แบบจำลองควบคุมสัญญาณไฟจราจร

TRAFFIC LIGHT MODEL CONTROL

โดย

นางสาวทิพย์ งึดเกาะ

นายไพศาล งามจรรยาภรณ์

วัน เดือน ปี 22 ต.ค. 2541

เลขทะเบียน 039096

เลขเรียกหนังสือ T 40336 ท 461 ป

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

039096

แบบจำลองควบคุมสัญญาณไฟจราจร

TRAFFIC LIGHT MODEL CONTROL

โดย

นางสาวทิพย์ งึดเกาะ 37014147

นายไพศาล งามจรรยาภรณ์ 37014310

อาจารย์ที่ปรึกษา

รศ. สมยศ จุณณะปิยะ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2540

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **แบบจำลองควบคุมสัญญาณไฟจราจร**

TRAFFIC LIGHT MODEL CONTROL

ผู้จัดทำ

1. นางสาวทิพย์ งึดเกาะ 37014147
2. นายไพศาล งามจรรยาภรณ์ 37014310

อาจารย์ที่ปรึกษา

(รศ.สมยศ จุณณะปิยะ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบจำลองควบคุมสัญญาณไฟจราจร

TRAFFIC LIGHT MODEL CONTROL

โดย นางสาวทิพย์ งึดเกาะ 37014147
นายไพศาล งามจรรยาภรณ์ 37014310

อาจารย์ที่ปรึกษา รศ.สมชศ จุณณะปิยะ

บทคัดย่อ

โครงงานนี้ทำการเสนอแบบจำลองควบคุมสัญญาณไฟจราจร โดยออกแบบให้ชุดควบคุมสามารถควบคุมสัญญาณไฟจราจรในแต่ละจุด และมีการเชื่อมต่อการควบคุมกันได้ 3 จุดด้วยชุดควบคุมหลัก ซึ่งจะส่งสัญญาณควบคุมไปยังชุดควบคุมอีก 2 จุด และชุดควบคุมย่อยทั้ง 2 จุดจะส่งสถานะปัจจุบันกลับมาให้ชุดควบคุมหลักโดยใช้ชิพ SN 75176A ในการติดต่อระหว่างชุดควบคุมหลักกับชุดควบคุมย่อย และชุดควบคุมหลักจะส่งค่าสถานะปัจจุบันของทั้ง 3 จุดให้แสดงยังหน้าจอคอมพิวเตอร์ โดยใช้ RS-232 module ในการติดต่อระหว่างชุดควบคุมหลักกับคอมพิวเตอร์ ในแบบจำลองควบคุมสัญญาณไฟจราจรนี้มีรูปแบบการเปิดสัญญาณไฟจราจร 8 รูปแบบ และสามารถเลือกควบคุมสัญญาณไฟจราจรได้ทั้งแบบ manual และแบบ auto โดยที่ชุดควบคุมจะประกอบด้วยส่วนของสวิตช์ควบคุม ส่วนควบคุมและประมวลผล ส่วนแสดงผล ส่วนเชื่อมต่อระหว่างชุดควบคุม และส่วนเชื่อมต่อระหว่างชุดควบคุมหลักกับคอมพิวเตอร์

ABSTRACT

This project presents the traffic light model control that is designed to control traffic light in each point and link 3 points together by master controller that will send the control signal to 2 slave controllers, and slave controllers can send the current status to the master controller, the signal linking between master controller and slave controllers use SN 75176A. The master controller has link to computer that use RS-232 module for send the current status of 8 points to display on monitor. In this model has 8 patterns for open the traffic light and can select manual control or automatic control. The control point consists of control switches , control and processing unit , traffic light model display , the linking part between master controller and slave controllers and the linking part between master and computer.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีหรือหลักการ	2
2.1 หลักการทำงานโดยทั่วไปของอุปกรณ์ควบคุมสัญญาณไฟจราจร	2
2.2 โครงสร้างของ MCS – 51	3
2.4 การเชื่อมโยง 8255 เข้ากับ MCS – 51	5
2.5 โหมดการทำงานของ 8255	6
2.6 การเชื่อมโยง 8255 เข้ากับ CPU	7
2.7 การสื่อสารข้อมูลแบบอนุกรม	8
2.8 พอร์ตสื่อสารแบบอนุกรม	11
2.9 มาตรฐานของสายสัญญาณรับ – ส่งข้อมูล	12
บทที่ 3 การออกแบบและการสร้าง	15
3.1 การออกแบบระบบ	15
3.2 ส่วนประกอบหลักของชุดควบคุมสัญญาณไฟจราจร	20
บทที่ 4 การทดลองและผลการทดลอง	25
4.1 การทดลอง4.2	25
4.3 ผลการทดลอง4.4	25
บทที่ 5 บทวิจารณ์และบทสรุป	33
ภาคผนวก	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 แสดงโครงสร้างภายในของ MCS - 51	4
รูปที่ 2.2 การจัดวางขาของ 8051	5
รูปที่ 2.3 การจัดวางขาและโครงสร้างของ 8255	5
รูปที่ 2.4 แสดงการเชื่อมโยง 8255 เข้ากับซีพียู	7
รูปที่ 3.1 ก. ลักษณะของสี่แยกที่ใช้ในการออกแบบ	15
รูปที่ 3.1 ข. สัญญาณไฟจราจรที่ใช้ในการออกแบบ	15
รูปที่ 3.2 ระบบควบคุมสัญญาณไฟจราจร	16
รูปที่ 3.3 รูปแบบการเปิดสัญญาณไฟจราจรทั้ง 8 รูปแบบ	17
รูปที่ 3.4 บล็อคไดอะแกรมของชุดควบคุมสัญญาณไฟจราจร	20
รูปที่ 3.5 การเชื่อมต่อส่วนของสวิตช์	21
รูปที่ 3.6 แสดงการจัดวางขาของชิพ SN 75176	22
รูปที่ 3.7 แสดงการเชื่อมต่อกันระหว่างชุดควบคุมหลักและชุดควบคุมที่แยกอื่น ๆ	23
รูปที่ 3.8 แสดงวงจรที่ใช้ในการควบคุมสัญญาณไฟจราจร	24
รูปที่ 4.1 แสดงค่าแอดเดรส #AAH ที่ตัวแม่ส่งให้กับลูกตัวที่ 1	25
รูปที่ 4.2 แสดงค่าแอดเดรส #AAH ที่ตัวแม่ส่งให้กับลูกตัวที่ 1	26
รูปที่ 4.3 แสดงค่าแอดเดรส #BBH ที่ตัวแม่ส่งให้กับลูกตัวที่ 2	26
รูปที่ 4.4 แสดงค่าแอดเดรส #BBH ที่ตัวแม่ส่งให้กับลูกตัวที่ 2	27
รูปที่ 4.5 แสดงค่าแอดเดรส #01H ที่ตัวลูกส่งให้กับตัวแม่	27
รูปที่ 4.6 แสดงค่าแอดเดรส #01H ที่ตัวลูกส่งให้กับตัวแม่	28
รูปที่ 4.7 แสดงค่าของดาต้า #22H ที่ตัวลูกส่งให้กับตัวแม่	28
รูปที่ 4.8 แสดงค่าของดาต้า #22H ที่ตัวลูกส่งให้กับตัวแม่	29
รูปที่ 4.9 แสดงค่าของดาต้า #33H ที่ตัวลูกส่งให้กับตัวแม่	29
รูปที่ 4.10 แสดงค่าของดาต้า #33H ที่ตัวลูกส่งให้กับตัวแม่	30
รูปที่ 4.11 แสดงค่าแอดเดรส #00H ที่ตัวแม่ส่งให้กับคอมพิวเตอร์	30
รูปที่ 4.12 แสดงค่าดาต้า #A1H ที่ตัวแม่ส่งให้กับคอมพิวเตอร์	31
รูปที่ 4.13 แสดงค่าดาต้า #22H ที่ตัวแม่ส่งให้กับคอมพิวเตอร์	31
รูปที่ 4.14 แสดงค่าดาต้า #33H ที่ตัวแม่ส่งให้กับคอมพิวเตอร์	32
รูปที่ 4.15 แสดงผลหน้าจอคอมพิวเตอร์	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 ตารางของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวในตระกูล 51	3
ตารางที่ 2.2 สรุปโหมดการทำงานของ 8255	6
ตารางที่ 2.3 แสดงแอดเดรสของ 8255	8
ตารางที่ 2.4 แสดงอัตราเร็วในการรับส่งข้อมูล	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

เนื่องจากในปัจจุบันนี้ ปัญหาการจราจรนับว่าเป็นปัญหาที่ต้องแก้ไขอย่างเร่งด่วนที่สุดเนื่องจากว่าจะทวีความรุนแรงมากขึ้นทุกวัน ระบบการควบคุมการจราจรที่มีประสิทธิภาพจึงเป็นสิ่งจำเป็นอย่างยิ่งที่จะช่วยบรรเทาปัญหาการจราจรที่เป็นอยู่ในทุกวันนี้ให้บรรเทาลง ซึ่งสัญญาณไฟจราจรที่ใช้กันอยู่ในปัจจุบันนี้เป็นระบบขนาดเล็กที่ทำงานควบคุมได้แค่เพียงสี่แยกเดียว โดยใช้ตำรวจจราจรคอยควบคุมหรือทำงานตามโปรแกรมที่ตั้งไว้แล้วโดยอัตโนมัติ ซึ่งปัญหาที่เกิดขึ้นนั้นก็คือจะสามารถควบคุมการจราจรได้เฉพาะสี่แยกใดสี่แยกหนึ่งเท่านั้นโดยไม่มีการประสานงานกัน ระบบควบคุมสัญญาณไฟจราจรที่ทำขึ้นมานี้จึงประดิษฐ์ขึ้นมาเพื่อให้ทำงานได้ดีกว่าระบบขนาดเล็กที่ใช้อยู่อย่างในปัจจุบัน ซึ่งแม้ว่าประสิทธิภาพจะไม่ดีเทียบเท่ากับระบบขนาดใหญ่ที่มีอยู่แล้วแต่ก็ช่วยลดปัญหาที่มีอยู่ได้บ้าง เนื่องจากว่าอุปกรณ์ที่ประดิษฐ์ขึ้นมานี้จะมีระบบการทำงานที่เพิ่มเติมขึ้นมาจากระบบขนาดเล็กโดยสามารถควบคุมการจราจรได้ 3 สี่แยก เมื่อแยกที่มีเครื่องควบคุมสัญญาณหลักทำการปล่อยรถไปยังแยกถัดไปแล้ว เครื่องควบคุมหลักจะทำการสร้างสัญญาณและส่งสัญญาณไปแจ้งแก่แยกถัดไปว่าขณะนี้ได้ทำการปล่อยรถออกไปแล้ว ให้แยกถัดไปทำการนับเวลาซึ่งรถจะใช้ในการวิ่งระหว่างแยกหนึ่งไปยังอีกแยกหนึ่งแล้วเปิดสัญญาณเป็นไฟเขียวเพื่อให้รถสามารถวิ่งทะลุผ่านไปได้โดยไม่มีการติดสัญญาณไฟยังแยกนี้อีก โดยตัวควบคุมหลักจะทำการส่งสัญญาณแจ้งไปยังแยกถัดไปต่าง ๆ ข้างหน้าได้ถึง 2 แยกให้รับรู้และสามารถทำงานให้สอดคล้องกับแยกแรกในการเปิดสัญญาณไฟให้รถวิ่งผ่านออกไปตลอดโดยไม่ติดขัดได้ โดยในอุปกรณ์ควบคุมสัญญาณไฟจราจรที่ประดิษฐ์ขึ้นมานี้จะมีระบบการทำงานเหมือนกับไฟจราจรที่มีใช้กันอยู่ทั่วไปแต่จะสามารถเชื่อมต่อไปยังสี่แยกต่างๆ นอกจากนี้ยังได้นำสัญญาณนาฬิกาเข้ามาช่วยควบคุมระบบเพื่อให้อุปกรณ์สามารถทำงานได้ตามเวลาที่เราตั้งโปรแกรมไว้ ในกรณีที่ไม่มีคนคอยควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีหรือหลักการ

2.1 หลักการทำงานโดยทั่วไปของอุปกรณ์ควบคุมสัญญาณไฟจราจร

อุปกรณ์ควบคุมสัญญาณไฟจราจรนี้สามารถทำการควบคุมเชื่อมต่อกันได้ 3 แยก โดยมีอุปกรณ์ควบคุมหลัก 1 ตัวเป็นตัวคอยควบคุม เมื่อแยกที่มีเครื่องควบคุมสัญญาณหลักทำการปล่อยรถไปยังแยกถัดไปแล้ว เครื่องควบคุมหลักจะทำการสร้างสัญญาณและส่งสัญญาณไปแจ้งแก่แยกถัดไป ว่าในขณะนี้ได้ทำการปล่อยรถไปแล้ว ให้แยกถัดไปทำการนับเวลาซึ่งรถจะใช้ในการวิ่งระหว่างแยกหนึ่งไปยังอีกแยกหนึ่งแล้วเปิดสัญญาณเป็นไฟเขียวเพื่อให้รถสามารถวิ่งทะลุผ่านไปได้ โดยไม่มีการติดสัญญาณไฟยังแยกนี้อีก โดยตัวควบคุมหลักจะทำการส่งสัญญาณแจ้งไปยังแยกต่างๆ ข้างหน้าได้ถึง 3 แยกให้รับรู้และสามารถทำงานให้สอดคล้องกับแยกแรกในการเปิดสัญญาณไฟให้รถวิ่งผ่านออกไปตลอดโดยไม่ติดขัดได้ ภายในอุปกรณ์ควบคุมสัญญาณไฟจราจรนั้นจะมีส่วนประกอบต่างๆ ที่ใช้ควบคุมการทำงานแบ่งเป็นส่วนควบคุมและประมวลผล(Central Processing- Unit) กับส่วนที่เป็นส่วนแสดงผล (Display Unit)

ส่วนควบคุมและประมวลผลของอุปกรณ์ควบคุมสัญญาณไฟจราจรนั้น เราใช้ไมโครคอนโทรลเลอร์ชิพเดี่ยวเบอร์ 8031 ทำหน้าที่เป็นตัวควบคุมและประมวลผลการทำงาน ในการควบคุมอุปกรณ์ควบคุมสัญญาณไฟจราจรนั้น หน่วยประมวลผลจะต้องอ่านข้อมูลจากหน่วยความจำที่ใช้เก็บคำสั่งจากภายนอก หน่วยความจำแบบนี้จะต้องเป็นแบบอ่านอย่างเดียว (Read Only Memory : ROM) ซึ่งเมื่อปิดไฟแล้วข้อมูลก็ไม่มีการสูญหาย ในการควบคุมการทำงานนั้น โปรแกรมที่ใช้ในการควบคุมจะต้องถูกเขียนลงบนหน่วยความจำโปรแกรม (Program Memory) ซึ่งได้แก่ อีพรอม (EPROM) 8031 จะอ่านโปรแกรมเหล่านี้ออกมาแล้วทำการปฏิบัติงานตามที่โปรแกรมสั่งงานเอาไว้ ในการควบคุมสัญญาณไฟของอุปกรณ์ควบคุมสัญญาณไฟจราจรจะทำงานผ่านส่วนอินพุทเอาท์พุท ซึ่งอุปกรณ์ควบคุมอินพุทเอาท์พุทนี้ เราใช้ไอซีเบอร์ 8255 ในการควบคุมสัญญาณไฟ 1 ดวงของ 8255 นั้นจะใช้ข้อมูลจำนวน 1 บิตในการควบคุมไฟแต่ละดวง โดยจะทำงานตามโปรแกรมการสั่งงานผ่านไมโครคอนโทรลเลอร์เบอร์ 8031 นอกจากนี้ในการทำงานของเครื่องควบคุมสัญญาณไฟจราจรในโหมดอัตโนมัตินั้น จะต้องทำงานตามที่ได้มีการกำหนดไว้ และยังสามารถตั้งเวลาของการเปิดปิดระหว่างไฟแดง เหลือง และเขียวได้ เนื่องจากว่าการควบคุมการจราจรในแต่ละช่วงเวลาของแต่ละวันนั้นต้องการโปรแกรมที่ใช้ในการเปิดปิดสัญญาณที่ต่างกัน เช่นในช่วงเช้าของวันธรรมดา เส้นทางหลักจะมีการจราจรหนาแน่นกว่าเส้นทางรอง แต่ในช่วงเวลาสายๆ ทั้งสองเส้นทางอาจจะมีความหนาแน่นของการจราจรพอๆ กัน ดังนั้นในช่วงเช้าจะต้องมีการเปิดปิดสัญญาณไฟจราจรรูปแบบหนึ่ง ในขณะที่ในช่วงสายต้องใช้รูปแบบสัญญาณไฟจราจรอีกรูปแบบหนึ่ง เวลาที่จะต้องนำมาใช้งานจะต้องเป็นเวลาที่แท้จริง ดังนั้นจะต้องมีอุปกรณ์ที่ทำหน้าที่ให้สัญญาณนาฬิกาเพื่อให้ได้เวลาตามความเป็นจริงในการนำไปใช้งาน เพื่อให้ผู้ควบคุมสะดวกต่อการใช้งานและสามารถตั้งเวลาได้ ซึ่งเราได้นำอุปกรณ์ไอซีสำเร็จรูป เบอร์ DS1202 มาใช้ในการให้สัญญาณนาฬิกา

ส่วนแสดงผลนั้น จะแสดงผลออกทางหน้าจอมอนิเตอร์ โดยตัวควบคุมหลักจะทำการเชื่อมต่อกับคอมพิวเตอร์ทางพอร์ตอนุกรม เพื่อที่จะนำค่าของตัวลูกมาแสดงผลทางหน้าจอว่าในขณะนี้นั้นตัวลูกที่แยกไหนมีรูปแบบการเดินรถแบบใด นอกจากนี้ที่แต่ละแยกนั้นจะใช้แบบจำลองที่ประกอบด้วย LED ซึ่งต่อกับ 8255 ของบอร์ดควบคุมเพื่อแสดงรูปแบบการเดินรถของแยกนั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โครงสร้างของ MCS-51

ไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล MCS-51 นี้ผลิตโดยบริษัทอินเทลมีอยู่ด้วยกันหลายเบอร์ ซึ่งมีรายละเอียดดังตารางที่ 2.1

ตารางที่ 2.1 ตารางของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวในตระกูล 51

Device	ROMless Version	EPROM Version	ROM Bytes	RAM Bytes	8-Bit I/O Ports	16-Bit Timer/ Counters	Programmable Counter Array (PCA)	UART	Serial Expansion Port (SEP)	Global Serial Channel (GSC)	DMA Channels	A/D Channels	Interrupt Sources/ Vectors	Power Down and Idle Modes
8051	8031	—	4K	128	4	2		✓					6/5	
8051AH	8031AH	8751H 8751BH	4K	128	4	2		✓					6/5	
8052AH	8032AH	8752BH	8K	256	4	3		✓					8/6	
80C51BH	80C31BH	87C51	4K	128	4	2		✓					6/5	✓
80C52	80C32	—	8K	256	4	3		✓					8/6	✓
83C51FA	80C51FA	87C51FA	8K	256	4	3	✓	✓					14/7	✓
83C51FB	80C51FA	87C51FB	16K	256	4	3	✓	✓					14/7	✓
83C152JA	80C152JA	—	3K	256	5	2		✓		✓	2		19/11	✓
—	80C152JB	—	—	256	7	2		✓		✓	2		19/11	✓
83C152JC	80C152JC	—	8K	256	5	2		✓		✓	2		19/11	✓
—	80C152JD	—	—	256	7	2		✓		✓	2		19/11	✓
83C452	80C452	87C452P	8K	256	5	2		✓					9/8	✓

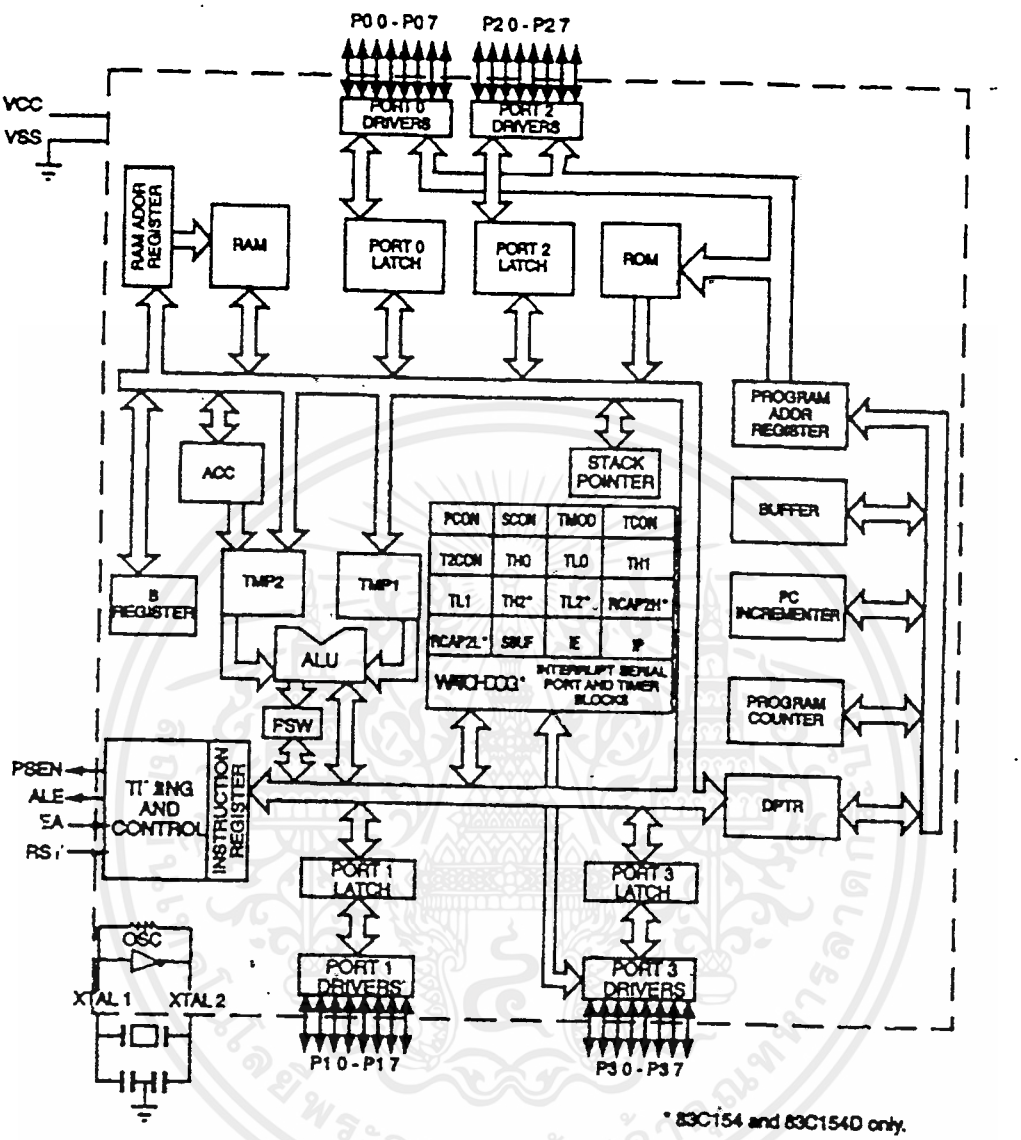
คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS -51

- ต้องการแหล่งจ่ายไฟ +5 โวลต์ ชุดเดียว
- มีหน่วยความจำโปรแกรม (Program Memory) ขนาด 5 กิโลไบต์ สำหรับเบอร์ 8051 และสำหรับเบอร์ 8031และ 8032 นั้นไม่มีหน่วยความจำชุดนี้ ส่วน 8052 มีหน่วยความจำถึง 8 กิโลไบต์
- มีหน่วยความจำสำหรับเก็บข้อมูล (Data Memory) ขนาด 128 ไบต์ สำหรับ 8052 มีถึง 256 ไบต์
- หน่วยความจำสำหรับโปรแกรมและดาต้า (Program Memory and Data Memory) แยกจากกันอย่างละ 64 กิโลไบต์
- คำสั่งที่ใช้เวลาน้อยที่สุดประมาณ 1 ไมโครวินาที เมื่อทำงานที่ความถี่ 12 เมกะเฮิร์ต
- มีไทม์เมอร์/เคาน์เตอร์ ขนาด 16 บิต 2 ชุด (สำหรับ 8052 มี 3 ชุด) ทำงานได้ 4 โหมด
- รับอินเตอร์รัปต์ได้ 6 แหล่ง 5 เวคเตอร์
- มีพอร์ตรับส่งข้อมูลอนุกรม (UART) 2 พอร์ต ทั้งรับและส่งในเวลาเดียวกัน (Full Duplex) เลือกรูปแบบการส่งข้อมูลได้ 4 โหมด
- มีคำสั่งในการทำ AND , OR หรือ COMPLEMENT ได้ทั้งแบบ 8 บิต และ 1 บิต

2.3 โครงสร้างภายในของ 8051

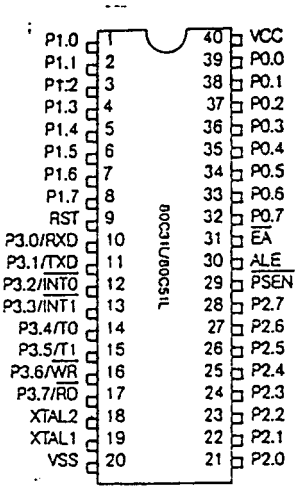
MCS-51 ใช้เทคโนโลยีการผลิตแบบ NMOS หรือ CMOS เบอร์ 8032 และ 8052 จะมี ROM BASIC อยู่ภายในจึงสะดวกสำหรับโปรแกรมเมอร์ที่จะเขียนโปรแกรมด้วยภาษาเบสิก โครงสร้างภายในสำหรับเบอร์ 8051 แสดงในรูปที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงโครงสร้างภายในของ MCS - 51

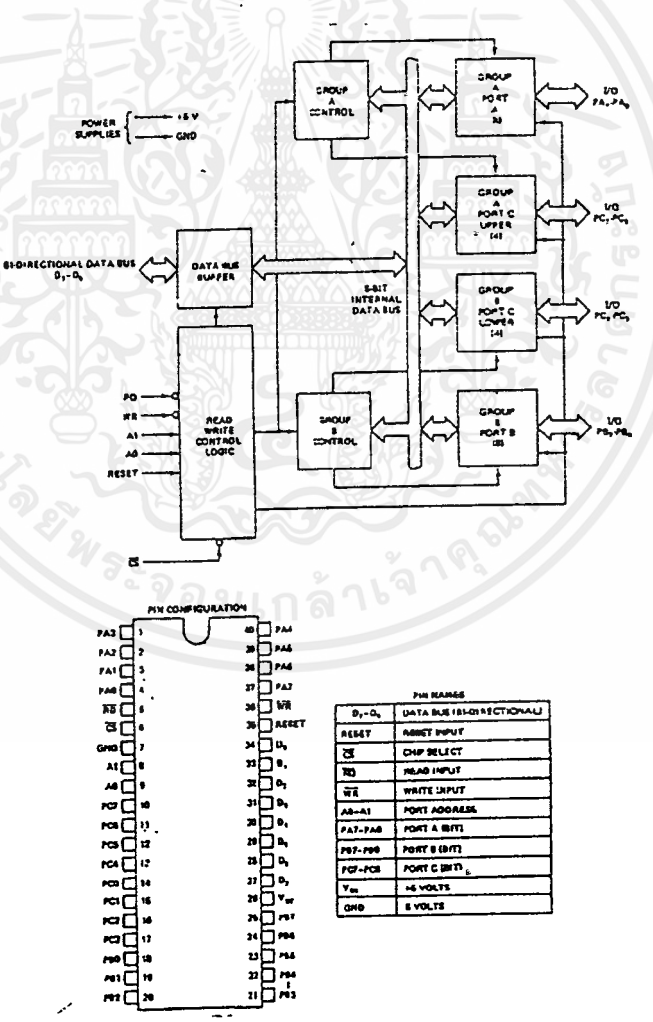
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงการจัดวางขาของ 8051

2.4 การเชื่อมโยง 8255 กับ MCS-51

8255 เป็นชิพขนาด 40 ขา มีอยู่ 3 พอร์ตคือ A, B และ C เป็นพอร์ต 8 บิต ที่สามารถโปรแกรมให้เป็นอินพุทหรือเอาท์พุทก็ได้ โดยที่พอร์ต C ยังแบ่งเป็น 4 บิทล่างและ 4 บิทบน โดยมีโครงสร้างตามรูปที่ 2.3



รูปที่ 2.3 แสดงการจัดวางขาและโครงสร้างของ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 โหมดการทำงานของ 8255

การทำงานของ 8255 มีอยู่ด้วยกัน 3 โหมดดังนี้

โหมด 0 มีการทำงานแบบ BASIC I/O ไม่มี handshake

โหมด 1 โหมดนี้ใช้พอร์ต A และ B ในการรับส่งข้อมูล และใช้พอร์ต C ในการตรวจสัญญาณ(handshake)

โหมด 2 โหมดนี้ใช้พอร์ต A ในการรับส่งข้อมูล 2 ทิศทางและใช้พอร์ต B ในการรับส่งข้อมูล และใช้ พอร์ต C บิต 0,1และ 2 ในการรับส่งข้อมูลบิตและบิต 4,5 และ 6 เป็นสัญญาณ handshake

ตารางที่ 2.2 สรุปโหมดการทำงานของ 8255

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA_0	IN	OUT	IN	OUT	↔	
PA_1	IN	OUT	IN	OUT	↔	
PA_2	IN	OUT	IN	OUT	↔	
PA_3	IN	OUT	IN	OUT	↔	
PA_4	IN	OUT	IN	OUT	↔	
PA_5	IN	OUT	IN	OUT	↔	
PA_6	IN	OUT	IN	OUT	↔	
PA_7	IN	OUT	IN	OUT	↔	
PB_0	IN	OUT	IN	OUT	—	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	—	
PB_2	IN	OUT	IN	OUT	—	
PB_3	IN	OUT	IN	OUT	—	
PB_4	IN	OUT	IN	OUT	—	
PB_5	IN	OUT	IN	OUT	—	
PB_6	IN	OUT	IN	OUT	—	
PB_7	IN	OUT	IN	OUT	—	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O	
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O	
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O	
PC_3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$	
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$	
PC_5	IN	OUT	IBF_A	I/O	IBF_A	
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$	
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$	

สัญญาณต่างๆ ของ 8255

D0 - D7 บัสข้อมูลเชื่อมโยงกับ CPU

A0 -A7 ใช้เลือกพอร์ต A, B, C และพอร์ตควบคุม

RESET เมื่อขานี้ได้รับสัญญาณกระตุ้นลอจิก 1 จะทำให้ 8255 ถูกรีเซ็ตมีผลทำให้ทุกพอร์ตเป็นอินพุททันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PA0 - PA7 เป็นพอร์ตขนาน 8 บิต

PB0 - PB7 เป็นพอร์ตขนาน 8 บิต

PC0 - PC7 เป็นพอร์ตขนาน 8 บิต

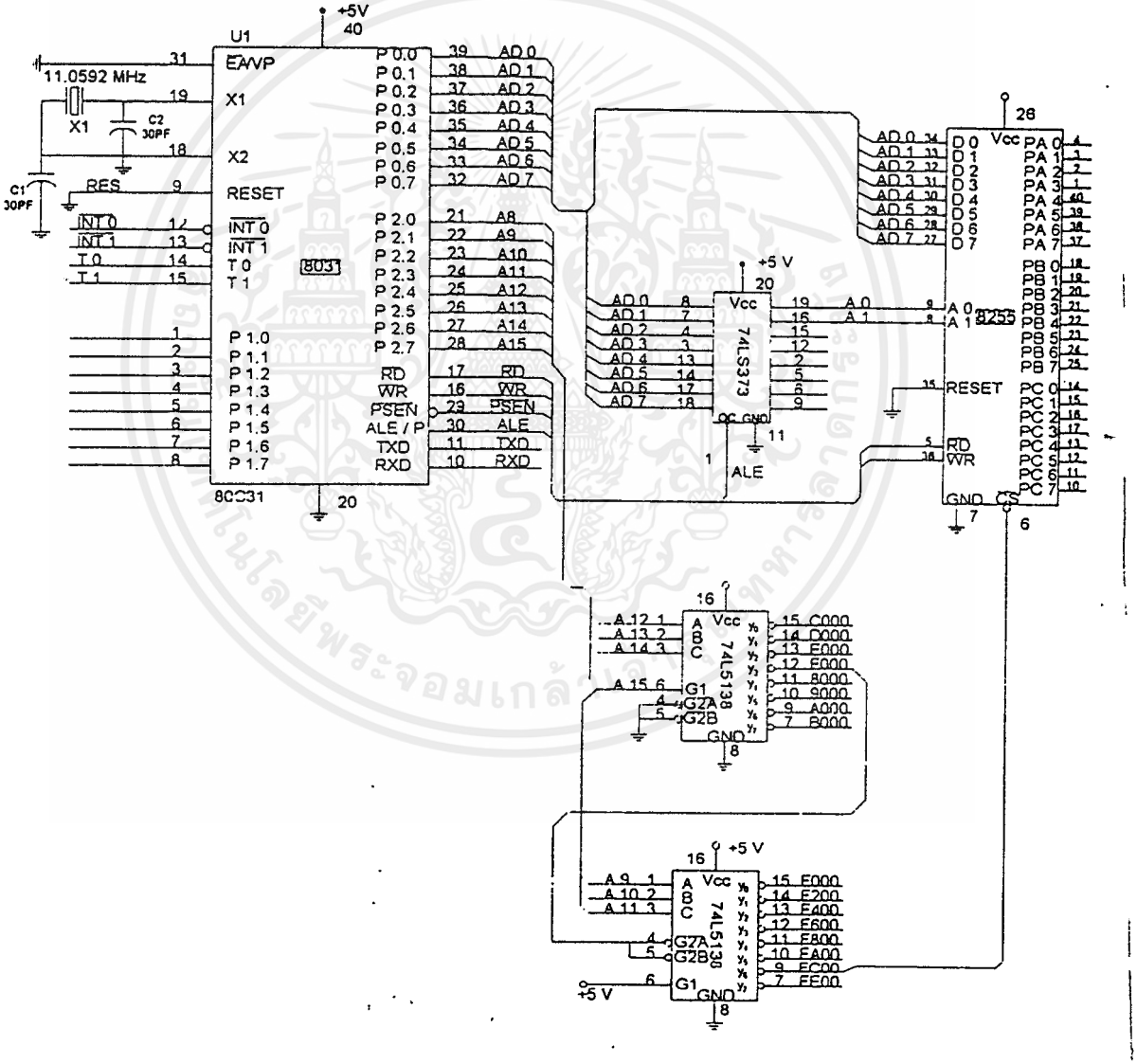
$\overline{RD}$ ในการอ่านข้อมูลที่พอร์ตขนานของ 8255 ต้องทำให้เป็นลอจิก 0 พร้อมกับ $\overline{CS}$

$\overline{WR}$ ในการเขียนข้อมูลหรือโปรแกรมลงบน 8255 ต้องทำให้ขานี้เป็นลอจิก 0 พร้อมกับ $\overline{CS}$

$\overline{CS}$ เป็นขาเลือกชิพ 8255 ได้ ขานี้มักจะต่อกับ I/O DECODER

2.6 การเชื่อมโยง 8255 เข้ากับ CPU

การเชื่อมโยง 8255 เข้ากับ CPU แสดงดังรูปที่ 2.4 และ อินพุทเอาท์พุทแอดเดรสแสดงดังตารางที่ 2.3



รูปที่ 2.4 แสดงการเชื่อมโยง 8255 เข้ากับซีพียู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 แสดงแอดเดรสของ 8255

I/O ADDRESS	8255 PORT
FC00h	A
FC01h	B
FC02h	C
FC03h	CONTROL

2.7 การสื่อสารข้อมูลแบบอนุกรม

การสื่อสารข้อมูลแบบอนุกรมเป็นการรับหรือส่งข้อมูลในลักษณะกลุ่มของบิตคราวละหนึ่งบิตเรียงลำดับเรื่อยไปจนสิ้นสุด การสื่อสารแบบนี้จะมีข้อแตกต่างจากการสื่อสารแบบขนานเป็นอย่างมาก เนื่องจากข้อมูลมีการโอนย้ายมาพร้อมกัน จึงมีความจำเป็นต้องใช้จำนวนเส้นสัญญาณมากขึ้นตามจำนวนบิตของข้อมูลด้วย ในขณะที่การสื่อสารแบบอนุกรมนั้นต้องการเส้นสัญญาณเพียงสองหรือสามเส้นเท่านั้น ดังนั้นการสื่อสารแบบขนานจึงไม่เหมาะสมในการสื่อสารกับอุปกรณ์ภายนอกเป็นระยะทางไกลๆ เพราะจะทำให้สิ้นเปลืองค่าใช้จ่ายมาก

2.7.1 ความเร็วของการสื่อสารข้อมูลอนุกรม

เนื่องจากการสื่อสารแบบอุกรมเป็นการรับส่งข้อมูลในลักษณะกลุ่มของบิตข้อมูล(Bit Stream) ดังนั้นจึงต้องให้ความสนใจในการพิจารณาถึงเรื่องอัตราความเร็วในการรับ - ส่งบิตเหล่านี้ เป็นลำดับแรก โดยทั่วไปมักจะระบุกันในหน่วยของจำนวนบิตข้อมูลภายในเวลาหนึ่งวินาที เรียกว่า อัตราบอด อัตราบอดตามค่ามาตราฐานเหล่านี้ได้แก่ 110, 150, 300, 1200, 2400, 4800, 9600, 19200 บอด ข้อมูลทั้ง 8 บิตนี้หากว่าทำการส่งออกมาด้วยอัตรา 9600 บอด จะใช้เวลาในการส่งข้อมูลหนึ่งบิต มีค่าเท่ากับ 1/9600 หรือ 104 uS และเวลาที่ใช้ส่งข้อมูลทั้ง 8 บิตมีค่าเท่ากับ 832 uS

2.7.2 รูปแบบของการส่งข้อมูลอนุกรม

การสื่อสารอนุกรมแบบอะซิงโครนัส จะใช้การแปลงข้อมูลขนานให้เป็นอนุกรม แล้วเติมบิตบางอย่างร่วมไปกับการส่งข้อมูลจริงซึ่งได้แก่

1. บิตเริ่มต้น (Start Bit)

   บิตเริ่มต้นมีหน้าที่สำหรับการบ่งบอกให้ทราบถึงตำแหน่งจุดเริ่มต้นก่อนบิตข้อมูล ตามปกติแล้วค่าของบิตเริ่มต้นจะเป็นระดับลอจิกต่ำ

2. บิตแสดงสถานะความเป็นเลขคู่หรือเลขคี่ (Parity Bit)

   บิตนี้มีหน้าที่เพื่อการตรวจสอบความถูกต้องของข้อมูล โดยทั่วไปมักเรียกว่า บิตพาริตี้ และจะนำไปต่อท้ายบิตข้อมูล ค่าของบิตนี้ขึ้นอยู่กับจำนวนค่าของบิตที่เป็นหนึ่ง ซึ่งจะเป็นได้สองลักษณะคือพาริตี้คู่ (Even- Parity) หรือพาริตี้คี่ (Odd Parity)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. บิตสุดท้าย (Stop Bit)

เป็นบิตที่เพิ่มขึ้นเพื่อระบุถึงขอบเขตการสิ้นสุดของกลุ่มบิตข้อมูล บิตสุดท้ายนี้สามารถโปรแกรมได้ คือ 1 บิต 1 1/2 บิต และ 2 บิต

2.7.3 การส่งข้อมูลอนุกรมของ 8051

พอร์ตอนุกรมของ 8051 มีโครงสร้างการทำงานในแบบที่เรียกว่า ฟูลดูเพล็กซ์ (Full Duplex) ในการรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน โดยทางด้านวงจรของตัวส่ง (Transmitter) ประกอบด้วยข้อมูลออกไปยังพอร์ตอนุกรม ทางขาสัญญาณ TxD (พอร์ต 3.1) ส่วนวงจรด้านตัวรับ(Receiver) ประกอบด้วย SBUF เช่นเดียวกันข้อมูลอนุกรมจะรับเข้ามาทางขาสัญญาณ RxD (พอร์ต 3.0)

พอร์ตอนุกรมของ 8051 สามารถโปรแกรมการทำงานได้หลายโหมดด้วยกัน โดยเลือกที่บิต SM0 และ SM1 ซึ่งอยู่ในรีจิสเตอร์ควบคุม SCON การทำงานทั้ง 4 โหมดของพอร์ตอนุกรมมีดังนี้

โหมด 0 ใช้รับส่งข้อมูล 8 บิต โดยการส่งจะเลื่อนออกทีละบิต โดยส่งบิต D0 ออกไปก่อนทางขา RxD และไม่มีการส่ง บิตเริ่มต้น แต่จะส่งคล็อคทางขา TxD ความเร็ว 1/12 เท่า ของซีพียูคล็อค

โหมด 1 ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART(Universal AsynchronousReceiver/Transmitte) โดยส่งแบบ 10 บิต ข้อมูล 8 บิต บิตเริ่มต้น 1 บิตและบิตสุดท้ายอีก 1 บิต และสามารถเปลี่ยนแปลงอัตราความเร็วในการส่งข้อมูลได้ โดยขึ้นอยู่กับบิต SMOD ใน PCON และอัตราโอเวอร์โฟลว์ของ Timer1

โหมด 2 ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART โดยการใช้กลุ่มข้อมูลแบบ 11 บิตและกำหนดอัตราความเร็วในการส่งข้อมูลเท่ากับ 1/32 และ 1/64 ของซีพียูคล็อคโดยการโปรแกรมที่บิต SMOD ใน PCON

โหมด 3 ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART โดยการใช้กลุ่มข้อมูลแบบ 11 บิตและสามารถเปลี่ยนแปลงอัตราความเร็วในการส่งข้อมูลได้ โดยควบคุมที่บิต SMOD และอัตราโอเวอร์โฟวล์ของ Timer1 นอกจากนี้โหมด 2 และ 3 ยังมีการดำเนินการอีกแบบหนึ่ง โดยสามารถนำมาใช้ประโยชน์ในการสื่อสารข้อมูลแบบที่มีไมโครโปรเซสเซอร์หลายตัวทำงานร่วมกันได้ ซึ่งมีชื่อ เรียกว่า มัลติโปรเซสเซอร์โหมด (Multi processor Mode)

2.7.4 กระบวนการรับและส่งข้อมูลอนุกรมของ 8051

การส่งข้อมูลออกทางพอร์ตอนุกรมของ 8051 จะเริ่มต้นเมื่อมีการเขียนข้อมูลลงใน SBUF ข้อมูลนี้จะถูกเลื่อนทีละบิต และส่งออกไปภายนอกโดยอัตโนมัติ เมื่อข้อมูลเหล่านี้ได้ส่งออกครบแล้วจะทำให้ค่าของแฟล็ก TI ให้เป็น 1 เพื่อแจ้งให้ทราบว่าขณะนี้ SBUF ว่างและพร้อมที่จะส่งข้อมูลไบต์ต่อไปแล้ว ในกรณีที่ใช้เขียนข้อมูลใหม่ลงในรีจิสเตอร์ SBUF โดยไม่รอให้แฟล็ก TI มีค่าเป็น 1 ก่อนจะมีผลทำให้ข้อมูลที่ส่งไปผิดพลาดได้

สำหรับการรับข้อมูลจากพอร์ตอนุกรมจะต้องเริ่มต้นโดยการกำหนดเซ็ตค่าดังนี้ REN (Receiver Enable) ให้มีค่าเป็นหนึ่งก่อน หลังจากนั้นเมื่อมีข้อมูลภายนอกถูกส่งเข้ามายัง 8051 ทีละบิตจนครบและเมื่อบิตสุดท้ายถูกเลื่อนเข้ามาเรียบร้อยแล้ว ข้อมูลนั้นจะถูกย้ายมาเก็บไว้ยังรีจิสเตอร์ SBUF และแฟล็ก RI จะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าเป็น 1 (ถูกเซ็ต) หลังจากนั้นก็จะเกิดการอินเตอร์รัปต์ขึ้น เนื่องจากในโครงงานนี้ใช้การรับส่งข้อมูลอนุกรมโหมด 1 ดังนั้นจึงจะกล่าวถึงโหมด 1 เท่านั้น

การรับส่งข้อมูลพอร์ตอนุกรมโหมด 1

การทำงานในโหมด 1 เป็นการสื่อสารข้อมูลอนุกรมจำนวน 10 บิต ประกอบด้วยบิตเริ่มต้น 1 บิต บิตข้อมูลจำนวน 8 บิต และบิตสุดท้ายจำนวน 1 บิต โดยข้อมูลจะถูกส่งออกทาง TxD และรับเข้ามาทางขาสัญญาณ RxD ในส่วนของข้อมูล 8 บิตที่ได้รับหรือทำการส่งออกจะเป็นบิตนัยสำคัญต่ำเป็นลำดับแรก ทางฝ่ายรับค่าของสต๊อปบิตจะส่งเข้ามาจัดเก็บไว้ในบิต RB8 ภายในรีจิสเตอร์ SCON สำหรับอัตราความเร็วในการส่งข้อมูลของโหมด 1 นั้นสามารถกำหนดเลือกได้จากไทม์เมอร์ 1 โดยการใช้ไทม์เมอร์ 1 ทำหน้าที่เป็นตัวกำเนิดอัตราการส่งข้อมูล และใช้แฟล็กที่เกิดจากการโอเวอร์โฟลว์ของไทม์เมอร์ 1 โดยโปรแกรมไทม์เมอร์ 1 ทำงานในโหมด 2 (8 บิต automatic reload) และการเปลี่ยนแปลงความเร็วดูได้จากตารางที่ 2.4 หรือคำนวณจากสูตรดังนี้

เมื่อกำหนดให้ Timer1 ทำงานโหมด 2 Auto Reload

$$\text{บอดเรต (โหมด 1)} = \frac{(2^{SMOD})(\text{ความถี่ออสซิเลเตอร์ที่ใช้})}{(32)(12)(256 - TH1)}$$

โดย SMOD เป็นค่าของบิตภายในรีจิสเตอร์ PCON (มีค่าเป็น 0 หรือ 1)

ตารางที่ 2.4 แสดงอัตราเร็วในการรับ - ส่งข้อมูลอนุกรม

Baud Rate	Fosc	SMOD	TIMER 1		
			$C/\overline{T}$	MODE	Reload Value
(MODE 0) Max : 1 MHz	12 MHz	X	X	X	X
(MODE 2) Max : 375 KHz	12 MHz	1	X	X	X
(MODE2) Min : 187.5 KHz	12 MHz	0	X	X	X
MODE 1, 3 : 62.5 K	12 MHz	1	0	2	FFH
19.2 K	11.059 MHz	1	0	2	FDH
9.6 K	11.059 MHz	0	0	2	FDH
4.8 K	11.059 MHz	0	0	2	FAH
2.4 K	11.059 MHz	0	0	2	F4H
1.2 K	11.059 MHz	0	0	2	E8H
137.5	11.059 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 พอร์ตสื่อสารแบบอนุกรม

ในเครื่องคอมพิวเตอร์ทั่วไปจะมีอะแดปเตอร์สื่อสารแบบอะซิงโครนัส (Asynchronous) เป็นการ์ดอะแดปเตอร์แบบอนุกรมและขนานในการ์ดอันเดียวกัน สำหรับอะแดปเตอร์แบบอนุกรมจะแบ่งได้เป็น 3 ประเภทดังนี้

- ประเภทแรกเป็นอะแดปเตอร์ที่ใช้ชิพ (Chip) เบอร์ 8250 หรือ 16450 UART เป็นอะแดปเตอร์ที่เป็นพื้นฐานที่สุดสำหรับการสื่อสารทั่วไป

- ประเภทที่สองคล้าย ๆ กับประเภทแรกแต่จะใช้ชิพเบอร์ 16550 และ FIFO (First In First Out) บัฟเฟอร์ซึ่งมีประสิทธิภาพมากกว่า ใช้สำหรับการสื่อสารโมเด็มความเร็วสูงและสภาวะแวดล้อม การทำงานแบบหลาย ๆ งาน (Multitasking Environments) แต่จะมีราคาแพงกว่าประเภทแรก

-ประเภทที่สามประกอบด้วยอะแดปเตอร์แบบพิเศษคือตัวควบคุมพอร์ทอนุกรม(Serial PortController) และเฮส์9 เอนฮานซด์ซีเรียลอินเตอร์เฟซ (Hayes Enhanced Serial Interface : HESI) สำหรับประเภทที่สามนี้จะสนับสนุนการถ่ายเทข้อมูลแบบ DMA (Direct Memory Acess)และมีโฟลว์คอนโทรล (Flow Control) ของตัวเอง HESI เป็นโคโพรเซสเซอร์ (Coprocessor)ทางการสื่อสารที่สมบูรณ์สามารถใช้ไมโครโพรเซสเซอร์ของตนเองในการจัดการสื่อสารโดยไม่ขึ้นอยู่กับประเภทของคอมพิวเตอร์

พอร์ตอนุกรมประกอบไปด้วยโครงสร้างหลายอย่างด้วยกัน คือ

2.8.1 ไอโอแอดเดรส (I/O Address) ในคอมพิวเตอร์จะมีไอโอแอดเดรสสำหรับใช้อ้างถึงอะแดปเตอร์แบบอนุกรมอยู่ 4 ช่วง ดังตารางที่ 2.5 ไอโอแอดเดรสดังกล่าวเชื่อมต่อกับ UART (Universal Asyncronous Receiver transmitter) ซึ่งทำหน้าที่ในการติดต่อสื่อสารแบบอะซิงโครนัสและอินเตอร์เฟสแบบ RS -232 เมื่อเราทำการติดตั้งพอร์ตอนุกรมอันแรก (COM1) จะมีฐานแอดเดรสอยู่ที่ 03F8H โดยเราสามารถอ้างถึงรีจิสเตอร์ใน UART จากฐานของแอดเดรสนี้ เช่น รีจิสเตอร์ตัวที่สี่ของ COM1 จะมีแอดเดรสตรงกับพีซี I/O แอดเดรสที่ 03F8H+4 = 03FCH

หมายเหตุ COM1, COM2, COM3 และ COM4 เป็นชื่อของอุรปกรณ์ (Devive) ที่ตั้งโดยระบบ ปฏิบัติการ (MS- DOS) สำหรับพอร์ตอนุรกรมอันที่ 1 ถึงอันที่ 4 ตามลำดับที่ติดตั้งอยู่ ในระบบ เพราะโปรแกรมส่วนใหญ่จะติดต่อกับฮาร์ดแวร์โดยตรง ดังนั้นชื่อเหล่านี้จึง เป็นที่รู้จักโดยทั่วไปเมื่ออ้างถึงการสื่อสารแบบอะซิงโครนัส

2.8.2 การเชื่อมต่อกับพีซีบัส (Bus Interface) การอ่านและการเขียนข้อมูลผ่านรีจิสเตอร์ของ UART ไอโอแอดเดรสของรีจิสเตอร์นั้นจะวางอยู่บนพีซีแอดเดรสบัส พีซีจะสโตรป (Storbe) สัญญาณ IOR (I/O read) หรือ (I/O write) เมื่อส่วนวงจรถอดรหัสแอดเดรส (Address Decoding Logic)ที่อยู่บนการ์ดอนุกรมตรวจสอบพบว่าเป็นแอดเดรสของตัวมันเอง มันจะทำการกระตุ้นสัญญาณเลือกชิป (Chip Select) ของ UART(CS2) การอ้างถึงรีจิสเตอร์ภายในของ UART สามารถทำได้โดยการถอดรหัส(Decode Address) จากเส้นแอดเดรส A2-A0 ซึ่งเป็น 3 บิตต่ำของไอโอแอดเดรส

2.8.3 การเชื่อมต่ออินเตอร์รัปต์ (Interrupt Interface) สำหรับแต่ละพอร์ตอนุกรมจะมีสัญญาณอินเตอร์รัปต์โดยต่อสัญญาณ INTR เข้ากับบัสของพีซี โดยกำหนดให้ IRQ4 ใช้ร่วมกันระหว่าง COM1 และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ COM3 ส่วน IRQ3 ใช้ร่วมกันระหว่าง COM2 และ COM4 หมายความว่าเราไม่สามารถที่จะใช้ COM1 COM3 โดยใช้แบบอินเตอร์รัปต์ได้ในเวลาเดียวกัน

จากบล็อคไดอะแกรมโครงสร้างพอร์ตอนุกรม จะเห็นว่าสัญญาณ INTR ไม่ได้ต่อโดยตรงกับสัญญาณ IRQ ที่อยู่บนบัสของพีซี แต่จะมีลอจิกเกต (Logic Gate) คั่นกลาง ซึ่งควบคุมการเปิดปิดของเกตนี้โดยใช้สัญญาณ OUT2 ของ UARTดังนั้นในการที่จะทำให้เกิดการอินเตอร์รัปต์ขึ้นจึงต้องมีการโปรแกรมให้เซ็ตบิต OUT2 ที่อยู่ในรีจิสเตอร์ควบคุมโมเด็ม (Modem Control Register) เป็น 1 เสียก่อน

2.8.4 อินเตอร์เฟส RS - 232 ขา DTR, RTS และ SOUT ของ UART จะต่อกับอินเตอร์เฟสแบบ RS - 232 โดยผ่านตัวขับสัญญาณ (Line Drivers) สำหรับตัวขับสัญญาณนี้จะแปลงสัญญาณ 0 โวลต์ และ 5 โวลต์ของ UART ไปเป็น -12 โวลต์ และ +12 โวลต์ ตามมาตรฐานของ RS - 232 เช่นเดียวกันกับขาสัญญาณอินพุท CD, DSR, CTS, RI และ RD ของ RS - 232 ก็จะต่อกับขาของ UART ที่สอดคล้องกัน ในการออกแบบตัวขับสัญญาณของ RS - 232 นั้นจะใช้ตัวขับสัญญาณที่มีลอจิกตรงข้าม คือมันจะทำการกลับค่าลอจิกอินพุทที่มีมาจาก UART

2.8.5 รายละเอียดของรีจิสเตอร์ของ UART ในรายละเอียดของรีจีสเตอร์จะเป็นการกล่าวถึงหน้าที่ของแต่ละบิตที่อยู่ภายในรีจีสเตอร์ ซึ่งจะกล่าวถึงโดยอ้างจากฐานแอดเดรสของพอร์ต COM1 (03F8h - 03FFh) ส่วนพอร์ตอนุกรม มาตรฐานของสายสัญญาณรับ - ส่งข้อมูลแบบอนุกรม

2.9 มาตรฐานของสายสัญญาณรับ - ส่งข้อมูลแบบอนุกรม

2.9.1 มาตรฐาน RS - 232

มาตรฐาน RS - 232 ได้จัดพิมพ์ขึ้นเมื่อปี ค.ศ. 1969 เป็นมาตรฐานที่กำหนดขึ้นโดยสมาคมของโรงงานอุตสาหกรรมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์แห่งอเมริกา (EIA : The Electronics Industries Association) ซึ่งใช้กันแพร่หลายในระบบการสื่อสารข้อมูลคอมพิวเตอร์ RS ย่อมาจาก Recommended Standard ส่วน 232 เป็นหมายเลขบ่งบอกมาตรฐานของตัวนี้ C เป็นหมายเลขของฉบับท้ายสุดของมาตรฐาน โดยจะกล่าวถึงมาตรฐานของลักษณะของสัญญาณไฟฟ้าในการอินเตอร์เฟสเทอร์มินอลเข้ากับไมโครคอมพิวเตอร์ หรืออินเตอร์เฟสเครื่องพิมพ์เข้ากับคอมพิวเตอร์เป็นต้น โดยจะทำการส่งข้อมูลแบบอนุกรม จุดประสงค์ของมาตรฐานนี้ก็เพื่อบรรยายคุณลักษณะของการเชื่อมต่ออุปกรณ์รับส่งข้อมูลปลายทาง (Data Terminal Equipment : DTE) กับอุปกรณ์สื่อสารข้อมูล (Data Communication Equipment : DCE) สำหรับผู้ใช้ไมโครคอมพิวเตอร์ DTE หมายถึงตัวไมโครคอมพิวเตอร์ และ DCE หมายถึงโมเด็ม อุปกรณ์อื่นๆ เช่นเครื่องพิมพ์ที่รับสัญญาณแบบอนุกรมอาจจะเป็นได้ทั้ง DTE และ DCE ขึ้นอยู่กับผู้ผลิต ซึ่งความเร็วและระยะทางของการเชื่อมต่อของ RS - 232 สามารถเชื่อมต่อการถ่ายโอนข้อมูลได้จาก 0 - 20,000 บิตต่อวินาที ซึ่งเพียงพอสำหรับไมโครคอมพิวเตอร์ที่มีอัตราบอด 110 ถึง 9600 บอด ความยาวของสายเชื่อมต่อสัญญาณตามมาตรฐานของ RS - 232 จำกัดอยู่ประมาณ 50 ฟุต

2.9.1.1 ลักษณะของสัญญาณ RS - 232

การที่จะให้ข้อมูลส่งไปอย่างถูกต้องและอุปกรณ์ควบคุมนั้นได้ถูกใช้งานอย่างเหมาะสม เราจำเป็นต้องทำให้สัญญาณที่จะใช้เข้ากันได้เสียก่อน มาตรฐาน RS - 232 จะให้แรงดันซึ่งนับเป็นจำนวนโวลต์แก่ข้อมูลและควบคุมสัญญาณเพื่อที่จะตอบสนองความต้องการนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS - 232 มีหน้าที่หลักคือ เปลี่ยนแปลงสัญญาณไฟฟ้า ซึ่งได้แก่ สัญญาณดิจิตอลที่ได้จาก Universal Asynchronous Receiver Transmitter (UART) ไปเป็นระดับสัญญาณมาตรฐาน (EIA) ลักษณะของสัญญาณทางไฟฟ้า (Electrical Characteristic) ที่ใช้กับ RS - 232 จะเป็นดังนี้

2.9.1.1.1 ระดับโวลต์เตจของสัญญาณที่ใช้เป็นทั้งค่าบวก และค่าลบ โดยอยู่ระหว่าง 3 โวลต์ และ 15 โวลต์

2.9.1.1.2 ระดับโวลต์เตจระหว่าง +3 โวลต์ และ -3 โวลต์ ไม่มีความหมาย

2.9.1.1.3 ค่าสัญญาณเป็นข้อมูลทางดิจิตอล สถานะหนึ่งเป็น marking state จะกำหนดระดับโวลต์เตจที่เป็นลบ และสถานะ 0 จะกำหนดด้วยโวลต์เตจที่เป็นบวก

2.9.1.1.4 ถ้าสัญญาณนั้นเป็นสัญญาณควบคุมระดับโวลต์เตจที่เป็นบวกจะแสดงสภาวะเปิดของสัญญาณ และระดับโวลต์เตจที่เป็นลบจะแสดงสภาวะปิดของสัญญาณ

2.9.2 มาตรฐาน RS - 485 และ RS - 422

มาตรฐาน RS - 485 นั้นมีความใกล้เคียงกับมาตรฐาน RS - 422 แต่ต่างกันที่เอาท์พุทไดร์ฟเวอร์ (Out Driver) ได้ถูกออกแบบมาเพื่อเอาท์พุตแบบไตรสเตท (Tristate output) โดยลักษณะร่วมกันของมาตรฐาน RS - 232 และ RS - 485 มีดังนี้

มาตรฐาน RS - 485 และ RS - 422 จะใช้เส้นส่งสัญญาณแบบสัญญาณผลต่าง (Differential signaling) ซึ่งจะใช้คู่หนึ่งสำหรับส่งข้อมูล และใช้เส้นส่งสัญญาณอีกคู่หนึ่งรับสัญญาณจากภายนอกอีกด้วย (แต่ในโครงงานนี้จะใช้เพียงแค่อย่างละเส้น)

มาตรฐาน RS - 422 ได้กำหนดไว้ว่าถ้าความต่างศักย์ระหว่างเส้นส่งสัญญาณคู่ใด ๆ มีค่าเป็นบวก แสดงว่าเส้นส่งสัญญาณคู่นั้นมีค่าทางตรรกเป็นหนึ่ง ซึ่งหมายความว่าสาย A จะต้องมีศักดาไฟฟ้าเป็นทั้งบวกและลบ ในมาตรฐาน RS - 422 นี้จะใช้สัญญาณอยู่ในช่วง 0 ถึง 5 โวลต์ ซึ่งหมายความว่า RS - 422 จะสามารถใช้ศักดาไฟฟ้าที่ใช้ในระบบไมโครโปรเซสเซอร์และวงจรอื่น ๆ ได้

สัญญาณข้อมูลจะถูกส่งไปในรูปของผลต่างของระดับศักดาไฟฟ้าในสายส่งทั้งสองเส้นที่พันเกลียวกันไปก็ไม่มีประโยชน์อันใด ยกเว้นแต่ว่าค่าศักดาไฟฟ้าที่เทียบกับกราวด์ในสายส่งเหล่านั้นจะมีค่ามากเกินกว่าที่วงจรรับสัญญาณจะทนได้ นั่นหมายความว่า RS - 422 นี้สามารถทำงานได้ถึงแม้ว่าจะมีการรบกวนแบบโหมดร่วมคือ การเปลี่ยนแปลงค่าระดับศักดาไฟฟ้าในเส้นส่งสัญญาณทั้งสองเส้นจำนวนพอ ๆ กัน แต่จะทำให้ค่าผลต่างของระดับศักดาไฟฟ้าในเส้นส่งสัญญาณทั้งสองนี้เปลี่ยนแปลง เพราะฉะนั้นข้อมูล (ที่จะส่งไปในรูปของผลต่างของระดับศักดาไฟฟ้าในสายส่งสัญญาณทั้งสองนี้) จึงไม่มีการเปลี่ยนแปลง

มาตรฐาน RS - 422 ต่างจากมาตรฐาน RS - 232 ตรงที่มาตรฐาน RS - 422 จะใช้สายรับสัญญาณ 1 คู่ และสายส่งสัญญาณ 1 คู่เท่านั้น ไม่มีการระบุการใช้เส้นส่งสัญญาณควบคุมใด ๆ และไม่ได้มีการกำหนดชนิดของตัวเชื่อมต่อที่ต้องใช้

เนื่องจากไม่มีการใช้สายส่งสัญญาณตอบรับ เราจึงต้องส่งข้อมูลควบคุมไปกับข้อมูลจริง เพื่อให้การทำงานเป็นไปดังโปรโตคอล การไม่มีสายส่งสัญญาณตอบรับทำให้การส่งข้อมูลในระบบ RS - 422 มีโอเวอร์เฮดสูงกว่าส่งข้อมูลในระบบ RS - 232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

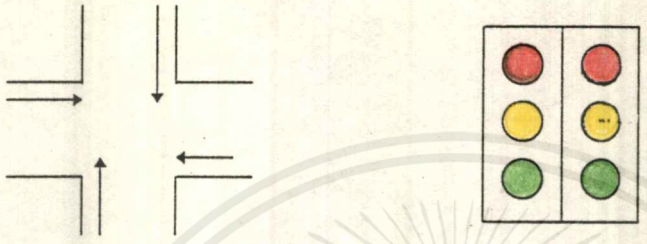
การส่งข้อมูลในระบบ RS - 422 ที่ใช้สัญญาณผลต่างทำให้เราสามารถส่งข้อมูลได้ด้วยอัตราเร็วกว่า และระยะทางไกลกว่าในระบบ RS -232 ระบบ RS - 422 นี้สามารถส่งข้อมูลด้วยอัตรา 38.4 กิโลบอด ในระยะทางประมาณ 4000 ฟุต แต่ในระบบ RS -232 จะสามารถส่งข้อมูลด้วยอัตรา 38.4 กิโลบอด ภายในระยะทางเพียง 12.5 ฟุต โดยระยะทางสูงสุดของสายขึ้นอยู่กับอัตราข้อมูล จำนวนสัญญาณรบกวนและความเพี้ยนที่สามารถทนได้ และความไม่สมดุลของสายเคเบิล ถ้าเพิ่มความยาวของเคเบิลจะเกิดความเพี้ยนมากขึ้น ดังนั้นสายเคเบิลจึงควรให้สั้นที่สุดเพื่อขจัดปัญหาที่อาจเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการสร้าง

3.1 การออกแบบระบบ

โครงงานนี้เป็นการออกแบบระบบ และสร้างระบบควบคุมสัญญาณไฟจราจรของสี่แยกสี่ช่องทางไปกลับดังรูป 3.1 ก และสัญญาณไฟจราจรที่ใช้จะแบ่งออกเป็น 2 ชุดสำหรับให้สัญญาณไฟจราจรแก่รถที่ต้องการตรงไป และสำหรับให้สัญญาณไฟจราจรแก่รถที่ต้องการเลี้ยวขวาดังรูป 3.1 ข

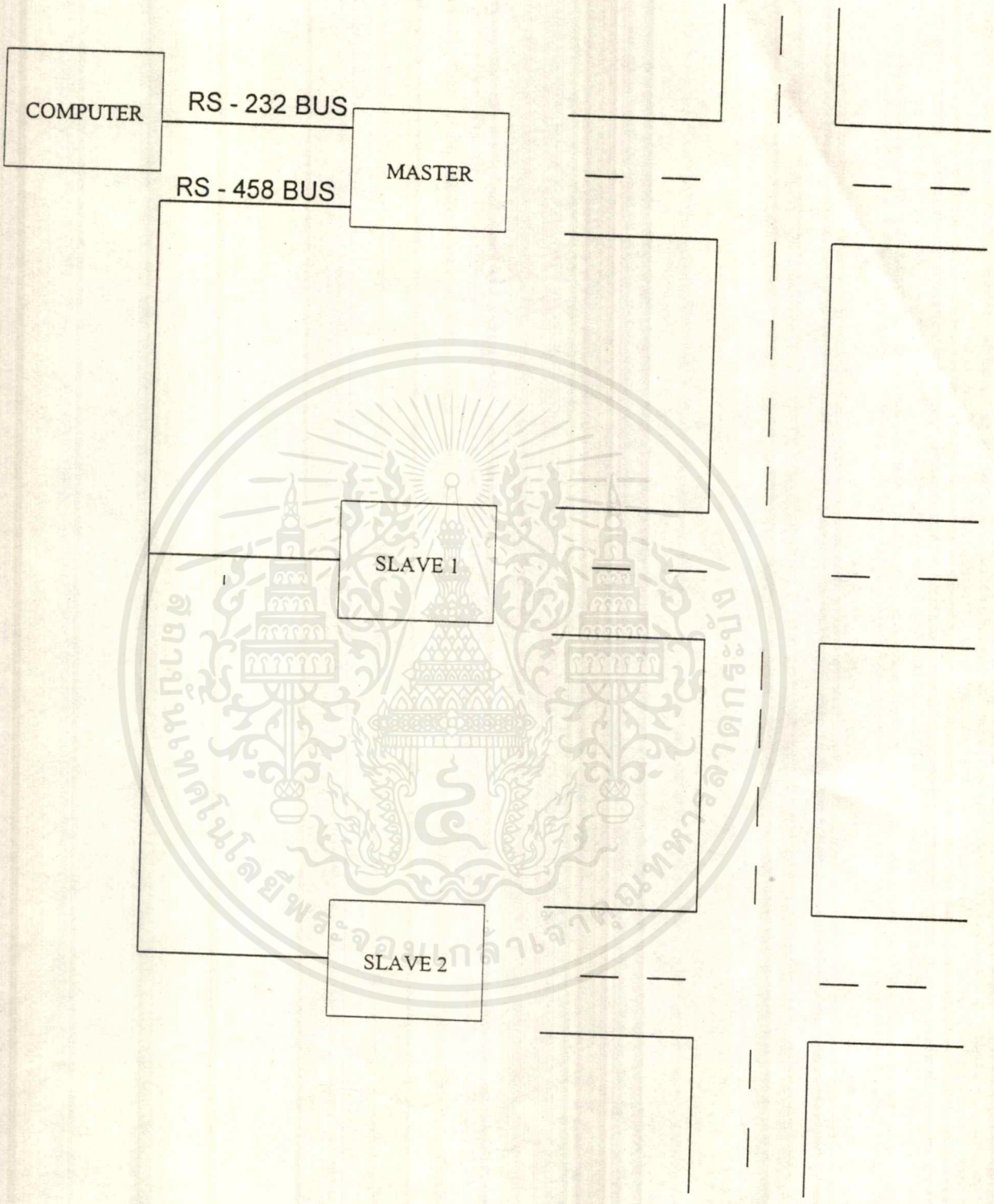


ก. ข.

รูป 3.1 ก. ลักษณะของสี่แยกที่ใช้ในการออกแบบ
ข. สัญญาณไฟจราจรที่ใช้ในการออกแบบ

การออกแบบได้ออกแบบให้ระบบควบคุมสัญญาณไฟจราจรนี้สามารถ ทำการควบคุมสัญญาณไฟจราจรได้ในแต่ละสี่แยก และยังสามารถทำการเชื่อมต่อกันได้ทั้งหมด 3 สี่แยกเพื่อให้การควบคุมสัญญาณไฟจราจรเป็นไปอย่างต่อเนื่องและสอดคล้องกัน โดยการควบคุมจากตัวควบคุมที่มีอยู่ในแต่ละสี่แยก และรับสัญญาณควบคุมจากตัวควบคุมหลักซึ่งจะทำการส่งสัญญาณควบคุมไปยังสี่แยกอีก 2 สี่แยก เพื่อให้การควบคุมสัญญาณไฟจราจรเป็นไปอย่างต่อเนื่องและสอดคล้องกันตามรูป 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

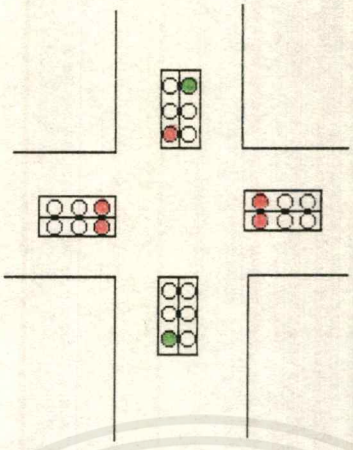


รูป 3.2 ระบบควบคุมสัญญาณไฟจราจร

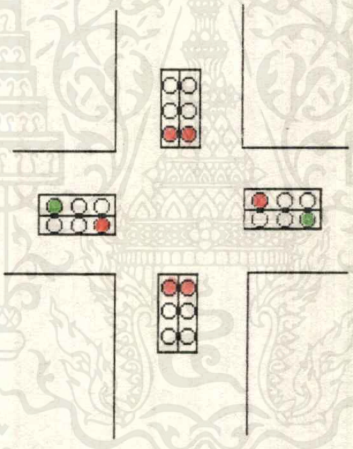
และในการออกแบบได้ทำการกำหนดรูปแบบในการเปิดสัญญาณไฟจราจรเป็นรูปแบบต่าง ๆ ทั้งหมด 8 รูปแบบตามรูป 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

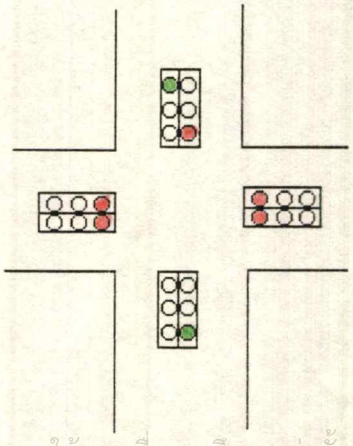
การเปิดสัญญาณไฟจราจรรูปแบบที่ 1



การเปิดสัญญาณไฟจราจรรูปแบบที่ 2

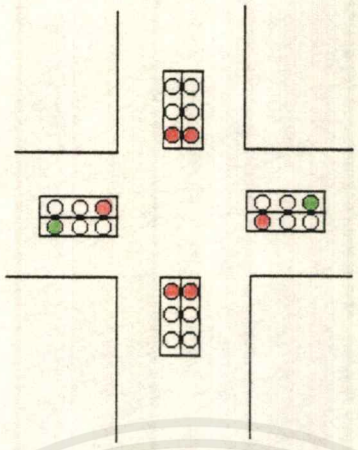


การเปิดสัญญาณไฟจราจรรูปแบบที่ 3

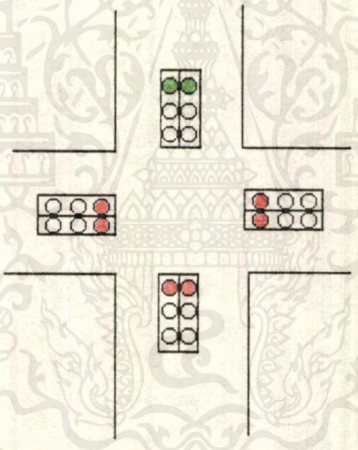


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

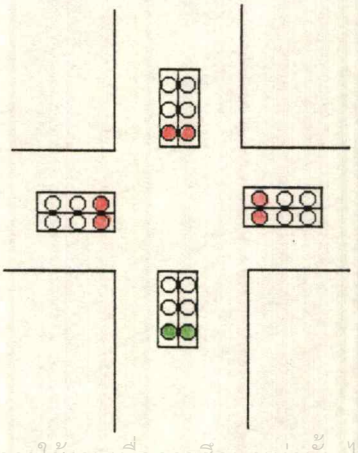
การเปิดสัญญาณไฟจราจรรูปแบบที่ 4



การเปิดสัญญาณไฟจราจรรูปแบบที่ 5

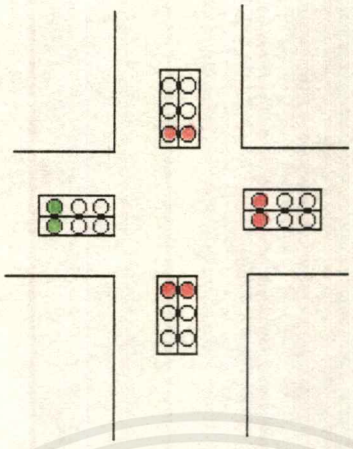


การเปิดสัญญาณไฟจราจรรูปแบบที่ 6

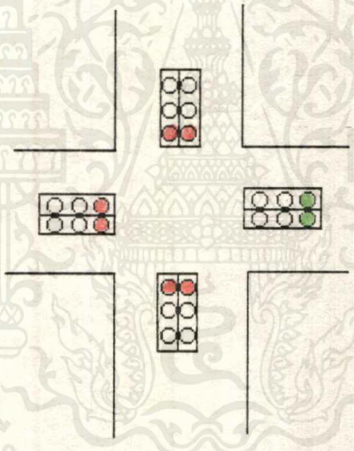


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปิดสัญญาณไฟจราจรรูปแบบที่ 7



การเปิดสัญญาณไฟจราจรรูปแบบที่ 8



รูป 3.3 รูปแบบการเปิดสัญญาณไฟจราจรทั้ง 8 รูปแบบ

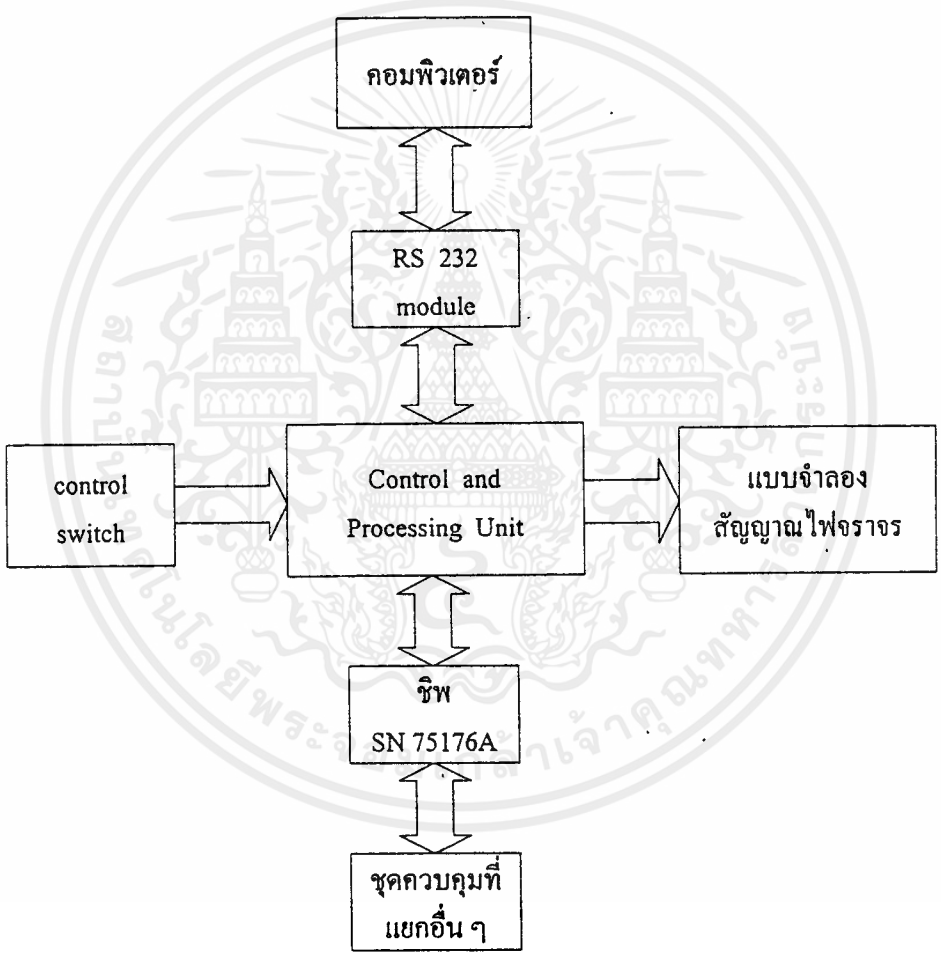
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
039096

3.2 ส่วนประกอบหลักของชุดควบคุมสัญญาณไฟจราจร

ชุดควบคุมสัญญาณไฟจราจรที่ออกแบบขึ้นนี้ประกอบด้วยส่วนประกอบหลัก 5 ส่วน ดังนี้

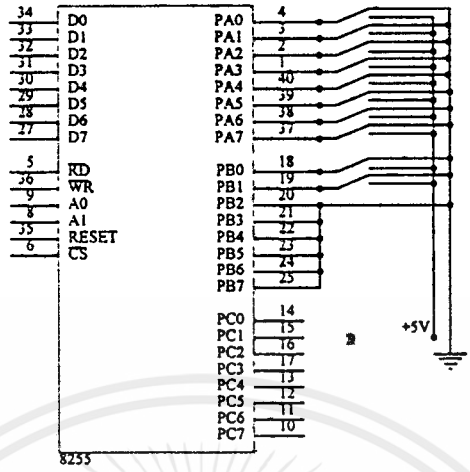
1. ส่วนของสวิตช์ควบคุม
2. ส่วนควบคุมและประมวลผล
3. ส่วนแสดงผล
4. ส่วนเชื่อมต่อระหว่างชุดควบคุม
5. ส่วนเชื่อมต่อระหว่างตัวควบคุมหลักและคอมพิวเตอร์

ซึ่งสามารถเขียนแสดงด้วยบล็อกไดอะแกรมได้ดังรูป 3.4



รูป 3.4 บล็อกไดอะแกรมของชุดควบคุมสัญญาณไฟจราจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 การเชื่อมต่อส่วนของสวิตช์

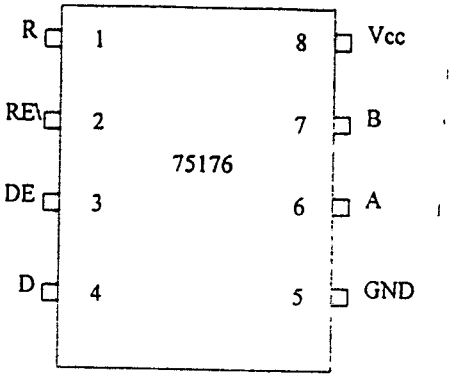
3.2.1 ส่วนของสวิตช์ควบคุม เป็นส่วนที่ป้อนสัญญาณอินพุทให้แก่ส่วนควบคุมและประมวลผล ประกอบด้วยสวิตช์ทั้งหมด 10 ตัว ซึ่งมีการเชื่อมต่อกับส่วนควบคุมและประมวลผลตามรูป 3.5 โดยจะต่อสวิตช์กับพอร์ท A ของ 8255 จำนวน 8 บิต และพอร์ท B ของ 8255 อีกจำนวน 2 บิต เนื่องจากบิทที่เหลือจะถูกใช้กับงานอื่น ดังนั้นเมื่อต้องการอินพุทสวิตช์ 10 ตัวจึงต้องทำการต่อสวิตช์ดังรูป โดยสวิตช์ตัวที่ 1 ถึง 8 เป็นสวิตช์สำหรับเลือกรูปแบบการเปิดสัญญาณไฟจราจรตั้งแต่รูปแบบที่ 1 ถึง 8 และสวิตช์ตัวที่ 9 เป็นสวิตช์เลือกการควบคุมแบบ auto และสวิตช์ตัวที่ 10 เป็นสวิตช์เลือกการทำงานให้ไฟเหลืองกระพริบ

3.2.2 ส่วนควบคุมและประมวลผล ประกอบด้วยส่วนต่าง ๆ ในวงจรตามรูปที่ 3.8 และชิพเบอร์ DS 1202 เป็น real time clock ซึ่งใช้เป็นเวลาอ้างอิงสำหรับส่วนควบคุม ในการควบคุมการเปิดสัญญาณไฟจราจร ซึ่งชิพตัวนี้จะต่ออยู่บนบอร์ดควบคุม ส่วนควบคุมจะเป็นตัวที่ทำการประมวลผลจากสัญญาณอินพุทที่ได้รับเข้ามาจากสวิตช์ควบคุม ทำให้การทำงานเป็นไปตามโปรแกรมที่เก็บอยู่ภายในหน่วยความจำบนบอร์ดควบคุมได้ สัญญาณเอาท์พุทส่งออกทางพอร์ท 8255 ให้มีการเปิดปิดสัญญาณไฟจราจรตามรูปแบบที่ผู้ควบคุมเลือก

3.2.3 ส่วนแสดงผล ซึ่งจะแสดงผลออกมาทางหน้าจอมอนิเตอร์ โดยตัวควบคุมหลักจะทำการเชื่อมต่อกับคอมพิวเตอร์ทางพอร์ตอนุกรม เพื่อจะนำค่าของตัวลูกมาแสดงผลทางหน้าจอว่าในขณะนี้นั้นตัวลูกที่แยกไหนมีรูปแบบการเดินรถแบบใด นอกจากนี้แล้วที่แต่ละแยกนั้นจะใช้แบบจำลองที่ประกอบด้วย LED ซึ่งต่อกับ 8255 ของบอร์ดควบคุมเพื่อแสดงรูปแบบการเดินรถของแยกนั้น ๆ

3.2.4 ส่วนเชื่อมต่อกันระหว่างชุดควบคุมหลักกับชุดควบคุมที่สี่แยกอื่นอีก 7 สี่แยก ในปริญญานิพนธ์นี้จะใช้ชิพ SN 75176 A มาควบคุมโดยชิพ SN 75176 A จะมีลักษณะดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงการจัดวางขาของชิพ SN 75176

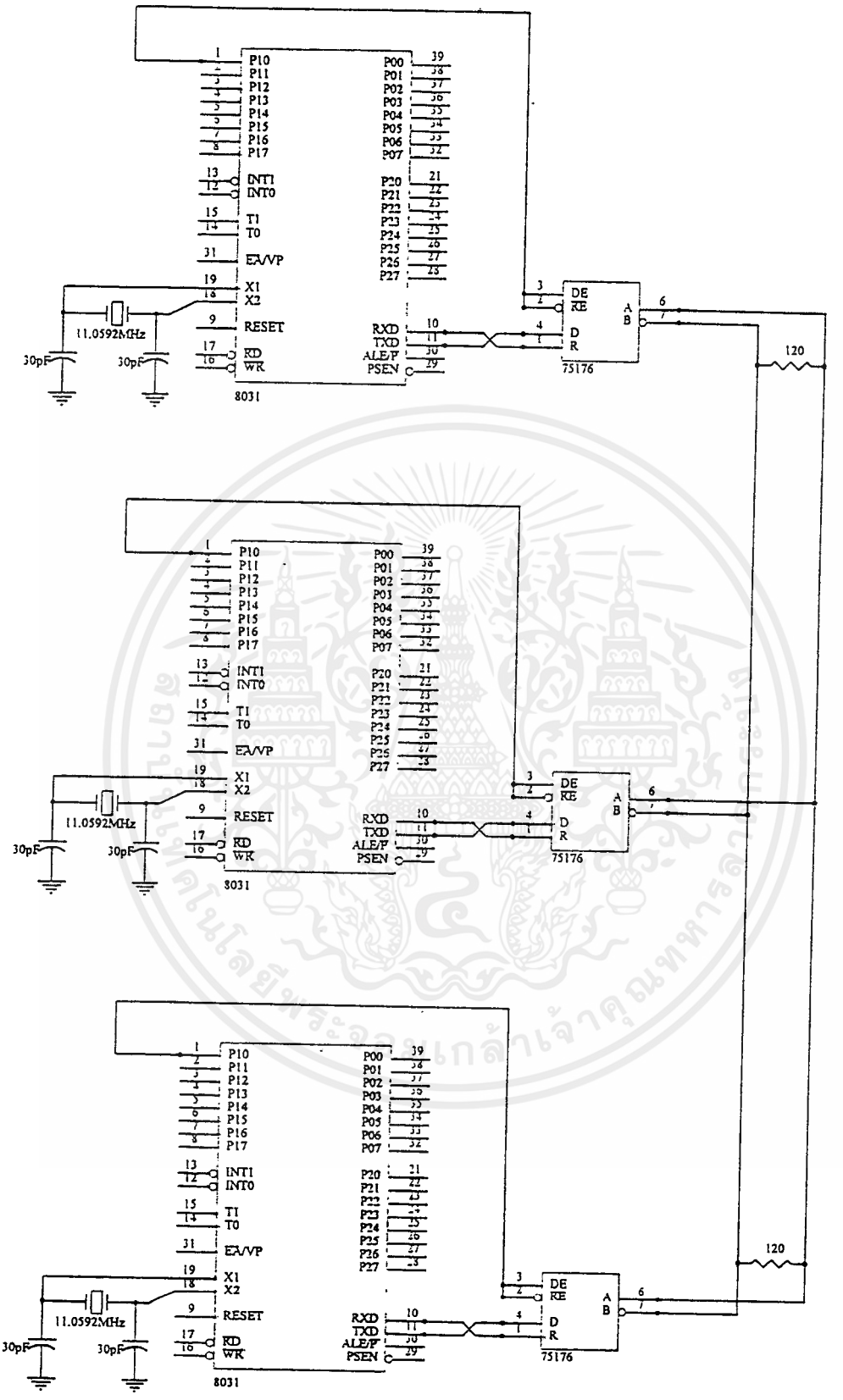
ในการใช้งานเราจะต่อขา DE (ขา 3) และ RE\ (ขา 2) ไว้ด้วยกัน แล้วควบคุมโดยไมโครโปรเซสเซอร์ โดยใช้ขา P1.0 ในการควบคุม

ทางด้านส่ง ขา Tx จากไมโครโปรเซสเซอร์จะต่อเข้ากับขา D ของ 75176 ซึ่งจะแปลงอินพุทนี้ให้เป็นแบบ RS - 485 ออกที่ขา A, B เมื่อขา DE แอคทีฟ และเอาท์พุทจะมีลักษณะเป็นไฮอิมพีแดนซ์ เมื่อขา DE ไม่แอคทีฟซึ่งจะทำหน้าที่เปรียบเสมือนตัวควบคุมทิศทาง

ทางด้านรับ เมื่อขา RE\ แอคทีฟสัญญาณจากขา A, B จะถูกแปลงกลับจากขา A, B จะแปลงกลับเป็น TTL ออกทางขา R ของ 75176 และเอาท์พุทที่ขา R จะเป็นไฮอิมพีแดนซ์เมื่อขา RE\ ไม่แอคทีฟ โดยการต่อทั้งหมดจะมีลักษณะดังรูปที่ 3.7

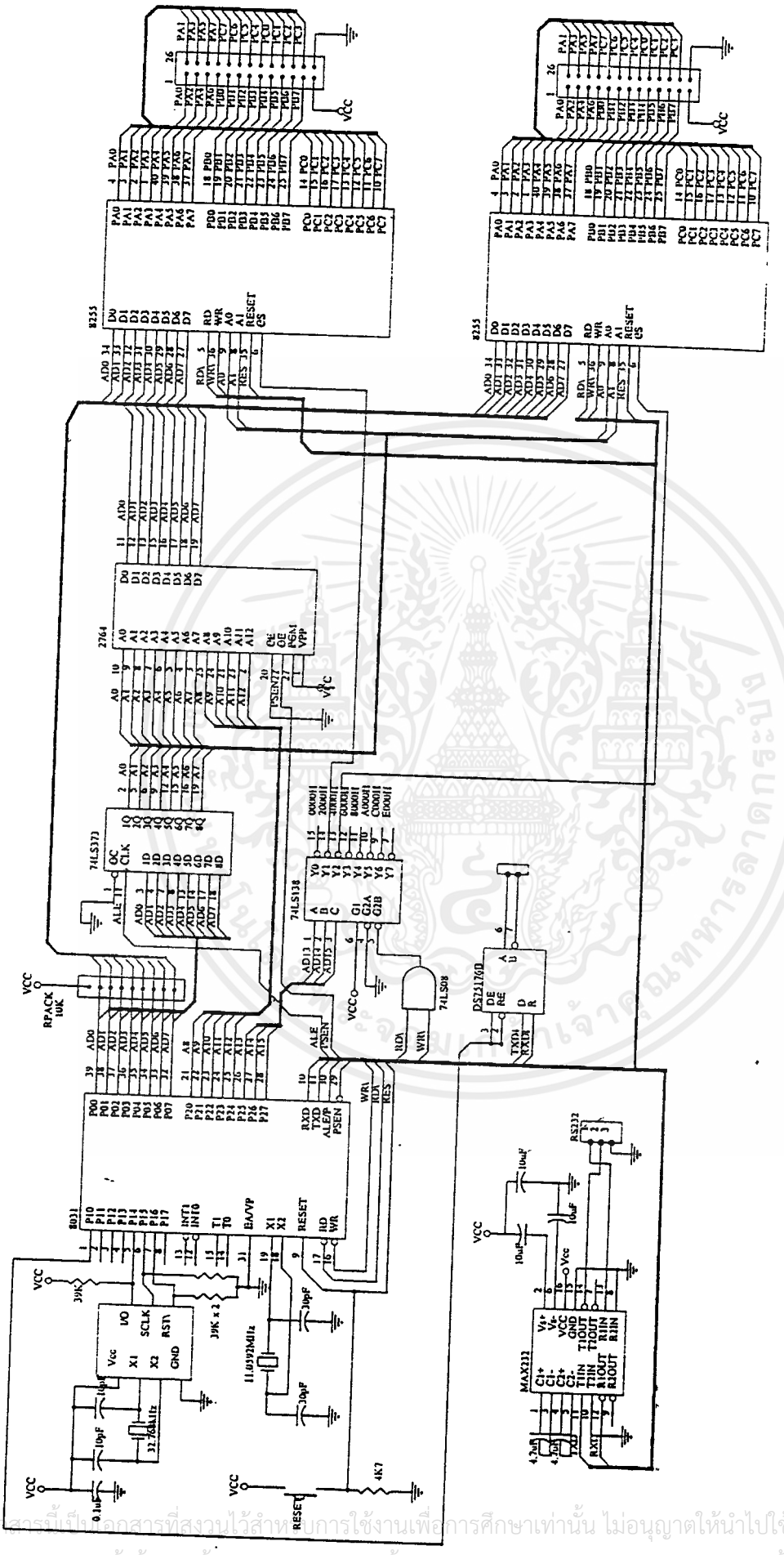
3.2.5 ส่วนเชื่อมต่อระหว่างตัวควบคุมหลักและคอมพิวเตอร์ ประกอบด้วย RS232 module ทำหน้าที่ส่งสัญญาณจากชุดควบคุมหลัก และรับสัญญาณควบคุมจากชุดควบคุมหลัก เพื่อให้การควบคุมสัญญาณไฟจราจรของทั้ง 3 สี่แยกเป็นไปอย่างต่อเนื่องและสอดคล้องกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงการเชื่อมต่อกันระหว่างชุดควบคุมหลักและชุดควบคุมที่แยกอื่น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงวงจรที่ใช้ในการควบคุมสัญญาณไฟจราจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่4

การทดลองและผลการทดลอง

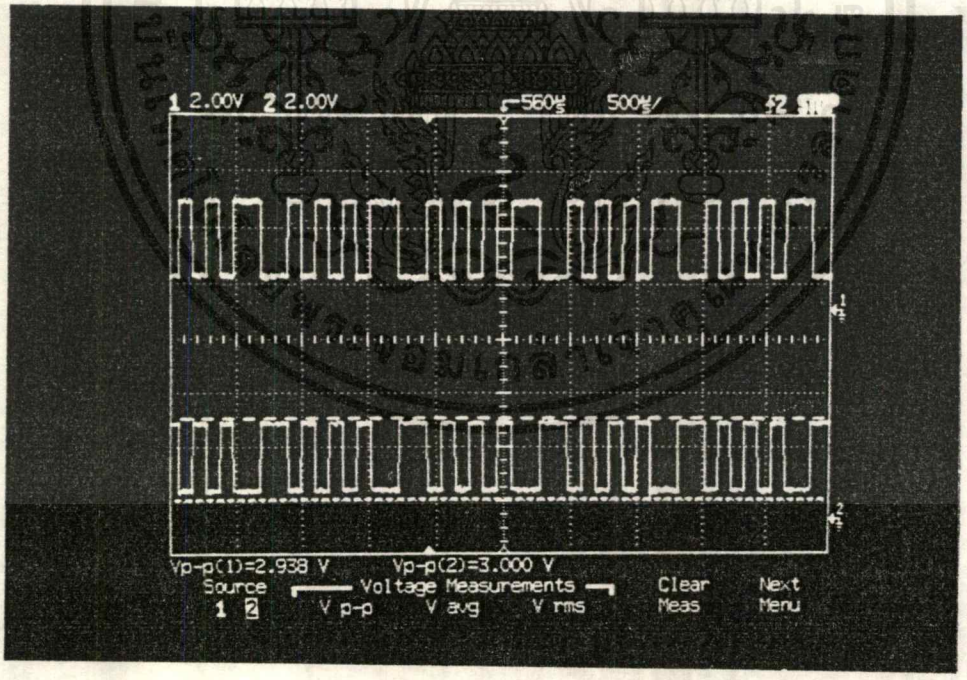
4.1 การทดลอง

ทำการทดลองโปรแกรมที่ใช้ควบคุมสัญญาณไฟจราจร โดยขั้นตอนการทดลองมีดังนี้

1. ต่อสวิทช์เข้าพอร์ต A และพอร์ต B ของ 8255 บนบอร์ดควบคุม
2. ต่อแบบจำลองสัญญาณไฟจราจรเข้ากับเอาท์พุทพอร์ต 8255 ของบอร์ดควบคุม
3. ต่ออีพรอมอีมูเลเตอร์เข้ากับบอร์ดควบคุม
4. เชื่อมต่อตัวควบคุมหลัก (ตัวแม่) เข้ากับชุดควบคุมที่แยกอื่น ๆ โดยใช้ ชิพ SN 75176A
5. เชื่อมต่อตัวควบคุมหลักเข้ากับคอมพิวเตอร์โดยใช้ RS - 232 module
6. จ่ายไฟดีซี +5 โวลต์ ให้บอร์ดควบคุม
7. โหลดโปรแกรมจากคอมพิวเตอร์ผ่านอีพรอมอีมูเลเตอร์
8. สวิทช์เลือกรูปแบบการเปิดสัญญาณไฟจราจรตั้งแต่สวิทช์ที่ 1 ถึง 10

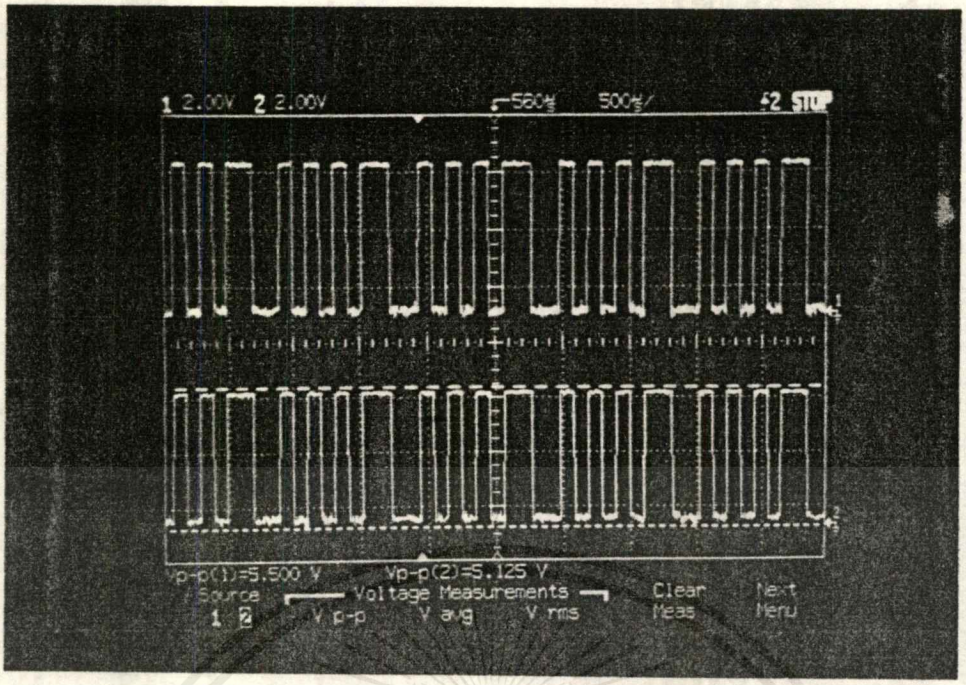
4.2 ผลการทดลอง

เมื่อทำการทดลองตามขั้นตอนดังกล่าว แล้วทำการวัดสัญญาณต่าง ๆ จะได้ผลการทดลองดังนี้

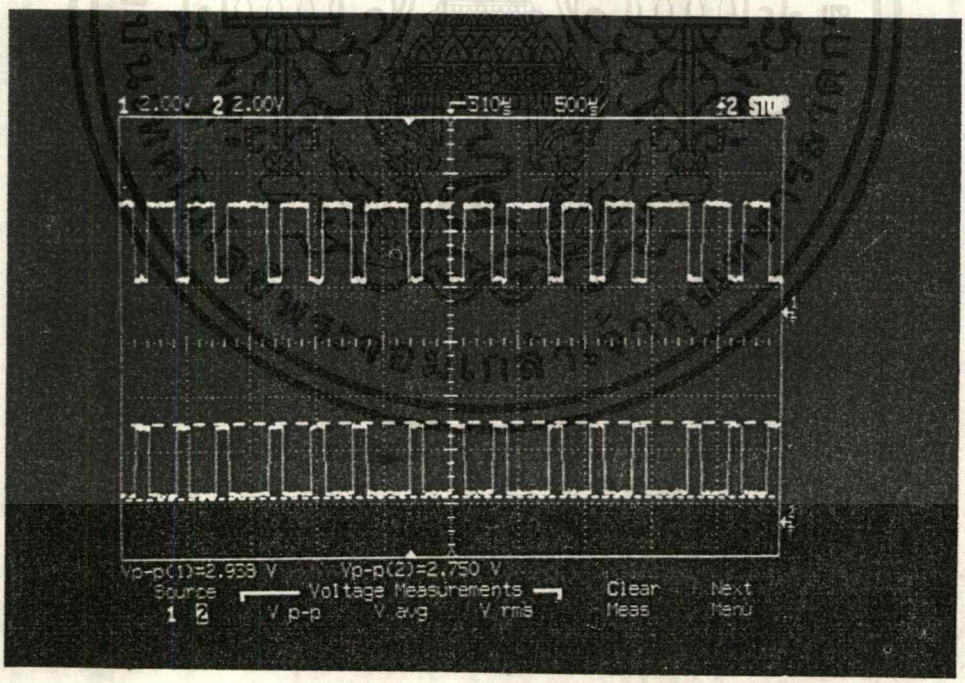


รูปที่ 4.1 แสดงค่าแอดเดรส #AAH ที่ตัวแม่ส่งให้กับลูกตัวที่ 1
เมื่อชันแนล 1 วัดที่ขา A และชันแนล 2 วัดที่ขา B ของชิพ SN 75176A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

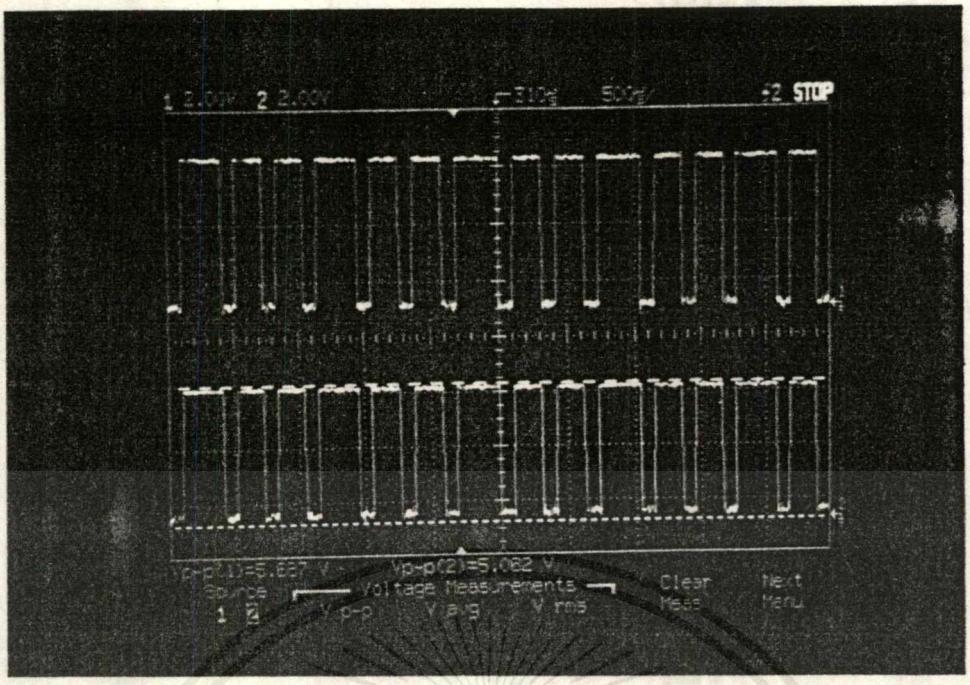


รูปที่ 4.2 แสดงค่าแอดเดรส #AAH ที่ตัวแม่ส่งให้กับลูกตัวที่ 1
เมื่อชันแนล 1 วัดที่ขา Tx ของตัวแม่ และชันแนล 2 วัดที่ขา Rx ของตัวลูก

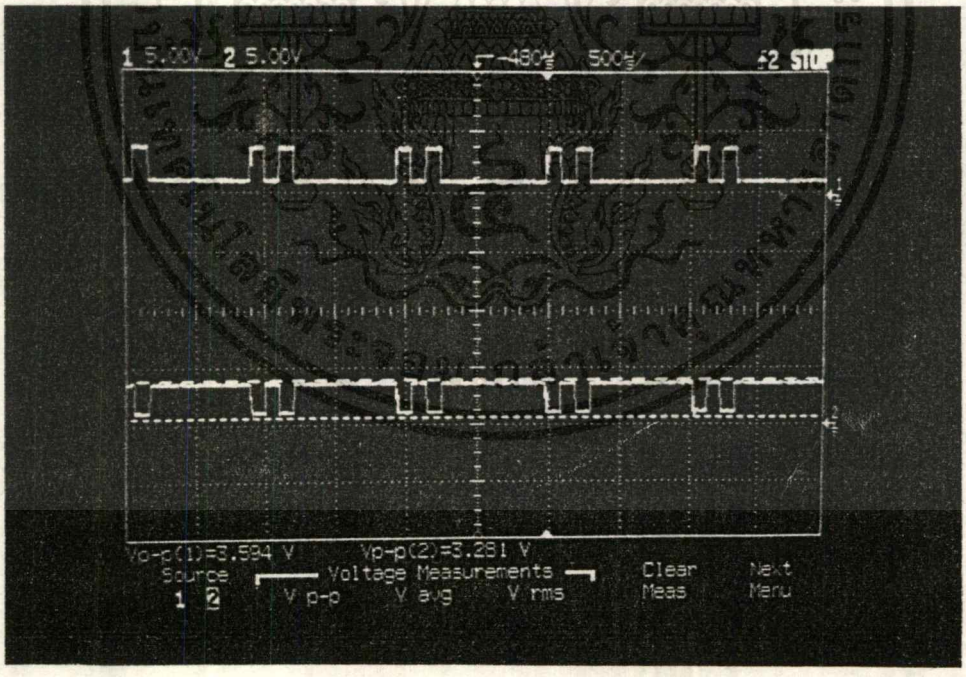


รูปที่ 4.3 แสดงค่าแอดเดรส #BBH ที่ตัวแม่ส่งให้กับลูกตัวที่ 2
เมื่อชันแนล 1 วัดที่ขา A และชันแนล 2 วัดที่ขา B ของชิพ SN 75176A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

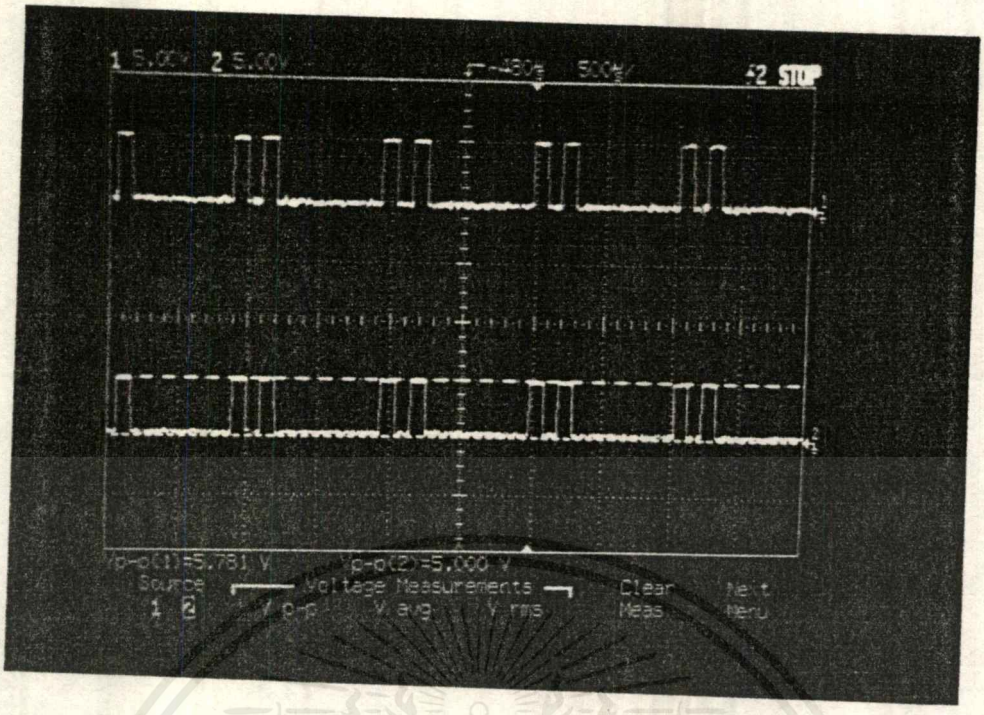


รูปที่ 4.4 แสดงค่าแอดเดรส #BBH ที่ตัวแม่ส่งให้กับลูกตัวที่ 2
เมื่อชันแนล 1 วัดที่ขา Tx ของตัวแม่ และชันแนล 2 วัดที่ขา Rx ของตัวลูก

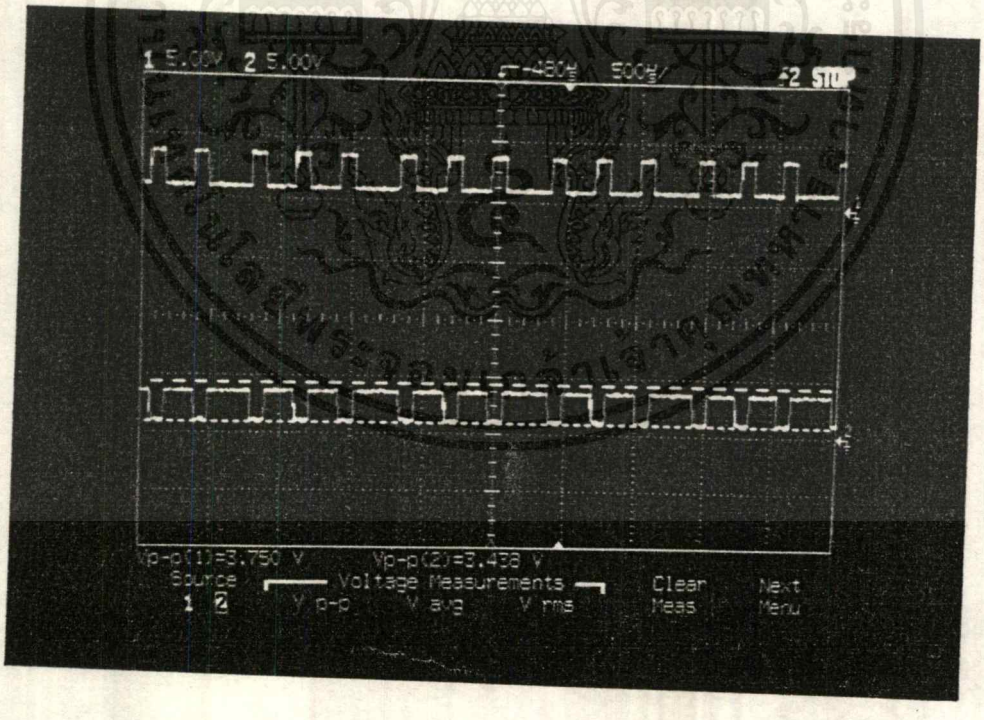


รูปที่ 4.5 แสดงค่าแอดเดรส #01H ที่ตัวลูกส่งให้กับตัวแม่
เมื่อชันแนล 1 วัดที่ขา A และชันแนล 2 วัดที่ขา B ของชิพ SN 75176A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

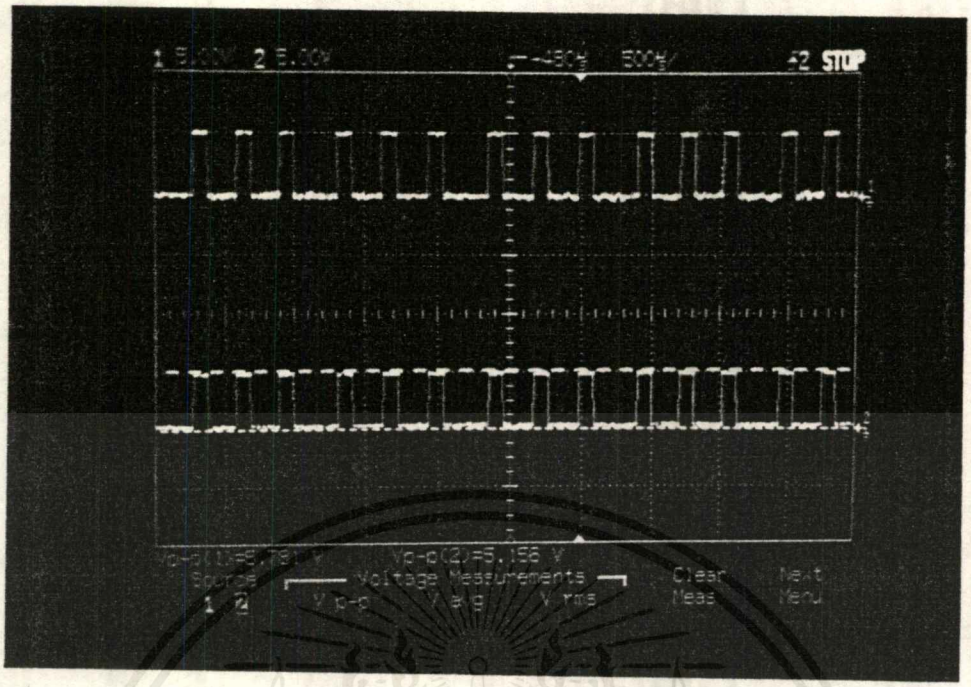


รูปที่ 4.6 แสดงค่าแอดเดรส #01H ที่ตัวลูกส่งให้กับตัวแม่
เมื่อชันแนล 1 วัดที่ขา Tx ของตัวลูก และชันแนล 2 วัดที่ขา Rx ของตัวแม่

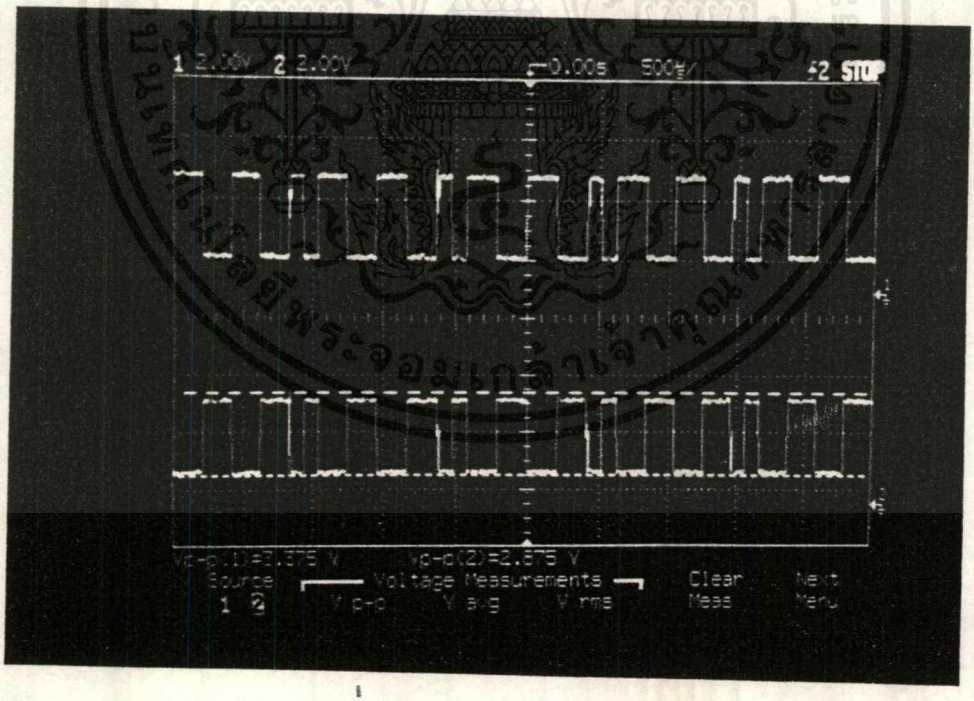


รูปที่ 4.7 แสดงค่าของดาต้า #22H ที่ตัวลูกส่งให้กับตัวแม่
เมื่อชันแนล 1 วัดที่ขา A และชันแนล 2 วัดที่ขา B ของชิพ SN 75176A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

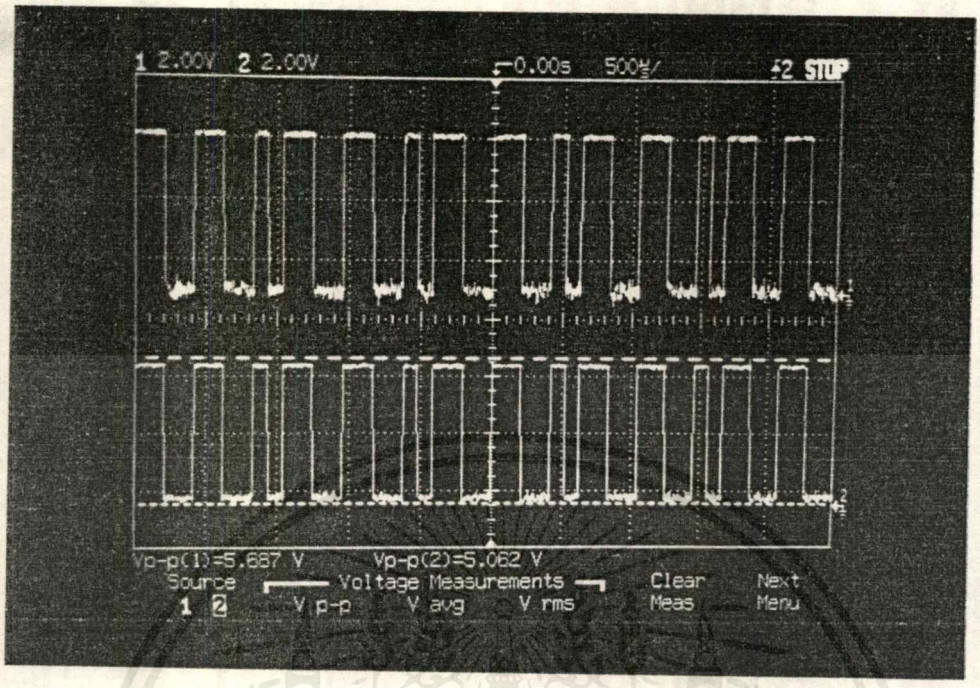


รูปที่ 4.8 แสดงค่าดาต้า #22H ที่ตัวลูกส่งให้กับลูกตัวแม่
เมื่อชันแนล 1 วัดที่ขา Tx ของตัวลูก และชันแนล 2 วัดที่ขา Rx ของตัวแม่

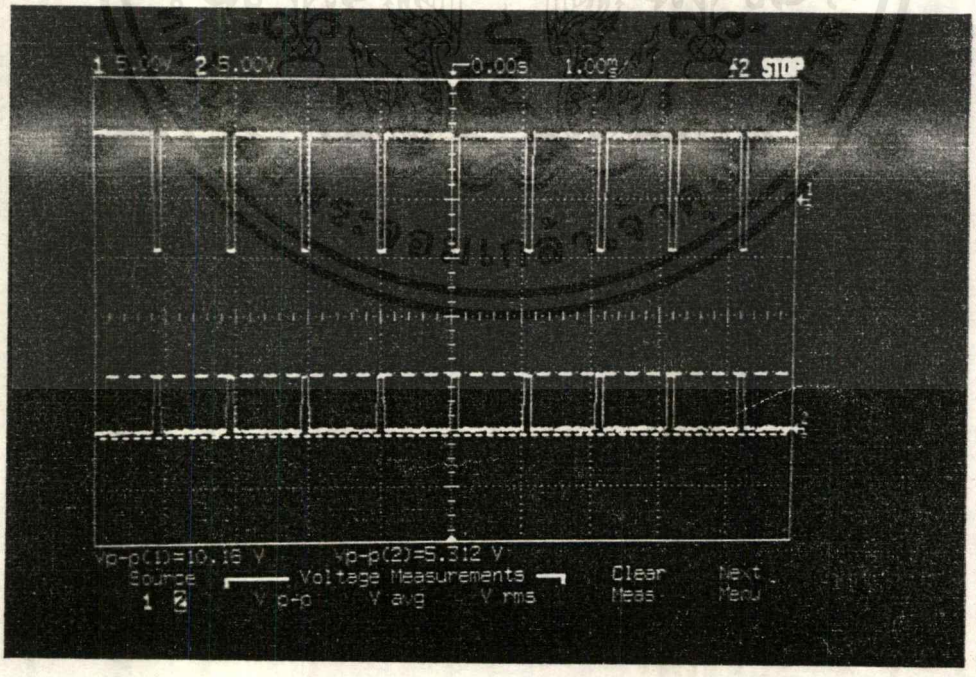


รูปที่ 4.9 แสดงค่าดาต้า #33H ที่ตัวลูกส่งให้กับตัวแม่
เมื่อชันแนล 1 วัดที่ขา A และชันแนล 2 วัดที่ขา B ของชิพ SN 75176A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

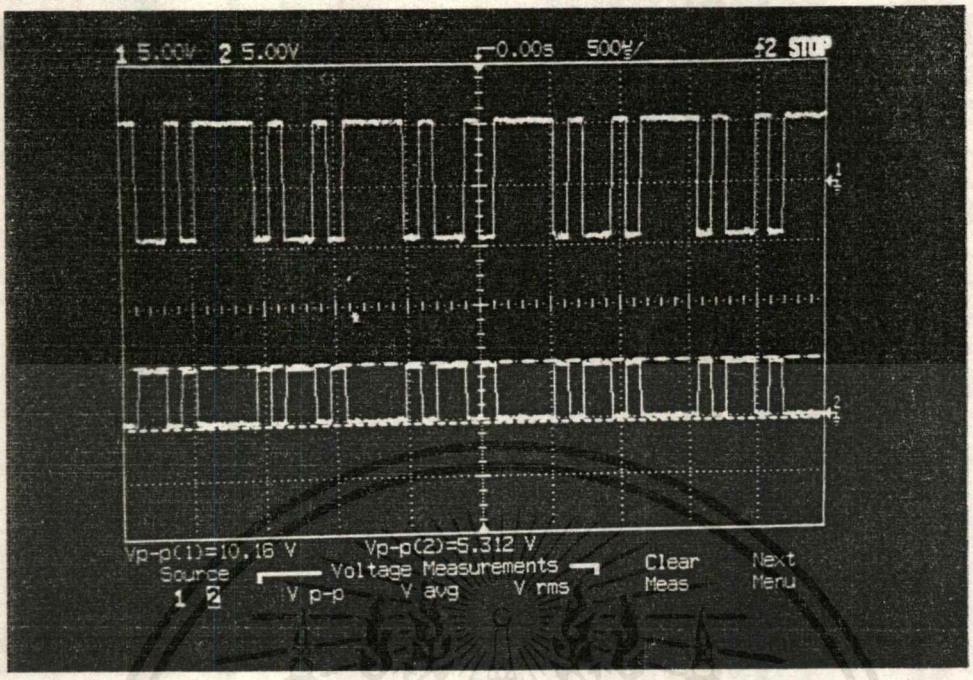


รูปที่ 4.10 แสดงค่าดาต้า #33H ที่ตัวลูกส่งให้กับลูกตัวแม่
เมื่อชันแนล 1 วัดที่ขา Tx ของตัวลูก และชันแนล 2 วัดที่ขา Rx ของตัวแม่

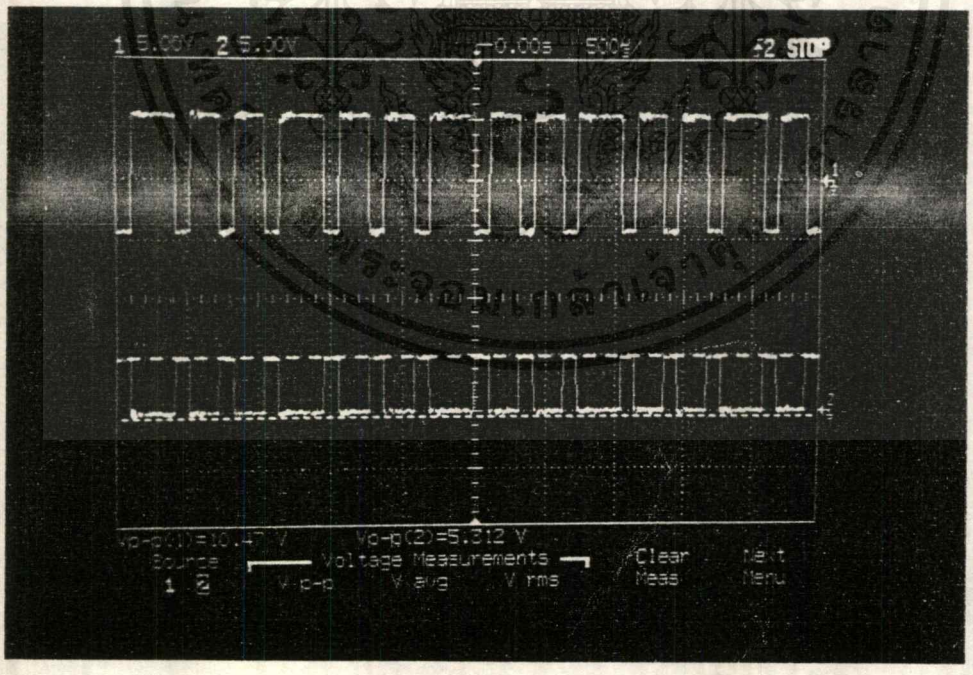


รูปที่ 4.11 แสดงค่าแอดเดรส #00H ที่ตัวแม่ส่งให้กับคอมพิวเตอร์
เมื่อชันแนล 1 วัดที่ขา Tx ของ RS - 232 และชันแนล 2 วัดที่ขา Tx ของ 8031

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

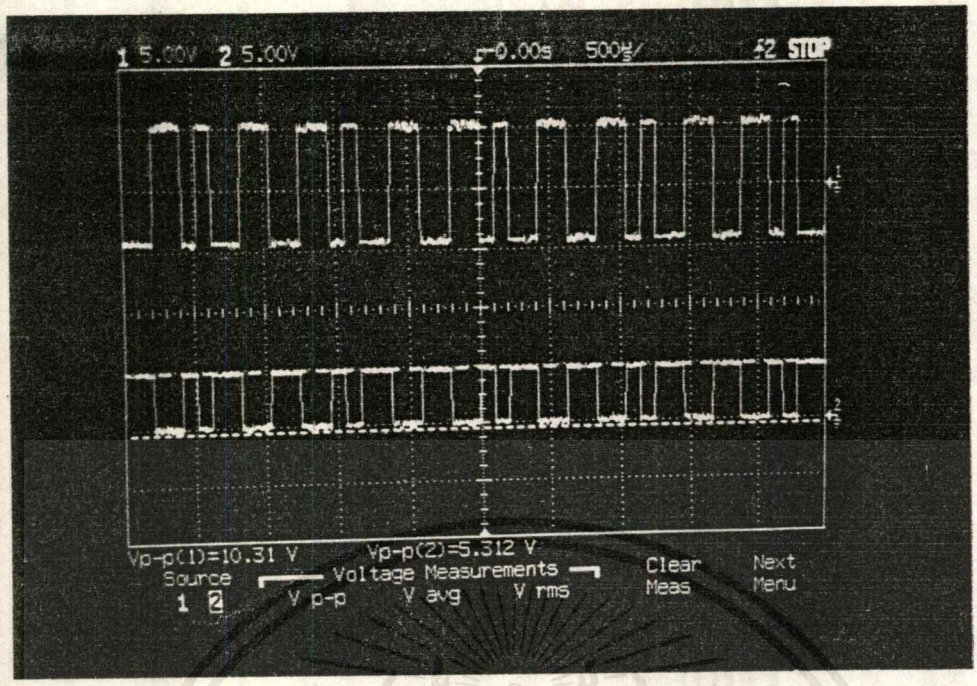


รูปที่ 4.12 แสดงค่าดาต้า #A1H ที่ตัวแม่ส่งให้กับคอมพิวเตอร์
เมื่อชันแนล 1 วัดที่ขา Tx ของ RS – 232 และชันแนล 2 วัดที่ขา Tx ของ 8031

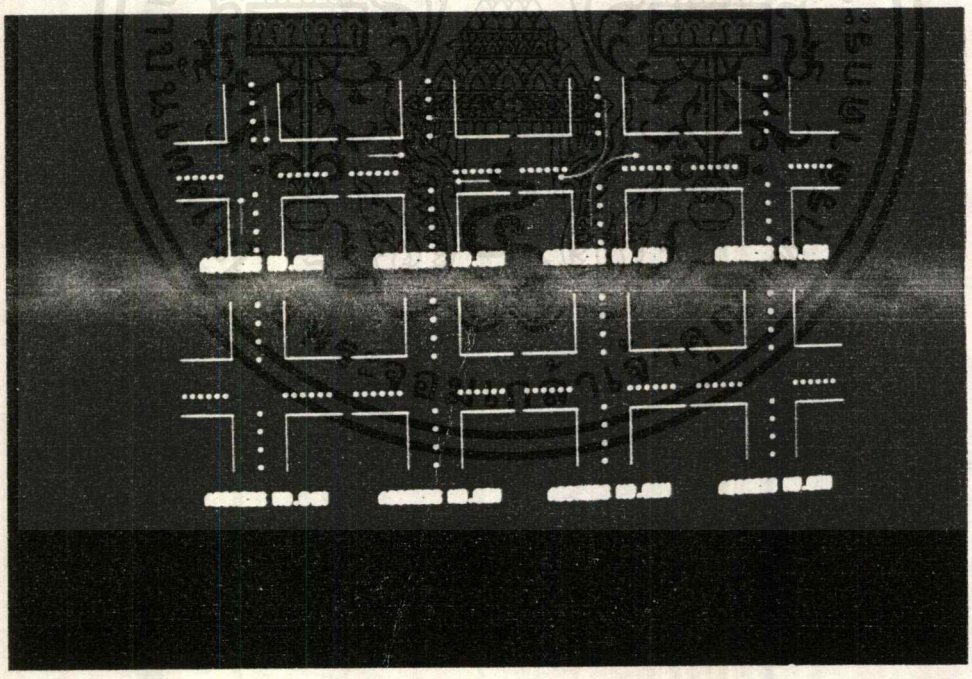


รูปที่ 4.13 แสดงค่าดาต้า #22H ที่ตัวแม่ส่งให้กับคอมพิวเตอร์
เมื่อชันแนล 1 วัดที่ขา Tx ของ RS -232 และชันแนล 2 วัดที่ขา Tx ของ 8031

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงค่าดาต้า #33H ที่ตัวแม่ส่งให้กับคอมพิวเตอร์
เมื่อชันแนล 1 วัดที่ขา Tx ของ RS – 232 และชันแนล 2 วัดที่ขา Tx ของ 8031



รูปที่ 4.15 แสดงผลหน้าจอคอมพิวเตอร์
เมื่อตัวแม่ส่งค่า A1H, 22H, 33H ให้กับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และบทสรุป

สรุปผลการทดลอง

จากผลการทดลองเราสามารถทำการควบคุมสัญญาณไฟจราจรได้ตามรูปแบบที่กำหนดไว้ทั้ง 8 รูปแบบ จากการเลือกสวิตช์ควบคุม และสามารถให้ชุดควบคุมทำการควบคุมสัญญาณไฟจราจรได้อย่างอัตโนมัติโดยการเลือกสวิตช์ควบคุมสำหรับการควบคุมแบบอัตโนมัติ ซึ่งในการเปิดสัญญาณไฟแต่ละแบบก็จะมีการส่งค่าจากชุดควบคุมตัวลูกแต่ละจุด ไปยังชุดควบคุมตัวแม่แล้วส่งไปแสดงผลบนหน้าจอคอมพิวเตอร์ และในกรณีที่ทั้งชุดควบคุมตัวแม่และชุดควบคุมตัวลูกทำการควบคุมสัญญาณไฟจราจรแบบอัตโนมัติ จะมีการส่งสัญญาณควบคุมจากชุดควบคุมตัวแม่ไปยังชุดควบคุมตัวลูก เพื่อให้การควบคุมสัญญาณไฟจราจรมีความสัมพันธ์กันทุกจุด ซึ่งจะทำให้การเคลื่อนตัวของรถยนต์เป็นไปอย่างต่อเนื่อง

ในการทดลองได้กำหนดค่าแอดเดรสของคอมพิวเตอร์ไว้เท่ากับ 00H ค่าแอดเดรสของชุดควบคุมตัวแม่เท่ากับ 01H ค่าแอดเดรสของชุดควบคุมตัวลูกชุดที่ 1 เท่ากับ AAH ค่าแอดเดรสของชุดควบคุมตัวลูกชุดที่ 2 เท่ากับ BBH และค่าแสดงสถานะของชุดควบคุมตัวแม่เท่ากับ A1H-A8H ค่าแสดงสถานะของชุดควบคุมตัวลูกชุดที่ 1 เท่ากับ 21H-28H ค่าแสดงสถานะของชุดควบคุมตัวลูกชุดที่ 2 เท่ากับ 31H-38H และได้ทำการเขียนโปรแกรมให้มีการรับส่งกันในจุดต่างๆ และทำการวัดสัญญาณที่จุดต่างๆได้ตามผลการทดลองในบทที่ 4 ซึ่งสัญญาณที่วัดได้ในจุดต่างๆก็ถูกต้องตามที่กำหนดไว้ ชุดควบคุมตัวแม่สามารถรับค่าจากชุดควบคุมตัวลูกได้ และสามารถส่งค่าไปแสดงหน้าจอคอมพิวเตอร์ได้ และชุดควบคุมตัวแม่ยังสามารถส่งสัญญาณควบคุมไปให้ชุดควบคุมตัวลูกสามารถทำการควบคุมสัญญาณไฟจราจรได้อย่างต่อเนื่อง

ปัญหาในการทดลอง

การส่งสัญญาณจากชุดควบคุมตัวลูกไปยังชุดควบคุมตัวแม่ ในการส่งสัญญาณเพียงครั้งเดียวทางชุดควบคุมตัวแม่ไม่สามารถรับได้ ชุดควบคุมตัวลูกจึงต้องทำการวนส่งค่าเป็นช่วงเวลาหนึ่ง ทางชุดควบคุมตัวแม่จึงจะสามารถรับค่านั้นและส่งต่อไปยังคอมพิวเตอร์ได้

แนวทางในการพัฒนา

พัฒนาโปรแกรมให้สามารถควบคุมสัญญาณไฟจราจรผ่านทางคอมพิวเตอร์ได้โดยตรง และสามารถที่จะนำการควบคุมแบบมัลติโปรเซสเซอร์ที่ใช้ในโครงงานนี้ไปประยุกต์ใช้กับการใช้งานจริงได้ ซึ่งสามารถเชื่อมต่อโปรเซสเซอร์ได้ถึง 32 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

[1] ผศ. สมยศ จุณณะปิยะ, *การใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51*, กรุงเทพ : โครงการตำราสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2537

[2] พ.อ.เจนวิทย์ เหลืองอร่าม,*การใช้ Turbo C++ เขียนโปรแกรมภาษาซี*,กรุงเทพ :สำนักพิมพ์สุขภาพใจ, 2538

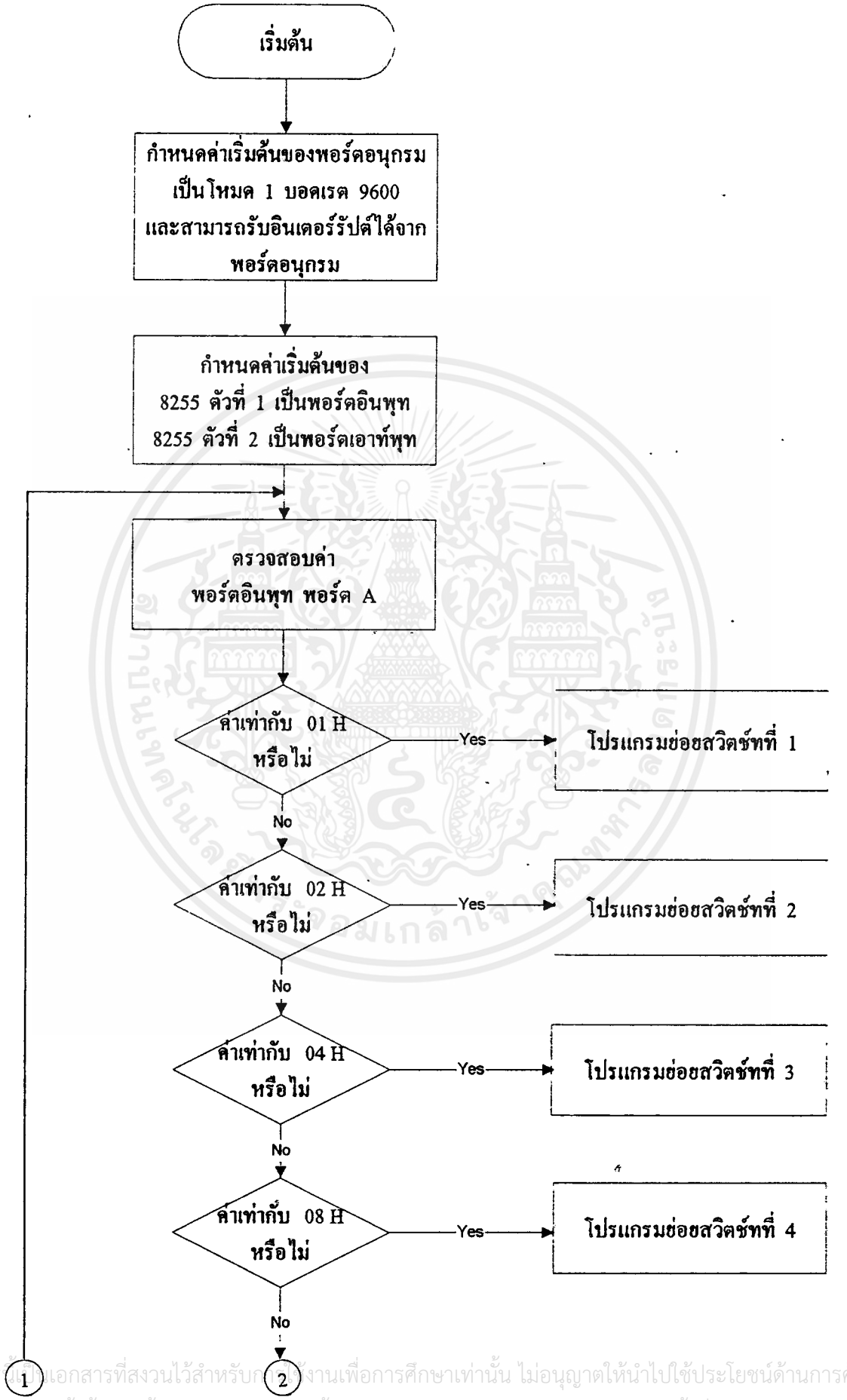
[3] ธันวา ศรีประมง , *การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม* , กรุงเทพ : โครงการตำรามหาวิทยาลัยเทคโนโลยีมหานคร , 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

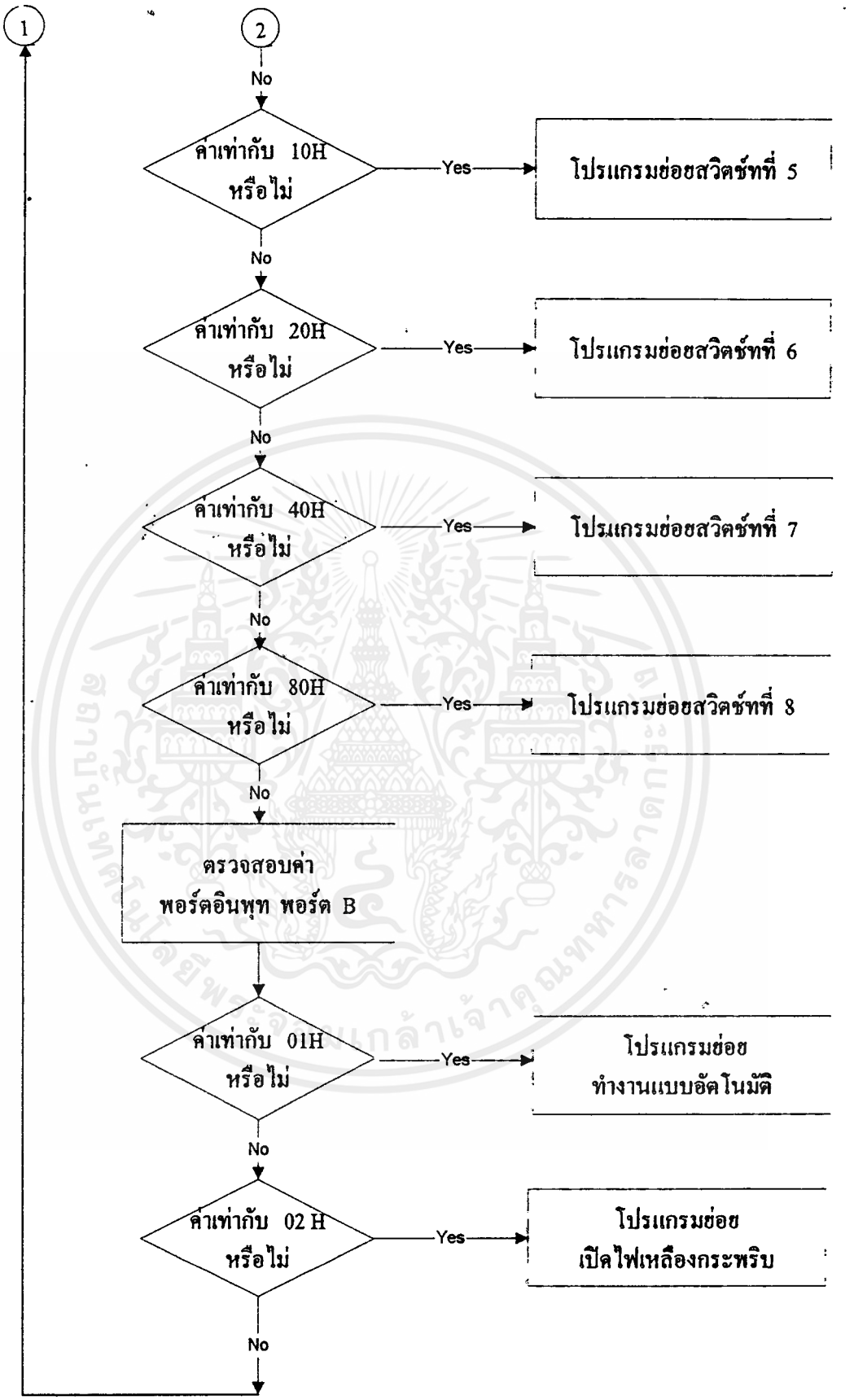
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของโปรแกรมหลักที่ใช้ในการควบคุมสัญญาณไฟจราจร

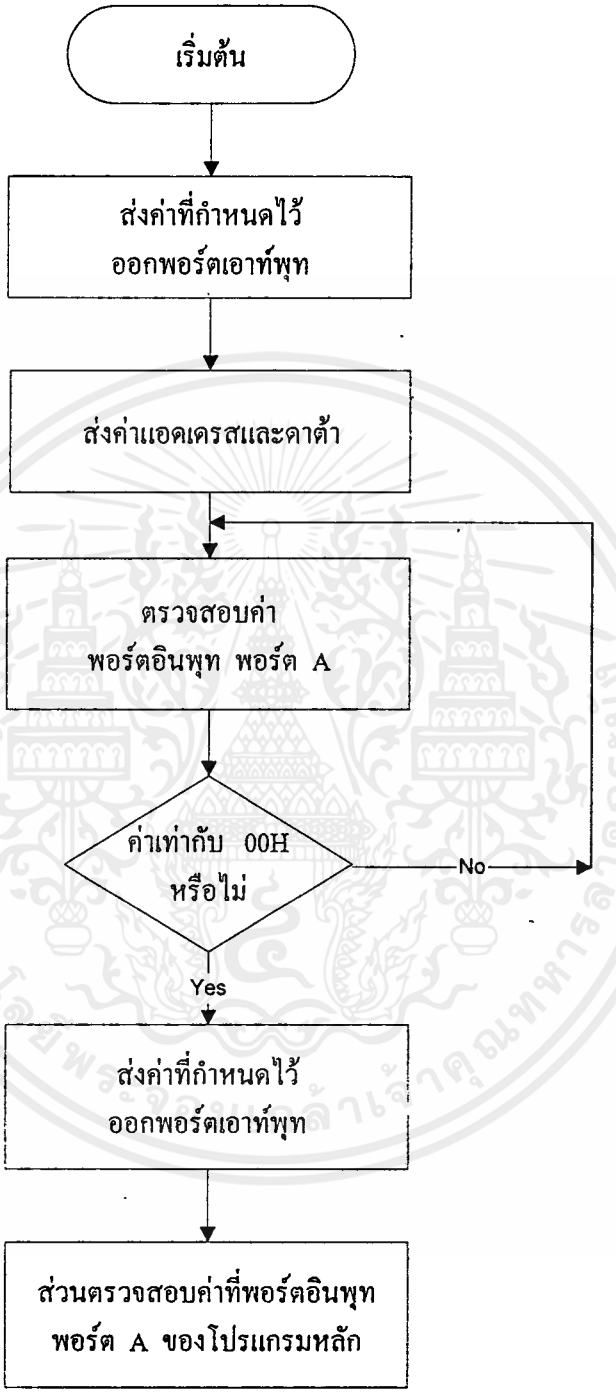


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



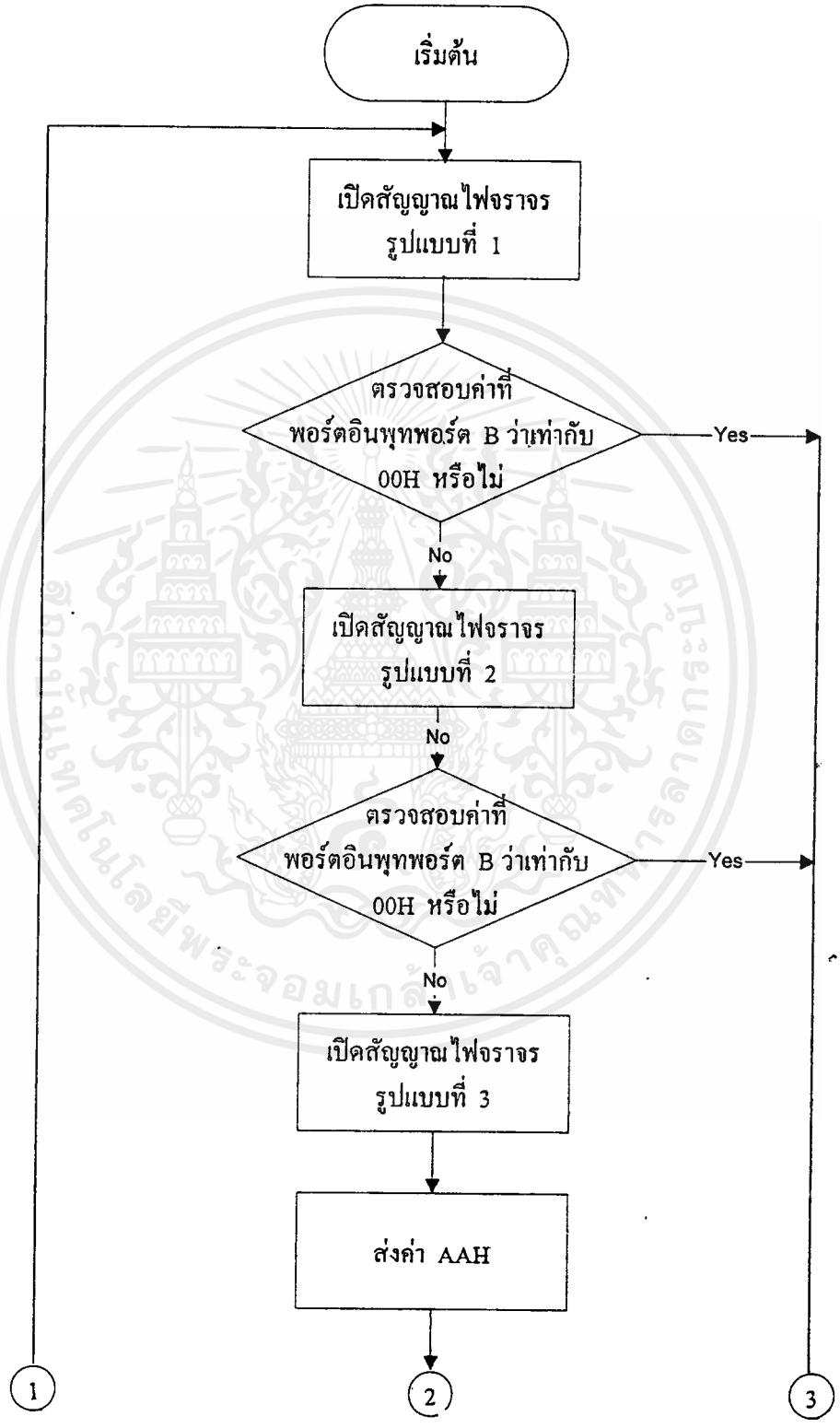
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของโปรแกรมย่อยสวิตช์ที่ 1 ถึง 8

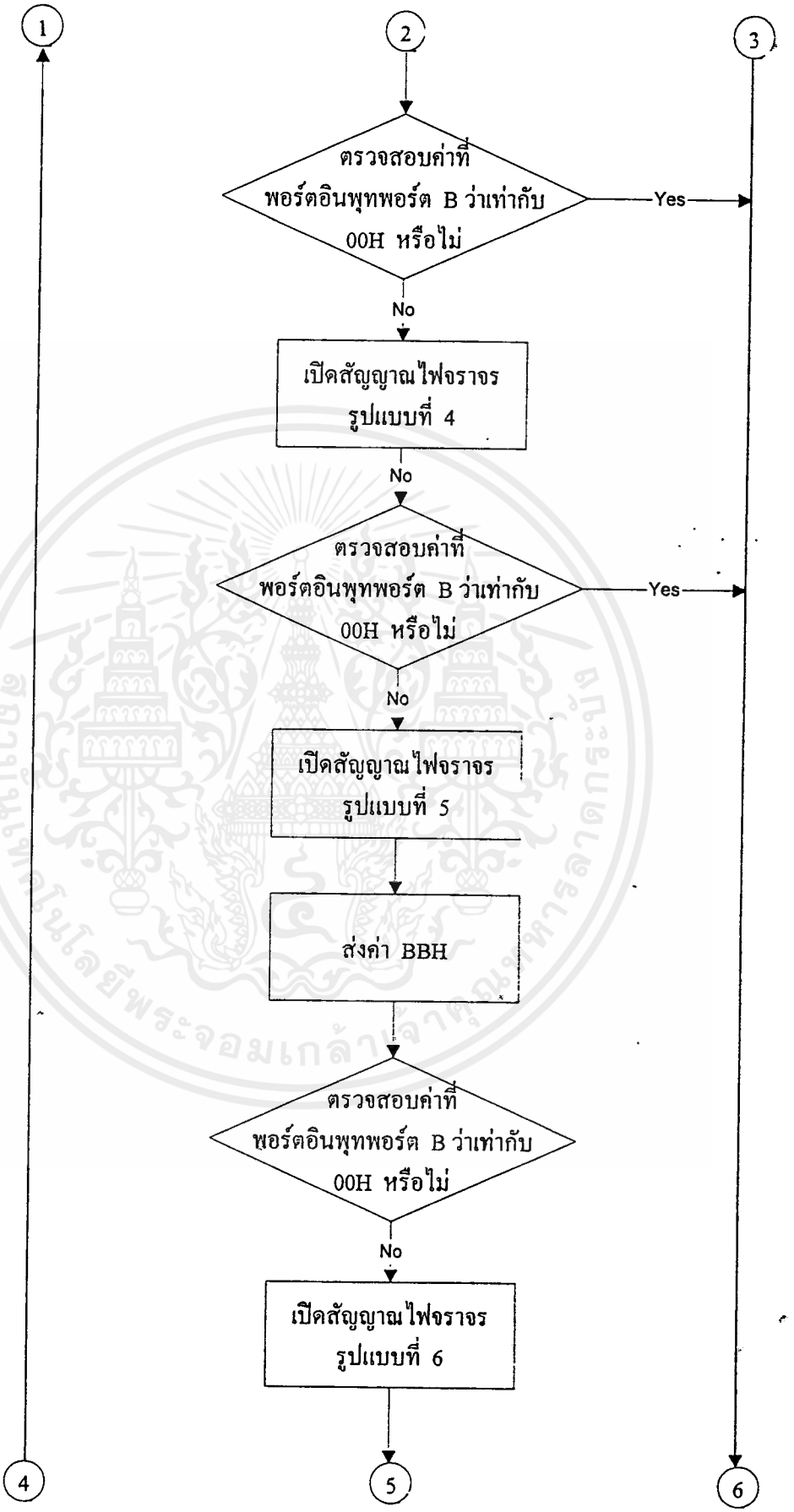


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

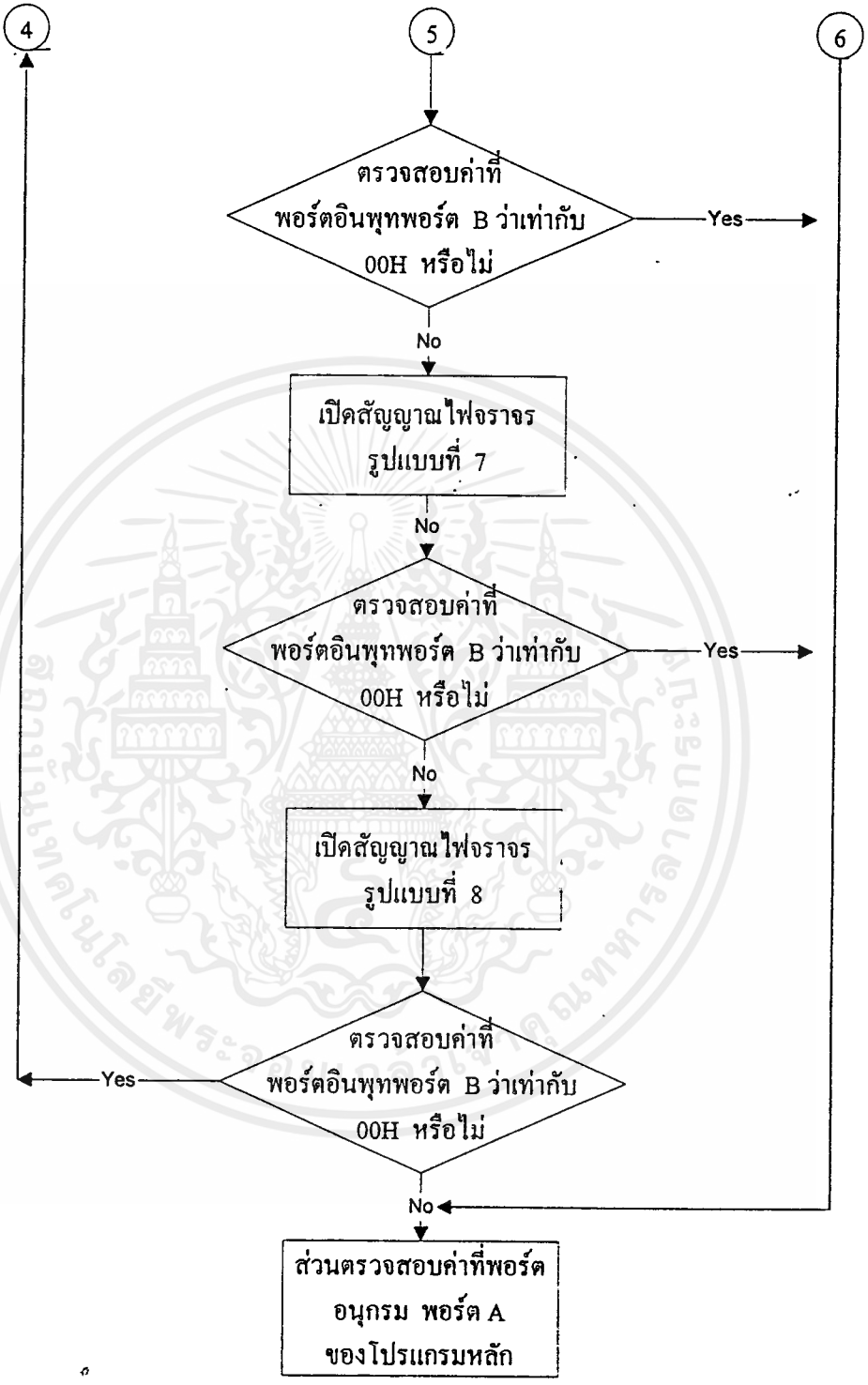
ขั้นตอนการทำงานแบบอัตโนมัติของตัวแม่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

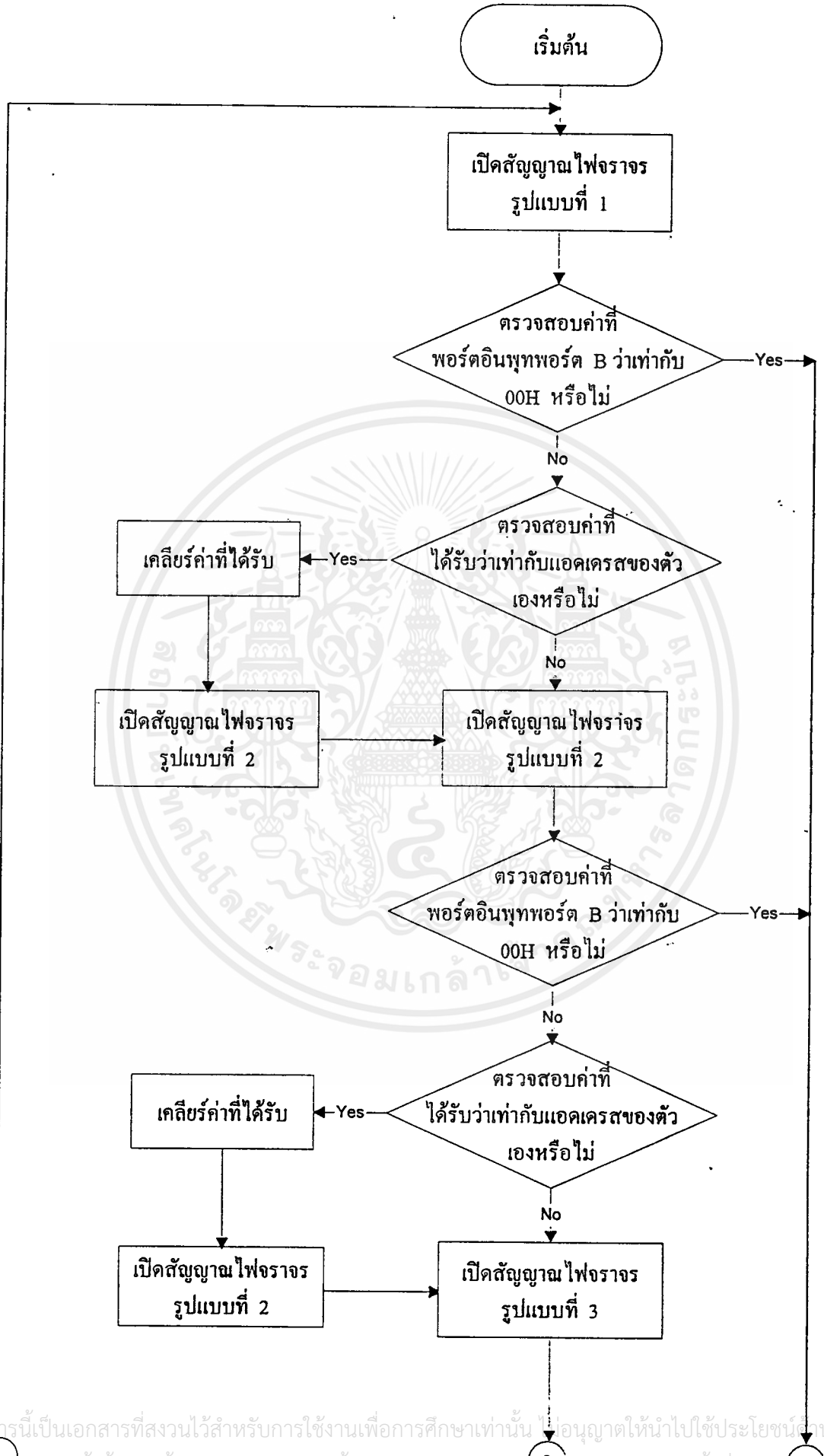


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

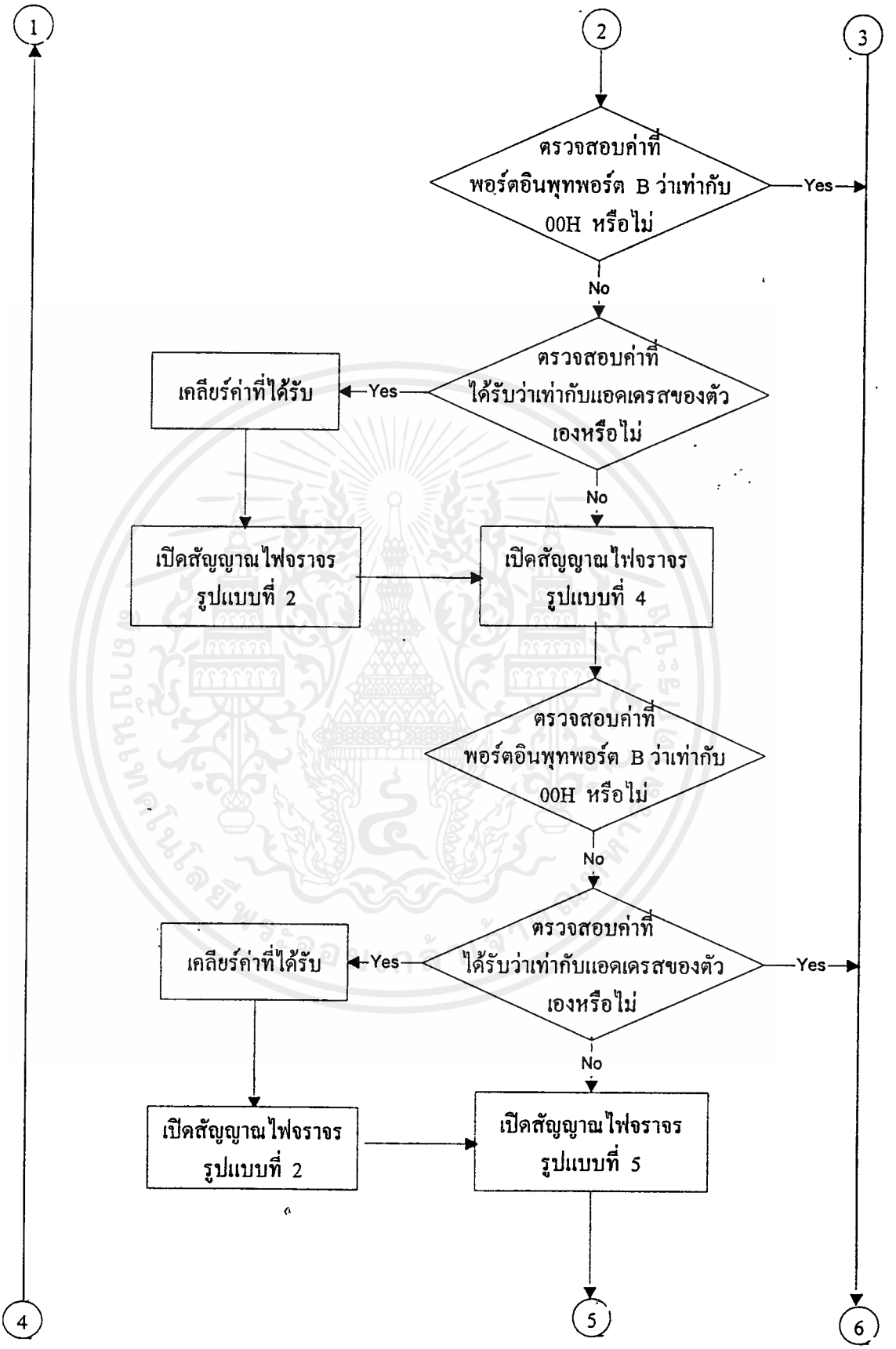


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

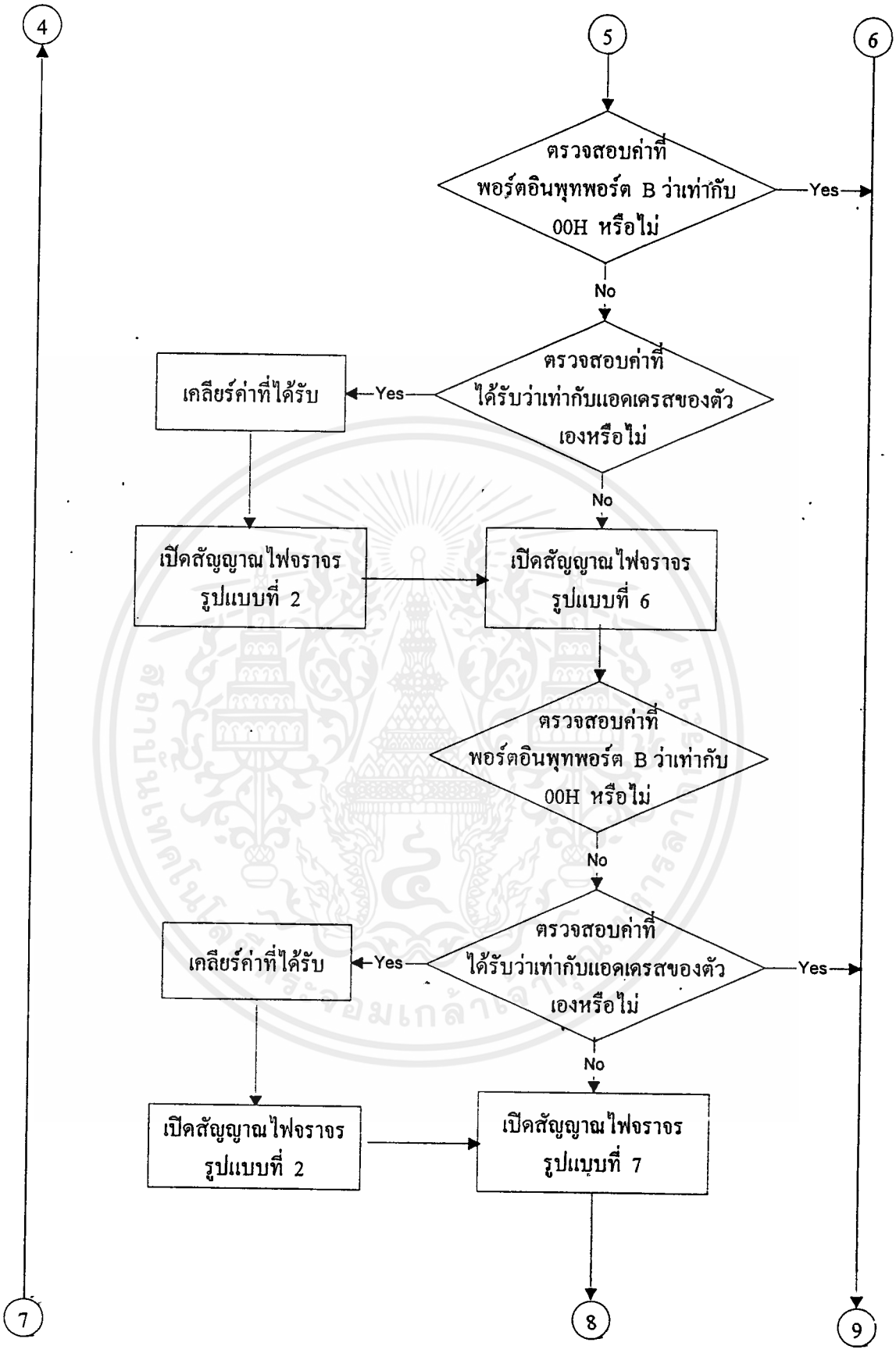
ขั้นตอนการทำงานแบบอัตโนมัติของตัวลูก



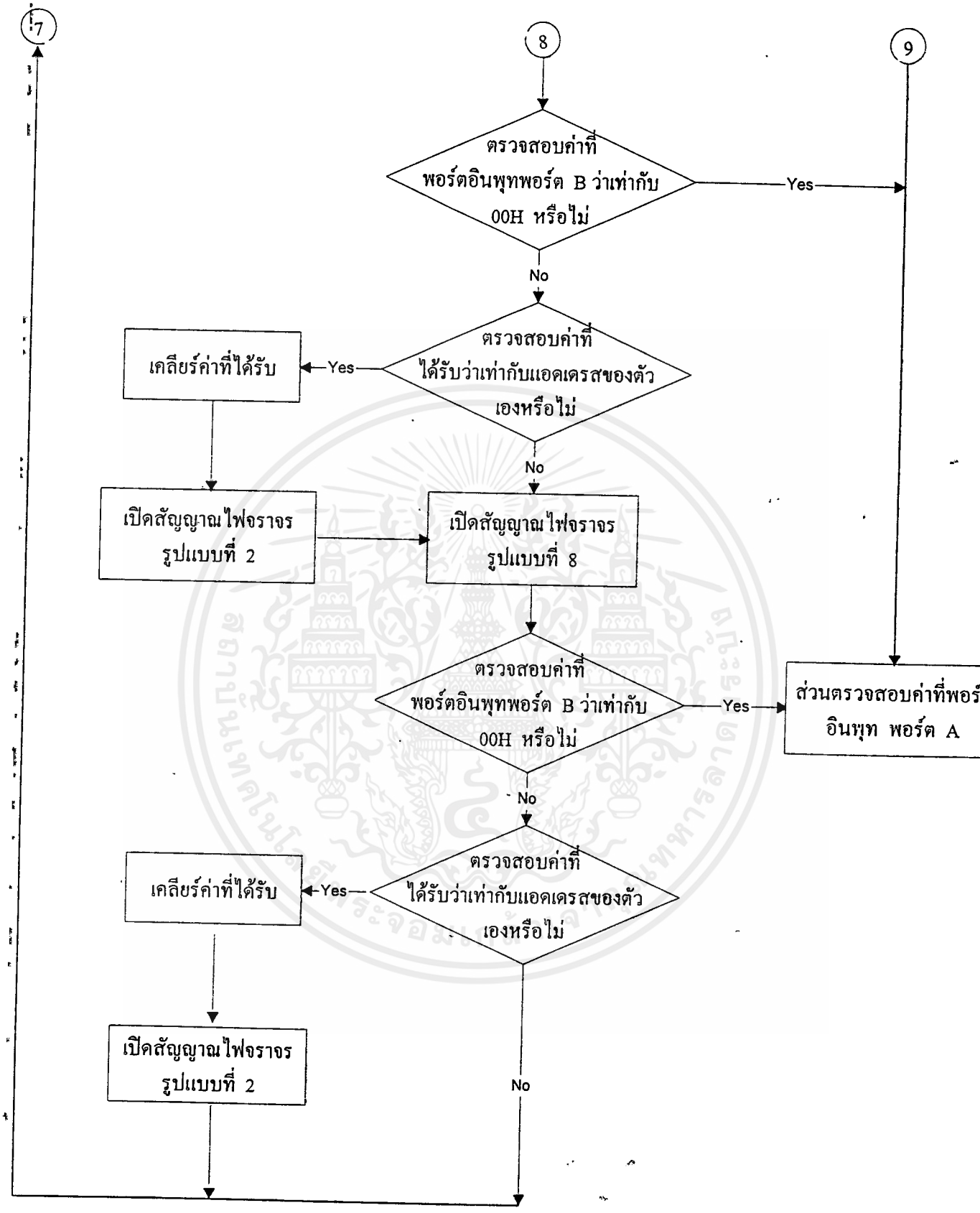
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

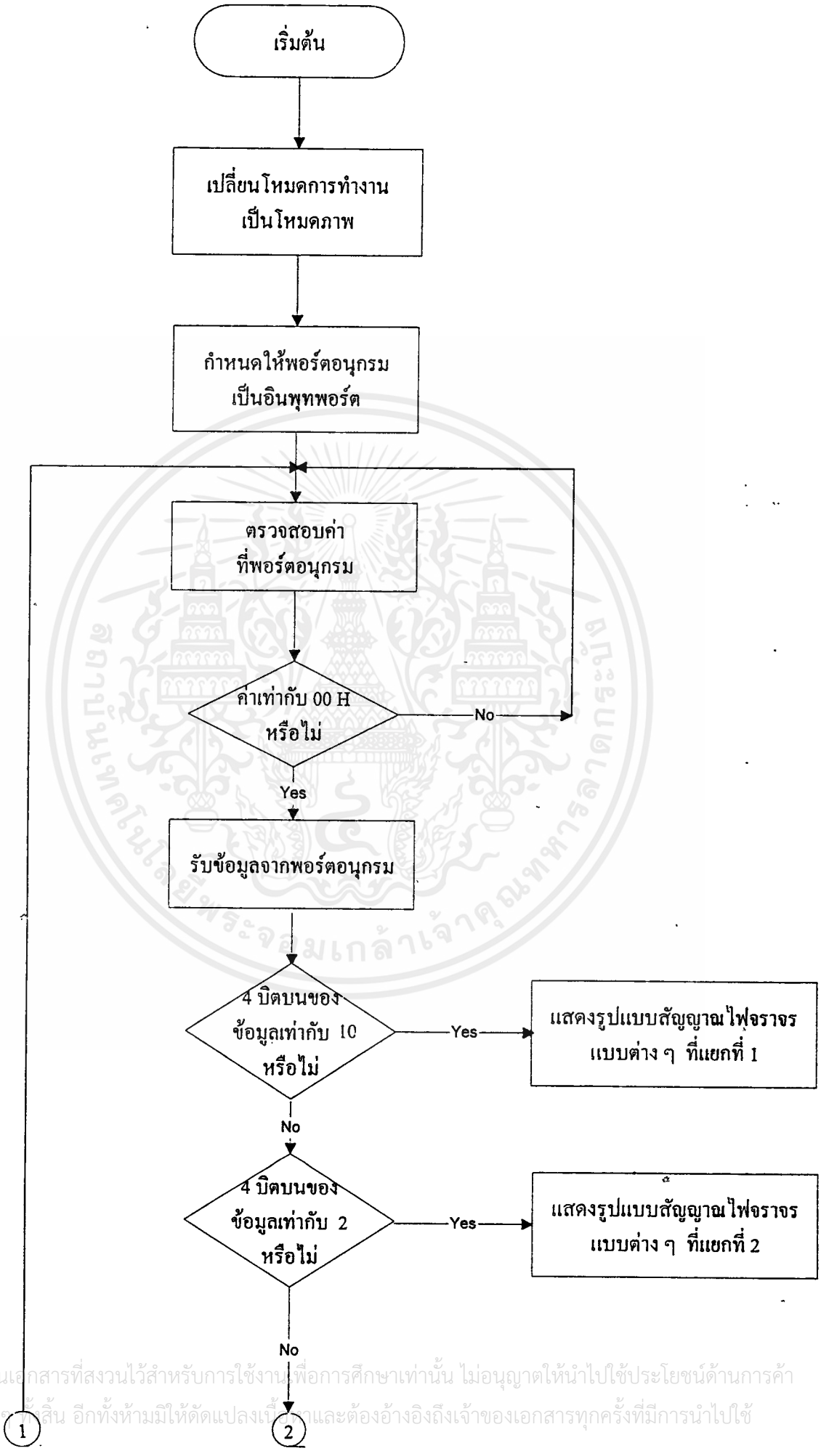


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

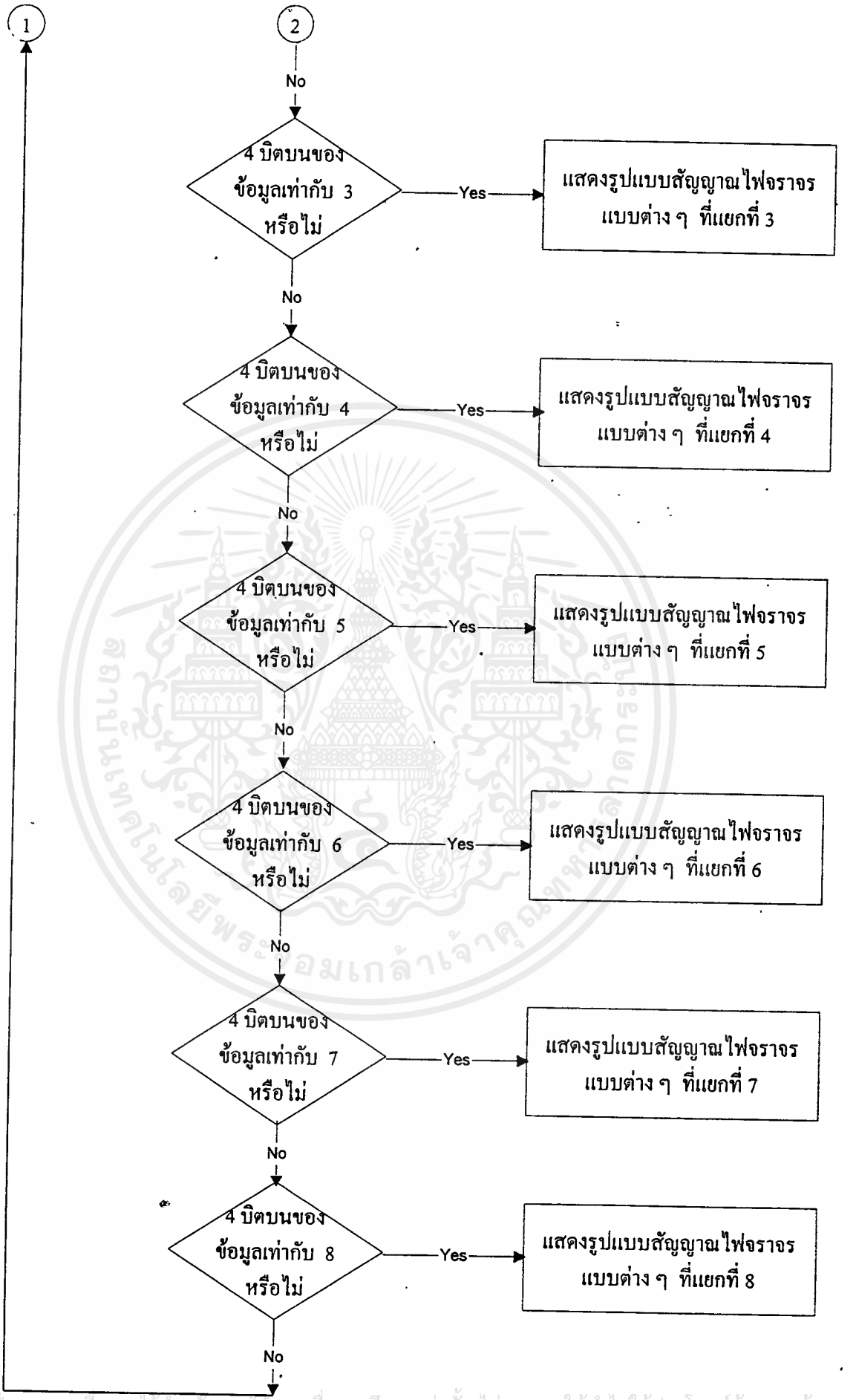


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของการแสดงผลบนหน้าจอคอมพิวเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;FILENAME        MASTER.ASM
;DESCRIPTION     THE 1st POINT TRAFFIC LIGHT CONTROL
;ASSEMBLER       SXA51 ON PC-DOS

            USP1A   EQU   2000H
            USP1B   EQU   2001H
            USP1C   EQU   2002H
            USP1P   EQU   2003H

            USP2A   EQU   4000H
            USP2B   EQU   4001H
            USP2C   EQU   4002H
            USP2P   EQU   4003H

            IO      EQU   P1.4
            CLK     EQU   P1.5
            RST     EQU   P1.6

                ORG 0000H

;INITIAL SERIAL MODE1 TRANSMIT AND RECEIVE ENABLE, 9600 BAUD

            MOV     PCON,#00H           ;SMOD=0
            MOV     SCON,#050H          ;SERIAL MODE1
            MOV     TMOD,#20H           ;TIMER1 (MODE2)
            MOV     TH1,#0FDH           ;9600 BAUD
            SETB    TR1                 ;START TIMER1
            SETB    EA                  ;INTERRUPT ENABLE
            SETB    ES                  ;INTERRUPT FROM SERIAL PORT
            CLR     P1.0                ;RECEIVE
            LJMP    MAIN

;************************************************************
;********** INTERRUPT SERVICE ROUTINE ***********************
;************ FOR RECEIVE AND SEND **************************
;************************************************************

                ORG 0023H

 INTSER:    PUSH    ACC
            PUSH    PSW
            JB      TI,SEND

 ;************* FOR RECEIVE ***********************

            CLR     RI
            JNB     RI,$
            MOV     A,SBUF
            MOV     R6,A                ;MOVE THE FIRST BYTE TO R6
            CLR     RI
            JNB     RI,$
            MOV     A,SBUF
            MOV     R7,A                ;MOVE THE SECOND BYTE TO R7
            CLR     RI
            CJNE    R6,#01H,OUTISR      ;MASTER'S ADDRESS OR ;NOT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;**** SEND THE CURRENT STATUS OF SLAVE TO PC ****

STOPC:      MOV     A,#00H              ;PC'S ADDRESS
            CALL    SBYTE               ;SEND BYTE
            MOV     A,R7                ;DATA FROM SLAVE
            CALL    SBYTE               ;SEND BYTE
OUTISR:     POP     PSW
            POP     ACC
            RETI

;*************** FOR SEND ************************
;****** SEND THE CURRENT STATUS TO PC ************

SEND:       CLR     TI
            CJNE    A,#0AAH,SSL2        ;SEND SLAVE1'S ADDRESS TO LINK
            CALL    SBYTE               ;TO SLAVE1
            LJMP    OUTSEND
SSL2:       CJNE    A,#0BBH,S1          ;SEND SLAVE2'S ADDRESS TO LINK
            CALL    SBYTE               ;TO SLAVE2
            LJMP    OUTSEND
S1:         CJNE    A,#22H,S2           ;SEND CURRENT STATUS TO PC
            MOV     R7,#0A1H
            SJMP    STOPC
S2:         CJNE    A,#88H,S3
            MOV     R7,#0A2H
            SJMP    STOPC
S3:         CJNE    A,#11H,S4
            MOV     R7,#0A3H
            SJMP    STOPC
S4:         CJNE    A,#44H,S5
            MOV     R7,#0A4H
            SJMP    STOPC
S5:         CJNE    A,#30H,S6
            MOV     R7,#0A5H
            SJMP    STOPC
S6:         CJNE    A,#03H,S7
            MOV     R7,#0A6H
            SJMP    STOPC
S7:         CJNE    A,#0C0H,S8
            MOV     R7,#0A7H
            SJMP    STOPC
S8:         CJNE    A,#0CH,OUTSEND
            MOV     R7,#0A8H
            SJMP    STOPC

SBYTE:      SETB    P1.0
            MOV     SBUF,A
            JNB     TI,$
            CLR     TI
            CLR     P1.0
            RET

OUTSEND:    POP     PSW
            POP     ACC
            RETI

;*****************************************************
;*****************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MAIN:   MOV     A,#9BH          ;INITIAL 8255 TO BE INPUT PORT
        MOV     DPTR,#USP1P
        MOVX    @DPTR,A
        MOV     A,#80H          ;INITIAL 8255 TO BE OUTPUT PORT
        MOV     DPTR,#USP2P
        MOVX    @DPTR,A
        LCALL   RTC_INIT

SW_ON:  MOV     DPTR,#USP1A     ;INPUT FROM SWITCH
        MOVX    A,@DPTR

SW1:    CJNE    A,#01H,SW2
        LCALL   PROG1
        LCALL   SW_OFF
        LCALL   YEL1
        LJMP    SW_ON

SW2:    CJNE    A,#02H,SW3
        LCALL   PROG2
        LCALL   SW_OFF
        LCALL   YEL2
        LJMP    SW_ON

SW3:    CJNE    A,#04H,SW4
        LCALL   PROG3
        LCALL   SW_OFF
        LCALL   YEL3
        LJMP    SW_ON

SW4:    CJNE    A,#08H,SW5
        LCALL   PROG4
        LCALL   SW_OFF
        LCALL   YEL4
        LJMP    SW_ON

SW5:    CJNE    A,#10H,SW6
        LCALL   PROG5
        LCALL   SW_OFF
        LCALL   YEL5
        LJMP    SW_ON

SW6:    CJNE    A,#20H,SW7
        LCALL   PROG6
        LCALL   SW_OFF
        LCALL   YEL6
        LJMP    SW_ON

SW7:    CJNE    A,#40H,SW8
        LCALL   PROG7
        LCALL   SW_OFF
        LCALL   YEL7
        LJMP    SW_ON
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SW8:        CJNE    A,#80H,YELLOW
            LCALL   PROG8
            LCALL   SW_OFF
            LCALL   YEL8
SW:         LJMP    SW_ON

YELLOW:     MOV     DPTR,#USP1B
            MOVX    A,@DPTR
            CJNE    A,#02H,AUTO
START:      MOV     A,#00H
            LCALL   OUT2A
            LCALL   OUT2C
            MOV     A,#0FFH
            LCALL   OUT2B
            MOV     R5,#1D
            LCALL   TIMER
            MOV     A,#00H
            LCALL   OUT2B
            MOV     R5,#1D
            LCALL   TIMER
            LCALL   AUTO_OFF
            SJMP    START

AUTO:       CJNE    A,#01H,SW
RUNAUTO:    LCALL   PROG1
            MOV     R5,#80D
            LCALL   TIMER
            LCALL   YEL1
            MOV     R5,#2D
            LCALL   TIMER
            LCALL   AUTO_OFF

            LCALL   PROG2
            MOV     R5,#80D
            LCALL   TIMER
            LCALL   YEL2
            MOV     R5,#2D
            LCALL   TIMER
            LCALL   AUTO_OFF

            LCALL   PROG3
            MOV     R5,#80D
            LCALL   LINK1                   ;LINK TO SLAVE1
            LCALL   TIMER
            LCALL   YEL3
            MOV     R5,#2D
            LCALL   TIMER
            LCALL   AUTO_OFF

            LCALL   PROG4
            MOV     R5,#80D
            LCALL   TIMER
            LCALL   YEL4
            MOV     R5,#2D
            LCALL   TIMER
            LCALL   AUTO_OFF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    LCALL   PROG5
                    MOV     R5,#80D
                    LCALL   LINK2                   ;LINK TO. SLAVE2
                    LCALL   TIMER
                    LCALL   YEL5
                    MOV     R5,#2D
                    LCALL   TIMER
                    LCALL   AUTO_OFF

                    LCALL   PROG6
                    MOV     R5,#80D
                    LCALL   TIMER
                    LCALL   YEL6
                    MOV     R5,#2D
                    LCALL   TIMER
                    LCALL   AUTO_OFF

                    LCALL   PROG7
                    MOV     R5,#80D
                    LCALL   TIMER
                    LCALL   YEL7
                    MOV     R5,#2D
                    LCALL   TIMER
                    LCALL   AUTO_OFF

                    LCALL   PROG8
                    MOV     R5,#80D
                    LCALL   TIMER
                    LCALL   YEL8
                    MOV     R5,#2D
                    LCALL   TIMER
                    LCALL   AUTO_OFF
                    LJMP    RUNAUTO

LINK1:              MOV     A,#0AAH
                    SETB    TI
                    RET
LINK2:              MOV     A,#0BBH
                    SETB    TI
                    RET

PROG1:              MOV     A,#0DDH
                    LCALL   OUT2A
                    MOV     A,#00H
                    LCALL   OUT2B
                    MOV     A,#22H
                    LCALL   OUT2C
                    SETB    TI
                    RET

PROG2:              MOV     A,#77H
                    LCALL   OUT2A
                    MOV     A,#00H
                    LCALL   OUT2B
                    MOV     A,#88H
                    LCALL   OUT2C
                    SETB    TI
                    RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PROG3:      MOV     A,#0EEH
            LCALL   OUT2A
            MOV     A,#00H
            LCALL   OUT2B
            MOV     A,#11H
            LCALL   OUT2C
            SETB    TI
            RET

PROG4:      MOV     A,#0BBH
            LCALL   OUT2A
            MOV     A,#00H
            LCALL   OUT2B
            MOV     A,#44H
            LCALL   OUT2C
            SETB    TI
            RET

PROG5:      MOV     A,#0FCH
            LCALL   OUT2A
            MOV     A,#00H
            LCALL   OUT2B
            MOV     A,#03H
            LCALL   OUT2C
            SETB    TI
            RET

PROG6:      MOV     A,#0CFH
            LCALL   OUT2A
            MOV     A,#00H
            LCALL   OUT2B
            MOV     A,#30H
            LCALL   OUT2C
            SETB    TI
            RET

PROG7:      MOV     A,#3FH
            LCALL   OUT2A
            MOV     A,#00H
            LCALL   OUT2B
            MOV     A,#0C0H
            LCALL   OUT2C
            SETB    TI
            RET

PROG8:      MOV     A,#0F3H
            LCALL   OUT2A
            MOV     A,#00H
            LCALL   OUT2B
            MOV     A,#0CH
            LCALL   OUT2C
            SETB    TI
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SW_OFF:     MOV     DPTR,#USP1A
            MOVX    A,@DPTR
            CJNE    A,#00H,SW_OFF
            RET

YEL1:       MOV     A,#22H
            LCALL   OUT2B
            MOV     A,#00H
            LCALL   OUT2C
            RET
YEL2:       MOV     A,#88H
            LCALL   OUT2B
            MOV     A,#00H
            LCALL   OUT2C
            RET
YEL3:       MOV     A,#11H
            LCALL   OUT2B
            MOV     A,#00H
            LCALL   OUT2C
            RET
YEL4:       MOV     A,#44H
            LCALL   OUT2B
            MOV     A,#00H
            LCALL   OUT2C
            RET
YEL5:       MOV     A,#03H
            LCALL   OUT2B
            MOV     A,#00H
            LCALL   OUT2C
            RET
YEL6:       MOV     A,#30H
            LCALL   OUT2B
            MOV     A,#00H
            LCALL   OUT2C
            RET
YEL7:       MOV     A,#0C0H
            LCALL   OUT2B
            MOV     A,#00H
            LCALL   OUT2C
            RET
YEL8:       MOV     A,#0CH
            LCALL   OUT2B
            MOV     A,#00H
            LCALL   OUT2C
            RET

OUT2A:      MOV     DPTR,#USP2A
            MOVX    @DPTR,A
            RET
OUT2B:      MOV     DPTR,#USP2B
            MOVX    @DPTR,A
            RET
OUT2C:      MOV     DPTR,#USP2C
            MOVX    @DPTR,A
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RTC_INIT:   CLR     RST
            SETB    CLK
            MOV     R2,#8EH
            MOV     R3,#00H
            LCALL   WRT
            MOV     R2,#80H
            MOV     R3,#00H
            LCALL   WRT
            MOV     R2,#82H
            MOV     R3,#00H
            LCALL   WRT
            MOV     R2,#84H
            MOV     R3,#00H
            LCALL   WRT
            MOV     R2,#8EH
            MOV     R3,#80H
            LCALL   WRT
            RET
WRT:        CLR     CLK
            NOP
            SETB    RST
            NOP
            MOV     A,R2
            LCALL   RTCWR8
            MOV     A,R3
            LCALL   RTCWR8
            CLR     RST
            NOP
            RET

TIMER:      CALL    RTC
            CALL    COMP
BO1:        CALL    RTC
            ANL     A,#01H
            CJNE    A,30H,EXIT
            SJMP    BO1
COMP:       ANL     A,#01H
            MOV     30H,A
            RET
EXIT:       MOV     DPTR,#USP1B
            MOVX    A,@DPTR
            CJNE    A,#00H,BO2
            MOV     R5,#1D
BO2:        DEC     R5
            CJNE    R5,#0H,TIMER
            RET

RTC:        MOV     R2,#81H
            CLR     CLK
            NOP
            SETB    RST
            NOP
            MOV     A,R2
            CALL    RTCWR8
            MOV     R4,#08H
            CLR     A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RTCRD:      CLR     CLK                 ;READ FROM RTC
            NOP
            MOV     C,IO
            RRC     A
            SETB    CLK
            NOP
            DJNZ    R4,RTCRD
            CLR     RST
            NOP
            RET
RTCWR8:     MOV     R4,#08H             ;WRITE TO RTC
WR8:        RRC     A
            MOV     IO,C
            SETB    CLK
            NOP
            CLR     CLK
            NOP
            DJNZ    R4,WR8
            RET

AUTO_OFF:   MOV     DPTR,#USP1B
            MOVX    A,@DPTR
            CJNE    A,#00H,BACK
            LJMP    SW_ON
BACK:       RET

            END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;FILENAME       SLAVE1.ASM
;DESCIRPTION    THE 2nd POINT TRAFFIC LIGHT CONTROL
;ASSEMBLER      SXA51 ON PC-DOS

               USP1A   EQU   2000H
               USP1B   EQU   2001H
               USP1C   EQU   2002H
               USP1P   EQU   2003H

               USP2A   EQU   4000H
               USP2B   EQU   4001H
               USP2C   EQU   4002H
               USP2P   EQU   4003H

               IO      EQU   P1.4
               CLK     EQU   P1.5
               RST     EQU   P1.6

                   ORG 0000H

;INITIAL SERIAL MODE1 TRANSMIT AND RECEIVE ENABLE, 9600 BAUD

               MOV     PCON,#00H          ;SMOD=0
               MOV     SCON,#50H          ;SERIAL MODE1
               MOV     TMOD,#20H          ;TIMER1 (MODE2)
               MOV     TH1,#0FDH          ;9600 BAUD
               SETB    TR1                ;START TIMER1
               SETB    EA                 ;INTERRUPT ENABLE
               SETB    ES                 ;INTERRUPT FROM SERIAL PORT
               CLR     P1.0               ;RECEIVE
               LJMP    MAIN

;***********************************************
;********** INTERRUPT SERVICE ROUTINE ***********
;***********************************************

                   ORG 0023H

INTSER:        PUSH    ACC
               PUSH    PSW
               JB      TI,SEND
               CLR     RI
               MOV     A,SBUF
               CJNE    A,#0AAH,OUTISR     ;RECEIVE LINK FROM MASTER
               MOV     R6,A
OUTISR:        CLR     RI
               POP     PSW
               POP     ACC
               RETI
SEND:          CLR     TI                 ;SEND THE CURRENT STATUS TO MASTER
               MOV     R2,#4H
               CJNE    A,#22H,S2
TX1:           MOV     A,#01H
               CALL    SBYTE
               MOV     A,#21H
               CALL    SBYTE
               DJNZ    R2,TX1
               LJMP    OUTSEND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
S2:     CJNE    A,#88H,S3
TX2:    MOV     A,#01H
        CALL    SBYTE
        MOV     A,#22H
        CALL    SBYTE
        DJNZ    R2,TX2
        SJMP    OUTSEND
S3:     CJNE    A,#11H,S4
TX3:    MOV     A,#01H
        CALL    SBYTE
        MOV     A,#23H
        CALL    SBYTE
        DJNZ    R2,TX3
        SJMP    OUTSEND
S4:     CJNE    A,#44H,S5
TX4:    MOV     A,#01H
        CALL    SBYTE
        MOV     A,#24H
        CALL    SBYTE
        DJNZ    R2,TX4
        SJMP    OUTSEND
S5:     CJNE    A,#03H,S6
TX5:    MOV     A,#01H
        CALL    SBYTE
        MOV     A,#25H
        CALL    SBYTE
        DJNZ    R2,TX5
        SJMP    OUTSEND
S6:     CJNE    A,#30H,S7
TX6:    MOV     A,#01H
        CALL    SBYTE
        MOV     A,#26H
        CALL    SBYTE
        DJNZ    R2,TX6
        SJMP    OUTSEND
S7:     CJNE    A,#0C0H,S8
TX7:    MOV     A,#01H
        CALL    SBYTE
        MOV     A,#27H
        CALL    SBYTE
        DJNZ    R2,TX7
        SJMP    OUTSEND
S8:     CJNE    A,#0CH,OUTSEND
TX8:    MOV     A,#01H
        CALL    SBYTE
        MOV     A,#28H
        CALL    SBYTE
        DJNZ    R2,TX8
        SJMP    OUTSEND

SBYTE:  SETB    P1.0
        MOV     SBUF,A
        JNB     TI,$
        CLR     TI
        CLR     P1.0
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OUTSEND:    POP     PSW
            POP     ACC
            RETI

;*****************************************************
;*****************************************************

MAIN:       MOV     A,#9BH          ;INITIAL 8255 TO BE INPUT PORT
            MOV     DPTR,#USP1P
            MOVX    @DPTR,A
            MOV     A,#80H          ;INITIAL 8255 TO BE OUTPUT PORT
            MOV     DPTR,#USP2P
            MOVX    @DPTR,A
            LCALL   RTC_INIT

SW_ON:      MOV     DPTR,#USP1A     ;INPUT FROM SWITCH
            MOVX    A,@DPTR
SW1:        CJNE    A,#01H,SW2
            LCALL   PROG1
            LCALL   SW_OFF
            LCALL   YEL1
            LJMP    SW_ON

SW2:        CJNE    A,#02H,SW3
            LCALL   PROG2
            LCALL   SW_OFF
            LCALL   YEL2
            LJMP    SW_ON

SW3:        CJNE    A,#04H,SW4
            LCALL   PROG3
            LCALL   SW_OFF
            LCALL   YEL3
            LJMP    SW_ON

SW4:        CJNE    A,#08H,SW5
            LCALL   PROG4
            LCALL   SW_OFF
            LCALL   YEL4
            LJMP    SW_ON

SW5:        CJNE    A,#10H,SW6
            LCALL   PROG5
            LCALL   SW_OFF
            LCALL   YEL5
            LJMP    SW_ON

SW6:        CJNE    A,#20H,SW7
            LCALL   PROG6
            LCALL   SW_OFF
            LCALL   YEL6
            LJMP    SW_ON

SW7:        CJNE    A,#40H,SW8
            LCALL   PROG7
            LCALL   SW_OFF
            LCALL   YEL7
            LJMP    SW_ON
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SW8:        CJNE    A,#80H,YELLOW
            LCALL   PROG8
            LCALL   SW_OFF
            LCALL   YEL8
SW:         LJMP    SW_ON

YELLOW:     MOV     DPTR,#USP1B
            MOVX    A,@DPTR
            CJNE    A,#02H,AUTO
START:      MOV     A,#00H
            LCALL   OUT2A
            LCALL   OUT2C
            MOV     A,#0FFH
            LCALL   OUT2B
            MOV     R5,#1D
            LCALL   TIMER
            MOV     A,#00H
            LCALL   OUT2B
            MOV     R5,#1D
            LCALL   TIMER
            LCALL   AUTO_OFF
            SJMP    START

AUTO:       CJNE    A,#01H,SW
AUTO1:      LCALL   PROG1
            MOV     R5,#80D
            LCALL   TIMER
            LCALL   YEL1
            MOV     R5,#2D
            LCALL   TIMER
            LCALL   AUTO_OFF
            MOV     R6,#00H

AUTO2:      LCALL   PROG2
            MOV     R5,#80D
            LCALL   TIMER
            LCALL   YEL2
            MOV     R5,#2D
            LCALL   TIMER
            LCALL   AUTO_OFF
            CJNE    R6,#0AAH,AUTO3
            CALL    LINK

AUTO3:      LCALL   PROG3
            MOV     R5,#80D
            LCALL   TIMER
            LCALL   YEL3
            MOV     R5,#2D
            LCALL   TIMER
            LCALL   AUTO_OFF
            CJNE    R6,#0AAH,AUTO4
            CALL    LINK

AUTO4:      LCALL   PROG4
            MOV     R5,#80D
            LCALL   TIMER
            LCALL   YEL4
            MOV     R5,#2D
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCALL   TIMER
                LCALL   AUTO_OFF
                CJNE    R6,#0AAH,AUTO5
                CALL    LINK

AUTO5:          LCALL   PROG5
                MOV     R5,#80D
                LCALL   TIMER
                LCALL   YEL5
                MOV     R5,#2D
                LCALL   TIMER
                LCALL   AUTO_OFF
                CJNE    R6,#0AAH,AUTO6
                CALL    LINK

AUTO6:          LCALL   PROG6
                MOV     R5,#80D
                LCALL   TIMER
                LCALL   YEL6
                MOV     R5,#2D
                LCALL   TIMER
                LCALL   AUTO_OFF
                CJNE    R6,#0AAH,AUTO7
                CALL    LINK

AUTO7:          LCALL   PROG7
                MOV     R5,#80D
                LCALL   TIMER
                LCALL   YEL7
                MOV     R5,#2D
                LCALL   TIMER
                LCALL   AUTO_OFF
                CJNE    R6,#0AAH,AUTO8
                CALL    LINK

AUTO8:          LCALL   PROG8
                MOV     R5,#80D
                LCALL   TIMER
                LCALL   YEL8
                MOV     R5,#2D
                LCALL   TIMER
                LCALL   AUTO_OFF
                CJNE    R6,#0AAH,RUNAUTO
                CALL    LINK
RUNAUTO:        LJMP    AUTO1

LINK:           MOV     R6,#00H
                LCALL   PROG2
                MOV     R5,#80D
                LCALL   TIMER
                LCALL   YEL2
                MOV     R5,#2D
                LCALL   TIMER
                LCALL   AUTO_OFF
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PROG1:  MOV     A,#0DDH
        LCALL   OUT2A
        MOV     A,#00H
        LCALL   OUT2B
        MOV     A,#22H
        LCALL   OUT2C
        SETB    TI
        RET

PROG2:  MOV     A,#77H
        LCALL   OUT2A
        MOV     A,#00H
        LCALL   OUT2B
        MOV     A,#88H
        LCALL   OUT2C
        SETB    TI
        RET

PROG3:  MOV     A,#0EEH
        LCALL   OUT2A
        MOV     A,#00H
        LCALL   OUT2B
        MOV     A,#11H
        LCALL   OUT2C
        SETB    TI
        RET

PROG4:  MOV     A,#0BBH
        LCALL   OUT2A
        MOV     A,#00H
        LCALL   OUT2B
        MOV     A,#44H
        LCALL   OUT2C
        SETB    TI
        RET

PROG5:  MOV     A,#0FCH
        LCALL   OUT2A
        MOV     A,#00H
        LCALL   OUT2B
        MOV     A,#03H
        LCALL   OUT2C
        SETB    TI
        RET

PROG6:  MOV     A,#0CFH
        LCALL   OUT2A
        MOV     A,#00H
        LCALL   OUT2B
        MOV     A,#30H
        LCALL   OUT2C
        SETB    TI
        RET

PROG7:  MOV     A,#3FH
        LCALL   OUT2A
        MOV     A,#00H
        LCALL   OUT2B
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     A,#0C0H
                LCALL   OUT2C
                SETB    TI
                RET

PROG8:          MOV     A,#0F3H
                LCALL   OUT2A
                MOV     A,#00H
                LCALL   OUT2B
                MOV     A,#0CH
                LCALL   OUT2C
                SETB    TI
                RET
SW_OFF:         MOV     DPTR,#USP1A
                MOVX    A,@DPTR
                CJNE    A,#00H,SW_OFF
                RET

YEL1:           MOV     A,#22H
                LCALL   OUT2B
                MOV     A,#00H
                LCALL   OUT2C
                RET
YEL2:           MOV     A,#88H
                LCALL   OUT2B
                MOV     A,#00H
                LCALL   OUT2C
                RET
YEL3:           MOV     A,#11H
                LCALL   OUT2B
                MOV     A,#00H
                LCALL   OUT2C
                RET
YEL4:           MOV     A,#44H
                LCALL   OUT2B
                MOV     A,#00H
                LCALL   OUT2C
                RET
YEL5:           MOV     A,#03H
                LCALL   OUT2B
                MOV     A,#00H
                LCALL   OUT2C
                RET
YEL6:           MOV     A,#30H
                LCALL   OUT2B
                MOV     A,#00H
                LCALL   OUT2C
                RET
YEL7:           MOV     A,#0C0H
                LCALL   OUT2B
                MOV     A,#00H
                LCALL   OUT2C
                RET
YEL8:           MOV     A,#0CH
                LCALL   OUT2B
                MOV     A,#00H
                LCALL   OUT2C
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OUT2A:      MOV     DPTR,#USP2A
            MOVX    @DPTR,A
            RET
OUT2B:      MOV     DPTR,#USP2B
            MOVX    @DPTR,A
            RET
OUT2C:      MOV     DPTR,#USP2C
            MOVX    @DPTR,A
            RET

RTC_INIT:   CLR     RST
            SETB    CLK
            MOV     R2,#8EH
            MOV     R3,#00H
            LCALL   WRT
            MOV     R2,#80H
            MOV     R3,#00H
            LCALL   WRT
            MOV     R2,#82H
            MOV     R3,#00H
            LCALL   WRT
            MOV     R2,#84H
            MOV     R3,#00H
            LCALL   WRT
            MOV     R2,#8EH
            MOV     R3,#80H
            LCALL   WRT
            RET
WRT:        CLR     CLK
            NOP
            SETB    RST
            NOP
            MOV     A,R2
            LCALL   RTCWR8
            MOV     A,R3
            LCALL   RTCWR8
            CLR     RST
            NOP
            RET

TIMER:      CALL    RTC
            CALL    COMP
BO1:        CALL    RTC
            ANL     A,#01H
            CJNE    A,30H,EXIT
            SJMP    BO1
COMP:       ANL     A,#01H
            MOV     30H,A
            RET
EXIT:       MOV     DPTR,#USP1B
            MOVX    A,@DPTR
            CJNE    A,#00H,BO2
            MOV     R5,#1D
BO2:        DEC     R5
            CJNE    R5,#0H,TIMER
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RTC:        MOV     R2,#81H
            CLR     CLK
            NOP
            SETB    RST
            NOP
            MOV     A,R2
            CALL    RTCWR8
            MOV     R4,#08H
            CLR     A
RTCRD:      CLR     CLK                 ;READ FROM RTC
            NOP
            MOV     C,IO
            RRC     A
            SETB    CLK
            NOP
            DJNZ    R4,RTCRD
            CLR     RST
            NOP
            RET
RTCWR8:     MOV     R4,#08H             ;WRITE TO RTC
WR8:        RRC     A
            MOV     IO,C
            SETB    CLK
            NOP
            CLR     CLK
            NOP
            DJNZ    R4,WR8
            RET

AUTO_OFF:   MOV     DPTR,#USP1B
            MOVX    A,@DPTR
            CJNE    A,#00H,BACK
            LJMP    SW_ON
BACK:       RET

            END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;FILENAME       SLAVE2.ASM
;DESCIRPTION    THE 3rd POINT TRAFFIC LIGHT CONTROL
;ASSEMBLER      SXA51 ON PC-DOS

            USP1A   EQU   2000H
            USP1B   EQU   2001H
            USP1C   EQU   2002H
            USP1P   EQU   2003H

            USP2A   EQU   4000H
            USP2B   EQU   4001H
            USP2C   EQU   4002H
            USP2P   EQU   4003H

            IO      EQU   P1.4
            CLK     EQU   P1.5
            RST     EQU   P1.6

                ORG 0000H

;INITIAL SERIAL MODE1 TRANSMIT AND RECEIVE ENABLE, 9600 BAUD

            MOV     PCON,#00H           ;SMOD=0
            MOV     SCON,#50H           ;SERIAL MODE1
            MOV     TMOD,#20H           ;TIMER1 (MODE2)
            MOV     TH1,#0FDH           ;9600 BAUD
            SETB    TR1                 ;START TIMER1
            SETB    EA                  ;INTERRUPT ENABLE
            SETB    ES                  ;INTERRUPT FROM SERIAL PORT
            CLR     P1.0                ;RECEIVE
            LJMP    MAIN

;************************************************
;********** INTERRUPT SERVICE ROUTINE ***********
;************************************************

                ORG 0023H

INTSER:     PUSH    ACC
            PUSH    PSW
            JB      TI,SEND
            CLR     RI
            MOV     A,SBUF
            CJNE    A,#0BBH,OUTISR      ;RECEIVE LINK FROM MASTER
            MOV     R6,A
OUTISR:     CLR     RI
            POP     PSW
            POP     ACC
            RETI
SEND:       CLR     TI                  ;SEND THE CURRENT STATUS TO MASTER
            MOV     R2,#4H
            CJNE    A,#22H,S2
TX1:        MOV     A,#01H
            CALL    SBYTE
            MOV     A,#31H
            CALL    SBYTE
            DJNZ    R2,TX1
            LJMP    OUTSEND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
S2:     CJNE    A,#88H,S3
TX2:    MOV     A,#01H
        CALL    SBYTE
        MOV     A,#32H
        CALL    SBYTE
        DJNZ    R2,TX2
        SJMP    OUTSEND
S3:     CJNE    A,#11H,S4
TX3:    MOV     A,#01H
        CALL    SBYTE
        MOV     A,#33H
        CALL    SBYTE
        DJNZ    R2,TX3
        SJMP    OUTSEND
S4:     CJNE    A,#44H,S5
TX4:    MOV     A,#01H
        CALL    SBYTE
        MOV     A,#34H
        CALL    SBYTE
        DJNZ    R2,TX4
        SJMP    OUTSEND
S5:     CJNE    A,#03H,S6
TX5:    MOV     A,#01H
        CALL    SBYTE
        MOV     A,#35H
        CALL    SBYTE
        DJNZ    R2,TX5
        SJMP    OUTSEND
S6:     CJNE    A,#30H,S7
TX6:    MOV     A,#01H
        CALL    SBYTE
        MOV     A,#36H
        CALL    SBYTE
        DJNZ    R2,TX6
        SJMP    OUTSEND
S7:     CJNE    A,#0C0H,S8
TX7:    MOV     A,#01H
        CALL    SBYTE
        MOV     A,#37H
        CALL    SBYTE
        DJNZ    R2,TX7
        SJMP    OUTSEND
S8:     CJNE    A,#0CH,OUTSEND
TX8:    MOV     A,#01H
        CALL    SBYTE
        MOV     A,#38H
        CALL    SBYTE
        DJNZ    R2,TX8
        SJMP    OUTSEND

SBYTE:  SETB    P1.0
        MOV     SBUF,A
        JNB     TI,$
        CLR     TI
        CLR     P1.0
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OUTSEND:    POP     PSW
            POP     ACC
            RETI

;*****************************************************
;*****************************************************

MAIN:       MOV     A,#9BH              ;INITIAL 8255 TO BE INPUT PORT
            MOV     DPTR,#USP1P
            MOVX    @DPTR,A
            MOV     A,#80H              ;INITIAL 8255 TO BE OUTPUT PORT
            MOV     DPTR,#USP2P
            MOVX    @DPTR,A
            LCALL   RTC_INIT

SW_ON:      MOV     DPTR,#USP1A         ;INPUT FROM SWITCH
            MOVX    A,@DPTR
SW1:        CJNE    A,#01H,SW2
            LCALL   PROG1
            LCALL   SW_OFF
            LCALL   YEL1
            LJMP    SW_ON

SW2:        CJNE    A,#02H,SW3
            LCALL   PROG2
            LCALL   SW_OFF
            LCALL   YEL2
            LJMP    SW_ON

SW3:        CJNE    A,#04H,SW4
            LCALL   PROG3
            LCALL   SW_OFF
            LCALL   YEL3
            LJMP    SW_ON

SW4:        CJNE    A,#08H,SW5
            LCALL   PROG4
            LCALL   SW_OFF
            LCALL   YEL4
            LJMP    SW_ON

SW5:        CJNE    A,#10H,SW6
            LCALL   PROG5
            LCALL   SW_OFF
            LCALL   YEL5
            LJMP    SW_ON

SW6:        CJNE    A,#20H,SW7
            LCALL   PROG6
            LCALL   SW_OFF
            LCALL   YEL6
            LJMP    SW_ON

SW7:        CJNE    A,#40H,SW8
            LCALL   PROG7
            LCALL   SW_OFF
            LCALL   YEL7
            LJMP    SW_ON
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SW8:        CJNE    A,#80H,YELLOW
            LCALL   PROG8
            LCALL   SW_OFF
            LCALL   YEL8
SW:         LJMP    SW_ON

YELLOW:     MOV     DPTR,#USP1B
            MOVX    A,@DPTR
            CJNE    A,#02H,AUTO
START:      MOV     A,#00H
            LCALL   OUT2A
            LCALL   OUT2C
            MOV     A,#0FFH
            LCALL   OUT2B
            MOV     R5,#1D
            LCALL   TIMER
            MOV     A,#00H
            LCALL   OUT2B
            MOV     R5,#1D
            LCALL   TIMER
            LCALL   AUTO_OFF
            SJMP    START

AUTO:       CJNE    A,#01H,SW
AUTO1:      LCALL   PROG1
            MOV     R5,#80D
            LCALL   TIMER
            LCALL   YEL1
            MOV     R5,#2D
            LCALL   TIMER
            LCALL   AUTO_OFF
            MOV     R6,#00H

AUTO2:      LCALL   PROG2
            MOV     R5,#80D
            LCALL   TIMER
            LCALL   YEL2
            MOV     R5,#2D
            LCALL   TIMER
            LCALL   AUTO_OFF
            CJNE    R6,#0BBH,AUTO3
            CALL    LINK

AUTO3:      LCALL   PROG3
            MOV     R5,#80D
            LCALL   TIMER
            LCALL   YEL3
            MOV     R5,#2D
            LCALL   TIMER
            LCALL   AUTO_OFF
            CJNE    R6,#0BBH,AUTO4
            CALL    LINK

AUTO4:      LCALL   PROG4
            MOV     R5,#80D
            LCALL   TIMER
            LCALL   YEL4
            MOV     R5,#2D
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCALL   TIMER
                LCALL   AUTO_OFF
                CJNE    R6,#0BBH,AUTO5
                CALL    LINK

AUTO5:          LCALL   PROG5
                MOV     R5,#80D
                LCALL   TIMER
                LCALL   YEL5
                MOV     R5,#2D
                LCALL   TIMER
                LCALL   AUTO_OFF
                CJNE    R6,#0BBH,AUTO6
                CALL    LINK

AUTO6:          LCALL   PROG6
                MOV     R5,#80D
                LCALL   TIMER
                LCALL   YEL6
                MOV     R5,#2D
                LCALL   TIMER
                LCALL   AUTO_OFF
                CJNE    R6,#0BBH,AUTO7
                CALL    LINK

AUTO7:          LCALL   PROG7
                MOV     R5,#80D
                LCALL   TIMER
                LCALL   YEL7
                MOV     R5,#2D
                LCALL   TIMER
                LCALL   AUTO_OFF
                CJNE    R6,#0BBH,AUTO8
                CALL    LINK

AUTO8:          LCALL   PROG8
                MOV     R5,#80D
                LCALL   TIMER
                LCALL   YEL8
                MOV     R5,#2D
                LCALL   TIMER
                LCALL   AUTO_OFF
                CJNE    R6,#0BBH,RUNAUTO
                CALL    LINK
RUNAUTO:        LJMP    AUTO1

LINK:           MOV     R6,#00H
                LCALL   PROG2
                MOV     R5,#80D
                LCALL   TIMER
                LCALL   YEL2
                MOV     R5,#2D
                LCALL   TIMER
                LCALL   AUTO_OFF
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PROG1:  MOV     A,#0DDH
        LCALL   OUT2A
        MOV     A,#00H
        LCALL   OUT2B
        MOV     A,#22H
        LCALL   OUT2C
        SETB    TI
        RET

PROG2:  MOV     A,#77H
        LCALL   OUT2A
        MOV     A,#00H
        LCALL   OUT2B
        MOV     A,#88H
        LCALL   OUT2C
        SETB    TI
        RET

PROG3:  MOV     A,#0EEH
        LCALL   OUT2A
        MOV     A,#00H
        LCALL   OUT2B
        MOV     A,#11H
        LCALL   OUT2C
        SETB    TI
        RET

PROG4:  MOV     A,#0BBH
        LCALL   OUT2A
        MOV     A,#00H
        LCALL   OUT2B
        MOV     A,#44H
        LCALL   OUT2C
        SETB    TI
        RET

PROG5:  MOV     A,#0FCH
        LCALL   OUT2A
        MOV     A,#00H
        LCALL   OUT2B
        MOV     A,#03H
        LCALL   OUT2C
        SETB    TI
        RET

PROG6:  MOV     A,#0CFH
        LCALL   OUT2A
        MOV     A,#00H
        LCALL   OUT2B
        MOV     A,#30H
        LCALL   OUT2C
        SETB    TI
        RET

PROG7:  MOV     A,#3FH
        LCALL   OUT2A
        MOV     A,#00H
        LCALL   OUT2B
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     A,#0C0H
                LCALL   OUT2C
                SETB    TI
                RET

PROG8:          MOV     A,#0F3H
                LCALL   OUT2A
                MOV     A,#00H
                LCALL   OUT2B
                MOV     A,#0CH
                LCALL   OUT2C
                SETB    TI
                RET
SW_OFF:         MOV     DPTR,#USP1A
                MOVX    A,@DPTR
                CJNE    A,#00H,SW_OFF
                RET

YEL1:           MOV     A,#22H
                LCALL   OUT2B
                MOV     A,#00H
                LCALL   OUT2C
                RET
YEL2:           MOV     A,#88H
                LCALL   OUT2B
                MOV     A,#00H
                LCALL   OUT2C
                RET
YEL3:           MOV     A,#11H
                LCALL   OUT2B
                MOV     A,#00H
                LCALL   OUT2C
                RET
YEL4:           MOV     A,#44H
                LCALL   OUT2B
                MOV     A,#00H
                LCALL   OUT2C
                RET
YEL5:           MOV     A,#03H
                LCALL   OUT2B
                MOV     A,#00H
                LCALL   OUT2C
                RET
YEL6:           MOV     A,#30H
                LCALL   OUT2B
                MOV     A,#00H
                LCALL   OUT2C
                RET
YEL7:           MOV     A,#0C0H
                LCALL   OUT2B
                MOV     A,#00H
                LCALL   OUT2C
                RET
YEL8:           MOV     A,#0CH
                LCALL   OUT2B
                MOV     A,#00H
                LCALL   OUT2C
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OUT2A:      MOV     DPTR,#USP2A
            MOVX    @DPTR,A
            RET
OUT2B:      MOV     DPTR,#USP2B
            MOVX    @DPTR,A
            RET
OUT2C:      MOV     DPTR,#USP2C
            MOVX    @DPTR,A
            RET

RTC_INIT:   CLR     RST
            SETB    CLK
            MOV     R2,#8EH
            MOV     R3,#00H
            LCALL   WRT
            MOV     R2,#80H
            MOV     R3,#00H
            LCALL   WRT
            MOV     R2,#82H
            MOV     R3,#00H
            LCALL   WRT
            MOV     R2,#84H
            MOV     R3,#00H
            LCALL   WRT
            MOV     R2,#8EH
            MOV     R3,#80H
            LCALL   WRT
            RET
WRT:        CLR     CLK
            NOP
            SETB    RST
            NOP
            MOV     A,R2
            LCALL   RTCWR8
            MOV     A,R3
            LCALL   RTCWR8
            CLR     RST
            NOP
            RET

TIMER:      CALL    RTC
            CALL    COMP
BO1:        CALL    RTC
            ANL     A,#01H
            CJNE    A,30H,EXIT
            SJMP    BO1
COMP:       ANL     A,#01H
            MOV     30H,A
            RET
EXIT:       MOV     DPTR,#USP1B
            MOVX    A,@DPTR
            CJNE    A,#00H,BO2
            MOV     R5,#1D
BO2:        DEC     R5
            CJNE    R5,#0H,TIMER
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RTC:        MOV     R2,#81H
            CLR     CLK
            NOP
            SETB    RST
            NOP
            MOV     A,R2
            CALL    RTCWR8
            MOV     R4,#08H
            CLR     A
RTCRD:      CLR     CLK                 ;READ FROM RTC
            NOP
            MOV     C,IO
            RRC     A
            SETB    CLK
            NOP
            DJNZ    R4,RTCRD
            CLR     RST
            NOP
            RET
RTCWR8:     MOV     R4,#08H             ;WRITE TO RTC
WR8:        RRC     A
            MOV     IO,C
            SETB    CLK
            NOP
            CLR     CLK
            NOP
            DJNZ    R4,WR8
            RET

AUTO_OFF:   MOV     DPTR,#USP1B
            MOVX    A,@DPTR
            CJNE    A,#00H,BACK
            LJMP    SW_ON
BACK:       RET

            END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include<conio.h>
#include<graphics.h>
#include<stdio.h>
#include<stdlib.h>
#include<dos.h>

#define baud 0xff;

void cross_road_no1(int);
void cross_road_no2(int);
void cross_road_no3(int);

void arrowp1c1no1(void);
void arrowp1c1no2(void);
void arrowp1c2no1(void);
void arrowp1c2no2(void);
void arrowp1c3no1(void);
void arrowp1c3no2(void);
void arrowp1c4no1(void);
void arrowp1c4no2(void);
void arrowp1c5no1(void);
void arrowp1c5no2(void);
void arrowp1c6no1(void);
void arrowp1c6no2(void);
void arrowp1c7no1(void);
void arrowp1c7no2(void);
void arrowp1c8no1(void);
void arrowp1c8no2(void);
void arrowp2c1no1(void);
void arrowp2c1no2(void);
void arrowp2c2no1(void);
void arrowp2c2no2(void);
void arrowp2c3no1(void);
void arrowp2c3no2(void);
void arrowp2c4no1(void);
void arrowp2c4no2(void);
void arrowp2c5no1(void);
void arrowp2c5no2(void);
void arrowp2c6no1(void);
void arrowp2c6no2(void);
void arrowp2c7no1(void);
void arrowp2c7no2(void);
void arrowp2c8no1(void);
void arrowp2c8no2(void);
void arrowp3c1no1(void);
void arrowp3c1no2(void);
void arrowp3c2no1(void);
void arrowp3c2no2(void);
void arrowp3c3no1(void);
void arrowp3c3no2(void);
void arrowp3c4no1(void);
void arrowp3c4no2(void);
void arrowp3c5no1(void);
void arrowp3c5no2(void);
void arrowp3c6no1(void);
void arrowp3c6no2(void);
void arrowp3c7no1(void);
void arrowp3c7no2(void);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void arrowp3c8no1(void);
void arrowp3c8no2(void);

void cross_road_no1_case1();
void cross_road_no1_case1();
void cross_road_no1_case1();
void cross_road_no1_case1();
void cross_road_no1_case1();
void cross_road_no1_case1();
void cross_road_no1_case1();
void cross_road_no1_case1();
void cross_road_no1_case1();
void cross_road_no1_case1();
void cross_road_no1_case1();
void cross_road_no1_case1();
void cross_road_no1_case1();
void cross_road_no1_case1();
void cross_road_no1_case1();
void cross_road_no1_case2();
void cross_road_no1_case3();
void cross_road_no1_case4();
void cross_road_no1_case5();
void cross_road_no1_case6();
void cross_road_no1_case7();
void cross_road_no1_case8();
void cross_road_no2_case1();
void cross_road_no2_case2();
void cross_road_no2_case3();
void cross_road_no2_case4();
void cross_road_no2_case5();
void cross_road_no2_case6();
void cross_road_no2_case7();
void cross_road_no2_case8();
void cross_road_no3_case1();
void cross_road_no3_case2();
void cross_road_no3_case3();
void cross_road_no3_case4();
void cross_road_no3_case5();
void cross_road_no3_case6();
void cross_road_no3_case7();
void cross_road_no3_case8();

void draw_cross1();
void draw_cross2();
void draw_cross3();
void clear_cross1();
void clear_cross2();
void clear_cross3();

void initial_port();
check_port();
void clear_port();
receive();
void check_case();
void check_type();
void check_address();

void draw_cross1()                    /*draw cross road no.1*/
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    int x,y;
    setcolor(2);
    line(50,90,50,140);
    line(50,200,50,250);
    line(115,90,115,140);
    line(115,200,115,250);
    line(0,140,50,140);
    line(0,200,50,200);
    line(115,140,165,140);
    line(115,200,165,200);

    for(x=11;x<=11;++x)
    {
      for(y=6;y<=9;++y)
      {
      gotoxy(x,y);
      printf(".");
      }
    }

    for(x=11;x<=11;++x)
    {
      for(y=13;y<=16;++y)
      {
      gotoxy(x,y);
      printf(".");
      }
    }

    for(y=11;y<=11;++y)
    {
      for(x=2;x<=7;++x)
      {
      gotoxy(x,y);
      printf(".");
      }
    }

    for(y=11;y<=11;++y)
    {
      for(x=15;x<=20;++x)
      {
      gotoxy(x,y);
      printf(".");
      }
    }

    x=4;
    y=18;
    gotoxy(x,y);
    printf("CROSS ROAD NO. 1");

}
void cross_road_no1(int data)
{
    switch(data)                    /*select case of data from port
com1*/
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case 161 : clear_cross1();
                   draw_cross1();
                   cross_road_no1_case1();
                   break;
        case 162 : clear_cross1();
                   draw_cross1();
                   cross_road_no1_case2();
                   break;
        case 163 : clear_cross1();
                   draw_cross1();
                   cross_road_no1_case3();
                   break;
        case 164 : clear_cross1();
                   draw_cross1();
                   cross_road_no1_case4();
                   break;
        case 165 : clear_cross1();
                   draw_cross1();
                   cross_road_no1_case5();
                   break;
        case 166 : clear_cross1();
                   draw_cross1();
                   cross_road_no1_case6();
                   break;
        case 167 : clear_cross1();
                   draw_cross1();
                   cross_road_no1_case7();
                   break;
        case 168 : clear_cross1();
                   draw_cross1();
                   cross_road_no1_case8();
                   break;
      }
}

void cross_road_no1_case1()
{
     setcolor(4);
     line(98,110,98,140);
     arrowp1c1no1();
     line(66,200,66,230);
     arrowp1c1no2();
}

void cross_road_no1_case2()
{
     setcolor(4);
     line(20,155,50,155);
     arrowp1c2no1();
     line(115,185,145,185);
     arrowp1c2no2();

}

void cross_road_no1_case3()
{
      setcolor(4);
     line(93,110,93,140);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
     arc(48,138,270,360,45);
     line(50,184,30,184);
     arrowp1c3no1();
     line(72,230,72,200);
     arc(117,202,90,180,45);
     line(115,157,135,157);
     arrowp1c3no2();
}

void cross_road_no1_case4()
{
     setcolor(4);
     line(20,157,50,157);
     arc(48,202,0,90,45);
     line(93,200,93,220);
     arrowp1c4no1();
     line(145,183,115,183);
     arc(117,138,180,270,45);
     line(72,140,72,120);
     arrowp1c4no2();
}

void cross_road_no1_case5()
{
     setcolor(4);
     line(98,110,98,220);
     arrowp1c5no1();
     arc(50,140,270,360,48);
     line(50,188,30,188);
     arrowp1c5no2();
}

void cross_road_no1_case6()
{
     setcolor(4);
     line(66,230,66,120);
     arrowp1c6no1();
     arc(115,200,90,180,48);
     line(115,152,135,152);
     arrowp1c6no2();
}

void cross_road_no1_case7()
{
     setcolor(4);
     line(20,155,135,155);
     arrowp1c7no1();
     arc(48,202,0,90,47);
     line(96,200,96,220);
     arrowp1c7no2();
}

void cross_road_no1_case8()
{
     setcolor(4);
     line(145,185,30,185);
     arrowp1c8no1();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    arc(117,138,180,270,47);
    line(70,140,70,120);
    arrowp1c8no2();
}

void arrowp1c1no1()
{
    int poly[6];
      poly[0]=98;
      poly[1]=140;
      poly[2]=95;
      poly[3]=137;
      poly[4]=101;
      poly[5]=137;
    fillpoly(3,poly);
    return;
}

void arrowp1c1no2()
{
    int poly[6];
      poly[0]=66;
      poly[1]=200;
      poly[2]=63;
      poly[3]=203;
      poly[4]=69;
      poly[5]=203;

    fillpoly(3,poly);
    return;
}

void arrowp1c2no1()
{
    int poly[6];
      poly[0]=50;
      poly[1]=155;
      poly[2]=47;
      poly[3]=152;
      poly[4]=47;
      poly[5]=158;
    fillpoly(3,poly);
    return;
}

void arrowp1c2no2()
{
    int poly[6];
      poly[0]=115;
      poly[1]=185;
      poly[2]=118;
      poly[3]=182;
      poly[4]=118;
      poly[5]=188;
    fillpoly(3,poly);
    return;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void arrowp1c3no1()
{
     int poly[6];
       poly[0]=30;
       poly[1]=184;
       poly[2]=33;
       poly[3]=187;
       poly[4]=33;
       poly[5]=181;
     fillpoly(3,poly);
     return;
}

void arrowp1c3no2()
{
     int poly[6];
       poly[0]=135;
       poly[1]=157;
       poly[2]=132;
       poly[3]=154;
       poly[4]=132;
       poly[5]=160;
     fillpoly(3,poly);
     return;
}

void arrowp1c4no1()
{
          int poly[6];
       poly[0]=93;
       poly[1]=220;
       poly[2]=90;
       poly[3]=218;
       poly[4]=95;
       poly[5]=218;
     fillpoly(3,poly);
     return;

}

void arrowp1c4no2()
{
     int poly[6];
       poly[0]=72;
       poly[1]=120;
       poly[2]=69;
       poly[3]=123;
       poly[4]=75;
       poly[5]=123;
     fillpoly(3,poly);
     return;
}

void arrowp1c5no1()
{
     int poly[6];
       poly[0]=98;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        poly[1]=220;
        poly[2]=95;
        poly[3]=217;
        poly[4]=101;
        poly[5]=217;
      fillpoly(3,poly);
      return;
}

void arrowp1c5no2()
{
      int poly[6];
        poly[0]=30;
        poly[1]=188;
        poly[2]=33;
        poly[3]=185;
        poly[4]=33;
        poly[5]=191;
      fillpoly(3,poly);
      return;
}

void arrowp1c6no1()
{
      int poly[6];
        poly[0]=66;
        poly[1]=120;
        poly[2]=63;
        poly[3]=123;
        poly[4]=69;
        poly[5]=123;
      fillpoly(3,poly);
      return;
}

void arrowp1c6no2()
{
      int poly[6];
        poly[0]=135;
        poly[1]=152;
        poly[2]=132;
        poly[3]=149;
        poly[4]=132;
        poly[5]=155;
      fillpoly(3,poly);
      return;
}

void arrowp1c7no1()
{
      int poly[6];
        poly[0]=135;
        poly[1]=155;
        poly[2]=132;
        poly[3]=158;
        poly[4]=132;
        poly[5]=152;
      fillpoly(3,poly);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return;
}

void arrowp1c7no2()
{
    int poly[6];
      poly[0]=96;
      poly[1]=220;
      poly[2]=99;
      poly[3]=217;
      poly[4]=93;
      poly[5]=217;
    fillpoly(3,poly);
    return;
}

void arrowp1c8no1()
{
    int poly[6];
      poly[0]=30;
      poly[1]=185;
      poly[2]=33;
      poly[3]=188;
      poly[4]=33;
      poly[5]=182;
    fillpoly(3,poly);
    return;
}

void arrowp1c8no2()
{
    int poly[6];
      poly[0]=70;
      poly[1]=120;
      poly[2]=67;
      poly[3]=123;
      poly[4]=73;
      poly[5]=123;
    fillpoly(3,poly);
    return;
}


void draw_cross2()                  /*draw cross road no.2*/
{
    int x,y;
    setcolor(2);
    line(282,90,282,140);
    line(282,200,282,250);
    line(347,90,347,140);
    line(347,200,347,250);
    line(230,140,282,140);
    line(230,200,282,200);
    line(347,140,397,140);
    line(347,200,397,200);

    for(x=40;x<=40;++x)
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      for(y=6;y<=9;++y)
      {
      gotoxy(x,y);
      printf(".");
      }
    }

    for(x=40;x<=40;++x)
    {
      for(y=13;y<=16;++y)
      {
      gotoxy(x,y);
      printf(".");
      }
    }

    for(y=11;y<=11;++y)
    {
      for(x=30;x<=35;++x)
      {
      gotoxy(x,y);
      printf(".");
      }
    }

    for(y=11;y<=11;++y)
    {
      for(x=44;x<=49;++x)
      {
      gotoxy(x,y);
      printf(".");
      }
    }
    x=33;
    y=18;
    gotoxy(x,y);
    printf("CROSS ROAD NO. 2");

}

void cross_road_no2(int data)
{
    switch(data)                    /*select case of data from port com1*/
    {
      case 33 :  clear_cross2();
                 draw_cross2();
                 cross_road_no2_case1();
                 break;
      case 34 :  clear_cross2();
                 draw_cross2();
                 cross_road_no2_case2();
                 break;
      case 35 :  clear_cross2();
                 draw_cross2();
                 cross_road_no2_case3();
                 break;
      case 36 :  clear_cross2();
                 draw_cross2();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                cross_road_no2_case4();
                break;
        case 37 : clear_cross2();
                draw_cross2();
                cross_road_no2_case5();
                break;
        case 38 : clear_cross2();
                draw_cross2();
                cross_road_no2_case6();
                break;
        case 39 : clear_cross2();
                draw_cross2();
                cross_road_no2_case7();
                break;
        case 40 : clear_cross2();
                draw_cross2();
                cross_road_no2_case8();
                break;
    }
}

void cross_road_no2_case1()
{
    setcolor(4);
    line(330,110,330,140);
    arrowp2c1no1();
    line(298,200,298,230);
    arrowp2c1no2();
}

void cross_road_no2_case2()
{
   setcolor(4);
    line(252,155,282,155);
    arrowp2c2no1();
    line(347,185,377,185);
    arrowp2c2no2();
}

void cross_road_no2_case3()
{
    setcolor(4);
    line(325,110,325,140);
    arc(280,138,270,360,45);
    line(282,184,262,184);
    arrowp2c3no1();
    line(304,230,304,200);
    arc(349,202,90,180,45);
    line(347,157,367,157);
    arrowp2c3no2();
}

void cross_road_no2_case4()
{
    setcolor(4);
    line(252,157,282,157);
    arc(280,202,0,90,45);
    line(325,200,325,220);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    arrowp2c4no1();
    line(347,183,377,183);
    arc(349,138,180,270,45);
    line(304,140,304,120);
    arrowp2c4no2();
}

void cross_road_no2_case5()
{
    setcolor(4);
    line(330,220,330,110);
    arrowp2c5no1();
    arc(282,140,270,360,48);
    line(282,188,262,188);
    arrowp2c5no2();
}

void cross_road_no2_case6()
{
    setcolor(4);
    line(298,230,298,120);
    arrowp2c6no1();
    arc(347,200,90,180,48);
    line(347,152,367,152);
    arrowp2c6no2();
}

void cross_road_no2_case7()
{
   setcolor(4);
    line(252,155,367,155);
    arrowp2c7no1();
    arc(280,202,0,90,47);
    line(328,200,328,220);
    arrowp2c7no2();
}

void cross_road_no2_case8()
{
    setcolor(4);
    line(377,185,262,185);
    arrowp2c8no1();
    arc(349,138,180,270,47);
    line(302,140,302,120);
    arrowp2c8no2();
}

void arrowp2c1no1()
{
    int poly[6];
      poly[0]=330;
      poly[1]=140;
      poly[2]=327;
      poly[3]=137;
      poly[4]=333;
      poly[5]=137;
    fillpoly(3,poly);
    return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

void arrowp2c1no2()
{
     int poly[6];
       poly[0]=298;
       poly[1]=200;
       poly[2]=295;
       poly[3]=203;
       poly[4]=301;
       poly[5]=203;
     fillpoly(3,poly);
     return;
}

void arrowp2c2no1()
{
     int poly[6];
       poly[0]=282;
       poly[1]=155;
       poly[2]=279;
       poly[3]=152;
       poly[4]=279;
       poly[5]=158;
     fillpoly(3,poly);
     return;
}

void arrowp2c2no2()
{
     int poly[6];
       poly[0]=347;
       poly[1]=185;
       poly[2]=350;
       poly[3]=182;
       poly[4]=350;
       poly[5]=188;
     fillpoly(3,poly);
     return;
}

void arrowp2c3no1()
{
     int poly[6];
       poly[0]=262;
       poly[1]=184;
       poly[2]=265;
       poly[3]=181;
       poly[4]=265;
       poly[5]=187;
     fillpoly(3,poly);
     return;
}

void arrowp2c3no2()
{
     int poly[6];
       poly[0]=367;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      poly[1]=157;
      poly[2]=364;
      poly[3]=154;
      poly[4]=364;
      poly[5]=160;
    fillpoly(3,poly);
    return;
}

void arrowp2c4no1()
{
    int poly[6];
      poly[0]=325;
      poly[1]=220;
      poly[2]=322;
      poly[3]=217;
      poly[4]=328;
      poly[5]=217;
    fillpoly(3,poly);
    return;
}

void arrowp2c4no2()
{
    int poly[6];
      poly[0]=304;
      poly[1]=120;
      poly[2]=301;
      poly[3]=123;
      poly[4]=307;
      poly[5]=123;
    fillpoly(3,poly);
    return;
}

void arrowp2c5no1()
{
    int poly[6];
      poly[0]=330;
      poly[1]=220;
      poly[2]=327;
      poly[3]=217;
      poly[4]=333;
      poly[5]=217;
    fillpoly(3,poly);
    return;
}

void arrowp2c5no2()
{
    int poly[6];
      poly[0]=262;
      poly[1]=188;
      poly[2]=265;
      poly[3]=189;
      poly[4]=265;
      poly[5]=183;
    fillpoly(3,poly);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return;
}

void arrowp2c6no1()
{
    int poly[6];
      poly[0]=298;
      poly[1]=120;
      poly[2]=301;
      poly[3]=123;
      poly[4]=295;
      poly[5]=123;
    fillpoly(3,poly);
    return;
}

void arrowp2c6no2()
{   int poly[6];
      poly[0]=367;
      poly[1]=152;
      poly[2]=364;
      poly[3]=155;
      poly[4]=364;
      poly[5]=149;
    fillpoly(3,poly);
    return;

}

void arrowp2c7no1()
{
    int poly[6];
      poly[0]=367;
      poly[1]=155;
      poly[2]=364;
      poly[3]=152;
      poly[4]=364;
      poly[5]=158;
    fillpoly(3,poly);
    return;
}

void arrowp2c7no2()
{
    int poly[6];
      poly[0]=328;
      poly[1]=220;
      poly[2]=325;
      poly[3]=217;
      poly[4]=331;
      poly[5]=217;
    fillpoly(3,poly);
    return;
}

void arrowp2c8no1()
{
    int poly[6];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      poly[0]=262;
      poly[1]=185;
      poly[2]=265;
      poly[3]=188;
      poly[4]=265;
      poly[5]=182;
    fillpoly(3,poly);
    return;
}

void arrowp2c8no2()
{
    int poly[6];
      poly[0]=302;
      poly[1]=120;
      poly[2]=299;
      poly[3]=123;
      poly[4]=305;
      poly[5]=123;
    fillpoly(3,poly);
    return;
}

void draw_cross3()              /*draw cross road no.3*/
{
    int x,y;
    setcolor(2);
    line(515,90,515,140);
    line(515,200,515,250);
    line(580,90,580,140);
    line(580,200,580,250);
    line(465,140,515,140);
    line(465,200,515,200);
    line(580,140,630,140);
    line(580,200,630,200);

    for(x=69;x<=69;++x)
    {
      for(y=6;y<=9;++y)
      {
      gotoxy(x,y);
      printf(".");
      }
    }

    for(x=69;x<=69;++x)
    {
      for(y=13;y<=16;++y)
      {
      gotoxy(x,y);
      printf(".");
      }
    }

    for(y=11;y<=11;++y)
    {
      for(x=60;x<=65;++x)
      {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        gotoxy(x,y);
        printf(".");
        }
    }

    for(y=11;y<=11;++y)
    {
      for(x=73;x<=78;++x)
      {
      gotoxy(x,y);
      printf(".");
      }
    }
    x=62;
    y=18;
    gotoxy(x,y);
    printf("CROSS ROAD NO. 3");

}

void cross_road_no3(int data)
{
    switch(data)                    /*select case of data from port com1*/
    {
      case 49 :  clear_cross3();
                 draw_cross3();
                 cross_road_no3_case1();
                 break;
      case 50 :  clear_cross3();
                 draw_cross3();
                 cross_road_no3_case2();
                 break;
      case 51 :  clear_cross3();
                 draw_cross3();
                 cross_road_no3_case3();
                 break;
      case 52 :  clear_cross3();
                 draw_cross3();
                 cross_road_no3_case4();
                 break;
      case 53 :  clear_cross3();
                 draw_cross3();
                 cross_road_no3_case5();
                 break;
      case 54 :  clear_cross3();
                 draw_cross3();
                 cross_road_no3_case6();
                 break;
      case 55 :  clear_cross3();
                 draw_cross3();
                 cross_road_no3_case7();
                 break;
      case 56 :  clear_cross3();
                 draw_cross3();
                 cross_road_no3_case8();
                 break;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

void cross_road_no3_case1()
{
     setcolor(4);
     line(563,110,563,140);
     arrowp3c1no1();
     line(531,200,531,230);
     arrowp3c1no2();
}

void cross_road_no3_case2()
{
     setcolor(4);
     line(485,155,515,155);
     arrowp3c2no1();
     line(580,185,610,185);
     arrowp3c2no2();
}

void cross_road_no3_case3()
{   setcolor(4);
     line(558,110,558,140);
     arc(513,138,270,360,45);
     line(515,184,495,184);
     arrowp3c3no1();
     line(537,230,537,200);
     arc(582,202,90,180,45);
     line(580,157,600,157);
     arrowp3c3no2();
}

void cross_road_no3_case4()
{
     setcolor(4);
     line(485,157,515,157);
     arc(513,202,0,90,45);
     line(558,200,558,220);
     arrowp3c4no1();
     line(610,183,580,183);
     arc(582,138,180,270,45);
     line(537,140,537,120);
     arrowp3c4no2();
}

void cross_road_no3_case5()
{
     setcolor(4);
     line(563,110,563,220);
     arrowp3c5no1();
     arc(515,140,270,360,48);
     line(515,188,495,188);
     arrowp3c5no2();
}

void cross_road_no3_case6()
{
     setcolor(4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
     line(531,230,531,120);
     arrowp3c6no1();
     arc(580,200,90,180,48);
     line(580,152,600,152);
     arrowp3c6no2();
}

void cross_road_no3_case7()
{
     setcolor(4);
     line(485,155,600,155);
     arrowp3c7no1();
     arc(513,202,0,90,47);
     line(561,200,561,220);
     arrowp3c7no2();
}

void cross_road_no3_case8()
{
     setcolor(4);
     line(610,185,495,185);
     arrowp3c8no1();
     arc(582,138,180,270,47);
     line(535,140,535,120);
     arrowp3c8no2();
}

void arrowp3c1no1()
{
     int poly[6];
       poly[0]=563;
       poly[1]=140;
       poly[2]=560;
       poly[3]=137;
       poly[4]=566;
       poly[5]=137;
     fillpoly(3,poly);
     return;
}

void arrowp3c1no2()
{
     int poly[6];
       poly[0]=531;
       poly[1]=200;
       poly[2]=529;
       poly[3]=203;
       poly[4]=534;
       poly[5]=203;
     fillpoly(3,poly);
     return;

}

void arrowp3c2no1()
{
     int poly[6];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      poly[0]=515;
      poly[1]=155;
      poly[2]=512;
      poly[3]=152;
      poly[4]=512;
      poly[5]=158;
    fillpoly(3,poly);
    return;
}

void arrowp3c2no2()
{
    int poly[6];
      poly[0]=580;
      poly[1]=185;
      poly[2]=583;
      poly[3]=182;
      poly[4]=583;
      poly[5]=188;
    fillpoly(3,poly);
    return;
}

void arrowp3c3no1()
{
    int poly[6];
      poly[0]=495;
      poly[1]=184;
      poly[2]=498;
      poly[3]=181;
      poly[4]=498;
      poly[5]=187;
    fillpoly(3,poly);
    return;
}

void arrowp3c3no2()
{
    int poly[6];
      poly[0]=600;
      poly[1]=157;
      poly[2]=597;
      poly[3]=154;
      poly[4]=597;
      poly[5]=160;
    fillpoly(3,poly);
    return;
}

void arrowp3c4no1()
{
    int poly[6];
      poly[0]=558;
      poly[1]=220;
      poly[2]=555;
      poly[3]=217;
      poly[4]=561;
      poly[5]=217;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    fillpoly(3,poly);
    return;
}

void arrowp3c4no2()
{
    int poly[6];
      poly[0]=537;
      poly[1]=120;
      poly[2]=534;
      poly[3]=123;
      poly[4]=540;
      poly[5]=123;
    fillpoly(3,poly);
    return;
}

void arrowp3c5no1()
{
    int poly[6];
      poly[0]=563;
      poly[1]=220;
      poly[2]=560;
      poly[3]=217;
      poly[4]=566;
      poly[5]=217;
    fillpoly(3,poly);
    return;
}

void arrowp3c5no2()
{
    int poly[6];
      poly[0]=495;
      poly[1]=188;
      poly[2]=498;
      poly[3]=185;
      poly[4]=498;
      poly[5]=191;
    fillpoly(3,poly);
    return;
}

void arrowp3c6no1()
{
    int poly[6];
      poly[0]=531;
      poly[1]=120;
      poly[2]=528;
      poly[3]=123;
      poly[4]=534;
      poly[5]=123;
    fillpoly(3,poly);
    return;
}

void arrowp3c6no2()
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    int poly[6];
      poly[0]=600;
      poly[1]=152;
      poly[2]=597;
      poly[3]=149;
      poly[4]=597;
      poly[5]=155;
    fillpoly(3,poly);
    return;
}

void arrowp3c7no1()
{
    int poly[6];
      poly[0]=600;
      poly[1]=155;
      poly[2]=598;
      poly[3]=152;
      poly[4]=598;
      poly[5]=158;
    fillpoly(3,poly);
    return;
}

void arrowp3c7no2()
{
    int poly[8];
      poly[0]=405;
      poly[1]=121;
      poly[2]=403;
      poly[3]=119;
      poly[4]=407;
      poly[5]=119;
    fillpoly(3,poly);
    return;
}

void arrowp3c8no1()
{   int poly[6];
      poly[0]=495;
      poly[1]=185;
      poly[2]=498;
      poly[3]=182;
      poly[4]=498;
      poly[5]=188;
    fillpoly(3,poly);
    return;

}

void arrowp3c8no2()
{
    int poly[6];
      poly[0]=535;
      poly[1]=120;
      poly[2]=532;
      poly[3]=123;
      poly[4]=538;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      poly[5]=123;
    fillpoly(3,poly);
    return;
}


void main()
{
    int gdriver=DETECT,gmode,x,y;
    int c;
    initgraph(&gdriver,&gmode,"");  /*initial to be graphic mode*/
    setviewport(0,0,639,479,1);
    textbackground(1);
    setcolor(2);
    draw_cross1();
    draw_cross2();
    draw_cross3();

    initial_port();
    clear_port();
    check_port();

    for(;;)
    {
      check_address();
    }
}

void initial_port()
{
    union REGS r;
    r.x.dx=0;                   /*port com1*/
    r.h.ah=0;                   /*ah=0 =>initial port*/
    r.h.al=baud;                /* baud=FFH*/
    int86(0x14,&r,&r);
}

void clear_port()
{
    union REGS r;
    r.x.dx=0;
    r.h.ah=2;
    int86(0x14,&r,&r);
}

receive()
{
    union REGS r;
    r.x.dx=0;
    r.h.ah=2;
    int86(0x14,&r,&r);
    return r.h.al;
}

check_port()
{
    union REGS r;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    r.x.dx=0;
    r.h.ah=3;
    int86(0x14,&r,&r);
    return r.x.ax;
}

void check_address()
{
    int b;
    b=receive();
    switch(b)
    {
      case 0:check_type();
             break;

    }
}
void check_type()
{

    int ch;
    ch=receive();
    check_case(ch);
}

void check_case(int max)
{
    int num;
    switch(max)
    {
      case 17 : num=17;                 /*no.1,case 1*/
                cross_road_no1(num);
                break;
      case 18 : num=18;                 /*no.1,case 2*/
                cross_road_no1(num);
                break;
      case 19 : num=19;                 /*no.1,case 3*/
                cross_road_no1(num);
                break;
      case 20 : num=20;                 /*no.1,case 4*/
                cross_road_no1(num);
                break;
      case 21 : num=21;                 /*no.1,case 5*/
                cross_road_no1(num);
                break;
      case 22 : num=22;                 /*no.1,case 6*/
                cross_road_no1(num);
                break;
      case 23 : num=23;                 /*no.1,case 7*/
                cross_road_no1(num);
                break;
      case 24 : num=24;                 /*no.1,case 8*/
                cross_road_no1(num);
                break;

      case 33 : num=33;                 /*no.2,case 1*/
                cross_road_no2(num);
                break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      case 34 : num=34;                /*no.2,case 2*/
                cross_road_no2(num);
                break;
      case 35 : num=35;                /*no.2,case 3*/
                    cross_road_no2(num);
                    break;
      case 36 : num=36;                /*no.2,case 4*/
                cross_road_no2(num);
                break;
      case 37 : num=37;                /*no.2,case 5*/
                cross_road_no2(num);
                break;
      case 38 : num=38;                /*no.2,case 6*/
                cross_road_no2(num);
                break;
      case 39 : num=39;                /*no.2,case 7*/
                cross_road_no2(num);
                break;
      case 40 : num=40;                /*no.2,case 8*/
                cross_road_no2(num);
                break;

      case 49 : num=49;                /*no.3,case 1*/
                cross_road_no3(num);
                break;
      case 50 : num=50;                /*no.3,case 2*/
                cross_road_no3(num);
                break;
      case 51 : num=51;                /*no.3,case 3*/
                cross_road_no3(num);
                break;
      case 52 : num=52;                /*no.3,case 4*/
                cross_road_no3(num);
                break;
      case 53 : num=53;                /*no.3,case 5*/
                cross_road_no3(num);
                break;
      case 54 : num=54;                /*no.3,case 6*/
                cross_road_no3(num);
                break;
      case 55 : num=55;                /*no.3,case 7*/
                cross_road_no3(num);
                break;
      case 56 : num=56;                /*no.3,case 8*/
                cross_road_no3(num);
                break;
     }

}

void clear_cross1()
{
     int x,y;
     for(x=53;x<=112;x++)
     {
       for(y=90;y<=250;y++)
       {
           setcolor(0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            line(x,y,x,y++);
        }
    }
    for(x=0;x<=165;x++)
    {
      for(y=143;y<=197;y++)
      {
            setcolor(0);
            line(x,y,x,y++);
      }
    }

}
void clear_cross2()
{
    int x,y;
    for(x=285;x<=344;x++)
    {
      for(y=90;y<=250;y++)
      {
            setcolor(0);
            line(x,y,x,y++);
      }
    }
    for(x=230;x<=397;x++)
    {
      for(y=143;y<=197;y++)
      {
            setcolor(0);
            line(x,y,x,y++);
      }
    }

}
void clear_cross3()
{
    int x,y;
    for(x=518;x<=577;x++)
    {
      for(y=90;y<=250;y++)
      {
            setcolor(0);
            line(x,y,x,y++);
      }
    }
    for(x=465;x<=630;x++)
    {
      for(y=143;y<=197;y++)
      {
            setcolor(0);
            line(x,y,x,y++);
      }
    }

}
```

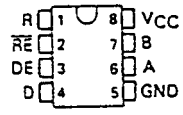
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SN75176A
DIFFERENTIAL BUS TRANSCEIVER

D2619, JUNE 1984—REVISED AUGUST 1989

- Bidirectional Transceiver
- Meets EIA Standards RS-422A and CCITT Recommendations V.11 and X.27
- Designed for Multipoint Transmission on Long Bus Lines in Noisy Environments
- 3-State Driver and Receiver Outputs
- Individual Driver and Receiver Enables
- Wide Positive and Negative Input/Output Bus Voltage Ranges
- Driver Output Capability. . . ±60 mA Max
- Thermal Shutdown Protection
- Driver Positive and Negative Current Limiting
- Receiver Input Impedance . . . 12 kΩ Min
- Receiver Input Sensitivity . . . ±200 mV
- Receiver Input Hysteresis . . . 50 mV Typ
- Operates from Single 5-V Supply
- Low Power Requirements

D OR P
DUAL-IN-LINE PACKAGE
(TOP VIEW)



FUNCTION TABLE (DRIVER)

INPUT	ENABLE	OUTPUTS	
D	DE	A	B
H	H	H	L
L	H	L	H
X	L	Z	Z

FUNCTION TABLE (RECEIVER)

DIFFERENTIAL INPUTS	ENABLE	OUTPUT
A − B	$\overline{RE}$	R
$V_{ID} > 0.2$ V	L	H
-0.2 V $< V_{ID} < 0.2$ V	L	?
$V_{ID} < -0.2$ V	L	L
X	H	Z

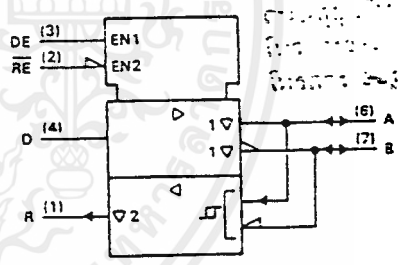
H = high level, L = low level, ? = indeterminate,
X = irrelevant, Z = high impedance (off)

description

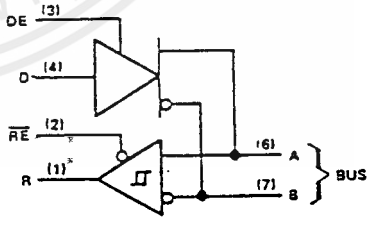
The SN75176A differential bus transceiver is a monolithic integrated circuit designed for bidirectional data communication on multipoint bus transmission lines. It is designed for balanced transmission lines and meets EIA Standard RS-422A and CCITT Recommendations V.11 and X.27.

The SN75176A combines a 3-state differential line driver and a differential-input line receiver both of which operate from a single 5-V power supply. The driver and receiver have active-high and active-low enables, respectively, that can be externally connected together to function as direction control. The driver differential outputs and the receiver differential inputs are connected internally to form differential input/output (I/O bus ports that are designed to offer minimum loading to the bus whenever the driver is disabled or V_{CC} = 0. These ports feature wide positive and negative common-mode voltage ranges making the device suitable for party-line applications.

logic symbol



logic diagram (positive logic)



PRODUCTION DATA documents contain information 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 © 1989, Texas Instruments Incorporated

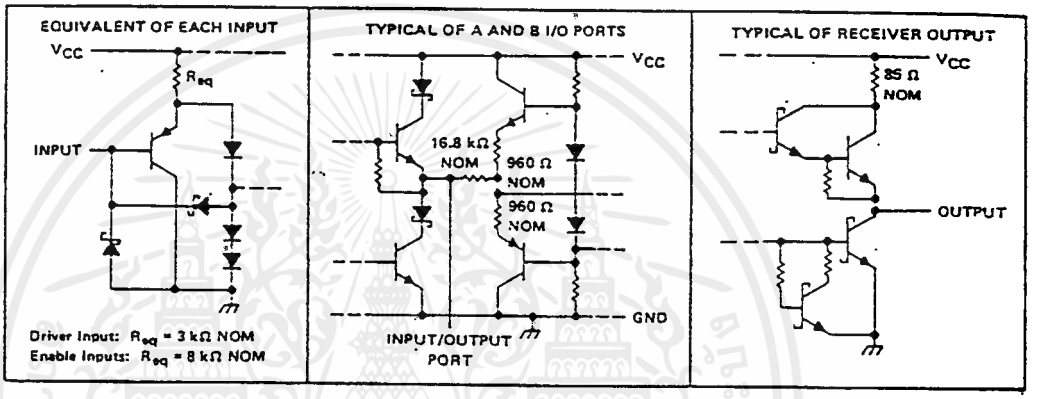
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

description (continued)

The driver is designed to handle loads up to 60 mA of sink or source current. The driver features positive- and negative-current limiting and thermal shutdown for protection from line fault conditions. Thermal shutdown is designed to occur at a junction temperature of approximately 150 °C. The receiver features a minimum input impedance of 12 kΩ, input sensitivity of ±200 mV, and a typical input hysteresis of 50 mV.

The SN75176A can be used in transmission line applications employing the SN75172 and SN75174 quadruple differential line drivers and the SN75173 and SN75175 quadruple differential line receivers.

schematics of inputs and outputs



TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

Supply voltage, V_{CC} (see Note 1) . . . . . 7 V
Voltage at any bus terminal . . . . . −10 V to 15 V
Enable input voltage . . . . . 5.5 V
Continuous total dissipation at (or below) 25°C free-air temperature (see Note 2):
D package . . . . . 725 mW
P package . . . . . 1000 mW
Operating free-air temperature range . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds . . . . . 260°C

NOTES: 1. All voltage values, except differential input/output bus voltage, are with respect to network ground terminal.
2. For operation above 25°C free-air temperature, derate the D package to 464 mW at 70°C at the rate of 5.8 mW/°C and derate the P package to 640 mW at 70°C at the rate of 8.0 mW/°C.

recommended operating conditions

		MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}		4.75	5	5.25	V
Voltage at any bus terminal (separately or common-mode), V_I or V_{IC}		−7		12	V
High-level input voltage, V_{IH}	D, DE, and $\overline{RE}$	2			V
Low-level input voltage, V_{IL}	D, DE, and $\overline{RE}$			0.8	V
Differential input voltage, V_{ID} (see Note 3)				±12	V
High-level output current, I_{OH}	Driver			−60	mA
	Receiver			−400	μA
Low-level output current, I_{OL}	Driver			60	mA
	Receiver			8	
Operating free-air temperature, T_A		0		70	°C

NOTE 3: Differential-input/output bus voltage is measured at the noninverting terminal A with respect to the inverting terminal B.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DRIVER SECTION

driver electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted).

PARAMETER		TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{IK}	Input clamp voltage	I_I = −18 mA				−1.5	V
V_{OH}	High-level output voltage	V_{IH} = 2 V, I_{OH} = −33 mA	V_{IL} = 0.8 V,		3.7		V
V_{OL}	Low-level output voltage	V_{IH} = 2 V, I_{OL} = 33 mA	V_{IL} = 0.8 V,		1.1		V
$\lvert V_{OD1}\rvert$‡	Differential output voltage	I_O = 0				2 V_{OD2}	V
$\lvert V_{OD2}\rvert$	Differential output voltage	R_L = 100 Ω,	See Figure 1	2	2.7		V
		R_L = 54 Ω,	See Figure 1	1.5	2.4		
$\Delta\lvert V_{OD}\rvert$	Change in magnitude of differential output voltage‡					±0.2	V
V_{OC}	Common-mode output voltage§	R_L = 54 Ω or 100 Ω,	See Figure 1			3	V
$\Delta\lvert V_{OC}\rvert$	Change in magnitude of common-mode output voltage‡					±0.2	V
I_O	Output current	Output disabled, See Note 4	V_O = 12 V			1	mA
			V_O = −7 V			−0.8	
I_{IH}	High-level input current	V_I = 2.4 V				20	μA
I_{IL}	Low-level input current	V_I = 0.4 V				−400	μA
I_{OS}	Short-circuit output current	V_O = −7 V				−250	mA
		V_O = V_{CC}				250	
		V_O = 12 V				500	
I_{CC}	Supply current (total package)	No load	Outputs enabled		35	50	mA
			Outputs disabled		26	40	

†All typical values are at V_{CC} = 5 V and T_A = 25°C.
‡$\Delta\lvert V_{OD}\rvert$ and $\Delta\lvert V_{OC}\rvert$ are the changes in magnitude of V_{OD} and V_{OC} respectively, that occur when the input is changed from a high level to a low level.
§In EIA Standard RS-422A, V_{OC}, which is the average of the two output voltages with respect to ground, is called output offset voltage, V_{OS}.
NOTE 4: This applies for both power on and power off. Refer to EIA Standard RS-422A for exact conditions.

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

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
t_{DD}	Differential-output delay time	R_L = 60 Ω,	See Figure 3		40	60	ns
t_{TD}	Differential-output transition time				65	95	ns
t_{PZH}	Output enable time to high level	R_L = 110 Ω,	See Figure 4		55	90	ns
t_{PZL}	Output enable time to low level	R_L = 110 Ω,	See Figure 5		30	50	ns
t_{PHZ}	Output disable time from high level	R_L = 110 Ω,	See Figure 4		85	130	ns
t_{PLZ}	Output disable time from low level	R_L = 110 Ω,	See Figure 5		20	40	ns

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RECEIVER SECTION

receiver electrical characteristics over recommended ranges of common-mode input voltage, supply voltage, and operating free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS			MIN	TYP†	MAX	UNIT
V_{TH}	Differential-input high-threshold voltage	V_O = 2.7 V,	I_O = −0.4 mA				0.2	V
V_{TL}	Differential-input low-threshold voltage	V_O = 0.5 V,	I_O = 8 mA		−0.2‡			V
$V_{T+} - V_{T-}$	Hysteresis§					50		mV
V_{IK}	Enable-input clamp voltage	I_I = −18 mA					−1.5	V
V_{OH}	High-level output voltage	V_{ID} = −200 mV, See Figure 2	I_{OH} = −400 μA,		2.7			V
V_{OL}	Low-level output voltage	V_{ID} = −200 mV, See Figure 2	I_{OL} = 8 mA,				0.45	V
I_{OZ}	High-impedance-state output current	V_O = 0.4 V to 2.4 V					±20	μA
I_I	Line input current	Other input = 0 V, See Note 4		V_I = 12 V			1	mA
				V_I = −7 V			−0.8	
I_{IH}	High-level enable-input current	V_{IH} = 2.7 V					20	μA
I_{IL}	Low-level enable-input current	V_{IL} = 0.4 V					−100	μA
r_i	Input resistance				12			kΩ
I_{OS}	Short-circuit output current				−15		−85	mA
I_{CC}	Supply current (total package)	No load		Outputs enabled		35	50	mA
				Outputs disabled		26	40	

†All typical values are at V_{CC} = 5 V, T_A = 25°C.
‡The algebraic convention, where the less-positive (more-negative) limit is designated minimum, is used in this data sheet for common-mode input voltage and threshold voltage levels only.
§Hysteresis is the difference between the positive-going input threshold voltage, V_{T+}, and the negative-going input threshold voltage, V_{T-}. See Figure 4.
NOTE 4: This applies for both power on and power off. Refer to EIA Standard RS-422A for exact conditions.

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

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
t_{PLH}	Propagation delay time, low-to-high-level output	V_{ID} = −1.5 V to 1.5 V, C_L = 15 pF,	See Figure 6		21	35	ns
t_{PHL}	Propagation delay time, high-to-low-level output				23	35	ns
t_{PZH}	Output enable time to high level	C_L = 15 pF,	See Figure 7		10	30	ns
t_{PZL}	Output enable time to low level				12	30	ns
t_{PHZ}	Output disable time from high level	C_L = 15 pF,	See Figure 7		20	35	ns
t_{PLZ}	Output disable time from low level				17	35	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION

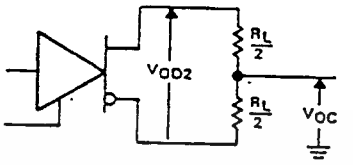


FIGURE 1. DRIVER V_{OD} AND V_{OC}

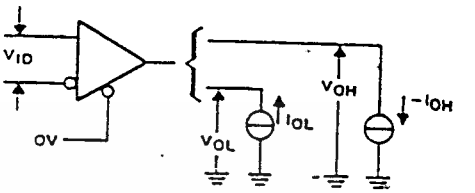


FIGURE 2. RECEIVER V_{OH} AND V_{OL}

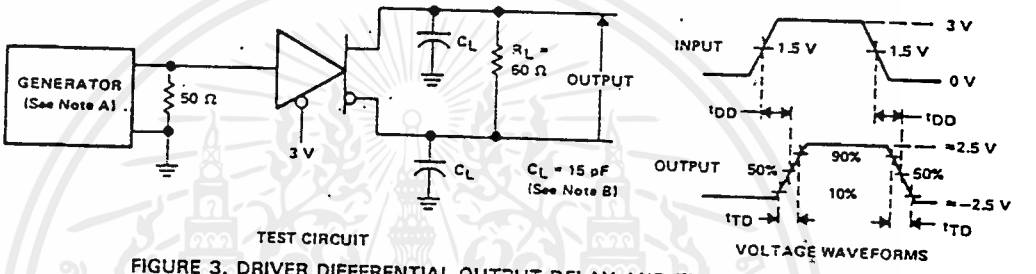


FIGURE 3. DRIVER DIFFERENTIAL-OUTPUT DELAY AND TRANSITION TIMES

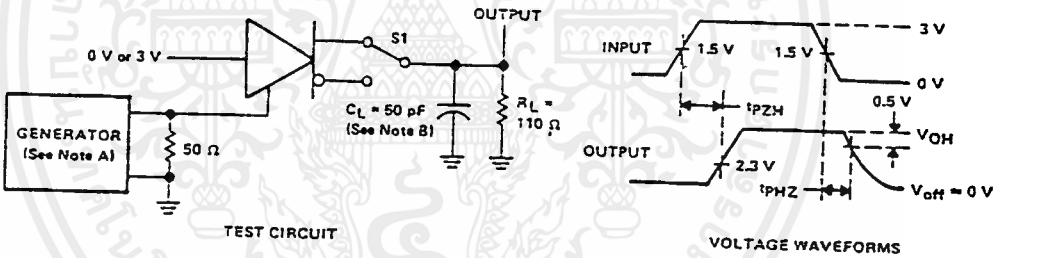


FIGURE 4. DRIVER ENABLE AND DISABLE TIMES

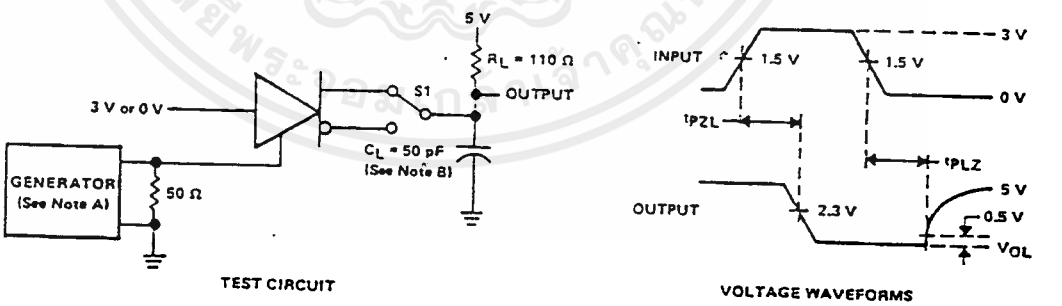


FIGURE 5. DRIVER ENABLE AND DISABLE TIMES

NOTES: A. The input pulse is supplied by a generator having the following characteristics: PRR = 1 MHz, 50% duty cycle, t_r ≤ 6 ns, t_f ≤ 6 ns, Z_{out} = 50 Ω.
B. C_L includes probe and jig capacitance.

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION

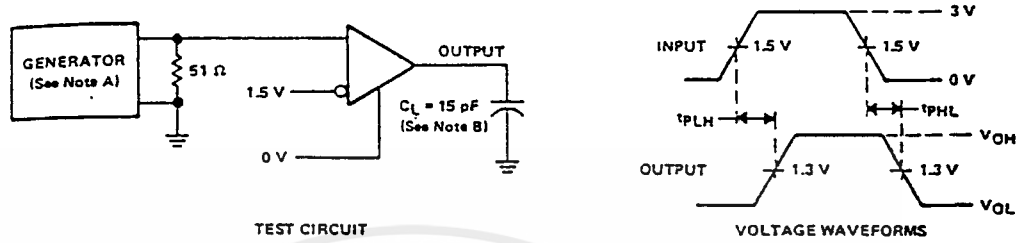


FIGURE 6. RECEIVER PROPAGATION DELAY TIMES

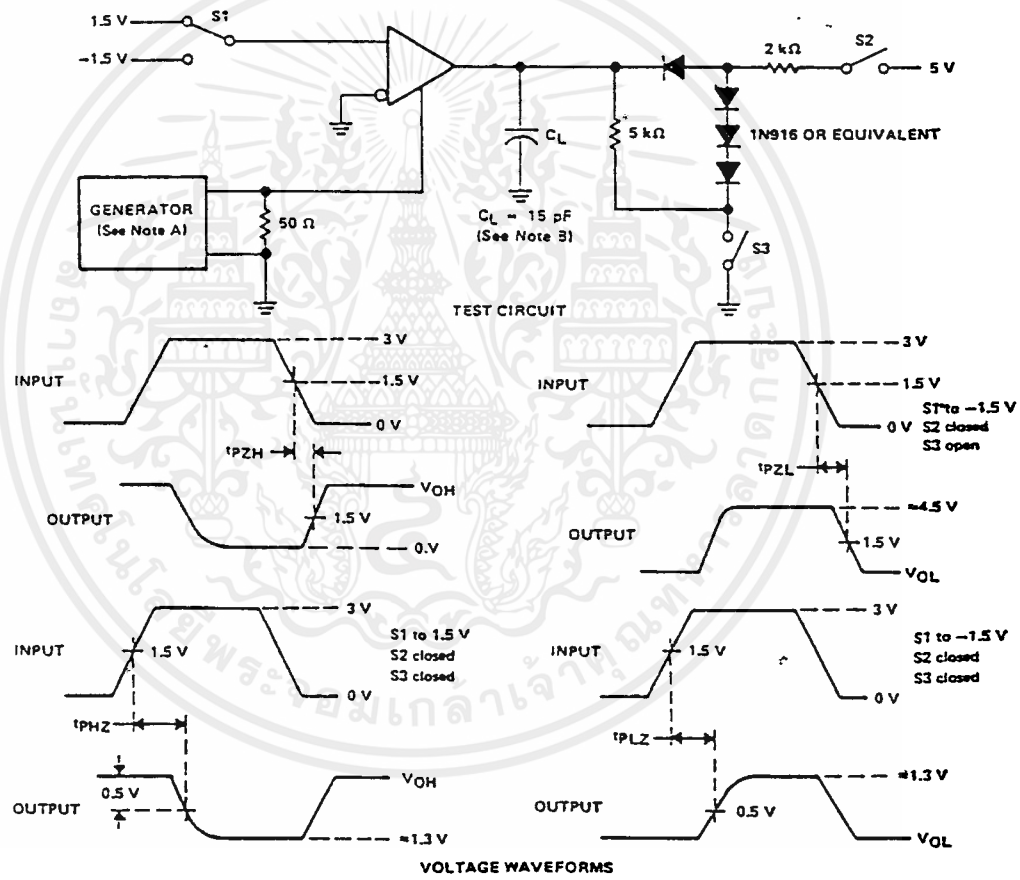


FIGURE 7. RECEIVER OUTPUT ENABLE AND DISABLE TIMES

NOTES: A. The input pulse is supplied by a generator having the following characteristics: PRR = 1 MHz, 50% duty cycle, t_r ≤6 ns, t_f ≤6 ns, Z_{out} = 50 Ω.
B. C_L includes probe and jig capacitance.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL CHARACTERISTICS

DRIVER HIGH-LEVEL OUTPUT VOLTAGE
vs
DRIVER HIGH-LEVEL OUTPUT CURRENT

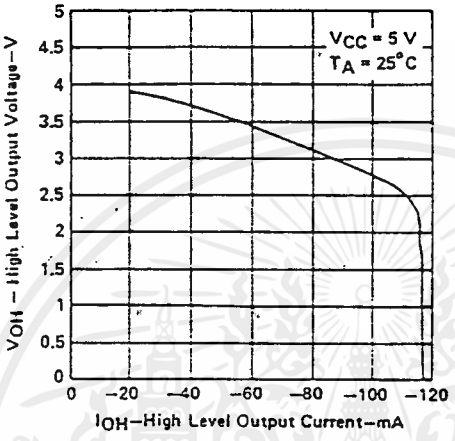


FIGURE 8

DRIVER LOW-LEVEL OUTPUT VOLTAGE
vs
DRIVER LOW-LEVEL OUTPUT CURRENT

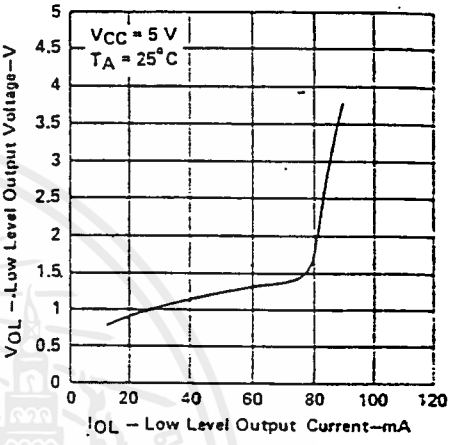


FIGURE 9

DRIVER DIFFERENTIAL OUTPUT VOLTAGE
vs
DRIVER OUTPUT CURRENT

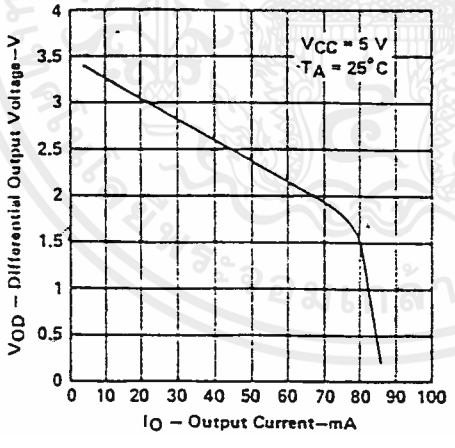


FIGURE 10

RECEIVER LOW-LEVEL OUTPUT VOLTAGE
vs
RECEIVER LOW-LEVEL OUTPUT CURRENT

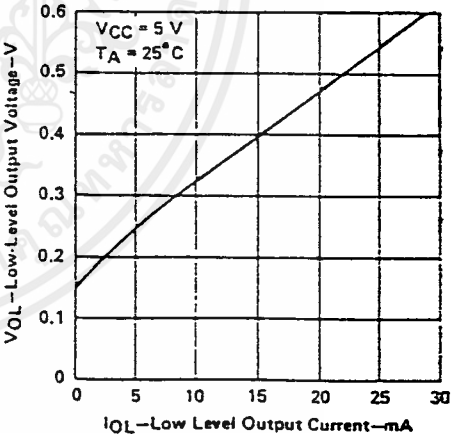


FIGURE 11

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL CHARACTERISTICS

RECEIVER LOW-LEVEL OUTPUT VOLTAGE
vs
FREE-AIR TEMPERATURE

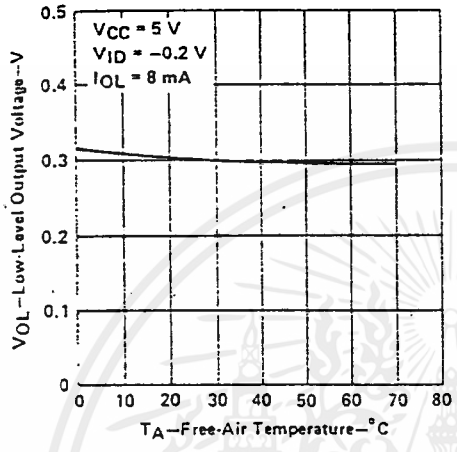


FIGURE 12

RECEIVER OUTPUT VOLTAGE
vs
ENABLE VOLTAGE

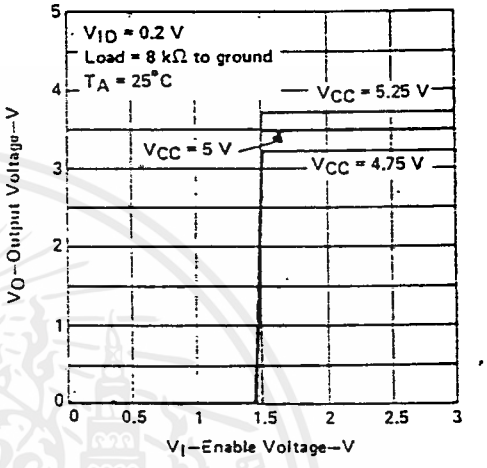


FIGURE 13

RECEIVER OUTPUT VOLTAGE
vs
ENABLE VOLTAGE

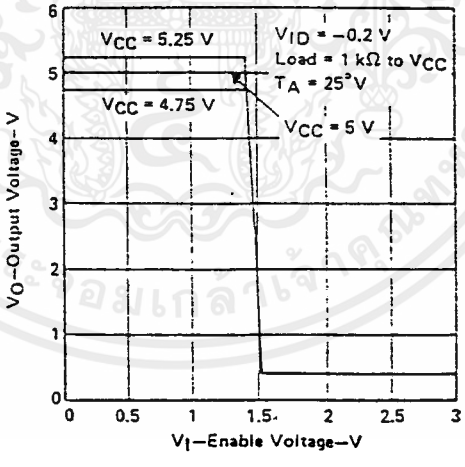
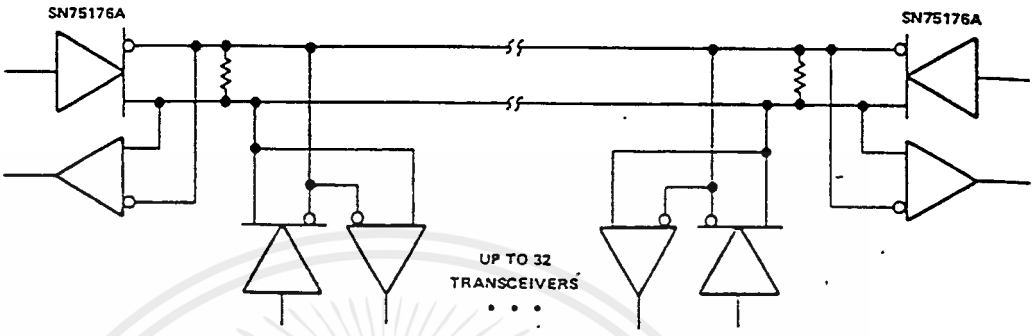


FIGURE 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL APPLICATION



NOTE: The line should be terminated at both ends in its characteristic impedance. Stub lengths off the main line should be kept as short as possible.

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ รศ.สมยศ จุณณะปิยะ ที่เอื้อเฟื้อสถานที่ อุปกรณ์ หนังสืออ้างอิง และให้คำปรึกษาในการทำโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้