สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การสื่อสารผ่านเครือข่ายอินเตอร์เน็ตควบคุมอุปกรณ์ไฟฟ้า

Control of electric appliances through the internet

นาย จิรสิทธิ์ ฤทธิ์มนตรี

นาย สิริชัย ธรรมารักษ์วัฒนะ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน 42235
วัน, เดือน, ป 15 พ.ค. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

สาขาวิชาเทคโนโลยีโทรคมนาคม ภาควิชาเทคนิคอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTROL OF ELECTRIC APPLIANCES THROUGH THE INTERNET

Mr. JIRASIT RITMONTRI

Mr. SIRICHAI TUMMARUCKWATTANA

A THESIS SUBMITTED IN PARTIAL FULFILMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF THE TECHNOLOGY TELECOMMUNICATIONS

FACULTY OF ENGINEERING

KING MONKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ การสื่อสารผ่านเครือข่ายอินเตอร์เน็ตควบคุมอุปกรณ์ไฟฟ้า
Control of electric appliances through the internet

นักศึกษา นายจิรสิทธิ์ ฤทธิ์มนตรี เลขที่ประจำตัว 41013329
นายสิริชัย ธรรมรักษ์วัฒนะ เลขที่ประจำตัว 41013354

อาจารย์ที่ปรึกษา อาจารย์มยุรี เลิศเวชกุล
อาจารย์พิทักษ์ ธรรมวาริน

ภาควิชา เทคนิคอุตสาหกรรม
ปีการศึกษา 2543

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

.....

คณะกรรมการสอบปริญญานิพนธ์

..... อาจารย์ที่ปรึกษา
(อาจารย์มยุรี เลิศเวชกุล)

..... อาจารย์ที่ปรึกษา
(อาจารย์พิทักษ์ ธรรมวาริน)

..... กรรมการ
( )

..... กรรมการ
( )

..... กรรมการ
( )

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ การสื่อสารผ่านเครือข่ายอินเตอร์เน็ตควบคุมอุปกรณ์ไฟฟ้า
Control of electric appliances through the internet

นักศึกษา นายจิรสิทธิ์ ฤทธิ์มนตรี เลขที่ประจำตัว 41013329
นายสิริชัย ธรรมรักษ์วัฒนะ เลขที่ประจำตัว 41013354

อาจารย์ที่ปรึกษา อาจารย์มยุรี เลิศเวชกุล
อาจารย์พิทักษ์ ธรรมวาริน

ภาควิชา เทคนิคอุตสาหกรรม

ปีการศึกษา 2543

บทคัดย่อ

การควบคุมระยะไกลซึ่งทำหน้าที่เพื่อควบคุมอุปกรณ์เครื่องใช้ไฟฟ้า โครงงานนี้ได้ทำการควบคุมระยะไกล โดยใช้วิธีสั่งการควบคุมผ่านอินเตอร์เน็ตไปยังคอมพิวเตอร์และคอนโทรลเลอร์ ทำให้เราสามารถควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าได้ โดยในส่วนของคอนโทรลเลอร์ทำการควบคุมโดยใช้ชิพคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89C2051 เป็นไมโครคอนโทรลเลอร์ 8 บิท ขนาด 20 ขา มีหน่วยความจำภายในขนาด 2 Kbyte ทำการรับข้อมูลจากพอร์ต RS-232 โดยโปรแกรมส่งคำสั่งเปิดหรือปิด มีความเร็ว 9600 บิทต่อวินาที ไปยังสถานที่ๆจะทำการควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title Control of electric appliances through the internet

Student Mr. Jirasit Ritmontri ID 41013329
Mr. Sirichai Tammaruckwattana ID 41013354

Advisor Ms. Mayuree Lertwachakul
Mr. Pitak Tamwarin

Academic Year 2000

ABSTRACT

This project is involved in remote control of electric devices from the internet to computer and controller. The controller is commanded by chip controller family MCS-51 number AT89C2051. It's an eight-bit and 20 pin micro-controller with a 2 Kbyte built-in memory. It receives data from RS-232 port as the programme orders to open or shut down at a speed of 9600 bit per second to the location of electric devices that are to be controlled.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี ขอขอบพระคุณ อาจารย์ที่ปรึกษา ที่ให้คำแนะนำและความช่วยเหลือที่ดีเสมอมาตลอดจนอีกทั้งยังได้ชี้แนะแนวทางในการแก้ไขปัญหาต่างๆมาโดยตลอด

ขอขอบพระคุณพี่วิสัน, พี่สำฤทธิ์, พี่เต๋า และพี่ๆ ที่สำนักวิจัยฯ ที่คอยให้คำปรึกษาและชี้แนะแนวทางในการแก้ไขปัญหาต่างๆ ด้วยดีมาตลอด

ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ที่คอยห่วงใยและให้การสนับสนุนในการศึกษา รวมทั้งขอขอบคุณญาติสนิทและพี่ๆทุกคนที่เป็นกำลังใจพร้อมทั้งให้ความช่วยเหลือในด้านต่างๆมาโดยตลอด

สุดท้ายขอขอบคุณเพื่อนๆ ห้อง 1U ทุกคน ที่คอยช่วยเหลือกันมาตลอดและนายปริญญา นายวิเชียรและนายวิทยากรณ์ ที่ให้คำปรึกษาในการออกแบบวงจร

นาย จิรสิทธิ์ ฤทธิ์มนตรี
นาย สิริชัย ธรรมารักษ์วัฒนะ

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์	1
1.2 แนวความคิดและที่มา	1
1.3 สถาปัตยกรรมของระบบ	1
1.4 ขอบเขตของโครงงาน	2
บทที่ 2 JAVA Servlet	3
2.1 จุดเด่นของ Java Servlet	4
2.2 สถาปัตยกรรมของ Java Servlet	4
2.3 การใช้ Java Servlets กับการพัฒนาระบบงานบนเครือข่ายอินเตอร์เน็ต	5
บทที่ 3 การสื่อสารข้อมูล	13
3.1 การสื่อสารแบบขนาน	13
3.2 การสื่อสารแบบอนุกรม	14
3.3 การอินเตอร์เฟส	17
3.4 การรับส่งข้อมูล 2 ทิศทาง	19
3.5 ลักษณะการเชื่อมต่อและหน้าที่การทำงานที่สำคัญ	23
บทที่ 4 วงจรรับข้อมูลจากพอร์ตอนุกรม และ แหล่งจ่ายไฟ	25
4.1 คุณสมบัติของ AT89C2051	26
4.2 แหล่งจ่ายไฟ	28
4.3 วงจรรีเลย์	28
4.4 แหล่งจ่ายไฟสำหรับวงจรรีเลย์	31
บทที่ 5 โปรแกรมควบคุม	32
5.1 โปรแกรมเขียนข้อมูลขนาด 1 ไบต์ บนพอร์ตอนุกรม	33
5.2 โปรแกรมบนเซอร์ฟเวอร์	34
5.3 โปรแกรม JAVA Servlet	34
5.4 รายชื่อไฟล์สำคัญ	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6 การติดตั้งและการใช้งาน 35
6.1 การใช้งานระบบควบคุมอุปกรณ์ไฟฟ้าผ่านอินเตอร์เน็ต 36
6.2 ผลการทดสอบ 38
บทที่ 7 สรุปผลการทดลองและข้อเสนอแนะ 39
7.1 สรุปผลการทดลอง 39
7.2 ข้อเสนอแนะ 39
บรรณานุกรม 40
ภาคผนวก ก. รายละเอียดของวงจร
ภาคผนวก ข. โปรแกรมการใช้งาน
ภาคผนวก ค. เอกสารประกอบการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3.1 ตารางเปรียบเทียบการสื่อสารข้อมูลแบบขนานกับอนุกรม	16
3.2 ตารางมาตราฐานของการใช้แรงดันไฟฟ้า RS-232	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
1.1	ระบบโดยรวมของโครงงาน	2
2.1	แสดงการทำงานของ Java Servlet	3
3.1	แสดงการสื่อสารแบบขนาน	14
3.2	แสดงการสื่อสารข้อมูลแบบอนุกรม	15
3.3	แสดงอุปกรณ์ DTE เบื้องต้น	18
3.4	แสดงอุปกรณ์ทั้ง DTE และ DCE	19
3.5	แสดงอุปกรณ์ DTE และ DCE ซึ่งเป็นคู่อุปกรณ์ที่ทำงานตรงกันข้ามกัน	19
3.6	แสดงการทำงานของอุปกรณ์ที่สามารถส่งและรับข้อมูลได้ทั้งสองทิศทาง	20
3.7	แสดงคำจำกัดความค่าตรรกะที่เอาต์พุทของ RS-232	21
3.8	แสดงคำจำกัดความค่าตรรกะที่อินพุทของ RS-232	22
3.9	แสดงการเชื่อมต่อทั้งแบบ 25 Pin และ แบบ 9 Pin	24
4.1	ส่วนประกอบของฮาร์ดแวร์	26
4.2	วงจรไมโครคอนโทรลเลอร์	27
4.3	แหล่งจ่ายไฟวงจรไมโครคอนโทรลเลอร์	28
4.4	วงจรรีเลย์	29
4.5	วงจร LED แสดงสถานะรีเลย์และ Supply	30
4.6	แหล่งจ่ายไฟวงจรรีเลย์	31
5.1	แสดงผลวงจรทดสอบรับข้อมูลผ่านพอร์ตอนุกรม	33
6.1	แสดงการส่งสัญญาณควบคุมอุปกรณ์ไฟฟ้า	35
6.2	แสดงหน้าจอหลักระบบควบคุมอุปกรณ์ไฟฟ้าผ่านอินเตอร์เน็ต	36
6.3	แสดงหน้าจอเมื่อใส่รหัสผ่านผิดพลาด	36
6.4	แสดงระบบควบคุมอุปกรณ์ไฟฟ้าผ่านอินเตอร์เน็ต	37
6.5	แสดงหน้าจอการเลือกควบคุมอุปกรณ์ไฟฟ้า	37
6.6	แสดงสถานะการควบคุมอุปกรณ์ไฟฟ้าแก่ผู้ใช้งาน	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 วัตถุประสงค์

1. ศึกษาการรับและส่งข้อมูลผ่านเครือข่ายอินเตอร์เน็ต
2. ศึกษาโปรแกรมควบคุมรีเลย์
3. ศึกษาการส่งข้อมูลผ่านพอร์ตอนุกรม
4. ศึกษาการทำงานของวงจรไมโครคอนโทรลเลอร์
5. ศึกษาการส่งข้อมูลสถานะของรีเลย์ เพื่อแจ้งแก่ผู้ใช้ซึ่งอยู่ห่างออกไป

1.2 แนวความคิดและที่มา

ในอดีต COMPUTER ทำงานโดยเก็บข้อมูลขนาดใหญ่ๆ อยู่ในตัวมันซึ่งเป็นการเก็บที่มีระเบียบแบบแผนค้นหาง่าย ต่อมา COMPUTER ก็ได้พัฒนา SOFTWARE และ HARDWARE ที่มีความสามารถขึ้นมาเรื่อยๆ ความสะดวกสบายนี้เอง COMPUTER จึงได้เข้ามามีบทบาทในชีวิตมากขึ้นและในปัจจุบันมีการใช้ COMPUTER ติดต่อสื่อสารทางอินเตอร์เน็ตซึ่งครอบคลุมได้ทั่วโลก คุณประโยชน์นี้เองจึงได้เป็นแนวทางในการประยุกต์ใช้อินเตอร์เน็ตควบคุมรีเลย์ แต่เดิมการควบคุมรีเลย์เปิด - ปิดไฟฟ้าอยู่ในขอบเขตพื้นที่จำกัด

โดยโครงงานนี้เราได้นำเครือข่ายอินเตอร์เน็ตมาใช้ประโยชน์ในการควบคุมรีเลย์ อีกทั้งยังเป็นแนวทางในการควบคุมที่ซับซ้อนยิ่งขึ้นได้ต่อไป

1.3 สถาปัตยกรรมของระบบ

1.3.1 จุดประสงค์ของโครงงานนี้ คือ การควบคุมอุปกรณ์ไฟฟ้าผ่านเครือข่ายอินเตอร์เน็ต

1.3.2 สิ่งที่ระบบต้องการ (system Requirement) ได้แก่

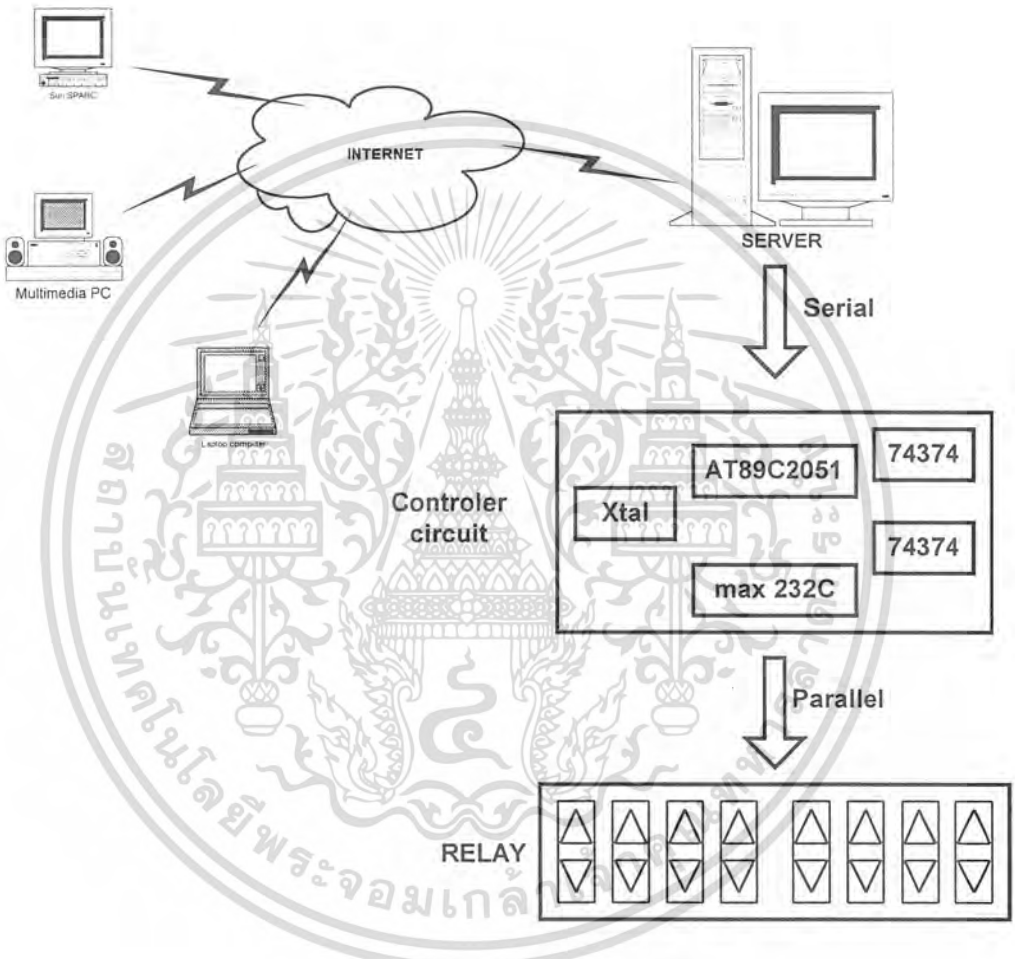
- เซอร์ฟเวอร์ (server) ที่มีพอร์ทอนุกรมแบบ RS-232 ชนิด DB-25 หรือ DB-9
- ระบบปฏิบัติการ windows 95 , windows 98 หรือ windows NT
- โปรแกรมภาษาจาวา (JAVA)
- โปรแกรมภาษาซี (turbo C)

ทางฝั่งผู้ใช้จะควบคุมอุปกรณ์ระยะไกลโดยใช้โปรแกรมเว็บเบราเซอร์ เช่น อินเตอร์เน็ตเอ็กซ์พลอยร์เลอร์ (microsoft Internet Explorer) หรือ เน็ทส์เคป เนวิเกเตอร์ (Netscape Navigator)

ซึ่งจะเขียนโปรแกรมรับส่งข้อมูลด้วยภาษา JAVA ไว้ที่เครื่องเซอร์ฟเวอร์แล้วสร้างไฟล์ HTML ไว้บนเครื่องเซอร์ฟเวอร์ ซึ่งมีเว็บเซอร์ฟเวอร์ (Sambar) เป็นตัวให้บริการ เมื่อผู้ใช้ที่อยู่ห่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกไปใช้เว็บเบราเซอร์ติดต่อเข้ามาทางอินเตอร์เน็ต (Internet) เซอร์ฟเวอร์จะโหลดไฟล์ HTML นี้ให้กับผู้ใช้ โดยผู้ใช้จะต้องใส่รหัสผู้ใช้ และรหัสผ่านก่อนที่จะส่งข้อมูลควบคุมอุปกรณ์ไฟฟ้า กลับมายังเซอร์ฟเวอร์ โปรแกรม Servlet จะทำการรับข้อมูลจากไฟล์ Html และส่งต่อให้ฮาร์ดแวร์ ซึ่งต่ออยู่กับพอร์ตอนุกรม com2 โดยจะมีการแสดงสถานะการควบคุมส่งกลับมายังผู้ใช้



รูปที่ 1.1 ระบบโดยรวมของโครงงาน

1.4 ขอบเขตของโครงงาน

ออกแบบวงจรรีเลย์ซึ่งทำหน้าที่รับข้อมูลจากคอมพิวเตอร์ ในการควบคุมรีเลย์และใช้ Java Servlet สำหรับควบคุมรีเลย์ไฟฟ้าจากระยะไกล โดยผ่านทางเครือข่ายอินเตอร์เน็ตและใช้โปรแกรมที่พัฒนาด้วยภาษา C เป็นตัวควบคุมไมโครคอนโทรเลอร์ และอุปกรณ์ปลายทาง

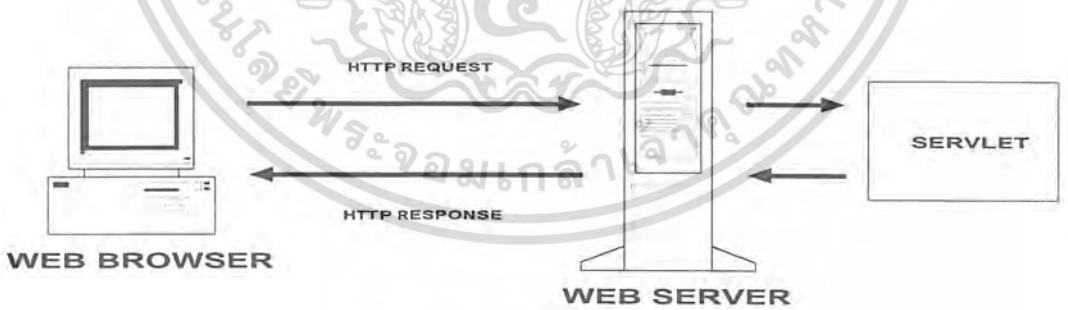
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

Java Servlet

Java servlet คือ โปรแกรมภาษาจาวาที่ถูกสร้างมาให้ทำงานได้บน HTTP Server โดยการทำงานของ Servlet มีลักษณะเป็น Server-side ซึ่งต่างจาก Java Appplet โดยทั่วไปที่ทำงานในลักษณะ Client-side โดย Servlet เป็น modules ที่ทำงานอยู่ภายในกระบวนการ request/response oriented server และ Servlet ยังสามารถตอบสนองในการส่งข้อมูลโดยให้อยู่ในรูปของภาษา HTML โดย Servlet ยังสามารถนำมาประยุกต์ใช้ในการพัฒนาฐานข้อมูล (database) ต่างๆ ได้อีกด้วย

ประโยชน์ของ Servlet นั้นช่วยให้เราเขียน Program ที่สั่งให้ HTTP Server ทำงานหรือโต้ตอบกับ Program ของ Client เหมือนที่ CGI (Common Gateway Interface) ทำได้ ซึ่ง Servlet ถูกนำมาใช้แทน CGI scripts ได้อย่างมีประสิทธิภาพและให้ความสะดวกได้อย่างมากคือสามารถเขียนได้ง่ายกว่าและทำการ RUN ได้เร็วกว่า CGI และ servlet ยังได้มีการจัดการในส่วนของการโปรแกรมทางด้าน server-side ไว้โดยเฉพาะ Servlet ได้ถูกพัฒนาโดยใช้ Java Servlet API ซึ่งเป็นส่วนเพิ่มเติมมาตราฐานของ Java



รูปที่ 2.1 แสดงการทำงานของ Servlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 จุดเด่นของ Servlet

1. Servlet สามารถที่จะฝังตัวทำงานอยู่บน server เมื่อทำการให้บริการ Client เสร็จสิ้นแล้วจะคอยรับ Request ใหม่ที่อาจจะมีมาจาก Client อีกโดยไม่ต้องถูกสร้างขึ้นมาใหม่จึงทำให้ Servlet ใช้ทรัพยากรของ server น้อยมาก
2. Servlet สามารถจัดการกับงานประเภท Multiple Connection ได้ดี โดยอาศัยความสามารถของ Thread ที่มีอยู่ในตัว Java
3. Servlet สามารถติดต่อกับ Applet บน Client ได้อย่างต่อเนื่องเป็นผลให้การรับส่งข้อมูลระหว่างกันดำเนินไปอย่างต่อเนื่อง
4. Servlet สามารถถูก Upload จาก Client เพื่อสั่งให้ไปทำงานบน server ใดๆ บน Network ได้ประโยชน์คือเราจะสามารถเขียน Program เพื่อค้นหาข้อมูลใน Web Host ใดๆ โดยไม่ต้องส่งข้อมูลมาประมวลที่เครื่องต้นทาง สามารถทำได้ที่ตัว Host นั้นๆ เลย เสร็จแล้วค่อยส่งผลรับกลับคืน จึงสามารถลดเวลาและ Traffic บนระบบ Network ลงอย่างมาก

2.2 สถาปัตยกรรมของ Servlet

ใจความสำคัญของ Servlet API คือ การสนับสนุนการ interface ของ Servlet ทั้งแบบทางตรงหรือทางอ้อม โดยประกอบไปด้วยคลาสที่สนับสนุนการ interface เช่น Http Servlet ซึ่ง Servlet interface ประกอบด้วย method ที่ใช้เป็นตัวจัดการ Servlet และติดต่อสื่อสารกับ client ผู้เขียน Servlet จะกำหนด method นี้ทั้งหมดหรือบางส่วนเมื่อต้องการพัฒนา Servlet เมื่อ Servlet ได้รับการร้องเรียกจาก client จะมีการรับ object อยู่ 2 object ซึ่งอันแรกคือ Servlet Request และอีกอันคือ Servlet Response โดยคลาส Servlet Request นั้นจะเป็นตัวติดต่อสื่อสารจาก client ไปยัง server ในขณะที่คลาส Servlet Response เป็นตัวติดต่อสื่อสารจาก Servlet กลับไปยัง client

Servlet Request interface อนุญาตให้ Servlet สามารถเข้าถึงข้อมูลได้ เช่น ชื่อของพารามิเตอร์ที่ผ่านเข้ามราโดย client , โปรโตคอล และชื่อ host ของ client ที่เป็นตัวส่ง request และของ server ที่เป็นตัวรับ request ด้วยรวมไปถึงการกำหนดให้ Servlet เข้าถึง input stream ได้ด้วย โดย Servlet Input Stream โดย Servlet ทำหน้าที่รับข้อมูลจาก client โดยผ่านทางโปรโตคอล เช่น HTTP , POST และ PUT method โดยคลาสย่อยของคลาส Servlet Request จะอนุญาตให้ Servlet สามารถรับข้อมูลที่เป็นลักษณะเฉพาะของโปรโตคอล กลับมาได้

Servlet Response interface ประกอบด้วย method สำหรับการตอบกลับไปยัง client โดยอนุญาตให้ Servlet กำหนดขนาด และ mime type ในการตอบกลับ และเป็นตัวจัดเตรียม output stream ด้วย โดยผู้เขียน Servlet สามารถใช้ Servlet Output Stream ในการส่งข้อมูลกลับได้ sub คลาส ของ Servlet Response ทำให้ Servlet มีโปรโตคอลที่แตกต่างกันได้หลายชนิด ตัวอย่างเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Http Servlet Response ประกอบด้วย method ที่อนุญาตให้ Servlet สามารถถ่ายเทข้อความจาก HTTP-specific header ได้

2.3 การใช้ Java Servlets กับการพัฒนาระบบงานบนเครือข่ายอินเตอร์เน็ต

Servlet API

Servlet API สามารถใช้ได้บน server พื้นฐานทั่วไปที่มีการสนับสนุน Servlet และถูกกำหนดโดยคลาสที่อยู่ในแพคเกจดังนี้ java.Servlet , java . Servlet.http และ java. Servlet.htm

Package : java.servlet

Java.Servlet เป็นแพคเกจที่กำหนดการทำงานของ Servlet จากสัดส่วนของ protocol-independent ในระดับนี้โปรโตคอลที่ถูกใช้ในการเข้าถึง Servlet ได้แก่ HTTP , โปรโตคอลของระบบไฟล์เครือข่าย หรือ โปรโตคอล ที่ใช้ในการขนส่งอื่น ๆ Servlet ถูกออกแบบให้ทำงานบน server-side โดยใช้คลาสในแพคเกจนี้ซึ่งประกอบไปด้วย interfaces class 4 คลาส และ implementation class 3 คลาส ดังนี้

1. *Servlet Interface Class* ประกอบไปด้วย method พื้นฐาน 4 method ดังนี้คือ
   - init(ServletStub) ถูกเรียกโดยอัตโนมัติเมื่อระบบต้องการหาค่าเริ่มต้นของ Servlet โดยกำหนดตัวแปรเริ่มต้นของ Servlet ในไฟล์ servlet.properties ได้โดยการเรียก getniParameter() หรือ getlnitParameters()
   - destroy() ถูกเรียกโดยระบบเพื่อคืนทรัพยากรให้ระบบเมื่อ Servlet ถูกยกเลิกการใช้งาน
   - service(ServletRequest , Servlet Response) ถูกเรียกมาใช้ในกระบวนการรับ request ที่เข้ามาและสร้าง response ตอบกลับไป
   - getServletInfo() ถูกเรียกเพื่อรับข้อมูล (information) ที่เป็น String ซึ่งบอกลักษณะของ Servlet
2. *Servlet Context Interface Class* เป็นการกำหนด method เพื่อให้ Servlet สามารถทำงานแบบสลับฉาก (interact) ได้โดย method ต่าง ๆ มีดังนี้
   - getMime Type(String filename) ส่งคืน MIME type ของแต่ละไฟล์โดยใส่ชื่อของไฟล์ (filename) ลงใน mime.properties file
   - getRealPath (String path) การประยุกต์ alias rules ใน alias.properties file เพื่อเป็นการกำหนด virtual path และตอบสนองด้วยการส่ง real path หากการส่งข้อมูลเกิดการผิดพลาดค่าที่ส่งกลับคืน null
   - getServerInfo() ส่งคืนค่าที่เป็น string ที่บ่งบอกชื่อ , version และฐานของ server software

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- getServlet (String) ส่งคืน Servlet ที่ตรงกับชื่อที่เรียกมา
- getServlets() ส่งคืน Servlets ทั้งหมดที่ถูกโหลดใน server โดยรวมไปถึง Servlet ที่ถูกเรียกด้วย
- log (Servlet , String) เขียนข้อความลงใน Servlet log ของ Servlet ที่ถูกกำหนด

3. *Servlet Request Interface Class* ประกอบไปด้วย method ที่ใช้สำหรับค้นหาข้อมูลที่เกียวกับ Servlet request ซึ่งเป็นอิสระจากโปรโตคอลที่ใช้อยู่จริงในการส่งผ่านข้อมูลระหว่าง client และ server โดย request จะถูกนำพาโดย IP พื้นฐานของโปรโตคอลเช่น HTTP ซึ่งการส่งผ่านข้อมูลของ ServletRequest โดยอาศัยโปรโตคอลเช่น HTTP นั้นจะต้องมี method ที่มีความสัมพันธ์กับโปโตคอลดังนี้

- getRemoteAddr() ส่งคืนค่า IP address ของ client ที่ส่ง request
- getRemoteHost() ส่งคืน Host name ของตัวแทนที่ส่ง request
- getServerName() ส่งคืน Host name ของ server ที่รับ request
- getServerPort() ส่งคืนค่า port number ที่รับ request
- getProtocol() ส่งคืนค่าที่เป็น string ซึ่งแสดง protocol และ version

request ประกอบไปด้วยพารามิเตอร์และข้อมูลต่าง ๆ โดย Servlet สามารถหาขนาดความยาวของ content ได้โดย getContentLength() และหาชนิดของ content ได้จาก getContentType() เมื่อหา content พบ Servlet สามารถอ่านได้โดยใช้ HttpInputStream ซึ่งได้จาก getInputStream() โดย method ที่มีความสัมพันธ์กับ content มีดังนี้

- getParameter (String) ส่งคืนพารามิเตอร์ที่กำหนดมาสำหรับ request
- getParameters() ส่งคืนพารามิเตอร์ทั้งหมดของ request
- getContentType () ส่งคืน MIME type ของข้อมูลที่รวมมากับ request หรือค่า null ถ้าไม่มีการกำหนด MIME type มากับ request
- getInputStream() ส่งคืน input stream เพื่ออ่านข้อมูลที่มากับ request

4. ServletResponse Interface class เป็นตัวกำหนดการสร้าง response สำหรับตอบสนอง request โดยทั่วไปแล้ว response จะประกอบไปด้วยส่วนที่บอกลักษณะของ MIME และขนาดของ content โดยการเขียน content ใน response นั้น Servlet จะเรียก getOutputStream() เพื่อทำหน้าที่รับ ServletOutputStream จากนั้นเรียก print() เพื่อส่ง response โดย method ที่ใช้ในการสร้าง response มีดังนี้

- setContentType (String) เซ็ท MIME type ของ response
- setContentLength(int) เซ็ทขนาดความยาวของ response
- getOutputStream() ส่งคืน ServletOutputStream เพื่อทำการเขียนข้อมูล response ให้กลับไปยัง client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. GenericServlet Class เป็นคลาสพื้นฐานที่สนับสนุนการส่งผ่านข้อมูลของ Servlet และใช้สำหรับการปลุก Servlet จาก server-side include โดยประกอบด้วย method ดังนี้

- init() ถูกเรียกหลังจาก Servlet ได้รับการโหลดให้เป็นขั้นเริ่มต้น โดยจะข้าม method นี้ไปได้เมื่อมีการรับพารามิเตอร์เริ่มต้นต่าง ๆ หรือมีการทำงานที่ฟังก์ชั่นอื่น ๆ ซึ่งในแต่ละครั้งเมื่อ Servlet ถูกโหลด Servlet จะมีค่าอยู่ที่ค่าเริ่มต้นหนึ่งครั้งเท่านั้น
- getlnitParameter(String name) รับค่าพารามิเตอร์เริ่มต้นของ Servlet ทั้งแบบ servet.properties file และอยู่ในรูปของ <servlet> tag สำหรับ server-side includes servlets
- GetlnitParameters() ส่งคืนชื่อ และค่าพารามิเตอร์เริ่มต้นของ FOCUS01>
- GetServletContext () ส่งคืน servlet context.
- Log (String) log ข้อความลงใน servlet log file

6. ServletInputStream เป็น subclass ของ InputStream ที่กำหนด readLine(byte[] ,int offset,int count) method ซึ่งทำการอ่าน byte ลงใน byte array จนกระทั่ง array เต็มหรือมีการอ่านบรรทัดใหม่เข้ามา

7. ServletOutputStream Abstract Class เป็น subclass ของ OutputStream ซึ่งคล้ายกับ PrintStream แต่เป็นเพียงการกำหนด method ที่ใช้ print ข้อมูลโดยทั่วไปชนิดของข้อมูลจะใช้ ints , longs และ strings การทำงานของ ServletOutputStream จะทำการเปลี่ยน code ที่ใช้ในโปรแกรม (internal) ให้เป็น US-ASCII ซึ่งเป็น code ที่ใช้กันทั่วไปในระบบ Internet protocol โดยประกอบด้วย method ดังนี้

- print(int) , println (int)
- Print (long) , println (long)
- Println (String) , println (String)
- Println ()

โดยสามารถแปลงข้อมูลชนิดอื่นก่อนการสั่ง print ออกทาง ServletOutputStream ได้โดยใช้ toString()

Package : java.servlet.http

Java.servlet.http เป็นแพคเกจที่สนับสนุน Servlet ในการใช้ HTTP โดยเฉพาะ แพคเกจนี้ประกอบไปด้วย interface class 2 คลาส implementation class 2 คลาส และคลาสสนับสนุนที่เป็น subclass ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. HttpServlet Abstract Class เป็นเครื่องมือที่ใช้ในการค้นหาและแปลงข้อมูลระหว่าง ServletRequest และ ServletResponse โดยประกอบด้วย method ดังนี้

- service (ServletRequest , ServletResponse) ทำหน้าที่ค้นหาและแปลงข้อมูลทั่ว ๆ ไประหว่าง HTTP-specific request กับ response โดยทั่วไปแล้วจะไม่มีการเขียนทับ method นี้
- service (HttpServletRequest , HttpServletResponse). เขียนทับ abstract method เพื่อรับ HTTP service request

2. HttpServletRequest Interface Class เป็น subclass ของ ServletRequest ซึ่งเป็นตัวจัดการข้อมูลของ extra request และแสดงลักษณะของ content โดยแบ่งประเภทของข้อมูลที่หาได้จาก HttpServletRequest เป็น 4 ลำดับดังนี้

2.1 General Request Information ประกอบด้วย method ที่ให้ข้อมูลทั่วไปเกี่ยวกับ request รวมไปถึงชื่อของ method ที่ใช้ในการเข้าถึงข้อมูลและคำถามต่าง ๆ มีดังนี้คือ

- getMethod () ส่งคืน string ที่บรรจุ method ที่ใช้ใน request เช่น "GET" , "HEAD" หรือ "POST"
- getQueryString (). ส่งคืน query string ซึ่งเป็นส่วนหนึ่งของ URI หรือ ถ้าไม่มีจะส่งค่า null โดยที่ query string จะตามหลังเครื่องหมาย ? ใน URL

2.2 URL Information ประกอบด้วย method ที่ให้ข้อมูลเกี่ยวกับการร้องขอ URL ดังนี้คือ

- getRequestURL () ส่งคืน request URL ส่วนที่อ้างอิงถึงการปลุก Servlet
- getServletPath () ส่งคืน request URL ส่วนที่อ้างอิงถึงการปลุก Servlet
- getPathInfo () ส่งคืนข้อมูลของ extra path ที่ตามหลัง servlet path ซึ่งนำหน้า query string
- getPathTranslated () ส่งคืนข้อมูลของ extra path ที่ส่งผ่านไปยัง real path โดยใช้ alias.properties rule
- getRequestPath () ส่งคืน request URL ส่วนที่ตอบสนอง servlet path ที่เพิ่มเติมด้วยข้อมูลของ extra path

2.3 Authentication-Related Information การเข้าถึงข้อมูลที่เกี่ยวกับชื่อ และที่อยู่ของ client นั้น Servlet สามารถทำได้โดยใช้ getHostName() และ getHostAddress() method แต่สำหรับชนิดของ HTTP level authenication สามารถแสดงได้โดยใช้ getAuthType() และการติดต่อในแบบ basic mode จะมีการส่งคืนชื่อผู้ใช้โดย getRemoteUser() method

- getAuthType() ส่งคืนรูปแบบของการรับรอง (Authenication) ของ request ในกรณีที่หาไม่พบค่าที่ส่งคืนคือ null

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- getRemoteUser() ส่งคืนชื่อของผู้ใช้ที่สร้าง request หรือถ้าไม่มีชื่อมากับ request นั้นจะให้ค่าเป็น null

2.4 Access to HTTP Request Header Fields สำหรับ HttpServletRequest มีการเปิดช่องทางให้มีการเข้าถึง HTTP request header fields โดยตรงเพื่อรับข้อมูลที่ได้รับการสนับสนุนเพียงเล็กน้อยจาก HttpServletRequest method ตัวอย่างเช่น include user-agent , referer , preferred languages และ cookie (สถานะของ client) ซึ่งสิ่งเหล่านี้มี header fields ที่ไม่แน่นอน

ต่อไปจะกล่าวถึง method 2 method ที่ใช้ในการเข้าถึง header เริ่มต้นเพื่อที่จะใส่ค่าดัชนีให้กับ header นั้นโดยค่าดัชนีที่เป็น 0 จะส่งคืน header field อันดับแรก และถ้าค่าดัชนีมีค่ามากกว่าจำนวนของ field ที่เป็นไปได้ค่าที่ส่งคืนคือ null

- getHeader(int) ส่งคืนจำนวนของ header field (n) ถ้ามีค่าน้อยกว่า header field (n) ค่าที่ส่งคืนคือ null
- getHeaderName(int) ส่งคืนชื่อของจำนวน (n) ถ้ามีค่าน้อยกว่า (n) ค่าที่ส่งกลับคือ null

ในกรณีต่อไปเป็นการค้นหา request header ที่ตรงกันกับชื่อของ header field และส่งคืนค่าของ header field นั้นกลับไป โดยเลือกใช้ method ที่กำหนดให้ต่อไปนี้ซึ่งขึ้นอยู่กับรูปแบบประเภทของข้อมูลที่ต้องการจะได้กลับคืนมา

- getHeader(String fieldname) ส่งคืนค่าของ header field ถ้าไม่มีการกำหนด method นี้ ค่าที่ส่งคืนคือ null
- getIntHeader(String fieldname , int default) ส่งคืนค่าที่เป็น integer ซึ่งเป็นค่าของ header field หลังจากมีการแปลงข้อมูลที่เป็น string ให้เป็น int ถ้าไม่พบ field นี้ค่าที่ส่งคืนจะไม่ได้ถูกกำหนดไว้
- getDateHeader(String fieldname , long default) ส่งคืนค่าวันเดือนปีของ header field หลังจากแปลงข้อมูลที่เป็น string ให้เป็น date ถ้าไม่พบ field นี้จะไม่มีการกำหนดค่าที่ส่งกลับ

3. HttpServletResponse Interface Class เป็น subclass ของ ServletResponse และเป็นเครื่องมือในการสร้าง HTTP response header โดยแบ่งตามการนำไปใช้ประโยชน์ได้ 4 แบบ

3.1 Setting the Response Status เป็นการเซ็ทสถานะของ response โดยค่าที่ส่งคืนคือ 200 หรือ OK. ถ้ามี error ส่งคืนหรือ OK. อยู่ในตำแหน่งที่ไม่ถูกต้อง Servlet สามารถระบุแยกรายละเอียดของทั้ง error code และ ข้อความที่บอกถึงลักษณะของ response ได้เอง ถ้ามีเพียง error เกิดขึ้นเท่านั้น Servlet จะแปลง code ให้อยู่ในรูปข้อความที่เป็น string แทนซึ่ง method ต่างมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- SetStatus (status Code) เซ็ท status code ในการส่งคืน HTTP header โดย status code ถูกแปลงให้เป็นข้อความที่เป็น string
- SetStatus (statusCode , messageString) เซ็ท status code ในการส่งคืน HTTP header

3.2 Specifying Non-Standard Header Fields การรวม field ต่าง ๆ กับ response header นั้น Servlet ได้กำหนดให้ใช้ setHeader () method แต่สำหรับ non-standard header field ซึ่งไม่ได้รับการสนับสนุนจาก HttpServletResponse ซึ่งกำหนดให้ชื่อของ header fiald (ไม่มีเครื่องหมาย ;) อยู่ที่ argument แรก และข้อมูลอยู่ที่ argument ที่ 2 setHeader ที่ต่างกันนั้นแยกด้วยชนิดของพารามิเตอร์ที่เป็น String , int และ date โดยประกอบด้วย method ดังนี้

- SetHeader (String fieldname , String) เซ็ทค่าของชื่อ header field เพื่อแสดงค่าออกมา
- SetIntHeader (String fieldname , int) เซ็ทค่าของชื่อ header field ที่เป็น integer หลังจากมีการแปลงค่าให้เป็น string
- SetDateHeader (String fieldname , long) เซ็ทค่าของชื่อ date field หลังจากแปลง long เป็น Internet-standard date-time string
- UnsetHeader(String fieldname) ย้ายชื่อของ header field ออกจาก response

3.3 Returning Error Response การสร้างและส่ง standard error response เพื่อส่งคืน eroor นั้นมี method ที่ทำหน้าที่นี้อยู่ภายใต้ HttpServletResponse คือ sendError() method ซึ่งประกอบด้วย status cod และข้อความที่อธิบายรายละเอียดของ error ซึ่งการส่งค่าคืนนี้จะส่งไปบน page โดยประกอบด้วย header ซึ่งแสดงค่า status code และข้อความมาตรฐานของ status code ในขณะที่ข้อความที่อธิบายรายละเอียดของ error นั้นจะอยู่บน body ของ page ที่ส่งคืน ตัวอย่างเช่น เมื่อมีการเรียก sendError (404 , "Couldn't find it!") จะมีการสร้าง page ซึ่งประกอบด้วย header "404 Not Found" และ ข้อความคือ "Couldn't find it!" โดย method ที่กล่าวถึงมีดังนี้

- sendError(int) ส่ง error response ไปที่ client โดยระบุแต่ status code และไม่มีข้อความเพิ่มเติม
- sendError(int , String) ส่ง error response ไปที่ client โดยระบุทั้ง status code และข้อความที่อธิบายรายละเอียดของ error

3.4 Returning Redirect Response ในการสร้างและส่ง standard redirect response นั้น HttpServletResponse ได้กำหนดให้ใช้ sendRedirect() method โดยจะมีการสร้างและส่งค่า "302" เป็น response ค่าที่ส่งไปจะเป็น string ซึ่งเป็นตัวแสดงตำแหน่งใหม่ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

page โดย string ที่กล่าวถึงนี้ก็คือ URL นั่นเองเนื่องจาก URL คือที่อยู่ที่ต้องการส่ง page ไปเหมือนกับการ link นั่นเอง อย่างไรก็ตาม sendRedirect() ไม่สามารถตรวจสอบได้ว่า string ที่เข้ามานั้นตรงกับ URL ที่ถูกต้องหรือไม่ ถ้าผู้เขียนมี URL ที่ถูกต้องแล้วให้แปลง URL นั้นเป็น string โดยใช้ toString() ก่อนที่จะเรียก sendRedirect() โดยการเรียก method นี้จะอยู่ในรูป sendRedirect (String) ซึ่งจะส่ง redirect response ไปยัง client โดยการระบุที่อยู่ของ URL ที่ต้องการมาด้วย

4. FormServlet Class เป็นคลาสของ Servlet ที่ถูกระบุให้เป็นส่วนหนึ่งของ Servlet API และด้วยเหตุนี้มันจึงสามารถอยู่บน web server ทุกชนิดที่สนับสนุน Servlet โดยทำหน้าที่สร้างแบบฟอร์มการทำงานของ Servlet โดย subclass ของ FormServlet และเขียนทับบน sendResponse() method ซึ่ง method นี้ทำหน้าที่ควบคุมแบบฟอร์มของข้อความและสร้าง response ซึ่งเป็น subclass ของ HttpServlet โดย method ต่อไปนี้ถูกกำหนดไว้ใน FormServlet Class

- getServletInfo() เขียนทับ method นี้เพื่อส่งคืน string ที่บรรจุอยู่ในข้อความของ Servlet เช่น ชื่อผู้เขียนหรือเวอร์ชั่น
- service (HttpServletRequest , HttpServletResponse) ผู้เขียนสามารถเขียนทับ method นี้เมื่อมีความต้องการ special request processingsendResponse (HttpServletResponse , Hashtable) เขียนทับ method นี้เมื่อต้องการแปลง code เพื่อจัดรูปแบบและสร้าง response 5 HttpUtils Class เป็นคลาสที่ทำหน้าที่กำหนด static utility routine ในการสร้าง Servlet

Package : java.servlet.html

แพคเกจลำดับที่ 3 ซึ่งกำหนดใน Servlet API จะกล่าวถึงต่อไปนี้คือ java.servlet.html โดยแพคเกจนี้ประกอบด้วยคลาสที่สนับสนุน HTML 13 คลาส ซึ่งสนับสนุนการปฏิบัติการของ HTML เช่น ตาราง , การแทรก รูปภาพ และ ฟอร์ม ผู้เขียนจึงสามารถสร้าง HTML page โดยใช้ภาษา HTML จากนั้นจึงส่ง HTML page นั้นเขียนลง ServletOutputStream โดยคลาสต่างๆ มีดังนี้

1. HtmlElement สนับสนุนการกำหนด basic interface ประกอบด้วย 2 method ดังนี้
   - wrap() สำหรับรวม HTML tag pairs (เช่น <h1> และ </h1>) เข้ากับ element โดย tag ถูกระบุให้เป็น argument ใน wrap ()
   - write() เขียน HTML ไปที่ OutputStream
2. HtmlText สนับสนุนข้อความ (text) ที่อยู่ภายใน tag ประกอบด้วย 2 method ดังนี้
   - add(String) , add (String tag) เพิ่มข้อความ (text) ซึ่งรวมไปถึงข้อความที่อยู่ใน tag
   - addTag(String), addTag(String ,String) เพิ่ม tag เดี่ยวที่ไม่เป็นคู่ (non-paired tag) เช่น <p>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. HtmlContainer Class สนับสนุน list ของ HtmlElement ที่อยู่ภายใน tags ซึ่ง HtmlContainer ประกอบด้วย beginning tags , list ของ HtmlElement และ ending tags Htmlcontainer สนับสนุน wrap (), write () และชนิดต่าง ๆ ของ add() และ addTag() method โดย HtmlContainer ประกอบด้วย method ต่าง ๆ ดังนี้

- addImg (String) เพิ่ม <IMG> tag โดยที่ String คือ URL ที่บรรจุรูปภาพที่ต้องการ
- addLink (String text , String URL) เพิ่ม link เมื่อ "text" อยู่ระหว่าง tags และ URL โดยเป็น URL ที่ต้องการจะ Link ไป

4. HtmlApplet Class สนับสนุน applet HTML tag กับรายละเอียดของความกว้างและความสูง
5. HtmlChoicd Class สนับสนุน HTML tag ที่ถูกเลือก
6. HtmlDefinitionList Class สนับสนุน dl , dt และ dd HTML tags
7. HtmlForm Class สนับสนุนการสร้าง HTML form ซึ่งประกอบด้วย input fields , check boxes , radio buttons , text area , selections และ submit buttons
8. HtmlFramest Class สนับสนุน client-side image map HTML tags
9. Html ImageMap Class สนับสนุน client-side image map HTML tags
10. HtmlList Class สนับสนุน ul และ ol HTML list tags
11. HtmlPages Class สนับสนุนการสร้าง web page ที่ประกอบด้วย header และ body areas
12. HTMLTable class สนับสนุนการสร้างตารางจาก row element
13. HtmlRow Class สนับสนุน HTML เพื่อกำหนดแถวของตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การสื่อสารข้อมูล

การสื่อสารข้อมูลที่จะกล่าวถึงในบทนี้เป็นการสื่อสารข้อมูลระหว่างคอมพิวเตอร์กับอุปกรณ์ภายนอก โดยมีรูปแบบของการติดต่อกับอุปกรณ์ภายนอกที่นิยมมี 2 รูปแบบคือ

1 การสื่อสารแบบขนาน

2 การสื่อสารแบบอนุกรม

แต่ก่อนที่จะทำความรู้จักกับการสื่อสารทั้งสองควรจะมีความเข้าใจเกี่ยวกับคำส่งหรือข้อมูลที่อยู่ในรูปของบิท ซึ่งประกอบด้วยหลายบิตมาประกอบกัน ข้อมูลในการสื่อสารแต่ละข้อมูลจะถูกแปลงให้อยู่ในรูปแบบของเลขฐานสอง แล้วนำมาประกอบกัน เช่น ถ้าข้อมูลที่ประกอบด้วย 4 บิต เราจะเรียกว่า 1 นิบเบิล หรือถ้าหากข้อมูลที่ประกอบด้วย 8 บิตเราจะเรียกว่า 1 ไบต์ เป็นต้น

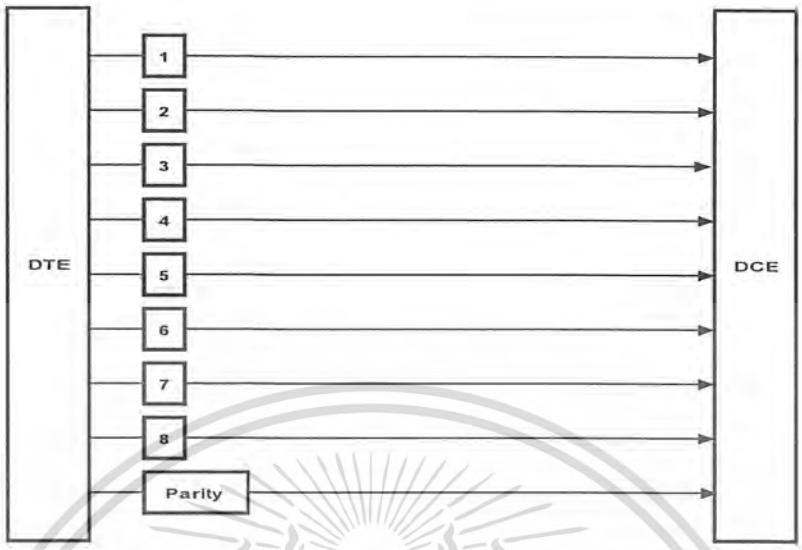
3.1 การสื่อสารแบบขนาน

การสื่อสารแบบขนานจะมีรูปแบบการส่งข้อมูลครั้งละ 1 ไบต์ก็คือจะทำการส่งข้อมูลครั้งละ 8 บิตนั่นเอง ซึ่งในการส่งต้องใช้สายไฟในการส่งข้อมูล 8 เส้น แล้วยังต้องใช้สายไฟอีก 1 เส้นในการควบคุมเช่น ใช้เป็นพาริตี้บิต หรืออาจจะมีมากกว่านั้นเพื่อใช้ในการควบคุมการโต้ตอบของการทำงาน (Hand-shake) ซึ่งรายละเอียดจะบอกใช้ทราบต่อไปในเรื่องของการ Hand-shake จึงสรุปได้ว่าในการสื่อสารแบบขนานนั้นต้องใช้สายอย่างน้อยที่สุด 9 เส้น

ดั้งนั้นในการส่งข้อมูลทีละ 1 ไบต์นั้นทำให้ข้อมูลทั้ง 8 บิตมาถึงปลายทางพร้อมกัน ทำให้ข้อมูลแบบขนานสามารถทำได้ด้วยความเร็วที่สูงมาก แต่ปัญหาที่สำคัญของการส่งข้อมูลแบบขนานคือคุณสมบัติของบิตกับแรงดัน เวลาที่บิตหรือแรงดันไฟฟ้าที่ใช้ในการเปลี่ยนสถานะจากหนึ่งเป็นศูนย์นั้นสั้นมาก โดยเร็วถึงระดับนาโนวินาที (หนึ่งในพันล้านของวินาที) การเปลี่ยนแปลงที่รวดเร็วนี้เป็นส่วนที่สำคัญมากต่อการส่งข้อมูล เพราะการเปลี่ยนแปลงระหว่างศูนย์และหนึ่งอย่างช้า ๆ จะไม่ถูกอ่านเป็นข้อมูลเลย และเมื่อสายไฟที่ใช้ส่งข้อมูลยาวขึ้น คุณสมบัติทางไฟฟ้าของสายไฟเช่น ค่าความจุไฟฟ้าและค่าความเหนี่ยวนำจะจำกัดความเร็วในการเปลี่ยนแปลงระหว่างศูนย์และหนึ่งของบิต ซึ่งจะทำให้ข้อมูลอาจสูญหายหรือทำให้การส่งข้อมูลล้มเหลวได้ ดังนั้นการส่งข้อมูลบนสายยาวอาจจะเป็นปัญหาได้หากใช้วิธีการสื่อสารแบบขนาน

เนื่องจากข้อเสียของการส่งข้อมูลแบบขนานสองอย่างคือค่าใช้จ่ายที่สูงและการสูญหายของข้อมูล ทำให้การใช้งานของมันถูกจำกัดอยู่กับอุปกรณ์เพียงไม่กี่ชนิด เช่น เครื่องพิมพ์ที่มักจะอยู่ใกล้กับเครื่องคอมพิวเตอร์และต้องทำงานที่ความเร็วสูง แต่เรายังใช้วิธีการส่งข้อมูลแบบขนานนี้ภายในเครื่องคอมพิวเตอร์เนื่องจากไม่ต้องใช้สายไฟขนาดยาว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



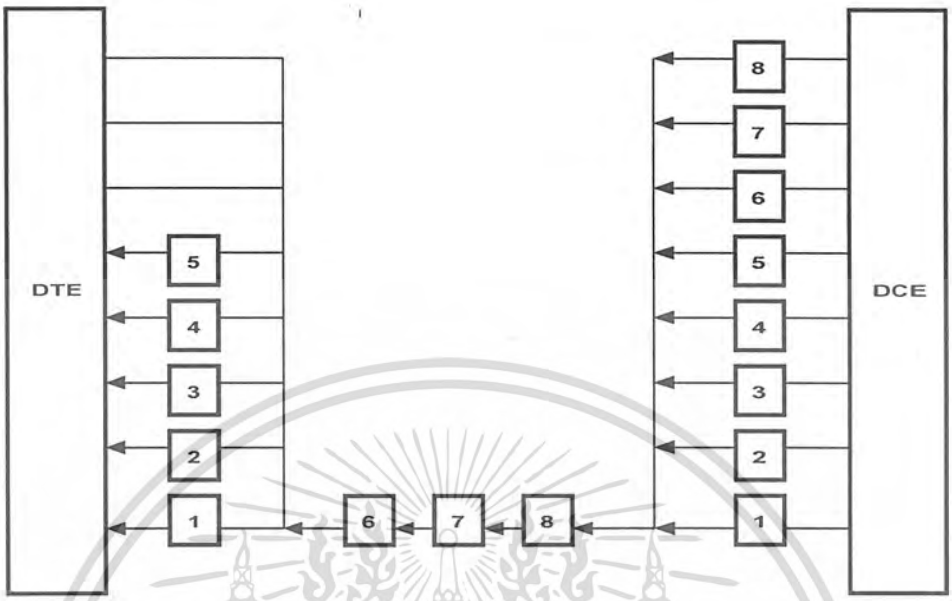
รูปที่ 3.1 แสดงการสื่อสารแบบขนาน

3.2 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมเป็นการส่งข้อมูลทีละบิต และข้อมูลจะถูกต่อรวมเข้าเป็นไบต์ใหม่ ด้วยวิธีการส่งข้อมูลทีละบิตนี้ ทำให้สามารถใช้สายไฟเพียงสองเส้นในการส่งข้อมูล ซึ่งช่วยให้เราประหยัดค่าสายไฟไปได้มาก แต่ก็ทำให้ประสิทธิภาพการทำงานลดลงไปด้วย เพราะการส่งข้อมูลวิธีนี้ต้องใช้เวลานานขึ้นอย่างน้อยแปดเท่าของการส่งข้อมูลแบบขนาน แต่ความเร็วที่ลดลงไปนี้ยังไม่ถือว่าเป็นข้อจำกัดที่สำคัญทางการใช้งานนัก เพราะหากว่าเราพิจารณาที่อุปกรณ์ทั่วไป จะพบว่าอุปกรณ์ส่วนใหญ่ทำงานช้ามากเมื่อเทียบกับความเร็วในการทำงานภายในไมโครโปรเซสเซอร์ อุปกรณ์แต่ละตัวมีขั้นตอนการทำงานที่กินเวลานาน ซึ่งโดยทั่วไปมักจะเป็นกระบวนการทางกลไก (mechanic)ที่เป็นตัวจำกัดความเร็วของเครื่องลงไปอย่างมาก

ตัวอย่างเช่น ความเร็วของเครื่องพิมพ์ถูกจำกัดที่ความเร็วของหัวพิมพ์ (printhead)ความเร็วของโมเด็มถูกจำกัดโดยขีดจำกัดความถี่ของสายโทรศัพท์ และความเร็วของดิสก์ไดรฟ์ถูกจำกัดโดยอัตราเร็วการหมุนของไดรฟ์ ดังนั้นความเร็วที่ได้มาจากการส่งข้อมูลแบบขนานจะเสียไปโดยเปล่าประโยชน์เมื่อนำมาใช้งานได้ แม้ว่าอัตราเร็วการส่งข้อมูลจะลดลงแต่ก็ยังสามารถใช้งานร่วมกับอุปกรณ์ประเภทนี้ได้อย่างมีประสิทธิภาพ ข้อเสียจากความเร็วที่ลดลงไปไม่อาจเทียบได้กับผลพวงที่ได้จากคุณภาพการส่งและระยะทางการส่งข้อมูลที่เพิ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงการสื่อสารข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 ตารางเปรียบเทียบการทำงานระหว่างการสื่อสารข้อมูลแบบขนานและแบบอนุกรม

การใช้งาน	แบบขนาน	แบบอนุกรม
1.ระยะทาง	จะใช้งานได้ในระยะไม่เกิน100 ฟุต	จะสามารถใช้งานได้ตั้งแต่ในระยะใกล้ ๆ ไปจะถึงระยะทางที่มาก ๆ จนถึงหลักไมล์
2.ความเร็ว	อัตราความเร็วสูงมากในระยะที่ไม่ไกลมากนักกำหนดได้เป็นจำนวนบิทต่อวินาที	อัตราความเร็วของข้อมูลที่ใช้กันอยู่ทั่วไปจะอยู่ในช่วง 0-2 ล้านบิทต่อวินาที
3.ระดับของสัญญาณ	ในการอินเตอร์เฟสจะใช้ระดับของสัญญาณที่ใช้กับอุปกรณ์ TTL คือสัญญาณลอจิก 1 และ 0 จะแทนด้วยระดับแรงดัน +5V และ 0V	ในมาตราฐานของ EIA – RS 232C ระบุว่ามีระดับสัญญาณไฟฟ้าขนาด 12V หรือใช้มาตราฐาน 20 mA current loop
4.ความผิดพลาดของสัญญาณ	ในการใช้งานระยะไกลๆ ความผิดพลาดของข้อมูลจะเกิดขึ้นได้ง่าย	การใช้งานจะเกิดการผิดพลาดของสัญญาณจะมีน้อยมาก
5.ค่าใช้จ่าย	ค่าใช้จ่ายจะสูงมากเพราะจะต้องใช้สายส่งสัญญาณหลายเส้น โดยเฉพาะการส่งในระยะทางไกล ๆ	สิ้นเปลืองน้อยกว่ามาก ถึงแม้ว่าจะต้องใช้อุปกรณ์เปลี่ยนสัญญาณของข้อมูลจากขนานไปเป็นอนุกรมแล้วส่งผ่านสายส่ง แล้วกลับสัญญาณมาเป็นขนานอีกครั้งก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3.3 การอินเตอร์เฟส

การส่งข้อมูลอยู่ภายในเครื่องคอมพิวเตอร์สามารถกระทำได้ง่าย เนื่องจากเราสามารถคาดเดาสภาพแวดล้อมภายในเครื่องได้ แต่ในการส่งข้อมูลสู่ภายนอก เราไม่สามารถคาดการณ์ได้ว่าเครื่องคอมพิวเตอร์และตัวข้อมูลจะต้องพบกับสภาพเช่นไรและจะมีผลกระทบต่อตัวข้อมูลและเครื่องคอมพิวเตอร์อย่างไร ดังนั้นในการออกแบบวงจรจึงมีสิ่งที่ต้องพิจารณาคือ จะต้องหาวิธีในการแยกข้อมูลออกจากสภาพแวดล้อมและสัญญาณรบกวนโดยเฉพาะอย่างยิ่งในกรณีที่มีสายส่งข้อมูลขนาดยาว สิ่งที่จะต้องพิจารณาอีกสิ่งหนึ่งก็คือ จะต้องหาวิธีการป้องกันคอมพิวเตอร์จากสภาพแวดล้อมอันไม่พึงประสงค์ด้วย นั่นก็คือจะต้องมีตัวอินเตอร์เฟส ซึ่งจะมีหน้าที่เป็นจุดเชื่อมต่อระหว่างคอมพิวเตอร์กับสิ่งแวดล้อมภายนอก และเนื่องจากอุปกรณ์อินเตอร์เฟสมีหน้าที่คล้ายกับเป็นประตูของเครื่องคอมพิวเตอร์ บางครั้งมันจึงถูกเรียกว่า I/O พอร์ต (I/O PORT)หรือบางครั้งเรียกสั้นๆ ว่าพอร์ต (PORT)

วัตถุประสงค์หลักของการอินเตอร์เฟสก็คือ การใช้อุปกรณ์อินเตอร์เฟสเป็นสื่อกลางของการส่งข้อมูล และวัตถุประสงค์ที่สำคัญอีกอย่างหนึ่งของการอินเตอร์เฟสก็คือ ความง่ายต่อการใช้งาน และเมื่อเราสามารถทำการอินเตอร์เฟสได้สำเร็จ ก็จะสามารถที่จะส่งข้อมูลสู่ภายนอกได้

มาตราฐานการอินเตอร์เฟส RS-232

มาตราฐานการอินเตอร์เฟส RS-232 ได้เกิดขึ้นตั้งแต่ปี 1969 EIA (Electronic Industries Association) ห้องวิจัย Bell และบรรดาผู้ผลิตอุปกรณ์สื่อสารได้ร่วมกันจัดมาตราฐาน EIA RS-232 ซึ่งต่อมาไม่นานนักก็ได้มีการปรับปรุงแก้ไขอีกเล็กน้อยกลายเป็น RS-232 C และเมื่อไม่นานมานี้ก็ได้ออกมาตราฐาน RS-232 D

เพื่อทำความเข้าใจกับวัตถุประสงค์หลักของ RS-232 ได้ดียิ่งขึ้น ควรทำความเข้าใจกับวัตถุประสงค์หลักของ RS-232 ก่อน ซึ่งได้แสดงไว้อย่างชัดเจนในหัวข้อของเอกสารคือ

Interface Between Data Terminal Equipment and Data Communications Equipment Employing Serial Binary Data Interchange (การอินเตอร์เฟสระหว่างอุปกรณ์เทอร์มินัลและอุปกรณ์สื่อสารข้อมูลที่ใช้วิธีการแลกเปลี่ยนข้อมูลไบนารีแบบอนุกรม)

ซึ่งในหัวข้อสำคัญจะอธิบายถึงการอินเตอร์เฟสระหว่างเทอร์มินัล (Data Terminal Equipment หรือ DTE) กับโมเด็ม (Data Communications Equipment หรือ DCE) เพื่อใช้ส่งข้อมูลแบบอนุกรม โดยจะประกอบด้วย 4 ส่วนคือ

- คุณสมบัติทางไฟฟ้าของสัญญาณ (Electrical Signal Characteristics) ซึ่งจะอธิบายถึงรูปแบบของสัญญาณไฟฟ้าที่ตัวอินเตอร์เฟสจะส่งออก และรับเข้ามาจากภายนอก ระดับแรงดันไฟฟ้าที่แสดงถึง ตรรกะ 0 และ 1 ก็จะมีกำหนดไว้ในส่วนนี้ด้วย
- คุณสมบัติทางกลไกการอินเตอร์เฟส : คอนเน็กเตอร์ (Interface Mechanical Characteristics : Connectors) ซึ่งเป็นข้อกำหนดเกี่ยวกับตัวอินเตอร์เฟส ประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42235

ส่วนที่เป็นปลั๊ก (plug) และเต้าเสียบ (receptacle) โดยเต้าเสียบจะต้องอยู่บน DCE สำหรับ RS-232 A-C ไม่ได้มีการกำหนดคอนเน็กเตอร์รูปตัว D (D-Shaped) ซึ่งมีใช้กันอยู่ทั่วไปนั้น ทั้งนี้เพราะว่าอุปกรณ์ตัวนี้ได้รับการคุ้มครองโดยลิขสิทธิ์ และเมื่อสิทธิบัตรนั้นหมดอายุลงใน RS-232 D จึงได้เพิ่มข้อกำหนดคอนเน็กเตอร์ DB-25 เข้าไว้ในมาตราฐานด้วย

- หน้าที่การทำงานของวงจรการแลกเปลี่ยน (Functional Description of Interchange Circuit) ในส่วนนี้กำหนดหน้าที่และตั้งชื่อให้กับสัญญาณไฟฟ้าต่าง ๆ ที่นำมาใช้ เช่น Transmitted Data (ข้อมูลส่งออก) ได้ถูกกำหนดไว้ให้กับขา 2
- มาตราฐานการอินเตอร์เฟสสำหรับระบบการสื่อสารเฉพาะอย่าง (Standard Interffaces for Selected Communications System Configurations) ในส่วนนี้เป็นรายละเอียดต่าง ๆ สำหรับการติดต่อระหว่างโมเด็มกับเทอร์มินัลทั่วไป

พื้นฐานการอินเตอร์เฟส RS-232

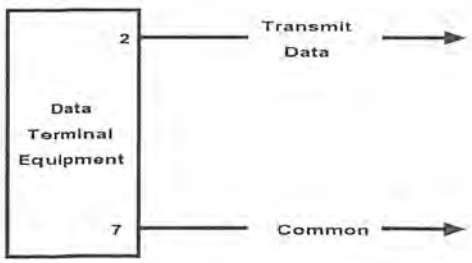
ก่อนที่จะทำความรู้จักกับพื้นฐานการอินเตอร์เฟส RS-232 ก่อนอื่นต้องทำความเข้าใจกับคำว่า DTE และ DCE ก่อน

DTE = Data Terminal Equipment ซึ่งก็คือ คอมพิวเตอร์ นั่นเอง (ตัวส่งข้อมูล)

DCE = Data Communications Equipment อุปกรณ์เหล่านี้ ได้แก่ โมเด็ม,TA อะแดปเตอร์, พลีอตเตอร์ เป็นต้น (ตัวรับข้อมูล)

ตามมาตราฐานของ EIA ได้กำหนดไว้ว่า อุปกรณ์ DTE หมายถึง อุปกรณ์ที่ข้อมูลมาสิ้นสุดและอุปกรณ์ DCE เป็นอุปกรณ์ที่ใช้ส่งผ่านข้อมูล ดังนั้นคอมพิวเตอร์ซึ่งสามารถทำได้ทั้งส่งผ่านข้อมูลและรับข้อมูล จึงไม่อาจระบุได้ว่าเป็นอุปกรณ์ DCE หรือ DTE

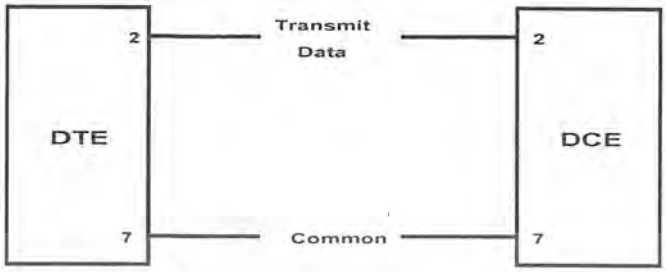
ในการพิจารณาโครงสร้างเบื้องต้นของการอินเตอร์เฟส RS-232 นั้นจะประกอบด้วยเส้นสายไฟเพียง 2 เส้น เส้นหนึ่งใช้ในการส่งข้อมูล และอีกเส้นหนึ่งสำหรับอ้างอิงแรงดันของวงจรอินเตอร์เฟส (circuit common) ซึ่งมักจะมีความเข้าใจผิดว่าคือกราน์ด (ground) แต่แท้จริงแล้วไม่ใช่ ดังรูปที่ 3.3



รูปที่ 3.3 แสดงอุปกรณ์ DTE เบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.3 เราจะพบว่ามีเพียงตัวส่งข้อมูล แต่ในความจริงแล้วในการอินเตอร์เฟสข้อมูลใดๆ จะต้องมีส่วนที่รับข้อมูลด้วย ซึ่งก็คือ DCE นั่นเอง ดังรูป 3.4



รูปที่ 3.4 แสดงอุปกรณ์ทั้ง DTE และ DCE

เมื่อพิจารณารูปที่ 3.4 จะพบว่าข้อมูลที่ถูกส่งออกจากขา 2 ของ DTE จะรับเข้าไปยังขา 2 ของ DCE เช่นกัน โดยข้อมูลที่รับทาง DCE แน่นอนจะต้องเป็นข้อมูลเดียวกันที่ส่งออกมาจาก DTE ทำให้สรุปได้ว่า " Transmit Data" มิได้มีส่วนในการกำหนดว่าอุปกรณ์ใดเป็นตัวต้นทางหรือปลายทาง แต่จะขึ้นอยู่กับว่าจะพิจารณาเช่นไร

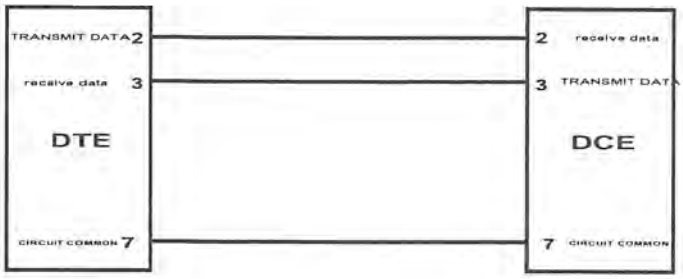
ในที่นี้จะใช้การพิจารณาดังนี้คือ สัญญาณที่ส่งออกไปเรียกว่า "เอาต์พุต" และสัญญาณที่รับเข้ามาเรียกว่า "อินพุต" โดยถือว่าสัญญาณเป็นกิจกรรมทางไฟฟ้าที่เกิดขึ้นในกระบวนการอินเตอร์เฟส ดังรูปที่ 3.5

รูปที่ 3.5 แสดงอุปกรณ์ DTE และ DCE ซึ่งเป็นคู่อุปกรณ์ที่ทำงานตรงข้ามกัน

3.4 การรับส่งข้อมูลสองทิศทาง

เมื่อพิจารณารูปที่ 3.5 แล้วเราจะพบว่าเมื่ออุปกรณ์ DTE รับข้อมูลแล้วจะต้องส่งผ่านไปยังอุปกรณ์ DCE (โมเด็ม) แล้วโมเด็มก็จะทำหน้าที่ในการส่งข้อมูลต่อออกไปยังสายโทรศัพท์ ซึ่งจะเป็นได้ว่าอุปกรณ์ DTE และอุปกรณ์ DCE สามารถเป็นได้ทั้งอุปกรณ์ส่งและรับข้อมูล และยังสามารถส่งและรับข้อมูลในทิศทางตรงกันข้ามได้อีกด้วย ดังรูปที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงการทำงานของอุปกรณ์ที่สามารถส่งและรับข้อมูลได้ทั้งสองทิศทาง

ซึ่งจากรูปที่ 3.6 ก็คงจะพอสรุปความแตกต่างระหว่างอุปกรณ์ DTE และ DCE ได้ดังนี้

DTE ส่งเอาต์พุตที่ขา 2 และรับอินพุตที่ขา 3

DCE ส่งเอาต์พุตที่ขา 3 และรับอินพุตที่ขา 2

โดยในการอินเตอร์เฟสนั้นจะต้องทำการตรวจสอบทิศทางของสัญญาณข้อมูลที่ขา 2 และที่ขา 3 ก่อนเสมอ

การแฮนด์เชคใน RS-232

การแฮนด์เชคหมายถึง กระบวนการที่อุปกรณ์หนึ่งใช้ตรวจสอบสถานะของอีกอุปกรณ์ที่ต่อเข้าด้วยกัน และตอบสนองสถานะนั้นอย่างเหมาะสมและถูกจังหวะเวลา ซึ่งก็คือวิธีการควบคุมการทำงานของอุปกรณ์สองตัวให้สัมพันธ์กันในการรับส่งข้อมูลนั่นเอง การแฮนด์เชคนั้นแบ่งออกเป็น 2 ประเภทคือ การแฮนด์เชคทางฮาร์ดแวร์ และ การแฮนด์เชคทางซอฟต์แวร์

การแฮนด์เชคทางซอฟต์แวร์ (software handshaking) เป็นวิธีการหนึ่งในการควบคุมการทำงานของอุปกรณ์รับข้อมูลโดยส่งผ่านสัญญาณควบคุมไปพร้อมกับตัวข้อมูลที่ต้องการส่ง

การแฮนด์เชคทางฮาร์ดแวร์ (hardware handshaking) สามารถควบคุมได้ตั้งแต่ระดับฮาร์ดแวร์โดยการเปลี่ยนระดับแรงดันในสายสัญญาณควบคุมเป็นตัวระงับไม่ให้คอมพิวเตอร์ส่งข้อมูลเพิ่มเข้ามาอีก ซึ่งเป็นการหลีกเลี่ยงการใช้รหัสหรือโปรแกรม แต่การแฮนด์เชคทางฮาร์ดแวร์นั้นมีข้อจำกัด คือจำเป็นต้องมีสายสัญญาณควบคุมต่างหากโดยเฉพาะ ทำให้วิธีนี้ไม่เหมาะที่จะนำมาใช้ในการอินเตอร์เฟสกับโมเด็ม

ขอบเขตความคอมแพติเบิล (compatible) กับ RS-232

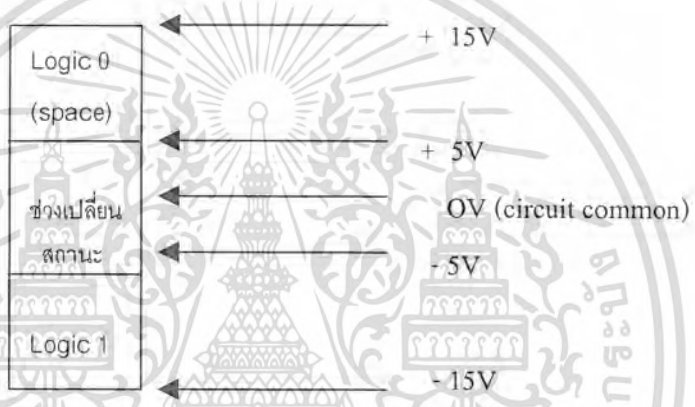
คุณสมบัติทางไฟฟ้า (ระดับแรงดัน ฯลฯ) ของการอินเตอร์เฟสได้รับการตรวจรับรอง ถ้าอุปกรณ์นั้นถูกอ้างว่าคอมแพติเบิลกับ RS-232 ย่อมหมายความว่าเราสามารถนำอุปกรณ์นั้นไปติดต่อกับอุปกรณ์อื่นที่คอมแพติเบิลกับ RS-232 ได้ โดยไม่ทำให้เกิดความเสียหายแก่อุปกรณ์ทั้งคู่ ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เงื่อนไขนี้ช่วยให้เรามั่นใจได้ว่าอุปกรณ์ทั้งคู่มีระบบไฟฟ้าที่ทำงานด้วยกันได้ โดยไม่ทำให้เกิดความบกพร่องในการรับส่งข้อมูล ระดับแรงดันสำหรับค่า "ศูนย์" และ "หนึ่ง" ต้องเป็นไปตามที่ระบุไว้ในมาตราฐาน

มาตราฐานระดับตรรกะ (logic) ใน RS-232

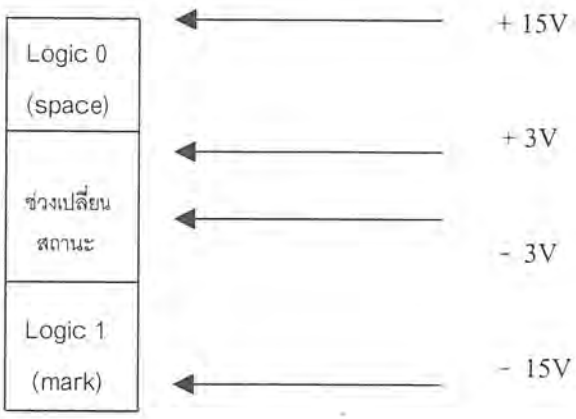
การส่งข้อมูลจากวงจรอินเตอร์เฟสมีลักษณะ " กลับหัว " กับวงจรที่ใช้งานกันอยู่ทั่วไป โดยความสัมพันธ์ระหว่างแรงดันกับค่าตรรกะบนอินเตอร์เฟสคือ แรงดันบวกบนอินเตอร์เฟสจะถูกแทนด้วย 0 ในขณะที่แรงดันลบแทนด้วยค่า 1 ดังรูปที่ 3.7



รูปที่ 3.7 แสดงคำจำกัดความค่าตรรกะที่เอาต์พุตของ RS-232

ขอให้สังเกตค่าตรรกะที่กลับกันให้ดีคือ แรงดันลบแทนด้วยค่า 1 และแรงดันบวกแทนด้วยค่า 0 เพื่อให้แน่ใจค่าตรรกะ 0 แรงดันไฟที่เอาต์พุตจะต้องอยู่ในช่วง +5V ถึง +15V และในทำนองเดียวกันในการแทนระดับ 1 ระดับแรงดันที่เอาต์พุตจะต้องอยู่ในช่วง –5V ถึง –15V สำหรับช่องว่างหรือ dead-band ที่อยู่ในช่วง +5V ถึง –5V มีชื่อเรียกว่า ช่วงเปลี่ยนสถานะ (transition region) เป็นที่ไม่สามารถกำหนดค่าตรรกะได้ ซึ่งหมายความว่าค่าแรงดันเอาต์พุตในช่วง +5V ถึง -5V นั้นอาจถูกแปลความหมายให้เป็น 0 หรือ 1 ก็ได้ ดังรูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงคำจำกัดความค่าตรรกะที่อินพุตของ RS-232

ข้อแตกต่างเพียงข้อเดียวระหว่างคำจำกัดความสำหรับอินพุตกับเอาต์พุตคือ ความกว้างของช่วงเปลี่ยนสถานะ (transition region) โดยช่วงที่ไม่สามารถกำหนดค่าตรรกะได้ของอินพุตกว้างเพียง 6 V (+3V ถึง –3V) ในขณะที่ช่วงเดียวกันนี้สำหรับเอาต์พุตกว้างถึง 10 V (+ 5V ถึง –5V) ซึ่งความแตกต่างนี้มีความสำคัญอย่างมากทีเดียว

ช่วงการยอมรับสัญญาณรบกวน

ความแตกต่างระหว่างคำจำกัดความของแรงดันต่ำสุดที่วงจรยอมรับได้เรียกว่า ช่วงการยอมรับสัญญาณรบกวน (noise margin) ซึ่งหมายความว่าวงจรยอมให้มีสัญญาณรบกวนออกจากเอาต์พุตเข้าสู่อินพุตได้โดยไม่ส่งผลต่อค่าตรรกะที่อินพุต ซึ่งคุณสมบัติข้อนี้มีประโยชน์มากในเวลาที่จำเป็นต้องเดินสายข้อมูลผ่านอุปกรณ์ที่เป็นตัวสร้างสัญญาณรบกวน เช่น มอเตอร์ไฟฟ้า หลอดไฟฟลูออเรสเซนต์ วงจรหรี่ไฟ และอุปกรณ์การสื่อสารต่าง ๆ

ส่วนต่างระหว่างช่วงเปลี่ยนสถานะของอินพุตและเอาต์พุตนอกจากจะทำหน้าที่เป็นช่วงยอมรับสัญญาณรบกวนแล้ว ยังทำหน้าที่เป็นช่วงปลอดภัย (safety margin) ด้วย โดยการให้แรงดันเผื่อสำหรับแรงดันที่ตกคร่อมสายเคเบิล ทำให้วงจรสามารถรับแรงดันที่ลดลงจากเอาต์พุตได้ถึงสองโวลต์โดยข้อมูลไม่ตกเข้าสู่ช่วงที่กำหนดตรรกะไม่ได้ของอินพุต

เนื่องจากแรงดันไฟกระแสตรง (direct current voltage) สูญเสียไปน้อยมากในสายเคเบิลจนสามารถตัดทิ้งไปได้แม้ในสายไฟขนาดยาวๆ ดังนั้นมาตราฐาน RS-232 จึงมีข้อกำหนดสำหรับสัญญาณควบคุมน้อยกว่าสัญญาณข้อมูล เนื่องจากสัญญาณควบคุมและสัญญาณการแฮนด์เชคเป็นสัญญาณแรงดันไฟฟ้ากระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mark และ Space

การรักษาระดับกระแสไฟให้คงที่ในระหว่างช่วงรอทำงาน (idle) ซึ่งเป็นช่วงที่ไม่มีการส่งข้อมูลออกมาของการส่งข้อมูล สามารถเพิ่มความน่าเชื่อถือของอุปกรณ์จะเพิ่มขึ้นอย่างมาก การส่งข้อมูลสามารถทำได้โดยการขัดจังหวะ (interrupt) กระแสไฟขณะหยุดรอการทำงานนี้ (มีกระแสไฟไหล) ได้มีการกำหนดให้มีชื่อว่า Mark ในทางกลับกัน เมื่อไม่มีกระแสไหล (เช่นในช่วงที่มีการส่งข้อมูลจริง) สถานะทางตรรกะได้ถูกกำหนดให้มีค่าเป็น 0 หรือสถานะ Space (Mark = logic "1" , Space = logic "0")

ตารางที่ 3.2 ตารางมาตราฐานของการใช้แรงดันไฟฟ้า RS-232

แรงดันไฟฟ้า	สถานภาพลอจิก	สถานภาพของสัญญาณ	ฟังก์ชั่นในการควบคุม
บวก	0	**Mark**	on
ลบ	1	**Space**	off

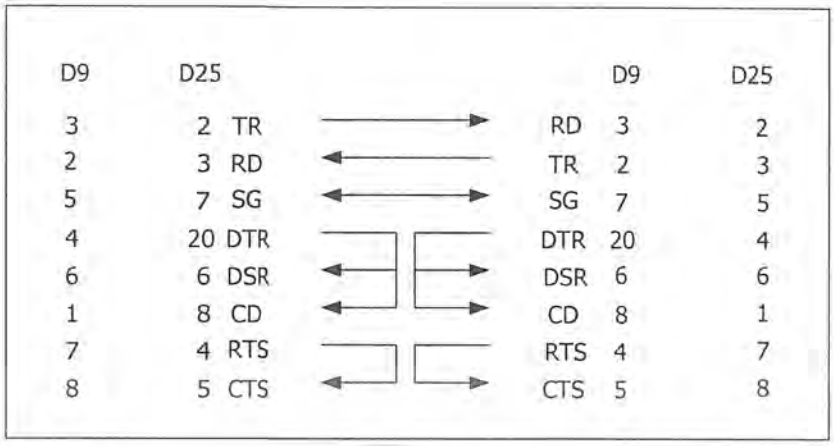
UART (Universal Asynchronous Receiver/Trasmitter) เป็นอุปกรณ์ที่มีความสำคัญมากในการสื่อสารแบบอนุกรม มีหน้าที่หลักอยู่ 3 อย่างที่สำคัญคือ

- Transmitter (ตัวส่งข้อมูล) ทำหน้าที่แปลงไบต์ข้อมูลขนาด 8 บิตไปเป็นอนุกรมของข้อมูล 8 บิต
- Receiver (ตัวรับข้อมูล) จะทำหน้าที่ตรงกันข้ามกับ Transmitter คือทำหน้าที่แปลงข้อมูลขนาด 8 บิต ไปเป็นไบต์ข้อมูล
- Controland Status มีหน้าที่ในการเฝ้าดูสถานะทางตรรกะของขาอินพุทต่างๆ และเมื่อโปรแกรมถูกเรียก ก็จะทำหน้าที่เปลี่ยนสถานะทางตรรกะของขาเอาต์พุต

3.5 ลักษณะการเชื่อมต่อและหน้าที่การทำงานของแต่ละขาที่สำคัญ

ในปัจจุบันพอร์ตอนุกรมนั้นจะมีอยู่ด้วยกัน 2 ขนาดคือคอนเน็คเตอร์แบบ D-type ตัวผู้ขนาด 25 Pin และ ตัวผู้ขนาด 9 Pin ซึ่งแสดงการเชื่อมต่อดังรูปที่ 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แสดงการเชื่อมต่อทั้งแบบ 25 Pin และ 9 Pin

หน้าที่การทำงานของแต่ละขา (ของชนิด 25 Pin) โดยพิจารณาด้าน DTE

- ขา 2 Transmitted Data (TD) ส่งข้อมูลจาก DTE ไป DCE
- ขา 3 Received Data (RD) ส่งข้อมูลจาก DCE ไป DTE
- ขา 4 Request to Send (RTS) เอาต์พุตอเนกประสงค์ สามารถนำไปประยุกต์ใช้ได้หลากหลาย ในโมเด็มแบบ half duplex ใช้สัญญาณนี้แสดงความต้องการส่งข้อมูล
- ขา 5 Clear to Send (CTS) อินพุตอเนกประสงค์ นำไปใช้งานได้หลากหลาย ในโมเด็มแบบ half duplex สัญญาณนี้ใช้อนุญาตให้ส่งข้อมูลได้
- ขา 6 Data Set Ready (DSR) อินพุตอเนกประสงค์ที่ใช้แจ้ง DTE ว่าอุปกรณ์ DCE มีไฟเลี้ยงและพร้อมที่จะทำงาน
- ขา 7 Signal Groun (SG) จุดอ้างอิงแรงดันสำหรับทุกสัญญาณในกระบวนการอินเฟส (ต้องมี)
- ขา 8 Data Carrier Detect (CD) สำหรับโมเด็มจะส่งสัญญาณ DCD เมื่อมันรับรู้การติดต่อกับโมเด็มที่อยู่ห่างออกไป สำหรับ DTE สัญญาณ DCD จะถูกนำไปใช้ในการยกเลิกการรับข้อมูล
- ขา 20 Data Terminal Ready (DTR) เอาต์พุตอเนกประสงค์โดยทั่วไปใช้เป็นสัญญาณบอก DCE ว่าอุปกรณ์ DTE ที่มันอินเตอร์เฟสด้วยมีไฟเลี้ยงและพร้อมที่จะทำงานด

นอกจาก 9 ขาที่กล่าวถึงข้างต้น ยังมีขาอื่นๆ อีกที่ใช้ในการอินเตอร์เฟส แต่สัญญาณสำคัญต่างๆ ที่มีการนำไปใช้เป็นประจำก็มักมาจาก 9 ขานี้เท่านั้น ซึ่งของขาคอนเน็กเตอร์ขา 2,3,4,5,6,7,8 และ 20 ว่ากลุ่ม " BIGEIGHT " ส่วนขาสัญญาณอื่นๆ มีไว้สำหรับเป็นทางเลือกที่ผู้ผลิต แต่ละรายจะนำไปประยุกต์ใช้ได้ตามความต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
วงจรรับข้อมูลจากพอร์ตอนุกรมและแหล่งจ่ายไฟ

วงจรข้อมูลอนุกรมจะทำหน้าที่รอรับข้อมูลอนุกรมหนึ่งไบต์ ซึ่งจะถูกส่งออกจากเซอร์ฟเวอร์ แล้วทำการแปลงเป็นข้อมูลขนานเพื่อส่งต่อไปอุปกรณ์ควบคุมชุดรีเลย์

วงจรกำหนดสัญญาณนาฬิกา

ใช้ผลึก Crystal ความถี่ 11.0592 MHz เป็นออสซิเลเตอร์

วงจรแปลงสัญญาณแบบ RS-232 ไปเป็น TTL

ใช้ไอซี Max 232 ซึ่งมีข้อดีคือมีทั้ง driver และ receiver ในตัวเองและใช้ โวลท์เตจระดับเดียวคือ 5 โวลท์ การนำเอา max 232 ไปใช้นั้นต้องมีการต่อตัวเก็บประจุเพิ่มเข้าไปอีกเล็กน้อย

วงจร latch ข้อมูล

ใช้ไอซี 74374 ทำหน้าที่เป็น ดี-ฟลิปฟลอป (D- FlipFlop) เพื่อคงค่าของข้อมูลขนาด 8 บิท ก่อนที่จะส่งให้กับวงจรรีเลย์

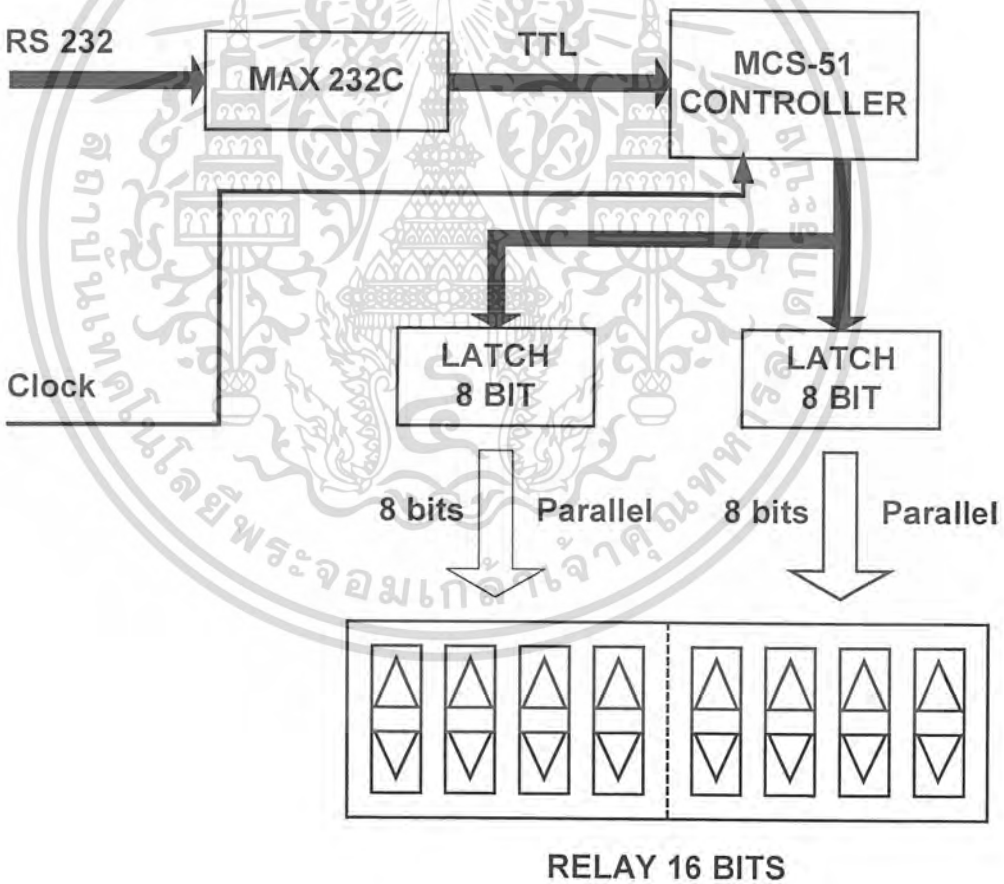
Microcontroller 8 bit At892051

At89C2051 เป็นไมโครคอนโทรลเลอร์ 8 บิท ขนาด 20 ขา สร้างด้วย Atmel's wigh Density nonvolatic memory technology โดยมีหน่วยความจำภายในขนาด 2kbyte ที่เรียกว่า อีพรอม (EPROM : Epasable and Programable Read only memory) ผลิตโดย Atmel ชุดคำสั่งและสถาปัตยกรรมภายในเหมือนไมโครคอนโทรลเลอร์ตระกูล MCS-51 ซึ่งผลิตโดย Intel โดยสามารถเขียนและลบได้ไม่น้อยกว่า 1000 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

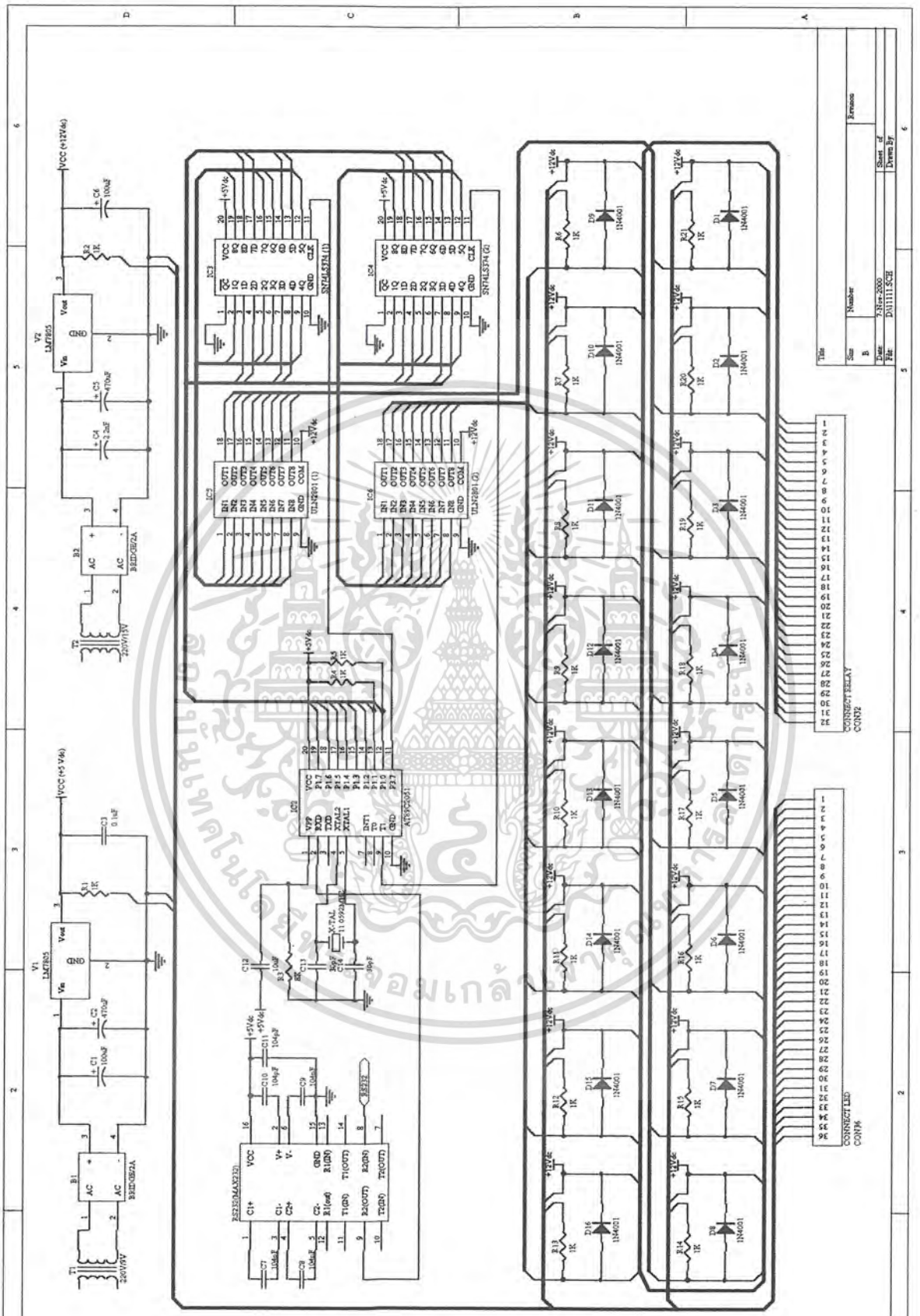
4.1 คุณสมบัติของ AT89C2051

1. สามารถใช้แทนไมโครคอนโทรลเลอร์ตระกูล MCS-51
2. หน่วยความจำโปรแกรมเป็นแบบ flash memory ขนาด 2Kbyte โดยสามารถเขียนหรือลบโปรแกรมได้ถึง 1000 ครั้ง
3. ใช้ไฟเลี้ยงตั้งแต่ 2.7 - 6.0 โวลท์
4. ทำงานได้ในช่วงความถี่ 0 Hz ถึง 24 MHz
5. สามารถล็อคโปรแกรมได้ 2 ระดับ
6. มีแรมภายใน 128* 8 บิท
7. มีอินเทอร์รัพท์ (Interupt) 5 ตำแหน่ง
8. มี Counter/Timer ขนาด 16 บิท 2 ตัว



รูปที่ 4.1 ส่วนประกอบของฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

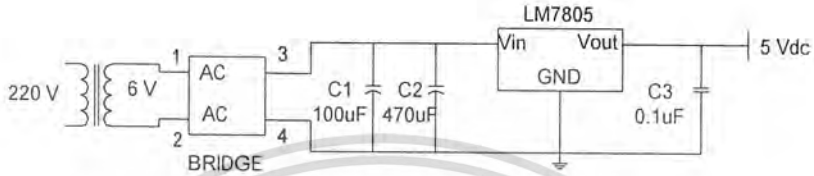


รูปที่ 4.2 แสดงวงจรไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 แหล่งจ่ายไฟ

จากหม้อแปลงกระแสสลับ 220V / 6V โดยใช้ไดโอดบริดจ์ , ไอซี่ เบอร์ 7805 และตัวเก็บประจุต่อร่วมกันเป็นวงจรควบคุมแรงดัน เพื่อจ่ายโวลท์เตจกระแสตรง 5 Vdc เป็นไฟเลี้ยงให้กับวงจรไมโครคอนโทรลเลอร์



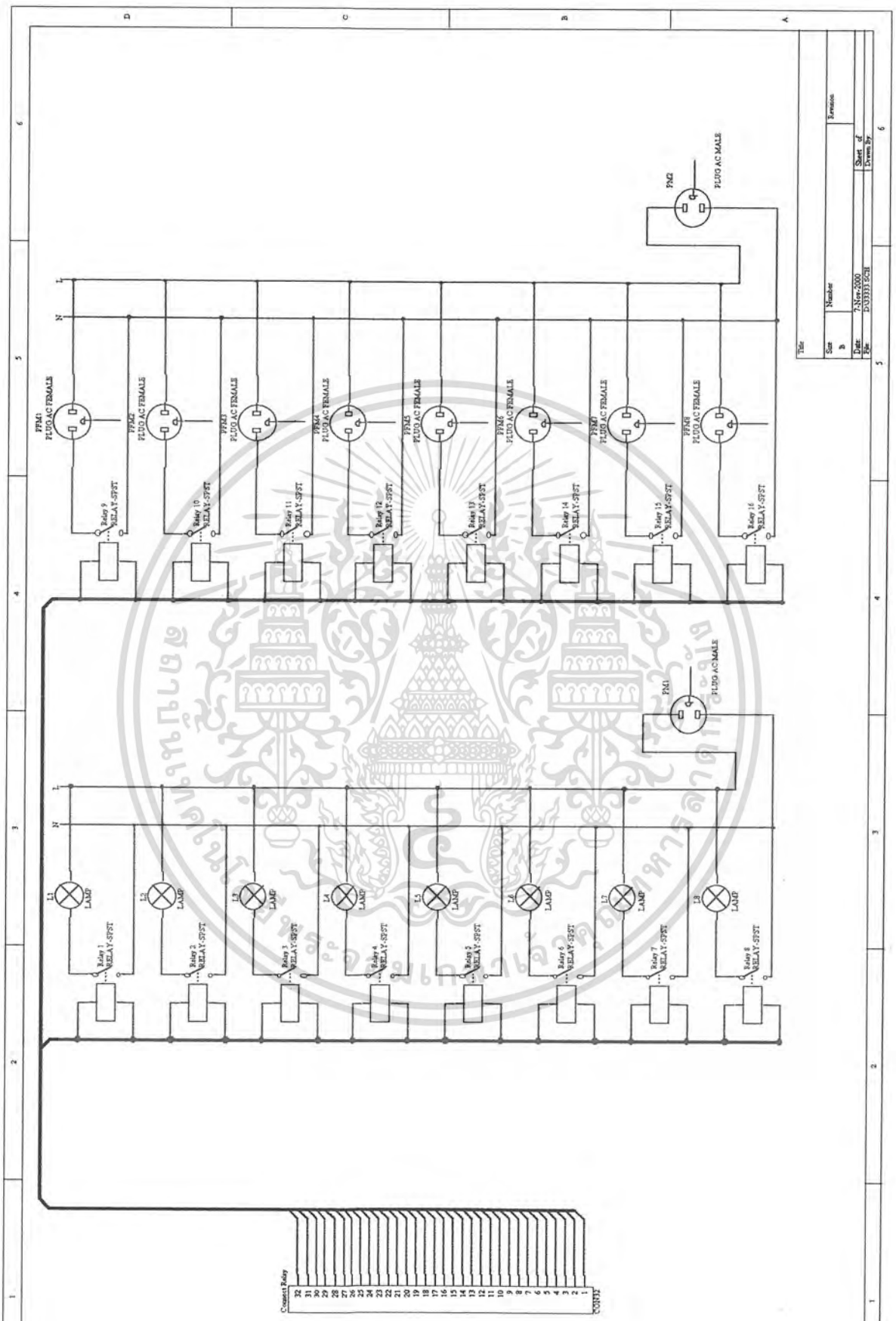
รูปที่ 4. 3 แหล่งจ่ายไฟวงจรไมโครคอนโทรลเลอร์

4.3 วงจรรีเลย์

เอาท์พุทจากไอซี ULN 2803 AN ทั้ง 2 ตัวจะทำงานเสมือนเป็นกราวด์ของสวิทซ์รีเลย์ เมื่อเอาท์พุทจาก latch บิทใดมีลอจิก 1 เพื่อผ่านคอนเวอร์เตอร์ของไอซี ULN 2803 AN จะเป็นศูนย์โวลท์จะมีกระแสไหล ผ่านขดลวดรีเลย์ ขดลวดของรีเลย์จะสร้างสนามแม่เหล็กดูดหน้าสัมผัสรีเลย์เข้าหากัน หน้าสัมผัสนี้จะถูกนำไปใช้เพื่อควบคุมการจ่ายกำลังให้กับเครื่องไฟฟ้าที่เราต้องการจะควบคุม

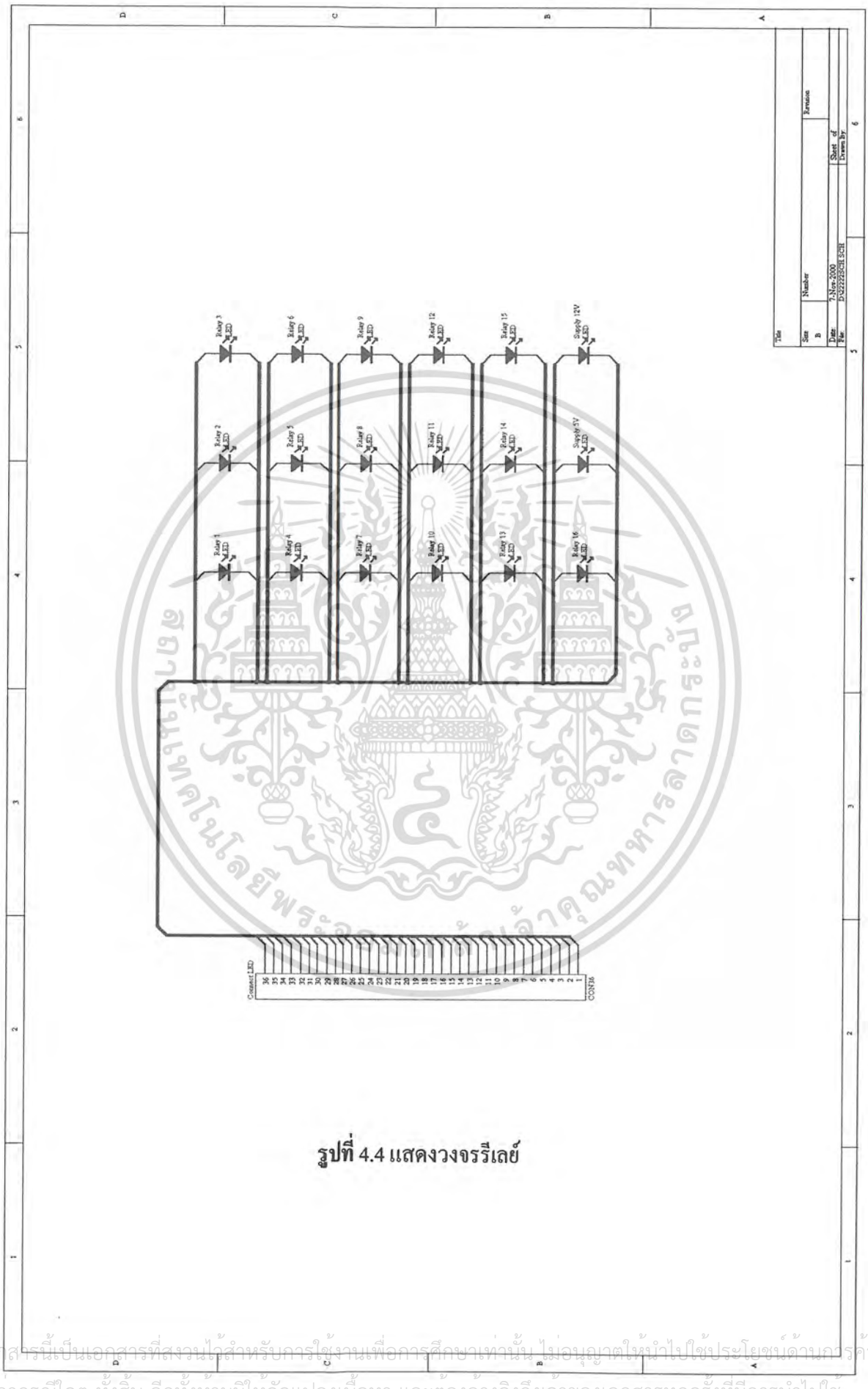
ให้อนุกรมหน้าสัมผัสเข้ากับแหล่งจ่ายแรงดัน 220 โวลท์กับหลอดไฟ และปลั๊กตัวเมียตามบอร์ดทดลอง ซึ่งสามารถควบคุมการเปิด-ปิด เครื่องไฟฟ้านี้ได้จากเทอร์มินัล ซึ่งอยู่ห่างออกไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงวงจรรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

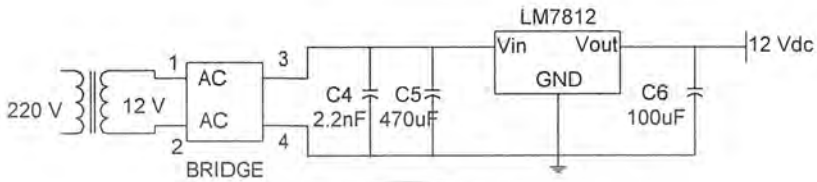


รูปที่ 4.4 แสดงวงจรรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 แหล่งจ่ายไฟสำหรับวงจรรีเลย์

จากหม้อแปลงกระแสสลับ 220 V / 12 V เราใช้ไดโอดบริดจ์ , ไอซี่เบอร์ 7812 และตัวเก็บประจุต่อร่วมกันเป็นวงจรควบคุมแรงดัน เพื่อจ่ายโวลท์เตจกระแสตรง 12 Vdc เป็นไฟเลี้ยงให้กับวงจรรีเลย์



รูปที่ 4.6 แหล่งจ่ายไฟวงจรรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
โปรแกรมควบคุม

ภาษาจาวา (JAVA) เป็นภาษาโปรแกรมซึ่งไม่ยึดติดกับระบบปฏิบัติการใด ๆ จาวา จึงไม่รู้จักพอร์ตอนุกรม Com2 ของดอส เราแก้ปัญหานี้ด้วยการอินเตอร์เฟสจาวาเข้ากับภาษาซี เป็นมาตรฐานซึ่งเข้าถึงพอร์ตอนุกรมได้

โครงงานนี้ จึงเลือกใช้ ภาษาซี เพื่อส่งข้อมูลไปควบคุมอุปกรณ์ไฟฟ้า โดยสร้างฟอร์มเพื่อทำการอ่านไฟล์ตัวอักขระ (text file) และจะนำข้อมูลในไฟล์ที่อ่านได้มาเก็บไว้ในตัวแปร แล้วส่งข้อมูลออกเอาท์พุทเป็นสัญญาณอนุกรม RS-232 ทางพอร์ตอนุกรม Com2 ซึ่งเป็นการส่งข้อมูลที่มีอัตราความเร็ว 9600 บิต/วินาที

โปรแกรมควบคุม

โปรแกรมควบคุมแบ่งออกเป็น 3 ส่วน ได้แก่

1. โปรแกรมเขียนข้อมูล 1 ไบต์ บนพอร์ตอนุกรม Com2
2. โปรแกรมบนเซอร์ฟเวอร์ คอยให้บริการรับส่งข้อมูลจากผู้ใช้ซึ่งอยู่ห่างออกไป
3. โปรแกรม JAVA Servlet จะทำหน้าที่เก็บข้อมูลสถานะรีเลย์จากผู้ใช้เพื่อส่งกับมายังเซอร์ฟเวอร์ ซึ่งต้องใช้ JDK1.1.6 (Java Developer's Toolkit version 1.1.6) ในการรันและคอมไพล์

Source code ที่พัฒนาจากภาษาจาวาจะมีชื่อนามสกุลเป็น .java ซึ่งในการนำไปใช้งานจะต้องคอมไพล์ด้วย JAVA Compiler (javac.exe) เสียก่อน ผลการคอมไพล์ได้ไฟล์ที่มีนามสกุลเป็น .class (ซึ่งเป็น Execute file)

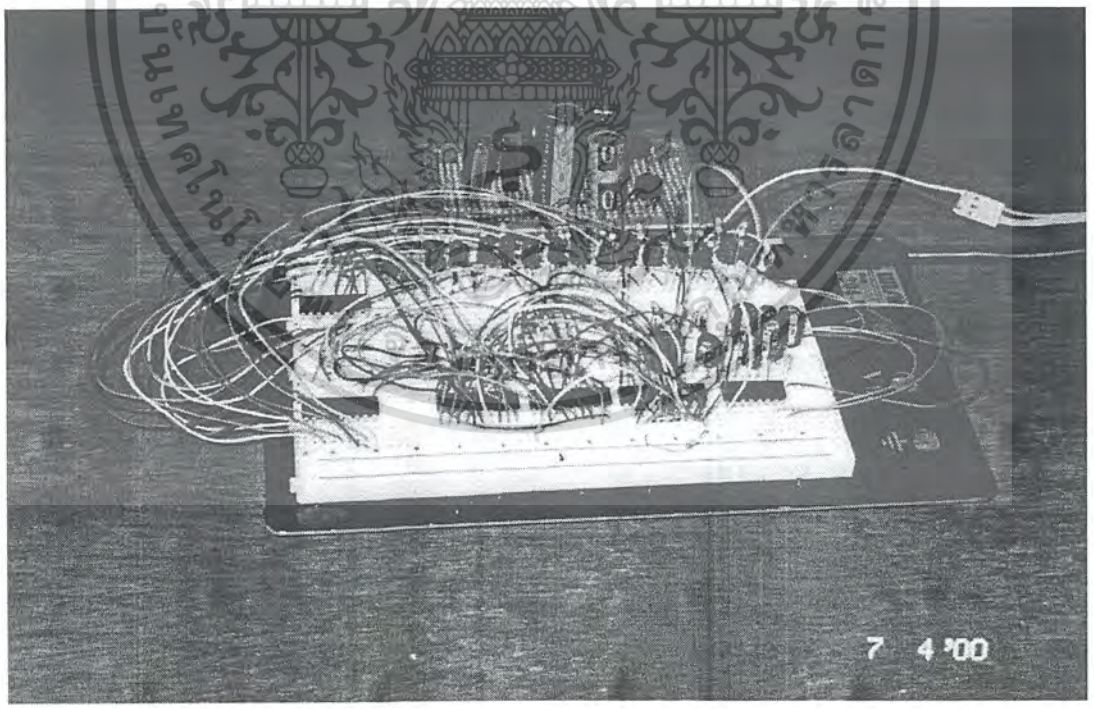
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1 โปรแกรมเขียนข้อมูลขนาด 1 ไบต์บนพอร์ตอนุกรม Com2

จาวาเป็นภาษาโปรแกรมที่ไม่ยึดติดกับระบบปฏิบัติการใดๆ จาวาจึงไม่รู้จักพอร์ตอนุกรม Com2 ของดอส จึงแก้ปัญหานี้ด้วยการอินเตอร์เฟสจาวาเข้ากับภาษาซี เป็นมาตรฐานซึ่งสามารถเข้าถึงพอร์ตอนุกรมได้ โดยทำการสร้างโปรแกรมที่ชื่อว่า ChkServlet (ChkServlet.class) เพื่อเป็นตัวรับส่งข้อมูลจากผู้ใช้ ซึ่งโปรแกรมนี้มีการอินเตอร์เฟสเข้ากับโปรแกรมควบคุมอุปกรณ์ที่พัฒนาจากภาษาซี โดยโปรแกรมควบคุมอุปกรณ์ที่พัฒนาจากภาษาซีนี้คือไฟล์ RS-232.C เป็นตัวควบคุมอุปกรณ์ที่แท้จริง

ในการทดสอบการทำงานของวงจรรับข้อมูลแบบอนุกรม มีการออกแบบวงจรรับข้อมูลโดยต่อ LED ที่เอาต์พุทของแลตช์ให้เอาต์พุทจากแลตช์ผ่าน LED ลง Ground การต่อแบบนี้ LED จะติดสว่าง เมื่อเอาต์พุทจากแลตช์มีค่าเป็นหนึ่ง หลังจากประกอบวงจรเสร็จเรียบร้อยแล้วนำมาทดสอบกับโปรแกรม ChkServlet.class ได้

จากผลเปรียบเทียบข้อมูลที่ส่งออกไปทดสอบวงจรรับข้อมูลได้ผลตรงกันหรือไม่ถ้าได้ผลตรงกันเป็นอันว่าส่งข้อมูลผ่านพอร์ตอนุกรม Com2 ได้ ซึ่งจากการทดสอบพบว่าผลของข้อมูลที่ส่งออกและวงจรที่รับข้อมูลมีผลตรงกัน



รูปที่ 5.1 แสดงวงจรทดสอบการรับข้อมูลผ่านพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 โปรแกรมบนเซอร์ฟเวอร์

โปรแกรมที่ทำหน้าที่ให้บริการบนเซอร์ฟเวอร์มี 2 ส่วนคือ

1. โปรแกรมเว็บเซอร์ฟเวอร์ ทำหน้าที่ให้บริการด้าน Web ต่างๆ
2. โปรแกรม Servletrunner เป็นตัวสตาร์ท Web Server Thread ให้รอรับ Package ซึ่งผู้ใช้ส่งมา

ในการทำงานของเซอร์ฟเวอร์ จะต้องรันโปรแกรม Webserver และ Servletrunner ค้างทิ้งไว้ที่เซอร์ฟเวอร์

5.3 โปรแกรม Java Servlet

หลังจากใช้โปรแกรมที่พัฒนาด้วยภาษาซี (RS232.C) ทำการควบคุมรีเลย์โดยเครื่องตัวมันเองได้แล้ว ในการใช้คุณสมบัติของJava Servlet จุดประสงค์เพื่อช่วยให้ความสะดวกสบายแก่ผู้ใช้งาน และสามารถใช้งานได้จากทั่วทุกมุมโลก โดยผ่านเบราเซอร์ทั่วไปได้ในการควบคุมรีเลย์ผ่านทางอินเตอร์เน็ต ซึ่งจะต้องใช้โปรแกรมที่พัฒนาด้วยภาษาจาวา เป็นตัวรับส่งข้อมูลผ่านเครือข่ายอินเตอร์เน็ต

5.4 รายชื่อไฟล์สำคัญ

1. Index.html เป็นไฟล์ html ซึ่งเป็น Home page ของระบบควบคุมอุปกรณ์ไฟฟ้าผ่านอินเตอร์เน็ต
2. ChkServlet.java เป็นโปรแกรมที่พัฒนาขึ้นมาจากภาษาจาวา ใช้ตรวจสอบรหัสผ่านผู้ใช้งานและรับส่งข้อมูลในการควบคุมอุปกรณ์รีเลย์ ในรูปแบบของฟอร์ม
3. ChkServlet.class เป็นโปรแกรมที่เกิดจากการ Compile โปรแกรม ChkServlet.java เพื่อใช้รับส่งข้อมูลบน HTTP Server หรือผ่าน Internet นั่นเอง
4. InfoServlet.java เป็นโปรแกรมที่พัฒนาขึ้นมาจากภาษาจาวา ใช้รับส่งข้อมูลในการควบคุมอุปกรณ์ไฟฟ้า และเป็นตัวบอกสถานะการควบคุมอุปกรณ์ไฟฟ้า
5. InfoServlet.class เป็นโปรแกรมที่เกิดจากการ Compile โปรแกรม InfoServlet.java เพื่อใช้รับส่งข้อมูลบน HTTP Server หรือผ่าน Internet
6. RS232.c เป็นโปรแกรมที่พัฒนาขึ้นมาจากภาษาซี ใช้รับส่งข้อมูลผ่านพอร์ตอนุกรม
7. RS232.exe เป็นโปรแกรมที่เกิดจากการ Compile โปรแกรม RS232.c เพื่อรับส่งข้อมูลที่อินเตอร์เฟสกับ InfoServlet.class ผ่านพอร์ตอนุกรม ควบคุมรีเลย์
8. IC.hex เป็นโปรแกรมที่พัฒนาจากภาษาแอสแซมบี้ สำหรับ IC เบอร์ AT89C2051

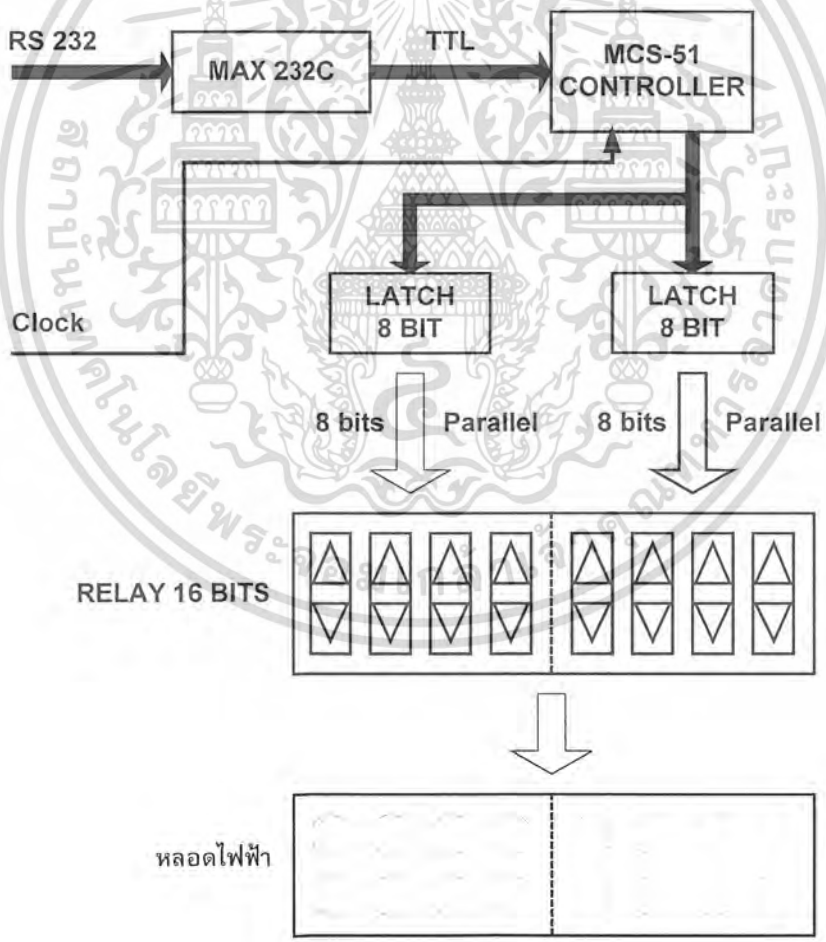
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การติดตั้งและการใช้งาน

หลังจากประกอบวงจรรับส่งข้อมูล, วงจรรีเลย์, แหล่งจ่ายไฟและคอนเน็คเตอร์ เข้าด้วยกันเรียบร้อยแล้ว ให้ต่อคอนเน็คเตอร์เข้ากับพอร์ตอนุกรม COM 2

รันโปรแกรม Web Server (Sambar) ตามปกติ เพื่อคอยโหลด Webpages ให้ผู้ใช้ที่ติดต่อเข้ามา รันโปรแกมจาวาที่ชื่อ Servletrunner โดยโปรแกรม Servletrunner จะเป็นตัวสตาร์ท Web Server Thread ให้รอรับ Package ซึ่งผู้ใช้ส่งมา

นำเครื่องไฟฟ้าที่ต้องการควบคุมระยะไกล เสียบเข้ากับปลั๊ก และเปิดสวิทช์เครื่องไฟฟ้านั้นไว้ด้วย หรือหลอดไฟฟ้าที่ต่อตามบอร์ด ดังรูป ซึ่งต่ออยู่กับวงจรรีเลย์

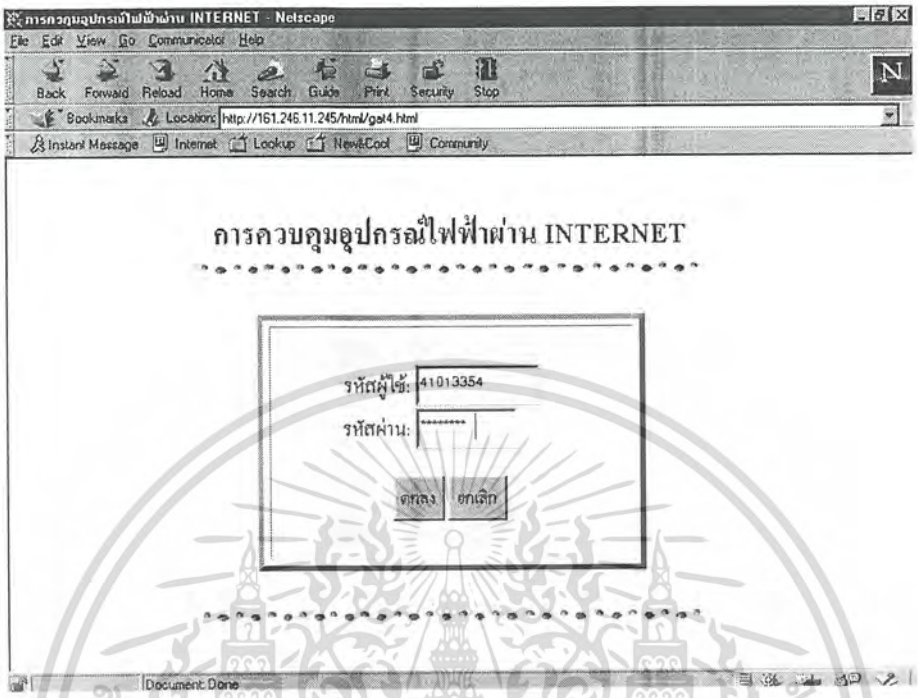


รูปที่ 6.1 แสดงการส่งสัญญาณควบคุมอุปกรณ์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

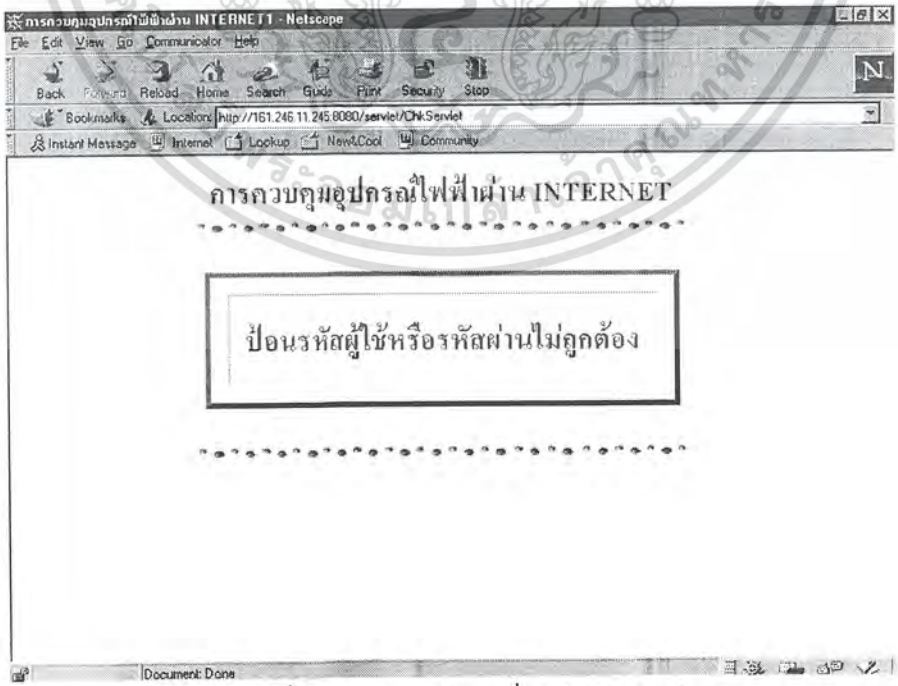
6.1 การใช้งานระบบควบคุมอุปกรณ์ไฟฟ้าผ่านอินเตอร์เน็ต

รันโปรแกรม Web Browser แล้วเรียกที่ http://161.246.11.245/index.html ดังรูปที่ 6.2



รูปที่ 6.2 แสดงหน้าจอหลักระบบควบคุมอุปกรณ์ไฟฟ้าผ่าน Internet

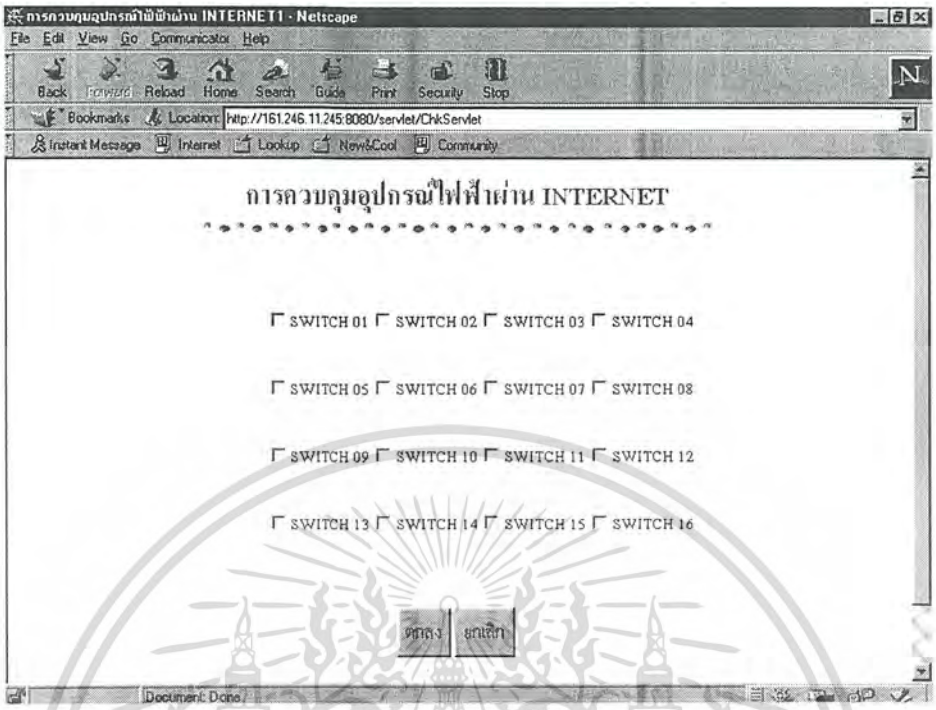
ผู้ใช้ต้องใส่ รหัสผู้ใช้ และรหัสผ่านให้ถูกต้องก่อน ในโครงงานนี้จะใช้รหัสผ่านเป็นระบบรักษาความปลอดภัย ซึ่งถ้าไม่ถูกต้องหน้าจอจะแสดงผลดังรูปที่ 6.3



รูปที่ 6.3 แสดงหน้าจอเมื่อใส่รหัสผ่านผิดพลาด

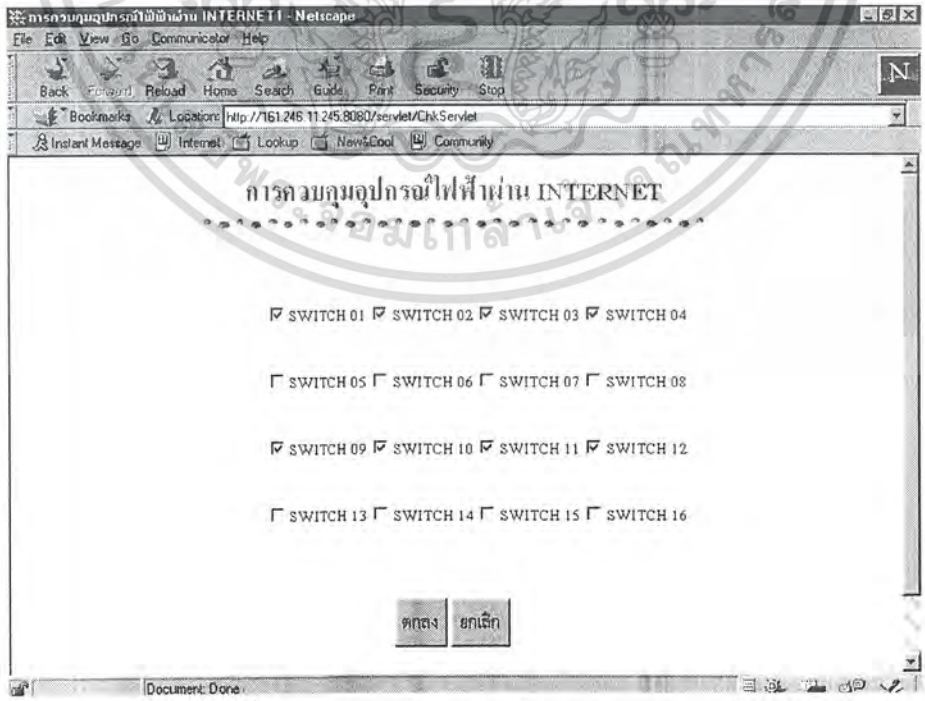
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อใส่รหัสถูกต้องที่หน้าจอผู้ใช้จะแสดงดังรูปที่ 6.4



รูปที่ 6.4 แสดงหน้าจอระบบควบคุมอุปกรณ์ไฟฟ้าผ่าน Internet

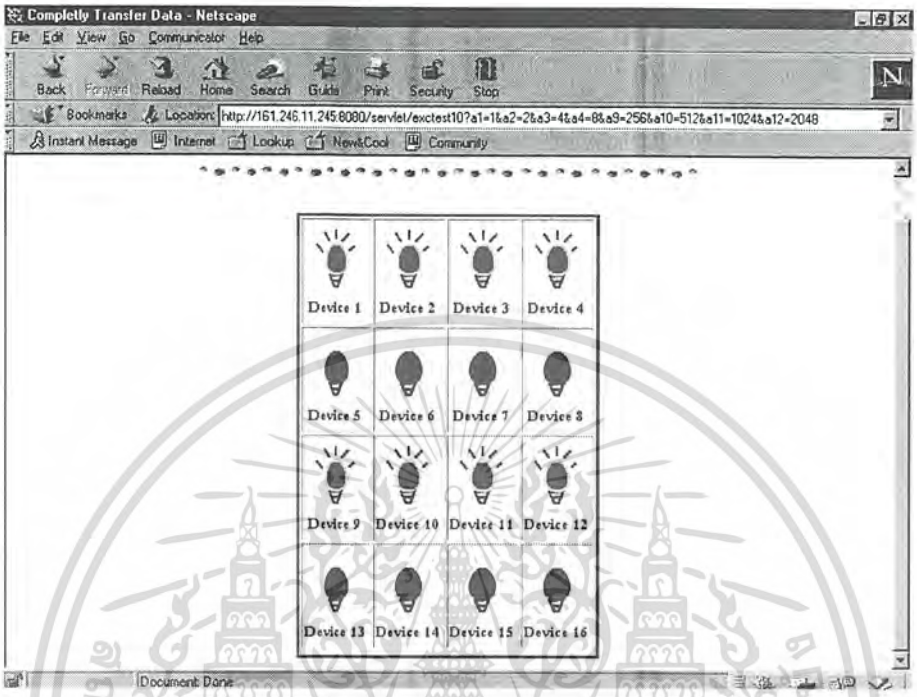
ผู้ใช้สามารถควบคุมอุปกรณ์ไฟฟ้าได้ด้วยการกำหนดสถานะใน Checkbox ทั้ง 16 ช่องได้



รูปที่ 6.5 แสดงหน้าจอการเลือกควบคุมอุปกรณ์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้เลือกควบคุมอุปกรณ์ไฟฟ้าที่ต้องการแล้ว ถ้าผู้ใช้กดปุ่ม "ยกเลิก" จะลบเครื่องหมายที่เลือกไว้ทั้งหมด แต่ถ้ากดปุ่ม "ตกลง" จะทำการส่งค่าที่ผู้ใช้เลือกไปควบคุมรีเลย์โดยผ่านพอร์ตอนุกรม ซึ่งจะแสดงสถานะของการควบคุมอุปกรณ์ไฟฟ้าดังรูปที่ 6.6



รูปที่ 6.6 แสดงสถานะการควบคุมอุปกรณ์ไฟฟ้าแก่ผู้ใช้งาน

6.2 ผลการทดสอบ

จากผลการทดสอบของการควบคุมอุปกรณ์ไฟฟ้าผ่านเครือข่าย Internet กำหนดการทดสอบใช้การสั่งควบคุมเปิดและปิดไฟฟ้าดวงที่ 1 ซึ่งส่งข้อมูลออกไปทางพอร์ตอนุกรมไปยังรีเลย์ซึ่งควบคุมเครื่องไฟฟ้าเป็นจำนวน 100 ครั้ง แล้วทำการตรวจสอบความถูกต้องของข้อมูลที่รับได้จากการตรวจสอบผลที่รับข้อมูลได้ตรงกันทั้ง 100 ครั้ง จากนั้นได้ทำการทดลองในกรณีเดียวกันนี้จนครบ 16 ดวง ซึ่งผลการทดสอบตรงกันทุกครั้งจากผลการทดสอบควบคุมการเปิดปิดไฟฟ้าทีละดวงนี้เป็นผลที่น่าพอใจ และยังได้ทำการทดสอบในการควบคุมเปิดปิดไฟฟ้าทีละ 4 ดวงเราได้สุ่มการทำงาน 100 ครั้งสลับกันไปในจำนวน 16 ดวง ซึ่งผลการตรวจสอบความถูกต้องของข้อมูลที่รับได้ปรากฏว่าถูกต้องถึง 98 ครั้ง จากทั้งหมด 100 ครั้ง มีผิดพลาดประมาณ 2 % ซึ่งการผิดพลาดใน 2 % นี้ ทางผู้จัดทำได้หาสาเหตุแล้วพบว่าเกิดจากการทำงานของรีเลย์ผิดพลาดเมื่อทำการรับข้อมูลรีเลย์จะดึงหน้าสัมผัสมาต่อวงจรจากนั้นมันจะดีดหน้าสัมผัสออกซึ่งเกิดขึ้นเร็วมาก จึงทำให้เกิดความผิดพลาดขึ้น จากการทดสอบการควบคุมอุปกรณ์ไฟฟ้าผ่านอินเตอร์เนตนี้ ได้ผลเป็นที่น่าพอใจสามารถทำงานได้ดีตรงตามวัตถุประสงค์ของโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
สรุปผลการทดลองและข้อเสนอแนะ

7.1 สรุปผลการทดลอง

จากการออกแบบและใช้งานระบบควบคุมอุปกรณ์ไฟฟ้าผ่านเครือข่ายอินเตอร์เน็ต พบว่าสามารถควบคุมการเปิด / ปิด อุปกรณ์ไฟฟ้าที่อยู่ในระยะไกลได้ดี

อุปกรณ์ไฟฟ้าที่เลือกใช้ในโครงงานนี้คือ

1. หลอดไฟฟ้าที่ใช้ตามบ้านทั่วไป ที่ใช้ไฟ 220 โวลท์
2. เครื่องใช้ไฟฟ้าที่ใช้ตามบ้านทั่วไป ที่ใช้ไฟ 220 โวลท์ เช่น พัดลม วิทยุ

ปัญหาที่เกิดขึ้นและแนวทางแก้ไข

1. การปิด / เปิดอุปกรณ์ไฟฟ้า ทั้ง 16 ดวง ที่มีความผิดพลาด 2 % เกิดจากอุปกรณ์รีเลย์ ที่มีคุณสมบัติต่ำ ถ้าต้องการไม่ให้เกิดความผิดพลาดเลย ควรใช้อุปกรณ์รีเลย์ที่มีคุณสมบัติดีกว่านี้ แต่จะมีราคาแพง
2. ถ้าอุปกรณ์ปลายทางที่สั่งงานไม่ทำงาน ผู้ใช้ไม่สามารถรับรู้ได้ วิธีแก้ไขคือ ต้องออกแบบให้มีกล้องจับภาพติดไว้ที่ปลายทาง และส่งภาพกลับมายังผู้ควบคุม โดยแสดงผลบนหน้าจอ

7.2 ข้อเสนอแนะ

1. จากโครงงานนี้เป็นการควบคุมอุปกรณ์ไฟฟ้าเพียง 16 จุด ซึ่งสามารถที่ออกแบบให้ควบคุมได้ถึง 32 จุด โดยการต่ออุปกรณ์เพิ่มที่พอร์ตของ IC - AT89C2051 ที่เหลืออีก 2 พอร์ตได้
2. จากโครงงานนี้เป็นการควบคุมอุปกรณ์ไฟฟ้าผ่านเครือข่ายอินเตอร์เน็ต ที่ใช้คอมพิวเตอร์เป็นตัวสั่งการควบคุม แต่เนื่องจากปัจจุบัน เทคโนโลยีทางด้าน Mobile ได้พัฒนาก้าวหน้าไปเป็นอันมาก สามารถรับส่งข้อมูลต่างๆ ผ่านทาง mobile เช่น โทรศัพท์มือถือ ได้แล้ว ดังนั้นโครงงานนี้จึงเป็นแนวทางเพื่อให้มีการพัฒนาการควบคุมผ่านทางระบบ Mobile ต่อไป
3. นอกจากนี้ยังสามารถเขียนโปรแกรมตั้งเวลาการเปิด/ปิดควบคุมอุปกรณ์ไฟฟ้า ซึ่งสามารถจะทำให้ตั้งเวลาการเปิด/ปิดอุปกรณ์ไฟฟ้าได้อีกทางหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

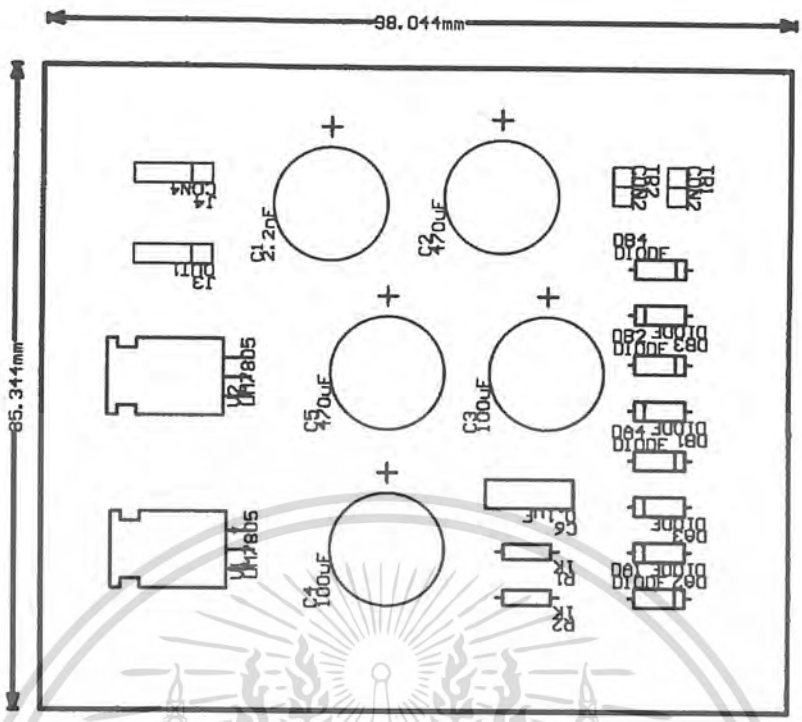
บรรณานุกรม

1. วิทยา เรืองพรวิสุทธิ์ , "คู่มือโปรแกรมภาษาซี" ,บ.ซีเอ็ดยูเคชัน จำกัด, 2539
2. ดร.อิน มาน หยาง , ทศพล ปราชญสมพงษ์ , "การออกแบบระบบไมโครคอนโทรลเลอร์ ตระกูล 8051" ศูนย์หนังสือมหาวิทยาลัยรังสิต, 2539
3. รศ.สมยศ จุณณะปิยะ , "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51" คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2541
4. JavaServer Product Group , "The Java Servlet AP" Website
   http://jserv.javasoft.com/products/java-erver/documentation/webserver1.0.2/servlets/api.html

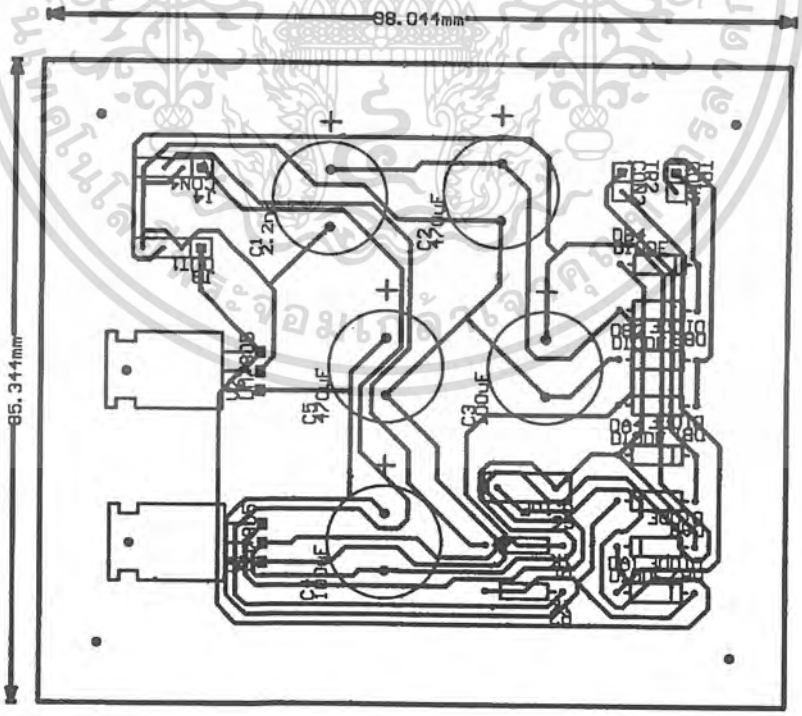
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

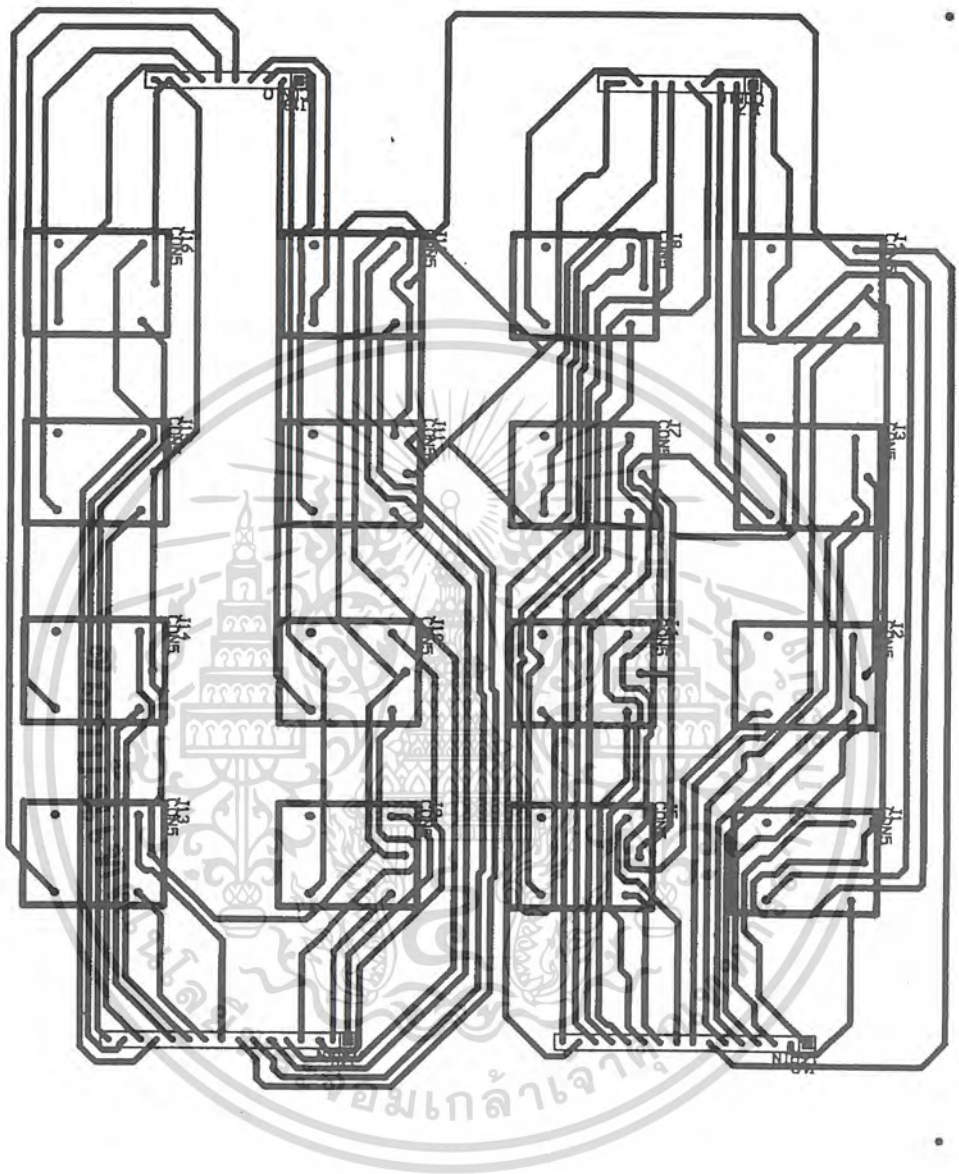


รูปที่ 1 แสดงสัญลักษณ์ตัวอุปกรณ์



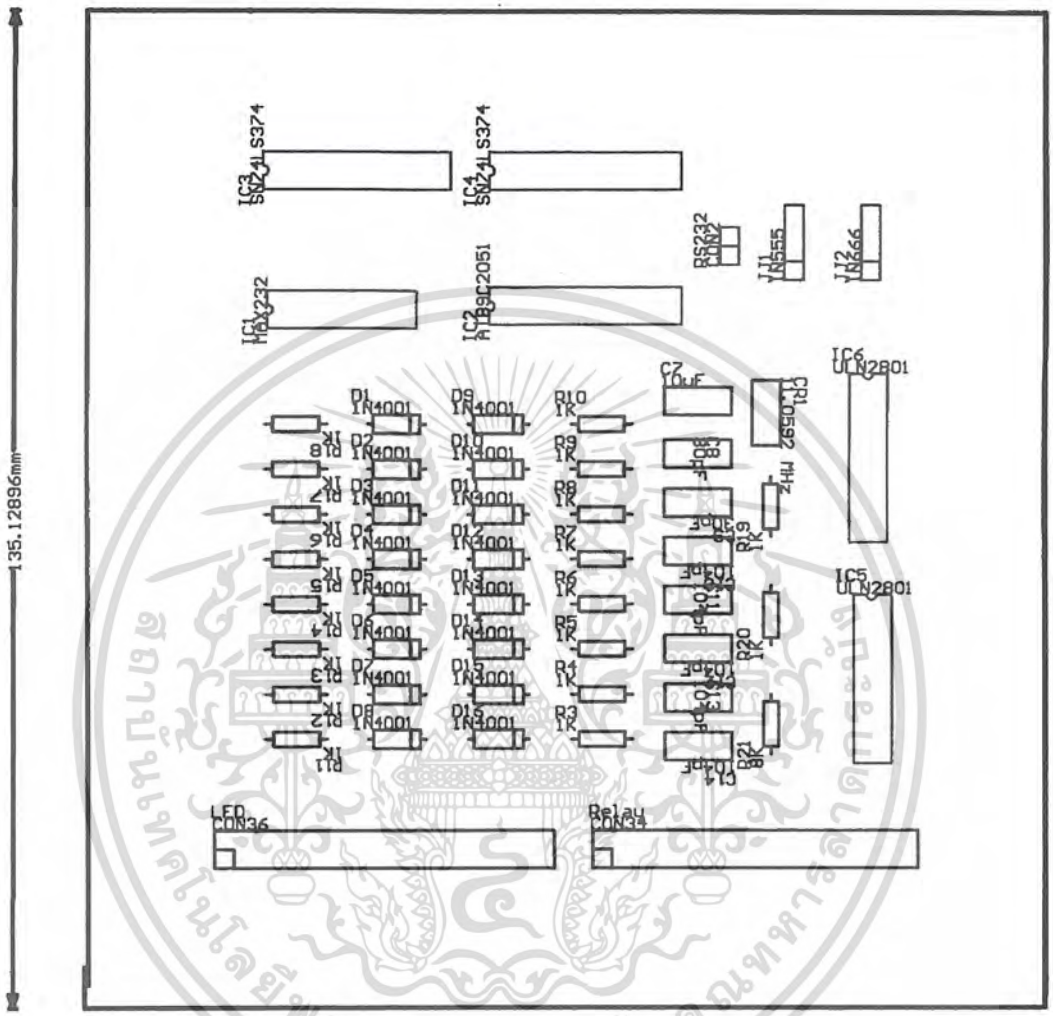
รูปที่ 2 แสดงลายวงจร Supply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



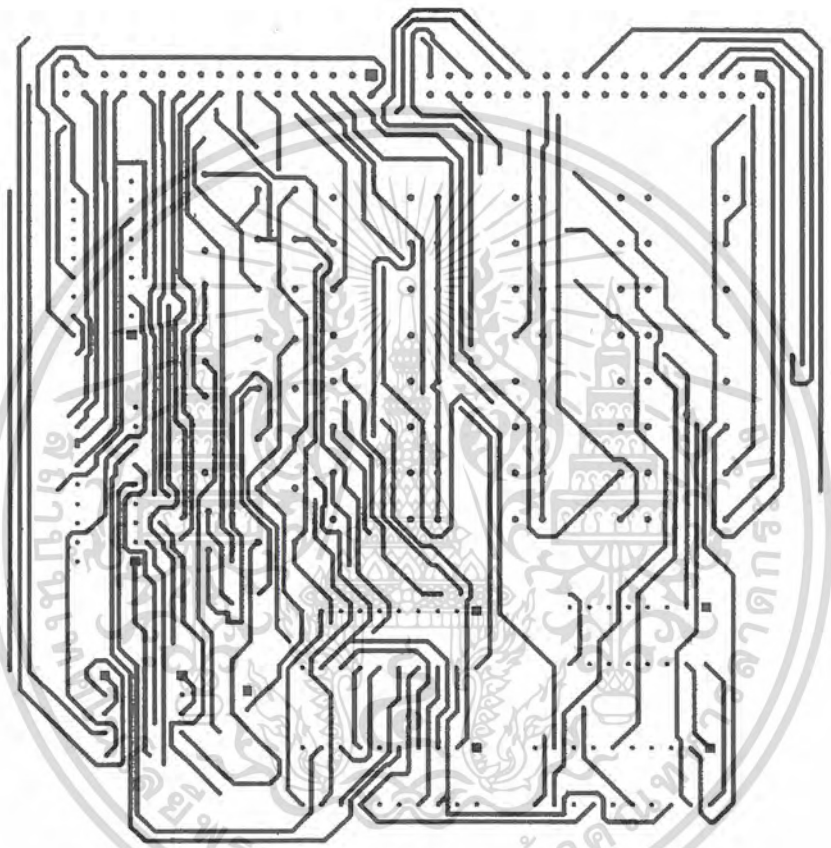
รูปที่ 3 แสดงลายวงจรรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



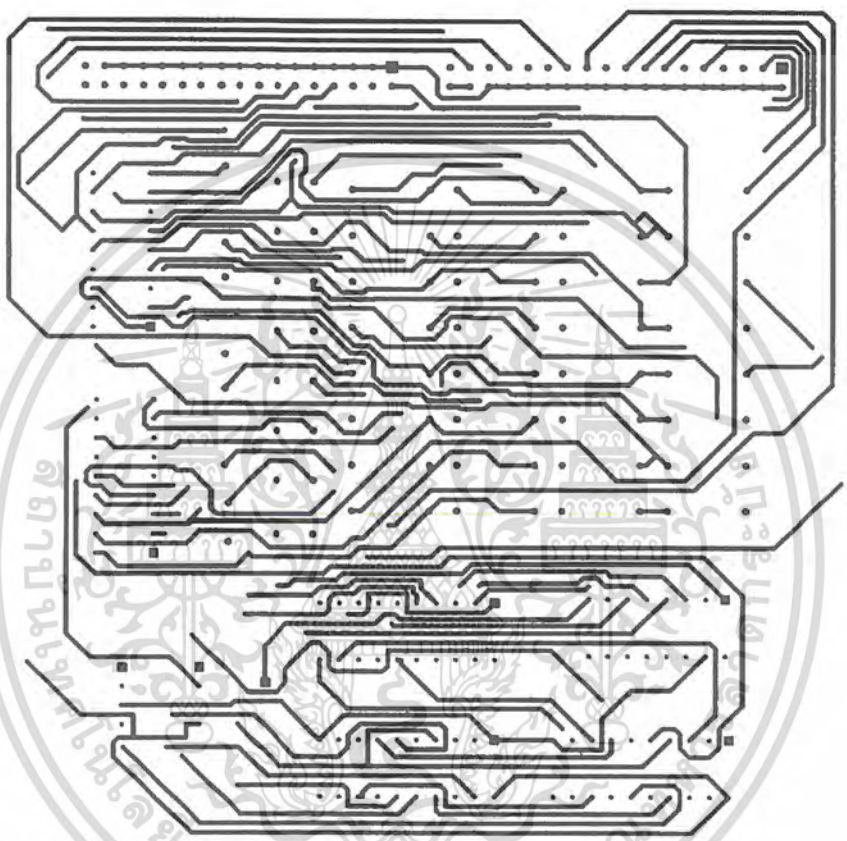
รูปที่ 4 แสดงสัญลักษณ์อุปกรณ์วงจรไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5 แสดงลายวงจรไมโครคอนโทรลเลอร์(ด้านบน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6 แสดงลายวงจรไมโครคอนโทรลเลอร์(ด้านล่าง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Index.html

```
<html>
<head><br>
<title>การควบคุมอุปกรณ์ไฟฟ้าผ่าน INTERNET</title>
<meta name="generator" content="Arachnophilia 3.9">
<meta name="formatter" content="Arachnophilia 3.9">
<meta name="author" content="s1013354( Sirichai Tammarugwattana )">
</head>

<body background="img/bg-wht2.gif" bgcolor="#ffffff" text="#000000" link="#bd0062"
vlink="#bd0062" alink="#ff0000">

<center>

<font size=5 color=#df0000><b>การควบคุมอุปกรณ์ไฟฟ้าผ่าน INTERNET</b></font><br>
<img src="img/line53.gif" width="447" height="14" border=0 alt="Line Break"><p>

<table border=5 bordercolor=#f00040 cellpadding=10 cellspacing=5>
<tr>  <td align=center>
      <table border=0 width=300 bgcolor=#fff0af cellpadding=0 cellspacing=0>
      <form  name="form1"  action="http://161.246.11.245:8080/servlet/ChkServlet"
method="post">
      <tr>  <td align=center colspan=2><font size=1> </font></td></tr>
      <tr>  <td width=120 align="right">รหัสผู้ใช้: </td>
            <td width=180 align="left"><input type="text" name="id" size=8 maxlength=8>
</td></tr>
      <tr>  <td width=120 align="right">รหัสผ่าน: </td>
            <td width=180 align="left"><input type="password" name="pass" size=6></td>
</tr>
      <tr>  <td align=center colspan=2><font size=1> </font></td></tr>
      <tr>  <td align=center colspan=2><font size=2><input type="submit" value="ตกลง">
<input type="reset" value="ยกเลิก"></td></tr>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        <tr>    <td align=center colspan=2><font size=1> </font></td></tr>
        </form></table>
</td></tr>
</table><br>
<img src="img/line53.gif" width="447" height="14" border=0 alt="Line Break"><br>
</center>
</body>
</html>
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ChkServlet.java

```
/*  ChkServlet.java
Project */
import java.io.*;
import java.util.*;
import java.math.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import oracle.jdbc.driver.*;

public class ChkServlet extends HttpServlet
{
   public void service(HttpServletRequest req, HttpServletResponse res) throws IOException
   {
      Enumeration keys;
      String key, reg_typeCheck;
      String value;
          PrintWriter out = new PrintWriter(res.getOutputStream());
      String line = null;
      res.setContentType("text/html");
           preamble(out, "CRSC Project ");
           out.println("<p>");
           String stuid = req.getParameter("id");
           String passwd = req.getParameter("pass");
           String user = "41013354";
           String pwd = "sirichai";
           boolean succeed = false;
     if ((stuid.equals(user)) &&(passwd.equals(pwd))){
                preamble3(out, "CRSC Project ");
                readyfail(out);
                postamble(out);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            return;
        }
    else{
                preamble3(out, "CRSC Project ");
                hoho(out);
                postamble(out);
            return;
                    }
    }
    public String getServletInfo() {
            return " CRSC Project ;) ";
    }
    private void preamble(PrintWriter out, String s) {
       out.println("<HTML>");
       out.println("<HEAD><TITLE>การควบคุมอุปกรณ์ไฟฟ้าผ่าน INTERNET1</TITLE>");
       out.println("<BASEFONT SIZE=3>");
    }

 /*   private void preamble2(PrintWriter out, String s) {
       out.println("</HEAD> ");
      out.println(" <BODY BACKGROUND=\"http://161.246.11.245/html/img/bg-wht2.gif\"
BGCOLOR=#ffffff TEXT=#000000 LINK=#0080ff VLINK=#8080ff ALINK=#ff0000
onLoad=\"init()\"> ");
       out.println("<BLOCKQUOTE> ");
       out.println("<CENTER> ");
       out.println("<font size=5 color=#0080ff><b>การควบคุมอุปกรไฟฟ้าผ่าน INTERNET</b>
</font><br> ");
      out.println("<img src=\"http://161.246.11.245/html/img/line53.gif\" width=\"447\"
height=\"14\" border=0 alt=\"Line Break\"><p> ");
    }         */

    private void preamble3(PrintWriter out, String s) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        out.println("</HEAD> ");
        out.println(" <BODY BACKGROUND=\"http://161.246.11.245/html/img/bg-wht2.gif \"
BGCOLOR=#ffffff TEXT=#000000 LINK=#0080ff VLINK=#8080ff ALINK=#ff0000 > ");
        out.println("<BLOCKQUOTE> ");
        out.println("<CENTER> ");
        out.println("<font size=5 color=#0080ff><b>การควบคุมอุปกรณ์ไฟฟ้าผ่าน INTERNET</b>
</font><br> ");
       out.println("<img src=\"http://161.246.11.245/html/img/line53.gif\" width=\"447\"
height=\"14\" border=0 alt=\"Line Break\"><p> ");
    }

    private void postamble(PrintWriter out) {
        out.println("<img src=\"http://161.246.11.245/html/img/line53.gif\" width=\"447\"
height=\"14\" border=0 alt=\"Line Break\"><br>");
        out.println("</BLOCKQUOTE>");
        out.println("</CENTER>");
        out.println("</BODY>");
        out.println("</HTML> ");
            out.flush();
    }

    private void readyfail(PrintWriter out) {
                    out.println("<FORM
NAME=\"Form2\"ACTION=\"http://161.246.11.245:8080/servlet/exctest10\"METHOD=\"get\">
");
              out.println("<CENTER><BR>");
                  out.println("<UL>");
                  out.println("<INPUT TYPE=\"CHECKBOX\" NAME=\"a1\" VALUE=\"1\">
SWITCH 01");
                  out.println("<INPUT TYPE=\"CHECKBOX\" NAME=\"a2\" VALUE=\"2\">
SWITCH 02");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                out.println("<INPUT TYPE=\"CHECKBOX\" NAME=\"a3\" VALUE=\"4\">
SWITCH 03");
                out.println("<INPUT TYPE=\"CHECKBOX\" NAME=\"a4\" VALUE=\"8\">
SWITCH 04");
                    out.println("</UL>");
                    out.println("<UL>");
                    out.println("<INPUT TYPE=\"CHECKBOX\" NAME=\"a5\" VALUE=\"16\">
SWITCH 05");
                    out.println("<INPUT TYPE=\"CHECKBOX\" NAME=\"a6\" VALUE=\"32\">
SWITCH 06");
                    out.println("<INPUT TYPE=\"CHECKBOX\" NAME=\"a7\" VALUE=\"64\">
SWITCH 07");
                    out.println("<INPUT TYPE=\"CHECKBOX\" NAME=\"a8\" VALUE=\"128\">
SWITCH 08");
                    out.println("</UL>");
                    out.println("<UL>");
                    out.println("<INPUT TYPE=\"CHECKBOX\" NAME=\"a9\" VALUE=\"256\">
SWITCH 09");
                    out.println("<INPUT TYPE=\"CHECKBOX\" NAME=\"a10\"
VALUE=\"512\">SWITCH 10");
                    out.println("<INPUT TYPE=\"CHECKBOX\" NAME=\"a11\"
VALUE=\"1024\">SWITCH 11");
                    out.println("<INPUT TYPE=\"CHECKBOX\" NAME=\"a12\"
VALUE=\"2048\">SWITCH 12");
                    out.println("</UL>");
                    out.println("<UL>");
                    out.println("<INPUT TYPE=\"CHECKBOX\" NAME=\"a13\"
VALUE=\"4096\">SWITCH 13");
                    out.println("<INPUT TYPE=\"CHECKBOX\" NAME=\"a14\"
VALUE=\"8192\">SWITCH 14");
                    out.println("<INPUT TYPE=\"CHECKBOX\" NAME=\"a15\"
VALUE=\"16384\">SWITCH 15");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                out.println("<INPUT TYPE=\"CHECKBOX\" NAME=\"a16\"
VALUE=\"32768\">SWITCH 16");
                out.println("</UL>");
                out.println("<br>");
                out.println("<INPUT TYPE=\"SUBMIT\" VALUE=\"ตกลง\">");
                out.println("<INPUT TYPE=\"RESET\" VALUE=\"ยกเลิก\">");
                out.println("</CENTER>");
                out.println("</FORM>");
                }

   private void  hoho(PrintWriter out) {
            out.println("<table border=5 cellpadding=15 cellspacing=15 bgcolor=#ffffe0
bordercolor=#0000ff> ");
            out.println("<tr><td align=center><font color=#ff0000 size=5><b>ป้อนรหัสผู้ใช้หรือ
รหัสผ่านไม่ถูกต้อง</b></font></td></tr> ");
            out.println("</table> <br>");
   }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

InfoServlet.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.lang.*;
import java.awt.*;
import java.io.*;

public class exctest10 extends HttpServlet
{
   public void doGet (HttpServletRequest request,HttpServletResponse response) throws
ServletException, IOException
   {
         PrintWriter   out;
         String    title = "Completly Transfer Data";
      String asciival = "";
      int  Switchval = 0;
      //String  b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16;
         //b1=b2=b3=b4=b5=b6=b7=b8=b9=b10=b11=b12=b13=b14=b15=b16= null
         String b1 = request.getParameter("a1");
         String b2 = request.getParameter("a2");
         String b3 = request.getParameter("a3");
         String b4 = request.getParameter("a4");
         String b5 = request.getParameter("a5");
         String b6 = request.getParameter("a6");
         String b7 = request.getParameter("a7");
         String b8 = request.getParameter("a8");
         String b9 = request.getParameter("a9");
         String b10 = request.getParameter("a10");
         String b11 = request.getParameter("a11");
         String b12 = request.getParameter("a12");
         String b13 = request.getParameter("a13");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            String b14 = request.getParameter("a14");
            String b15 = request.getParameter("a15");
            String b16 = request.getParameter("a16");
        for(int idx=1; idx <= 16; idx++ )
            {
               asciival = request.getParameter("a"+idx);
               if( asciival != null )
               Switchval = Switchval | Integer.parseInt(asciival);
            }
         asciival = Integer.toHexString(Switchval);
            for( int idx=asciival.length(); idx<4; idx++ )
                     asciival = "0" + asciival;
                asciival = "#"+asciival;
//       System.out.println( asciival,"#%04X",Switchval );
         try { Process dosProcess = Runtime.getRuntime().exec( "RS232.EXE " + asciival );
      }
//  if(asciival = reguest.getParameter("a"+idx)

         catch(IOException ex) {
                  System.err.println("Command failed with IO Exception.");
            }

         response.setContentType("text/html");
            out = response.getWriter();
         out.println("<HTML><HEAD><TITLE>");
            out.println(title);
            out.println("</TITLE></HEAD><BODY>");
         out.println(" <center> ");
        out.println("<font size=5 color=#0080ff><b>Show Control Device On Internet</b></font>
<br> ");
       out.println("<img src=\"http://161.246.11.245/html/img/line53.gif\" width=\"447\"
height=\"14\" border=0 alt=\"Line Break\"><p> ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        out.println(" <BODY BACKGROUND=\"http://161.246.11.245/html/img/bg-wht2.gif\">");
        out.println(" <table border = 3  bordercolor=black> ");
        out.println(" <tr>");


    if (b1 != null)
                {out.println("<th><img src=\"http://161.246.11.245/html/img/red.jpg\" width = 50
    height = 60><br>");
                out.println("Device 1");}
              else
             {out.println("<th><img src=\"http://161.246.11.245/html/img/black.jpg\" width = 50
    height = 60><br>");
                out.println("Device 1");}


    if (b2 != null)
                {out.println("<th><img src=\"http://161.246.11.245/html/img/red.jpg\" width = 50
    height = 60><br>");
                out.println("Device 2");}
              else
             {out.println("<th><img src=\"http://161.246.11.245/html/img/black.jpg\" width = 50
    height = 60><br>");
                out.println("Device 2");}
    if (b3 != null)
                {out.println("<th><img src=\"http://161.246.11.245/html/img/red.jpg\" width = 50
    height = 60><br>");
                out.println("Device 3");}
              else
             {out.println("<th><img src=\"http://161.246.11.245/html/img/black.jpg\" width = 50
    height = 60><br>");
                out.println("Device 3");}
    if (b4 != null)
                {out.println("<th><img src=\"http://161.246.11.245/html/img/red.jpg\" width = 50
    height = 60><br>");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            out.println("Device 4");}
          else
         {out.println("<th><img src=\"http://161.246.11.245/html/img/black.jpg\" width = 50
height = 60><br>");
            out.println("Device 4");}
      out.println(" </tr>");
      out.println(" <tr>");
if (b5 != null)
            {out.println("<th><img src=\"http://161.246.11.245/html/img/red.jpg\" width = 50
height = 60><br>");
            out.println("Device 5");}
          else
         {out.println("<th><img src=\"http://161.246.11.245/html/img/black.jpg\" width = 50
height = 60><br>");
            out.println("Device 5");}
if (b6 != null)
            {out.println("<th><img src=\"http://161.246.11.245/html/img/red.jpg\" width = 50
height = 60><br>");
            out.println("Device 6");}
          else
         {out.println("<th><img src=\"http://161.246.11.245/html/img/black.jpg\" width = 50
height = 60><br>");
            out.println("Device 6");}
if (b7 != null)
            {out.println("<th><img src=\"http://161.246.11.245/html/img/red.jpg\" width = 50
height = 60><br>");
            out.println("Device 7");}
          else
         {out.println("<th><img src=\"http://161.246.11.245/html/img/black.jpg\" width = 50
height = 60><br>");
            out.println("Device 7");}
if (b8 != null)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {out.println("<th><img src=\"http://161.246.11.245/html/img/red.jpg\" width = 50
height = 60><br>");
            out.println("Device 8");}
          else
         {out.println("<th><img src=\"http://161.246.11.245/html/img/black.jpg\" width = 50
height = 60><br>");
            out.println("Device 8");}
      out.println(" </tr>");
      out.println(" <tr>");

if (b9 != null)
            {out.println("<th><img src=\"http://161.246.11.245/html/img/red.jpg\" width = 50
height = 60><br>");
            out.println("Device 9");}
          else
         {out.println("<th><img src=\"http://161.246.11.245/html/img/black.jpg\" width = 50
height = 60><br>");
            out.println("Device 9");}
if (b10 != null)
            {out.println("<th><img src=\"http://161.246.11.245/html/img/red.jpg\" width = 50
height = 60><br>");
            out.println("Device 10");}
          else
         {out.println("<th><img src=\"http://161.246.11.245/html/img/black.jpg\" width = 50
height = 60><br>");
            out.println("Device 10");}
if (b11 != null)
            {out.println("<th><img src=\"http://161.246.11.245/html/img/red.jpg\" width = 50
height = 60><br>");
            out.println("Device 11");}
          else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {out.println("<th><img src=\"http://161.246.11.245/html/img/black.jpg\" width = 50
height = 60><br>");
            out.println("Device 11");}
if (b12 != null)
            {out.println("<th><img src=\"http://161.246.11.245/html/img/red.jpg\" width = 50
height = 60><br>");
            out.println("Device 12");}
        else
        {out.println("<th><img src=\"http://161.246.11.245/html/img/black.jpg\" width = 50
height = 60><br>");
            out.println("Device 12");}
    out.println(" </tr>");
    out.println(" <tr>");

if (b13 != null)
            {out.println("<th><img src=\"http://161.246.11.245/html/img/red.jpg\" width = 50
height = 60><br>");
            out.println("Device 13");}
        else
        {out.println("<th><img src=\"http://161.246.11.245/html/img/black.jpg\" width = 50
height = 60><br>");
            out.println("Device 13");}
if (b14 != null)
            {out.println("<th><img src=\"http://161.246.11.245/html/img/red.jpg\" width = 50
height = 60><br>");
            out.println("Device 14");}
        else
        {out.println("<th><img src=\"http://161.246.11.245/html/img/black.jpg\" width = 50
height = 60><br>");
            out.println("Device 14");}
if (b15 != null)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {out.println("<th><img src=\"http://161.246.11.245/html/img/red.jpg\" width = 50
height = 60><br>");
                out.println("Device 15");}
            else
           {out.println("<th><img src=\"http://161.246.11.245/html/img/black.jpg\" width = 50
height = 60><br>");
                out.println("Device 15");}
if (b16 != null)
                {out.println("<th><img src=\"http://161.246.11.245/html/img/red.jpg\" width = 50
height = 60><br>");
                out.println("Device 16");}
            else
           {out.println("<th><img src=\"http://161.246.11.245/html/img/black.jpg\" width = 50
height = 60><br>");
                out.println("Device 16");}
      out.println(" <tr>");
      out.println(" </table>");


           out.println("</BODY></HTML>");
           out.close();
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS232.c

```
#include <dos.h>
#include <stdio.h>
#include <conio.h>
#include <string.h>
#define PORT1  0x3F8

int PortSerial;
int argc;

/* ------ protiotype function ------*/

void InitSerial(int port);
void SendData(char data);


/*-------- sub function module -------*/

void InitSerial(int port)
{
 PortSerial = port;
 outportb(port+1,0x00);
 outportb(port+3,0x80);
 outportb(port+0,0x0C);
 outportb(port+1,0x00);
 outportb(port+3,0x03);
 outportb(port+2,0xC7);
 outportb(port+4,0x0B);
}

void SendData(char data)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  outportb(PortSerial,data);
  delay(1000);
}


/* ------- main program ------*/

main (int argc,char *argv[])
{
   char *s = argv[1];

   InitSerial(PORT1);

   SendData(s[0]);
   SendData(s[1]);
   SendData(s[2]);
   SendData(s[3]);
   SendData(s[4]);

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IC.asm

```
; NAME          AT89C2051 Project remote serial version 1.0
; Compiler : SXA51.EXE
; Date    : 01/04/2000

;----------<< defined baudrate >>----------
BAUD_1200   equ   0e8h          ; 1200 baud timer reload value
BAUD_2400   equ   0f4h          ; 2400 baud
BAUD_4800   equ   0fah          ; 4800 baud
BAUD_9600   equ   0fdh          ; 9600 baud

;-------<< defined bytes address >>---------
            org   00H
            ds    8             ; reg R0 - R7
stack:      ds    24            ; stack depth
Systemflag: ds    1             ; address bit

; ----------<< power on reset >>----------
            org   0000H         ; power on/reset vector
            jmp   start

            org   0003H         ; external interrupt 0 vector
            reti                ; undefined

            org   000BH         ; timer 0 overflow vector
            reti                ; undefined

            org   0013H         ; external interrupt 1 vector
            reti                ; undefined

            org   001BH         ; timer 1 overflow vector
            reti                ; undefined
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                org     0023H           ; serial I/O interrupt vector
                reti                    ; undefined


        org     0100H           ; begin code space
; ------<< subroutine initialze main program >>-------
initialize:   movPCON, #0       ; initialize power control register
                mov     IE, #0          ; deactivate all interrupts
                mov     SCON, #01010000b  ; serial port mode one
                mov     TMOD, #00100000b  ; timer one 8-bit auto-reload,
                                          ;  timer zero 16-bit
          mov   a, #BAUD_9600
                mov     TH1, a          ; timer one reload value
          mov   TL1, a          ; timer one reload value
          setb  TR1             ; Set timer 1 control run
          mov   p1, #0
          clr   p3.7
          clr   p3.5
          setb  p3.7
          nop
          clr   p3.7

          setb  p3.5
          nop
          clr   p3.5
          ret

;-----------<< subroutine TXD byte RS232 >>-----------
;input : acc    data byte for TX
;output: none
;reg.  : acc.
TXDByte:      push    IE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        clr   ES          ;disable serial Interrupt
        clr   TI
        mov   SBUF, a     ;send data at acc to serial port in SBUF
        jnb   TI, $
        clr   TI
        pop   IE
        ret

;-----------<< subroutine RXD byte RS232 >>-----------
;input : none
;output: acc    return data in serial port
;reg.  : acc,
RXDByte:    push  IE
        clr   ES          ; disable Interrupt
        jnb   RI, $       ; wait data
        clr   RI
        mov   a, SBUF     ;get data serial port form SBUF to acc
        pop   IE
        ret

;-----------<< subroutine RXD byte RS232 >>-----------
;Receive ASCII 2 bytes from serial port and convert to hex in acc
;input : serial port 2 bytes <high byte><low byte> of ascii
;output: acc    return data in serial port
;reg.  : acc,
RXD2ASCII:    push  b
        push  IE
        clr   ES
        jnb   RI, $       ;Wait data
        clr   RI
        mov   a, SBUF
        call  ASCIIToHex  ;Convert to hex
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        swap    a
        mov     b, a            ;Hi-nibble
        ;** second byte **
        jnb     RI, $           ;Wait for second byte
        clr     RI
        mov     a, SBUF
        call    ASCIIToHex      ;Convert ot hex
        orl     a, b            ;Merge
        pop     IE
        pop     b
        ret

;---------<< subroutine Hex_2ASC >>----------
;input : acc
;output: a = high byte , b = low byte
;reg.  : acc, b    , call  HexToASCII
Hex_2ASCII:   push    acc
        call    HexToASCII
        mov     b, a            ; B = Lo byte
        pop     acc
        swap    a
        call    HexToASCII
        ret
;
HexToASCII:   push    PSW
        clr     CY
        anl     a, #00001111b
        cjne    a, #09h,HtoA1
        setb    CY
 HtoA1:     jnb     CY,HtoA2      ;> 0AH
        orl     a, #30H         ;< 0AH
        sjmp    End_HtoA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
HtoA2:      subb    a, #09H         ;> 0AH
            orl     a, #40H
End_HtoA:   pop     PSW
            ret

;---------<< subroutine ASCII TO HEX >>----------
;Convert one byte of ASCII to one byte in hex.
;input : acc carry ASCII
;output: acc = HEX
;reg.  : acc
ASCIIToHex:   push    PSW
            clr     CY
            cjne    a, #41H, MORE_A     ; "A"
MORE_A:     jc      AH2
AH1:        add     a, #09
AH2:        anl     a, #00001111B
            pop     PSW
            ret

;----------<< subroutine delay_ms >>------------
; Delay for one mS times the value in the accumulator.
delay_ms:       push    acc
                push    b
                mov     b, #0
dd:         djnz  b, $              ; 500 uS @ 12 MHz
                djnz    b, $            ; 500 uS @ 12 MHz
                djnz    acc, dd
                pop     b
                pop     acc
                ret

;----------<< subroutine delay_sec >>-----------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; Delay for one second times the value in the accumulator.
delay_sec:    push      acc
              push      b
              mov       b, a
   ddd:       mov a, #250
              acall     delay_ms        ; 250 mS
              acall     delay_ms        ; 500 mS
              acall     delay_ms        ; 750 mS
              acall     delay_ms        ; 1000 mS
              djnz      b, ddd
              pop       b
              pop       acc
              ret


;------------<< main program >>-----------
start:        mov    r0, #8         ;clear Internal RAM
              mov    r2, #80h-8     ;and delay for oter device reset time
 res0:        mov    @r0, #0
              inc    r0
              djnz   r2, res0
              mov    sp, #stack          ; initialize stack pointer
                     call   initialize   ; initialize controller registers
             xx:
                     setb   p3.4
                     mov    a, #1
                     call   delay_Sec
                     clr    p3.4
                     mov    a, #1
                     call   delay_Sec
                     sjmp   xx
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
loop:   call   RXDByte
        cjne   a, #'#', loop
        call   RXD2ASCII
        mov    p1, a
        setb   p3.7
        nop
        clr    p3.7
        call   RXD2ASCII
        mov    p1, a
        setb   p3.5
        nop
        clr    p3.5
        sjmp   loop

        end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Compatible with MCS-51™ Products
- 2K Bytes of Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- 2.7V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Two-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 15 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-chip Analog Comparator
- Low-power Idle and Power-down Modes

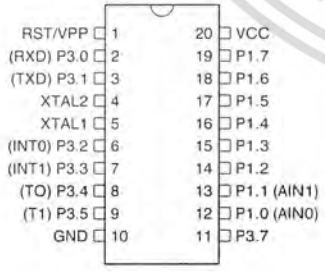
Description

The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2K 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. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C2051 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89C2051 provides the following standard features: 2K bytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C2051 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 Configuration

PDIP/SOIC



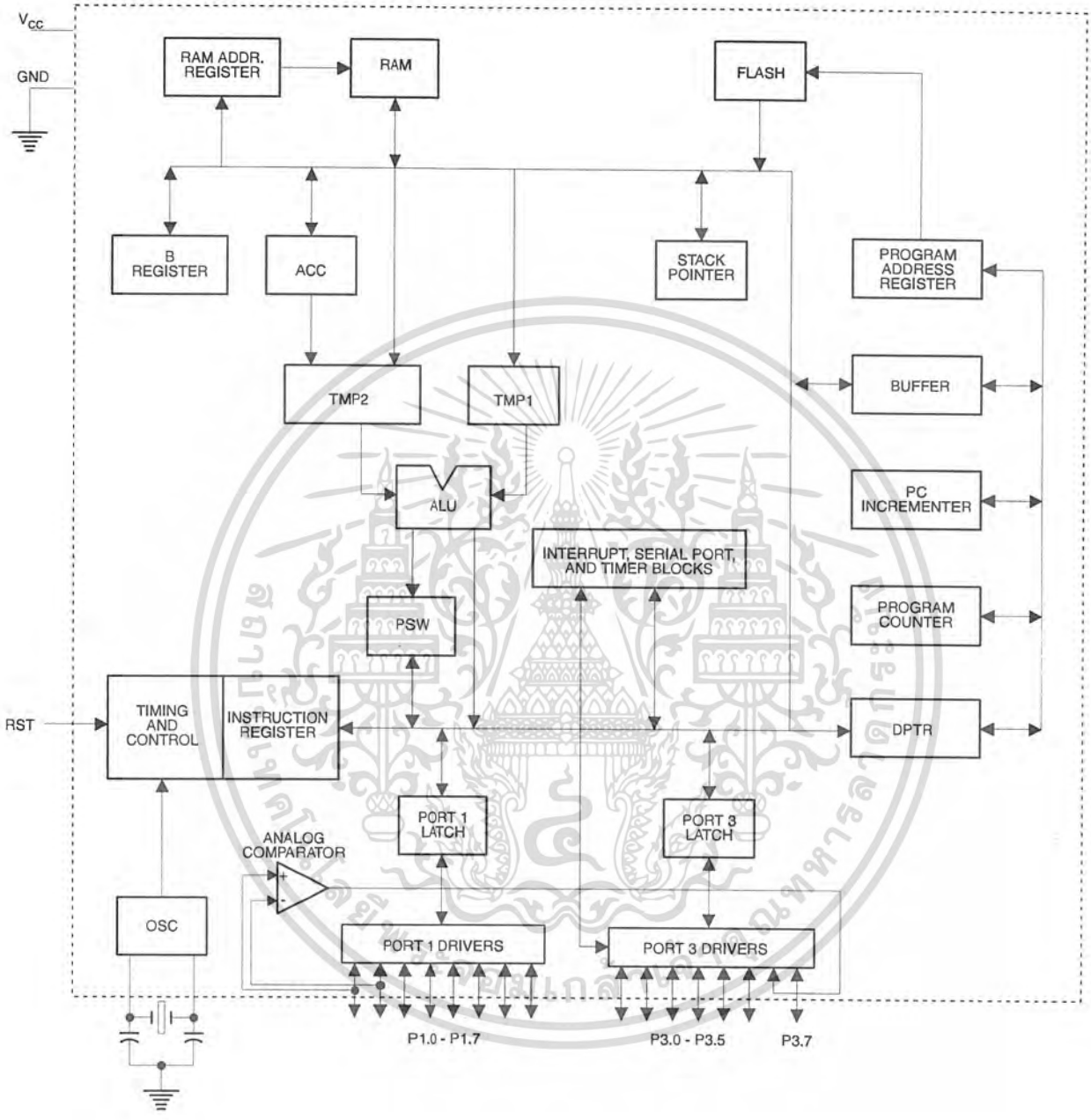
8-bit Microcontroller with 2K Bytes Flash

AT89C2051

Rev. 0368E–02/00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description

VCC

Supply voltage.

GND

Ground.

Port 1

Port 1 is an 8-bit bi-irectional I/O port. Port pins P1.2 to P1.7 provide internal pullups. P1.0 and P1.1 require external pullups. P1.0 and P1.1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output buffers can sink 20 mA and can drive LED displays directly. When 1s are written to Port 1 pins, they can be used as inputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current (I_{IL}) because of the internal pullups.

Port 1 also receives code data during Flash programming and verification.

Port 3

Port 3 pins P3.0 to P3.5, P3.7 are seven bi-irectional I/O pins with internal pullups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general purpose I/O pin. The Port 3 output buffers can sink 20 mA. 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 AT89C2051 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)

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

RST

Reset input. All I/O pins are reset to 1s as soon as RST goes high. Holding the RST pin high for two machine cycles while the oscillator is running resets the device.

Each machine cycle takes 12 oscillator or clock cycles.

XTAL1

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

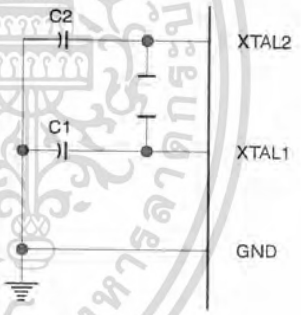
XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

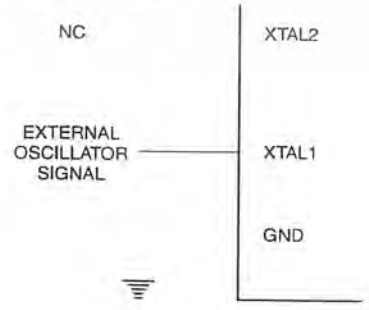
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected 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.

Figure 1. Oscillator Connections



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

Figure 2. External Clock Drive Configuration



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in the table below.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Table 1. AT89C2051 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H									0CFH
0C0H									0C7H
0B8H	IP XXX00000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0XX00000								0AFH
0A0H									0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
80H		SP 00000111	DPL 00000000	DPH 00000000				PCON 0XXX0000	87H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Restrictions on Certain Instructions

The AT89C2051 and is an economical and cost-effective member of Atmel's growing family of microcontrollers. It contains 2K bytes of flash program memory. It is fully compatible with the MCS-51 architecture, and can be programmed using the MCS-51 instruction set. However, there are a few considerations one must keep in mind when utilizing certain instructions to program this device.

All the instructions related to jumping or branching should be restricted such that the destination address falls within the physical program memory space of the device, which is 2K for the AT89C2051. This should be the responsibility of the software programmer. For example, LJMP 7E0H would be a valid instruction for the AT89C2051 (with 2K of memory), whereas LJMP 900H would not.

1. Branching instructions:

LCALL, LJMP, ACALL, AJMP, SJMP, JMP @A+DPTR

These unconditional branching instructions will execute correctly as long as the programmer keeps in mind that the destination branching address must fall within the physical boundaries of the program memory size (locations 00H to 7FFH for the 89C2051). Violating the physical space limits may cause unknown program behavior.

CJNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, JNZ With these conditional branching instructions the same rule above applies. Again, violating the memory boundaries may cause erratic execution.

For applications involving interrupts the normal interrupt service routine address locations of the 80C51 family architecture have been preserved.

2. MOVX-related instructions, Data Memory:

The AT89C2051 contains 128 bytes of internal data memory. Thus, in the AT89C2051 the stack depth is limited to 128 bytes, the amount of available RAM. External DATA memory access is not supported in this device, nor is external PROGRAM memory execution. Therefore, no MOVX [...] instructions should be included in the program.

A typical 80C51 assembler will still assemble instructions, even if they are written in violation of the restrictions mentioned above. It is the responsibility of the controller user to know the physical features and limitations of the device being used and adjust the instructions used correspondingly.

Program Memory Lock Bits

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

Lock Bit Protection Modes[1]

Program Lock Bits			
	LB1	LB2	Protection Type
1	U	U	No program lock features.
2	P	U	Further programming of the Flash is disabled.
3	P	P	Same as mode 2, also verify is disabled.

Note: 1. The Lock Bits can only be erased with the Chip Erase operation.

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.

P1.0 and P1.1 should be set to "0" if no external pullups are used, or set to "1" if external pullups are used.

It should be noted that when idle is terminated by a hardware 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.

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.

P1.0 and P1.1 should be set to "0" if no external pullups are used, or set to "1" if external pullups are used.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programming The Flash

The AT89C2051 is shipped with the 2K bytes of on-chip PEROM code memory array in the erased state (i.e., contents = FFH) and ready to be programmed. The code memory array is programmed one byte at a time. *Once the array is programmed, to re-program any non-blank byte, the entire memory array needs to be erased electrically.*

Internal Address Counter: The AT89C2051 contains an internal PEROM address counter which is always reset to 000H on the rising edge of RST and is advanced by applying a positive going pulse to pin XTAL1.

Programming Algorithm: To program the AT89C2051, the following sequence is recommended.

1. Power-up sequence:
   Apply power between V_{CC} and GND pins
   Set RST and XTAL1 to GND
2. Set pin RST to "H"
   Set pin P3.2 to "H"
3. Apply the appropriate combination of "H" or "L" logic levels to pins P3.3, P3.4, P3.5, P3.7 to select one of the programming operations shown in the PEROM Programming Modes table.

To Program and Verify the Array:

4. Apply data for Code byte at location 000H to P1.0 to P1.7.
5. Raise RST to 12V to enable programming.
6. Pulse P3.2 once to program a byte in the PEROM array or the lock bits. The byte-write cycle is self-timed and typically takes 1.2 ms.
7. To verify the programmed data, lower RST from 12V to logic "H" level and set pins P3.3 to P3.7 to the appropiate levels. Output data can be read at the port P1 pins.
8. To program a byte at the next address location, pulse XTAL1 pin once to advance the internal address counter. Apply new data to the port P1 pins.
9. Repeat steps 5 through 8, changing data and advancing the address counter for the entire 2K bytes array or until the end of the object file is reached.
10. Power-off sequence:
    set XTAL1 to "L"
    set RST to "L"
    Turn V_{CC} power off

$\overline{\text{Data}}$ Polling: The AT89C2051 features $\overline{\text{Data}}$ 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 data on P1.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. $\overline{\text{Data}}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{\text{Busy}}$: The Progress of byte programming can also be monitored by the RDY/$\overline{\text{BSY}}$ output signal. Pin P3.1 is pulled low after P3.2 goes High during programming to indicate BUSY. P3.1 is pulled High again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed code data can be read back via the data lines for verification:

1. Reset the internal address counter to 000H by bringing RST from "L" to "H".
2. Apply the appropriate control signals for Read Code data and read the output data at the port P1 pins.
3. Pulse pin XTAL1 once to advance the internal address counter.
4. Read the next code data byte at the port P1 pins.
5. Repeat steps 3 and 4 until the entire array is read.

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 PEROM array (2K bytes) and the two Lock Bits are erased electrically by using the proper combination of control signals and by holding P3.2 low for 10 ms. The code array is written with all "1"s in the Chip Erase operation and must be executed before any non-blank memory byte can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 001H, and 002H, except that P3.5 and P3.7 must be pulled to a logic low. The values returned are as follows.

(000H) = 1EH indicates manufactured by Atmel
(001H) = 21H indicates 89C2051

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/VPP	P3.2/$\overline{PROG}$	P3.3	P3.4	P3.5	P3.7
Write Code Data[1][3]		12V		L	H	H	H
Read Code Data[1]		H	H	L	L	H	H
Write Lock	Bit - 1	12V		H	H	H	H
	Bit - 2	12V		H	H	L	L
Chip Erase		12V	(2)	H	L	L	L
Read Signature Byte		H	H	L	L	L	L

Notes:
1. The internal PEROM address counter is reset to 000H on the rising edge of RST and is advanced by a positive pulse at XTAL 1 pin.
2. Chip Erase requires a 10 ms $\overline{PROG}$ pulse.
3. P3.1 is pulled Low during programming to indicate RDY/$\overline{BSY}$.

Figure 3. Programming the Flash Memory

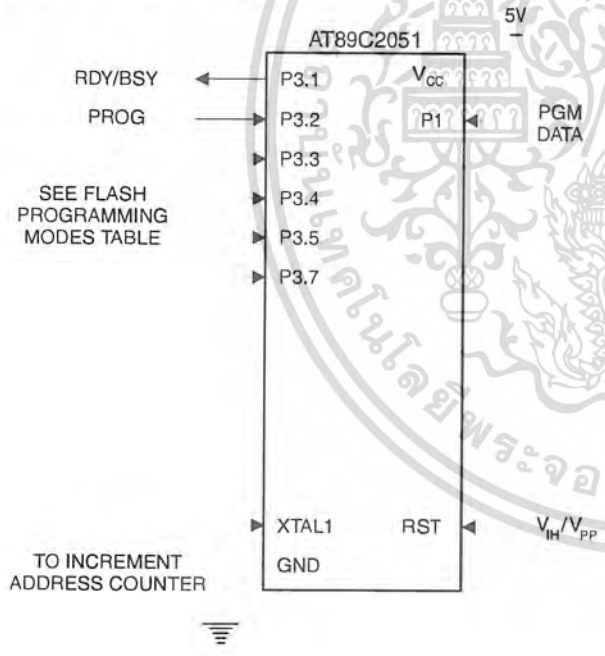
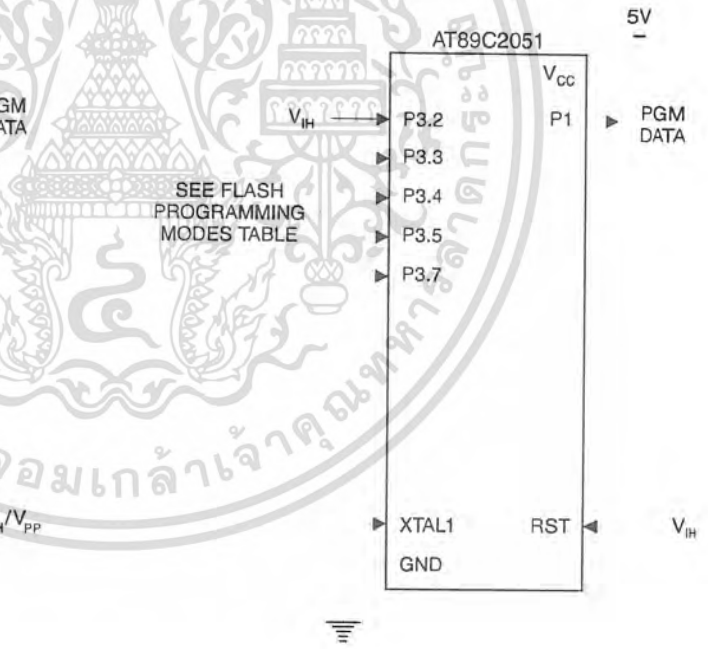


Figure 4. Verifying the Flash Memory



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

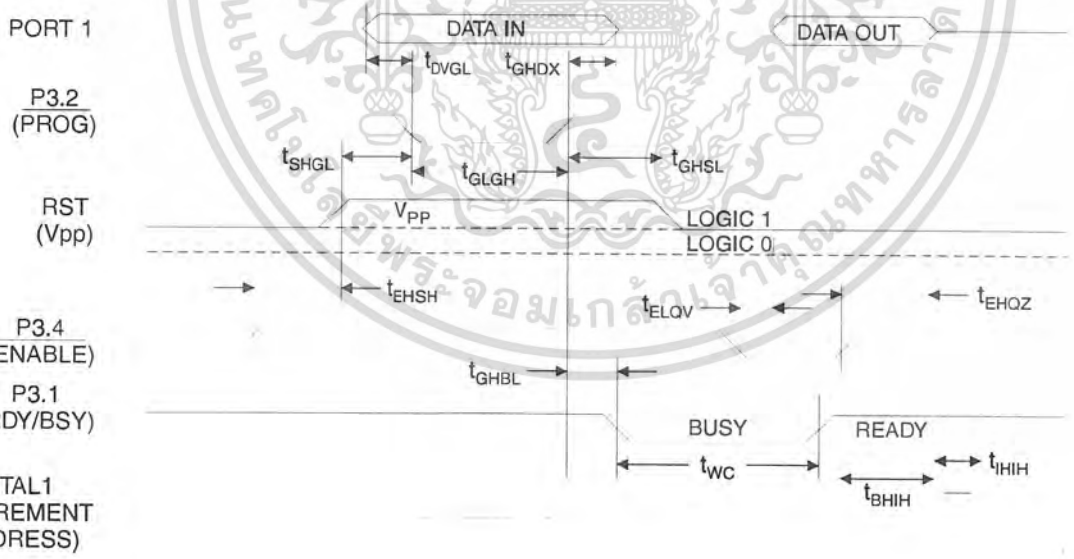
Flash Programming and Verification Characteristics

T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		250	µA
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	1.0		µs
t_{GHDX}	Data Hold after $\overline{PROG}$	1.0		µs
t_{EHSH}	P3.4 ($\overline{ENABLE}$) High to V_{PP}	1.0		µs
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}	V_{PP} Hold after $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		1.0	µs
t_{EHQZ}	Data Float after $\overline{ENABLE}$	0	1.0	µs
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		50	ns
t_{WC}	Byte Write Cycle Time		2.0	ms
t_{BHIH}	RDY/$\overline{BSY}$\ to Increment Clock Delay	1.0		µs
t_{IHIL}	Increment Clock High	200		ns

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

Flash Programming and Verification Waveforms



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	25.0 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

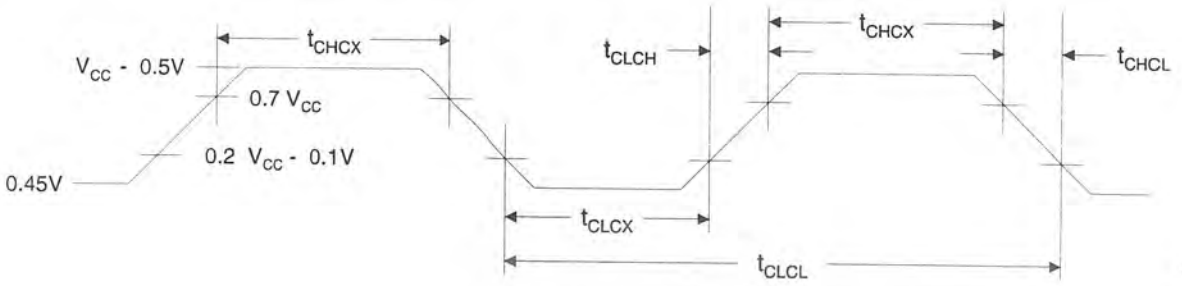
T_A = -40°C to 85°C, V_{CC} = 2.0V to 6.0V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage		-0.5	0.2 V_{CC} - 0.1	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, 3)	I_{OL} = 20 mA, V_{CC} = 5V I_{OL} = 10 mA, V_{CC} = 2.7V		0.5	V
V_{OH}	Output High-voltage (Ports 1, 3)	I_{OH} = -80 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -30 µA	0.75 V_{CC}		V
		I_{OH} = -12 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1, 3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 3)	V_{IN} = 2V, V_{CC} = 5V ± 10%		-750	µA
I_{LI}	Input Leakage Current (Port P1.0, P1.1)	0 < V_{IN} < V_{CC}		±10	µA
V_{OS}	Comparator Input Offset Voltage	V_{CC} = 5V		20	mV
V_{CM}	Comparator Input Common Mode Voltage		0	V_{CC}	V
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, V_{CC} = 6V/3V		15/5.5	mA
		Idle Mode, 12 MHz, V_{CC} = 6V/3V P1.0 & P1.1 = 0V or V_{CC}		5/1	mA
	Power-down Mode[2]	V_{CC} = 6V P1.0 & P1.1 = 0V or V_{CC}		100	µA
		V_{CC} = 3V P1.0 & P1.1 = 0V or V_{CC}		20	µA

Notes:
1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 20 mA
   Maximum total I_{OL} for all output pins: 80 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

External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	V_{CC} = 2.7V to 6.0V Min	V_{CC} = 2.7V to 6.0V Max	V_{CC} = 4.0V to 6.0V Min	V_{CC} = 4.0V to 6.0V Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	12	0	24	MHz
t_{CLCL}	Clock Period	83.3		41.6		ns
t_{CHCX}	High Time	30		15		ns
t_{CLCX}	Low Time	30		15		ns
t_{CLCH}	Rise Time		20		20	ns
t_{CHCL}	Fall Time		20		20	ns

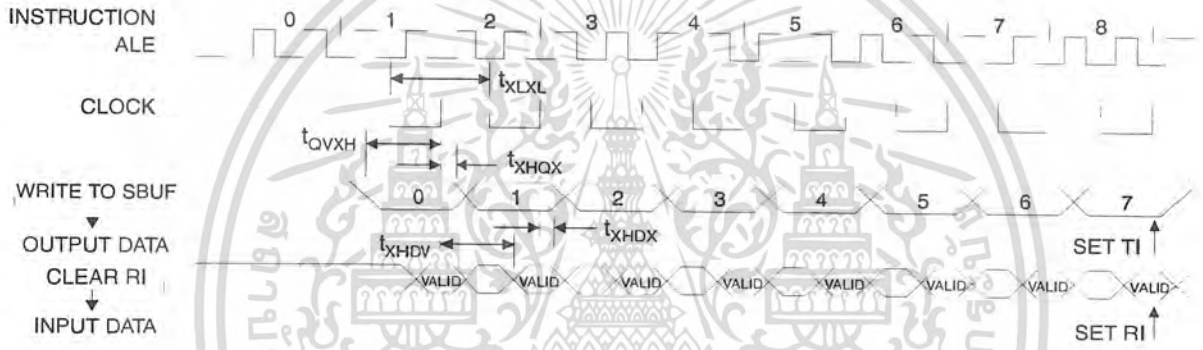
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

V_{CC} = 5.0V ± 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

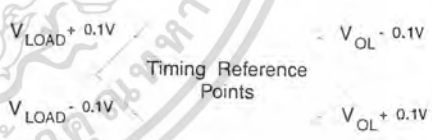
Shift Register Mode Timing Waveforms



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]

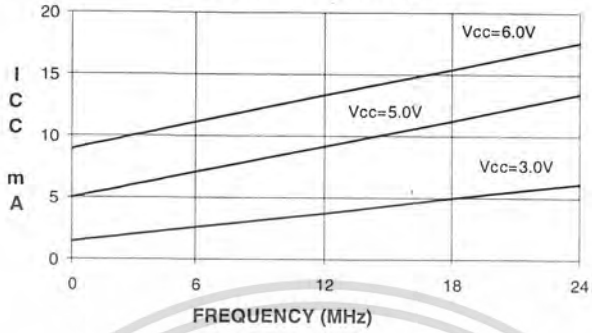


Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change frothe loaded V_{OH}/V_{OL} level occurs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

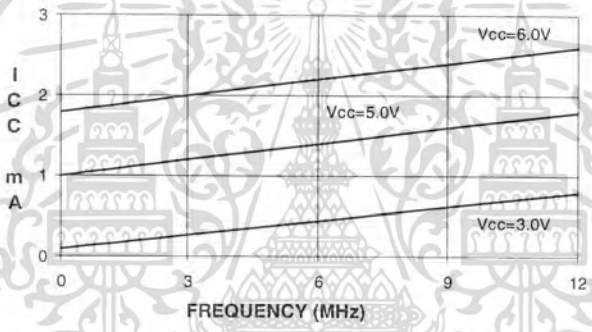
AT89C2051

TYPICAL ICC - ACTIVE (85°C)



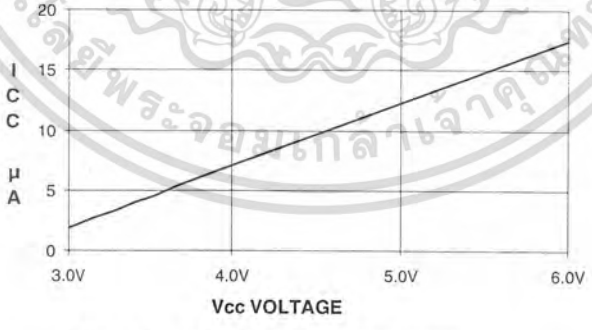
AT89C2051

TYPICAL ICC - IDLE (85°C)



AT89C2051

TYPICAL ICC vs. VOLTAGE- POWER DOWN (85°C)



Notes:
1. XTAL1 tied to GND for I_{CC} (power-down)
2. P.1.0 and P1.1 = V_{CC} or GND
3. Lock bits programmed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

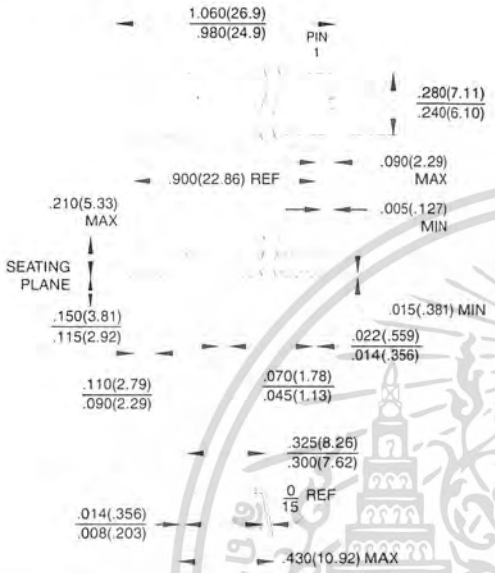
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	2.7V to 6.0V	AT89C2051-12PC AT89C2051-12SC	20P3 20S	Commercial (0°C to 70°C)
		AT89C2051-12PI AT89C2051-12SI	20P3 20S	Industrial (-40°C to 85°C)
24	4.0V to 6.0V	AT89C2051-24PC AT89C2051-24SC	20P3 20S	Commercial (0°C to 70°C)
		AT89C2051-24PI AT89C2051-24SI	20P3 20S	Industrial (-40°C to 85°C)

Package Type	
20P3	20-lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)
20S	20-lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)

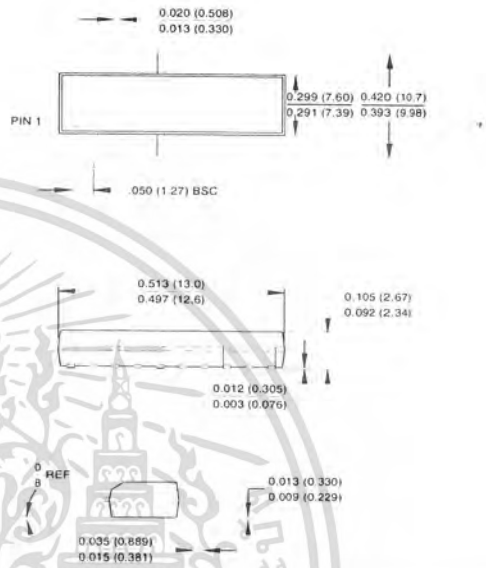
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Packaging Information

20P3, 20-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-001 AD



20S, 20-lead, 0.300" Wide, Plastic Gull WIng Small Outline (SOIC)
Dimensions in Inches and (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

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.

0368E–02/00/xM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SGS-THOMSON MICROELECTRONICS

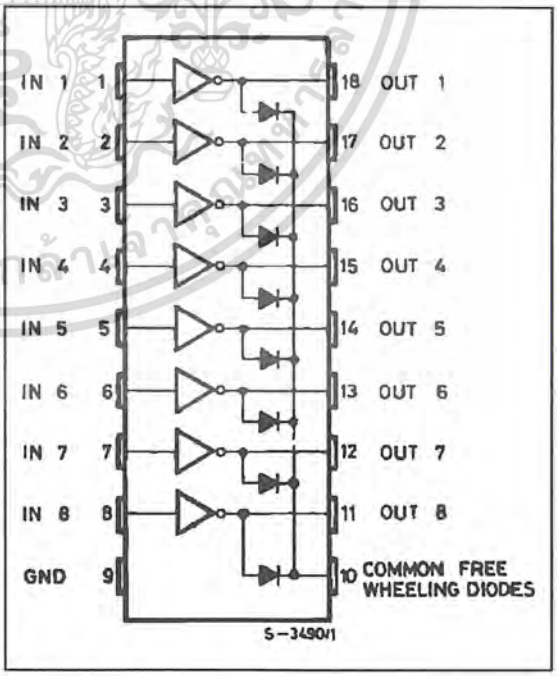
ULN2801A
ULN2802A - ULN2803A
ULN2804A - ULN2805A

EIGHT DARLINGTON ARRAYS

- EIGHT DARLINGTONS WITH COMMON EMITTERS
- OUTPUT CURRENT TO 500 mA
- OUTPUT VOLTAGE TO 50 V
- INTEGRAL SUPPRESSION DIODES
- VERSIONS FOR ALL POPULAR LOGIC FAMILIES
- OUTPUT CAN BE PARALLELED
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY BOARD LAYOUT

DIP18

PIN CONNECTION (top view)



DESCRIPTION

The ULN2801A-ULN2805A each contain eight darlington transistors with common emitters and integral suppression diodes for inductive loads. Each darlington features a peak load current rating of 600mA (500mA continuous) and can withstand at least 50V in the off state. Outputs may be paralleled for higher current capability.

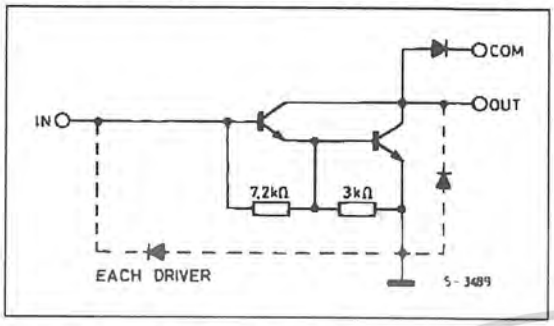
Five versions are available to simplify interfacing to standard logic families : the ULN2801A is designed for general purpose applications with a current limit resistor ; the ULN2802A has a 10.5kΩ input resistor and zener for 14-25V PMOS ; the ULN2803A has a 2.7kΩ input resistor for 5V TTL and CMOS ; the ULN2804A has a 10.5kΩ input resistor for 6-15V CMOS and the ULN2805A is designed to sink a minimum of 350mA for standard and Schottky TTL where higher output current is required.

All types are supplied in a 18-lead plastic DIP with a copper lead from and feature the convenient input-opposite-output pinout to simplify board layout.

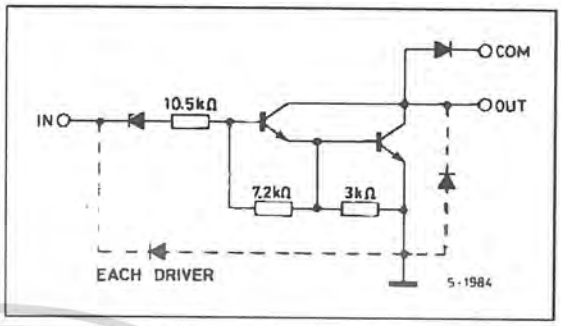
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCHEMATIC DIAGRAM AND ORDER CODES

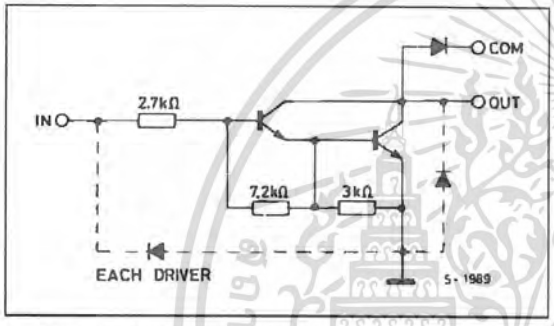
For ULN2801A (each driver for PMOS-CMOS)



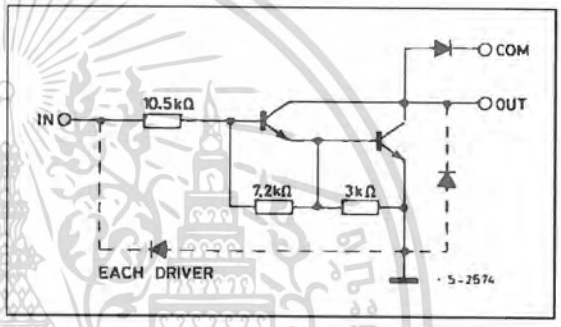
For ULN2802A (each driver for 14-15 V PMOS)



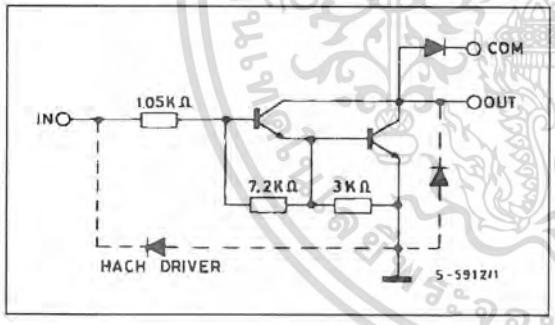
For ULN2803A (each driver for 5 V, TTL/CMOS)



For ULN2804A (each driver for 6-15 V CMOS/PMOS



For ULN2805A (each driver for high out TTL)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_o	Output Voltage	50	V
V_i	Input Voltage for ULN2802A, UL2803A, ULN2804A for ULN2805A	 30 15	V
I_C	Continuous Collector Current	500	mA
I_B	Continuous Base Current	25	mA
P_{tot}	Power Dissipation (one Darlington pair) (total package)	 1.0 2.25	W
T_{amb}	Operating Ambient Temperature Range	– 20 to 85	°C
T_{stg}	Storage Temperature Range	– 55 to 150	°C
Tj	Junction Temperature Range	– 20 to 150	°C

THERMAL DATA

Symbol	Parameter		Value	Unit
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient	Max.	55	°C/W

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

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit	Fig.
I_{CEX}	Output Leakage Current	V_{CE} = 50V			50	µA	1a
		T_{amb} = 70°C, V_{CE} = 50V			100	µA	1a
		T_{amb} = 70°C for ULN2802A V_{CE} = 50V, V_i = 6V			500	µA	1b
		for ULN2804A V_{CE} = 50V, V_i = 1V			500	µA	1b
$V_{CE(sat)}$	Collector-emitter Saturation Voltage	I_C = 100mA, I_B = 250µA		0.9	1.1	V	
		I_C = 200mA, I_B = 350µA		1.1	1.3	V	2
		I_C = 350mA, I_B = 500µA		1.3	1.6	V	
$I_{i(on)}$	Input Current	for ULN2802A V_i = 17V		0.82	1.25	mA	
		for ULN2803A V_i = 3.85V		0.93	1.35	mA	
		for ULN2804A V_i = 5V		0.35	0.5	mA	3
		V_i = 12V		1	1.45	mA	
		for ULN2805A V_i = 3V		1.5	2.4	mA	
$I_{i(off)}$	Input Current	T_{amb} = 70°C, I_C = 500µA	50	65		µA	4
$V_{i(on)}$	Input Voltage	V_{CE} = 2 V for ULN2802A I_C = 300mA			13	V	
		for ULN2803A I_C = 200mA			2.4	V	
		I_C = 250mA			2.7	V	
		I_C = 300mA			3	V	5
		for ULN2804A I_C = 125mA			5	V	
		I_C = 200mA			6	V	
		I_C = 275mA			7	V	
		I_C = 350mA			8	V	
		for ULN2805A I_C = 350mA			2.4	V	
h_{FE}	DC Forward Current Gain	for ULN2801A V_{CE} = 2V, I_C = 350mA	1000			–	2
C_i	Input Capacitance			15	25	pF	–
t_{PLH}	Turn-on Delay Time	0.5 V_i to 0.5 V_o		0.25	1	µs	–
t_{PHL}	Turn-off Delay Time	0.5 V_i to 0.5 V_o		0.25	1	µs	–
I_R	Clamp Diode Leakage Current	V_R = 50V			50	µA	6
		T_{amb} = 70°C, V_R = 50V			100	µA	6
V_F	Clamp Diode Forward Voltage	I_F = 350mA		1.7	2	V	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEST CIRCUITS

Figure 1a.

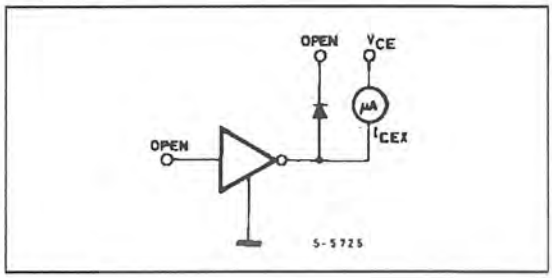


Figure 1b.

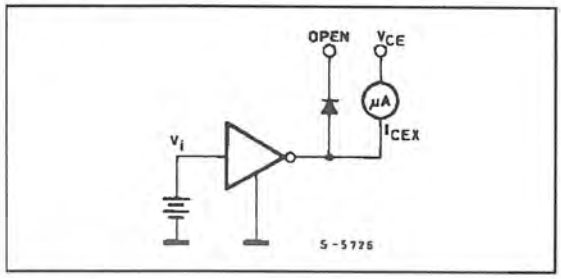


Figure 2.

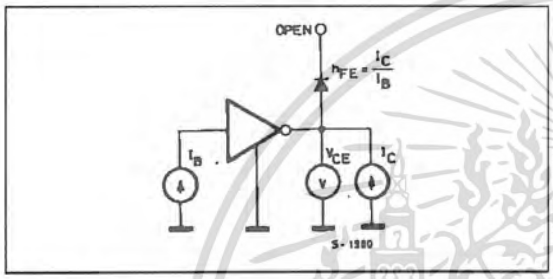


Figure 3.

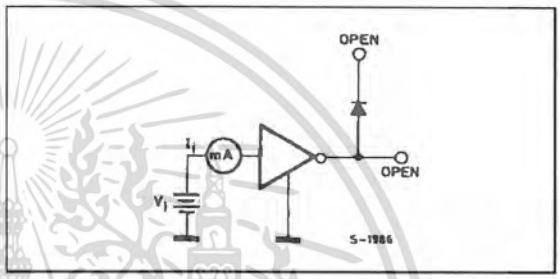


Figure 4.

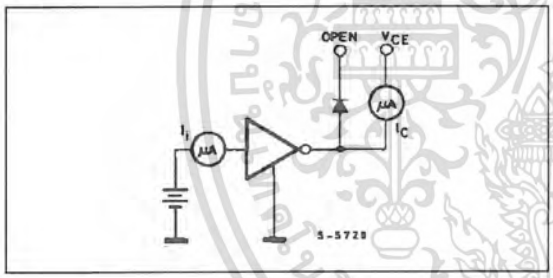


Figure 5.

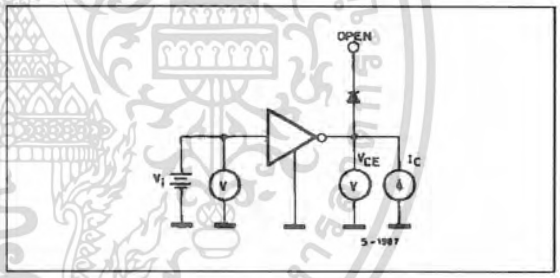


Figure 6.

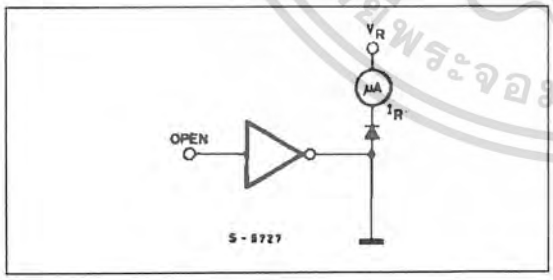
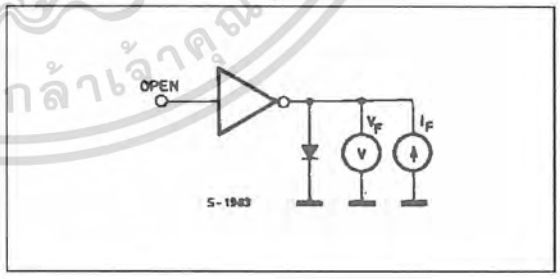


Figure 7.



SGS-THOMSON MICROELECTRONICS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 8 : Collector Current as a Function of Saturation Voltage.

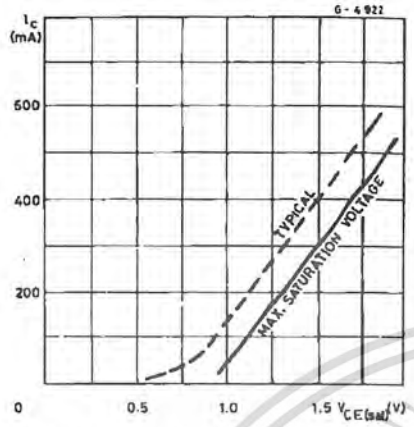


Figure 9 : Collector Current as a Function of Input Current.

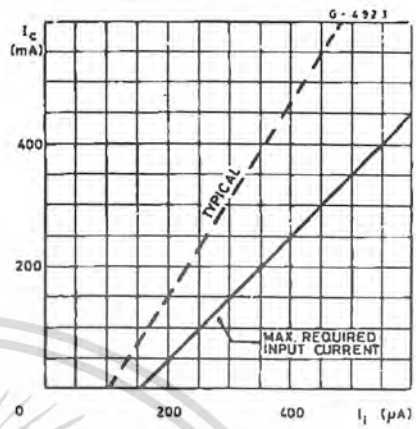


Figure 10 : Allowable Average Power Dissipation as a Function of Ambient Temperature.

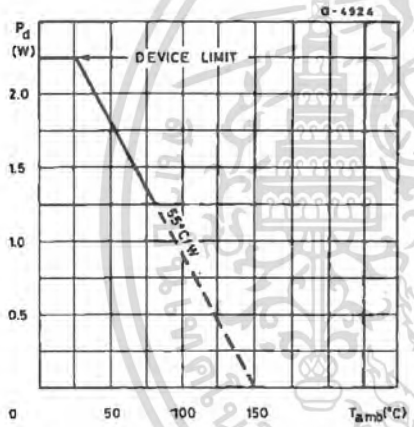


Figure 11 : Peak Collector Current as a Function of Duty Cycle.

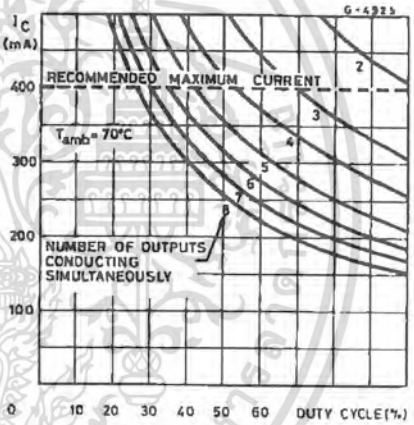


Figure 12 : Peak Collector Current as a Function of Duty.

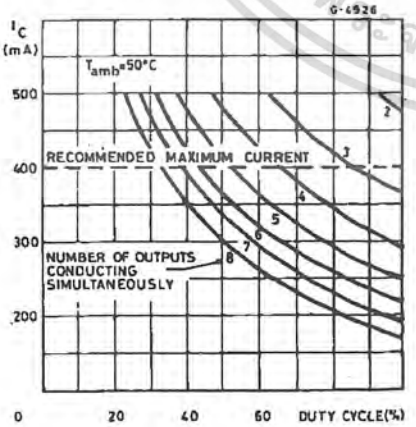
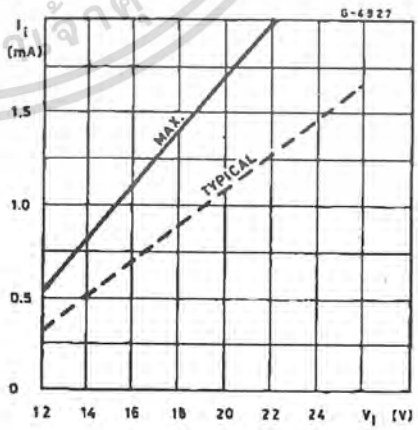


Figure 13 : Input Current as a Function of Input Voltage (for ULN2802A).



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 14 : Input Current as a Function of Input Voltage (for ULN2804A)

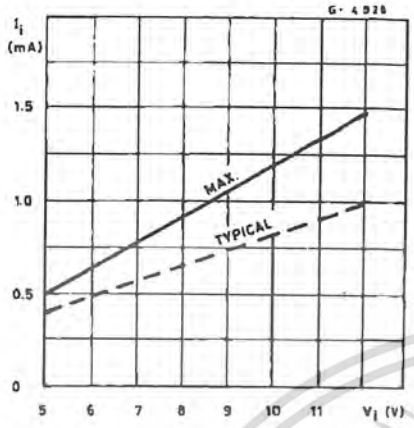


Figure 15 : Input Current as a Function of Input Voltage (for ULN2803A)

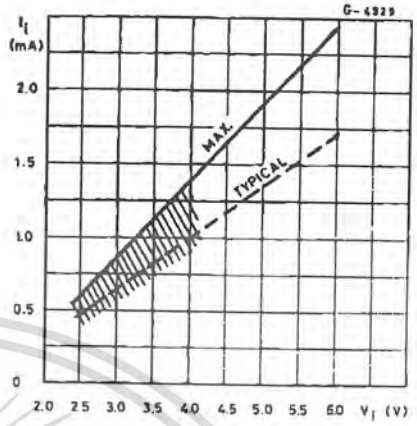
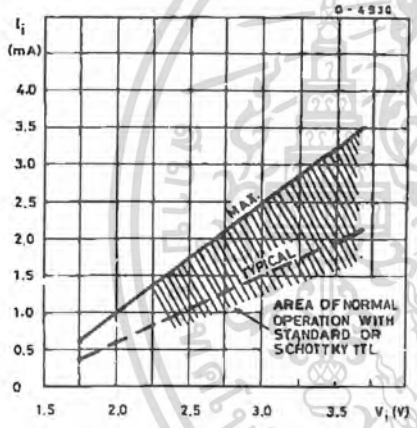


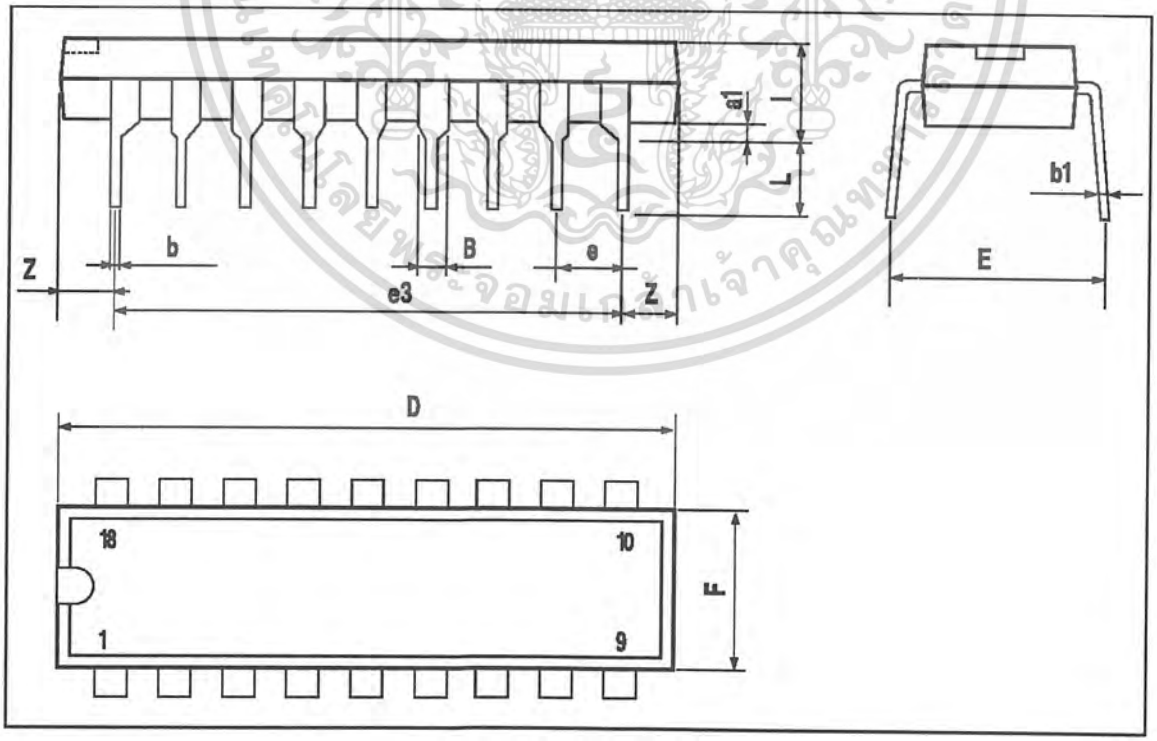
Figure 16 : Input Current as a Function of Input Voltage (for ULN2805A)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIP18 PACKAGE MECHANICAL DATA

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
a1	0.254			0.010		
B	1.39		1.65	0.055		0.065
b		0.46			0.018	
b1		0.25			0.010	
D			23.24			0.915
E		8.5			0.335	
e		2.54			0.100	
e3		20.32			0.800	
F			7.1			0.280
I			3.93			0.155
L		3.3			0.130	
Z		1.27	1.59		0.050	0.063



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Information furnished is believed to be accurate and reliable. However, SGS-THOMSON Microelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of SGS-THOMSON Microelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. SGS-THOMSON Microelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of SGS-THOMSON Microelectronics.

© 1997 SGS-THOMSON Microelectronics – Printed in Italy – All Rights Reserved

SGS-THOMSON Microelectronics GROUP OF COMPANIES

Australia - Brazil - Canada - China - France - Germany - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco - The Netherlands - Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.

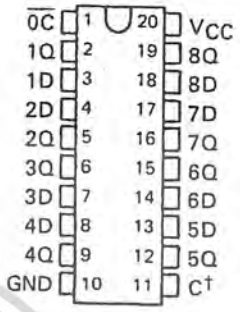
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SN54LS373, SN54LS374, SN54S373, SN54S374, SN74LS373, SN74LS374, SN74S373, SN74S374
OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS

SDLS165 – OCTOBER 1975 – REVISED MARCH 1988

- Choice of 8 Latches or 8 D-Type Flip-Flops In a Single Package
- 3-State Bus-Driving Outputs
- Full Parallel-Access for Loading
- Buffered Control Inputs
- Clock/Enable Input Has Hysteresis to Improve Noise Rejection ('S373 and 'S374)
- P-N-P Inputs Reduce D-C Loading on Data Lines ('S373 and 'S374)

SN54LS373, SN54LS374, SN54S373, SN54S374 . . . J OR W PACKAGE
SN74LS373, SN74LS374, SN74S373, SN74S374 . . . DW OR N PACKAGE
(TOP VIEW)



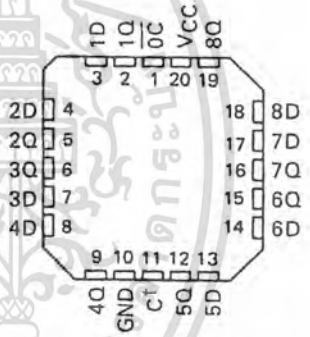
'LS373, 'S373
FUNCTION TABLE

OUTPUT ENABLE	ENABLE LATCH	D	OUTPUT
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

'LS374, 'S374
FUNCTION TABLE

OUTPUT ENABLE	CLOCK	D	OUTPUT
L	↑	H	H
L	↑	L	L
L	L	X	Q_0
H	X	X	Z

SN54LS373, SN54LS374, SN54S373, SN54S374 . . . FK PACKAGE
(TOP VIEW)



†C for 'LS373 and 'S373; CLK for 'LS374 and 'S374.

description

These 8-bit registers feature three-state outputs designed specifically for driving highly-capacitive or relatively low-impedance loads. The high-impedance third state and increased high-logic-level drive provide these registers with the capability of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pull-up components. They are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers.

The eight latches of the 'LS373 and 'S373 are transparent D-type latches meaning that while the enable (C) is high the Q outputs will follow the data (D) inputs. When the enable is taken low the output will be latched at the level of the data that was set up.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 1988, Texas Instruments Incorporated

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

description (continued)

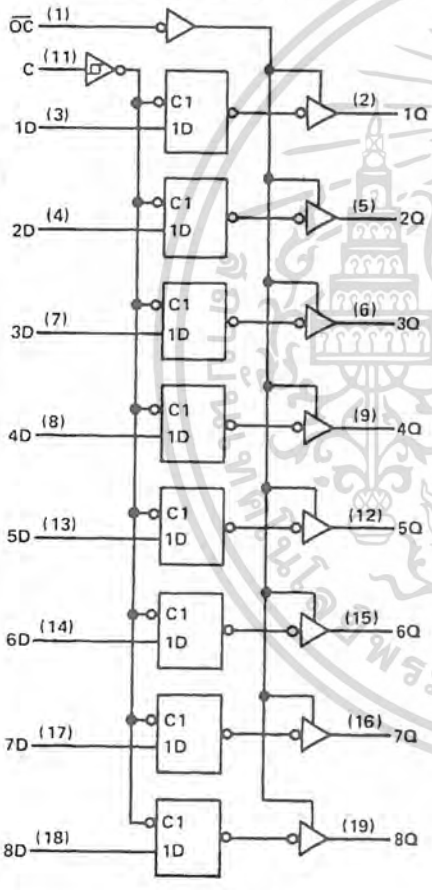
The eight flip-flops of the 'LS374 and 'S374 are edge-triggered D-type flip-flops. On the positive transition of the clock, the Q outputs will be set to the logic states that were setup at the D inputs.

Schmitt-trigger buffered inputs at the enable/clock lines of the 'S373 and 'S374 devices, simplify system design as ac and dc noise rejection is improved by typically 400 mV due to the input hysteresis. A buffered output control input can be used to place the eight outputs in either a normal logic state (high or low logic levels) or a high-impedance state. In the high-impedance state the outputs neither load nor drive the bus lines significantly.

The output control does not affect the internal operation of the latches or flip-flops. That is, the old data can be retained or new data can be entered even while the outputs are off.

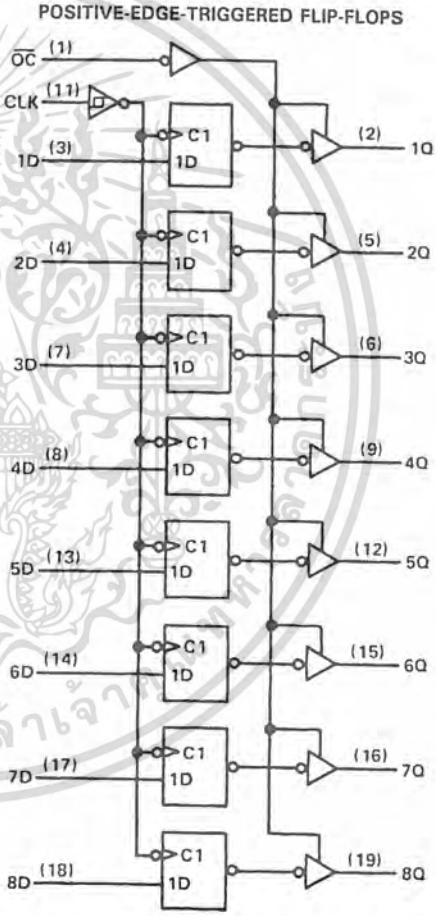
logic diagrams (positive logic)

'LS373, 'S373
TRANSPARENT LATCHES



⊔ for 'S373 only

'LS374, 'S374
POSITIVE-EDGE-TRIGGERED FLIP-FLOPS



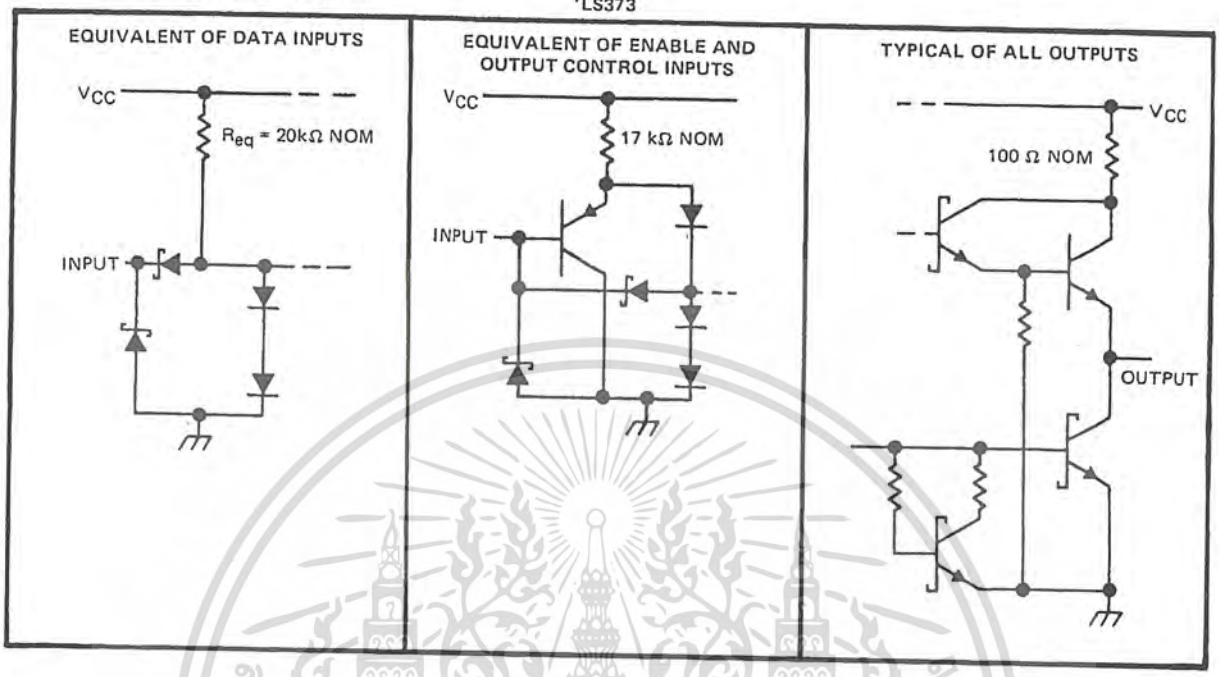
⊔ for 'S374 only

Pin numbers shown are for DW, J, N, and W packages.

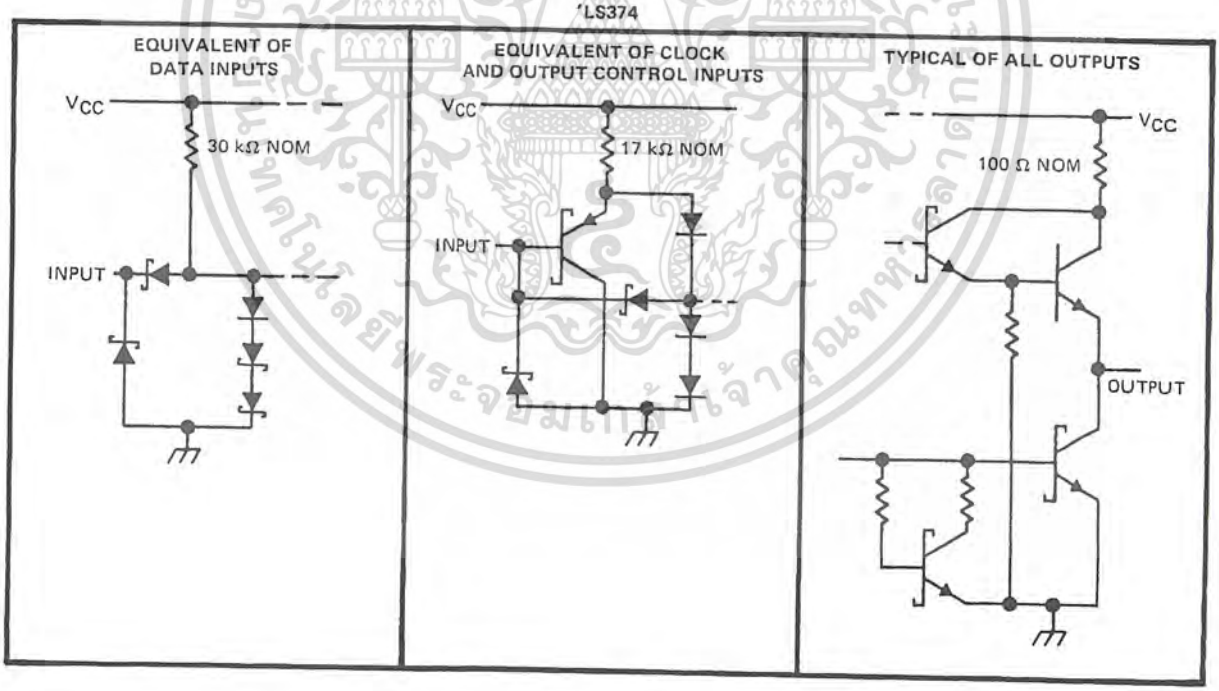
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

schematic of inputs and outputs

'LS373



'LS374



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1) . . . . . 7 V
Input voltage . . . . . 7 V
Off-state output voltage . . . . . 5.5 V
Operating free-air temperature range: SN54LS' . . . . . −55°C to 125°C
SN74LS' . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

recommended operating conditions

			SN54LS'			SN74LS'			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage		4.5	5	5.5	4.75	5	5.25	V
V_{OH}	High-level output voltage				5.5			5.5	V
I_{OH}	High-level output current				−1			−2.6	mA
I_{OL}	Low-level output current				12			24	mA
t_w	Pulse duration	CLK high	15			15			ns
		CLK low	15			15			
t_{su}	Data setup time	'LS373	5↓			5↓			ns
		'LS374	20↑			20↑			
t_h	Data hold time	'LS373	20↓			20↓			ns
		'LS374†	5↑			0↑			
T_A	Operating free-air temperature		−55		125	0		70	°C

†The t_h specification applies only for data frequency below 10 MHz. Designs above 10 MHz should use a minimum of 5 ns. (Commercial only)

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS†		SN54LS'			SN74LS'			UNIT
				MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH}	High-level input voltage			2			2			V
V_{IL}	Low-level input voltage					0.7			0.8	V
V_{IK}	Input clamp voltage	V_{CC} = MIN, I_I = −18 mA				−1.5			−1.5	V
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = V_{IL}max, I_{OH} = MAX		2.4	3.4		2.4	3.1		V
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = V_{IL}max	I_{OL} = 12 mA		0.25	0.4		0.25	0.4	V
			I_{OL} = 24 mA					0.35	0.5	
I_{OZH}	Off-state output current, high-level voltage applied	V_{CC} = MAX, V_{IH} = 2 V, V_O = 2.7 V				20			20	μA
I_{OZL}	Off-state output current, low-level voltage applied	V_{CC} = MAX, V_{IH} = 2 V, V_O = 0.4 V				−20			−20	μA
I_I	Input current at maximum input voltage	V_{CC} = MAX, V_I = 7 V				0.1			0.1	mA
I_{IH}	High-level input current	V_{CC} = MAX, V_I = 2.7 V				20			20	μA
I_{IL}	Low-level input current	V_{CC} = MAX, V_I = 0.4 V				−0.4			−0.4	mA
I_{OS}	Short-circuit output current§	V_{CC} = MAX		−30		−130	−30		−130	mA
I_{CC}	Supply current	V_{CC} = MAX, Output control at 4.5 V	'LS373		24	40		24	40	mA
			'LS374		27	40		27	40	

†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡All typical values are at V_{CC} = 5 V, T_A = 25°C.
§Not more than one output should be shorted at a time and duration of the short circuit should not exceed one second.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

switching characteristics, V_{CC} = 5 V, T_A = 25°C

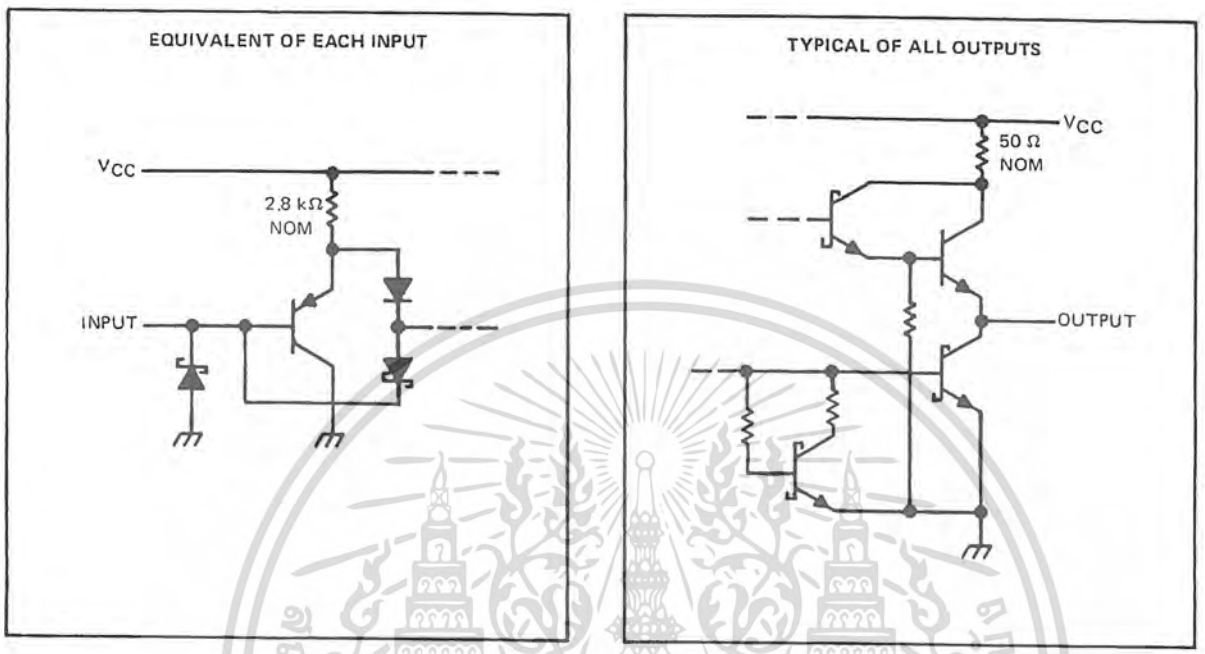
PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	'LS373			'LS374			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
f_{max}			C_L = 45 pF, R_L = 667 Ω See Notes 2 and 3				35	50		MHz
t_{PLH}	Data	Any Q			12	18				ns
t_{PHL}					12	18				
t_{PLH}	Clock or enable	Any Q			20	30		15	28	ns
t_{PHL}					18	30		19	28	
t_{PZH}	Output Control	Any Q			15	28		20	26	ns
t_{PZL}					25	36		21	28	
t_{PHZ}	Output Control	Any Q	C_L = 5 pF, R_L = 667 Ω See Note 3		15	25		15	28	ns
t_{PLZ}	Output Control	Any Q			12	20		12	20	ns

NOTES: 2. Maximum clock frequency is tested with all outputs loaded.
3. Load circuits and voltage waveforms are shown in Section 1.

f_{max} = maximum clock frequency
t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output
t_{PZH} = output enable time to high level
t_{PZL} = output enable time to low level
t_{PHZ} = output disable time from high level
t_{PLZ} = output disable time from low level

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

schematic of inputs and outputs



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1) . . . . . 7 V
Input voltage . . . . . 5.5 V
Off-state output voltage . . . . . 5.5 V
Operating free-air temperature range: SN54S' . . . . . −55°C to 125°C
SN74S' . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

recommended operating conditions

		SN54S'			SN74S'			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}		4.5	5	5.5	4.75	5	5.25	V
High-level output voltage, V_{OH}				5.5			5.5	V
High-level output current, I_{OH}				−2			−6.5	mA
Width of clock/enable pulse, t_w	High	6			6			ns
	Low	7.3			7.3			
Data setup time, t_{su}	'S373	0↓			0↓			ns
	'S374	5↑			5↑			
Data hold time, t_h	'S373	10↓			10↓			ns
	'S374	2↑			2↑			
Operating free-air temperature, T_A		−55		125	0		70	°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS†				MIN	TYP‡	MAX	UNIT
V_{IH}						2			V
V_{IL}								0.8	V
V_{IK}		V_{CC} = MIN, I_I = −18 mA						−1.2	V
V_{OH}	SN54S'	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OH} = MAX				2.4	3.4		V
	SN74S'					2.4	3.1		
V_{OL}		V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = 0.8 V, I_{OL} = 20 mA						0.5	V
I_{OZH}		V_{CC} = MAX, V_{IH} = 2 V, V_O = 2.4 V						50	μA
I_{OZL}		V_{CC} = MAX, V_{IH} = 2 V, V_O = 0.5 V						−50	μA
I_I		V_{CC} = MAX, V_I = 5.5 V						1	mA
I_{IH}		V_{CC} = MAX, V_I = 2.7 V						50	μA
I_{IL}		V_{CC} = MAX, V_I = 0.5 V						−250	μA
I_{OS}§		V_{CC} = MAX				−40		−100	mA
I_{CC}		V_{CC} = MAX	'S373		outputs high			160	mA
					outputs low			160	
					outputs disabled			190	
			'S374		outputs high			110	
					outputs low			140	
					outputs disabled			160	
				CLK and $\overline{OC}$ at 4 V, D inputs at 0 V				180	

†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡All typical values are at V_{CC} = 5 V, T_A = 25°C.
§Not more than one output should be shorted at a time and duration of the short-circuit should not exceed one second.

switching characteristics, V_{CC} = 5 V, T_A = 25°C

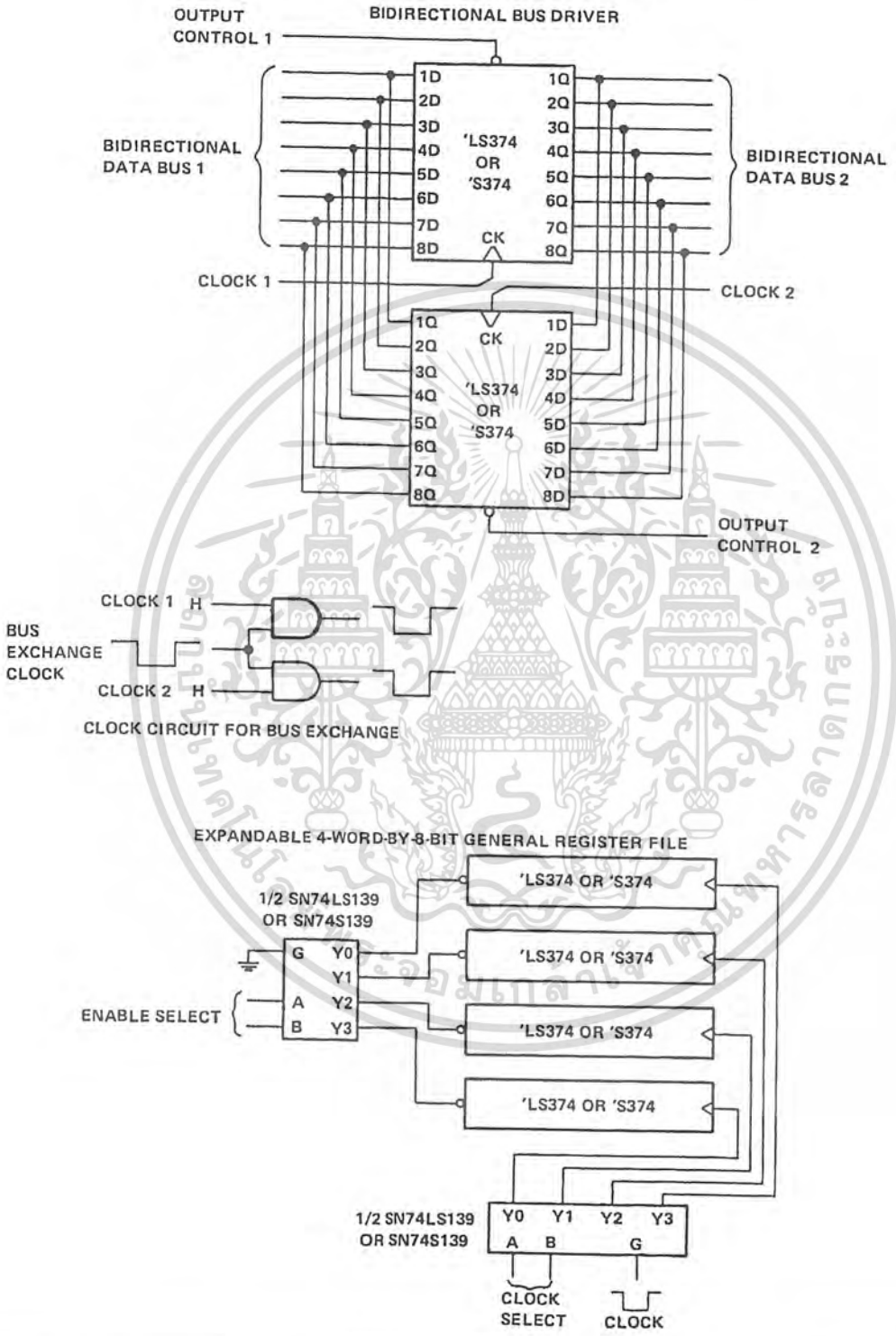
PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	'S373			'S374			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
f_{max}			C_L = 15 pF, R_L = 280 Ω, See Notes 2 and 4				75	100		MHz
t_{PLH}	Data	Any Q			7	12				ns
t_{PHL}					7	12				
t_{PLH}	Clock or enable	Any Q			7	14		8	15	ns
t_{PHL}					12	18		11	17	
t_{PZH}	Output Control	Any Q			8	15		8	15	ns
t_{PZL}					11	18		11	18	
t_{PHZ}	Output Control	Any Q	C_L = 5 pF, R_L = 280 Ω, See Note 3		6	9		5	9	ns
t_{PLZ}					8	12		7	12	

NOTES: 2. Maximum clock frequency is tested with all outputs loaded.
4. Load circuits and voltage waveforms are shown in Section 1.

f_{max} = maximum clock frequency
t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output
t_{PZH} = output enable time to high level
t_{PZL} = output enable time to low level
t_{PHZ} = output disable time from high level
t_{PLZ} = output disable time from low level

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL APPLICATION DATA



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

Copyright © 1999, Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้