลาดจากพาริตี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการใช้งานคำสั่ง

object.ParityReplace [= value]

บิตพาริตี้ เป็นบิตที่ทางภาคส่งข้อมูลทำการส่งมาพร้อมกับข้อมูล เพื่อตรวจสอบข้อผิดพลาดของข้อมูล โดยเมื่อมีการใช้บิตพาริตี้ คอนโทรล MSComm จะทำการบวกบิตทุกบิตที่มีค่าลอจิก "1" ในแต่ละไบต์และทำการตรวจสอบผลลัพธ์ว่าบิตที่อ่านได้นั้นมีจำนวนลอจิก "1" เป็นเลขคู่หรือคี่ และตรงกับค่าที่กำหนดไว้แต่ต้นหรือไม่ ถ้าค่าที่นำมาบวกแล้วมีพาริตี้ไม่ตรงแสดงว่าการรับส่งข้อมูลผิดพลาด

2.6 การสร้างแอพพลิเคชันฐานข้อมูล

2.6.1 ความหมายของฐานข้อมูล

ฐานข้อมูล คือ วิธีการจัดเก็บข้อมูลที่สัมพันธ์กันอย่างมีระเบียบ ซึ่งจะทำให้ง่ายต่อการใช้งานและค้นหาข้อมูล ซึ่งฐานข้อมูลที่คนส่วนใหญ่คุ้นเคยคือ ฐานขอมูลเชิงสัมพันธ์ เป็นรูปแบบการจัดเก็บข้อมูล ที่สัมพันธ์กัน โดยมองข้อมูลในลักษณะของตารางต่างๆ ที่มีความสัมพันธ์กัน

2.6.2 องค์ประกอบของฐานข้อมูล

ฐานข้อมูลเป็นเพียงวิธีคิดเท่านั้น แต่การใช้งานฐานข้อมูลจะต้องประกอบด้วยองค์ประกอบต่อไปนี้

- แอพพลิเคชันฐานข้อมูล (Database Application)

แอพพลิเคชันฐานข้อมูล เป็นแอพพลิเคชันที่สร้างไว้ให้ผู้ใช้งานสามารถติดต่อกับฐานข้อมูลได้อย่างสะดวก ซึ่งมีรูปแบบการติดต่อกับฐานข้อมูลแบบเมนู หรือกราฟิก โดยไม่จำเป็นต้องมีความรู้เกี่ยวกับฐานข้อมูลเลยก็สามารถเรียกใช้งานฐานข้อมูลได้

- ระบบจัดการฐานข้อมูล (Database Management System)

ระบบจัดการฐานข้อมูล เป็นซอฟต์แวร์ที่ทำหน้าที่จัดการข้อมูลในฐานข้อมูล ทั้งการจัดเก็บ , การแสดงผล , การค้นหา , การสำรองข้อมูลฯลฯ โดยจะเป็นเครื่องมือในการทำงานของผู้บริหารฐานข้อมูล และเป็นตัวกลางที่เชื่อมผ่านระหว่างแอพพลิเคชันฐานข้อมูลที่สร้างขึ้นกับตัวข้อมูลในฐานข้อมูล ตัวอย่างของ เช่น DBMS , Microsoft Access , FoxPro , SQLServer , Oracle , Informix , DB2 เป็นต้น

- ดาต้าเบสเซิร์ฟเวอร์ (Database Server)

ดาต้าเบสเซิร์ฟเวอร์ เป็นคอมพิวเตอร์ที่คอยให้บริการในการจัดการฐานข้อมูล ซึ่งก็คือคอมพิวเตอร์ที่ระบบจัดการฐานข้อมูลทำงานอยู่นั่นเอง เพราะฉะนั้นจึงมักจะเป็นคอมพิวเตอร์ที่มีประสิทธิภาพการทำงานสูงกว่าคอมพิวเตอร์ที่ใช้งานทั่วไป

- ข้อมูล (Data)

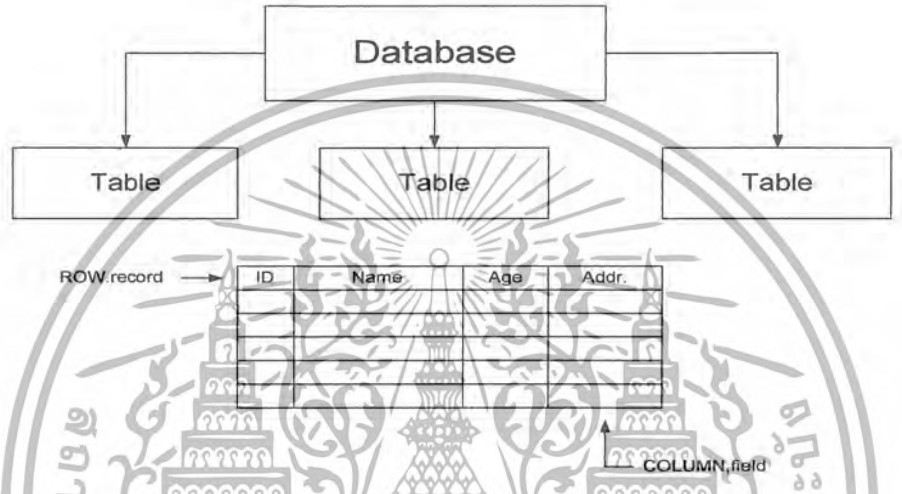
ข้อมูล คือตัวเนื้อของข้อมูลที่เราใช้งาน ซึ่งจะถูกเก็บไว้ในหน่วยความจำของดาต้าเบสเซิร์ฟเวอร์ โดยจะถูกเรียกมาใช้งานจากระบบการจัดการฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ผู้บริหารฐานข้อมูล (Database Administrator)

ผู้บริหารฐานข้อมูล เป็นคนที่ทำหน้าที่ดูแลข้อมูล ในฐานข้อมูลผ่านระบบจัดการฐานข้อมูล ซึ่งจะควบคุมให้การทำงานเป็นไปอย่างราบรื่น นอกจากนี้ยังทำหน้าที่ในการกำหนดผู้ที่จะมีสิทธิ์ใช้งานฐานข้อมูล , กำหนดในเรื่องความปลอดภัยของการใช้งาน พร้อมทั้งดูแลดาต้าเบสเซิร์ฟเวอร์ ให้ทำงานอย่างเป็นปกติด้วย

2.6.3 โปรแกรมวิชวลเบสิกกับการจัดการฐานข้อมูล



รูปที่ 2.19 แสดงมุมมองของโปรแกรมวิชวลเบสิกกับฐานข้อมูล

วิชวลเบสิกจะมองเห็นฐานข้อมูลหนึ่งๆ ซึ่งจะประกอบมาจากตารางหลายๆ อันมาประกอบกัน ในแต่ละตารางก็จะประกอบไปด้วยคอลัมน์ต่างๆ แยกข้อมูลแต่ละพวกออกจากกัน และในแต่ละตารางก็จะแบ่งข้อมูลแต่ละชุดออกเป็นแถวๆ (Row)

สำหรับการจัดการ และใช้งานฐานข้อมูลเบื้องต้นนี้วิชวลเบสิก ได้เตรียม ActiveX Control ที่มีชื่อว่า Data Control มาช่วย นอกจากนี้ยังสามารถใช้ ActiveX Control อื่นๆ มาช่วยในการแสดงข้อมูลภายในฐานข้อมูล

แต่ถ้างานที่ต้องการความยืดหยุ่น และประสิทธิภาพในการใช้งานฐานข้อมูลสูงๆ เราจะใช้การเขียนโปรแกรมโดยใช้กลุ่มของออบเจ็กต์ที่ วิชวลเบสิก เตรียมไว้ให้โดยเฉพาะที่เรียกว่า DAO (Data Access Object)

เมื่อฐานข้อมูลขยายออกไป และนำไปสู่ การใช้งานแบบไคลเอนท์/เซิร์ฟเวอร์ วิชวลเบสิกก็เตรียมกลุ่มออบเจ็กต์ที่จัดการกับฐานข้อมูลที่ใช้งานผ่านเครือข่ายที่มีชื่อว่า RDO (Remote Data Object)

และสุดท้ายเมื่ออินเตอร์เน็ตเข้ามามีบทบาทต่อการทำงาน การสร้างแอพพลิเคชันฐานข้อมูลเพื่อใช้งานร่วมกับอินเตอร์เน็ตก็สามารถทำได้โดยใช้กลุ่มของออบเจ็กต์ที่มีชื่อว่า ADO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.4 การใช้งานภาษา SQL กับฐานข้อมูล

ภาษา SQL (Strutured Query Language) เป็นภาษาที่ใช้ในการเรียกค้น (Query) และจัดการกับฐานข้อมูลเชิงสัมพันธ์ เป็นภาษาที่มีความคลึงกับภาษามนุษย์มาก โดยสามารถจะเขียนแทรกลงไปในแอพพลิเคชันที่ใช้ติดต่อกับฐานข้อมูล หรือสั่งงานระบบจัดการฐานข้อมูลได้

คำสั่งของภาษา SQL (SQL Statement) นั้นมีความใกล้เคียงกับภาษามนุษย์มาก โดยที่สามารถกำหนดลงไปในคำสั่งเลยว่าต้องการอะไร โดยภาษา SQL นั้นแบ่งตามความสามารถออกเป็น 2 ส่วน

1. ความสามารถในการจัดการลักษณะ และโครงสร้างฐานข้อมูล (DDL: Data Definition Language) เช่น Create table , Alter Table , Create View เป็นต้น
2. ความสามารถในการจัดการตัวข้อมูลในฐานข้อมูล (DML : Data Manipulation Language) เช่น การค้นหาข้อมูล , การ insert record , Delete record , การUpdate เป็นต้น

2.7 การติดต่อกับฐานข้อมูลด้วย ActiveX Data Object

การศึกษาวิธีการติดต่อกับฐานข้อมูลมีหลายวิธีด้วยกัน ซึ่งการติดต่อฐานข้อมูลวิธีนี้เป็นวิธีที่ถูกสร้างขึ้นมาใหม่ มีประสิทธิภาพการทำงานสูง และใช้งานง่าย ซึ่งถูกสร้างโดยใช้เทคโนโลยี OLE DB ซึ่งถูกออกแบบมาให้สามารถติดต่อกับฐานข้อมูลได้ทุกประเภทไม่ว่าจะเป็นฐานข้อมูลแบบ Relational หรือ Non-Relational รวมทั้งข้อมูลในรูปแบบ E-mail,Text และอื่นๆ นอกจากนี้ยังเป็นวิธีที่ใช้ Traffic ของระบบเครือข่ายน้อย รวมทั้งใช้หน่วยความจำในการทำงานน้อย ซึ่งเหมาะสำหรับสร้างโปรแกรมบนอินเตอร์เน็ตที่ติดต่อกับฐานข้อมูล

ADO นั้นถูกออกแบบมาเพื่อให้ Client สามารถเข้าถึง หรือจัดการกับฐานข้อมูลที่อยู่บนเครื่อง Server ผ่านทาง OLE DB Provide ซึ่งมีทั้ง Provider ของ Jet Engine , SQL Server , Oracl และอื่นๆ

2.7.1 ความสามารถต่างๆ ของ ActiveX Data Object

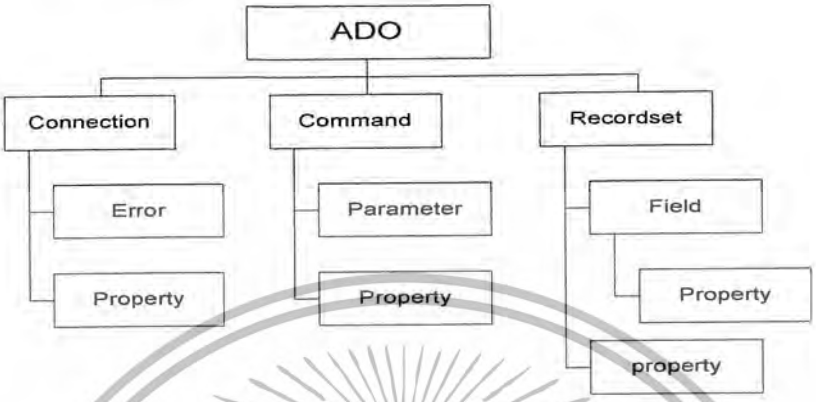
ความสามารถต่างๆ ของ ADO มีดังต่อไปนี้

- สามารถที่ใช้ในโปรแกรมที่ติดต่อฐานข้อมูลแบบ Client-Server รวมทั้งนำไปใช้ในโปรแกรมบนอินเตอร์เน็ตได้ด้วยเนื่องจากสนับสนุนการสร้างออบเจ็กต์แบบอิสระต่อกัน
- สนับสนุน Cursor ที่ใช้ชี้ไปยังข้อมูลในฐานข้อมูลหลายๆ ชนิด ทั้ง Cursor บน Server และ Client
- สามารถที่จะสร้างออบเจ็กต์ต่างๆ ได้อย่างอิสระ เนื่องจากโครงสร้างออบเจ็กต์เป็นแบบแบนราบไม่ใช่ลำดับ จึงไม่จำเป็นต้องสร้างออบเจ็กต์ Connectionก่อนแล้วจึงสร้างออบเจ็กต์ Resultset ทำให้ทำงานกับออบเจ็กต์ที่จำเป็นเท่านั้น จึงใช้หน่วยความจำในการทำงานน้อยลง
- สนับสนุนการเรียก ที่มีการส่งพารามิเตอร์เข้า/ออก รวมทั้งที่มีผลลัพธ์ส่งกลับมาได้
- สามารถจำกัดจำนวนของแถวของเรคคอร์ดที่ส่งกลับมาได้ ทำให้เราทราบเวลาที่ใช้ในการรันคิวรีได้ และง่ายต่อการจัดการกับทรัพยากรที่ต้องใช้
- มีอีเว็นต์ที่ใช้ในการจัดการคิวรีแบบ ทำให้เราสามารถควบคุมการรันคิวรีที่ต้องเสียเวลาในการทำงานนานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 โครงสร้างออบเจ็กต์ของ ActiveX Data Object

ADO นั้นจะมีโครงสร้างของออบเจ็กต์ต่างๆ ตามรูป 2.20 ซึ่งสามารถสร้างออบเจ็กต์แต่ละตัวได้เป็นอิสระจากกัน โดยมีรายละเอียดของออบเจ็กต์ดังต่อไปนี้



รูปที่ 2.20 แสดงโครงสร้างออบเจ็กต์ของ ActiveX Data Object

2.7.2.1 รายละเอียดของออบเจ็กต์ Connection

ออบเจ็กต์ Connection ใช้สำหรับเชื่อมต่อกับฐานข้อมูลที่จะทำงานด้วย ซึ่งต้องกำหนดคุณสมบัติต่างๆ ที่จำเป็นในการเชื่อมต่อกับฐานข้อมูล และออบเจ็กต์นี้ ยังใช้จัดการกับ Transaction ซึ่งจะช่วยให้ทำงานกับฐานข้อมูลได้ปลอดภัยยิ่งขึ้นด้วย ออบเจ็กต์นี้อาจจะกล่าวได้ว่า เป็นตัวแทนของการเชื่อมต่อกับเครื่อง Server ซึ่งคุณสมบัติ และเมดถอดบางอย่างอาจจะไม่สนับสนุนใน Provider หนึ่ง Provider แต่สนับสนุนใน อื่นๆ อีกก็ได้ เพราะฉะนั้นก็ขอให้ตรวจสอบก่อนการใช้งานในโปรแกรมด้วย โดยสำหรับในการเชื่อมต่อด้วยออบเจ็กต์ Connection จะต้องบอกด้วยว่า ใช้ Data Source ตัวใด , รหัสผ่าน และชื่อผู้ใช้เป็นอะไร ซึ่งต้องกำหนดค่าต่างๆเหล่านี้ให้อยู่ในรูปแบบที่เรียกว่า Connect String ซึ่งจะขอสรุปรูปแบบของ Connect String ซึ่งมีดังตารางที่ 2.3

ODBC Connect String	ใช้กำหนด
DSN=	กำหนดชื่อ Data Source ที่เรากำหนดใน Control Panel
UID=	กำหนดชื่อผู้ใช้ที่จะเข้าระบบ
PWD=	กำหนดรหัสผ่านที่เราใช้เข้าระบบ
DATABASE=	กำหนดชื่อฐานข้อมูลที่เราจะใช้ เช่นฐานข้อมูลOrders ใน SQL Server เป็นต้น
SERVER=	กำหนดชื่อเครื่อง Server ที่ฐานข้อมูลอยู่
DRIVER=	กำหนดชื่อไดรเวอร์ที่ใช้ติดต่อฐานข้อมูล เช่น {SQL Server} เป็นต้น

ตารางที่ 2.3 แสดงรูปแบบของ Connect String ของออบเจ็กต์ Connection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และออบเจ็กต์ connection มีคุณสมบัติที่สำคัญเกี่ยวกับข้อมูลต่างๆ ในการเชื่อมต่อ โดยมีรายละเอียดดังตารางที่ 2.4

ชื่อคุณสมบัติ	คำอธิบาย
CommandTimeout	กำหนดเวลาในหน่วยวินาทีที่รอคิวรีรันจนได้ผลลัพธ์กลับมา ซึ่งถ้ารันเกินเวลานี้จะหมดเวลารัน และเกิดข้อผิดพลาดขึ้นมา
DefaultDatabase	กำหนดชื่อฐานข้อมูลดีฟอลต์ที่ใช้
ConnectionString	ใช้กำหนด Data Source หรือฐานข้อมูลที่ใช้ ด้วยการส่งรายละเอียดของการเชื่อมต่อที่เราได้กล่าวมาแล้ว
ConnectionTimeout	กำหนดเวลาเป็นวินาทีที่รอจนกว่าการเชื่อมต่อจะเกิดขึ้น ซึ่งถ้าเกินเวลานี้ จะหมดเวลาเชื่อมต่อและเกิดข้อผิดพลาดขึ้นมา
Attributes	กำหนดวิธีการทำงานของ Transaction
Provider	กำหนดชื่อ Provider ที่ออบเจ็กต์ Connection ใช้ เช่น เราต้องกำหนดคุณสมบัตินี้เป็น SQLOLEDB ถ้าเราใช้ SQL Server OLE DB Provider เป็นต้น
CursorLocation	กำหนดให้ ADO สร้าง Cursor ที่ใช้จัดการกับฐานข้อมูลขึ้นที่ใด ซึ่งจะมีผลต่อประสิทธิภาพในการทำงาน
IsolationLevel	กำหนดวิธีที่ให้ ADO และ SQL Server จัดการกับ Transaction อย่างไร
Mode	บอกให้เราทราบว่า เราสามารถทำอะไรกับฐานข้อมูลที่เราเชื่อมต่อได้บ้าง
State	บอกให้เราทราบว่า สถานะการเชื่อมต่อเป็นอย่างไร เปิดอยู่ หรือยังไม่ว่าง
Version	บอกหมายเลขเวอร์ชันของ ADO ที่ใช้

ตารางที่ 2.4 แสดงคุณสมบัติที่สำคัญของออบเจ็กต์ connection

2.7.2.2 รายละเอียดของออบเจ็กต์ Recordset

ออบเจ็กต์ Recordset นี้ เป็น Recordset ที่เป็นผลลัพธ์ที่ส่งกลับมาจากการทำงานของ Select Query ซึ่งใน ADO นั้นจะสามารถสร้างออบเจ็กต์ Recordset หลายๆ ออบเจ็กต์จากการเชื่อมต่อกับฐานข้อมูลเดียวกัน

โดยออบเจ็กต์ Recordset นี้จะประกอบด้วย คอลเลคชั่น Fields ที่เก็บคอลัมน์ต่างๆ ใน Recordset นั้น และออบเจ็กต์ Fields ที่เก็บคุณสมบัติต่างๆ ในแต่ละคอลัมน์ภายในออบเจ็กต์ Recordset โดยคุณสมบัติที่สำคัญของออบเจ็กต์ Recordset ดังแสดงใน

ตารางที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อคุณสมบัติ	คำอธิบาย
BOF	ถ้ามีค่าเป็น True ตำแหน่งเรคคอร์ดปัจจุบันนั้นเป็นตำแหน่งก่อนเรคคอร์ดแรก
EOF	ถ้ามีค่าเป็น True ตำแหน่งเรคคอร์ดปัจจุบันนั้นเป็นตำแหน่งหลังเรคคอร์ดสุดท้าย
Filter	กำหนดข้อความที่ใช้กรองข้อมูลใน Recordset เช่น "CategorylD = 1" จะกรองข้อมูล ใน Recordset เฉพาะที่มีค่าฟิลด์ CategorylD เท่ากับ 1
MaxRecords	กำหนดจำนวนเรคคอร์ดสูงสุดในผลลัพธ์ที่ส่งกลับมา ถ้าเป็น 0 จะส่งเรคคอร์ดกลับมาทั้งหมด
RecordCount	บอกว่า จำนวนเรคคอร์ดใน Recordset เป็นเท่าไร
Sort	กำหนดฟิลด์ใน Recordset ที่ต้องการให้ Recordset เรียงข้อมูลตาม เช่น "CustomerName DESC" จะเป็นการเรียงข้อมูลตามฟิลด์ CustomerName DESC แบบมากไปน้อย

ตารางที่ 2.5 แสดงคุณสมบัติที่สำคัญของออบเจ็กต์ Recordset

2.7.2.3 รายละเอียดของออบเจ็กต์ Command

ออบเจ็กต์ Command จะใช้เก็บข้อมูลต่างๆ เกี่ยวกับคิวรี เช่น คำสั่งคิวรี , และพารามิเตอร์ที่จะใช้ในการรันคิวรี ในการรันคิวรีด้วยออบเจ็กต์ Command นั้น เราจะต้องส่งพารามิเตอร์ตัวหนึ่งเป็นออบเจ็กต์ Connection ที่เชื่อมต่อกับฐานข้อมูลเราต้องการนำผลลัพธ์มาใช้

ออบเจ็กต์ Command ยังสามารถรับพารามิเตอร์คิวรีที่ให้ผลลัพธ์ของคิวรีแตกต่างกันไปโดยผลลัพธ์จะขึ้นอยู่กับค่าในพารามิเตอร์ที่กำหนดไป และยังสามารถรัน Store Procedure ที่เป็นคำสั่งในการทำงานกับข้อมูลที่อยู่บนเครื่อง Server ได้อีกด้วย คุณสมบัติที่สำคัญของออบเจ็กต์ Command แสดงดังตารางที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อคุณสมบัติ	คำอธิบาย
ActiveConnection	ใช้กำหนดการเชื่อมต่อในรูป ConnectionString
CommandText	กำหนดข้อความคำสั่ง SQL หรือชื่อ Store Procedure ที่จะรันด้วยเมดถอด Execute
CommandTimeout	กำหนดช่วงเวลาที่ให้รอคิวรีที่กำลังรันอยู่ ถ้าเกินเวลานี้จะยังรันไม่เสร็จ จะเกิดขอผิดพลาดขึ้นมา
CommandType	กำหนดชนิดของคำสั่งที่เรากำหนดในคุณสมบัติ CommandText ซึ่งมีค่าต่างๆ ดังต่อไปนี้ - adCmdText บอกว่าเป็นคำสั่ง SQL - adCmdTable บอกว่าเป็นชื่อตารางที่มีในฐานข้อมูล - adCmdTableDirect คล้ายกันแบบ adCmdTable แต่ถ้า Provider นั้นสนับสนุนการเปิดตารางโดยตรงจะทำให้ประสิทธิภาพในการทำงานดีขึ้น - adCmdStoredProc บอกว่าเป็นชื่อ Store Procedure - adCmdUnknown ใช้กำหนดค่าเมื่อเราไม่ทราบชนิดของคำสั่งที่ใช้แน่นอน
Name	เป็นชื่อของออบเจ็กต์ Command ที่เราสามารถนำไปใช้อ้างอิงได้

ตารางที่ 2.6 แสดงคุณสมบัติที่สำคัญของออบเจ็กต์ Command

2.7.2.4 รายละเอียดของออบเจ็กต์ Field

เป็นออบเจ็กต์ที่ใช้เพื่อนำข้อมูลในฟิลด์ต่างๆ ใน Recordset ออกมาใช้งาน ซึ่งออบเจ็กต์ Recordset จะมีคอลเล็กชัน Fields ที่เก็บฟิลด์ต่างๆ ทั้งหมดไว้ ซึ่งในฟิลด์นี้จะมีคุณสมบัติต่างๆ ที่บอกถึงชนิดข้อมูล , ค่าข้อมูลที่เก็บอยู่ โดยคุณสมบัติที่สำคัญของออบเจ็กต์ Field แสดงในตารางที่ 2.7

ชื่อคุณสมบัติ	คำอธิบาย
Name	เป็นชื่อของฟิลด์
Type	เป็นชนิดข้อมูลของฟิลด์
Value	เป็นค่าของข้อมูลในฟิลด์
Attribute	บอกถึงลักษณะต่างๆ ของฟิลด์ เช่น ค่าในฟิลด์สามารถแก้ไขได้หรือไม่

ตารางที่ 2.7 แสดงคุณสมบัติที่สำคัญของออบเจ็กต์ Field

2.7.2.5 รายละเอียดของออบเจ็กต์ Parameter

ออบเจ็กต์นี้จะใช้สำหรับสร้างออบเจ็กต์ Command แบบมีพารามิเตอร์ขึ้นมา ซึ่งต้องกำหนดชนิดข้อมูล, ทิศทางการส่งพารามิเตอร์ โดยแสดงคุณสมบัติที่สำคัญของออบเจ็กต์ Parmeter ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.8

ชื่อคุณสมบัติ	คำอธิบาย
Name	เป็นชื่อของพารามิเตอร์
Type	เป็นชนิดข้อมูลของพารามิเตอร์
Value	เป็นค่าของข้อมูลในพารามิเตอร์
Attribute	บอกถึงลักษณะต่างๆ ของพารามิเตอร์ เช่นค่าในพารามิเตอร์เป็นค่า Null ได้หรือไม่
Direction	เป็นทิศทางของพารามิเตอร์ ซึ่งมีค่าต่างๆ ดังต่อไปนี้ - adParamInput เป็นพารามิเตอร์ที่ส่งค่าเข้าไปอย่างเดียว - adParamOutput เป็นพารามิเตอร์ที่ส่งค่าออกอย่างเดียว - adParamInputOutput เป็นพารามิเตอร์ที่ส่งค่าเข้าไป และรับค่าออกมา - adParamReturn เป็นพารามิเตอร์ที่ได้จากค่าที่ส่งกลับมาจากการทำงาน - adParamUnknown ใช้กำหนดค่าในกรณีที่ไม่ทราบทิศทางของพารามิเตอร์ที่แน่นอน

ตารางที่ 2.8 แสดงคุณสมบัติที่สำคัญของออบเจ็กต์ Parameter

2.7.2.6 รายละเอียดของออบเจ็กต์ Error

จะบอกข้อผิดพลาดต่างๆ ที่เกิดขึ้นจาก Provider แต่เนื่องจากคำสั่งหนึ่งอาจจะทำให้เกิดข้อผิดพลาดได้มากกว่าหนึ่ง ดังนั้นคอลเลคชั่น Errors จะสามารถเก็บออบเจ็กต์ข้อผิดพลาดได้มากกว่าหนึ่ง เพื่อให้เราตรวจสอบได้ว่ามีข้อผิดพลาดใดบ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

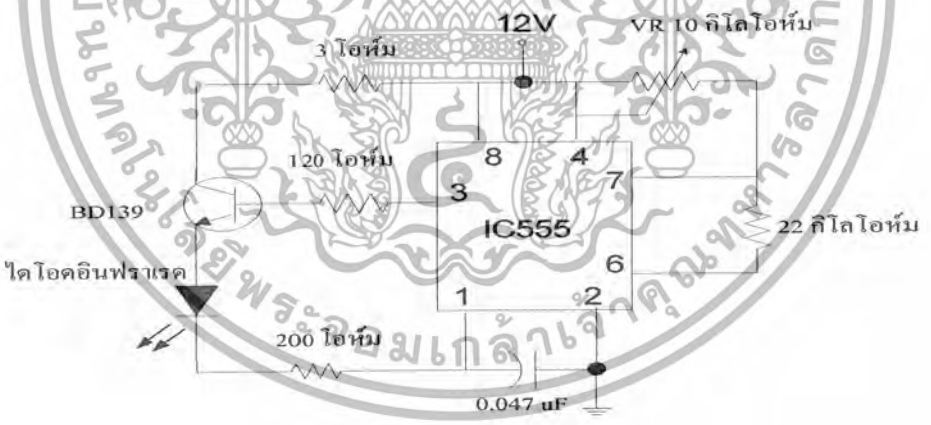
บทที่ 3
การสร้างและแนวความคิดในการออกแบบ

ในการออกแบบสร้างชุดบริการข้อมูลอัตโนมัติในปริญญานิพนธ์ฉบับนี้ ประกอบด้วยส่วนที่สำคัญ 3 ส่วนด้วยกันคือ ส่วนของวงจรอินฟราเรดเซนเซอร์ที่ส่งสัญญาณพัลส์ไปยังส่วนที่ 2 คือ ส่วนของชุดควบคุมไมโครคอนโทรลเลอร์ ซึ่งชุดควบคุมไมโครคอนโทรลเลอร์นี้จะเชื่อมต่อกับส่วนที่ 3 คือ คอมพิวเตอร์ผ่านทางพอร์ตอนุกรม RS-232 โดยชุดควบคุมไมโครคอนโทรลเลอร์จะทำหน้าที่รับสัญญาณพัลส์จากวงจรอินฟราเรดเซนเซอร์ ที่แปลงสัญญาณแสงอินฟราเรดเป็นสัญญาณทางไฟฟ้า ส่วนของคอมพิวเตอร์จะรับข้อมูลจากชุดควบคุมไมโครคอนโทรลเลอร์ เพื่อแสดงผลของข้อมูลบนหน้าจอมอนิเตอร์ พร้อมทั้งส่งข้อมูลกลับมายังชุดควบคุมไมโครคอนโทรลเลอร์เมื่อข้อมูลนั้นสิ้นสุดลงด้วย

3.1 การออกแบบวงจรอินฟราเรดเซนเซอร์

สามารถแบ่งการออกแบบวงจรได้เป็น 2 ส่วนดังนี้

3.1.1 ส่วนของวงจรภาคส่งอินฟราเรดเซนเซอร์



รูปที่ 3.1 แสดงวงจรภาคส่งอินฟราเรดเซนเซอร์

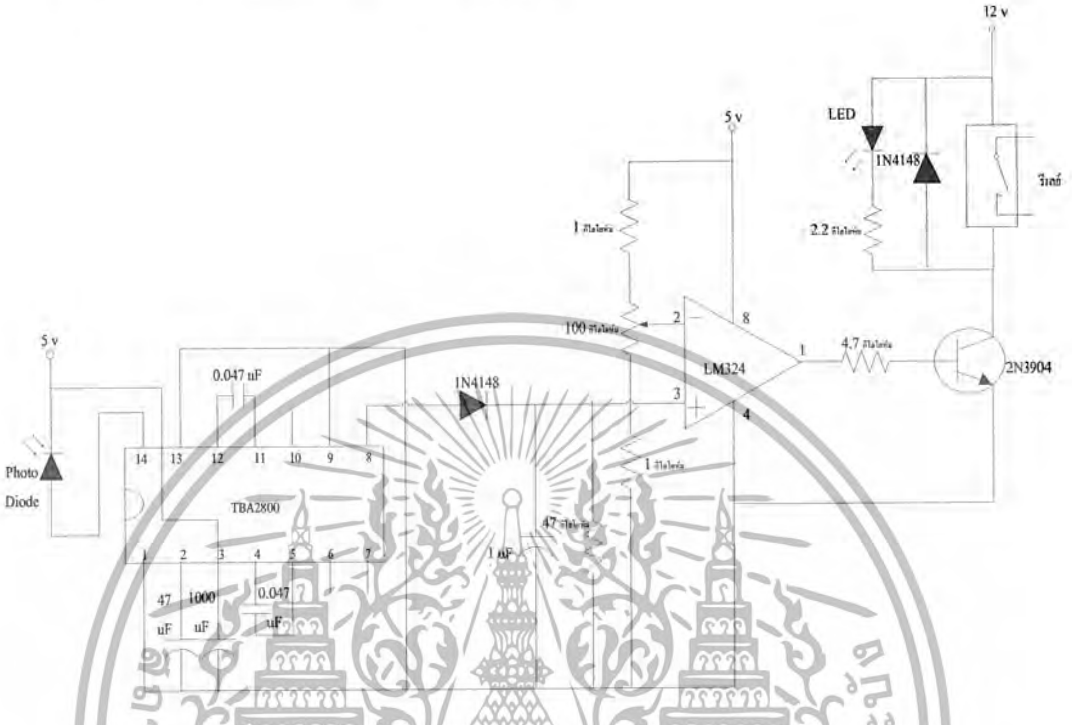
หลักการทำงานของภาคส่งอินฟราเรดเซนเซอร์

เริ่มจากภาคส่งส่งอินฟราเรดเซนเซอร์ใช้ ไอซี NE555 จัดเป็นวงจรอะสเตเบิล มัลติไวเบรเตอร์ ซึ่งวงจรในลักษณะนี้ เป็นวงจรสร้างสัญญาณ สี่เหลี่ยม (Squre Wave) โดยมีความต้านทานเกือกม้า 10 K ที่สามารถปรับเปลี่ยนความถี่ที่ต้องการได้ เอาท์พุท ที่ได้จะออกมาที่ขา 3 ของ ไอซีและผ่านมาที่ขาเบส ของทรานซิสเตอร์ เบอร BD139 ส่งต่อมายังขาอิมิตเตอร์ ของทรานซิสเตอร์และผ่านอินฟราเรดเซนเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งออกไป และเนื่องจากที่ใช้ อินฟราเรดเซนเซอร์ถึง 3 ตัว เพื่อให้มีกำลังส่งเพิ่มขึ้น และได้ระยะไกลออกไปอีกด้วย

3.1.2 ส่วนของวงจรภาครับอินฟราเรดเซนเซอร์



รูปที่ 3.2 แสดงวงจรภาครับอินฟราเรดเซนเซอร์

หลักการทำงานของภาครับอินฟราเรดเซนเซอร์

จากวงจรเมื่อมีความถี่ที่ส่งมาจากตัวอินฟราเรดเซนเซอร์ ผ่านเข้ามาตัวรับอินฟราเรดเซนเซอร์ ก็ยังไม่สามารถนำมาใช้งานได้โดยตรง เนื่องจากตัวรับอินฟราเรดเซนเซอร์สามารถรับแสงได้หลากหลาย โดยเฉพาะแสงที่มาจากนิออน (50 Hz) หรือแสงทั่วไปที่มีความไวมากจึงจำเป็นจะต้องมีวงจรกรองความถี่ (กรองเฉพาะความถี่ที่ใช้งาน) โดยในวงจรนี้ ใช้ไอซี เบอร์ TBA2800 เนื่องจากมีการใช้งานง่ายเพียงแค่ปรับเปลี่ยนค่าตัวเก็บประจุที่ต่อระหว่างขา 10, ขา 11 และขา 3, ขา 4 ของ ไอซีก็สามารถได้ความถี่ที่ต้องการใช้งาน สัญญาณจะถูกเรคติไฟครึ่งไซเคิล สัญญาณที่ได้จะถูกเปลี่ยนโดยตัวเก็บประจุค่า 1 ไมโครฟารัด จากพัลซ์ที่ดีมอดดูเรดแล้วมาเป็น ระดับแรงดัน ซึ่งระดับแรงดันจะมีค่ามากหรือน้อยก็จะขึ้นอยู่กับระยะของวัตถุที่สะท้อนตามระยะห่างจากอินฟราเรด แรงดันได้มาจากตัวเก็บประจุค่า 1 ไมโครฟารัด ก็จะผ่านมายัง ไอซีออปแอมป์เบอร์ LM358 ขาที่3 มาเปรียบเทียบแรงดันระหว่างขา 2 เกือกม้า ค่า 100 กิโลโอห์ม จะเป็นตัวกำหนดแรงดันที่ขา 2 ที่เราต้องการสามารถปรับเปลี่ยนได้ เมื่อแรงดันขา 3 มากกว่าขาที่ 2 เมื่อใด ขาที่ 1 จะมีเอาท์พุตออกจาก Trig จะเป็นไฮท์ (+4V DC)เพื่อจะนำกระแสสู่ทรานซิสเตอร์ เบอร์ 2N3904 ให้ทำงานจากนั้นรีเลย์จะทำงาน (ON)

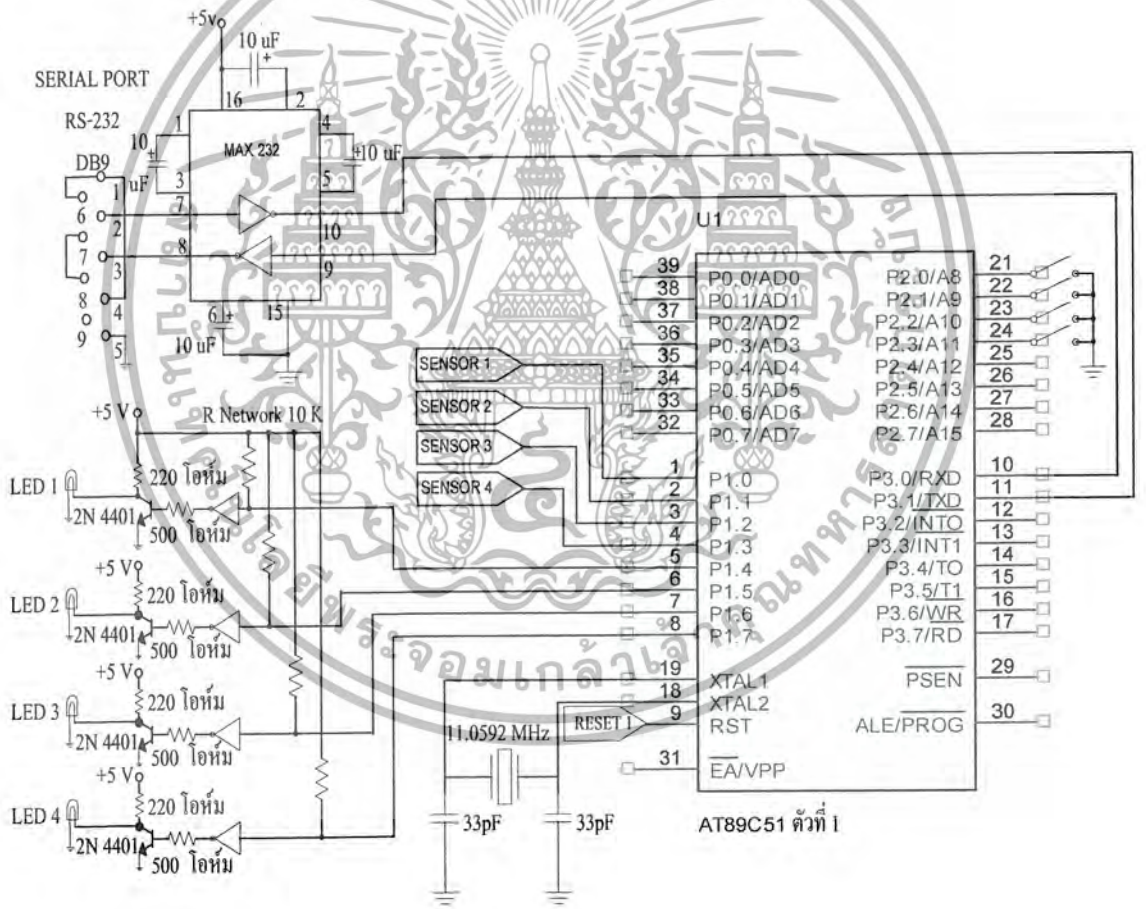
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบชุดควบคุมไมโครคอนโทรลเลอร์

ในการออกแบบชุดควบคุมไมโครคอนโทรลเลอร์จะประกอบไปด้วยวงจรไมโครคอนโทรลเลอร์ทั้งหมด 4 ชุด ที่ใช้ควบคุมการทำงานของวงจรอินฟราเรดเซนเซอร์ที่แต่ต่างกัน โดยมีส่วนของการออกแบบที่สำคัญ 2 ส่วน ดังนี้

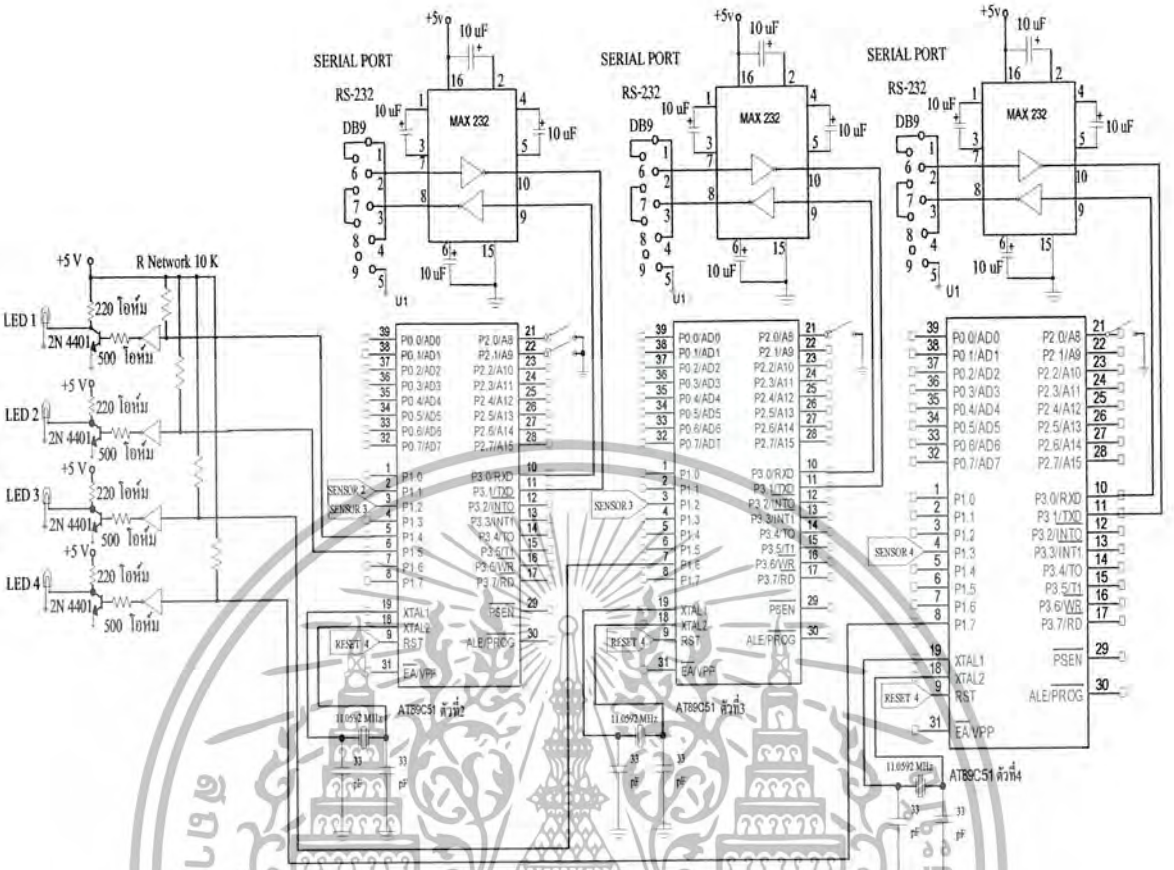
3.2.1 ส่วนการออกแบบฮาร์ดแวร์

โครงงานนี้จะนำไมโครคอนโทรลเลอร์เบอร์AT89C51 ทั้งหมด 4 ตัว มาประกอบเป็นชุดควบคุมการรับสัญญาณพัลส์จากวงจรอินฟราเรดเซนเซอร์ รวมไปถึงการส่งข้อมูลและรับข้อมูลกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมในมาตรฐานอาร์เอสสองสามสอง โดยใช้ไอซี MAX232 ทำหน้าที่ในการแปลงระดับสัญญาณให้สามารถส่งข้อมูลและรับข้อมูลผ่านทางพอร์ตอนุกรมได้ ซึ่งวงจรได้แสดงดังรูปที่ 3.3 และ รูปที่ 3. 4



รูปที่ 3. 3 แสดงวงจรไมโครคอนโทรลเลอร์ ตัวที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3. 4 แสดงวงจรไมโครคอนโทรลเลอร์ ตัวที่ 2,ตัวที่ 3 และ ตัวที่ 4

หน้าที่และการต่อขาต่างๆกับอุปกรณ์ภายนอกไมโครคอนโทรลเลอร์จะมีดังนี้

- ขา 1 (P1.0) บิตนี้ทำหน้าที่เป็นอินพุตรับคำสั่งจากวงจรอินฟราเรดเซนเซอร์ตัวที่ 1 เพื่อแสดงข้อมูลชุดที่ 1
- ขา 2 (P1.1) บิตนี้ทำหน้าที่เป็นอินพุตรับคำสั่งจากวงจรอินฟราเรดเซนเซอร์ตัวที่ 2 เพื่อแสดงข้อมูลชุดที่ 2
- ขา 3 (P1.2) บิตนี้ทำหน้าที่เป็นอินพุตรับคำสั่งจากวงจรอินฟราเรดเซนเซอร์ตัวที่ 3 เพื่อแสดงข้อมูลชุดที่ 3
- ขา 4 (P1.3) บิตนี้ทำหน้าที่เป็นอินพุตรับคำสั่งจากวงจรอินฟราเรดเซนเซอร์ตัวที่ 1 เพื่อแสดงข้อมูลชุดที่ 1
- ขา 5 ถึง ขา 8 (P1.4 ถึง P1.7) บิตนี้ทำหน้าที่เป็นเอาท์พุตแสดงผลการทำงานของวงจรอินฟราเรดเซนเซอร์แต่ละตัว เพื่อบอกว่ากำลังแสดงข้อมูลแต่ละชุดอยู่
- ขา 21ถึง ขา 24 (P2.0 ถึง P2.3) บิตนี้ทำหน้าที่เป็นอินพุตรับคำสั่งจากสวิตซ์แต่ละตัว เพื่อเริ่มต้นการทำงานของไมโครคอนโทรลเลอร์แต่ละตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา 9 (RESET) เป็นขารีเซตของวงจรไมโครคอนโทรลเลอร์แต่ละตัว จะต่อตัวต้านทานลงกราวด์และตัวเก็บประจุคร่อมสวิตซ์ที่ต่อกับไฟ 5 โวลต์
- ขา 10 (RxD) ใช้ทำหน้าที่เป็นตัวรับข้อมูลอนุกรม เพื่อรับข้อมูลจากคอมพิวเตอร์
- ขา 10 (TxD) ใช้ทำหน้าที่เป็นตัวส่งข้อมูลอนุกรม เพื่อส่งข้อมูลไปยังคอมพิวเตอร์
- ขา 18 และขา 19 (XTAL1และXTAL2) เป็นขาที่ต่อกับชุดออสซิสเลเตอร์ โดยใช้คริสตอล 11.0592 เมกกะเฮิร์ท และต่อตัวเก็บประจุ 33 ฟิกโคฟารัด ลงกราวด์

3.2.2 ส่วนการออกแบบโปรแกรม

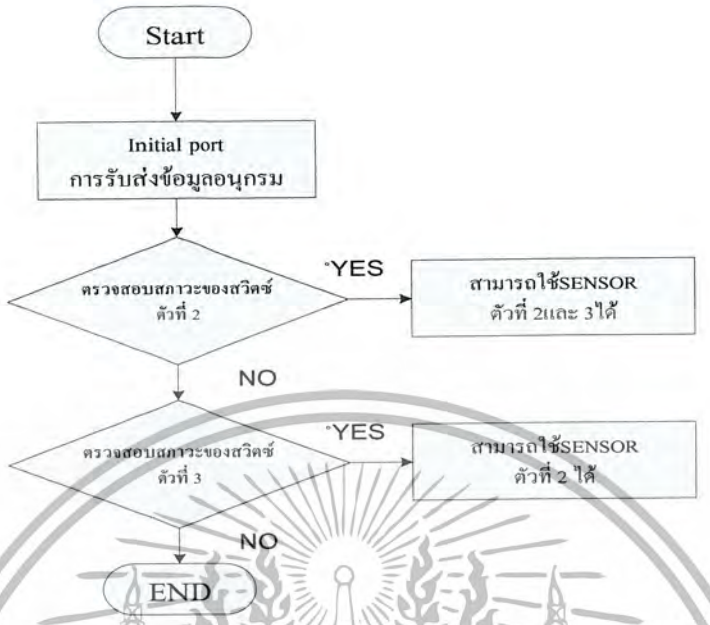
ส่วนของการออกแบบโปรแกรมของไมโครคอนโทรลเลอร์แต่ละตัว จะใช้ภาษาแอสเซมบลีในการเขียน ซึ่งจะส่งข้อมูลและรับข้อมูล 8 บิต เป็นเลข ฐาน 16 คือเลข 8, 4, 2 และ 1 ไปยังเครื่องคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมอาร์เอสสองสามสอง โดยมีการทำงานของโปรแกรมแสดงให้เห็นดังโฟลว์ชาร์ต ต่อไปนี้

3.2.2.1 การออกแบบโปรแกรมของไมโครคอนโทรลเลอร์ตัวที่ 1

รูปที่ 3.5 แสดงโฟลว์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์ตัวที่ 1

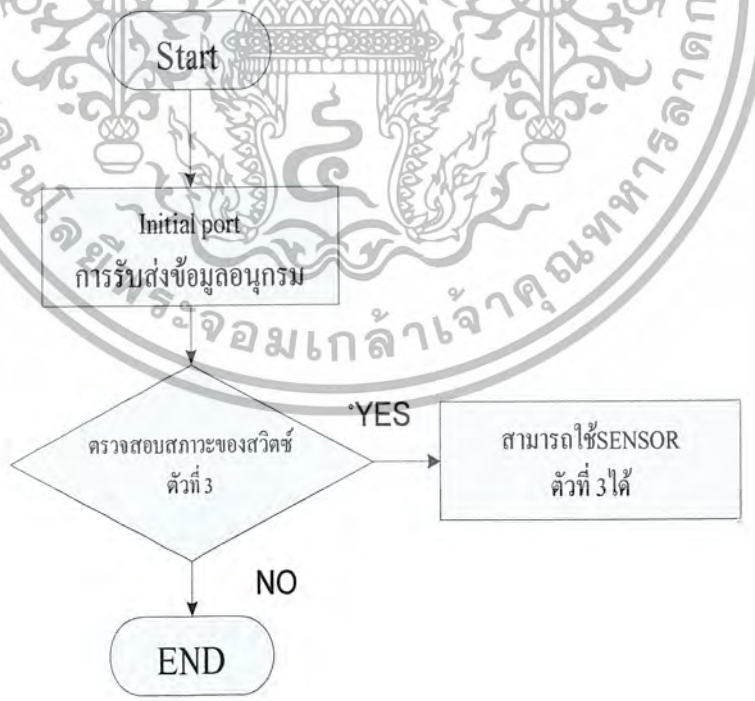
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.2 การออกแบบโปรแกรมของไมโครคอนโทรลเลอร์ตัวที่ 2



รูปที่ 3.6 แสดงโฟลว์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์ตัวที่ 2

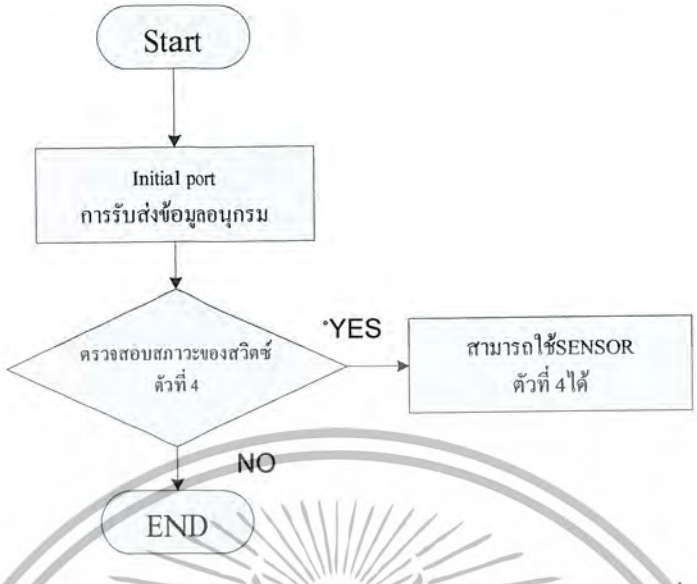
3.2.2.3 การออกแบบโปรแกรมของไมโครคอนโทรลเลอร์ตัวที่ 3



รูปที่ 3.7 แสดงโฟลว์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์ตัวที่ 3

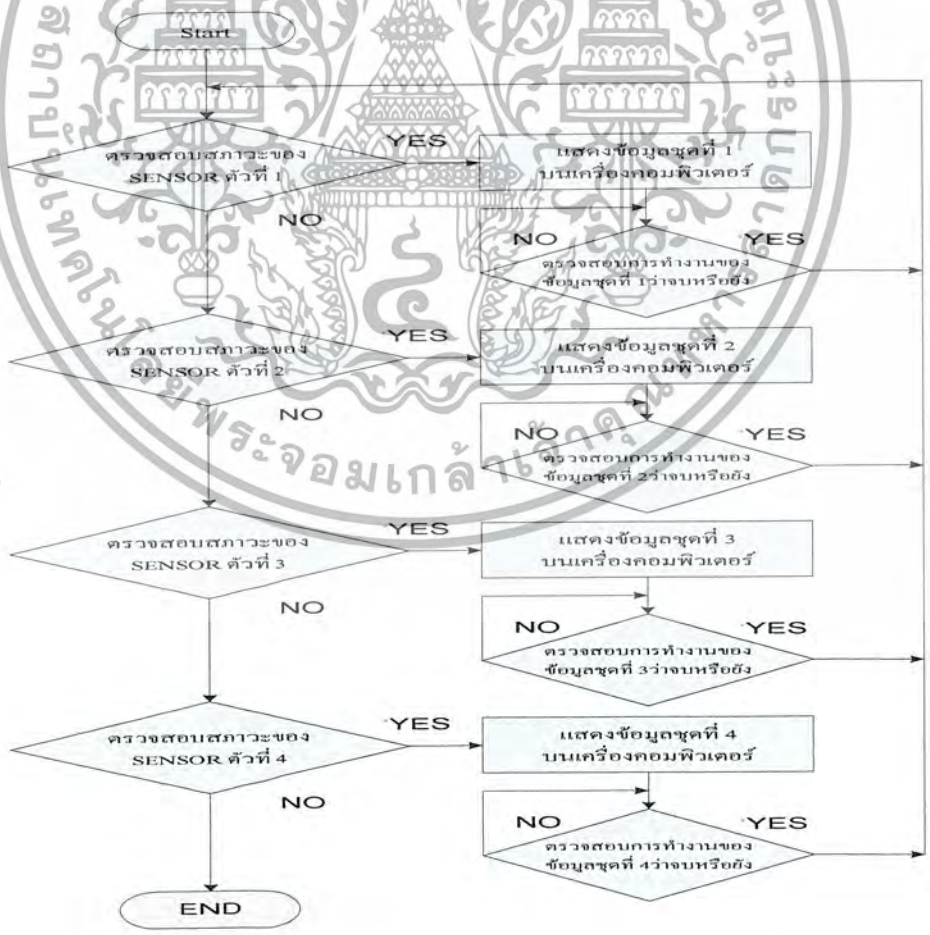
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.4 การออกแบบโปรแกรมของไมโครคอนโทรลเลอร์ตัวที่ 4



รูปที่ 3.8 แสดงโฟลว์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์ตัวที่ 4

3.2.2.5 การออกแบบโปรแกรมควบคุมการส่งและรับข้อมูลทางพอร์ตอนุกรม



รูปที่ 3.9 แสดงโฟลว์ชาร์ตการส่งและรับข้อมูลทางพอร์ตอนุกรม

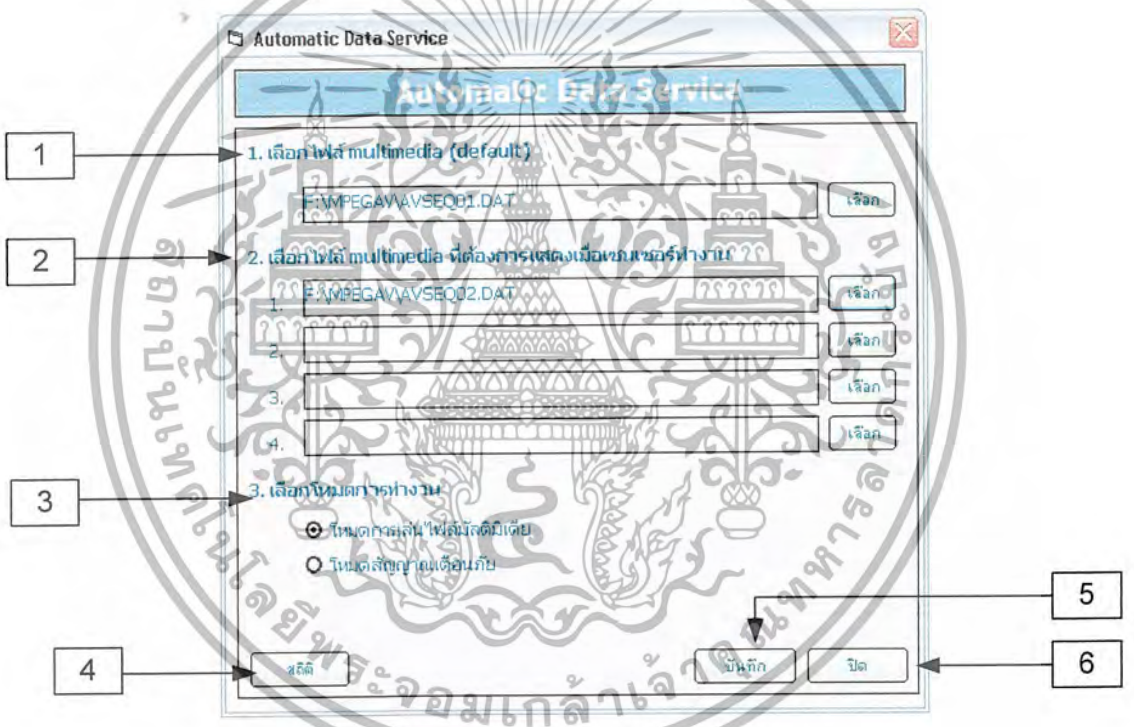
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบโปรแกรมคอมพิวเตอร์

ในส่วนของการออกแบบโปรแกรมคอมพิวเตอร์จะใช้โปรแกรมวิชวลเบสิก เวอร์ชั่น 6.0 ที่เป็นตัวแสดงผลบนหน้าจอมอนิเตอร์, รับข้อมูลและส่งข้อมูลจากไมโครคอนโทรลเลอร์แต่ละตัว รวมไปถึงการจัดเก็บข้อมูลที่ไมโครคอนโทรลเลอร์แต่ละตัวส่งข้อมูลเข้ามา โดยในส่วนของการออกแบบโปรแกรมวิชวลเบสิก เวอร์ชั่น 6.0 จะประกอบไปด้วยฟอร์ม(Form)ต่างๆ ที่เป็นพื้นที่ที่ใช้สำหรับการออกแบบ โดยการนำเอาคอนโทรลต่างๆมาวางในฟอร์ม ซึ่งการออกแบบฟอร์มและการทำงานของโปรแกรมในโครงงานนี้ สามารถแบ่งออกได้ 3 ส่วนดังนี้

3.3.1 ส่วนของโปรแกรมหน้าจอควบคุมหลัก

ในส่วนของโปรแกรมหน้าจอควบคุมหลักจะมีฟอร์มดังรูปที่ 3.10 ซึ่งจะมีรูปแบบการทำงานในส่วนต่างๆ ดังนี้



รูปที่ 3.10 แสดงส่วนของหน้าจอควบคุมหลัก

1) ใช้เลือกไฟล์มัลติมีเดีย (default)ที่อยู่ในเครื่องคอมพิวเตอร์ โดยไฟล์มัลติมีเดียที่ถูกเลือกขึ้นมาจะแสดงชื่อไฟล์มัลติมีเดียนั้นขึ้นมา ซึ่งการทำงานของไฟล์ดิฟอลต์นี้จะแสดงข้อมูลในลักษณะวนการทำงานไปเรื่อยๆจนกว่าจะได้รับข้อมูลจากไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรมจึงจะหยุดแสดงข้อมูลนั้น และจะแสดงข้อมูลในลักษณะวนการทำงานนี้อีกครั้งเมื่อมีการส่งข้อมูลของไมโครคอนโทรลเลอร์อีกครั้ง
2) ใช้เลือกไฟล์มัลติมีเดียที่ต้องการแสดงข้อมูลเมื่อได้รับข้อมูลจากไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรม โดยไฟล์มัลติมีเดียที่ถูกเลือกขึ้นมา จะแสดงชื่อไฟล์มัลติมีเดียนั้นขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งการทำงานของไฟล์นี้จะแสดงข้อมูลรอบเดียว กล่าวคือเมื่อแสดงข้อมูลนั้นเสร็จก็จะส่งข้อมูลกลับยังไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรม เพื่อจะทำการรับข้อมูลขึ้นมาใหม่อีกครั้ง โดยในส่วนนี้จะสามารถเลือกไฟล์มัลติมีเดียได้ถึง 4 ไฟล์ทำงาน

3) ใช้เลือกโหมดการทำงานจะมี 2 โหมดการทำงาน คือ โหมดการเล่นไฟล์มัลติมีเดีย และโหมดสัญญาณเตือนภัย โดยในสภาวะปกติต้องเลือกโหมดการเล่นไฟล์มัลติมีเดีย ส่วนโหมดสัญญาณเตือนภัยเป็นการประยุกต์ใช้เป็นสัญญาณกันขโมย โดยการทำงานของโหมดสัญญาณเตือนภัยนี้จะมีสัญญาณเตือนดังขึ้นเป็นเวลา 5 นาทีจึงหยุด เมื่อได้รับข้อมูลจากไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรม และจะดังขึ้นอีกครั้งเมื่อมีข้อมูลใหม่ขึ้นมา

4) ปุ่มสถิติจะมีไว้สำหรับเรียกดูข้อมูลต่างๆที่แสดงในฐานข้อมูล

5) ปุ่มบันทึกไฟล์มัลติมีเดียที่ต้องการลงในฐานข้อมูลเมื่อเลือกไฟล์มัลติมีเดียเรียบร้อยแล้ว

6) ปุ่มปิดการทำงานหลังจากที่ทำการเลือกไฟล์มัลติมีเดียและทำการบันทึกแล้ว

3.3.2 ส่วนของโปรแกรมในการจัดเก็บข้อมูลภายในฐานข้อมูล

ส่วนนี้จะเป็นการจัดเก็บข้อมูลที่อยู่ภายในฐานข้อมูล โดยใช้ ActiveX Data Objects (ADO) ซึ่งเป็นเทคโนโลยีการติดต่อกับระบบการจัดการฐานข้อมูลที่ใช้งานได้ง่าย และสามารถใช้งานร่วมกับโปรแกรมการจัดการฐานข้อมูลชนิดต่างๆได้มากมาย ในส่วนของการจัดเก็บข้อมูลภายในฐานข้อมูลดังรูปที่ 3.11 จะแสดงขึ้นมาเมื่อทำการเลือกปุ่มสถิติในส่วนของหน้าจอควบคุมหลัก ซึ่งจะมีรูปแบบการทำงานและฟอร์มในส่วนต่างๆ ดังนี้

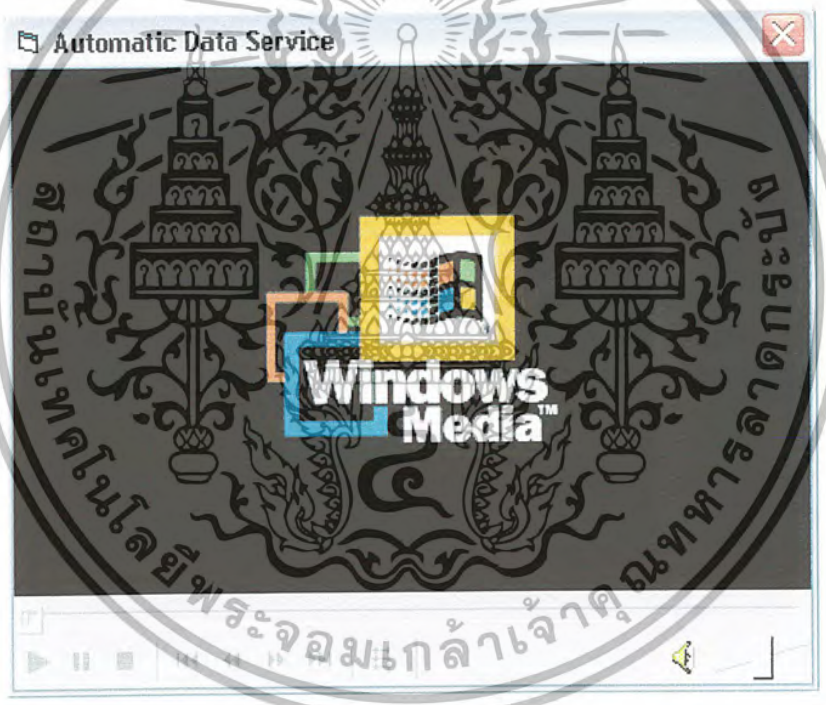
รูปที่ 3.11 แสดงส่วนของการจัดเก็บข้อมูลภายในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) เป็นส่วนของการแสดงข้อมูลไฟล์มัลติมีเดียที่ทำการบันทึกไว้ โดยจะประกอบไปด้วยลำดับที่บันทึกไฟล์มัลติมีเดียเข้ามา, ลำดับของไฟล์มัลติมีเดียที่ถูกบันทึก และชื่อไฟล์มัลติมีเดียนั้น นอกจากนี้ยังมีส่วนของการลบข้อมูลที่ต้องการทั้งแบบลบที่ละข้อมูลและแบบลบทั้งหมด
2) เป็นส่วนของการแสดงข้อมูลทางสถิติของแต่ละไฟล์มัลติมีเดีย ที่บอกวันและเวลาของไฟล์มัลติมีเดียแต่ละไฟล์ที่มีการแสดงข้อมูลออกไป โดยมีส่วนของการลบข้อมูลย่อยที่ต้องการทีละไฟล์มัลติมีเดียและ ลบข้อมูลย่อยออกทั้งหมด
3) ปุ่มปิดการทำงานหลังจากที่ทำการดูข้อมูลที่แสดง หรือจัดการลบข้อมูลเรียบร้อยแล้ว

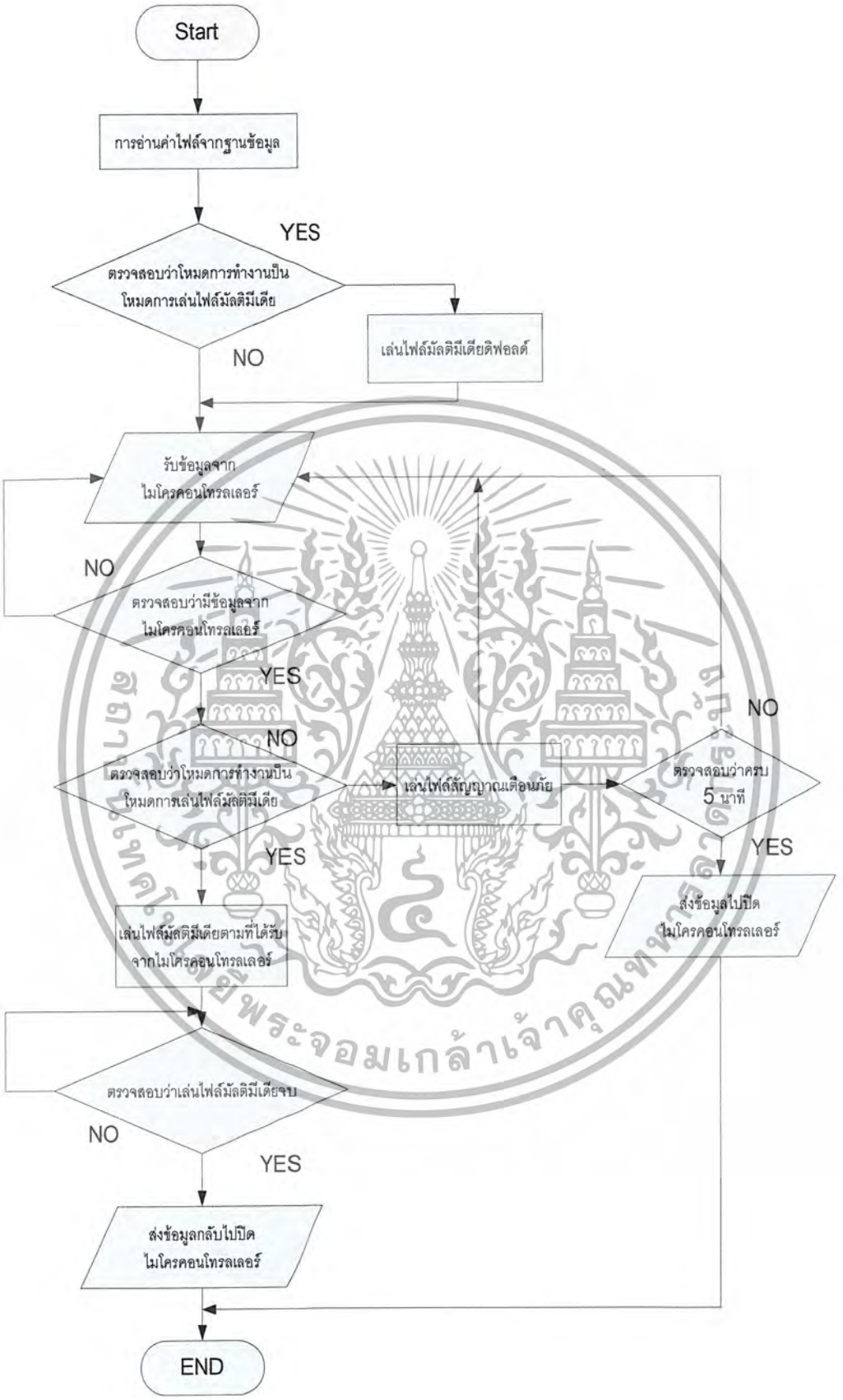
3.3.3 ส่วนของโปรแกรมแสดงผลของข้อมูล

ในส่วนโปรมแกรมการแสดงผลของข้อมูลจะใช้คอนโทรล Windows Media Player มาเป็นตัวแสดงผล โดยคอนโทรล Windows Media Player นี้จะเล่นได้ทั้งไฟล์หนังและเพลงประเภท Mp3



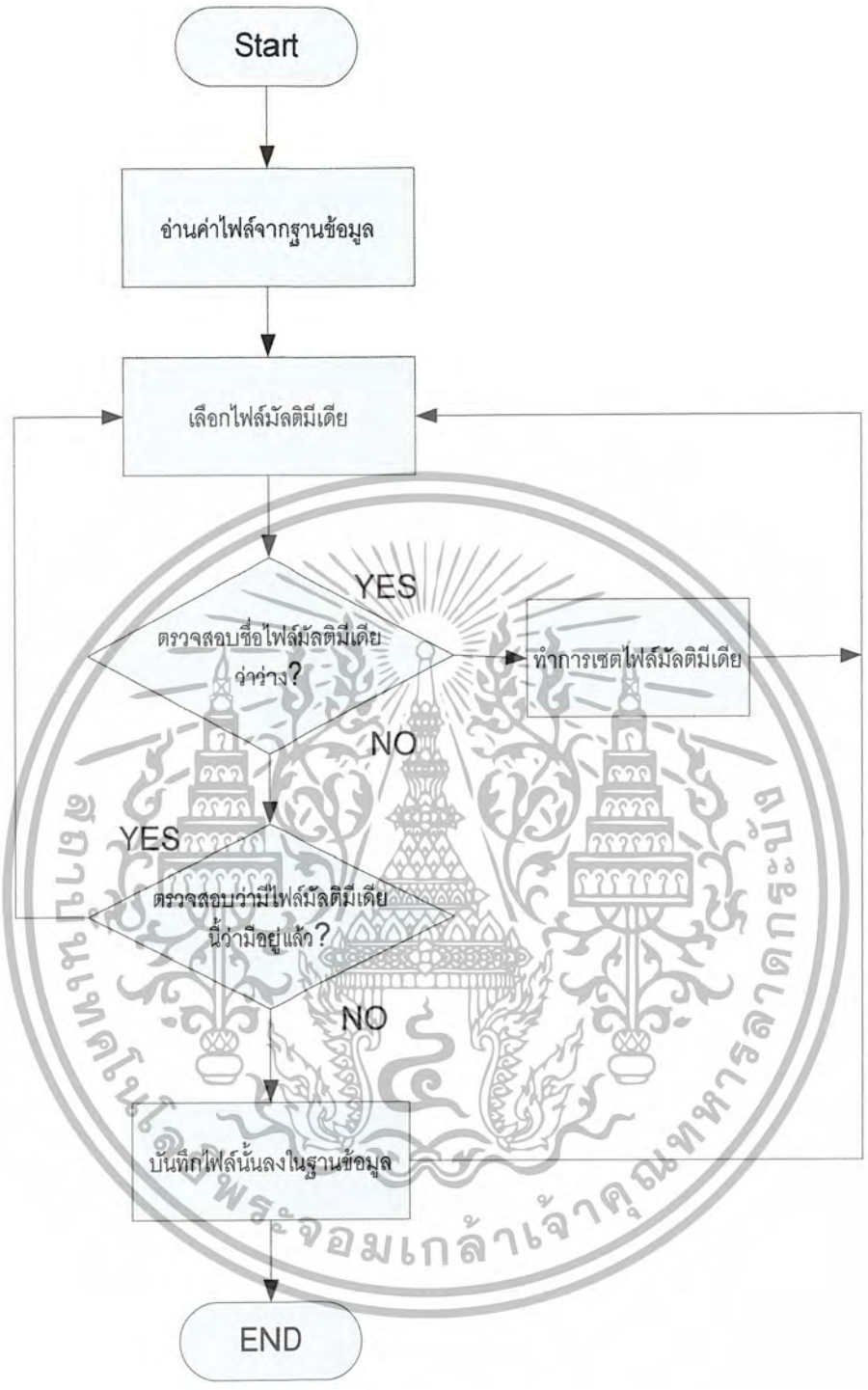
รูปที่ 3. 12 แสดงส่วนของการแสดงผลของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงโฟลว์ชาร์ต การทำงานของการแสดงผลในโหมดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดงโฟลว์ชาร์ตการเลือกไฟล์การทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

4.1 อุปกรณ์ที่ใช้ในการทดลอง

4.1.1 ออสซิลโลสโคป

4.1.2 ฟังก์ชั่นเจนเนอร์เรเตอร์

4.1.3 แหล่งจ่ายไฟตรง

4.1.4 แผ่นทดลองวงจร

4.1.5 เครื่องคอมพิวเตอร์

4.1.6 บอร์ดเบรีน์โปรมแกรมไมโครคอนโทรลเลอร์

4.2 วงจรภาคส่งอินฟราเรดเซนเซอร์

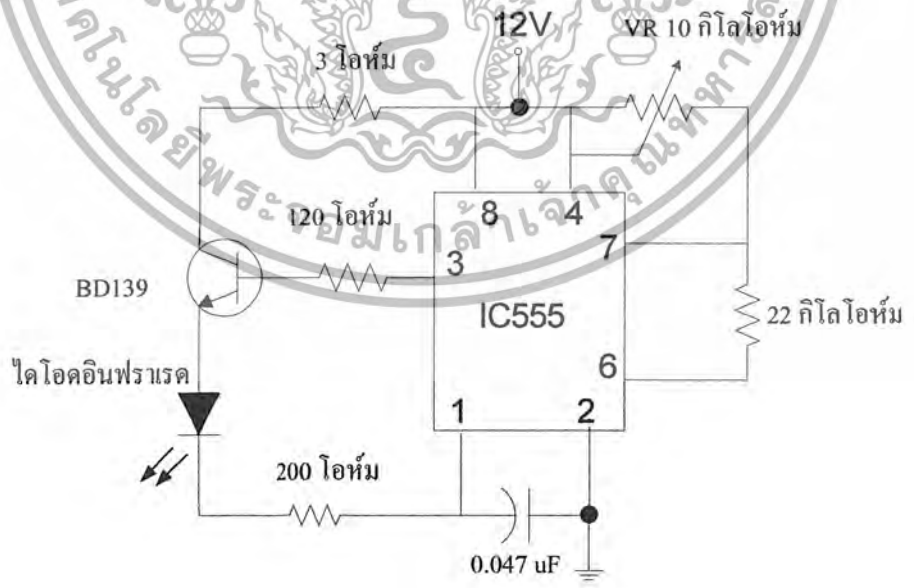
วัตถุประสงค์

เพื่อศึกษาการทำงานของภาคส่งอินฟราเรด

ขั้นตอนการทดลอง

4.2.1 ต่อวงจรตามรูปที่ 4.1

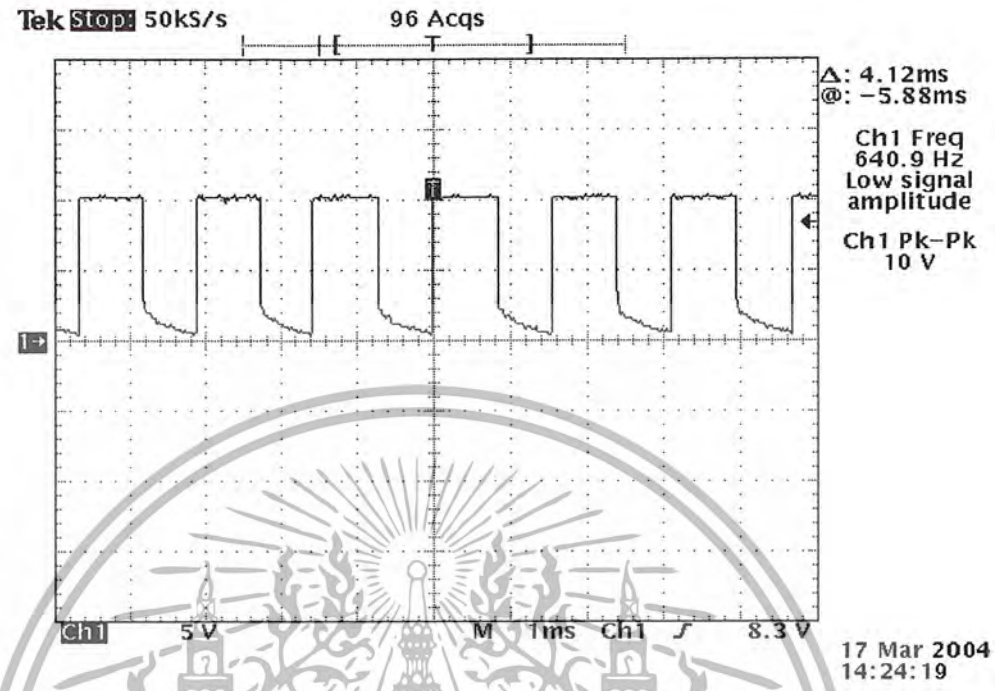
4.2.2 ทำการบันทึกสัญญาณเอาท์พุตด้วยออสซิลโลสโคปและทำการบันทึกผล



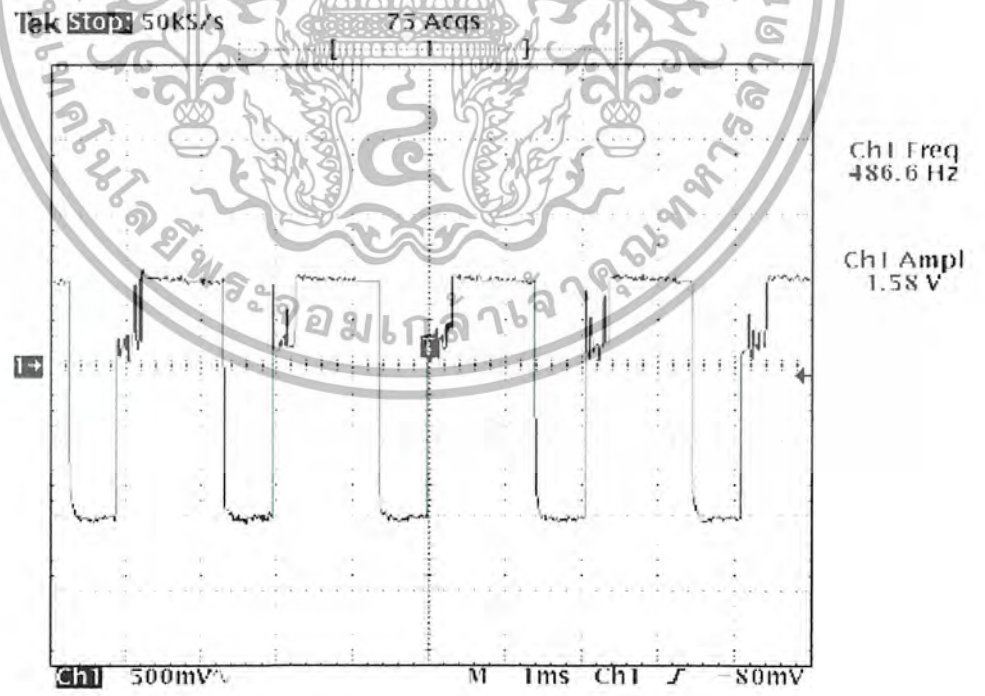
รูปที่ 4.1 แสดงวงจรภาคส่งอินฟราเรดเซนเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง



รูปที่ 4.2 แสดงสัญญาณที่ขาอิมิตเตอร์ของทานซิสเตอร์ เทียบกับกราวด์



รูปที่ 4.3 แสดงสัญญาณที่ตกคร่อมไดโอดอินฟราเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 วงจรภาครับอินฟราเรดเซนเซอร์

วัตถุประสงค์

เพื่อศึกษาการทำงานของภาครับอินฟราเรด

ขั้นตอนการทดลอง

4.2.1 ต่อวงจรตามรูปที่ 4.4

4.2.2 ทำการวัดแรงดันตกคร่อมและสังเกตุการเปลี่ยนแปลงแรงดันที่ขา 8 ของไอซีเบอร์ TBA2800

4.2.3 วัดและบันทึกผลระยะทางสูงสุดและต่ำสุดที่ภาครับอินฟราเรดสามารถทำงานได้

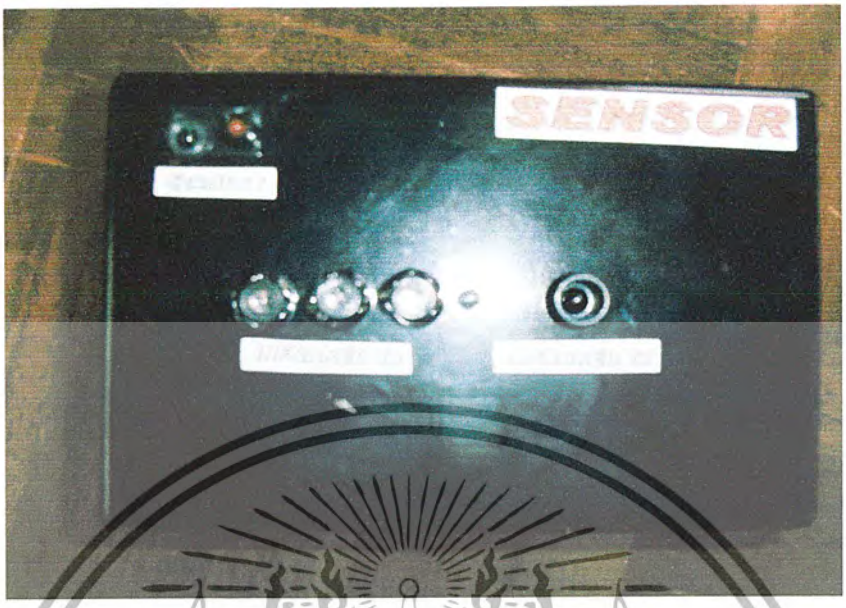
รูปที่ 4. 4 แสดงวงจรภาครับอินฟราเรดเซนเซอร์

ผลการทดลอง

ค่าแรงดันที่ขา 2 ของไอซี LM386 (โวลท์)	ค่าแรงดันที่ขา 3 ของไอซี LM358 (โวลท์)	ระยะทางที่ทดสอบได้ (ฟุต)	สภาวะวงจร
1.5	1.6	1	รีเลย์ทำงาน
1.5	1.4	มากกว่า 1	รีเลย์ไม่ทำงาน

ตารางที่ 4.1 แสดงผลการทดลองภาครับอินฟราเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงภาพถ่ายตัวอินฟราเรดเซนเซอร์

รูปที่ 4.6 แสดงภาพถ่ายด้านในภายในตัวอินฟราเรดเซนเซอร์

4.4 ชุดควบคุมไมโครคอนโทรลเลอร์

วัตถุประสงค์

เพื่อศึกษาการแสดงผลของแอลอีดีในชุดควบคุมไมโครคอนโทรลเลอร์แต่ละตัว เมื่อมีสัญญาณพัลส์จากวงจรอินฟราเรดเซนเซอร์เข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

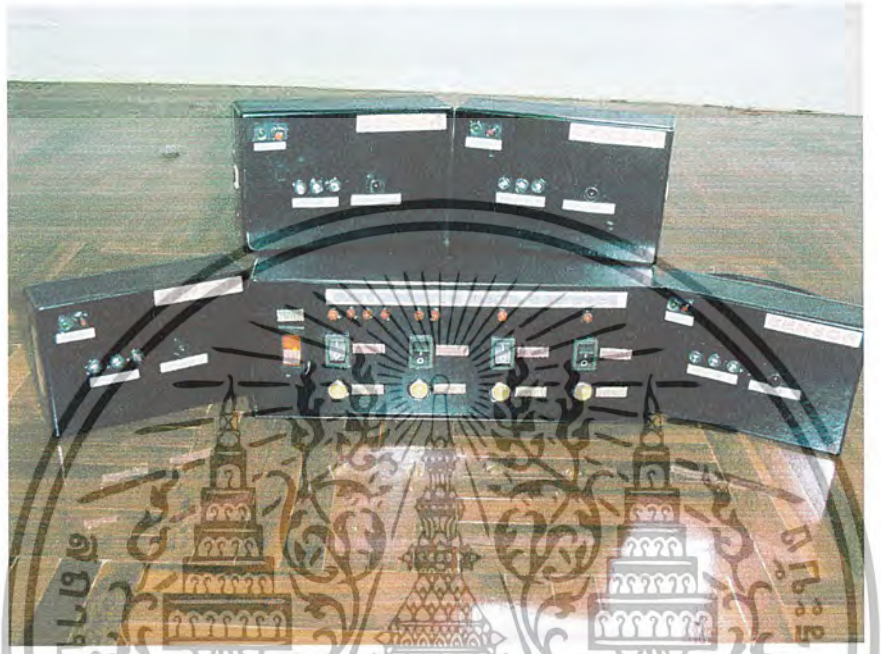
ขั้นตอนการทดลอง

4.4.1 นำชุดควบคุมไมโครคอนโทรลเลอร์ไปต่อกับเซนเซอร์ตามรูปที่ 4.7

4.4.2 ทำการเลือกกดสวิตซ์การทำงานของไมโครคอนโทรลเลอร์แต่ละตัว

4.4.3ทำการบันทึกผลการทดลองที่หลอดแอลอีดี เมื่อมีสัญญาณพัลส์ของเซนเซอร์แต่ละตัวเข้ามา

4.4.4 เมื่อทำการเริ่มการทำงานของไมโครคอนโทรลเลอร์แต่ละตัวใหม่จะต้องกดรีเซตทุกครั้ง



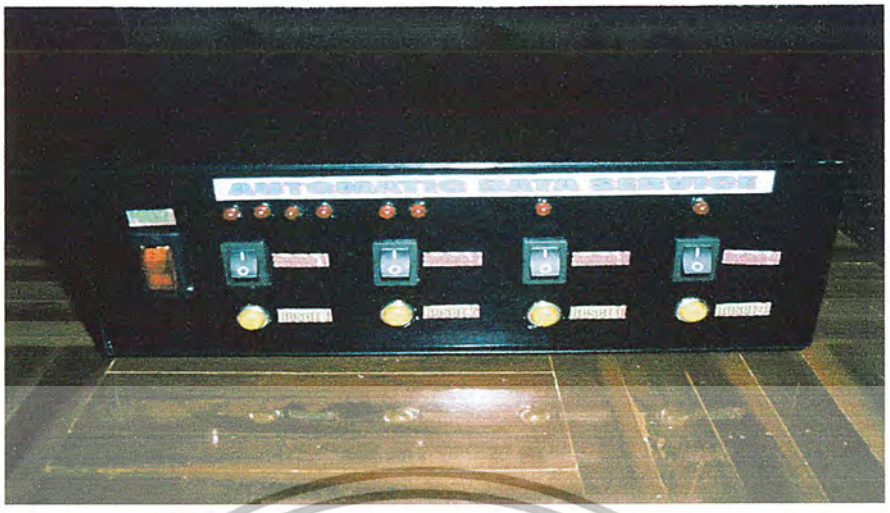
รูปที่ 4.7 แสดงด้านหน้าของชุดควบคุมไมโครคอนโทรลเลอร์ที่ต่อกับเซนเซอร์

ผลการทดลอง

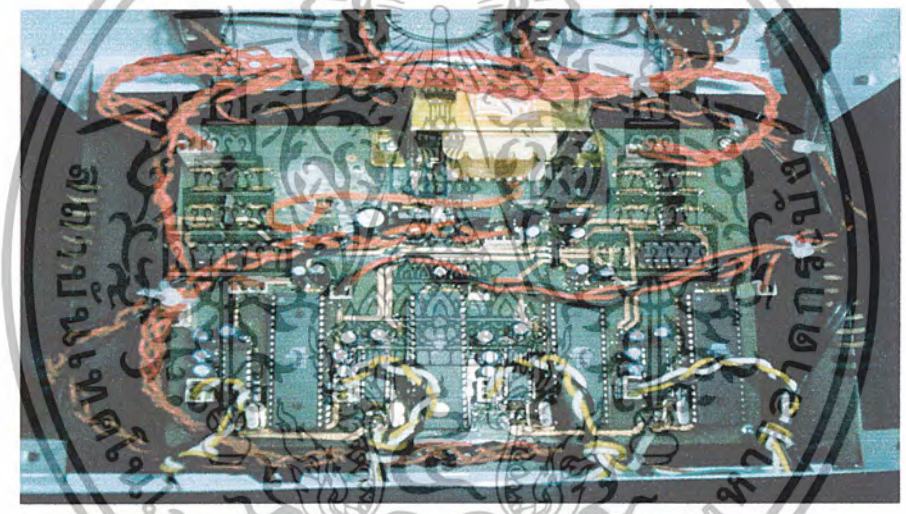
ไมโครคอนโทรลเลอร์	เลือกกดสวิตซ์	สถานะของหลอด LED
การทำงานของตัวที่ 1	Switch 1	หลอด LED แสดงสถานะติดได้ทั้งหมด 4 ดวง
	Switch 2	หลอด LED แสดงสถานะติดได้ทั้งหมด 2 ดวง
	Switch 3	หลอด LED แสดงสถานะติดได้ทั้งหมด 2 ดวง
	Switch 4	หลอด LED แสดงสถานะ ติดได้ทั้งหมด 1 ดวง
การทำงานของตัวที่ 2	Switch 2	หลอด LED แสดงสถานะติดได้ทั้งหมด 2 ดวง
	Switch 3	หลอด LED แสดงสถานะติดได้ทั้งหมด 1 ดวง
การทำงานของตัวที่ 3	Switch 3	หลอด LED แสดงสถานะติดได้ทั้งหมด 1 ดวง
การทำงานของตัวที่ 4	Switch 4	หลอด LED แสดงสถานะติดได้ทั้งหมด 1 ดวง

ตารางที่ 4.2 แสดงผลการทดลองชุดควบคุมไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 แสดงภาพถ่ายชุดควบคุมไมโครคอนโทรลเลอร์



รูปที่ 4.9 แสดงภาพถ่ายภายในของชุดควบคุมไมโครคอนโทรลเลอร์

รูปที่ 4.10 แสดงภาพถ่ายด้านหลังของชุดควบคุมไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

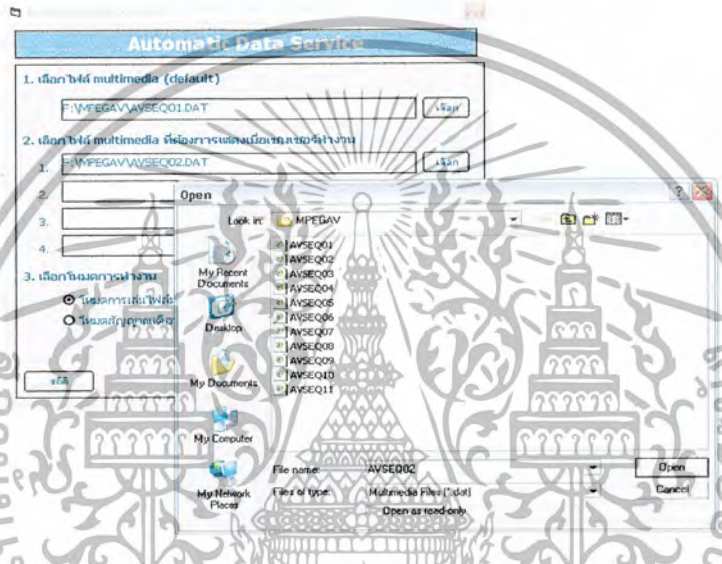
4.5 โปรแกรมวิชวลเบสิก เวอร์ชั่น 6.0

วัตถุประสงค์

เพื่อศึกษาการแสดงผลของโปรแกรมบนหน้าจอมอนิเตอร์ เมื่อชุดควบคุมไมโครคอนโทรลเลอร์ส่งข้อมูลผ่านทางพอร์ตอนุกรมอาร์เอสสองสามสองมา โดยลำดับการทดลองและผลการทดลองสามารถแบ่งออกเป็น 3 ส่วน ดังนี้

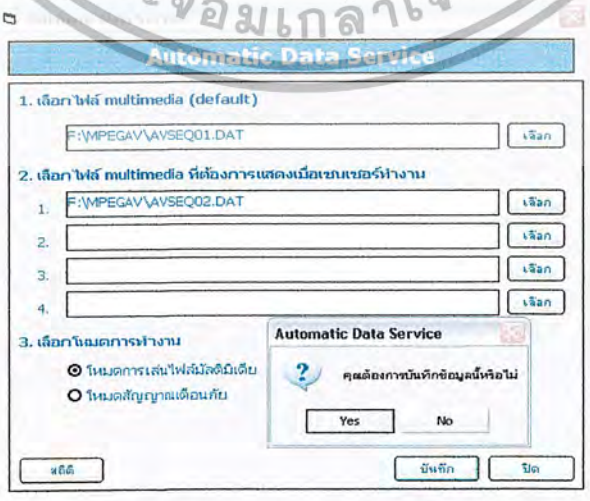
4.5.1 ส่วนของการแสดงผลหน้าจอควบคุมหลัก

- เมื่อทำการกดปุ่มเลือกไฟล์มัลติมีเดียต่างๆ ดังรูปที่ 4.11 ซึ่งสามารถค้นหาไฟล์มัลติมีเดียที่ต้องการแสดงข้อมูลได้



รูปที่ 4.11 แสดงผลการกดปุ่มเลือกไฟล์มัลติมีเดีย

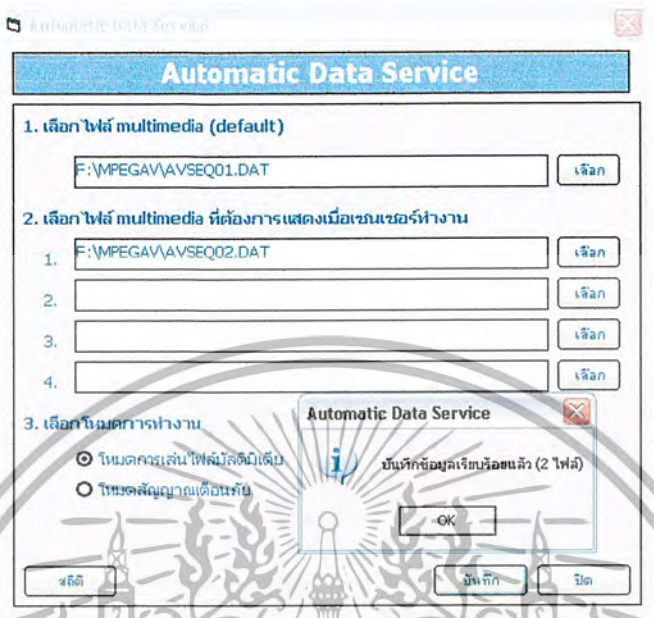
- หลังจากทำการเลือกไฟล์มัลติมีเดียที่ต้องการ แล้วทำการบันทึกข้อมูลที่ต้องการโดยกดปุ่มบันทึกจะแสดงผลดังรูปที่ 4.12



รูปที่ 4.12 แสดงผลการกดปุ่มบันทึกไฟล์มัลติมีเดีย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

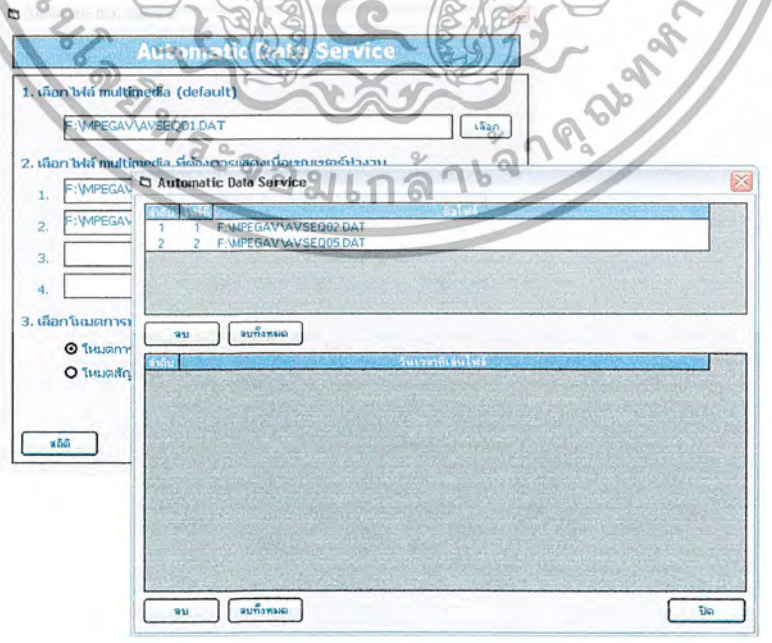
- เมื่อทำการยืนยันไฟล์มัลติมีเดียที่ต้องการบันทึก จะมีการแจ้งว่าได้บันทึกข้อมูลใหม่ลงไปในฐานข้อมูลกี่ไฟล์ ดังรูปที่ 4.13



รูปที่ 4.13 แสดงผลการบันทึกไฟล์มัลติมีเดียเรียบร้อยแล้ว

4.5.2 ส่วนของการแสดงผลในการจัดเก็บข้อมูลภายในฐานข้อมูล

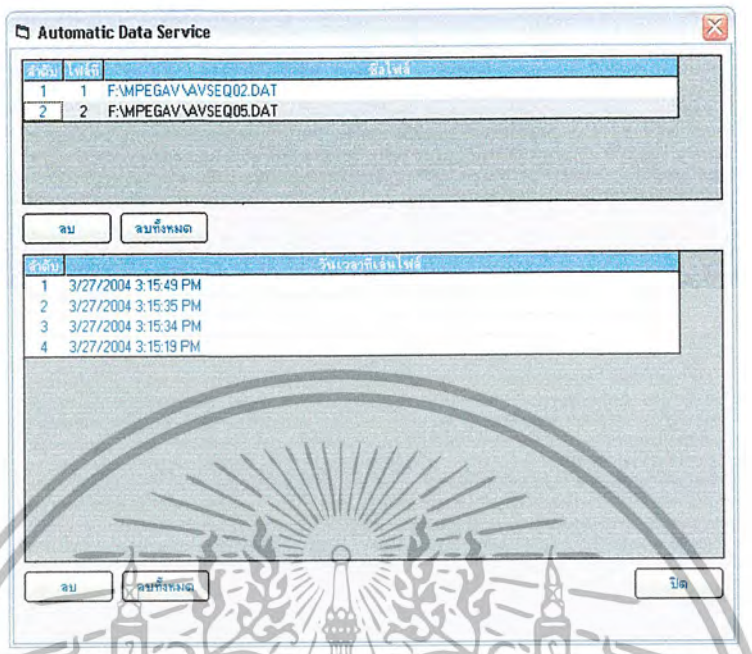
- เมื่อต้องการดูไฟล์มัลติมีเดียที่ถูกจัดเก็บไว้ในฐานข้อมูลทำได้โดยการกดปุ่มสถิติ ดังรูปที่ 4.14



รูปที่ 4.14 แสดงผลในการจัดเก็บข้อมูลภายในฐานข้อมูล

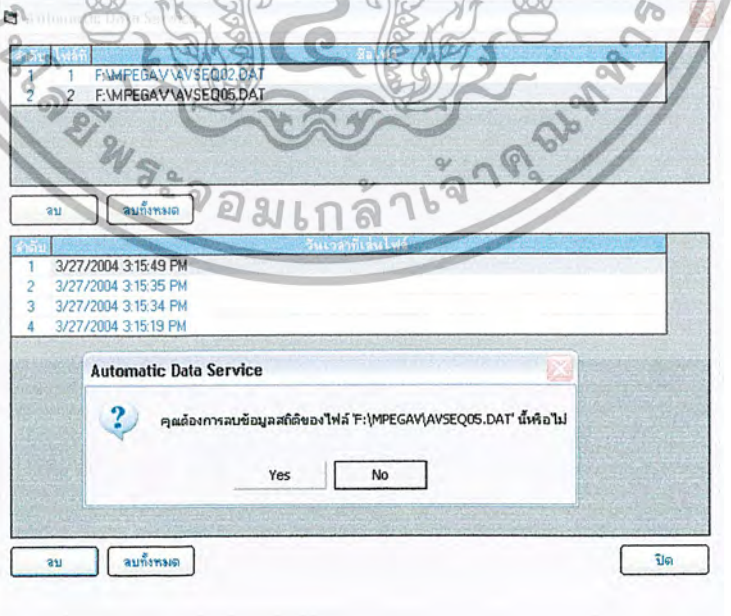
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ภายในฐานข้อมูลมีการแสดงผลเวลาและวันที่ของข้อมูลสถิติของไฟล์มัลติมีเดีย เมื่อไฟล์มัลติมีเดียนั้นๆถูกใช้งาน ดังรูปที่ 4.15



รูปที่ 4.15 แสดงผลข้อมูลสถิติของไฟล์มัลติมีเดีย

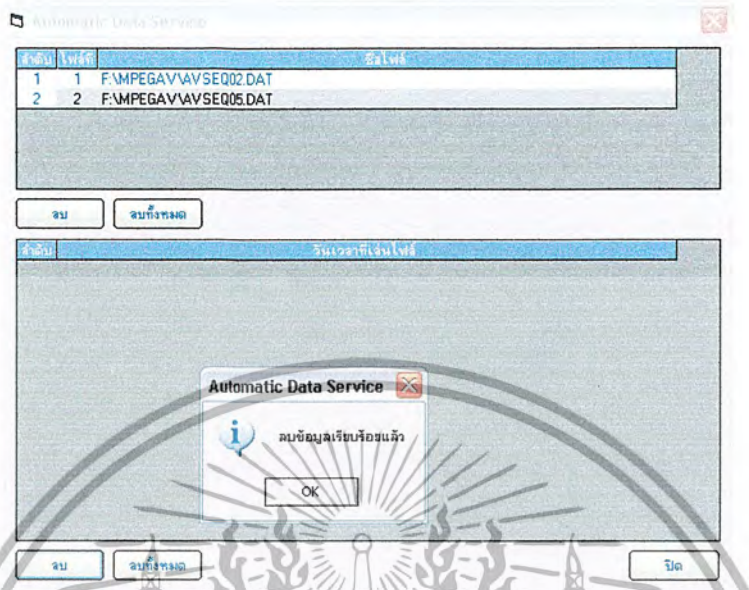
- ถ้าต้องการลบข้อมูลสถิติของไฟล์มัลติมีเดียใดออกจากฐานข้อมูล ก็ทำการกดปุ่มลบ ดังรูปที่ 4.16



รูปที่ 4.16 แสดงผลการกดปุ่มลบข้อมูลสถิติของไฟล์มัลติมีเดีย

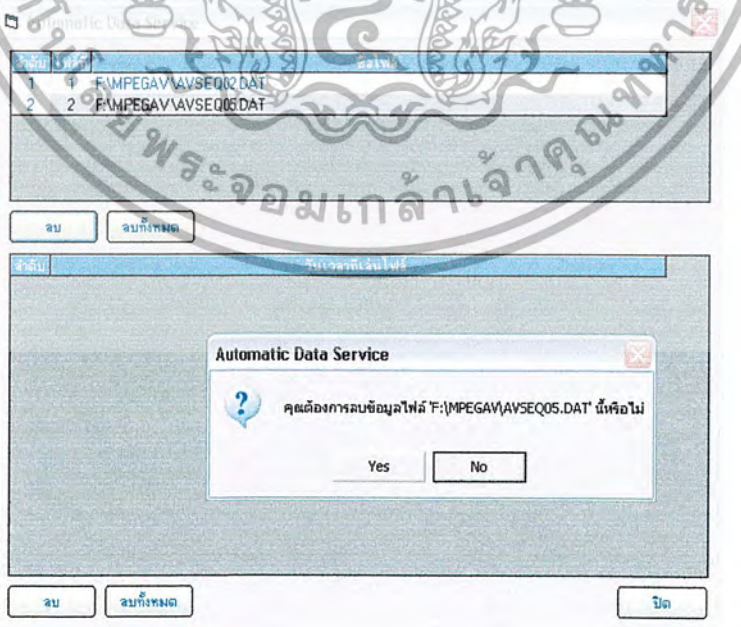
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หลังจากตกลงลบข้อมูลสถิติของไฟล์มัลติมีเดีย ดังรูปที่ 4.17 จะมีการแจ้งว่าข้อมูลนั้นได้ถูกลบออกจากฐานข้อมูลเรียบร้อยแล้ว



รูปที่ 4.17 แสดงผลหลังจากตกลงลบข้อมูลสถิติของไฟล์มัลติมีเดีย

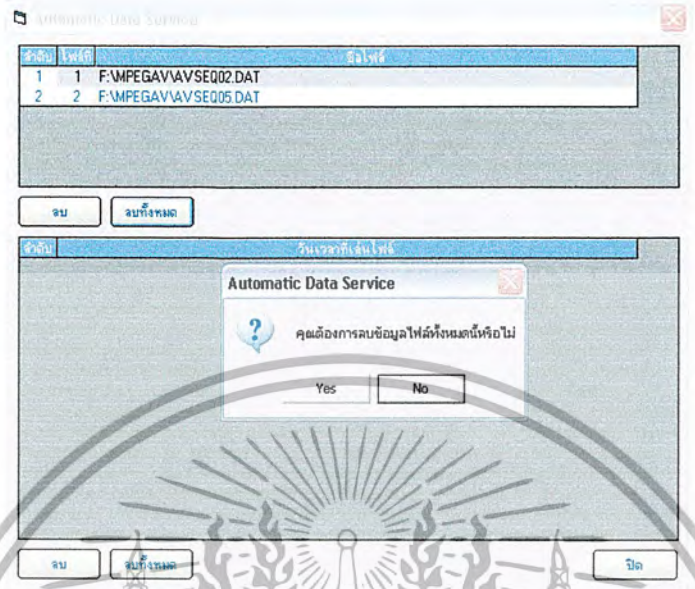
- ถ้าต้องการลบข้อมูลของไฟล์มัลติมีเดียใดออกจากฐานข้อมูล ก็ทำการกดปุ่มลบ ดังรูปที่ 4.18



รูปที่ 4.18 แสดงผลการกดปุ่มลบข้อมูลของไฟล์มัลติมีเดีย

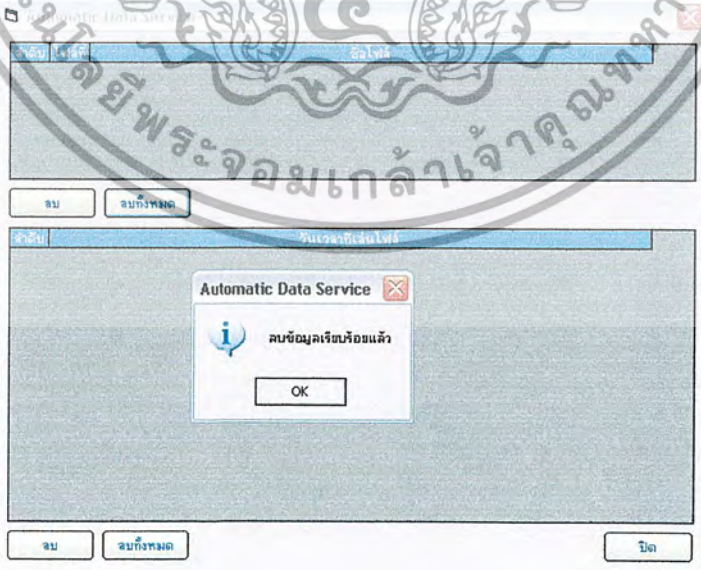
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถ้าต้องการลบข้อมูลของไฟล์มัลติมีเดียออกจากฐานข้อมูลทั้งหมด ก็ทำการกดปุ่มลบทั้งหมด ดังรูปที่ 4.19



รูปที่ 4.19 แสดงผลการกดปุ่มลบข้อมูลของไฟล์มัลติมีเดียทั้งหมด

- หลังจากตกลงลบข้อมูลของไฟล์มัลติมีเดียทั้งหมดแล้วดังรูปที่ 4.20 จะมีการแจ้งว่าข้อมูลนั้นได้ถูกลบออกจากฐานข้อมูลเรียบร้อยแล้ว



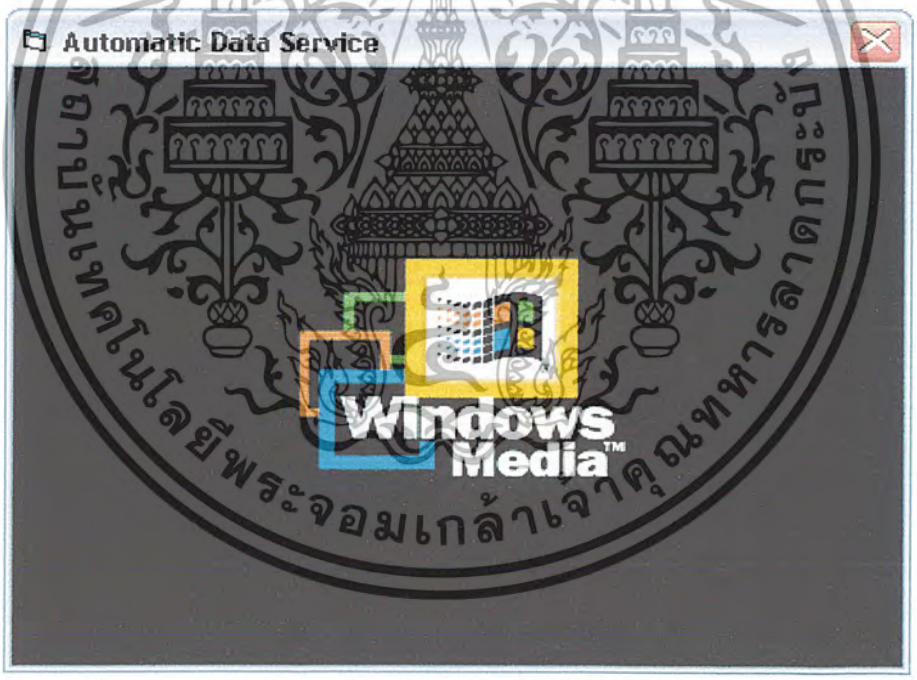
รูปที่ 4.20 แสดงผลหลังจากตกลงลบข้อมูลของไฟล์มัลติมีเดียทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.3 ส่วนของการแสดงผลของข้อมูลไฟล์มัลติมีเดีย

หลังจากที่ทำการเลือกไฟล์มัลติมีเดียที่ต้องการและทำการบันทึกเรียบร้อยแล้ว เมื่อทำการเปิดโปรแกรมในส่วนของการแสดงผลก็จะแสดงข้อมูลที่แบ่งได้ 3 ส่วน คือ

- การแสดงข้อมูลที่เป็นไฟล์มัลติมีเดียดิฟอลด์ ที่วนการทำงานเรื่อยๆ ซึ่งจะแสดงข้อมูลทันทีที่เปิดโปรมแกรมในส่วนแสดงผลขึ้นมาจนได้รับข้อมูลจากไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรม จึงหยุด
- การแสดงข้อมูลที่เป็นไฟล์มัลติมีเดีย เมื่อได้รับข้อมูลจากไมโครคอนโทรลเลอร์ ซึ่งข้อมูลที่จะทำการแสดงก็ขึ้นอยู่กับไฟล์มัลติมีเดียที่เลือกเอาไว้ เมื่อแสดงข้อมูลในส่วนนี้เสร็จก็จะกลับไปแสดงข้อมูลที่เป็นไฟล์มัลติมีเดียดิฟอลด์อีกครั้ง จนกว่าจะได้รับข้อมูลใหม่เข้ามา
- การแสดงข้อมูลที่เป็นโหมดสัญญาณเตือนภัย เมื่อทำการเลือกโหมดนี้จะมีสัญญาณเตือนภัยดังเป็นเวลา 5 นาทีขึ้น เมื่อได้รับข้อมูลจากไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรม



รูปที่ 4.21 แสดงผลของการแสดงผลของข้อมูลไฟล์มัลติมีเดีย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.22 แสดงภาพถ่ายชุดบริการข้อมูลอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

วิจารณ์และบทสรุปผลการทำงาน

การใช้งานความถี่อินฟราเรดความถี่ต่ำที่จะสามารถสะท้อนกับคนหรือวัตถุได้ โดยอาศัยการสะท้อนของลำคลื่นอินฟราเรดจากตัวส่งอินฟราเรดกลับมายังตัวรับอินฟราเรด โดยตัวรับอินฟราเรดจะเป็นตัวแปลงสัญญาณความถี่อินฟราเรดที่รับได้มาเป็นระดับสัญญาณทางไฟฟ้า ซึ่งจะมีค่ามากหรือน้อยขึ้นอยู่กับความเข้มของสัญญาณที่รับได้นั้น โดยเราสามารถที่จะกำหนดสภาวะการทำงานของวงจรได้ว่าจะให้ทำงานหรือไม่ให้ทำงาน โดยมีการนำค่าระดับแรงดันที่ตัวรับสามารถแปลงได้เทียบกับระดับแรงดันอ้างอิง หากระดับแรงดันที่ได้มากกว่าระดับแรงดันอ้างอิงสภาวะของวงจรจะทำงาน แต่ถ้าระดับแรงดันที่ได้น้อยกว่าระดับแรงดันอ้างอิงสภาวะของวงจรจะไม่ทำงาน.

การเลือกใช้แอลอีดีอินฟราเรดตัวส่งกำลังสูงจะทำให้คลื่นอินฟราเรดได้ระยะทางที่ไกลขึ้น โดยสามารถเพิ่มจำนวนของแอลอีดีอินฟราเรดได้ 2 ถึง 3 ตัว เพื่อให้มีความเข้มของคลื่นอินฟราเรดให้สูงขึ้น สามารถส่งกระทบกับคนหรือวัตถุ และสะท้อนกลับมายังตัวรับอินฟราเรดได้ดีขึ้น

การวางแอลอีดีทั้งตัวส่งและตัวรับที่อยู่ใกล้กัน จะทำให้เกิดการรบกวนจากตัวส่งอินฟราเรดไปยังตัวรับอินฟราเรดในแนวด้านข้างได้ ดังนั้นจึงควรป้องกันโดยใช้ปลอกแอลอีดีครอบ เพื่อให้ตัวส่งอินฟราเรดมีทิศทางที่แน่นอนและสามารถส่งได้ไกลขึ้น และเพื่อป้องกันการรบกวนของแสงอื่นๆที่มากระทบกับตัวรับอินฟราเรด.

แนวทางการพัฒนาต่อ

ในโครงงานนี้ได้เป็นตัวให้บริการข้อมูลที่ใช้กับการแสดงนิทรรศการในงานต่างๆโดยการทำงานที่อัตโนมัติ แต่อันที่จริงแล้วโครงงานนี้มีลักษณะเป็นวงกว้างครอบคลุมในการประยุกต์ใช้งานได้หลายอย่าง ไม่จำกัดอยู่ที่งานแสดงนิทรรศการเท่านั้น คือสามารถใช้ได้กับงานทุกประเภทที่มีการส่งข้อมูลและรับข้อมูลแล้วแสดงผลออกหน้าจอคอมพิวเตอร์ และเป็นลักษณะงานที่ต้องการความสะดวกสบายด้วยระบบที่อัตโนมัติ เช่น ใช้เป็นตัวเปิด-ปิดอุปกรณ์ไฟฟ้าอื่นๆภายในบ้าน หรืออุปกรณ์อื่นๆซึ่งต้องดูตามความเหมาะสม.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรมแกรมต่างๆของชุดบริการข้อมูลอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมโมโครคอนโทรลเลอร์ตัวที่ 1

```
sen1  bit   p1.0
sen2  bit   p1.1
sen3  bit   p1.2
sen4  bit   p1.3

      org   0000h

main: clr   tr1
            mov   scon,#00h
            mov   tmod,#021h
            mov   th1,#0fdh
            mov   tl1,#0fdh
            mov   p1,#0fh
            mov   a,#00h
            mov   p1,#0fh
            mov   p2,#0fh

loop_main:  mov   r0,p2
            cjne  r0,#00000111b,next_1
            ljmp  loop_a
next_1:     cjne  r0,#00000110b,next_2
            ljmp  loop_b
next_2:     cjne  r0,#00000101b,next_3
            ljmp  loop_c
next_3:     cjne  r0,#00000011b,next_4
            ljmp  loop_d
next_4:     cjne  r0,#00000100b,next_5
            ljmp  loop_b
next_5:     cjne  r0,#00000010b,next_6
            ljmp  loop_f
next_6:     cjne  r0,#00000001b,next_7
            ljmp  loop_g
next_7:     cjne  r0,#00000000b,loop_main
            ljmp  loop_f

loop_a:     jnb   sen1,mode1a
            jnb   sen2,mode2a
            jnb   sen3,mode3a
            jnb   sen4,mode4a
            mov   p2,#0fh
            ljmp  loop_main
mode1a:     ljmp  mode1
mode2a:     ljmp  mode2
mode3a:     ljmp  mode3
mode4a:     ljmp  mode4

loop_b:     jnb   sen1,mode1b
            jnb   sen4,mode4b
            mov   p2,#0fh
            ljmp  loop_main
mode1b:     ljmp  mode1
mode4b:     ljmp  mode4

loop_c:     jnb   sen1,mode1c
            jnb   sen2,mode2c
            jnb   sen4,mode4c
            mov   p2,#0fh
            ljmp  loop_main
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mode1c:     ljmp  mode1
mode2c:     ljmp  mode2
mode4c:     ljmp  mode4

loop_d:     jnb   sen1,mode1d
            jnb   sen2,mode2d
            jnb   sen3,mode3d
            mov   p2,#0fh
            ljmp  loop_main
mode1d:     ljmp  mode1
 mode2d:    ljmp  mode2
mode3d:     ljmp  mode3

loop_f:     jnb   sen1,mode1f
            mov   p2,#0fh
            ljmp  loop_main
mode1f:     ljmp  mode1

loop_g:     jnb   sen1,mode1g
            jnb   sen2,mode2g
            mov   p2,#0fh
            ljmp  loop_main
mode1g:     ljmp  mode1
mode2g:     ljmp  mode2

mode1:      lcall trb
            setb  p1.4
            setb  tr1
            mov   scon,#040h
            mov   dptr,#song1
            acall TX_TEXT
            setb  ren
loop1:      jnb   ri,$
            clr   ri
            mov   a,sbuf
            cjne  a,#08h,loop1
            clr   ren
            clr   ri
            ljmp  main
mode2:      lcall trb
            setb  p1.5
            setb  tr1
            mov   scon,#040h
            mov   dptr,#song2
            acall TX_TEXT
            setb  ren
loop2:      jnb   ri,$
            clr   ri
            mov   a,sbuf
            cjne  a,#04h,loop2
            clr   ren
            clr   ri
            ljmp  main
mode3:      lcall trb
            setb  p1.6
            setb  tr1
            mov   scon,#040h
            mov   dptr,#song3
            acall TX_TEXT
            setb  ren
loop3:      jnb   ri,$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            clr   ri
            mov   a,sbuf
            cjne  a,#02h,loop3
            clr   ren
            clr   ri
            ljmp  main
mode4:      lcall trb
            setb  p1.7
            setb  tr1
            mov   scon,#040h
            mov   dptr,#song4
            acall TX_TEXT
            setb  ren
loop4:      jnb   ri,$
            clr   ri
            mov   a,sbuf
            cjne  a,#01h,loop4
            clr   ren
            clr   ri
            ljmp  main
;-------------------------------------------------------------------------
--------
; TX Serial Text from ROM Pointer
;-------------------------------------------------------------------------
--------
TX_TEXT:          CLR         TI              ; Clear TI
TX_LOOP:          CLR         A               ; Clear ACC.
                  MOVC  A,@A+DPTR             ; Get Data from ROM
with Pointer
                  INC         DPTR            ; Increase Pointer
                  CJNE  A,#0FFH,TX_CHAR ; Check 0FFH End of
Text Char.
                  RET                         ; End => Return
TX_CHAR:          MOV         SBUF,A          ; Send Data to
SBUF
                  JNB         TI,$            ; Wait until TX
already (TI=1)
                  CLR         TI              ; Clear TI
                  ACALL DELAY_100ms           ; Delay
                  AJMP  TX_LOOP               ; Jump to TX_LOOP

;-------------------------------------------------------------------------
--------
; Dummy Delay time 100m
;-------------------------------------------------------------------------
--------
DELAY_100ms:      MOV               R7,#100             ; Do 100
times
DELAY_100ms_1:    MOV         R6,#0E6H          ; Each loop = 1 ms
DELAY_100ms_2:    NOP
                  NOP
                  DJNZ  R6,DELAY_100ms_2
                  DJNZ  R7,DELAY_100ms_1
                  RET
trb:  mov   r7,#20
mm:         mov   r6,#0f9h
nn:         nop
            nop
            djnz  r6,nn
            djnz  r7,mm
            ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------------------
--------
;Define Constant < Store in Flash EEPROM Program Memory >
;-------------------------------------------------------------------------
--------
song1:      DB    08h,0FFH
song2:      DB    04h,0FFH
song3:      DB    02h,0FFH
song4:      DB    01h,0FFH

      end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมโมโครคอนโทรลเลอร์ตัวที่ 2

```
      org   0000h

main: clr   tr1
            mov   scon,#00h
            mov   tmod,#021h
            mov   th1,#0fdh
            mov   tl1,#0fdh
            mov   p1,#0fh
            mov   a,#00h
            mov   p2,#00000011b

loop: jb    p2.0,loop
            jnb   p1.1,mode1
            jnb   p1.2,mode2
            setb  p2.0
            ljmp  loop

mode1:      lcall trb
            setb  p1.4
            setb  tr1
            mov   scon,#040h
            mov   dptr,#song1
            acall TX_TEXT
            setb  ren
loop1:      jnb   ri,$
            clr   ri
            mov   a,sbuf
            cjne  a,#08h,loop1
            clr   ren
            clr   ri
            ljmp  main
mode2:      jnb    p2.1 ,jj
            lcall trb
            setb  p1.5
            setb  tr1
            mov   scon,#040h
            mov   dptr,#song2
            acall TX_TEXT
            setb  ren
loop2:      jnb   ri,$
            clr   ri
            mov   a,sbuf
            cjne  a,#04h,loop2
            clr   ren
            clr   ri
jj:         ljmp  main

;-------------------------------------------------------------------------
--------
; TX Serial Text from ROM Pointer
;-------------------------------------------------------------------------
--------
TX_TEXT:          CLR         TI                ; Clear TI
TX_LOOP:          CLR         A                 ; Clear ACC.
                  MOVC  A,@A+DPTR               ; Get Data from ROM
with Pointer
                  INC         DPTR              ; Increase Pointer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        CJNE  A,#0FFH,TX_CHAR ; Check 0FFH End of
Text Char.
                        RET                   ; End => Return
TX_CHAR:          MOV       SBUF,A                ; Send Data to
SBUF
                        JNB         TI,$      ; Wait until TX
already (TI=1)
                        CLR         TI        ; Clear TI
                        ACALL DELAY_100ms     ; Delay
                        AJMP  TX_LOOP         ; Jump to TX_LOOP

;-------------------------------------------------------------------
--------
; Dummy Delay time 100m
;-------------------------------------------------------------------
--------
DELAY_100ms:      MOV               R7,#100             ; Do 100
times
DELAY_100ms_1:    MOV         R6,#0E6H          ; Each loop = 1 ms
DELAY_100ms_2:    NOP
                  NOP
                  DJNZ  R6,DELAY_100ms_2
                  DJNZ  R7,DELAY_100ms_1
                  RET

trb:  mov   r7,#20
mm:         mov   r6,#0f9h
nn:         nop
            nop
            djnz  r6,nn
            djnz  r7,mm
            ret
;-------------------------------------------------------------------
--------
;Define Constant < Store in Flash EEPROM Program Memory >
;-------------------------------------------------------------------
--------
song1:      DB    08h,0FFH
song2:      DB    04h,0FFH

      end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมไมโครคอนโทรลเลอร์ตัวที่ 3

```
        org     0000h
main:   clr     tr1
                mov     scon,#00h
                mov     tmod,#021h
                mov     th1,#0fdh
                mov     tl1,#0fdh
                mov     p1,#0fh
                mov     a,#00h
                mov     p2,#00000001b

loop:   jb      p2.0,loop
                jnb     p1.2,mode1
                setb    p2.0
                ljmp    loop

mode1:          lcall   trb
                setb    p1.6
                setb    tr1
                mov     scon,#040h
                mov     dptr,#song1
                acall   TX_TEXT
                setb    ren
loop1:          jnb     ri,$
                clr     ri
                mov     a,sbuf
                cjne    a,#08h,loop1
                clr     ren
                clr     ri
                ljmp    main

;-------------------------------------------------------------------------
--------
; TX Serial Text from ROM Pointer
;-------------------------------------------------------------------------
--------
TX_TEXT:        CLR     TI              ; Clear TI
TX_LOOP:        CLR     A               ; Clear ACC.
                MOVC    A,@A+DPTR       ; Get Data from ROM
with Pointer
                INC     DPTR            ; Increase Pointer
                CJNE    A,#0FFH,TX_CHAR ; Check 0FFH End of
Text Char.
                RET                     ; End => Return
TX_CHAR:        MOV     SBUF,A                  ; Send Data to
SBUF
                JNB     TI,$            ; Wait until TX
already (TI=1)
                CLR     TI              ; Clear TI
                ACALL   DELAY_100ms     ; Delay
                AJMP    TX_LOOP         ; Jump to TX_LOOP

;-------------------------------------------------------------------------
--------
; Dummy Delay time 100m
;-------------------------------------------------------------------------
--------
DELAY_100ms:    MOV             R7,#100                 ; Do 100
times
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DELAY_100ms_1:    MOV         R6,#0E6H          ; Each loop = 1 ms
DELAY_100ms_2:    NOP
                  NOP
                  DJNZ  R6,DELAY_100ms_2
                  DJNZ  R7,DELAY_100ms_1
                  RET
trb:  mov   r7,#20
mm:         mov   r6,#0f9h
nn:         nop
            nop
            djnz  r6,nn
            djnz  r7,mm
            ret
;-------------------------------------------------------------------------
--------
;Define Constant < Store in Flash EEPROM Program Memory >
;-------------------------------------------------------------------------
--------
song1:      DB    08h,0FFH
      end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมโมโครคอนโทรลเลอร์ตัวที่ 4

```
org   0000h
      main: clr   tr1
            mov   scon,#00h
            mov   tmod,#021h
            mov   th1,#0fdh
            mov   tl1,#0fdh
            mov   p1,#0fh
            mov   a,#00h
            mov   p2,#00000001b

loop: jb    p2.0,loop
            jnb   p1.2,mode1
            setb  p2.0
            ljmp  loop

mode1:      lcall trb
            setb  p1.7
            setb  tr1
            mov   scon,#040h
            mov   dptr,#song1
            acall TX_TEXT
            setb  ren
loop1:      jnb   ri,$
            clr   ri
            mov   a,sbuf
            cjne  a,#08h,loop1
            clr   ren
            clr   ri
            ljmp  main

;-------------------------------------------------------------------
--------
; TX Serial Text from ROM Pointer
;-------------------------------------------------------------------
--------
TX_TEXT:          CLR         TI                ; Clear TI
TX_LOOP:          CLR         A                 ; Clear ACC.
                  MOVC  A,@A+DPTR               ; Get Data from ROM
with Pointer
                  INC         DPTR              ; Increase Pointer
                  CJNE  A,#0FFH,TX_CHAR ; Check 0FFH End of
Text Char.
                  RET                           ; End => Return
TX_CHAR:          MOV         SBUF,A                  ; Send Data to
SBUF
                  JNB         TI,$              ; Wait until TX
already (TI=1)
                  CLR         TI                ; Clear TI
                  ACALL DELAY_100ms             ; Delay
                  AJMP  TX_LOOP                 ; Jump to TX_LOOP
;-------------------------------------------------------------------
--------
; Dummy Delay time 100m
;-------------------------------------------------------------------
--------
DELAY_100ms:      MOV               R7,#100                 ; Do 100
times
DELAY_100ms_1:    MOV         R6,#0E6H          ; Each loop = 1 ms
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DELAY_100ms_2:    NOP
                  NOP
                  DJNZ  R6,DELAY_100ms_2
                  DJNZ  R7,DELAY_100ms_1
                  RET
trb:  mov   r7,#20
mm:         mov   r6,#0f9h
nn:         nop
            nop
            djnz  r6,nn
            djnz  r7,mm
            ret
;-------------------------------------------------------------------------
--------
;Define Constant < Store in Flash EEPROM Program Memory >
;-------------------------------------------------------------------------
--------
song1:      DB    08h,0FFH
      end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมในส่วนของฟอร์มหน้าจอควบคุมหลัก

```
Private Const APP_HEADER = "Automatic Data Service"
Private tempPath(4) As String
' Mode Day=0 ,Night=1
Private mode As Integer
Private Sub cmdBrowse_Click(Index As Integer)
On Error GoTo ERR_CANCEL
    openDialog.Filter = "Multimedia Files (*.dat)|*.dat|All Files
(*.*)|*.*"
    openDialog.ShowOpen
    txtPath(Index).Text = openDialog.fileName
    Exit Sub
ERR_CANCEL:
End Sub
Private Sub cmdClose_Click()
    Unload Me
End Sub
Private Sub cmdSave_Click()
    If MsgBox("คุณต้องการบันทึกข้อมูลนี้หรือไม่", vbQuestion + vbYesNo) = vbYes Then
        Call SaveData
    End If
End Sub
Private Sub cmdStat_Click()
    frmStat.Show vbModal
End Sub
Private Sub Form_Load()
    Call InitializeControl
    Call LoadData
End Sub
Private Sub InitializeControl()
    Me.Caption = APP_HEADER
    lblHeader.Caption = APP_HEADER
    openDialog.CancelError = True
End Sub

Private Sub LoadData()
Dim obj As New Multimedia
Dim rs As ADODB.Recordset
Dim i As Integer
    Set rs = obj.SelectBySeq()
    If Not rs Is Nothing Then
        For i = 0 To rs.RecordCount - 1
            tempPath(rs("SEQ")) = rs("FILE_PATH")
            txtPath(rs("SEQ")).Text = rs("FILE_PATH")
            rs.MoveNext
        Next
    End If
    mode = GetSetting(APP_HEADER, "Setting", "Mode", "0")
    optMode(0).Value = mode = 0
    optMode(1).Value = mode = 1
    Set rs = Nothing
    Set obj = Nothing
End Sub
Private Sub SaveData()
Dim i As Integer
Dim row As Integer
    For i = 0 To 4
        If Trim(txtPath(i).Text) <> tempPath(i) Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            If SaveFilePath(i, Trim(txtPath(i).Text)) Then row = row
+ 1
        End If
    Next
    ' Mode Day=0 ,Night=1
    Call SaveSetting(APP_HEADER, "Setting", "Mode",
IIf(optMode(0).Value, "0", "1"))
    If row > 0 Then
        MsgBox "บันทึกข้อมูลเรียบร้อยแล้ว (" & row & "ไฟล์)", vbInformation
    End If
End Sub
Private Function SaveFilePath(ByVal Index As Integer, ByVal fileName
As String) As Boolean
Dim obj As New Multimedia
Dim rs As ADODB.Recordset
Dim ret As Boolean
    If tempPath(Index) <> "" And fileName = "" Then
        ret = obj.SetActiveFlag(False, tempPath(Index))
    ElseIf Not obj.IsPathExist(fileName) Then
        Call obj.SetActiveFlag(False, tempPath(Index))
        Set rs = obj.SelectEmpty
        rs.AddNew
        rs("SEQ") = Index
        rs("FILE_PATH") = fileName
        rs("IS_INACTIVE") = "0"
        ret = obj.Create(rs)
    End If
    Set rs = Nothing
    Set obj = Nothing
    SaveFilePath = ret
End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมในส่วนของการจัดเก็บข้อมูลในฐานข้อมูล

```
Option Explicit
Private Const APP_HEADER = "Automatic Data Service"
Private Sub cmdClose_Click()
    Unload Me
End Sub
Private Sub ResetMaster()
    If fxgMaster.row > 0 Then
        If MsgBox("คุณต้องการลบข้อมูลไฟล์'" & fxgMaster.TextMatrix(fxgMaster.row,
3) & "' นี้หรือไม่", vbQuestion + vbYesNo + vbDefaultButton2) = vbYes Then
            Dim id As Integer
            Dim ret As Boolean
            id = fxgMaster.TextMatrix(fxgMaster.row, 1)
            Dim obj As New Multimedia
            If obj.ResetFile(id) Then
                Call RefreshMaster(fxgMaster)
                MsgBox "ลบข้อมูลเรียบร้อยแล้ว", vbInformation
            End If
            Set obj = Nothing
        End If
    End If
End Sub
Private Sub ResetAllMasters()
    If MsgBox("คุณต้องการลบข้อมูลไฟล์ทั้งหมดนี้หรือไม่", vbQuestion + vbYesNo +
vbDefaultButton2) = vbYes Then
        Dim obj As New Multimedia
        If obj.ResetAllFiles Then
            Call RefreshMaster(fxgMaster)
            MsgBox "ลบข้อมูลเรียบร้อยแล้ว", vbInformation
        End If
        Set obj = Nothing
    End If
End Sub

Private Sub ResetDetail()
    If fxgMaster.row > 0 Then
        If MsgBox("คุณต้องการลบข้อมูลสถิติของไฟล์'" &
fxgMaster.TextMatrix(fxgMaster.row, 3) & "' นี้หรือไม่", vbQuestion +
vbYesNo + vbDefaultButton2) = vbYes Then
            Dim id As Integer
            Dim ret As Boolean
            id = fxgMaster.TextMatrix(fxgMaster.row, 1)
            Dim obj As New Multimedia
            If obj.ResetStat(id) Then
                If fxgMaster.row > 0 Then
                    Call RefreshDetail(fxgDetail,
CInt(fxgMaster.TextMatrix(fxgMaster.row, 1)))
                End If
                MsgBox "ลบข้อมูลเรียบร้อยแล้ว", vbInformation
            End If
            Set obj = Nothing
        End If
    End If
End Sub
Private Sub ResetAllDetails()
    If MsgBox("คุณต้องการลบข้อมูลสถิติทั้งหมดนี้หรือไม่", vbQuestion + vbYesNo +
vbDefaultButton2) = vbYes Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Dim obj As New Multimedia
        If obj.ResetAllStats Then
            If fxgMaster.row > 0 Then
                Call RefreshDetail(fxgDetail,
CInt(fxgMaster.TextMatrix(fxgMaster.row, 1)))
            End If
            MsgBox "ลบข้อมูลเรียบร้อยแล้ว", vbInformation
        End If
        Set obj = Nothing
    End If
End Sub
Private Sub cmdReset_Click(Index As Integer)
    Select Case Index
        Case 0: Call ResetMaster
        Case 1: Call ResetAllMasters
        Case 2: Call ResetDetail
        Case 3: Call ResetAllDetails
    End Select
End Sub
Private Sub Form_Load()
    Call InitializeControl
    Call InitializeMasterGrid(fxgMaster)
    Call RefreshMaster(fxgMaster)
    Call InitializeDetailGrid(fxgDetail)
End Sub
Private Sub InitializeControl()
    Me.Caption = APP_HEADER
End Sub
Private Sub InitializeMasterGrid(ByRef fxgMaster As MSFlexGrid)
Dim i As Integer
    fxgMaster.Clear
    fxgMaster.Cols = 4
    fxgMaster.ColWidth(0) = 500
    fxgMaster.ColWidth(1) = 0
    fxgMaster.ColWidth(2) = 500
    fxgMaster.ColWidth(3) = 7000
      For i = 0 To fxgMaster.Cols - 1
       fxgMaster.FixedAlignment(i) = flexAlignCenterTop
       fxgMaster.ColAlignment(i) = flexAlignLeftTop
    Next
    fxgMaster.ColAlignment(0) = flexAlignCenterTop
    fxgMaster.ColAlignment(2) = flexAlignCenterTop
     fxgMaster.TextMatrix(0, 0) = "ลำดับ"
    fxgMaster.TextMatrix(0, 2) = "ไฟล์ที่"
    fxgMaster.TextMatrix(0, 3) = "ชื่อไฟล์"
End Sub
Private Sub InitializeDetailGrid(ByRef fxgDetail As MSFlexGrid)
Dim i As Integer
    fxgDetail.Clear
    fxgDetail.Cols = 2
     fxgDetail.ColWidth(0) = 500
    fxgDetail.ColWidth(1) = 7500
    For i = 0 To fxgDetail.Cols - 1
        fxgDetail.FixedAlignment(i) = flexAlignCenterTop
        fxgDetail.ColAlignment(i) = flexAlignLeftTop
    Next
    fxgDetail.ColAlignment(0) = flexAlignCenterTop
     fxgDetail.TextMatrix(0, 0) = "ลำดับ"
    fxgDetail.TextMatrix(0, 1) = "วันเวลาที่เล่นไฟล์"
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub RefreshMaster(ByRef fxgMaster As MSFlexGrid)
    Dim obj As New Multimedia
    Dim rs As ADODB.Recordset
    Set rs = obj.SelectBySeqEx
    Dim i As Integer
    fxgMaster.Rows = rs.RecordCount + 1
    For i = 1 To fxgMaster.Rows - 1
        fxgMaster.TextMatrix(i, 0) = i
        fxgMaster.TextMatrix(i, 1) = rs("ID")
        fxgMaster.TextMatrix(i, 2) = rs("SEQ")
        fxgMaster.TextMatrix(i, 3) = rs("FILE_PATH")
        rs.MoveNext
    Next
    fxgMaster.row = 0
    Set rs = Nothing
    Set obj = Nothing
End Sub
Private Sub RefreshDetail(ByRef fxgDetail As MSFlexGrid, ByVal id As
Integer)
    Dim obj As New Multimedia
    Dim rs As ADODB.Recordset
    Set rs = obj.ReadStat(id)
    Dim i As Integer
    fxgDetail.Rows = rs.RecordCount + 1
    For i = 1 To fxgDetail.Rows - 1
        fxgDetail.TextMatrix(i, 0) = i
        fxgDetail.TextMatrix(i, 1) = rs("PLAY_DATE")
        rs.MoveNext
    Next
    fxgDetail.row = 0
    Set rs = Nothing
    Set obj = Nothing
End Sub
Private Sub fxgMaster_Click()
    If fxgMaster.row > 0 Then
        Call RefreshDetail(fxgDetail,
CInt(fxgMaster.TextMatrix(fxgMaster.row, 1)))
    End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมในส่วนของการแสดงผลการทำงาน

```
Option Explicit
Private Const APP_HEADER = "Automatic Data Service"
Private tempPath(4) As String
' Mode Day=0 ,Night=1
Private mode As Integer
Private filePlay As Integer
Private oncePlay As Boolean
Private startPlay As Date
Private Const warningMin As Integer = 5
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As
Long)
Private Sub Form_Load()
    Call InitializeControl
    Call LoadData
    comm.PortOpen = True
    tmrComm.Enabled = True
    If mode = 0 Then Call Play(0)
End Sub
Private Sub Form_Unload(Cancel As Integer)
    comm.PortOpen = False
End Sub
Private Sub InitializeControl()
    Me.Caption = APP_HEADER
End Sub
Private Sub Play(ByVal index As Integer)
On Error Resume Next
    mode = GetSetting(APP_HEADER, "Setting", "Mode", "0")
    Select Case mode
        Case 0
            If tempPath(index) <> "" Then
                Dim obj As New Multimedia
                Call obj.WriteStat(tempPath(index))
                Set obj = Nothing

                wmpMain.Stop
                wmpMain.PlayCount = IIf(index > 0, 1, 0)
                wmpMain.AutoRewind = (index = 0)
                wmpMain.Open tempPath(index)
            End If
        Case 1
            wmpMain.Stop
            Call Sleep(200)
            wmpMain.PlayCount = 0
            wmpMain.Open App.Path & "\Default.wav"
            If Not oncePlay Then
                startPlay = Now
                oncePlay = True
            End If
    End Select
End Sub

Private Sub LoadData()
Dim obj As New Multimedia
Dim rs As ADODB.Recordset
Dim i As Integer
    Set rs = obj.SelectBySeq()
    If Not rs Is Nothing Then
        For i = 0 To rs.RecordCount - 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            tempPath(rs("SEQ")) = rs("FILE_PATH")
            rs.MoveNext
        Next
    End If
    mode = GetSetting(APP_HEADER, "Setting", "Mode", "0")
    Set rs = Nothing
    Set obj = Nothing
End Sub
Private Sub tmrComm_Timer()
Dim data As String
    On Error Resume Next
    comm.DTREnable = False
    comm.DTREnable = True
    comm.InputLen = 0
    data = comm.Input
    data = Hex$(Asc(data))
    wmpMain.PlayCount = 1
    Select Case data
        Case 8: Call Play(1)
        Case 4: Call Play(2)
        Case 2: Call Play(3)
        Case 1: Call Play(4)
    End Select
    filePlay = CInt(data)
End Sub
'Private Sub wmpMain_PlayStateChange(ByVal OldState As Long, ByVal
NewState As Long)
    If NewState = 0 Then
        Select Case mode
            Case 0: Call PlayCompleted
            Case 1
        End Select
        If mode = 0 Then
            Call Play(0)
        Else
            ' เล่นไฟล์ Default.wav ต่อไปเรื่อยๆ
            If mode = 1 And oncePlay Then Call Play(0)
            If DateDiff("n", startPlay, Now) >= warningMin Then
                Call PlayCompleted
                oncePlay = False
            End If
        End If
    End If
End Sub

Private Sub PlayCompleted()
    Select Case filePlay
        Case 8: comm.Output = Chr$(&H8)
        Case 4: comm.Output = Chr$(&H4)
        Case 2: comm.Output = Chr$(&H2)
        Case 1: comm.Output = Chr$(&H1)
    End Select
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Class_Initialize()
Dim strPath As String
    strPath = App.Path  ' Left(App.Path, Len(App.Path) - 11)
    strConn = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &
strPath & "\DATA\ADS.mdb;Persist Security Info=False"
End Sub
Public Function FetchSQL(ByVal strSQL As String) As ADODB.Recordset
On Error GoTo FetchSQL_ERROR
Dim rs As ADODB.Recordset
    Set rs = CreateObject("ADODB.Recordset")
    With rs
        .ActiveConnection = strConn
        .CursorLocation = adUseClient
        .CursorType = adOpenForwardOnly
        .LockType = adLockOptimistic
        .Source = UCase(strSQL)
        .Open
        .ActiveConnection = Nothing
    End With
    Set FetchSQL = rs
    Set rs = Nothing
    Exit Function
FetchSQL_ERROR:
    MsgBox Err.Number & " : " & vbCrLf & Err.Description,
vbInformation, Err.Source
    'Err.Raise Err.Number, Err.Source, Err.Description
    Set rs = Nothing
    Set FetchSQL = Nothing
End Function
Public Function FetchEmpty(ByVal strSQL As String) As ADODB.Recordset
On Error GoTo FetchEmpty_ERROR
Dim rs As ADODB.Recordset
    Set rs = CreateObject("ADODB.Recordset")
    With rs
        .ActiveConnection = strConn
        .CursorLocation = adUseClient
        .CursorType = adOpenForwardOnly
        .LockType = adLockOptimistic
        .Source = UCase(strSQL)
        .Open
        .ActiveConnection = Nothing
    End With
    Set FetchEmpty = rs
    Set rs = Nothing
    Exit Function
FetchEmpty_ERROR:
    MsgBox Err.Number & " : " & vbCrLf & Err.Description,
vbInformation, Err.Source
    'Err.Raise Err.Number, Err.Source, Err.Description
    Set rs = Nothing
    Set FetchEmpty = Nothing
End Function
Public Function ExecSQL(ByVal strSQL As String) As Boolean
On Error GoTo ExecSQL_ERROR
Dim cmd As ADODB.Command
    Set cmd = CreateObject("ADODB.Command")
    With cmd
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        .ActiveConnection = strConn
        .CommandText = strSQL
        .CommandType = adCmdText
        .Execute
        .ActiveConnection = Nothing
    End With
    Set cmd = Nothing
    ExecSQL = True
    Exit Function
ExecSQL_ERROR:
    MsgBox Err.Number & " : " & vbCrLf & Err.Description,
vbInformation, Err.Source
    'Err.Raise Err.Number, Err.Source, Err.Description
    Set cmd = Nothing
    ExecSQL = False
End Function
Public Function rpRSQuoted(rs As ADODB.Recordset) As ADODB.Recordset
Dim i As Long
    For i = 0 To rs.Fields.Count - 1
        If Not IsNull(rs(i)) Then
            rs(i) = Replace(rs(i), Chr(39), Chr(39) & Chr(39))
        End If
    Next
    Set rpRSQuoted = rs
End Function
Public Function rpQuoted(ByVal str As String) As String
    rpQuoted = Replace(str, Chr(39), Chr(39) & Chr(39))
End Function

Private Sub Class_Initialize()
    Set ADO = New ADOService
End Sub
Private Sub Class_Terminate()
    Set ADO = Nothing
End Sub
Private Function rpQuoted(ByVal str As String) As String
    rpQuoted = Replace(str, Chr(39), Chr(39) & Chr(39))
End Function
' PUBLIC FUNCTION / SUB
Public Function SelectAll() As ADODB.Recordset
Dim rs As ADODB.Recordset
Dim strSQL As String
    strSQL = _
            "SELECT T.ID ,T.SEQ ,T.FILE_PATH ,T.IS_INACTIVE " & _
            "FROM MULTIMEDIA T " & _
            "ORDER BY T.SEQ "
    Set rs = ADO.FetchSQL(strSQL)
    Set SelectAll = rs
    Set rs = Nothing
End Function
Public Function SelectOne(ByVal sCode As String) As ADODB.Recordset
Dim rs As ADODB.Recordset
Dim strSQL As String
    strSQL = _
            "SELECT T.ID ,T.SEQ ,T.FILE_PATH ,T.IS_INACTIVE " & _
            "FROM MULTIMEDIA T " & _
            "WHERE T.ID = " & sCode & " "
    Set rs = ADO.FetchSQL(strSQL)
    Set SelectOne = rs
    Set rs = Nothing
End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Public Function SelectBySeq() As ADODB.Recordset
Dim rs As ADODB.Recordset
Dim strSQL As String
    strSQL = _
            "SELECT T.ID ,T.SEQ ,T.FILE_PATH ,T.IS_INACTIVE " & _
            "FROM MULTIMEDIA T " & _
            "WHERE T.ID > 0 " & _
            "AND T.IS_INACTIVE = 0 " & _
            "ORDER BY T.SEQ "
    Set rs = ADO.FetchSQL(strSQL)
    Set SelectBySeq = rs
    Set rs = Nothing
End Function
Public Function SelectBySeqEx() As ADODB.Recordset
Dim rs As ADODB.Recordset
Dim strSQL As String
    strSQL = _
            "SELECT T.ID ,T.SEQ ,T.FILE_PATH ,T.IS_INACTIVE " & _
            "FROM MULTIMEDIA T " & _
            "WHERE T.ID > 0 " & _
            "AND T.SEQ <> 0 " & _
            "ORDER BY T.SEQ "
    Set rs = ADO.FetchSQL(strSQL)
    Set SelectBySeqEx = rs
    Set rs = Nothing
End Function
Public Function SelectByPath(ByVal fileName As String) As
ADODB.Recordset
Dim rs As ADODB.Recordset
Dim strSQL As String
    strSQL = _
            "SELECT T.ID ,T.SEQ ,T.FILE_PATH ,T.IS_INACTIVE " & _
            "FROM MULTIMEDIA T " & _
            "WHERE T.FILE_PATH = '" & rpQuoted(fileName) & "' " & _
            "AND T.IS_INACTIVE = 0 "
    Set rs = ADO.FetchSQL(strSQL)
    Set SelectByPath = rs
    Set rs = Nothing
End Function
Public Function GetID(ByVal fileName As String) As Integer
Dim rs As ADODB.Recordset
    Set rs = SelectByPath(fileName)
    If Not rs Is Nothing Then
        GetID = rs("ID")
    End If
    Set rs = Nothing
End Function
Public Function IsExist(ByVal sCode As String) As Boolean
Dim rs As ADODB.Recordset
Dim strSQL As String
    Set rs = SelectOne(sCode)
    IsExist = rs.RecordCount > 0
    Set rs = Nothing
End Function
Public Function IsPathExist(ByVal fileName As String) As Boolean
Dim rs As ADODB.Recordset
Dim strSQL As String
    Set rs = SelectByPath(fileName)
    IsPathExist = rs.RecordCount > 0
    Set rs = Nothing
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Function
Public Function SelectEmpty() As ADODB.Recordset
Dim strSQL As String
    strSQL = _
            "SELECT T.ID ,T.SEQ ,T.FILE_PATH ,T.IS_INACTIVE " & _
            "FROM MULTIMEDIA T " & _
            "WHERE 'A' = 'B' "
    Set SelectEmpty = ADO.FetchEmpty(strSQL)
End Function
Public Function Delete(ByVal sCode As String) As Boolean
Dim strSQL As String
    strSQL = _
            "DELETE FROM MULTIMEDIA " & _
            "WHERE ID = " & sCode & " "
    Delete = ADO.ExecSQL(strSQL)
End Function
Public Function Create(rs As ADODB.Recordset) As Boolean
Dim strSQL As String
    Set rs = ADO.rpRSQuoted(rs)
    strSQL = _
            "INSERT INTO MULTIMEDIA " & _
            "(SEQ ,FILE_PATH ,IS_INACTIVE) " & _
            "VALUES (" & rs(1) & " ,'" & ADO.rpQuoted(rs(2)) & "' ,"
& rs(3) & ") "
    Create = ADO.ExecSQL(strSQL)
End Function
Public Function Update(rs As ADODB.Recordset) As Boolean
Dim strSQL As String
    Set rs = ADO.rpRSQuoted(rs)
    strSQL = _
            "UPDATE MULTIMEDIA SET " & _
            "SEQ = " & rs(1) & " ," & _
            "FILE_PATH = '" & ADO.rpQuoted(rs(2)) & "' ," & _
            "IS_INACTIVE = " & rs(3) & " " & _
            "WHERE ID = " & rs(0) & " "
    Update = ADO.ExecSQL(strSQL)
End Function
Public Function SetActiveFlag(ByVal flagActive As Boolean, ByVal
fileName As String) As Boolean
Dim strSQL As String
    strSQL = _
            "UPDATE MULTIMEDIA SET " & _
            "IS_INACTIVE = " & IIf(flagActive, "0", "1") & " " & _
            "WHERE FILE_PATH = '" & rpQuoted(fileName) & "' "
    SetActiveFlag = ADO.ExecSQL(strSQL)
End Function
Public Function WriteStat(ByVal fileName As String) As Boolean
Dim strSQL As String
Dim id As Integer
    id = GetID(fileName)
    strSQL = _
            "INSERT INTO MULTIMEDIA_STAT " & _
            "(MULTIMEDIA_ID ,PLAY_DATE) " & _
            "VALUES (" & id & " ,'" & Now & "')"
    WriteStat = ADO.ExecSQL(strSQL)
End Function
Public Function ReadStat(ByVal id As Integer) As ADODB.Recordset
Dim rs As ADODB.Recordset
Dim strSQL As String
    strSQL = _
            "SELECT T.ID ,T.MULTIMEDIA_ID ,T.PLAY_DATE " & _
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            "FROM MULTIMEDIA_STAT T " & _
            "WHERE T.MULTIMEDIA_ID = " & id & " " & _
            "ORDER BY T.PLAY_DATE DESC"
    Set rs = ADO.FetchSQL(strSQL)
    Set ReadStat = rs
    Set rs = Nothing
End Function
Public Function ResetStat(ByVal id As Integer) As Boolean
Dim strSQL As String
    strSQL = _
            "DELETE FROM MULTIMEDIA_STAT " & _
            "WHERE MULTIMEDIA_ID = " & id & " "
    ResetStat = ADO.ExecSQL(strSQL)
End Function
Public Function ResetAllStats() As Boolean
Dim strSQL As String
    strSQL = _
             "DELETE FROM MULTIMEDIA_STAT "
    ResetAllStats = ADO.ExecSQL(strSQL)
End Function
Public Function ResetFile(ByVal id As Integer) As Boolean
Dim strSQL As String
Dim ret As Boolean
    strSQL = _
             "DELETE FROM MULTIMEDIA " & _
             "WHERE ID = " & id & " "
    ret = ADO.ExecSQL(strSQL)
    If ret Then Call ResetStat(id)
    ResetFile = ret
End Function
Public Function ResetAllFiles() As Boolean
Dim strSQL As String
Dim ret As Boolean
    strSQL = _
             "DELETE FROM MULTIMEDIA "
    ret = ADO.ExecSQL(strSQL)
    If ret Then Call ResetAllStats
    ResetAllFiles = ret
End Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
รายละเอียดของอุปกรณ์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

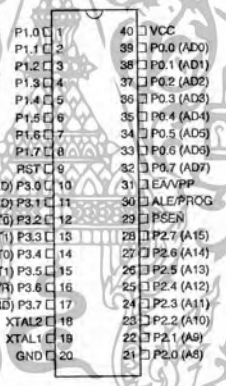
- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

Description

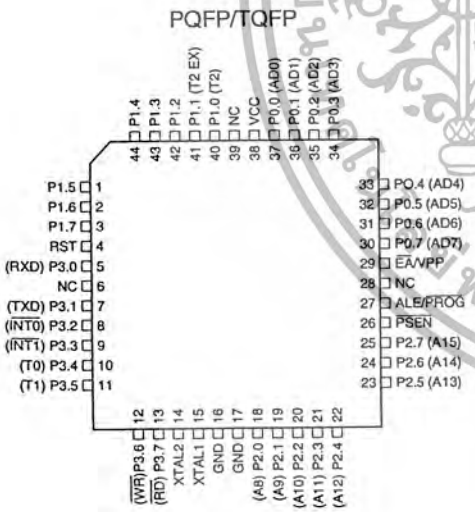
The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

Pin Configurations

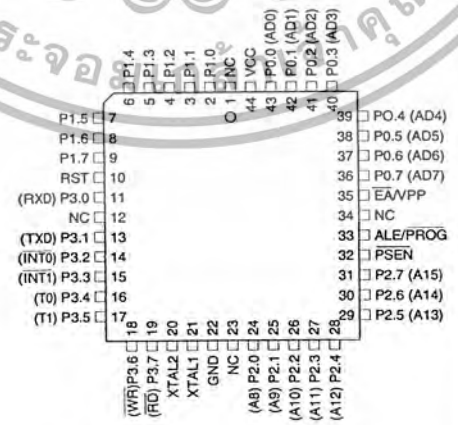
PDIP



PQFP/TQFP



PLCC



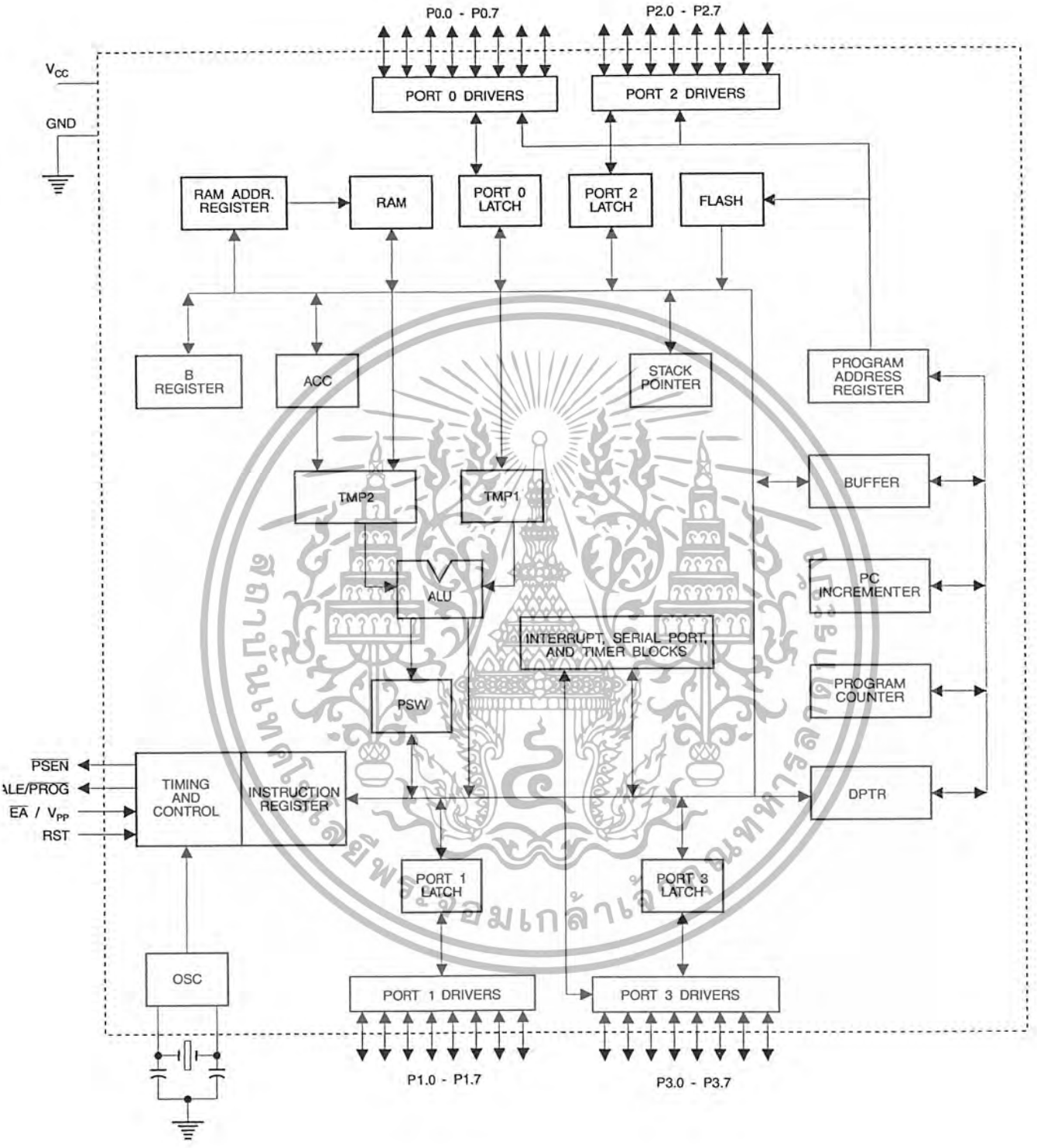
8-bit Microcontroller with 4K Bytes Flash

AT89C51

Rev. 0265G–02/00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power-down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open-drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

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

Port 3 also receives some control signals for Flash programming and verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pulse is skipped during each access to external Data Memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{\text{PSEN}}$

Program Store Enable is the read strobe to external program memory.

When the AT89C51 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.

$\overline{\text{EA}}$/VPP

External Access Enable. $\overline{\text{EA}}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{\text{EA}}$ will be internally latched on reset.

$\overline{\text{EA}}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming, for parts that require 12-volt V_{PP}.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

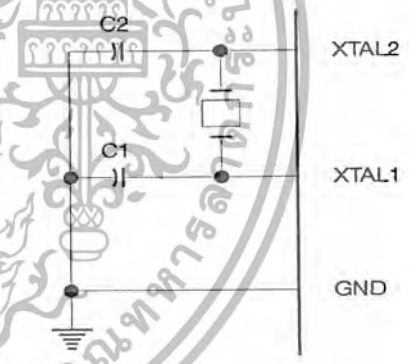
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

It should be noted that when idle is terminated by a hard ware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Figure 1. Oscillator Connections



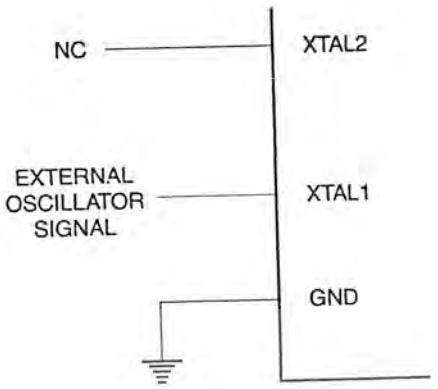
Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	$\overline{\text{PSEN}}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 2. External Clock Drive Configuration



Power-down Mode

In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below.

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of $\overline{EA}$ be in agreement with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

Program Lock Bits				
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash is disabled
3	P	P	U	Same as mode 2, also verify is disabled
4	P	P	P	Same as mode 3, also external execution is disabled

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low-voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third-party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V_{PP} = 12V	V_{PP} = 5V
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H) = 1EH (031H) = 51H (032H) =F FH	(030H) = 1EH (031H) = 51H (032H) = 05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figure 3 and Figure 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}/V_{PP}$ to 12V for the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{Data}$ Polling: The AT89C51 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/$\overline{PROG}$ low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 51H indicates 89C51
(032H) = FFH indicates 12V programming
(032H) = 05H indicates 5V programming

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming Modes

Mode		RST	PSEN	ALE/PROG	EA/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L		H/12V	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L		H/12V	H	H	H	H
	Bit - 2	H	L		H/12V	H	H	L	L
	Bit - 3	H	L		H/12V	H	L	H	L
Chip Erase		H	L	(1)	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10 ms PROG pulse.

Figure 3. Programming the Flash

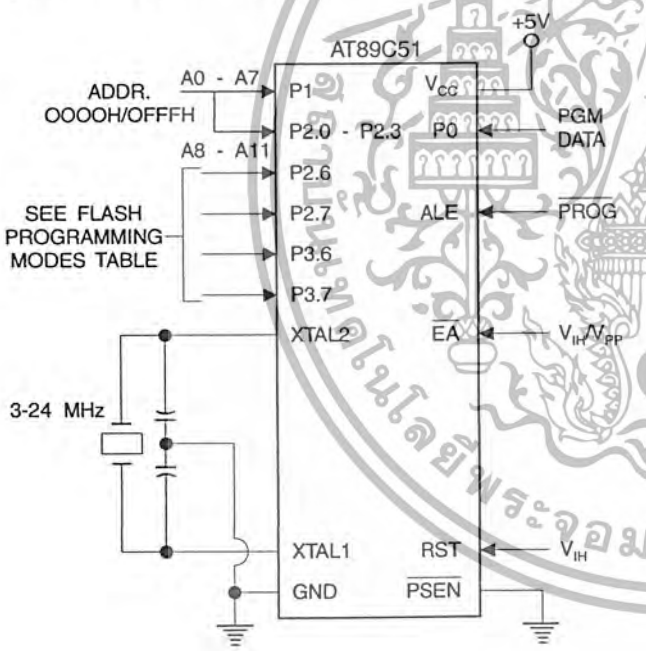
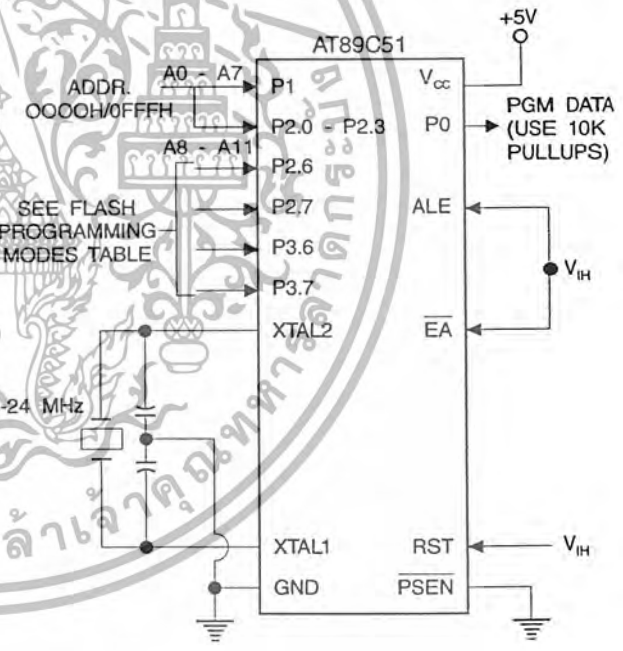
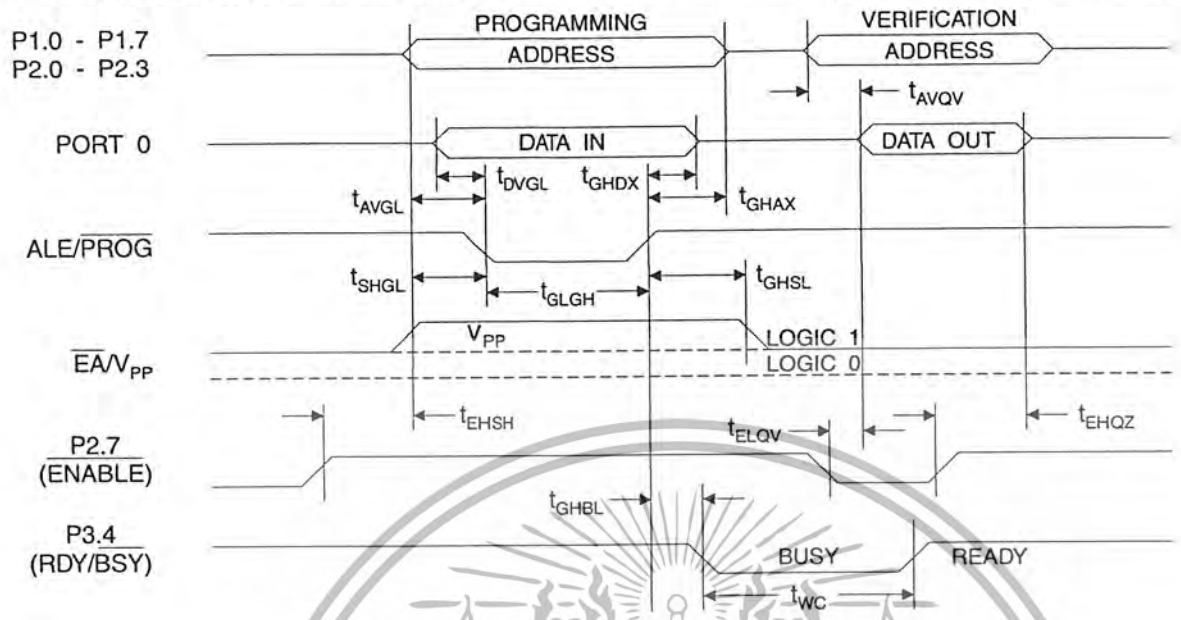


Figure 4. Verifying the Flash

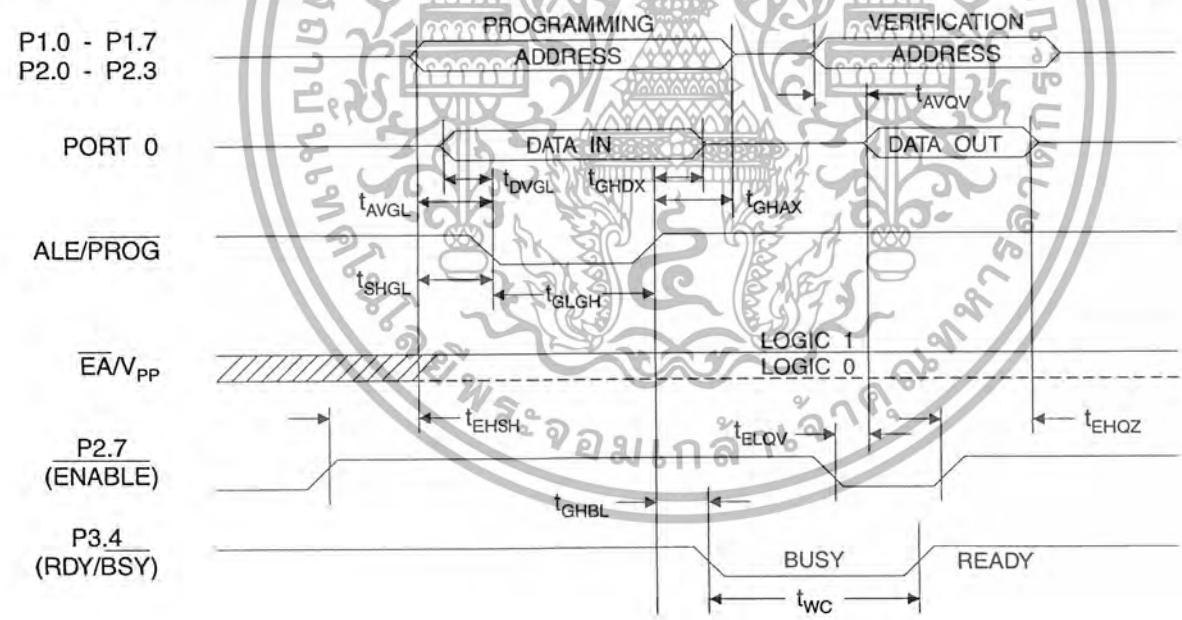


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Waveforms - High-voltage Mode (V_{PP} = 12V)



Flash Programming and Verification Waveforms - Low-voltage Mode (V_{PP} = 5V)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Characteristics

T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V_{PP}(1)	Programming Enable Voltage	11.5	12.5	V
I_{PP}(1)	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}(1)	V_{PP} Hold After $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

Note: 1. Only used in 12-volt programming mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 5.0V ± 20% (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC} - 0.1	V
V_{IL1}	Input Low-voltage ($\overline{EA}$)		-0.5	0.2 V_{CC} - 0.3	V
V_{IH}	Input High-voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
V_{IH1}	Input High-voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
V_{OL}	Output Low-voltage[1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low-voltage[1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High-voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High-voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, VCC = 5V ± 10%		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pull-down Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power-down Mode[2]	V_{CC} = 6V		100	µA
		V_{CC} = 3V		40	µA

Notes:
1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port: Port 0: 26 mA
   Ports 1, 2, 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Characteristics

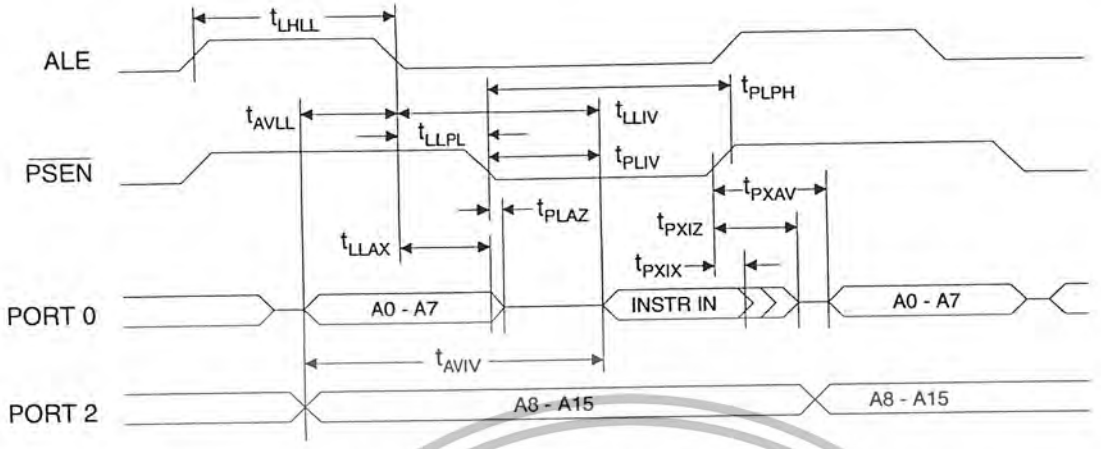
Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

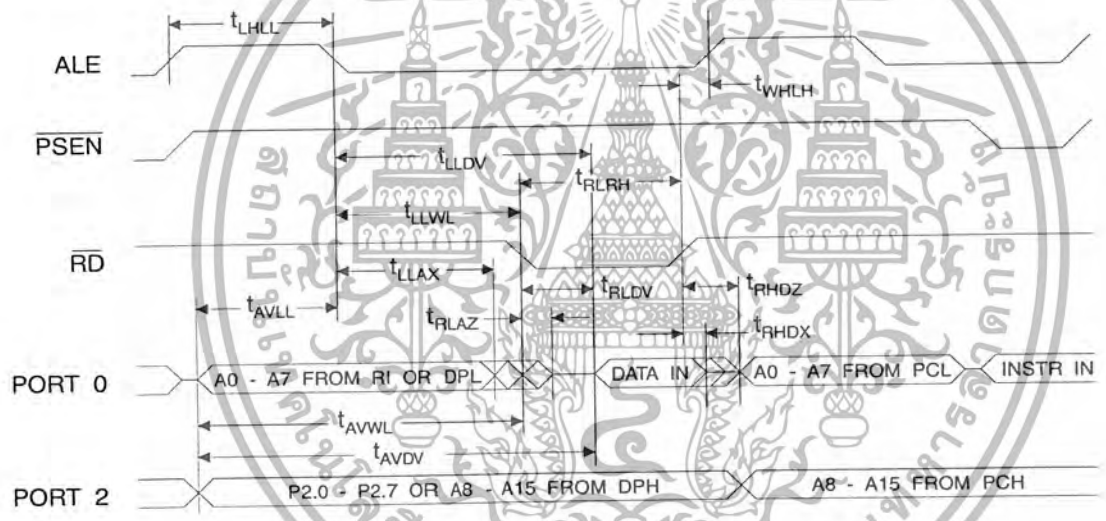
Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	43		t_{CLCL}-13		ns
t_{LLAX}	Address Hold After ALE Low	48		t_{CLCL}-20		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}$-65	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		t_{CLCL}-13		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}$-20		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}$-45	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		t_{CLCL}-10	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		t_{CLCL}-8		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}$-55	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL}$-90	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL}$-28	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}$-150	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}$-165	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}$-50	$3t_{CLCL}$+50	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL}$-75		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		t_{CLCL}-20		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}$-120		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		t_{CLCL}-20		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	t_{CLCL}-20	t_{CLCL}+25	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Program Memory Read Cycle

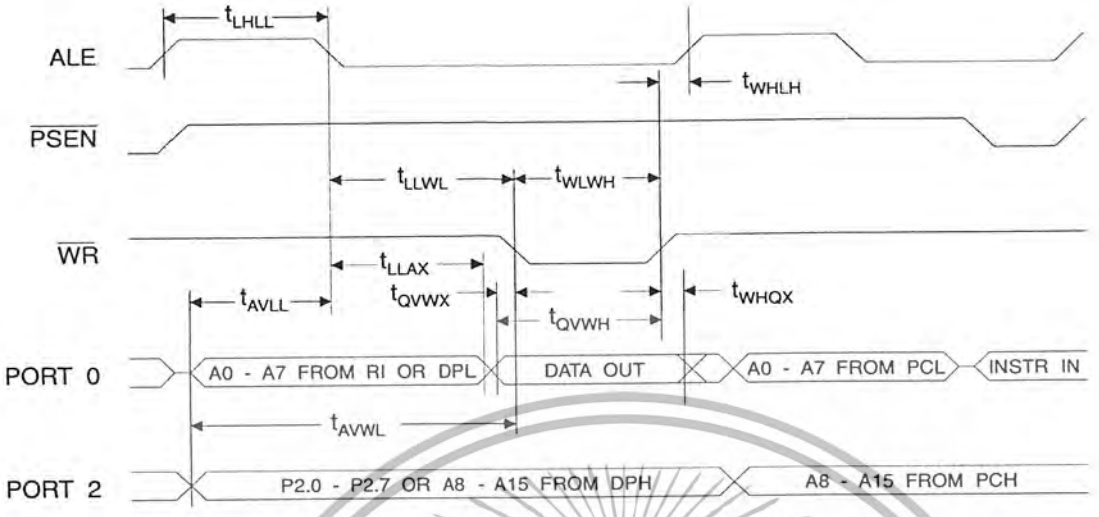


External Data Memory Read Cycle

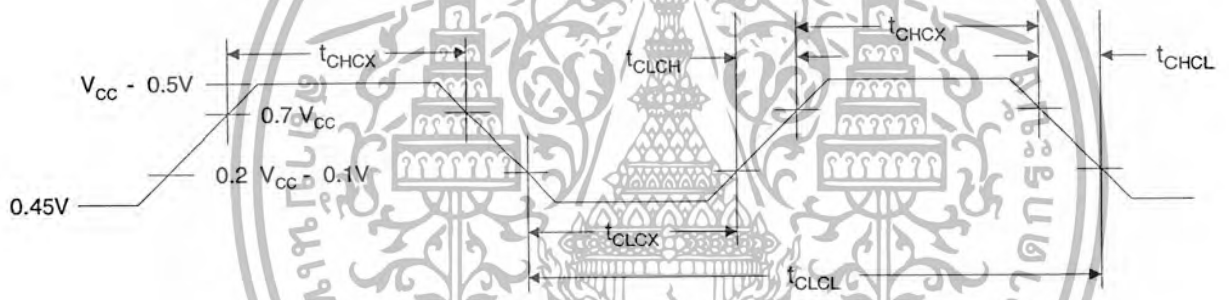


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

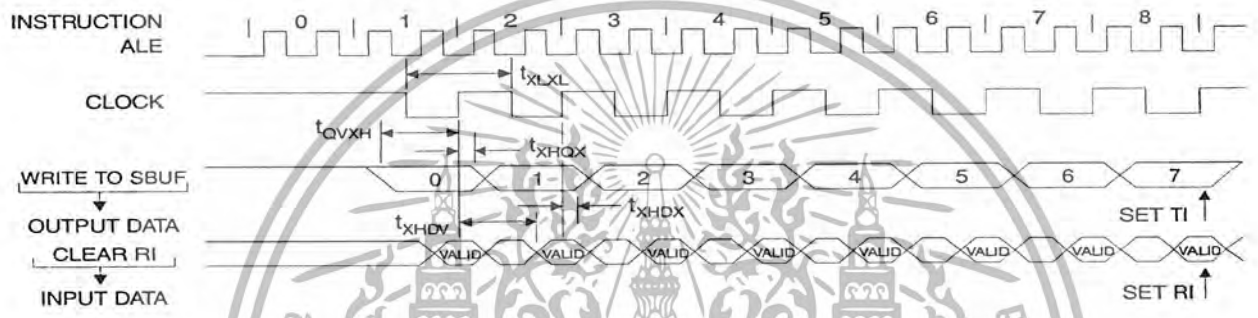
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

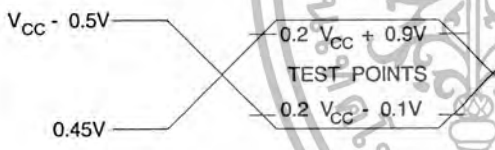
(V_{CC} = 5.0 V ± 20%; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-117		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

Shift Register Mode Timing Waveforms

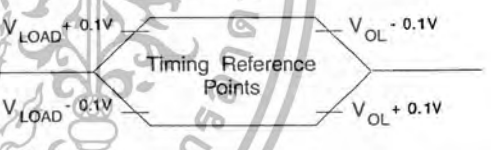


AC Testing Input/Output Waveforms[1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms[1]



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

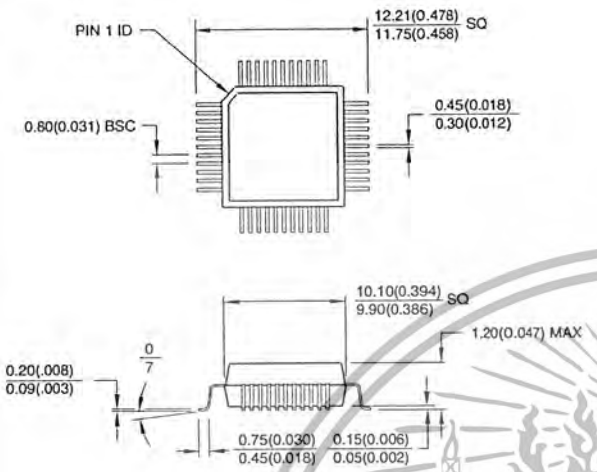
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5V ± 20%	AT89C51-12AC	44A	Commercial (0°C to 70°C)
		AT89C51-12JC	44J	
		AT89C51-12PC	40P6	
		AT89C51-12QC	44Q	
		AT89C51-12AI	44A	Industrial (-40°C to 85°C)
		AT89C51-12JI	44J	
		AT89C51-12PI	40P6	
		AT89C51-12QI	44Q	
16	5V ± 20%	AT89C51-16AC	44A	Commercial (0°C to 70°C)
		AT89C51-16JC	44J	
		AT89C51-16PC	40P6	
		AT89C51-16QC	44Q	
		AT89C51-16AI	44A	Industrial (-40°C to 85°C)
		AT89C51-16JI	44J	
		AT89C51-16PI	40P6	
		AT89C51-16QI	44Q	
20	5V ± 20%	AT89C51-20AC	44A	Commercial (0°C to 70°C)
		AT89C51-20JC	44J	
		AT89C51-20PC	40P6	
		AT89C51-20QC	44Q	
		AT89C51-20AI	44A	Industrial (-40°C to 85°C)
		AT89C51-20JI	44J	
		AT89C51-20PI	40P6	
		AT89C51-20QI	44Q	
24	5V ± 20%	AT89C51-24AC	44A	Commercial (0°C to 70°C)
		AT89C51-24JC	44J	
		AT89C51-24PC	40P6	
		AT89C51-24QC	44Q	
		AT89C51-24AI	44A	Industrial (-40°C to 85°C)
		AT89C51-24JI	44J	
		AT89C51-24PI	40P6	
		AT89C51-24QI	44Q	

Package Type	
44A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
40P6	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44-lead, Plastic Gull Wing Quad Flatpack (PQFP)

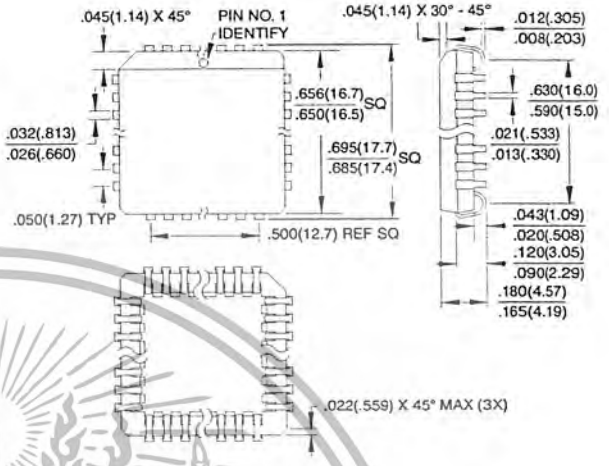
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Packaging Information

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flatpack (TQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-026 ACB

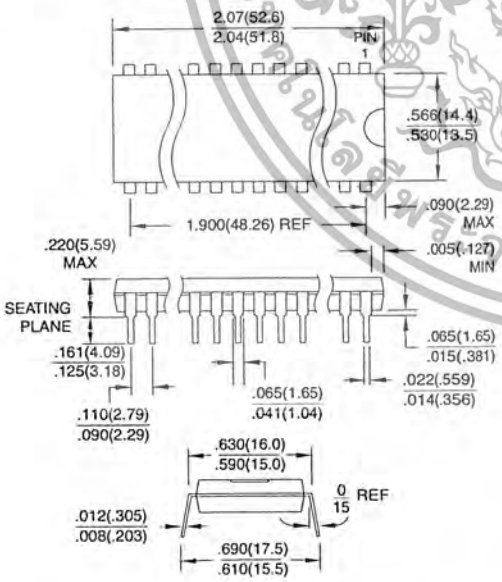


44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-018 AC

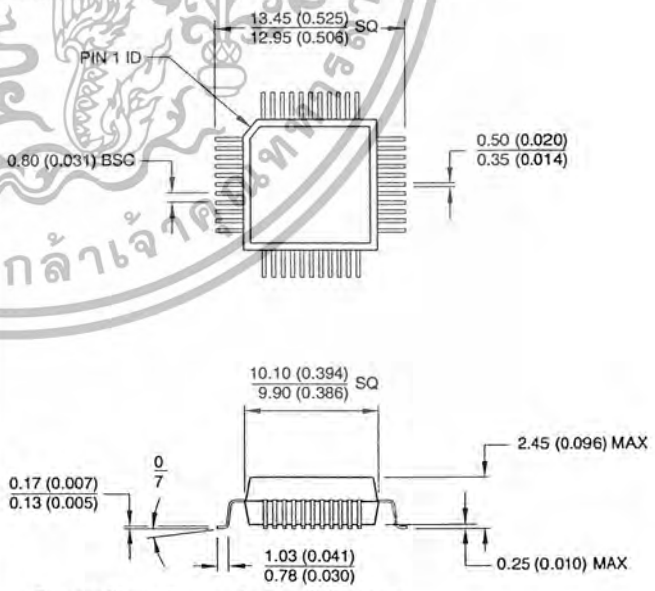


Controlling dimension: millimeters

40P6, 40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)



44Q, 44-lead, Plastic Quad Flat Package (PQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-022 AB



Controlling dimension: millimeters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Fax-on-Demand
North America:
1-(800) 292-8635

International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

© Atmel Corporation 2000.
Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.

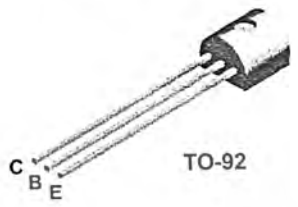
Printed on recycled paper.

0265G–02/00/xM

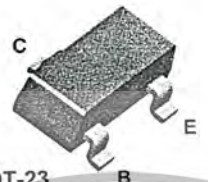
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FAIRCHILD
SEMICONDUCTOR™

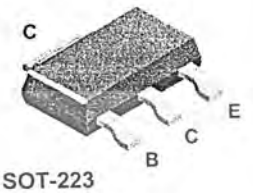
2N3904 MMBT3904 PZT3904



TO-92



SOT-23
Mark: 1A



SOT-223

NPN General Purpose Amplifier

This device is designed as a general purpose amplifier and switch. The useful dynamic range extends to 100 mA as a switch and to 100 MHz as an amplifier.

Absolute Maximum Ratings*

T_A = 25°C unless otherwise noted

Symbol	Parameter	Value	Units
V_{CEO}	Collector-Emitter Voltage	40	V
V_{CBO}	Collector-Base Voltage	60	V
V_{EBO}	Emitter-Base Voltage	6.0	V
I_C	Collector Current - Continuous	200	mA
T_J, T_{stg}	Operating and Storage Junction Temperature Range	-55 to +150	°C

*These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

NOTES:
1) These ratings are based on a maximum junction temperature of 150 degrees C.
2) These are steady state limits. The factory should be consulted on applications involving pulsed or low duty cycle operations.

Thermal Characteristics

T_A = 25°C unless otherwise noted

Symbol	Characteristic	Max			Units
		2N3904	*MMBT3904	**PZT3904	
P_D	Total Device Dissipation	625	350	1,000	mW
	Derate above 25°C	5.0	2.8	8.0	mW/°C
$R_{\theta JC}$	Thermal Resistance, Junction to Case	83.3			°C/W
$R_{\theta JA}$	Thermal Resistance, Junction to Ambient	200	357	125	°C/W

*Device mounted on FR-4 PCB 1.6" X 1.6" X 0.06."

**Device mounted on FR-4 PCB 36 mm X 18 mm X 1.5 mm; mounting pad for the collector lead min. 6 cm^2.

© 2001 Fairchild Semiconductor Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NPN General Purpose Amplifier
(continued)

Electrical Characteristics
T_A = 25°C unless otherwise noted

Symbol	Parameter	Test Conditions	Min	Max	Units
OFF CHARACTERISTICS					
$V_{(BR)CEO}$	Collector-Emitter Breakdown Voltage	I_C = 1.0 mA, I_B = 0	40		V
$V_{(BR)CBO}$	Collector-Base Breakdown Voltage	I_C = 10 µA, I_E = 0	60		V
$V_{(BR)EBO}$	Emitter-Base Breakdown Voltage	I_E = 10 µA, I_C = 0	6.0		V
I_{BL}	Base Cutoff Current	V_{CE} = 30 V, V_{EB} = 3V		50	nA
I_{CEX}	Collector Cutoff Current	V_{CE} = 30 V, V_{EB} = 3V		50	nA
ON CHARACTERISTICS*					
h_{FE}	DC Current Gain	I_C = 0.1 mA, V_{CE} = 1.0 V I_C = 1.0 mA, V_{CE} = 1.0 V I_C = 10 mA, V_{CE} = 1.0 V I_C = 50 mA, V_{CE} = 1.0 V I_C = 100 mA, V_{CE} = 1.0 V	40 70 100 60 30	 300 	
$V_{CE(sat)}$	Collector-Emitter Saturation Voltage	I_C = 10 mA, I_B = 1.0 mA I_C = 50 mA, I_B = 5.0 mA		0.2 0.3	V V
$V_{BE(sat)}$	Base-Emitter Saturation Voltage	I_C = 10 mA, I_B = 1.0 mA I_C = 50 mA, I_B = 5.0 mA	0.65	0.85 0.95	V V
SMALL SIGNAL CHARACTERISTICS					
f_T	Current Gain - Bandwidth Product	I_C = 10 mA, V_{CE} = 20 V, f = 100 MHz	300		MHz
C_{obo}	Output Capacitance	V_{CB} = 5.0 V, I_E = 0, f = 1.0 MHz		4.0	pF
C_{ibo}	Input Capacitance	V_{EB} = 0.5 V, I_C = 0, f = 1.0 MHz		8.0	pF
NF	Noise Figure	I_C = 100 µA, V_{CE} = 5.0 V, R_S = 1.0kΩ, f=10 Hz to 15.7kHz		5.0	dB
SWITCHING CHARACTERISTICS					
t_d	Delay Time	V_{CC} = 3.0 V, V_{BE} = 0.5 V,		35	ns
t_r	Rise Time	I_C = 10 mA, I_{B1} = 1.0 mA		35	ns
t_s	Storage Time	V_{CC} = 3.0 V, I_C = 10mA		200	ns
t_f	Fall Time	I_{B1} = I_{B2} = 1.0 mA		50	ns

*Pulse Test: Pulse Width ≤ 300 µs, Duty Cycle ≤ 2.0%

Spice Model

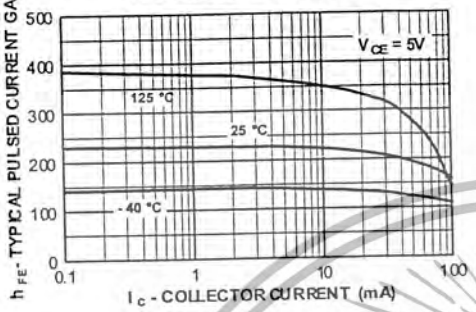
NPN (Is=6.734f Xti=3 Eg=1.11 Vaf=74.03 Bf=416.4 Ne=1.259 Ise=6.734 Ikf=66.78m Xtb=1.5 Br=.7371 Nc=2 Isc=0 Ikr=0 Rc=1 Cjc=3.638p Mjc=.3085 Vjc=.75 Fc=.5 Cje=4.493p Mje=.2593 Vje=.75 Tr=239.5n Tf=301.2p Itf=.4 Vtf=4 Xtf=2 Rb=10)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

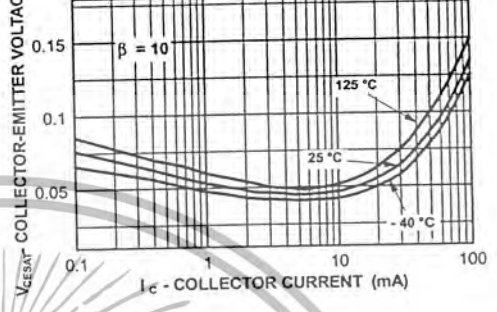
NPN General Purpose Amplifier
(continued)

Typical Characteristics

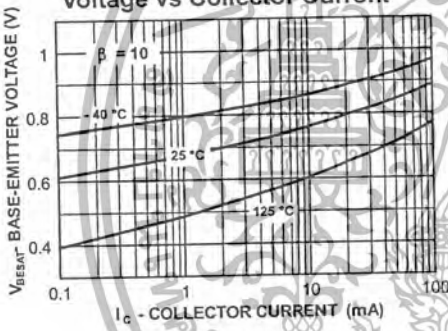
Typical Pulsed Current Gain vs Collector Current



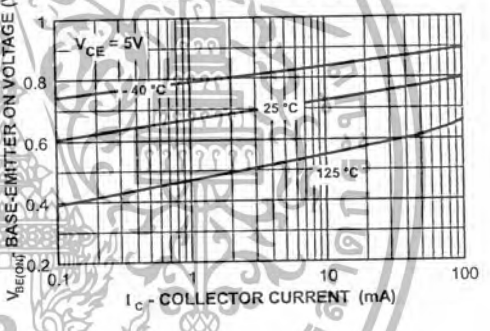
Collector-Emitter Saturation Voltage vs Collector Current



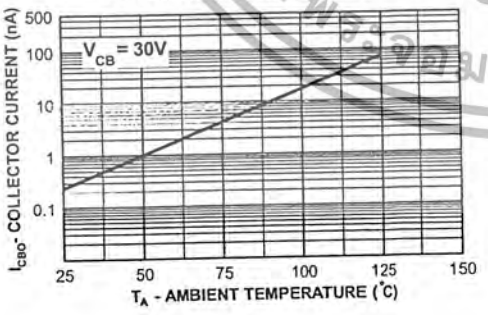
Base-Emitter Saturation Voltage vs Collector Current



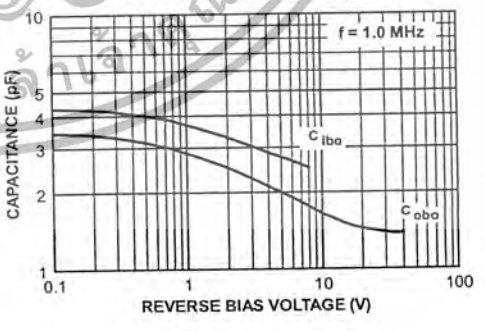
Base-Emitter ON Voltage vs Collector Current



Collector-Cutoff Current vs Ambient Temperature



Capacitance vs Reverse Bias Voltage

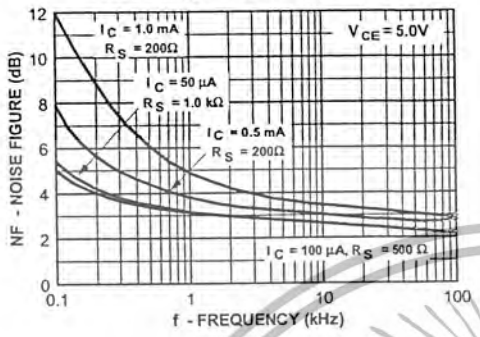


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

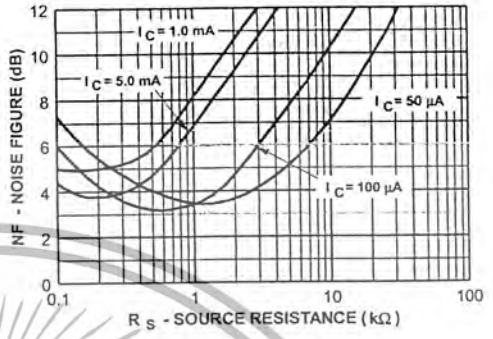
NPN General Purpose Amplifier
(continued)

Typical Characteristics (continued)

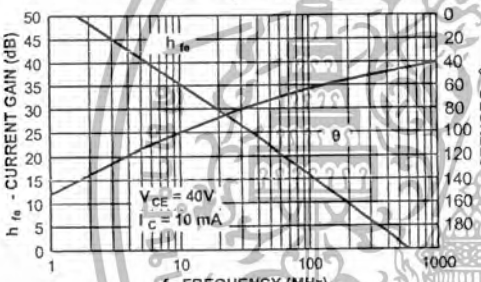
Noise Figure vs Frequency



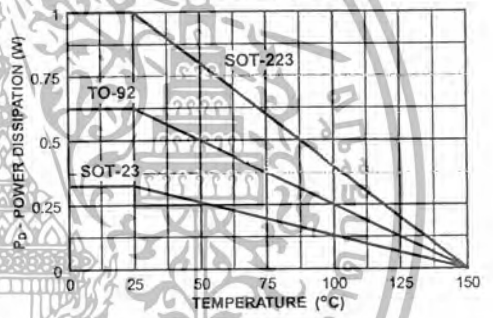
Noise Figure vs Source Resistance



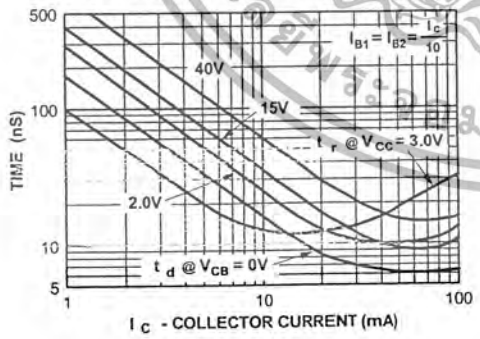
Current Gain and Phase Angle vs Frequency



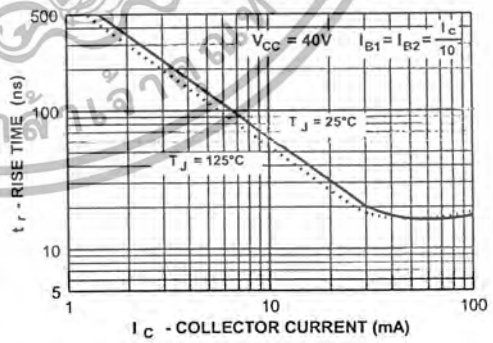
Power Dissipation vs Ambient Temperature



Turn-On Time vs Collector Current



Rise Time vs Collector Current

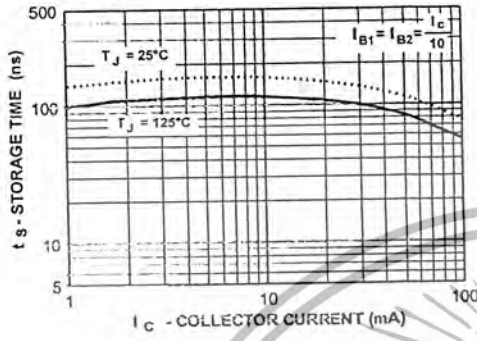


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

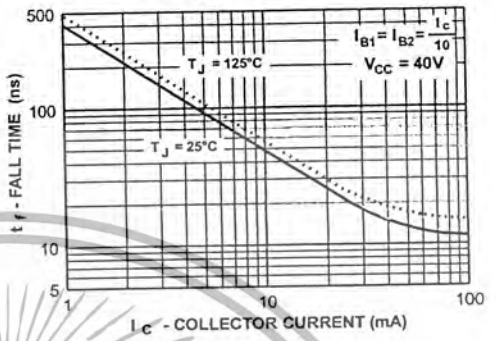
NPN General Purpose Amplifier
(continued)

Typical Characteristics (continued)

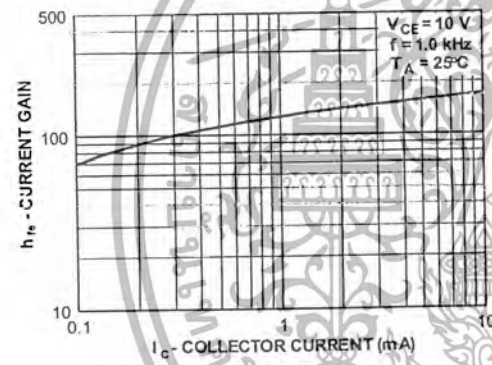
Storage Time vs Collector Current



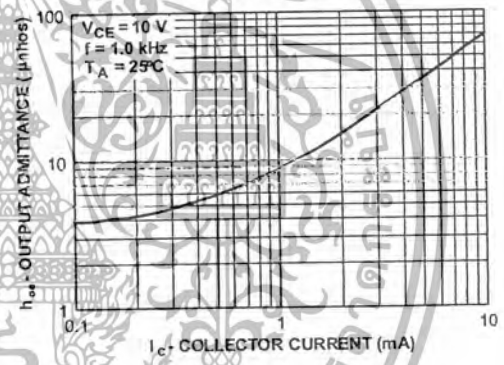
Fall Time vs Collector Current



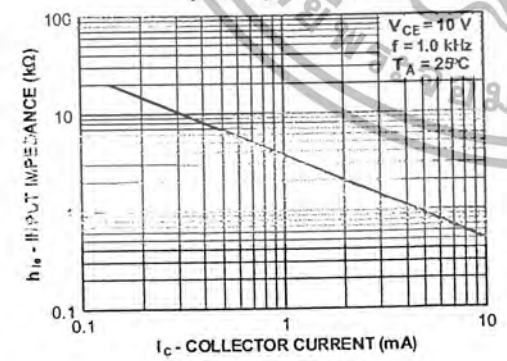
Current Gain



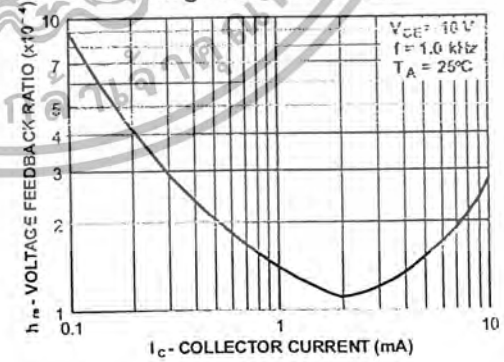
Output Admittance



Input Impedance



Voltage Feedback Ratio



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRADEMARKS

The following are registered and unregistered trademarks Fairchild Semiconductor owns or is authorized to use and is not intended to be an exhaustive list of all such trademarks.

ACEx™
Bottomless™
CoolFET™
CROSSVOLT™
DOME™
E²CMOS™
EnSigna™
FACT™
FACT Quiet Series™
FAST®
FASTr™
GlobalOptoisolator™
GTO™
HiSeC™
ISOPLANAR™
MICROWIRE™
OPTOLOGIC™
OPTOPLANAR™
PACMAN™
POP™
PowerTrench®
QFET™
QS™
QT Optoelectronics™
Quiet Series™
SILENT SWITCHER®
SMART START™
SuperSOT™-3
SuperSOT™-6
SuperSOT™-8
SyncFET™
TinyLogic™
UHC™
VCX™

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR CORPORATION.

As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, or (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

PRODUCT STATUS DEFINITIONS

Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This datasheet contains preliminary data, and supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
No Identification Needed	Full Production	This datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
Obsolete	Not In Production	This datasheet contains specifications on a product that has been discontinued by Fairchild semiconductor. The datasheet is printed for reference information only.

Rev. G

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SGS-THOMSON MICROELECTRONICS

BD135
BD137/BD139

NPN SILICON TRANSISTOR

- SGS-THOMSON PREFERRED SALESTYPES

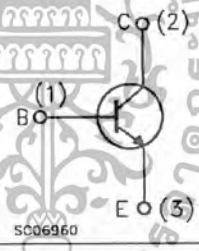
DESCRIPTION

The BD135, BD137 and BD139 are silicon epitaxial planar NPN transistors in Jedec SOT-32 plastic package, designed for audio amplifiers and drivers utilizing complementary or quasi compementary circuits.

The complementary PNP types are the BD136 BD138 and BD140.

SOT-32

INTERNAL SCHEMATIC DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value			Unit
		BD135	BD137	BD139	
V_{CBO}	Collector-Base Voltage (I_E = 0)	45	60	80	V
V_{CEO}	Collector-Emitter Voltage (I_B = 0)	45	60	80	V
V_{EBO}	Emitter-Base Voltage (I_C = 0)	5			V
I_C	Collector Current	1.5			A
I_{CM}	Collector Peak Current	3			A
I_B	Base Current	0.5			A
P_{tot}	Total Dissipation at $T_c \leq 25$ °C	12.5			W
P_{tot}	Total Dissipation at $T_{amb} \leq 25$ °C	1.25			W
T_{stg}	Storage Temperature	-65 to 150			°C
T_j	Max. Operating Junction Temperature	150			°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NPN General Purpose Amplifier
(continued)

Test Circuits

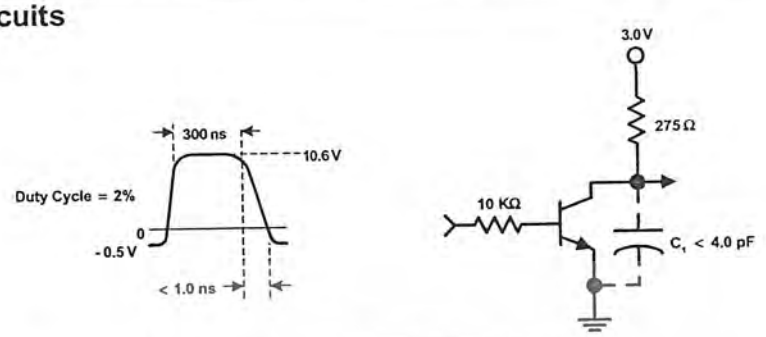


FIGURE 1: Delay and Rise Time Equivalent Test Circuit

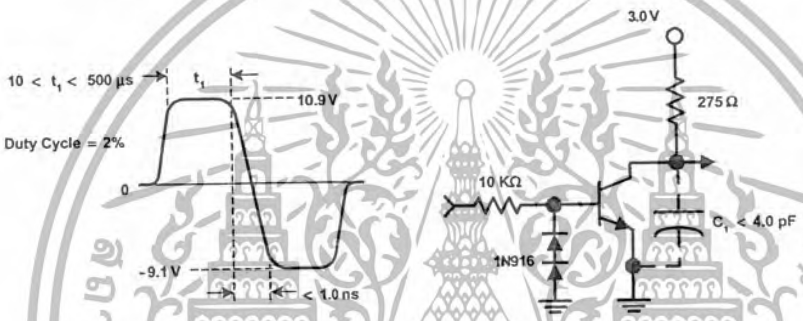


FIGURE 2: Storage and Fall Time Equivalent Test Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THERMAL DATA

$R_{thj\text{-}case}$	Thermal Resistance Junction-case	Max	10	°C/W

ELECTRICAL CHARACTERISTICS (T_{case} = 25 °C unless otherwise specified)

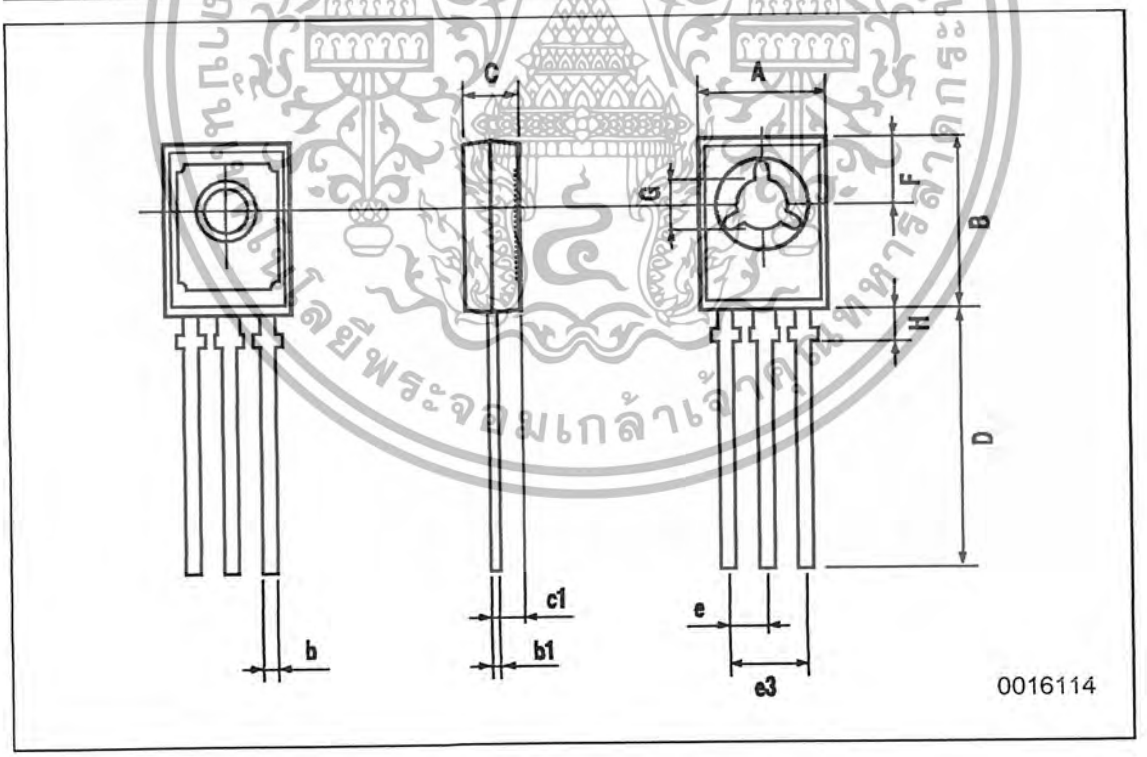
Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
I_{CBO}	Collector Cut-off Current (I_E = 0)	V_{CB} = 30 V V_{CB} = 30 V T_C = 125 °C			0.1 10	µA µA
I_{EBO}	Emitter Cut-off Current (I_C = 0)	V_{EB} = 5 V			10	µA
$V_{CEO(sus)}$*	Collector-Emitter Sustaining Voltage	I_C = 30 mA for BD135 for BD137 for BD139	 45 60 80			 V V V
$V_{CE(sat)}$*	Collector-Emitter Saturation Voltage	I_C = 0.5 A I_B = 0.05 A			0.5	V
V_{BE}*	Base-Emitter Voltage	I_C = 0.5 A V_{CE} = 2 V			1	V
h_{FE}*	DC Current Gain	I_C = 5 mA V_{CE} = 2 V I_C = 0.5 A V_{CE} = 2 V I_C = 150 mA V_{CE} = 2 V	25 25 40		 250	
h_{FE}	h_{FE} Groups	I_C = 150 mA V_{CE} = 2 V for BD139 group 10	 63		 160	

* Pulsed: Pulse duration = 300 µs, duty cycle 1.5 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOT-32 MECHANICAL DATA

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A	7.4		7.8	0.291		0.307
B	10.5		10.8	0.413		0.445
b	0.7		0.9	0.028		0.035
b1	0.49		0.75	0.019		0.030
C	2.4		2.7	0.04		0.106
c1		1.2			0.047	
D		15.7			0.618	
e		2.2			0.087	
e3		4.4			0.173	
F		3.8			0.150	
G	3		3.2	0.118		0.126
H			2.54			0.100



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Infrared Preamplifier IC

Bipolar integrated circuit, intended as a receiver preamplifier for Central Control Unit for the infrared remote-control systems designed with integrated circuits of ITT.

The TBA 2800 preamplifier IC contains four main parts: the gain-controlled amplifier I, the amplifier II, the pulse-separating amplifier III, and the inverter IV, as shown in Fig. 1. The amplifier I has a wide dynamic range and thus ensures interference-free operation, also at bright ambient light, 50-Hz-modulated light originating from fluorescent lamps, or at intensive infrared light, e. g. produced by infrared sound transmission. It is also possible, to approach almost directly the remote-control transmitter to the receiver without producing misfunction by overdriving the receiver.

The amplifier II further amplifies the signal, and amplifier III separates the pulse-shaped intelligence signal from noise and other unwanted parts. The inverter IV provided additionally inverts the output signal available at pin 7 as negative pulse, and thus delivers positive output pulses at pin 8. If an additional resistor is inserted between pin 6 and GND, the noise-immunity is increased, but the input sensitivity decreased. Pin 10 serves as test pin and must not be connected.

The capacitor connected from pin 2 to ground influences the automatic gain control of amplifier I contained in the TBA 2800. A capacitance of less than 1 μF will cause misfunction in the distance range of 0.2 m to 2 m. Fig. 3 shows the dependence of the transmission range on the capacitor at pin 2. Due to tolerances of the TBA 2800 itself, the transmitter diodes' efficiency and the receiver diode's sensitivity, the curves of Fig. 3 must be considered with a tolerance of about ±50%. The curves have been established by simulation of the distance between transmitter and receiver by means of infrared filter glass with specified attenuation inserted between transmitter and receiver.

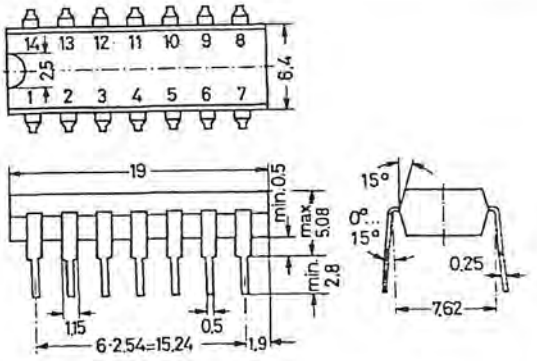


Fig. 2:
TBA 2800 in a 14-pin Dil Plastic Package TO-116, 20 A 14 according to DIN 41 866

Weight approx. 1.2 g
Dimensions in mm

Pin Connections

1	Input's Ground, 0
2	Capacitor Pin Amplifier I
3	Supply Voltage V_B
4	Input Amplifier III
5	Output Amplifier II
6	Pin for Adjusting the Separation Threshold
7	Negative Pulse Output
8	Positive Pulse Output
9	Output's Ground, 0
10	Test Pin, leave vacant
11	Input Amplifier II
12	Output Amplifier I
13	Ground, 0, of Amplifier II
14	Input

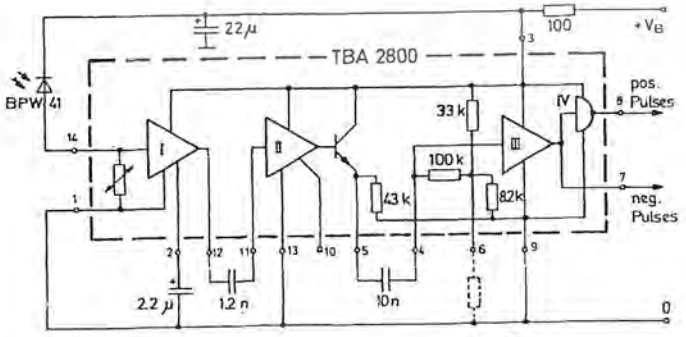


Fig. 1: TBA 2800 block diagram and application circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

All voltages are referred to GND (pins 1, 9, and 13).

Absolute Maximum Ratings

	Symbol	Value	Unit
Supply Voltage	V_3	6	V
Ambient Operating Temperature Range	T_A	0 to +65	°C
Storage Temperature Range	T_S	−30 to +125	°C

Recommended Operating Conditions

	Symbol	Min.	Typ.	Max.	Unit
Supply Voltage	V_3	4.5	5	5.5	V

Characteristics at $V_3 = 5$ V, $T_A = 25$ °C, photo diode BPW 41, in the circuit Fig. 1

	Symbol	Min.	Typ.	Max.	Unit
Current Consumption	I_2	–	1	2	mA
Gain between pins 14 and 7	G	70	–	–	dB
Output Resistance pins 7 and 8, formed by the pull-up resistor of an NPN transistor	R_O	–	20	–	kΩ
Output Low Voltage pins 7 and 8 at $I_{OL} = 1.6$ mA	V_{OL}	–	0.4	0.8	V
IR Transmission Range using the SAA 1250 IR Transmitter IC and a Transmission Current Amplitude of 1.5 A, and					
one Transmitter Diode CQY 99	L1	–	12	–	m
two Transmitter Diodes CQY 99	L2	–	19	–	m
three Transmitter Diodes CQY 99	L3	–	26	–	m

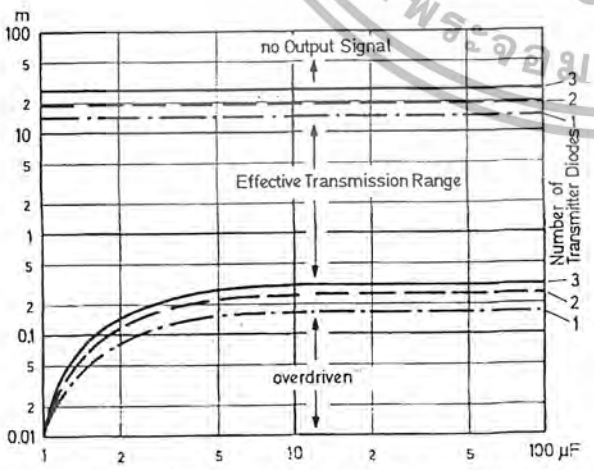


Fig. 3: Transmission range depending on capacitor at pin 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Sheet History

– page 3: Absolute Maximum Ratings
Ambient Operating Temperature Range (T_A) changed
from (–20 to +65°C)
to (0 to +65 °C)

MICRONAS INTERMETALL GmbH
Hans-Bunte-Strasse 19
D-79108 Freiburg (Germany)
P.O. Box 840
D-79008 Freiburg (Germany)
Tel. +49-761-517-0
Fax +49-761-517-2174
E-mail: docservice@intermetall.de
Internet: http://www.intermetall.de

Printed in Germany
by Simon Druck GmbH & Co., Freiburg (4/94)
Order No. 6251-203-6DS

All information and data contained in this data sheet are without any commitment, are not to be considered as an offer for conclusion of a contract nor shall they be construed as to create any liability. Any new issue of this data sheet invalidates previous issues. Product availability and delivery dates are exclusively subject to our respective order confirmation form; the same applies to orders based on development samples delivered. By this publication, MICRONAS INTERMETALL GmbH does not assume responsibility for patent infringements or other rights of third parties which may result from its use.
Reprinting is generally permitted, indicating the source. However, our prior consent must be obtained in all cases.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FAIRCHILD
SEMICONDUCTOR®

www.fairchildsemi.com

LM555/NE555/SA555

Single Timer

Features

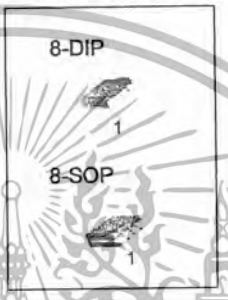
- High Current Drive Capability (200mA)
- Adjustable Duty Cycle
- Temperature Stability of 0.005%/°C
- Timing From μSec to Hours
- Turn off Time Less Than 2μSec

Applications

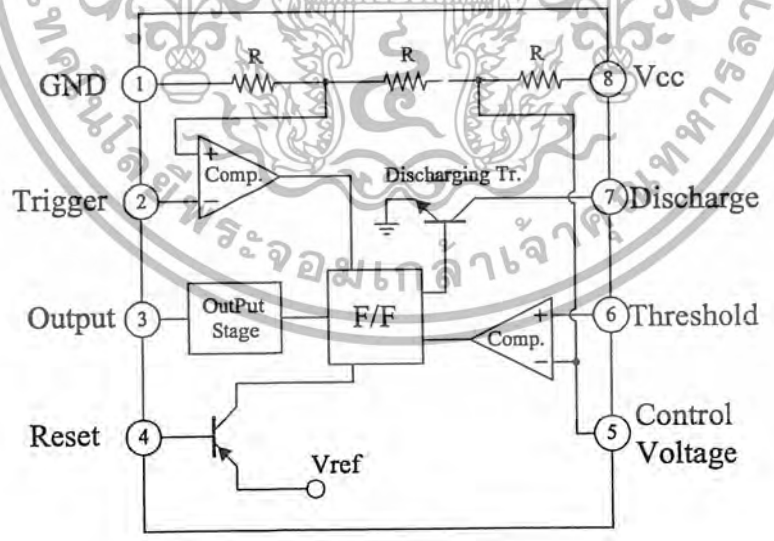
- Precision Timing
- Pulse Generation
- Time Delay Generation
- Sequential Timing

Description

The LM555/NE555/SA555 is a highly stable controller capable of producing accurate timing pulses. With a monostable operation, the time delay is controlled by one external resistor and one capacitor. With an astable operation, the frequency and duty cycle are accurately controlled by two external resistors and one capacitor.



Internal Block Diagram



Rev. 1.0.3

©2002 Fairchild Semiconductor Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Value	Unit
Supply Voltage	V_{CC}	16	V
Lead Temperature (Soldering 10sec)	T_{LEAD}	300	°C
Power Dissipation	P_D	600	mW
Operating Temperature Range LM555/NE555 SA555	T_{OPR}	 0 ~ +70 -40 ~ +85	°C
Storage Temperature Range	T_{STG}	-65 ~ +150	°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics

(T_A = 25°C, V_{CC} = 5 ~ 15V, unless otherwise specified)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Supply Voltage	V_{CC}	-	4.5	-	16	V
Supply Current (Low Stable) (Note1)	I_{CC}	V_{CC} = 5V, R_L = ∞	-	3	6	mA
		V_{CC} = 15V, R_L = ∞	-	7.5	15	mA
Timing Error (Monostable) Initial Accuracy (Note2) Drift with Temperature (Note4) Drift with Supply Voltage (Note4)	ACCUR $\Delta t/\Delta T$ $\Delta t/\Delta V_{CC}$	R_A = 1kΩ to100kΩ C = 0.1μF	-	1.0 50 0.1	3.0 0.5	% ppm/°C %/V
Timing Error (Astable) Intial Accuracy (Note2) Drift with Temperature (Note4) Drift with Supply Voltage (Note4)	ACCUR $\Delta t/\Delta T$ $\Delta t/\Delta V_{CC}$	R_A = 1kΩ to 100kΩ C = 0.1μF	-	2.25 150 0.3	-	% ppm/°C %/V
Control Voltage	V_C	V_{CC} = 15V	9.0	10.0	11.0	V
		V_{CC} = 5V	2.6	3.33	4.0	V
Threshold Voltage	V_{TH}	V_{CC} = 15V	-	10.0	-	V
		V_{CC} = 5V	-	3.33	-	V
Threshold Current (Note3)	I_{TH}	-	-	0.1	0.25	μA
Trigger Voltage	V_{TR}	V_{CC} = 5V	1.1	1.67	2.2	V
		V_{CC} = 15V	4.5	5	5.6	V
Trigger Current	I_{TR}	V_{TR} = 0V		0.01	2.0	μA
Reset Voltage	V_{RST}	-	0.4	0.7	1.0	V
Reset Current	I_{RST}	-		0.1	0.4	mA
Low Output Voltage	V_{OL}	V_{CC} = 15V I_{SINK} = 10mA I_{SINK} = 50mA	-	0.06 0.3	0.25 0.75	V V
		V_{CC} = 5V I_{SINK} = 5mA	-	0.05	0.35	V
High Output Voltage	V_{OH}	V_{CC} = 15V I_{SOURCE} = 200mA I_{SOURCE} = 100mA	 12.75	12.5 13.3	-	V V
		V_{CC} = 5V I_{SOURCE} = 100mA	2.75	3.3	-	V
Rise Time of Output (Note4)	t_R	-	-	100	-	ns
Fall Time of Output (Note4)	t_F	-	-	100	-	ns
Discharge Leakage Current	I_{LKG}	-	-	20	100	nA

Notes:

1. When the output is high, the supply current is typically 1mA less than at V_{CC} = 5V.
2. Tested at V_{CC} = 5.0V and V_{CC} = 15V.
3. This will determine the maximum value of $R_A + R_B$ for 15V operation, the max. total R = 20MΩ, and for 5V operation, the max. total R = 6.7MΩ.
4. These parameters, although guaranteed, are not 100% tested in production.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Application Information

Table 1 below is the basic operating table of 555 timer:

Table 1. Basic Operating Table

Threshold Voltage (Vth)(PIN 6)	Trigger Voltage (Vtr)(PIN 2)	Reset(PIN 4)	Output(PIN 3)	Discharging Tr. (PIN 7)
Don't care	Don't care	Low	Low	ON
Vth > 2Vcc / 3	Vth > 2Vcc / 3	High	Low	ON
Vcc / 3 < Vth < 2 Vcc / 3	Vcc / 3 < Vth < 2 Vcc / 3	High	-	-
Vth < Vcc / 3	Vth < Vcc / 3	High	High	OFF

When the low signal input is applied to the reset terminal, the timer output remains low regardless of the threshold voltage or the trigger voltage. Only when the high signal is applied to the reset terminal, the timer's output changes according to threshold voltage and trigger voltage.

When the threshold voltage exceeds 2/3 of the supply voltage while the timer output is high, the timer's internal discharge Tr. turns on, lowering the threshold voltage to below 1/3 of the supply voltage. During this time, the timer output is maintained low. Later, if a low signal is applied to the trigger voltage so that it becomes 1/3 of the supply voltage, the timer's internal discharge Tr. turns off, increasing the threshold voltage and driving the timer output again at high.

1. Monostable Operation

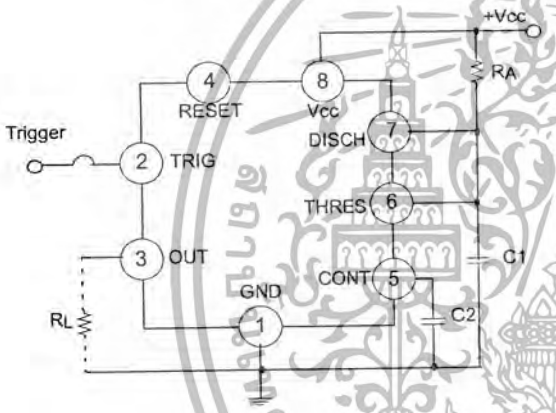


Figure 1. Monoatable Circuit

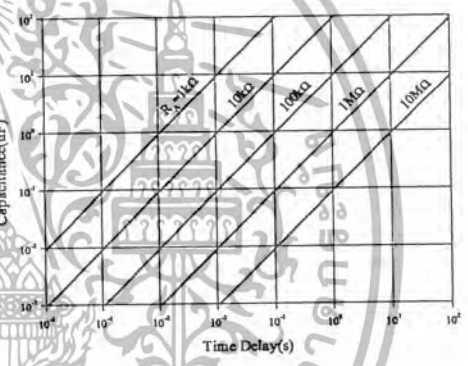


Figure 2. Resistance and Capacitance vs. Time delay(td)

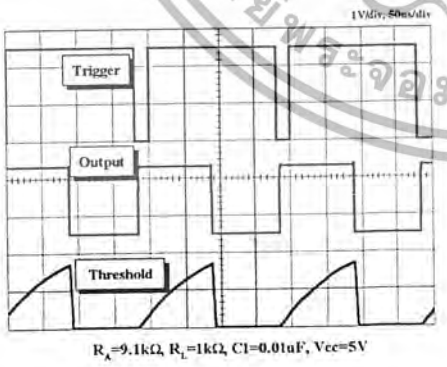


Figure 3. Waveforms of Monostable Operation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 1 illustrates a monostable circuit. In this mode, the timer generates a fixed pulse whenever the trigger voltage falls below Vcc/3. When the trigger pulse voltage applied to the #2 pin falls below Vcc/3 while the timer output is low, the timer's internal flip-flop turns the discharging Tr. off and causes the timer output to become high by charging the external capacitor C1 and setting the flip-flop output at the same time.

The voltage across the external capacitor C1, V_{C1} increases exponentially with the time constant $t=R_A*C$ and reaches 2Vcc/3 at $td=1.1R_A*C$. Hence, capacitor C1 is charged through resistor R_A. The greater the time constant R_AC, the longer it takes for the V_{C1} to reach 2Vcc/3. In other words, the time constant R_AC controls the output pulse width.

When the applied voltage to the capacitor C1 reaches 2Vcc/3, the comparator on the trigger terminal resets the flip-flop, turning the discharging Tr. on. At this time, C1 begins to discharge and the timer output converts to low.

In this way, the timer operating in the monostable repeats the above process. Figure 2 shows the time constant relationship based on R_A and C. Figure 3 shows the general waveforms during the monostable operation.

It must be noted that, for a normal operation, the trigger pulse voltage needs to maintain a minimum of Vcc/3 before the timer output turns low. That is, although the output remains unaffected even if a different trigger pulse is applied while the output is high, it may be affected and the waveform does not operate properly if the trigger pulse voltage at the end of the output pulse remains at below Vcc/3. Figure 4 shows such a timer output abnormality.

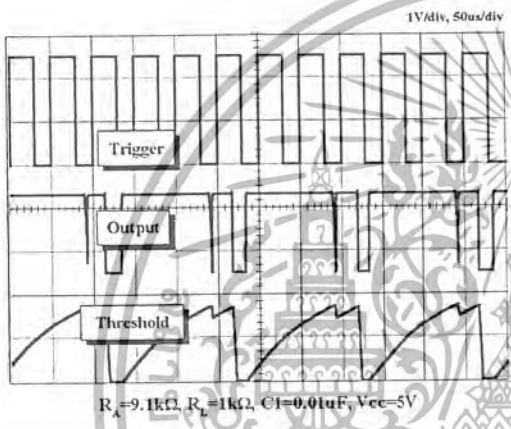


Figure 4. Waveforms of Monostable Operation (abnormal)

2. Astable Operation

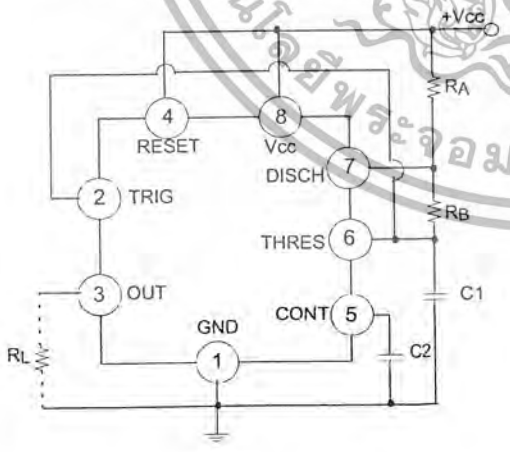


Figure 5. Astable Circuit

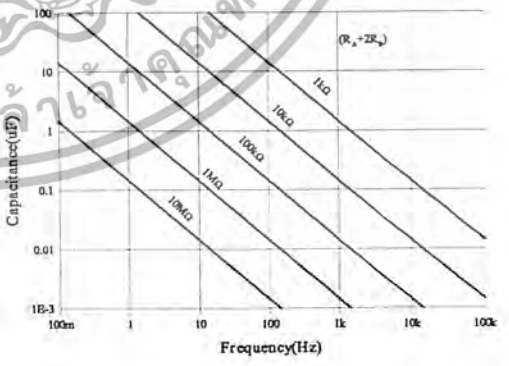
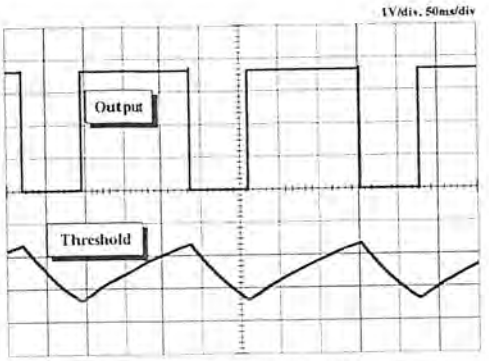


Figure 6. Capacitance and Resistance vs. Frequency

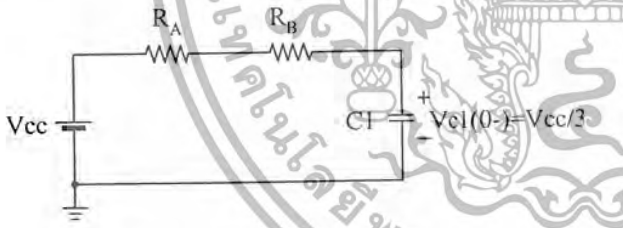
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



R_A=1kΩ, R_B=1kΩ, R_L=1kW, C1=1uF, Vcc=5V

Figure 7. Waveforms of Astable Operation

An astable timer operation is achieved by adding resistor R_B to Figure 1 and configuring as shown on Figure 5. In the astable operation, the trigger terminal and the threshold terminal are connected so that a self-trigger is formed, operating as a multi vibrator. When the timer output is high, its internal discharging Tr. turns off and the V_{C1} increases by exponential function with the time constant $(R_A+R_B)*C$.
When the V_{C1}, or the threshold voltage, reaches 2Vcc/3, the comparator output on the trigger terminal becomes high. resetting the F/F and causing the timer output to become low. This in turn turns on the discharging Tr. and the C1 discharges through the discharging channel formed by R_B and the discharging Tr. When the V_{C1} falls below Vcc/3, the comparator output on the trigger terminal becomes high and the timer output becomes high again. The discharging Tr. turns off and the V_{C1} rises again.
In the above process, the section where the timer output is high is the time it takes for the V_{C1} to rise from Vcc/3 to 2Vcc/3, and the section where the timer output is low is the time it takes for the V_{C1} to drop from 2Vcc/3 to Vcc/3. When timer output is high, the equivalent circuit for charging capacitor C1 is as follows:



$$C_1\frac{dv_{c1}}{dt} = \frac{V_{cc} - V(0-)}{R_A + R_B} \quad (1)$$

$$V_{C1}(0+) = V_{CC}/3 \quad (2)$$

$$V_{C1}(t) = V_{CC}\left(1 - \frac{2}{3}e^{-\left(\frac{t}{(R_A+R_B)C1}\right)}\right) \quad (3)$$

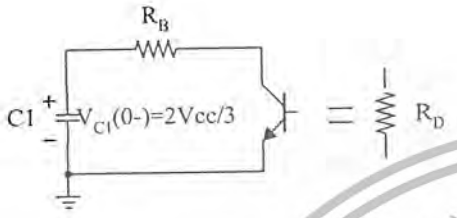
Since the duration of the timer output high state(t_H) is the amount of time it takes for the $V_{C1}(t)$ to reach 2Vcc/3,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$V_{C1}(t) = \frac{2}{3}V_{CC} = V_{CC}\left(1 - \frac{2}{3}e^{-\left(-\frac{t_H}{(R_A+R_B)C1}\right)}\right) \quad (4)$$

$$t_H = C_1(R_A+R_B)\ln 2 = 0.693(R_A+R_B)C_1 \quad (5)$$

The equivalent circuit for discharging capacitor C1, when timer output is low is, as follows:



$$C_1\frac{dv_{C1}}{dt} + \frac{1}{R_A+R_B}V_{C1} = 0 \quad (6)$$

$$V_{C1}(t) = \frac{2}{3}V_{CC}e^{-\frac{t}{(R_A+R_D)C1}} \quad (7)$$

Since the duration of the timer output low state(t_L) is the amount of time it takes for the $V_{C1}(t)$ to reach Vcc/3,

$$\frac{1}{3}V_{CC} = \frac{2}{3}V_{CC}e^{-\frac{t_L}{(R_A+R_D)C1}} \quad (8)$$

$$t_L = C_1(R_B+R_D)\ln 2 = 0.693(R_B+R_D)C_1 \quad (9)$$

Since R_D is normally $R_B >> R_D$ although related to the size of discharging Tr.,

$t_L = 0.693R_BC1$ (10)

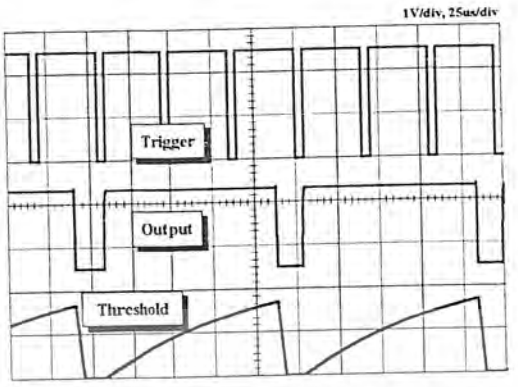
Consequently, if the timer operates in astable, the period is the same with
'$T=t_H+t_L=0.693(R_A+R_B)C_1+0.693R_BC_1=0.693(R_A+2R_B)C_1$' because the period is the sum of the charge time and discharge time. And since frequency is the reciprocal of the period, the following applies.

$$\text{frequency,} \quad f = \frac{1}{T} = \frac{1.44}{(R_A+2R_B)C_1} \quad (11)$$

3. Frequency divider

By adjusting the length of the timing cycle, the basic circuit of Figure 1 can be made to operate as a frequency divider. Figure 8. illustrates a divide-by-three circuit that makes use of the fact that retriggering cannot occur during the timing cycle.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



R_A=9.1kΩ, R_L=1kΩ, C1=0.01uF, Vcc=5V

Figure 8. Waveforms of Frequency Divider Operation

4. Pulse Width Modulation

The timer output waveform may be changed by modulating the control voltage applied to the timer's pin 5 and changing the reference of the timer's internal comparators. Figure 9 illustrates the pulse width modulation circuit.
When the continuous trigger pulse train is applied in the monostable mode, the timer output width is modulated according to the signal applied to the control terminal. Sine wave as well as other waveforms may be applied as a signal to the control terminal. Figure 10 shows the example of pulse width modulation waveform.

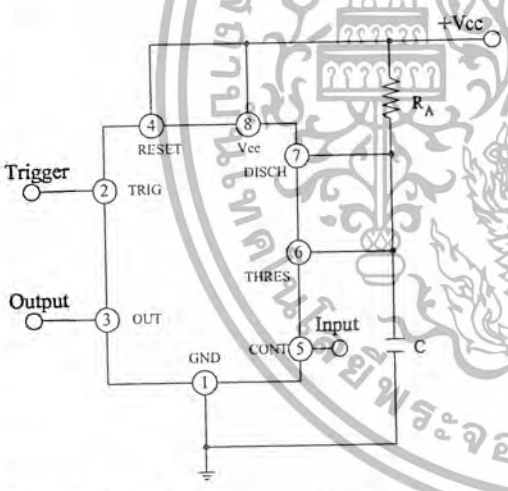
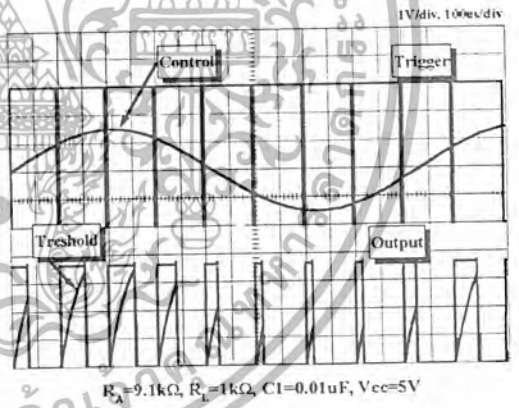


Figure 9. Circuit for Pulse Width Modulation



R_A=9.1kΩ, R_L=1kΩ, C1=0.01uF, Vcc=5V

Figure 10. Waveforms of Pulse Width Modulation

5. Pulse Position Modulation

If the modulating signal is applied to the control terminal while the timer is connected for the astable operation as in Figure 11, the timer becomes a pulse position modulator.
In the pulse position modulator, the reference of the timer's internal comparators is modulated which in turn modulates the timer output according to the modulation signal applied to the control terminal.
Figure 12 illustrates a sine wave for modulation signal and the resulting output pulse position modulation : however, any wave shape could be used.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

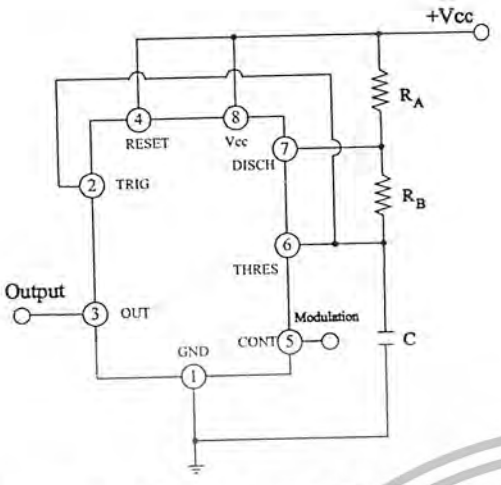


Figure 11. Circuit for Pulse Position Modulation

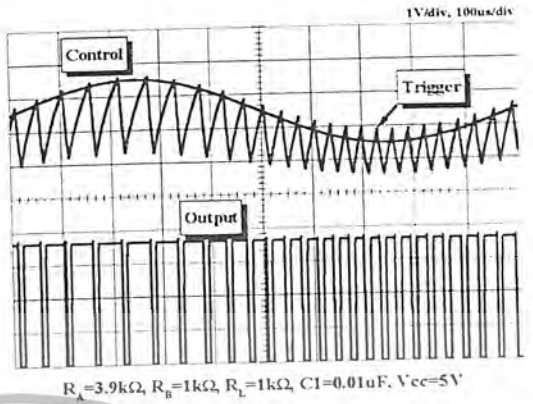


Figure 12. Waveforms of pulse position modulation

6. Linear Ramp

When the pull-up resistor RA in the monostable circuit shown in Figure 1 is replaced with constant current source, the V_{C1} increases linearly, generating a linear ramp. Figure 13 shows the linear ramp generating circuit and Figure 14 illustrates the generated linear ramp waveforms.

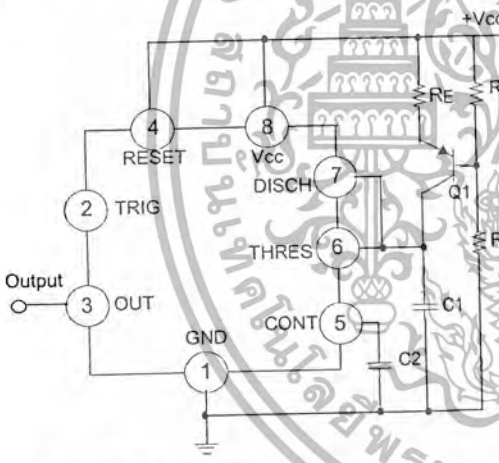


Figure 13. Circuit for Linear Ramp

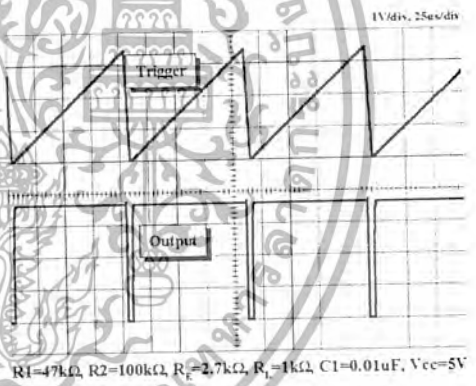


Figure 14. Waveforms of Linear Ramp

In Figure 13, current source is created by PNP transistor Q1 and resistor R1, R2, and R_E.

$$I_C = \frac{V_{CC} - V_E}{R_E} \qquad (12)$$

Here, V_E is

$$V_E = V_{BE} + \frac{R_2}{R_1 + R_2} V_{CC} \qquad (13)$$

For example, if Vcc=15V, R_E=20kΩ, R1=5kW, R2=10kΩ, and V_{BE}=0.7V,
V_E=0.7V+10V=10.7V
Ic=(15-10.7)/20k=0.215mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

When the trigger starts in a timer configured as shown in Figure 13, the current flowing through capacitor C1 becomes a constant current generated by PNP transistor and resistors.
Hence, the V_C is a linear ramp function as shown in Figure 14. The gradient S of the linear ramp function is defined as follows:

$$S = \frac{V_{p-p}}{T} \qquad (14)$$

Here the Vp-p is the peak-to-peak voltage.
If the electric charge amount accumulated in the capacitor is divided by the capacitance, the V_C comes out as follows:

$$V=Q/C \qquad (15)$$

The above equation divided on both sides by T gives us

$$\frac{V}{T} = \frac{Q/T}{C} \qquad (16)$$

and may be simplified into the following equation.

$$S=I/C \qquad (17)$$

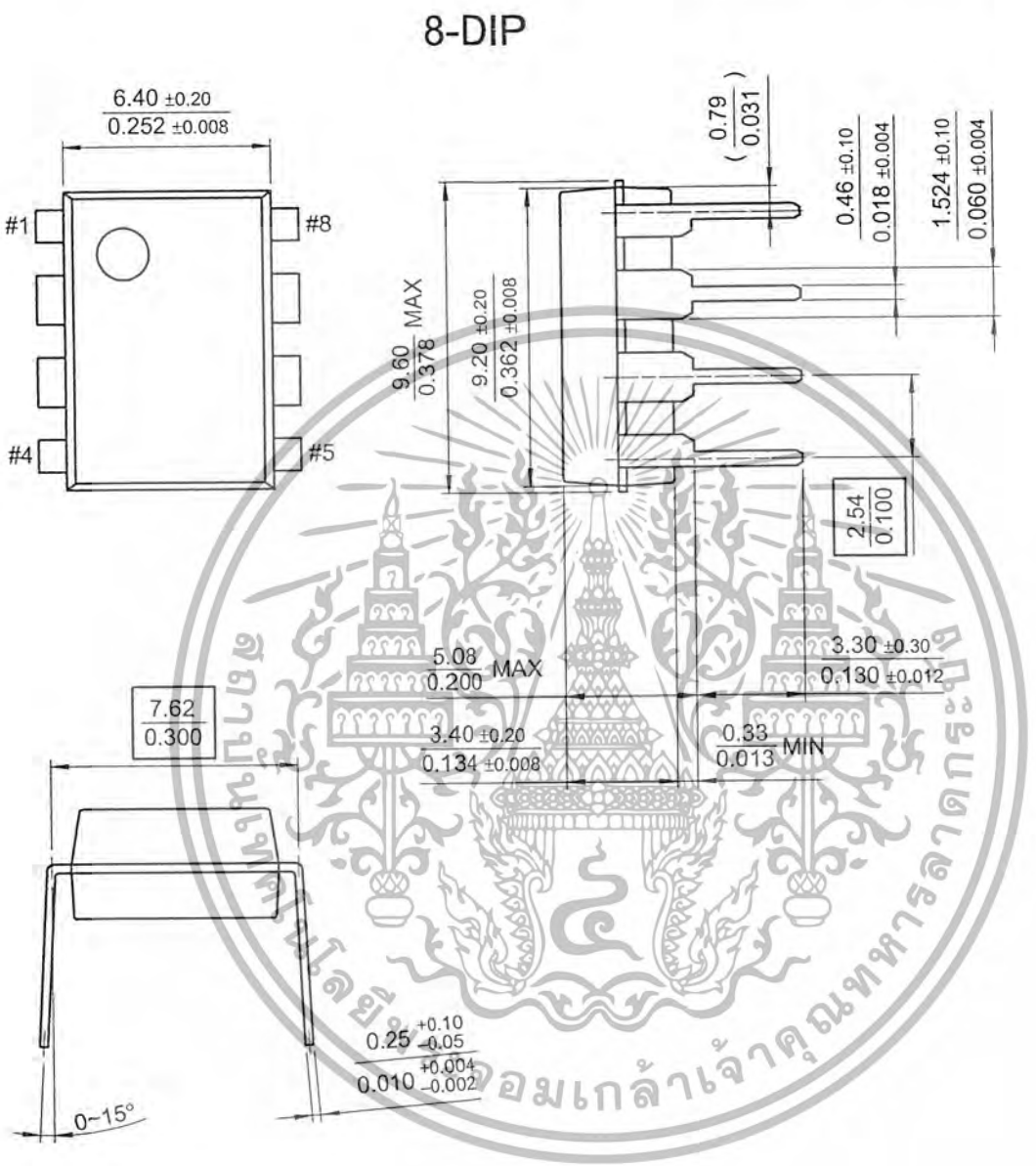
In other words, the gradient of the linear ramp function appearing across the capacitor can be obtained by using the constant current flowing through the capacitor.
If the constant current flow through the capacitor is 0.215mA and the capacitance is 0.02μF, the gradient of the ramp function at both ends of the capacitor is S = 0.215m/0.022μ = 9.77V/ms.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mechanical Dimensions

Package

Dimensions in millimeters



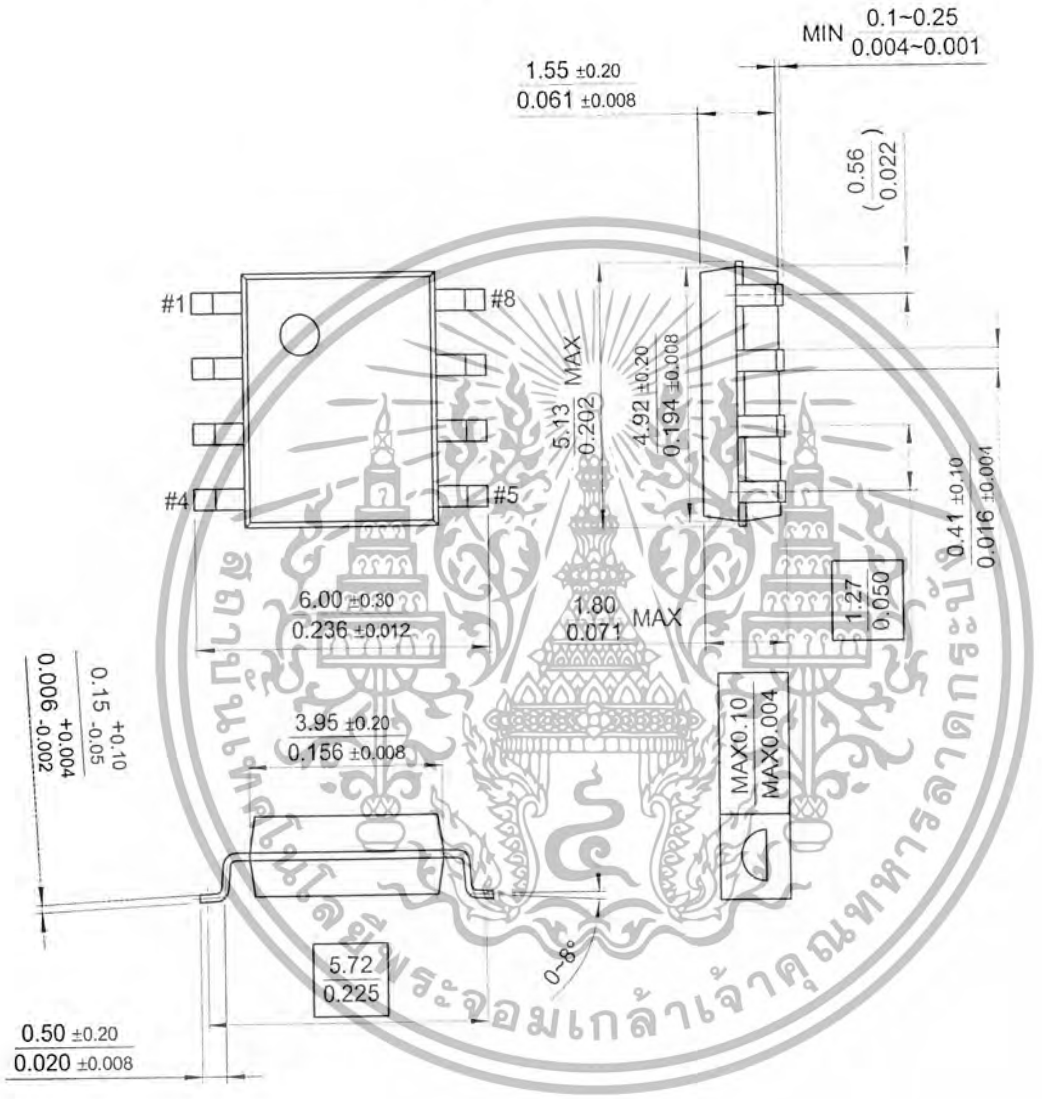
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mechanical Dimensions (Continued)

Package

Dimensions in millimeters

8-SOP



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และความร่วมมือจากหลายๆฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์สุรพล บุญจันทร์ อาจารย์ที่ปรึกษาที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา รวมถึงเพื่อนทุกๆคนที่ให้คำชี้แนะในปัญหาที่เกี่ยวกับโครงงาน และต้องขอขอบพระคุณบุคคลที่สำคัญที่สุดที่ทำให้ผู้จัดทำมีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ ผู้จัดทำขอกราบขอบพระคุณมา ณ ที่นี้ด้วย

และสุดท้ายนี้หวังว่าปริญญานิพนธ์ฉบับนี้คงจะเป็นประโยชน์และเป็นแนวทางแก่รุ่นน้องและผู้ที่ต้องการศึกษาต่อไป

นายพีรพัฒน์ อริยะกวี

นายเวชยันต์ พันรูปท้าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. วรพจน์ กรแก้ววัฒนกุล "เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลซ" บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด, กรุงเทพมหานคร, 2540.
2. รศ.สมยศ จุณณะปิยะ "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51" สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพมหานคร, 2543.
3. พัฒนพงศ์ พันธยะติวงศ์ " การเขียนโปรแกรมคอมพิวเตอร์ด้วย Visual Basic " บริษัท ซีเอ็ดยูเคชั่น จำกัด, กรุงเทพมหานคร, 2545.
4. สัจจะ จรัสรุ่งรวีวร " คู่มือการเขียนโปรแกรมและใช้งาน Visual Basic 6.0 " สำนักพิมพ์ อินโฟเพรส, กรุงเทพมหานคร, 2544.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้