ปีการศึกษา 2537

การสื่อสารข้อมูลด้วยไมโครคอนโทรลเลอร์

โดย

นายศักดิ์ชัย แววศักดิ์ 35.102118

นายไสว เหล่าไม้ 35.102125

อาจารย์ที่ปรึกษา

รศ. พินัฒน์ เลาหสงคราม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2537

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การสื่อสารข้อมูลด้วยไมโครคอนโทรลเลอร์

ผู้จัดทำ

1. นาย ศักดิ์ชัย แววศักดิ์ รหัสประจำตัว 35.102118

2. นาย ไสว เหล่าไม้ รหัสประจำตัว 35.102125

..... อาจารย์ที่ปรึกษา

(รศ. พิพัฒน์ เลาหสงคราม)

..... อาจารย์ที่ปรึกษา

( )

..... อาจารย์ที่ปรึกษา

( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารข้อมูลด้วยไมโครคอนโทรลเลอร์

ศักดิ์ชัย แวาศักดิ์ 35.102118
ไสว เหล่าไม้ 35.102125
รศ.พิพัฒน์ เลาหสงคราม อาจารย์ที่ปรึกษา
ปีการศึกษา 2537

บทคัดย่อ

อุปกรณ์ไมโครคอนโทรลเลอร์ได้ถูกนำมาใช้ในงานอุตสาหกรรมและในระบบควบคุมของอุปกรณ์เครื่องมือ-เครื่องใช้ต่าง ๆ ซึ่งการควบคุมและการใช้งานอุปกรณ์นี้หากมีการติดต่อกับอุปกรณ์ภายนอกแล้ว จะสามารถติดต่อได้เพียงชุดเดียว โดยรับส่งผ่านทางขารับ (RXD) และขาส่ง (TXD) ของตัวซีพียู (CPU) ดังนั้นปริญญานิพนธ์ฉบับนี้ได้นำเอาอุปกรณ์ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มาประยุกต์ใช้ในการติดต่อสื่อสาร เพื่อให้สามารถติดต่อแบบอนุกรม (SERIES) ได้หลายชุด ด้วยลักษณะการติดต่อแบบจุดต่อจุด (POINT-TO-POINT) ที่ใช้ในการติดต่อแบบอนุกรมในช่วงระยะทางใกล้ ๆ ระหว่างบอร์ดมาสเตอร์ (MASTER) กับบอร์ดไมโครคอนโทรเลอร์ หรือบอร์ดมาสเตอร์กับเครื่องคอมพิวเตอร์ (COMPUTER) มาตรฐาน RS-232-C และการติดต่อแบบจุดต่อหลายจุด(POINT-TO-MULTIPOINT)ที่ใช้ในการติดต่อในช่วงระยะทางไกล ๆ ในลักษณะบอร์ดมาสเตอร์กับบอร์ดสลาฟ (SLAVE) ตามมาตรฐาน RS-485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MICROCONTROLLER BASED COMMUNICATION

MR. SUKCHAI WAWSAK 35.102118

MR. SAWAI LAMAI 35.102125

ASSOC.PROF.PIPHAT LAUHASONGKRAM

YEAR 2537

ABSTRACT

MICROCONTROLLER DEVICE IS USED IN CONTROL SYSTEM OFINDUSTRIAL. THE CONTROL AND LINKING TO EXTERNAL-DEVICE CAN BE CONTROLLED ONLY ONE POINT WHICH RECEIVE BY RXD PIN AND TRANSMIT BY TXD PIN OF MICROPROCESSOR. THIS THESIS PRESENT APPLICATION OF COMMUNICATION FOR MULTI CHANNEL SERIES.THE POINT-TO-POINT COMMUNICSTION USES IN APPROACH AREA BETWEEN MASTERBOARD WITH OTHER BOARD AND MASTERBOARD WITH COMPUTER AS RS-232 STANDARD COMMUNICATION. THE POINT-TO-MULTIPOINT COMMUNICATION USES IN A LONG DISTANCE BETWEEN MASTERBOARD WITH SLAVEBOARD AS RS-485 STANDARD COMMUNICATION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันอุปกรณ์ไมโครคอนโทรลเลอร์เป็นอุปกรณ์ที่ถูกนำมาใช้ในงานทางด้านอุตสาหกรรมกันอย่างกว้างขวาง เช่นตามบอร์ดคอนโทรลต่าง ๆ ปริญญานิพนธ์ฉบับนี้จึงได้กล่าวถึงการนำอุปกรณ์ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 มาประยุกต์ใช้ทางการสื่อสารข้อมูลแบบอนุกรม ตามมาตราฐาน RS-232 และ RS-485 โดยมีรายละเอียดของแต่ละบทดังนี้

บทที่ 1 ซึ่งเป็นบทนำ กล่าวถึงวัตถุประสงค์โดยย่อของโครงงาน

บทที่ 2 ทฤษฎีการสื่อสารข้อมูลแบบอนุกรม กล่าวถึง ลักษณะของ MCS-51 พอร์ตสื่อสารข้อมูลแบบอนุกรม โปรโตรคอล การสื่อสารข้อมูลมาตราฐาน RS-232 และ RS-485

บทที่ 3 การออกแบบและการควบคุมด้วยซอฟแวร์ กล่าวถึงการออกแบบส่วนฮาร์ดแวร์และส่วนซอฟแวร์

บทที่ 4 ฟังก์ชันและการใช้งาน กล่าวถึงตำแหน่งและลักษณะการทำงานของคีย์บอร์ด การแสดงผล การต่อใช้งาน

บทที่ 5 บทสรุปและข้อแนะนำ กล่าวถึงการนำมาตราฐาน RS-232 RS-485 ไปใช้ให้เหมาะสมกับงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

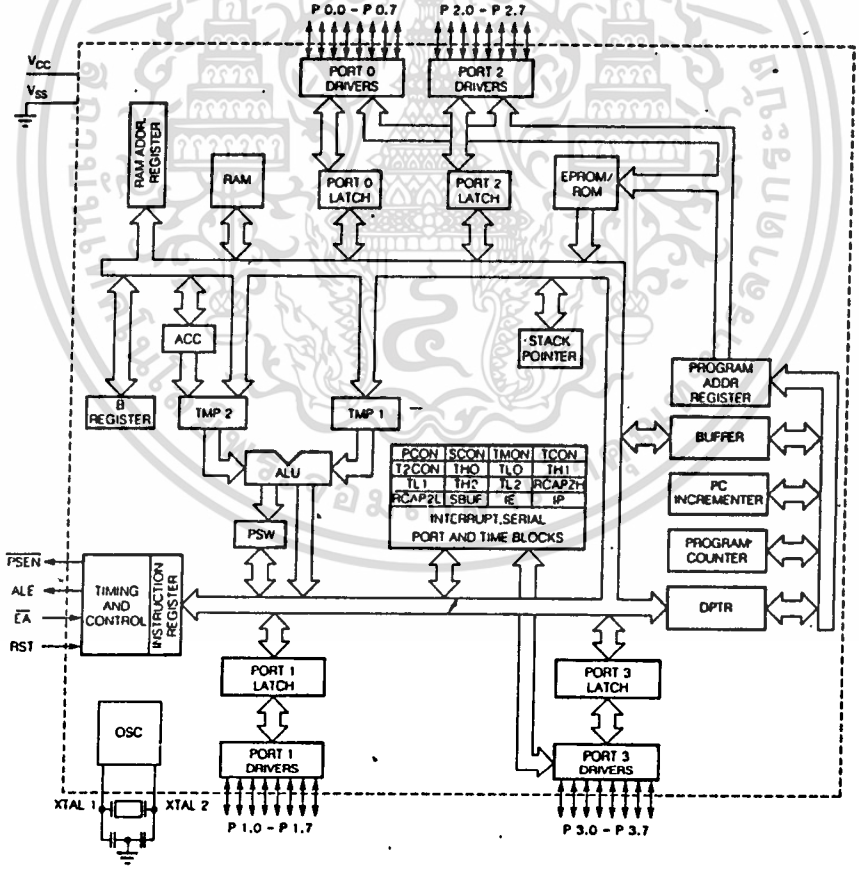
บทที่ 2

ทฤษฎีเกี่ยวกับการสื่อสารข้อมูล

2.1 ลักษณะของ MCS-51

โครงสร้างภายในของชิปไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีดังรูปที่ 2.1 และ เนื่องจาก MCS-51 แต่ละเบอร์มีลักษณะและโครงสร้างไม่เหมือนกันทีเดียว 8051s จะประกอบด้วยเบอร์ 8051,8051AH และ 80C51BH และ ROMLESS version คือเบอร์ 8031 (เป็น MCS-51 เบอร์ที่ไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายชิป) และ EPROM version คือเบอร์ 8751 (เป็น MCS-51 เบอร์ที่มีหน่วยความจำสำหรับเก็บโปรแกรมเป็น EPROM อยู่ในชิป)

8052s ประกอบด้วย 8052AH,8032AH และ 8752BH



รูปที่ 2.1 แสดงโครงสร้างภายในชิปไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 โครงสร้างและการทำงานของพอร์ต

โครงสร้างภายใน MCS-51 ที่เราจะกล่าวถึงเป็นอันดับแรกคือพอร์ตใน MCS-51 ซึ่งมีพอร์ตขนาด 8 บิตรวมทั้งสิ้น 4 พอร์ตแต่ละพอร์ตมีหน้าที่ติดต่อกับวงจรภายนอก โดยสามารถรับและส่งข้อมูลกับวงจรภายนอกได้ กล่าวคือ ทั้ง 4 พอร์ตจะเป็นพอร์ตแบบสองทิศทาง (bidirectional port) โครงสร้างของแต่ละพอร์ตจะประกอบด้วย

1. วงจรแลตช์ซึ่งสถานะของเอาต์พุตของวงจรแลตช์จะปรากฏที่รีจิสเตอร์ใช้งานเฉพาะ P0 ถึง P3
2. วงจรเอาต์พุตไดรเวอร์
3. วงจรอินพุตบัฟเฟอร์

เอาต์พุตไดรเวอร์ของพอร์ต 0 และพอร์ต 2 กับอินพุตบัฟเฟอร์ของพอร์ต 0 จะถูกใช้ในการติดต่อหน่วยความจำสำหรับเก็บโปรแกรมภายนอกชิป โดยมีการทำงานดังนี้

พอร์ต 0 ใช้ส่งค่าแอดเดรสไบต์ต่ำ (low byte - [A0-A7]) ของตำแหน่งความจำภายนอก (หน่วยความจำสำหรับเก็บโปรแกรมหรือข้อมูลภายนอกชิป) และยังถูกใช้เป็นดาต้าบัส (data bus) ซึ่งจะใช้ทั้งในการรับและส่งข้อมูลกับหน่วยความจำภายนอก โดยจะผลัดกันใช้คนละเวลาแบบ time multiplex

พอร์ต 2 ใช้ส่งค่าแอดเดรสไบต์สูง (high byte - [A8-A15]) ของตำแหน่งความจำภายนอก (หน่วยความจำสำหรับเก็บโปรแกรมหรือข้อมูลภายนอกชิป) เมื่อมีการใช้แอดเดรสขนาด 16 บิต และถ้าพอร์ต 2 ไม่ถูกใช้งาน (ติดต่อหน่วยความจำภายนอกน้อยกว่า 256 ไบต์) คือถูกใช้ไม่ครบ 16 บิต (ใช้หน่วยความจำภายนอกไม่ครบ 64 กิโลไบต์) พอร์ต 2 บิตที่ไม่ได้ใช้ก็จะส่งค่าภายในรีจีสเตอร์ใช้งานเฉพาะ P2 ออกมาที่แต่ละขา (ซึ่งก็คือค่าที่แลตช์อยู่ภายใน)

พอร์ต 3 ทั้งหมดรวมถึงพอร์ต 1 ของ 8052 อีก 2 ขา จะไม่เพียงแต่ใช้เป็นพอร์ตอินพุตเอาต์พุตเท่านั้น เพราะแต่ละบิตของพอร์ต 3 และพอร์ต 1 บางบิตยังมีหน้าที่พิเศษอื่นอีก ดังและในตาราง 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงหน้าที่พิเศษอื่น ๆ ของพอร์ต 1 และพอร์ต 3

พอร์ต	หน้าที่
P1.0 (T2)	เป็นอินพุตให้เคาน์เตอร์ของไทม์เมอร์ 2
P1.1 (T2EX)	เป็นสัญญาณควบคุมการทำงานของไทม์เมอร์ 2 รายละเอียดจะกล่าวต่อไป
P3.0 (RXD)	เป็นอินพุตของพอร์ตสื่อสารอนุกรม
P3.1 (TXD)	เป็นเอาต์พุตของพอร์ตสื่อสารอนุกรม
P3.2 (1NT0)	เป็นอินพุตของอินเตอร์รัปต์ภายนอกชนิด 0
P3.3 (1NT1)	เป็นอินพุตของอินเตอร์รัปต์ภายนอกชนิด 1
P3.4 (TO)	เป็นอินพุตให้เคาน์เตอร์ของไทม์เมอร์ 0
P3.5 (T1)	เป็นอินพุตให้เคาน์เตอร์ของไทม์เมอร์ 1
P3.6 (WR)	เป็นสัญญาณควบคุมการเขียนข้อมูลไปหน่วยความจำสำหรับเก็บข้อมูลภายนอก
P3.7 (RD)	เป็นสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำภายนอกเข้ามา

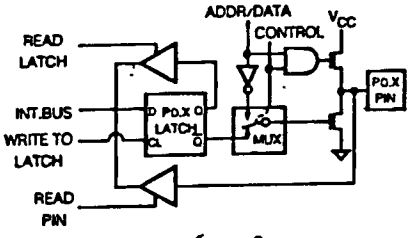
หน้าที่พิเศษเหล่านี้จะใช้ได้ก็ต่อเมื่อค่าในบิตแลตช์ของพอร์ตในบิตนั้น ๆ มีค่าเป็น 1 มิฉะนั้น ที่ขาของพอร์ตภายนอกชิปจะมีค่าเป็น 0

2.1.2 การกำหนดการใช้งานอินพุต/เอาต์พุต (I/O configuration)

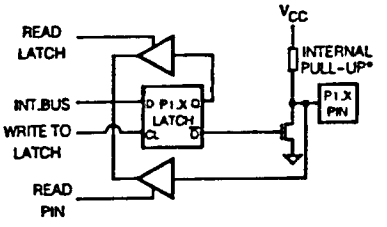
รูปที่ 2.2 แสดงการทำงานของบิตแลตช์และอินพุต/เอาพุตบัฟเฟอร์ของแต่ละพอร์ต ใน 4 พอร์ต (หนึ่งบิตในรีจิสเตอร์ใช้งานเฉพาะของพอร์ต) เป็น D flip-flop ซึ่งเปลี่ยนค่าของเอาต์พุตเป็นค่าตามค่าของ internal bus เมื่อมีสัญญาณ WRITE TO LATCH จากซีพียู และเอาต์พุตของฟลิปฟลอปจะมีค่าไปปรากฏบน interbal bus เมื่อมีสัญญาณ READ LATCH จากซีพียู ส่วนระดับของสัญญาณที่ขาของพอร์ตจะไปปรากฏที่ internal bus ก็ต่อเมื่อสัญญาณ READ PIN จากซีพียู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

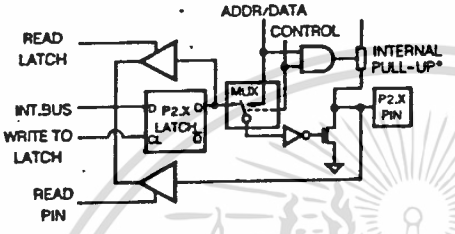
คำสั่งบางคำสั่งที่อ่านค่าจากพอร์ตจะทำให้เกิดสัญญาณ READ LATCH และที่เหลือจะไปทำให้เกิดสัญญาณ READ PIN



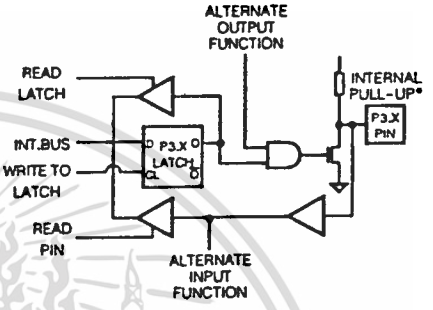
(ก) พอร์ต 0 บิต



(ข) พอร์ต 1 บิต



(ค) พอร์ต 2 บิต



(ง) พอร์ต 3 บิต

รูปที่ 2.2 แสดงภาพโครงสร้างแต่ละบิตของพอร์ตใน MCS-51

รูปที่ 2.2 เอาต์พุตไดรเวอร์ของพอร์ต 0 และพอร์ต 2 จะสามารถเปลี่ยนการทำงานไปเป็น Int bus และ Addr/Data Bus ได้จากสัญญาณควบคุมภายใน (สัญญาณ CONTROL) สำหรับใช้ในการติดต่อกับหน่วยความจำภายนอก (ทั้งหน่วยความจำสำหรับเก็บโปรแกรมและเก็บข้อมูลภายนอก) โดยค่าที่ขาของสัญญาณควบคุมภายในมีค่าเป็น 1 จะเปิดเกตให้ addr/data ออกมาได้ แต่ถ้าขานี้มีค่าเป็น 0 จะทำหน้าที่เป็นพอร์ตปกติ และระหว่างการติดต่อกับหน่วยความจำภายนอกนี้ ค่าของรีจิสเตอร์ใช้งานเฉพาะ P2 จะคงค่าเดิมไว้แต่รีจีสเตอร์ใช้งานเฉพาะ P0 จะถูกโหลดด้วยค่า 1 เพื่อ ทำให้พอร์ต 0 มีคุณสมบัติเป็น high impedance

สิ่งที่แสดงในรูปที่ 2.2 อีกอย่างคือ ถ้าบิตแลตช์ของพอร์ต 3 มีค่าเป็น 1 แล้ว ระดับของสัญญาณเอาต์พุตจะถูกควบคุมโดยสัญญาณ Alternate ouput function และระดับสัญญาณที่ขา P3.x จริง ๆ จะไปปรากฏเป็นสัญญาณ Alternate input function ที่แต่ละบิตของพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ต 1,2 และ 3 มีวงจรพูลอัพ (วงจรยกระดับแรงดันของสัญญาณ) ภายใน (internal pullups)พอร์ต 0 มีเอาต์พุตแบบ open drain และอินพุต/เอาต์พุตแต่ละเส้นสามารถใช้แยกอิสระจากกันสำหรับใช้เป็นอินพุตหรือเอาต์พุต (พอร์ต 0 และ 2 อาจไม่สามารถถูกใช้เป็นพอร์ตอินพุต/เอาต์พุตทั่วไปเมื่อถูกใช้ใน การติดต่อหน่วยความจำภายนอก) เมื่อพอร์ตถูกกำหนดการใช้งานเป็นอินพุต ค่าในบิตแลตช์ของพอร์ตต้องมีค่าเป็น 1 ซึ่งจะไป turn off เอาต์พุตไดรเวอร์ FET (ลอจิก 0 ที่ขาเกตของ FET จะปิดช่องทางนำกระแสของ field Effect Transistor เพราะเป็น Enhancement Mode n-Channel MOSFET ส่วนลอจิก 1 ที่ขาเกตจะเป็นตัวเปิดช่องทางนำกระแสและทำให้ FET มีคุณสมบัติเป็นตัวต้านทาน) ดังนั้นขาของพอร์ต 1,2, และ 3 จะถูกยกระดับสัญญาณขึ้นเป็น 1 โดยวงจรพูลอัพภายใน แต่สามารถถูกดึงระดับสัญญาณลงต่ำเป็น 0 ได้จากวงจรภายนอก

ในกรณีของพอร์ต 0 จะมีโครงสร้างภายในแตกต่างจากพอร์ตอื่นตรงที่ไม่มีวงจรพูลอัพภายในแต่ใช้ทรานซิสเตอร์ชนิด FET เป็นตัวยกระดับแรงดันแทน (pullup FET) ซึ่งจะถูกใช้งานเฉพาะเมื่อพอร์ต 0 ส่งค่า 1 ออกมาภายนอกเฉพาะระหว่างช่วงการติดต่อกับหน่วยความจำภายนอกเท่านั้น ส่วนในกรณีอื่น ๆ pullup FET จะไม่ถูกใช้งาน (off) ดังนั้นพอร์ต 0 ที่ถูกใช้เป็นเอาต์พุตพอร์ตจะเป็นเอาต์พุตชนิด open drain เพราะฉะนั้นการเขียนค่า 1 ไปยังแต่ละบิตแลตช์จะทำให้ FET ที่ทำหน้าเป็นเอาต์พุตถูกปิดช่องทางนำกระแส ทำให้ขาของพอร์ตมีสถานะ high impedance เพื่อใช้พอร์ตเป็นอินพุตในขณะทำหน้าที่เป็นดาต้าบัส

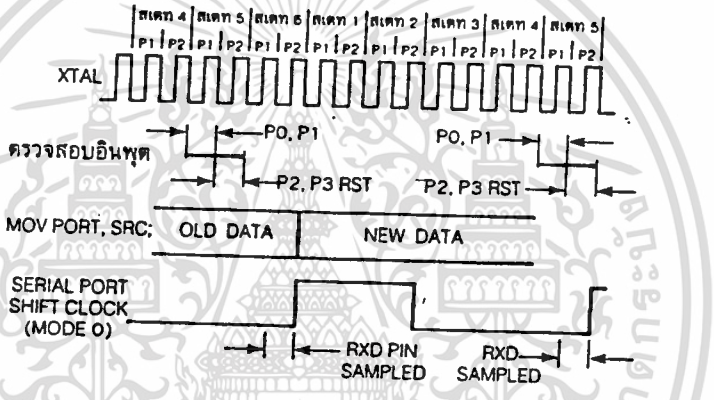
เพราะว่าพอร์ต 1,2 และ 3 มีวงจรพูลอัพภายในแบบถาวร จึงมักเรียกว่า quasi bidirectional port เมื่อถูกกำหนดเป็นอินพุต วงจรพูลอัพภายในนี้จะทำการยกระดับสัญญาณให้มีค่าสูง และจะจ่ายกระแสเมื่อถูกดึงระดับสัญญาณลงต่ำจากภายนอก แต่สำหรับพอร์ต 0 จะเรียกว่าเป็นพอร์ตแบบ bidirectional จริง ๆ เพราะว่าเมื่อเป็นอินพุตจะมีสถานะเป็น high impedance (float state)

พอร์ตแลตช์ทั้งหมดใน 8051 จะมีค่าเป็น 1 เมื่อ 8051 ถูกรีเซต (ซึ่งจะทำให้แต่ละบิตของทุกพอร์ตสามารถใช้งานเป็นอินพุตได้) ถ้ามีการเขียนค่า 0 ไปที่พอร์ตแลตช์ภายหลังการรีเซตเราจะสามารถทำให้พอร์ตมีคุณสมบัติเป็นอินพุตได้อีกครั้ง โดยเขียนค่า 1 ไปที่รีจีสเตอร์ใช้งานเฉพาะของแต่ละพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 การเขียนข้อมูลไปที่พอร์ต

ในการทำงานของคำสั่งที่มีการเปลี่ยนข้อมูลในพอร์ตแลตช์ ค่าใหม่จะไปปรากฏที่วงจรแลตช์ในช่วงสเตท 6 เฟส 2 ของไซเกิลสุดท้ายของคำสั่ง แต่ในความเป็นจริงพอร์ตแลตช์จะตรวจสอบ (simple) เอาต์พุตบัฟเฟอร์ของมันระหว่างเฟส 1 ของสัญญาณคาบใด ๆ เท่านั้น (ระหว่างเฟส 2 เอาต์พุตบัฟเฟอร์จะคงเป็นค่าที่มันเจอเมื่อเฟส 1 ก่อนหน้านี้) ผลก็คือค่าใหม่ที่พอร์ตแลตช์จะยังไม่ไปปรากฏที่ขาของพอร์ตจนกว่าจะถึงเฟส 1 ครั้งถัดไป ซึ่งจะเป็นสเตท 1 เฟส 1 ของแมชชีนไซเกิลถัดไป ดูรูปที่ 2.3 ในเรื่องแผนผังเวลาภายใน (internal timing)



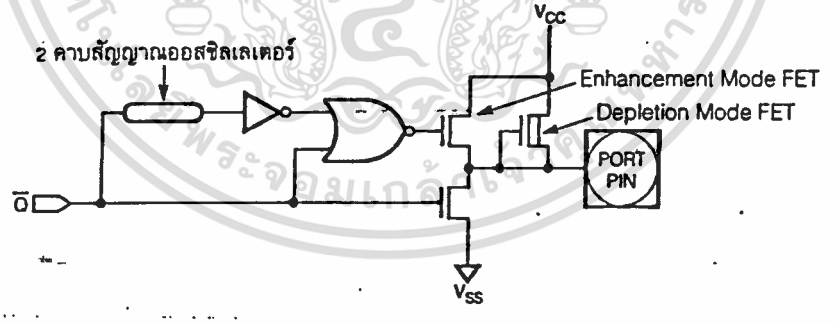
รูปที่ 2.3 แผนผังเวลาภายใน

ถ้าต้องมีการเปลี่ยนค่าจาก 0 เป็น 1 ในพอร์ต 1,2 และ 3 วงจรพูลอัพเพิ่มเติม (addition pullup) จะถูก turn on ระหว่างสเตท 1 เฟส 1 และสเตท 1 เฟส 2 ของไซเกิลที่เกิดการเปลี่ยนแปลง การทำเช่นนี้เพื่อเพิ่มความเร็วในการเปลี่ยนแปลง โดยวงจรพูลอัพที่มีเพิ่มขึ้นมา (extra pullup) นี้สามารถที่จะจ่ายกระแสได้ประมาณ 100 เท่าของที่วงจรพูลอัพธรรมดาสามารถจ่ายได้ และควรสังเกตว่าวงจรพูลอัพภายในนั้นเป็น FET ไม่ใช่ความต้านทานที่มีความเป็นเชิงเส้น (linear resistors) รูปแบบของการพูลอัพแสดงในรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

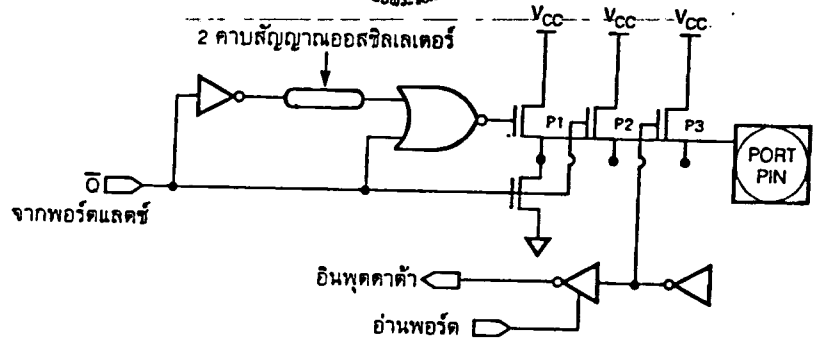
ใน MCS-51 ที่ผลิตโดยใช้เทคโนโลยี HMOS (HMOS version) ส่วนที่เป็นวงจรพูลอัพถาวรจะเป็น Depletion Mode Transistor ซึ่งขา Gate ต่ออยู่กับขา Source (จะทำหน้าที่เป็น Current Source เมื่อขา Source มีลอจิก 0 โดยที่ขา Drain ต่อกับ Vcc) เมื่อถูกต่อลงกราวด์โดยตรง ทรานซิสเตอร์แบบนี้จะจ่ายกระแสได้ประมาณ 0.25 มิลลิแอมแปร์ ส่วนวงจรที่ต่อขนานกับส่วนวงจรพูลอัพถาวรนี้จะเป็น Enhancement-mode Transistor ซึ่งจะทำงานที่สเตท 1 ในขณะที่พอร์ตบิตมีการเปลี่ยนค่าจาก 0 เป็น 1 ในช่วงนี้หากขาพอร์ตถูกต่อลงกราวด์โดยตรง ทรานซิสเตอร์พิเศษตัวนี้ (Enchancement Mode Transister) จะสามารถจ่ายกระแสเพิ่มได้อีก 30 มิลลิแอมแปร์

ใน MCS-51 ที่ผลิตโดยใช้เทคโนโลยี CHMOS (CHMOS version) วงจรพูลอัพจะประกอบด้วยทรานซิสเตอร์ชนิด pFET จำนวน 3 ตัว ควรจะสังเกตว่า n-channel FET (nFET) จะถูก turn on เมื่อให้ลอจิก 1 กับขา gate ของมันและจะ turn off เมื่อให้ลอจิก 0 กับขา gate ของมันส่วน p-channel FET (pFET) จะตรงข้ามกันคือจะ on เมื่อขา gate เป็น 0 และจะ off เมื่อขา gate เป็น 1



(ก) วงจรพูลอัพภายในของ MCS-51 HMOS ทรานซิสเตอร์ชนิด FET จะนำกระแสเป็นเวลา 2 คาบสัญญาณออสซิลเลเตอร์ ภายหลังจาก 0 เปลี่ยนสถานะ จาก 0 เป็น 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข) วงจรพูลอัพภายในของ MCS-51 ชนิด CHMOS ทรานซิสเตอร์ pFET1 จะนำกระแสเป็นเวลา 2 คาบสัญญาณออสซิลเลเตอร์ ภายหลัง 0 เปลี่ยนสถานจาก 0 เป็น 1 ในเวลาเดียวกันทรานซิสเตอร์ pFET จะไปบังคับให้ทรานซิสเตอร์ pFET3 นำกระแสผ่านวงจรอินเวอร์เตอร์ pFET2 นำกระแสด้วย

รูปที่ 2.4 รูปแบบของการพูลอัพ

pFET1 ในรูปที่ 2.4 เป็นทรานซิสเตอร์ซึ่งจะถูก turn on หลังจากการเปลี่ยนจาก 0 เป็น 1 ที่พอร์ตแลตช์เป็นเวลา 2 คาบสัญญาณออสซิลเลเตอร์ และเมื่อมัน on จะทำให้ pFET3 ถูก turn on (เป็นการพูลอัพที่จ่ายกระแสได้น้อย) ผ่านทางอินเวอร์เตอร์ (inverter) โดยอินเวอร์เตอร์ตัวนี้และ pFET จะรักษาสถานะซึ่งมีค่าเป็น 1 ไว้

สังเกตว่าถ้าขณะที่ขาของพอร์ตมีค่าลอจิก 1 สัญญาณรบกวนที่เป็นลอจิก 0 ที่ขาจากแหล่งกำเนิดสัญญาณภายนอกสามารถที่จะ turn off pFET3 ซึ่งทำให้ขาเป็นสถานะขาลอย (float state) pFET2 ในแบบของ CMOS ที่ใช้กันทั่วไปเป็นพูลอัพระดับต่ำ (จ่ายกระแสได้น้อย) ซึ่งจะ turn on เมื่อใดก็ตามที่ nPET off โดยจะมีความสามารถในการจ่ายกระแสประมาณ 1 ใน 10 ของ pFET3 หน้าที่ของมันคือการทำให้ขากลับมามีสถานเป็น 1 ตามเดิมในเหตุการณ์ที่ขามีค่าเป็น 1 และถูกเปลี่ยนสถานะเป็น 0 เนื่องจากสัญญาณรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
034754

2.1.4 การต่อใช้งานพอร์ต (port loading and interfacing)

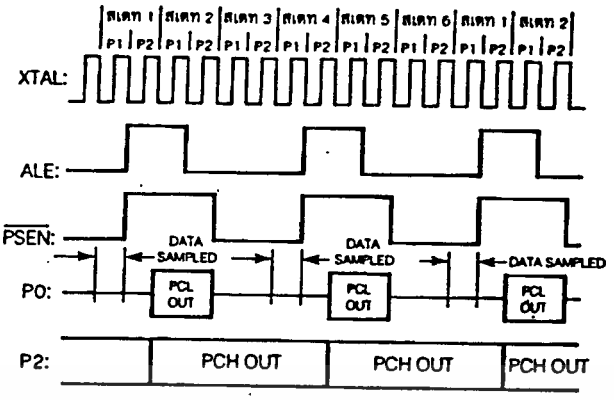
เอาต์พุตบัฟเฟอร์ของพอร์ต 1,2 และ 3 แต่ละตัวสามารถขับอินพุตไอซีชนิด LS TTL ได้ทั้งหมด 4 ตัว ใน MCS-51 ที่ผลิตโดยใช้เทคโนโลยีชนิด HMOS พอร์ตเหล่านี้สามารถถูกขับในแบบปกติโดยวงจร TTL หรือ NMOS แต่ละขาของทั้ง HMOS และ CHMOS สามารถถูกขับโดย open-collector และ open-drain outputs แต่ให้สังเกตว่าการเปลี่ยนแปลงจาก 0 เป็น 1 จะไม่เร็ว ในอุปกรณ์พวก HMOS ถ้าขาของพอร์ตถูกขับโดย open-collector output แล้วการเปลี่ยนแปลงจาก 0 เป็น 1 จะต้องถูกขับโดย depletion mode ที่อ่อนกว่าในรูปที่ 3.4 (ก) ส่วนในอุปกรณ์พวก CHMOS อินพุตที่เป็น 0 จะ turn off pullup pFET3 โดยปล่อยให้พูลอัพ pFET2 ที่อ่อนมาก ๆ ขับทรานซิสเตอร์เท่านั้น

ในช่วงการทำงานเพื่อติดต่อหน่วยความจำภายนอกของพอร์ต 0 เอาต์พุตบัฟเฟอร์ของพอร์ต 0 แต่ละตัวสามารถขับอินพุตของไอซีชนิด LS TTL ได้ 8 ตัว แต่ในช่วงการใช้งานเป็นพอร์ตอินพุตหรือเอาต์พุตทั่วไป พอร์ต 0 จะต้องการวงจรพูลอัพภายนอก ในการขับอินพุตใด ๆ เพิ่มเติม

2.1.5 การติดต่อหน่วยความจำภายนอกชิป (accessing external memory)

การติดต่อกับหน่วยความจำภายนอกมี 2 แบบ คือ การติดต่อกับหน่วยความจำสำหรับเก็บโปรแกรมภายนอกและการติดต่อกับหน่วยความจำสำหรับเก็บข้อมูลภายนอก การติดต่อกับหน่วยความจำสำหรับเก็บโปรแกรมภายนอกจะใช้สัญญาณ PSEN (Program Store Enable) เป็นสัญญาณ Read Stobe ส่วนการติดต่อกับหน่วยความจำสำหรับเก็บข้อมูลภายนอกจะใช้สัญญาณ RD หรือ WR (เป็น Altemate Function ของ P3.7 และ P3.6) เป็นสัญญาณควบคุมการอ่านและเขียนข้อมูลดังแสดงรายละเอียดในรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงสัญญาณติดต่อหน่วยความจำภายนอก

การอ่านโปรแกรมจากหน่วยความจำสำหรับ เก็บโปรแกรมภายนอกจะใช้แอดเดรสจำนวน 16 บิตเสมอ ส่วนการติดต่อกับหน่วยความจำสำหรับเก็บข้อมูลภายนอกสามารถใช้แอดเดรสจำนวน 16 บิต (โดยคำสั่ง MOVX @DPTR) หรือใช้แอดเดรสจำนวน 8 บิต (โดยใช้คำสั่ง MOVX @Ri)

เมื่อใดที่ใช้แอดเดรสจำนวน 16 บิต ค่าตำแหน่งหน่วยความจำไบต์สูงจะมาปรากฏที่พอร์ต 2 ซึ่งจะคงค่าอยู่ในช่วงไซเกิลของการอ่านหรือเขียน และให้สังเกตว่าไดรเวอร์ของพอร์ต 2 ใช้ตัวต้านทานพูลอัพที่มีความเข้มมาก ตลอดเวลาที่มันส่งค่า 1 ซึ่งก็เป็นการทำงานของคำสั่ง MOVX @DPTR ระหว่างเวลานี้ รีจิสเตอร์ใช้งานเฉพาะของพอร์ต 2 ไม่จำเป็นต้องมีค่าเป็น 1 และค่าของรีจีสเตอร์ใช้งานเฉพาะพอร์ต 2 จะไม่ถูกเปลี่ยน ถ้าช่วงไซเกิลการติดต่อหน่วยความจำภายนอกไม่ตามด้วยช่วงไซเกิลการติดต่อหน่วยความจำภายนอกอีกครั้งหนึ่งในทันที ค่าของพอร์ต 2 ที่ไม่ถูกเปลี่ยนก็จะกลับมาปรากฏในไซเกิลถัดไป

ถ้าใช้แอดเดรสขนาด 8 บิต (ใช้คำสั่ง MOVX @Ri) ข้อมูลในรีจิสเตอร์ใช้งานเฉพาะของพอร์ต 2 ก็จะคงปรากฏที่ขาของพอร์ต 2 ตลอดช่วงการติดต่อหน่วยความจำภายนอก ซึ่งสามารถทำให้ใช้วิธีการแบ่งช่วงหน่วยความจำ (paging) ได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีใดก็ตามค่าตำแหน่งหน่วยความจำไบต์ต่ำจะถูกผลัดกันใช้คนละเวลา (time-multiplex) กับดาต้าไบต์ที่พอร์ต 0 โดยที่สัญญาณ Address/Data จะขับ FET ทั้งสองในพอร์ต 0 เอาต์พุตบัฟเฟอร์ ดังนั้นในการกระทำเช่นนี้ขาของพอร์ต 0 จะไม่เป็น open-drain output และไม่ต้องการวงจรพูลอัพภายนอกเพิ่ม สัญญาณ ALE ควรจะถูกใช้ในการดักเก็บค่าแอดเดรสไบต์ไปไว้ที่วงจรแลตช์ภายนอก ซึ่งแอดเดรสไบต์จะใช้ได้ที่ขอบขาลงของสัญญาณ ALE ดังนั้นในช่วงการเขียนข้อมูล ข้อมูลที่จะเขียนปรากฏที่พอร์ต 0 ก่อนหน้าที่สัญญาณ WR จะแอกตีฟ และยังคงอยู่ที่นั่นจนกว่าสัญญาณ WR จะหยุดแอกตีฟ ส่วนในช่วงการอ่านข้อมูลไบต์ที่เข้ามาจะถูกรับเข้าที่พอร์ต 0 ก่อนหน้าที่สัญญาณ Read Strobe จะหยุดแอกตีฟ

ระหว่างการติดต่อหน่วยความจำภายนอกใด ๆ ซีพียูจะทำการเขียนค่า 0FFH ไปที่พอร์ต 0 แลตช์ (special function register) จึงไปลบค่าอะไรก็ตามที่มีอยู่ที่รีจิสเตอร์ใช้งานเฉพาะของพอร์ต 0 และถ้าผู้ใช้เขียนค่าไปที่พอร์ต 0 ระหว่างที่มีการเฟตช์คำสั่งจากหน่วยความจำภายนอก จะทำให้ค่าข้อมูลที่เป็นรหัสคำสั่งที่ได้มาผิดพลาด ดังนั้นจึงไม่ควรเขียนค่าไปที่พอร์ต 0 ถ้ามีการใช้โปรแกรมที่เก็บไว้ในหน่วยความจำสำหรับเก็บโปรแกรมภายนอก

หน่วยความจำสำหรับเก็บโปรแกรมภายนอกสามารถเข้าถึงได้ภายใต้เงื่อนไข 2 ข้อต่อไปนี้

1. เมื่อใดก็ตามที่สัญญาณ EA นั้นแอกตีฟ
2. เมื่อใดที่ pc program counter มีค่ามากกว่า 0FFFH (หรือมากกว่า 1FFFH สำหรับ 8052)

และสำหรับ MCS-51 ที่ไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป (ROMLESS version) จำเป็นจะต้องต่อขา EA ลงกราวด์ เพื่อบังคับให้เลือกโปรแกรมที่อยู่ในบริเวณ 4 กิโลไบต์แรก (หรือ 8 กิโลไบต์แรกสำหรับ 8032) จากหน่วยความจำสำหรับเก็บโปรแกรมที่อยู่ภายนอกชิป

เมื่อซีพียูปฏิบัติคำสั่งจากหน่วยความจำสำหรับเก็บโปรแกรมที่อยู่ภายนอกชิปทั้ง 8 บิตของพอร์ต 2 จะถูกใช้ทำหน้าที่เป็นเอาต์พุตและอาจไม่สามารถใช้เป็นพอร์ตอินพุตเอาต์พุตทั่วไป (General purpose I/O) ระหว่างการเฟตช์คำสั่งจากหน่วยความจำสำหรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เก็บโปรแกรมที่อยู่ภายนอกพอร์ต 2 จะส่งค่าแอดเดรสไบต์สูงของตำแหน่งหน่วยความจำที่ได้จากรีจิสเตอร์ใช้งานเฉพาะ PC และในเวลานี้เอาต์พุตไดรเวอร์ของพอร์ต2 ใช้วงจรพูลอัพอย่างเข้มในการส่งค่าของ PC บิตที่มีค่าเป็น 1

2.2 พอร์ตสื่อสารข้อมูลแบบอนุกรมใน MCS-51

MCS-51 มีพอร์ตสำหรับสื่อสารข้อมูลแบบอนุกรมที่สามารถรับและส่งข้อมูลแบบอนุกรมโดยผู้ใช้ไม่จำเป็นต้องต่อชิปที่ทำหน้าที่รับหรือส่งข้อมูลแบบอนุกรมโดยเฉพาะเพิ่มแต่อย่างใดเลย การนำ MCS-51 ไปประยุกต์ใช้งานที่ต้องมีการติดต่อสื่อสารข้อมูลแบบอนุกรมกับวงจรภายนอกอื่น ๆ จึงทำได้สะดวกและมีความคล่องตัวสูงมาก

พอร์ตสื่อสารข้อมูลแบบอนุกรมที่มีใน MCS-51 สามารถทำงานได้ในแบบ full duplex หมายความว่า MCS-51 สามารถรับและส่งข้อมูลได้พร้อม ๆ กัน โดยในการรับข้อมูลจะมีการบัฟเฟอร์ข้อมูลให้ด้วย จึงทำให้ MCS-51 สามารถกำหนดการรับข้อมูลไบต์ที่สองซึ่งถูกส่งตามเข้ามาก่อนที่ไบต์แรกที่ได้รับเข้ามาจะถูกอ่าน จากรีจิสเตอร์ใช้งานเฉพาะที่ใช้สำหรับรับข้อมูล (receive register) เพื่อนำไปเก็บไว้ในหน่วยความจำต่อไป (หากไบต์แรกยังไม่ถูกอ่านเมื่อได้รับไบต์ที่สองเรียบร้อยแล้วข้อมูลจะหายไปหนึ่งไบต์)

พอร์ตสื่อสารข้อมูลแบบอนุกรมใน MCS-51 ประกอบด้วยรีจิสเตอร์ขนาด 8 บิต จำนวนสองตัวแต่ละตัวมีชื่อเรียกตามหน้าที่ดังนี้คือ

- รีจิสเตอร์สำหรับรับข้อมูลใช้รับข้อมูลที่ส่งเข้ามาจากภายนอก
- รีจีสเตอร์สำหรับส่งข้อมูล (transmit register) ใช้ส่งข้อมูลจาก MCS-51 ออกไปภายนอก

รีจิสเตอร์ทั้งสองมีตำแหน่งเดียวกันในรีจิสเตอร์ใช้งานเฉพาะ คือ ตรงกับตำแหน่งของรีจิสเตอร์ใช้งานเฉพาะ SBUF (ตำแหน่ง 99H) ในหน่วยความจำสำหรับเก็บข้อมูลภายในชิปที่ใช้เป็นรีจิสเตอร์ใช้งานเฉพาะ การเข้าถึงข้อมูลในรีจิสเตอร์แต่ละตัว MCS-51 จะทราบเองว่าผู้ใช้ต้องการติดต่อกับรีจิสเตอร์ตัวใดโดยตรงสอบจากรหัสคำสั่ง ทั้งนี้เพราะ

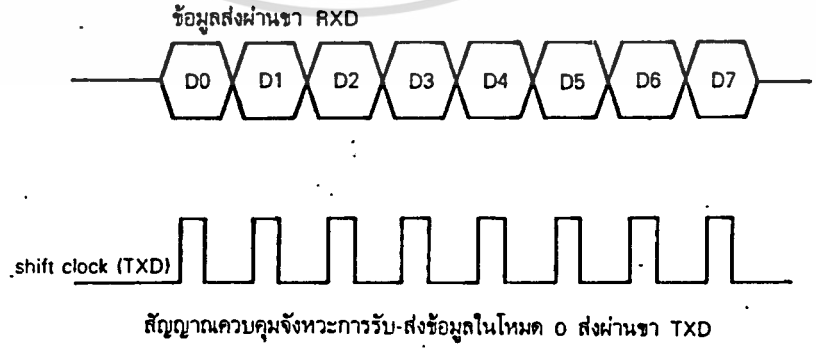
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเขียนข้อมูลไว้ในรีจิสเตอร์ใช้งานเฉพาะ SBUF หมายถึงการโหลดข้อมูลไปที่รีจิสเตอร์สำหรับส่งข้อมูลเพื่อส่งข้อมูลออกไปภายนอก ส่วนการอ่านข้อมูลจากรีจิสเตอร์ใช้งานเฉพาะ SBUF จะหมายถึงนำค่าที่รับเข้ามาได้จากภายนอกที่เก็บไว้ในรีจิสเตอร์สำหรับข้อมูลมาใช้งาน

การใช้งานพอร์ตสื่อสารข้อมูลแบบอนุการใน MCS-51 มีความสะดวกและคล่องตัวสูง ทั้งนี้เนื่องจากผู้ใช้สามารถกำหนดการทำงานที่แตกต่างกันได้ถึง 4 ประเภท โดยสามารถกำหนดได้จากค่าบิตในรีจีสเตอร์ใช้งานเฉพาะ SCON ดังแสดงในรูปที่ 2.6 การใช้งานที่แตกต่างกัน 4 ประเภทนี้มีจุดประสงค์เพื่อความคล่องตัวในการรับหรือส่งข้อมูลแบบอนุกรมแต่ละประเภทดังนี้

2.2.1 การทำงานพอร์ตสื่อสารข้อมูลแบบอนุกรมโหมด 0

โหมด 0 การทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมในโหมด 0 ขา RXD จะใช้สำหรับรับและส่งข้อมูล ส่วนขา TXD มีไว้เพื่อใช้สร้างสัญญาณ shift clock เพื่อกำหนดจังหวะในการรับและส่งข้อมูล (ข้อมูลจะถูกรับหรือส่งตามจังหวะของสัญญาณ shift clock) ในโหมดนี้การรับส่งข้อมูลจะเป็นแบบ 8 บิต (บิตข้อมูล 8 บิต) โดยเริ่มรับและส่งบิตต่ำสุดก่อน (LSB first) อัตราการรับส่งข้อมูลในการทำงานโหมด 0 ถูกกำหนดไว้ที่ 1/12 ของความถี่ออสซิลเลเตอร์ที่ใช้ การทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมในโหมด 0 จะไม่มีบิตเริ่มต้นของข้อมูล (start bit) และบิตสิ้นสุดของข้อมูล (stop bit) เพราะจังหวะการรับและส่งข้อมูลถูกกำหนดจากสัญญาณ shift clock แล้ว



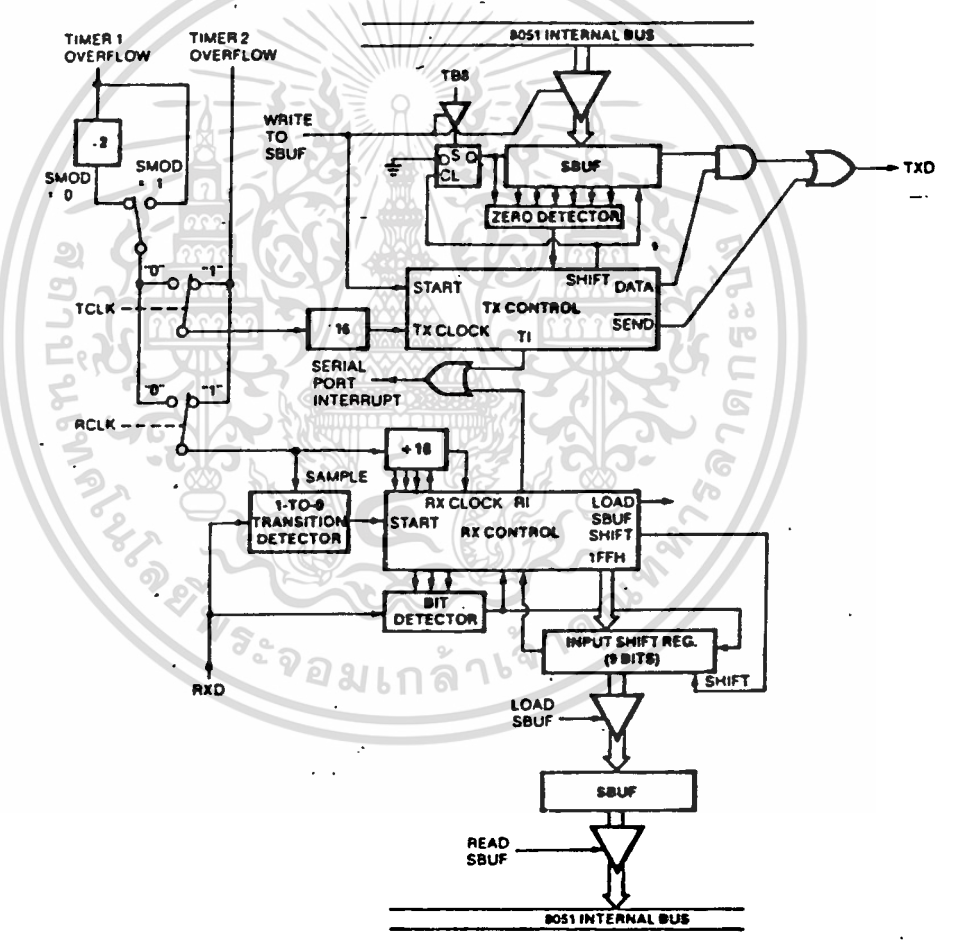
สัญญาณควบคุมจังหวะการรับ-ส่งข้อมูลในโหมด 0 ส่งผ่านขา TXD

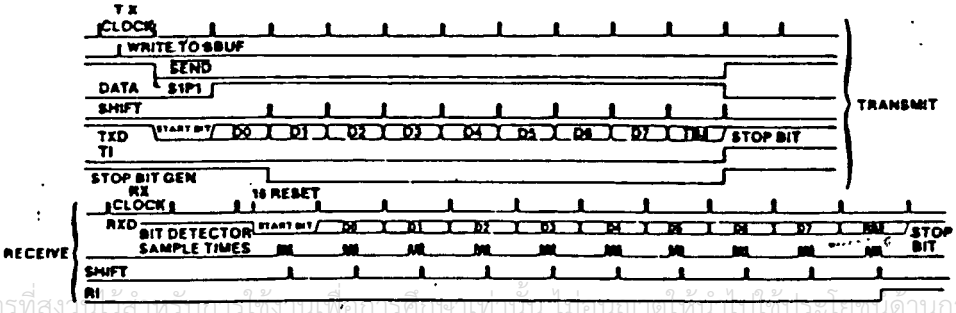
รูปที่ 2.6 แสดงข้อมูลที่รับและส่งในการทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมในโหมด 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมในโหมด 0 ข้อมูลถูกรับเข้าและส่งออกผ่านทางขา RXD ส่วนขา TXD ใช้เป็นตัวสร้าง shift clock เพื่อกำหนดจังหวะสำหรับการรับและส่งข้อมูลกับวงจรภายนอก ในการทำงานโหมด 0 ข้อมูลถูกรับและส่งโดยเริ่มจากบิตเริ่มต้นของข้อมูล (0) ตามด้วยบิตข้อมูลจำนวน 8 บิตที่เริ่มด้วยบิตต่ำสุดก่อน (LSB first) ดังแสดงในรูปที่ 5.1 ค่า baud rate ในการทำงานโหมดนี้จะคงที่เท่ากับ 1/12 ของความถี่ออสซิลเลเตอร์ที่ใช้

โครงสร้างการทำงานคร่าว ๆ ของพอร์ตสื่อสารข้อมูลแบบอนุกรมในโหมด 0 รวมทั้งแผนผังเวลา (timing diagram) ที่เกี่ยวข้องสำหรับการรับและส่งข้อมูล มีดังแสดงในรูปที่ 2.7 การทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมโหมด 0





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.7 การส่งข้อมูลเริ่มจากคำสั่งใด ๆ ที่มีการใช้รีจิสเตอร์ใช้งานเฉพาะ SBUF เป็นรีจิสเตอร์ปลายทาง (destination register) สัญญาณ write to SBUF ที่เห็นในรูปจะเกิดขึ้นขณะสเตท 6 เฟส 2 ของไซเกิลของคำสั่งที่มีการโหลดข้อมูลไปยังรีจิสเตอร์ SBUF ซึ่งจะทำให้มีการโหลดค่า 1 ไปยัง D-flipflop ซึ่งเป็นบิตที่ 9 ของรีจิสเตอร์สำหรับส่งข้อมูล (transmit shift register) ที่ประกอบขึ้นมาจากรีจีสเตอร์ SBUF (ขนาด 8 บิต) และ D-filpflop (ขนาด 1 บิต) สัญญาณ write to SBUF ยังถูกส่งไปกระตุ้นให้วงจรควบคุมการส่งข้อมูล (Tx control unit) เริ่มต้นทำงานด้วยแผนผังเวลาภายในที่แสดงรายละเอียดการส่งข้อมูลจะเห็นได้ว่าสัญญาณ write to SBUF ที่สเตท 6 เฟส 2 กับสัญญาณ SEND ที่แอกตีฟจะเกิดห่างกัน 1 แมชชีนไซเกิลเต็ม

สัญญาณ SEND ที่แอกตีฟ (มีค่าเป็น 1) จะเป็นสัญญาณออกจากวงจรควบคุมการส่งข้อมูลเพื่อไปควบคุมให้รีจิสเตอร์สำหรับส่งข้อมูลทำงานโดยส่งข้อมูลออกไปภายนอกทีละบิต ข้อมูลที่ส่งออกมาจะผ่านออกไปยังขา RXD ซึ่งตรงกับ P3.0 นอกจากนี้สัญญาณ SEND ยังไปควบคุมให้วงจร shift clock สร้างสัญญาณ shift clock ให้ผ่านออกไปทางขา TXD ซึ่งตรงกับ P3.1 โดยสัญญาณ shift clock จะเป็นสัญญาณที่มีค่าเป็น 0 ระหว่าง สเตท 3,4,5 และมีค่าเป็น ระหว่าง สเตท 6,1 และ 2 ของทุกแมชชีนไซเกิล ดังแสดงในรูปที่ 2.7 เมื่อถึงสเตท 6 เฟส 2 ของทุก ๆ แมชชีนไซเกิลซึ่งสัญญาณ SEND ยังแอกตีฟอยู่ ข้อมูลในรีจิสเตอร์สำหรับส่งข้อมูลจะถูกเลื่อนออกไปทางขวา 1 ตำแหน่ง ทำให้ข้อมูลถูกส่งผ่านออกมาภายนอกทางขา RXD ทีละบิต โดยเริ่มจากบิตต่ำสุดก่อน

ขณะที่ข้อมูลถูกเลื่อนออกไปทางขวาทีละบิต 0 จะถูกนำมาแทนที่ตำแหน่งซึ่งถูกเลื่อนออกไปโดยจะเลื่อนเข้ามาทำงานด้านซ้ายของรีจิสเตอร์สำหรับส่งข้อมูล (ค่าศูนย์ได้มาจาก D-flip flop ที่มีอินพุตต่อลงกราวด์) การส่งข้อมูลจะดำเนินไปเรื่อย ๆ จนกระทั่งข้อมูลที่เป็นบิตสูงสุด (MSB) ที่ถูกโหลดไว้ในรีจิสเตอร์ SBUF ถูกเลื่อนไปอยู่ที่ตำแหน่งเอาต์พุต (output position) ของรีจิสเตอร์ สำหรับส่งข้อมูลพร้อม ๆ กับ 1 ที่ถูกโหลดไปที่ตำแหน่งที่ 9 ในตอนเริ่มต้นการส่งข้อมูลที่กล่าวในตอนแรกจะอยู่ถัดจากบิตสูงสุดมาทางซ้าย และทุกตำแหน่งทางซ้ายของบิตที่ 9 มีค่าเป็นศูนย์หมด (มีศูนย์ทางซ้ายรวม 7 ตัว) ในสภาวะเช่นนี้จะส่งผลให้วงจรตรวจจับ 0 (Zero dectector) เริ่มต้นทำงาน (ตรวจจับเพียงแต่ 7 บิต) โดยส่งสัญญาณไปบอกวงจรควบคุมการส่งข้อมูลให้ทำการเลื่อนข้อมูลครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุดท้ายอีก 1 ครั้ง เพื่อส่งบิตสูงสุดออกไป ดังนั้นบิตที่ 9 จะไม่ถูกส่งตามออกมา โดยจะยังคงอยู่ที่ตำแหน่งเอาต์พุตของรีจิสเตอร์สำหรับส่งข้อมูลเท่านั้น จากนั้นการส่งข้อมูลจะสิ้นสุดลงโดยวงจรส่วนควบคุมการส่งข้อมูลจะบังคับให้สัญญาณ SEND หยุดแอกตีฟ เมื่อข้อมูลแต่ละบิตในรีจิสเตอร์ SBUF ถูกส่งออกไปภายนอกเรียบร้อยแล้ว บิต T1 จะถูกเซต เหตุการณ์ที่เกิดขึ้นขณะส่งข้อมูลเสร็จจะเกิดขึ้นที่ขณะสเตท 1 เฟส 1 ของแมชชีนไซเกิลที่ 10 หลังจากสัญญาณ write to SBUF เกิดขึ้น

การรับข้อมูลในโหมด 0 เกิดขึ้นได้ก็ต่อเมื่อบิต REN = 1 และ Ri = 0 (พิจารณาจากรูปที่ 2.7 โดยขณะสเตท 6 เฟส 2 ของแมชชีนไซเกิลถัดไป (ถัดจากไซเกิลที่บิต REN = 1 และ Ri = 0) วงจรควบคุมการรับข้อมูล RX control block) จะเริ่มเขียนข้อมูล 1111110B ไปที่รีจิสเตอร์สำหรับรับข้อมูล (receive shift register) และในสัญญาณนาฬิกาเฟสถัดไป receive จากวงจรควบคุมการรับข้อมูลจะเริ่มแอกตีฟ สัญญาณ receive ที่ได้จากวงจรควบคุมการรับข้อมูลจะทำให้วงจร shift clock ส่งสัญญาณ shoft clock ออกไปที่ขา TXD โดยควบคุมผ่านเกตแนนด์ สัญญาณ shift clock ที่ได้จะมีการเปลี่ยนสถานะทุกสเตท 3 เฟส 1 และสเตท 6 เฟส 1 ของทุก ๆ แมชชีนในช่วงการรับข้อมูลโดยขณะสเตท 6 เฟส 2 ของแต่ละแมชชีนไซเกิลซึ่งสัญญาณ Receive ยังคงแอกตีฟอยู่ ข้อมูลในรีจิสเตอร์สำหรับรับข้อมูลจะถูกเลื่อนไปทางซ้าย 1 ตำแหน่ง เนื่องจากข้อมูลที่รับได้จากภายนอกถูกเลื่อนเข้ามาจากทางขวาของรีจิสเตอร์สำหรับส่งข้อมูล ค่าของข้อมูลที่เข้ามาจากทางขวาจะเป็นข้อมูลซึ่งได้รับการตรวจสอบที่ขา RXD ในช่วงสเตท 5 เฟส 2 ของแมชชีนไซเกิลเดียวกัน

ขณะที่ข้อมูลถูกเลื่อนเข้ามาจากทางขวาทีละบิต ค่าของ 1 ซึ่งถูกโหลดไปไว้ใน รีจิสเตอร์สำหรับรับข้อมูลในตอนแรกจะถูกเลื่อนออกไปทางซ้าย (shift out) และเมื่อ 0 ซึ่งถูกโหลดไปไว้ที่บิตต่ำสุดตั้งแต่ตอนแรก (1111110) ถูกเลื่อนมาอยู่ในตำแหน่งซ้ายสุดของรีจิสเตอร์ตัวนี้จะทำให้เกิดสัญญาณลงไปบอกวงจรส่วนควบคุม การรับข้อมูลให้ทราบว่าขณะนี้ข้อมูลถูกรับเข้ามา 7 บิตแล้ว เมื่อวงจรควบคุมการรับข้อมูลได้รับสัญญาณนี้ก็จะควบคุมให้มีการรับข้อมูลอีก 1 บิต โดยการเลื่อนบิตมาทางซ้ายอีก 1 ครั้งและส่งสัญญาณ load SBUF เพื่อส่งข้อมูลที่รับมาได้เข้าไปไว้ในรีจิสเตอร์ SBUF เหตุการณ์ที่เกิดขึ้นขณะรับข้อมูลเสร็จจะเกิดขึ้นในช่วงสเตท 1 เฟส 1 ของแมชชีนไซเกิลที่ 10 หลังจากที่มีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

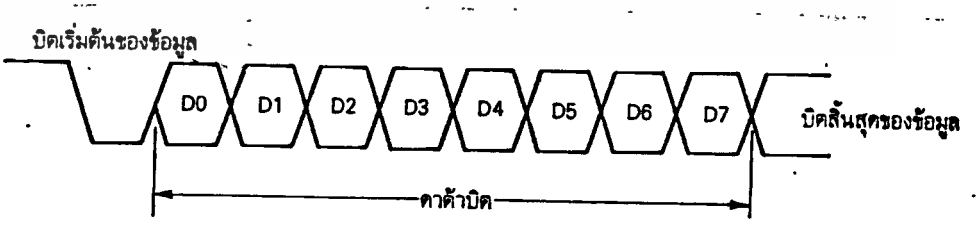
โหลดค่าไปที่รีจิสเตอร์ SCON ซึ่งไปเคลียร์บิต R1 เมื่อการรับข้อมูลสิ้นสุดลงแล้วสัญญาณ receive จะหยุดแอกตีฟและบิต R1 จะถูกเซต

ในการส่งและการรับข้อมูลของพอร์ตสื่อสารข้อมูลแบบอนุกรมในโหมด 0 นี้จะเห็นว่าจังหวะในการรับหรือส่งข้อมูลอยู่กับช่วงเวลาแต่ละแมชชีนไซเกิลของซีพียู เพราะข้อมูลจะถูกรับเข้ามาหรือส่งออกไปทุก ๆ แมชชีนไซเกิล จึงทำให้ค่า baud rate ของพอร์ตสื่อสารข้อมูลแบบอนุกรมในโหมด 0 มีค่าเท่ากับ 1/12 ของความถี่ออสซิลเลเตอร์ (เท่ากับความถี่ของแมชชีนไซเกิล)

สัญญาณ shift clock ที่เกิดขึ้นขณะส่งข้อมูลในโหมด 0 จะเป็นสัญญาณที่ MCS-51 สร้างขึ้นเพื่อควบคุมจังหวะการรับข้อมูลของวงจรภายนอก (MCS-51 ส่งข้อมูลให้วงจรภายนอก) ส่วนสัญญาณ shift clock ที่เกิดขึ้นขณะรับข้อมูลในโหมดนี้จะเป็นสัญญาณที่ MCS-51 สร้างขึ้นเพื่อใช้ควบคุมจังหวะการส่งข้อมูลของวงจรภายนอก (MCS-51 รับข้อมูลจากวงจรภายนอก)

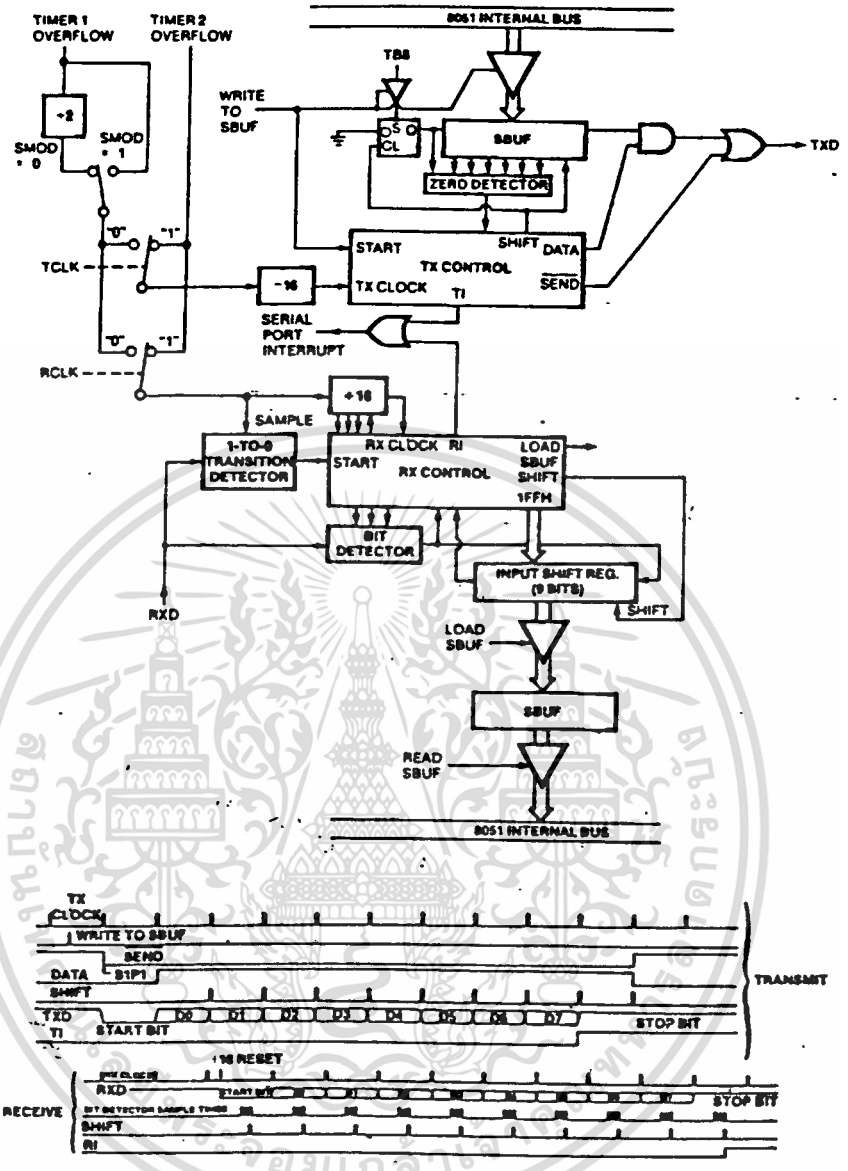
2.2.2 การทำงานพอร์ตสื่อสารข้อมูลแบบอนุกรมโหมด 1

โหมด 1 การทำงานแบบที่สองหรือการทำงานโหมด 1 นี้ มีการรับและส่งข้อมูลครั้งละ 10 บิต ข้อมูลจะส่งออกไปภายนอกผ่านทางขา TXD และรับข้อมูลเข้ามาทางขา RXD ข้อมูลทั้ง 10 บิต ประกอบด้วยบิตเริ่มต้นของข้อมูล 1 บิต (มีค่าเป็น 0 เสมอ) ในขณะทำการรับข้อมูล ค่าในบิตสิ้นสุดของข้อมูลที่รับได้จะไปอยู่ในบิต RB8 ของรีจิสเตอร์ใช้งานเฉพาะ SCON อัตราเร็วในการรับหรือส่งข้อมูลของพอร์ตสื่อสารข้อมูลแบบอนุกรมในโหมดนี้สามารถเปลี่ยนแปลงได้



รูปที่ 2.8 แสดงข้อมูลที่รับและส่งในการทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมโหมด 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 การทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมโหมด 1 (บิต TCLK,RCLK และไทม์เมอร์ 2 มีเฉพาะใน 8052/8032 เท่านั้น)

การทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมในโหมด 1 ข้อมูล 10 บิตถูกส่งออกไปภายนอกผ่านขา TXD และรับเข้ามาจากภายนอกผ่านทางขา RXD ข้อมูลที่รับหรือส่งจะประกอบด้วยบิตเริ่มต้นของข้อมูล 1 บิต (0) บิตข้อมูล 8 บิตที่เริ่มรับหรือส่งด้วยบิตต่ำสุดก่อน (LSB first) ตามด้วยบิตสิ้นสุดของข้อมูลอีก 1 บิต(1) ดังแสดงในรูปที่ 2.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการรับข้อมูล บิตสิ้นสุดของข้อมูลจะถูกนำไปไว้ในบิต RB8 ของรีจิสเตอร์ใช้งานเฉพาะ SCON ค่า baud rate ถูกกำหนดโดยอัตราการเกิด overflow ของไทม์เมอร์ 1 (timer 1 overflow rate) ส่วนใน 8052 ซึ่งมีไทม์เมอร์ 2 เพิ่มขึ้นมา ค่า baud rate สามารถถูกกำหนดโดยอัตราการเกิด overflow ของไทม์เมอร์ 1 หรือไทม์เมอร์ 2 อย่างใดอย่างหนึ่งหรือทั้ง 2 อย่าง ใช้ตัวหนึ่งสำหรับกำหนด baud rate ในการส่งข้อมูล และอีกตัวหนึ่งสำหรับการรับข้อมูล) โดยใช้บิต RCLK และบิต TCLX เป็นตัวเลือก

การส่งข้อมูลจะเริ่มต้นโดยคำสั่งใด ๆ ที่ใช้รีจิสเตอร์ SBUF เป็นรีจิสเตอร์ปลายทาง สัญญาณ write to SBUF ที่เกิดขึ้นจะทำให้ 1 ถูกโหลดไปไว้ที่ D-Flipflop ซึ่งเป็นตำแหน่งบิตที่ 9 ของรีจิสเตอร์สำหรับส่งข้อมูล (transmit shift register) ซึ่งประกอบขึ้นจากรีจิสเตอร์ SBUF (8 บิต) และ D-flipflow (1 บิต) ตำแหน่งบิตที่ 9 หรือ D-flipflop จะมีอินพุตต่อลงกราวด์ และขา S ต่อกับสัญญาณ write to SBUF นอกจากนี้สัญญาณ write to SBUF ยังถูกส่งไปบอกให้วงจรควบคุมการส่งข้อมูลทราบว่าขณะนี้กำลังมีความต้องการที่จะส่งข้อมูล การส่งข้อมูลจริง ๆ จะเริ่มต้นในช่วงสเตท 1 เฟส 1 ของแมชชีนไซเกิลที่ถัดจากการเกิด overflow ครั้งถัดไปในเคาน์เตอร์ที่ถูกหารด้วย 16 ในรูปที่ 2.10 (เคาน์เตอร์จะถูกเพิ่มค่าเมื่อนับครบ 16 ครั้ง) ดังนั้นช่วงจังหวะการส่งข้อมูลแต่ละบิตจะสอดคล้อง (synchronized) กับสัญญาณที่ได้จากเคาน์เตอร์ที่ถูกหารด้วย 16 ไม่ใช่กับสัญญาณ write to SBUF

การส่งข้อมูลเริ่มต้นด้วยสัญญาณ SEND (แอกตีฟที่สถานะลอจิก 0) ที่ถูกกระตุ้นให้เริ่มทำงาน โดยเริ่มบิตเริ่มต้นของข้อมูล (0)ไปยังขา TXD เมื่อเวลาของการส่งบิตแรกผ่านไป สัญญาณ data จะเริ่มแอกตีฟทำให้ข้อมูลถูกส่งออกมาทีละบิตจากรีจิสเตอร์สำหรับส่งข้อมูล (ควบคุมผ่านเกตแอนด์) ข้อมูลแต่ละบิตส่งออกมาจากตำแหน่งเอาต์พุตของรีจิสเตอร์สำหรับส่งข้อมูลเพื่อส่งผ่านออกไปผ่านขา TXD สัญญาณ shift จะสร้างพัลส์ทีละลูกภายหลังการส่งข้อมูลผ่านไป 1 บิต

ขณะที่บิตของข้อมูลถูกเลื่อนออกไปทางขวา ค่า 0 จะถูกเลื่อนเข้ามาทางซ้าย (จาก D-flipflop) เมื่อบิตสูงสุดของข้อมูลอยู่ที่ตำแหน่งเอาต์พุตของรีจิสเตอร์สำหรับส่งข้อมูลแล้ว ค่า 1 ซึ่งถูกโหลดไปไว้ในตำแหน่งที่ 9 ในตอนแรกจะอยู่ถัดจากบิตสูงสุดไปทางซ้าย ส่วนตำแหน่งอื่น ๆ ทางซ้ายมีค่าเป็น 0 หมด (ได้จาก D-flipflop ที่มีอินพุตต่อลงกราวด์) ในสภาวะเช่นนี้วงจรตรวจจับศูนย์จะเริ่มต้นทำงาน (ตรวจจับเพียง 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต) โดยการส่งสัญญาณไปบอกวงจรควบคุมการส่งข้อมูลให้ทำการเลื่อนข้อมูลเป็นครั้งสุดท้ายอีก 1 ครั้ง แล้วจึงหยุดการส่งข้อมูลและบังคับให้สัญญาณ SEND หยุดแอกตีฟ พร้อมทั้งเซตบิต TI เหตุการณ์ที่เกิดขึ้นขณะการส่งข้อมูลเสร็จสิ้นลงจะเกิดขึ้นในช่วงที่เกิด overflow ครั้งที่ 10 ของเคาน์เตอร์ที่หาร 16 หลังจากสัญญาณ write to SBUF เกิดขึ้น

การรับข้อมูลเริ่มต้นขึ้นเมื่อ MCS-51 ตรวจพบการเปลี่ยนสถานะจาก 1 เป็น 0 ที่ขา TXD (ตรวจพบบิตเริ่มต้นของข้อมูล) ซึ่งการตรวจสอบสถานะนี้ MCS-51 จะตรวจสอบด้วยความถี่ 16 เท่าของค่า baud rate ในขณะนั้น เมื่อการเปลี่ยนสถานะของสัญญาณถูกตรวจพบ เคาน์เตอร์หาร 16 จะถูกรีเซตในทันที และข้อมูล 1FFH จะถูกโหลดไปไว้ในรีจีสเตอร์สำหรับรับข้อมูลซึ่งมีขนาด 9 บิต การรีเซตเคาน์เตอร์หาร 16 ก็เพื่อตั้งอัตราการ overflow ให้เริ่มตรงกับการเริ่มต้นรับข้อมูลของบิตต่อไปที่จะรับเข้ามา

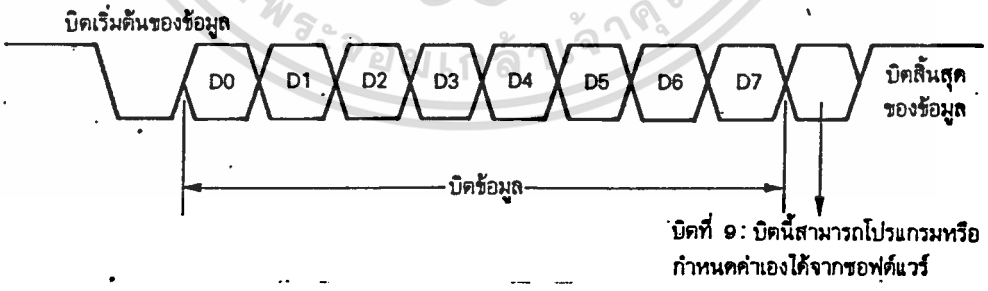
จากรูปที่ 2.9 เคาน์เตอร์จะถูกเพิ่มค่าอีก 1 ทุกๆ เมื่อไทม์เมอร์ 1 หรือไทม์เมอร์ 2 ที่ใช้เป็นตัวกำหนดค่า baud rate เกิด overflow ครบ 16 ครั้ง ช่วงเวลาในการรับข้อมูลแต่ละบิตจะนำเอาอัตราการเกิด overflow มาเป็นตัวกำหนด โดยช่วงเวลาในการรับข้อมูลแต่ละบิตจะถูกแบ่งออกเป็น 16 สเตทย่อย วงจรตรวจสอบข้อมูลแต่ละบิต (bit detector) จะทำการตรวจสอบค่าข้อมูลแต่ละบิตทุกๆ สเตทที่ 7,8,9 ในช่วงเวลาของการรับข้อมูลแต่ละบิต (bit time) วงจรตรวจสอบบิตข้อมูลจะตรวจสอบค่าสถานะที่ขา RXD ซึ่งมีข้อมูลรออยู่ ค่าที่รับเข้ามาจะถือว่าเป็นบิตข้อมูลที่ถูกต้องก็เมื่อตรวจสอบพบว่าเป็นค่าเดียวกันอย่างน้อย 2 ครั้งจากการตรวจสอบทั้งหมด 3 ครั้ง (ตรวจสอบสเตทที่ 7,8,9) การทำเช่นนี้เพื่อป้องการสัญญาณรบกวน (noice rejection) ที่อาจเกิดขึ้นในระหว่างการรับหรือส่งข้อมูล แต่ในช่วงของการรับบิตแรก (บิตเริ่มต้นของข้อมูล) หากค่าที่ถูกตรวจสอบได้ในสเตท 7,8,9 ปรากฏว่าไม่เป็นศูนย์ วงจรที่ทำหน้าที่รับข้อมูลจะถูกรีเซตและระบบจะกลับไปเริ่มตรวจหาการเปลี่ยนสถานะจาก 1 เป็น 0 ที่ขา RXD ต่อไป (กลับไปรอรับบิตเริ่มต้นของข้อมูลใหม่) ที่ต้องทำเช่นนี้ก็เพื่อยกเลิกการรับข้อมูลหากบิตเริ่มต้นของข้อมูลผิดพลาด แต่ถ้าบิตเริ่มต้นของข้อมูลถูกตรวจสอบแล้วปรากฏว่ามีค่าเป็นศูนย์ การรับข้อมูลบิตต่อไปจะเริ่มต้นทันทีโดยเริ่มต้นของข้อมูลจะถูกเลื่อนไปไว้ในรีจิสเตอร์สำหรับรับข้อมูล และบิตข้อมูลที่เหลือจะถูกรับตามมาเก็บไว้ในรีจิสเตอร์นี้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขณะที่บิตข้อมูลถูกรับเข้ามาทางขา บิตข้อมูลที่มีค่า 1 (ได้จากค่า 1FFH ที่ถูกโหลดไว้ก่อน) จะถูกเลื่อนออกไปทางซ้ายจนกระทั่งเมื่อบิตเริ่มต้นของข้อมูล (0) เลื่อนไปถึงตำแหน่งซ้านสุดในรีจิสเตอร์ สำหรับรับข้อมูล (ขนาด 9 บิต) มันจะส่งสัญญาณไปบอกวงจรควบคุมการรับข้อมูล (RX control block) ให้เลื่อนข้อมูลครั้งสุดท้ายอีก 1 ครั้ง เพื่อรับบิตสิ้นสุดของข้อมูล โหลดค่าในรีจิสเตอร์ใช้งานเฉพาะ SBUF ด้วยข้อมูลในรีจิสเตอร์สำหรับรับข้อมูล โหลดบิต RB8 ด้วยค่าบิตสิ้นสุดของข้อมูลและเซตบิต RI เป็นลำดับสุดท้ายสัญญาณที่จะโหลดข้อมูลไปไว้ในรีจิสเตอร์ SBUF และรับบิตสิ้นสุดของข้อมูลไปไว้ในบิต RB8 รวมทั้งการเซตบิต RI จะถูกสร้างขึ้นก็ต่อเมื่อเงื่อนไขต่อไปนี้เป็นจริงขณะสัญญาณ shilft pulse ลูกสุดท้ายถูกสร้างขึ้น

2.2.3 การทำงานพอร์ตสื่อสารข้อมูลแบบอนุกรม โหมด 2 และ 3

โหมด 2 การทำงานแบบที่สาม หรือการทำงานโหมด 2 จะมีการรับและส่งข้อมูลครั้งละ 11 บิต ข้อมูลจะถูกส่งออกภายนอกผ่านทางขา TXD และรับเข้ามาผ่านทางขา RXD ข้อมูลที่รับและส่งทั้ง 11 บิต ประกอบด้วยบิตเริ่มต้นของข้อมูล 1 บิต (มีค่าเป็น 0 เสมอ) บิตข้อมูล 11 บิต (รับหรือส่งบิตต่ำสุดก่อน) ตามด้วยบิตที่ 9 (ต่อจากบิตข้อมูลที่บิตสุดท้าย) ซึ่งเป็นบิตที่สามารถกำหนดให้มีค่าเป็นศูนย์หรือหนึ่งได้ (programmable 9th data bit) และบิตสุดท้ายคือบิตสิ้นสุดของข้อมูล (มีค่าเป็น 1 เสมอ) ดังนั้นจำนวนบิตที่รับส่งทั้งหมด 11 บิต จะประกอบด้วยบิตต่าง ๆ ดังนี้



รูปที่ 2.10 แสดงข้อมูลที่รับและส่งในการทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมโหมด 2,3

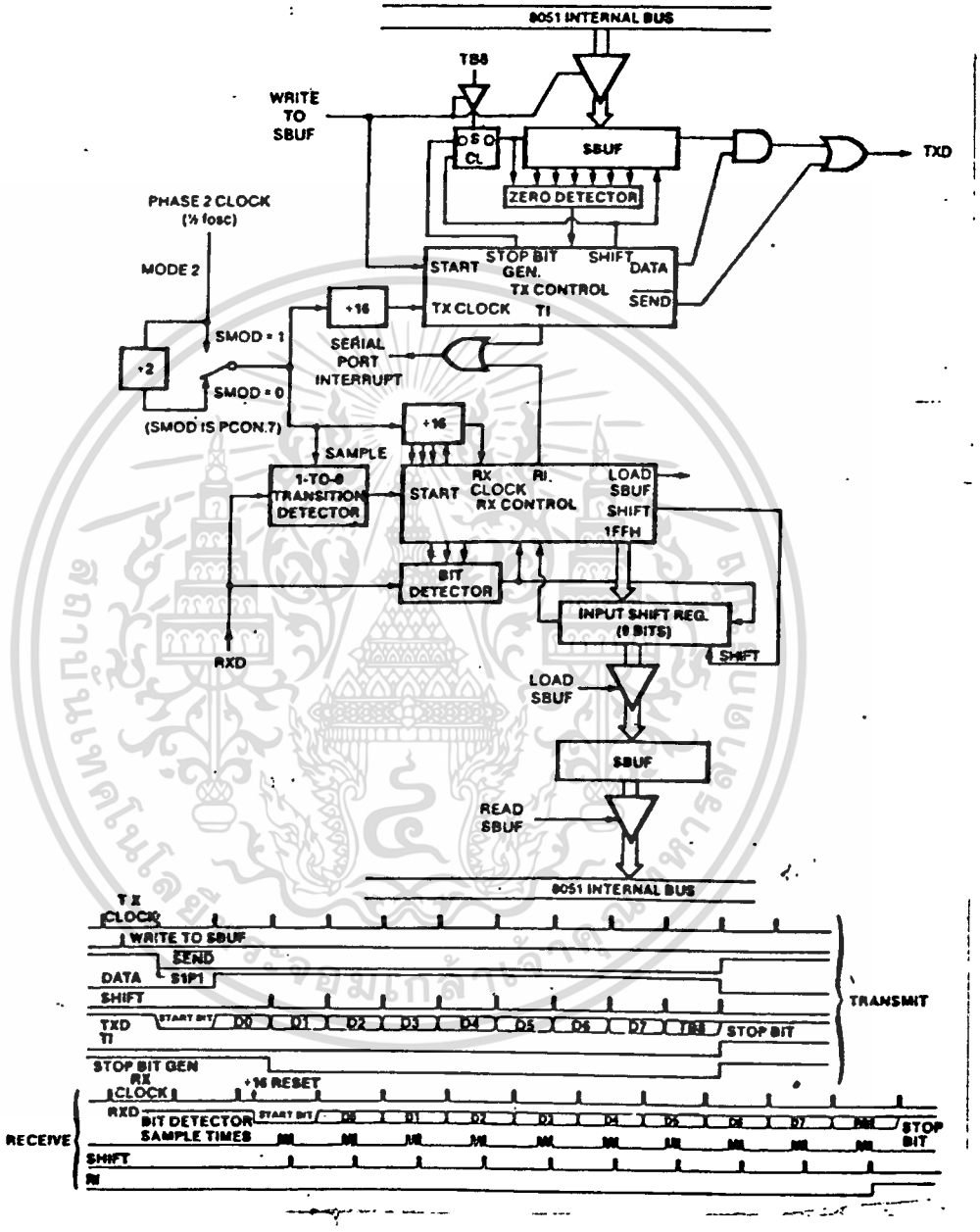
ในขณะทำการส่งข้อมูล บิตที่ 9 จะได้จากค่าในบิต TB8 ของรีจิสเตอร์ใช้งานเฉพาะSCON ดังแสดงในรูปที่ 2.7 บิตนี้สามารถถูกกำหนดให้มีค่าเป็น 0 หรือ 1 อย่างก็ได้ ส่วนใหญ่ในการใช้งานจริงมักจะใช้บิตนี้สำหรับตรวจสอบความถูกต้องของข้อมูลที่รับหรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่ง (parity bit) โดยจะนำบิต P (parity) ในรีจิสเตอร์ PSW ไปไว้ในบิต TB8 ส่วนในขณะรับข้อมูลบิตที่ 9 จะไปปรากฏอยู่ในบิต RB8 ของรีจิสเตอร์ SCON โดยไม่สนใจบิตสิ้นสุดของข้อมูล ค่าอัตราเร็วในการรับหรือส่งข้อมูลโหมดนี้ถูกกำหนดไว้ที่ 1/32 หรือ 1/64 ของความถี่ออสซิลเลเตอร์ที่ใช้

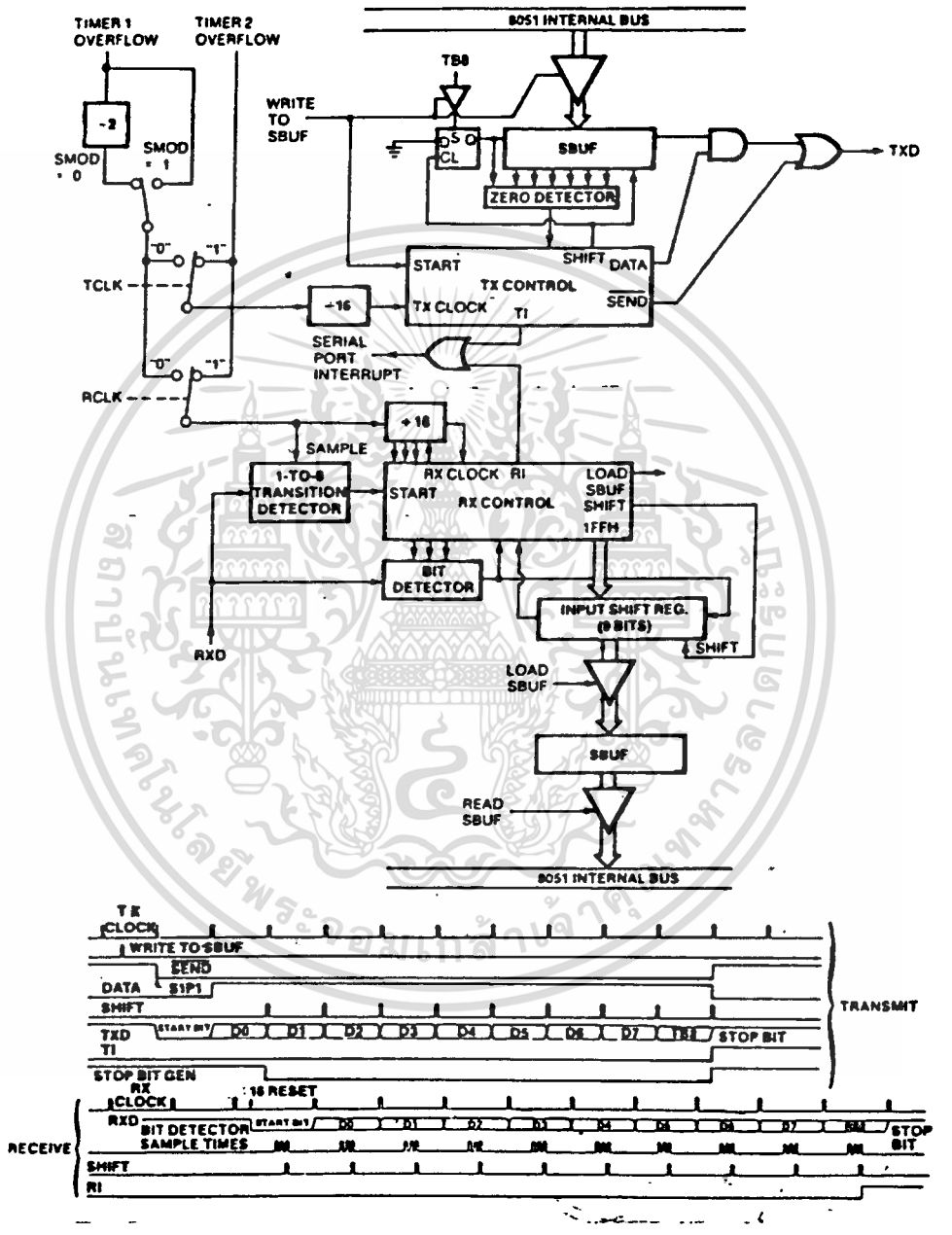
โหมด 3 การทำงานของพอร์ตสื่อสารข้อมูลอนุกรมแบบสุดท้าย คือการทำงานในโหมด 3 ในการทำงานโหมดนี้ ข้อมูลจำนวน 11 บิต ถูกส่งผ่านขา TXD และถูกรับเข้ามาทางขา RXD ข้อมูลทั้ง 11 บิตประกอบด้วยบิตเริ่มต้นของข้อมูล 1 บิต (เป็น 0 เสมอ) บิตข้อมูล 8 บิต (รับและส่งบิตต่ำสุดก่อน) ตามด้วยบิตที่ 9 ซึ่งเป็นบิตที่สามารถกำหนดค่าได้เหมือนในโหมด 2 (programmable 9th bit) และบิตสุดท้ายคือบิตสิ้นสุดของข้อมูล (เป็น 1 เสมอ) อัตราเร็วในการรับหรือส่งข้อมูลสามารถเปลี่ยนแปลงได้ ดังนั้นจะเห็นว่าในรูปแบบการรับส่งข้อมูลในโหมด 3 จะเหมือนกับโหมด 2 ทุกอย่าง แต่ในโหมดนี้สามารถกำหนดค่าอัตราเร็วในการรับหรือส่งข้อมูลได้ตามต้องการของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 การทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมโหมด 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 การทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมโหมด 3 (บิต TCLK,RCLK รวมทั้งไทม์เมอร์ 2 มีเฉพาะใน 8052/8032 เท่านั้น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บิต Rl ถูกเคลียร์ (Rl = 0) และ

- บิต SM2 = 0 หรือบิตสิ้นสุดของข้อมูล = 1

ถ้าเงื่อนไขอย่างใดอย่างหนึ่งของทั้งสองอย่างนี้ไม่ตรง ข้อมูลที่รับเข้ามาจะหายไปโดยไม่สามารถเรียกคืนได้ แต่ถ้าเงื่อนไขทั้งสองถูกต้องบิตสิ้นสุดของข้อมูลจะถูกนำไปไว้ในบิต RB8 ของรีจิสเตอร์ SCON ส่วนบิตที่เป็นข้อมูลจำนวน 8 บิต (บิตข้อมูล) จะถูกโหลดไปไว้ในรีจิสเตอร์ SBUF และบิต Rl จะถูกเซต เมื่อถึงตอนนี้ไม่ว่าเงื่อนไขข้างต้นจะตรงหรือไม่ ระบบการรับส่งข้อมูลจะถือว่าการรับข้อมูลจำนวน 1 ไบต์ (รับข้อมูลเข้ามา 10 บิต แต่เป็นบิตข้อมูลจริง ๆ เพียง 8 บิต) เสร็จสิ้นลงแล้วและเริ่มรีเซตตัวเองเพื่อกลับไปรอรับการเปลี่ยนสถานะจาก 1 เป็น 0 ที่ขา RXD เพื่อรับข้อมูลไบต์ต่อไป (กลับไปรอรับบิตเริ่มต้นของข้อมูลใหม่)

การทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมในโหลด 2 และ 3 มีการรับและส่งข้อมูลครั้งละ 11 บิต โดยส่งข้อมูลผ่านขา TXD และรับข้อมูลผ่านทางขา RXD ข้อมูลทั้ง 11 บิตจะประกอบไปด้วยบิตเริ่มต้นของข้อมูล 1 บิต (0) บิตข้อมูล 8 บิต (รับและส่งบิตต่ำสุดก่อน) บิตที่ 9 ซึ่งสามารถกำหนดค่าเองได้ 1 บิต (1 programmable 9th bit) และบิตสิ้นสุดของข้อมูล 1 บิต (1) ดังแสดงในรูปที่ 2.10

บิตข้อมูลที่มีเพิ่มขึ้นมาในโหมดนี้คือบิตที่ 9 ซึ่งสามารถกำหนดค่าล่วงหน้าได้ด้วยซอฟต์แวร์ในการส่งข้อมูล บิตที่ 9 (บิต TB8 ในรีจิสเตอร์ SCON) สามารถถูกกำหนดให้มีค่าเป็น 0 หรือ 1 ได้ส่วนในการรับข้อมูล บิตที่ 9 จะถูกนำไปเก็บไว้ที่บิต RB8 ในรีจิสเตอร์ใช้งานเฉพาะ SCON

ในการทำงานโหมด 2 ค่า baud rate สามารถกำหนดให้เป็น 1/32 หรือ 1/64 ของความถี่ออสซิลเลเตอร์ที่ใช้ ส่วนในโหมด 3 ค่า baud rate สามารถแปรค่าได้โดยถูกกำหนดด้วย ไทม์เมอร์ 1 หรือไทม์เมอร์ 2 อย่างใดอย่างหนึ่ง การเลือกให้ไทม์เมอร์ 1 หรือไทม์เมอร์ 2 เป็นตัวกำหนด baud rate ควบคุมได้จากบิต RCLK และ TCLK ในรีจิสเตอร์ใช้งานเฉพาะ T2CON

แผนผังแสดงการทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมในโหมด 2 และ 3 มีดังแสดงในรูปที่ 2.11 และ 2.12 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งเริ่มต้นด้วยคำสั่งใด ๆ ที่ใช้รีจิสเตอร์ใช้งานเฉพาะ SBUF เป็นรีจิสเตอร์ปลายทางสัญญาณ write to SBUF จะนำค่าในบิต TB8 (มีค่า 0 หรือ 1 ตามการกำหนดล่วงหน้าของผู้ใช้) ไปไว้ที่ตำแหน่งบิตที่ 9 ของรีจิสเตอร์สำหรับส่งข้อมูล (บิตที่ 9 ในรีจิสเตอร์สำหรับส่งข้อมูลคือ D-flipflop และส่งสัญญาณไปบอกวงจรควบคุมการส่งข้อมูล (TX control block) ว่าขณะนี้กำลังต้องการส่งข้อมูล การส่งข้อมูลจะเริ่มต้นที่สเตท 1 เฟส 1 ของแมชชีนไซเกิลถัดจากการเกิด overflow ในเคาน์เตอร์หาร 16 ครั้งถัดไป (ดังนั้นช่วงเวลาในการส่งแต่ละบิตจะซิงโครไนซ์กับเคาน์เตอร์หาร 16 ไม่ใช้กับสัญญาณ write to SBUF)

การส่งข้อมูลเริ่มต้นด้วยการกระตุ้นให้สัญญาณ SEND แอกตีฟ ซึ่งจะเริ่มใส่ค่าบิตเริ่มต้นของข้อมูลไปที่ขา TXD เมื่อช่วงเวลาการส่งบิตแรกผ่านไป สัญญาณ DATA จะเริ่มแอกตีฟทำให้ค่าบิตที่ตำแหน่งเอาต์พุตของรีจิสเตอร์สำหรับส่งข้อมูลถูกส่งไปที่ขา TXD และพัลส์ของสัญญาณ shift ลูกแรกเกิดขึ้น (ดูในรูปที่ 2.12) เมื่อเวลาในการส่งผ่านไป 1 บิต พัลส์ของสัญญาณ shift นี้จะไปทำให้ 1 ถูกนำไปไว้ในตำแหน่งบิตที่ 9 (ต่อไปจะกลายเป็นบิตสิ้นสุดของข้อมูล) ของรีจิสเตอร์ สำหรับส่งข้อมูล (ตำแหน่งบิตที 9 คือ D-flipflop) หลังจากนั้นจะมีเพียงศูนย์ที่ถูกใส่เข้าไปในรีจิสเตอร์สำหรับส่งข้อมูล ดังนั้นขณะที่บิตข้อมูลถูกเลื่อนออกไปทางขวา 0 จะถูกเลื่อนเข้าไปทางซ้าย การส่งจะดำเนินไปเรื่อย ๆจนกระทั่งบิต TB8 ซึ่งถูกใส่ไว้ในตำแหน่งบิตที่ 9 ของรีจิสเตอร์สำหรับส่งข้อมูลในตอนแรก ถูกเลื่อนไปอยู่ที่ตำแหน่งเอาต์พุตของรีจิสเตอร์สำหรับส่งข้อมูลและทางด้านซ้ายของบิต TB8 นี้จะมีค่า 1 (บิตสิ้นสุดของข้อมูล) จากการโหลดด้วยพัลส์ของสัญญาณ shift ครั้งแรก ส่วนตำแหน่งอื่นที่เหลือทางด้านซ้ายจะมีค่าเป็นศูนย์หมด สภาวะที่มีศูนย์อยู่ที่ด้านซ้ายของบิตสิ้นสุดของข้อมูลนี้จะถูกตรวจพบได้ด้วยวงจรตรวจจับ 0 (Zero detector) ซึ่งเมื่อตรวจจับได้ก็จะส่งสัญญาณไปบอกวงจรควบคุมการส่งข้อมูลให้ทำการเลื่อนข้อมูลครั้งสุดท้าย เหตุการณ์ที่เกิดขึ้นหลังจากตรวจจับศูนย์ได้ 7 บิตนี้ จะเกิดขึ้นขณะที่มีการ overflow ครั้งที่ 11 ของเคาน์เตอร์หาร 16 หลังจากสัญญาณ write to SBUF แอกตีฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับข้อมูลเริ่มต้นเมื่อมีการตรวจพบว่าการเปลี่ยนสถานะจาก 1 เป็น 0 ที่ขา RXD การตรวจสอบสถานะสัญญาณที่ขา RXD จะกระทำด้วยความถี่ 16 เท่าของค่า baud rate ในขณะนั้นจากนั้นเคาน์เตอร์หาร 16 จะถูกรีเซตทันที และค่า 1FFH จะถูกโหลดไปที่รีจิสเตอร์สำหรับข้อมูล

ช่วงเวลาในการตรวจสอบข้อมูลภายนอกที่ขา RXD แต่ละบิต จะถูกแบ่งออกเป็น 16 สเตทย่อย การตรวจสอบจะกระทำที่สเตทที่ 7,8,9 วงจรตรวจสอบค่าสัญญาณที่ขา RXD ซึ่งค่าที่ถูกรับเข้ามาเป็นข้อมูลจะต้องตรวจพบอย่างน้อย 2 ครั้งจากการตรวจสอบทั้งหมด 3 ครั้ง ทั้งนี้เพื่อป้องกันสัญญาณรบกวนที่เกิดขึ้น แต่ถ้าบิตแรกที่ถูกรับเข้ามาถูกตรวจสอบพบว่าไม่เป็น 0 วงจรควบคุมการรับข้อมูลจะถูกรีเซต และกลับไปเริ่มต้นตรวจการเปลี่ยนสถานะที่ขา RXD ตามเดิม ทั้งนี้เพื่อให้แน่ใจว่าบิตแรกที่เข้ามาเป็นบิตเริ่มต้นของข้อมูลแน่นอนนั่นเอง หากบิตแรกตรวจพบว่ามีค่าเป็น 0 นั่นคือบิตนี้เป็นบิตเริ่มต้นของข้อมูลแน่นอน การรับข้อมูลจะเริ่มขึ้น โดยเลื่อนบิตเริ่มต้นของข้อมูลที่รับเข้ามาได้ไปไว้ที่รีจิสเตอร์สำหรับรับข้อมูล

ขณะที่บิตข้อมูลถูกเลื่อนเข้ามาทางขวาในรีจิสเตอร์สำหรับรับข้อมูลจะทำให้ 1FFh ซึ่งถูกนำไปไว้ที่รีจิสเตอร์สำหรับรับข้อมูลในตอนแรกถูกเลื่อนออกไปทางซ้าย เมื่อบิตเริ่มต้นของข้อมูล (0) ถูกเลื่อนไปถึงตำแหน่งซ้ายสุดในรีจิสเตอร์สำหรับรับข้อมูล (ขนาด 9 บิต) จะมีสัญญาณส่งไปบอกวงจรควบคุมการรับข้อมูลให้ทำการเลื่อนข้อมูลเข้ามาเป็นครั้งสุดท้ายอีก 1 ครั้ง แล้วจึงส่งสัญญาณเพื่อนำข้อมูลจากรีจิสเตอร์สำหรับรับข้อมูลมาไว้ที่รีจิสเตอร์ใช้งานเฉพาะ SBUF และนำบิตที่ถูกรับเข้ามาครั้งสุดท้ายไปไว้ที่บิต RB8 รวมทั้งเซตบิต RI เหตุการณ์ทั้งหมดนี้เกิดขึ้นก็ต่อเมื่อเป็นไปตามเงื่อนไขต่อไปนี้เมื่อพัลส์ของสัญญาณ shift ลูกสุดท้ายถูกสร้างขึ้น

- บิต RI = 0
- บิต SM2 = 0 หรือ บิตที่ 9 ที่รับเข้ามามีค่าเป็น 1

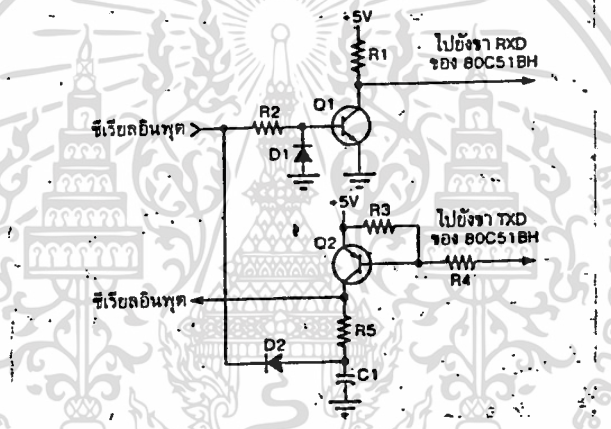
ถ้าเงื่อนไขใดไม่ตรงตามนี้ ข้อมูลไบต์ที่ถูกรับเข้ามาทั้งหมดจะหายไปโดยไม่สามารถเรียกคืนกลับมาได้ และบิต RI จะไม่ถูกเซต แต่ถ้าตรงตามเงื่อนไขทั้งสองอย่างนี้ บิตที่ 9 ที่รับเข้ามาจะถูกนำไปไว้ในบิต RB8 ของรีจิสเตอร์ SCON ส่วนบิตข้อมูลทั้ง 8 จะถูกนำไปไว้ในรีจิสเตอร์ใช้งานเฉพาะ SBUF ในช่วงเวลาของการรับข้อมูล 1 บิตต่อมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ว่าเงื่อนไขทั้งสองจะเป็นอย่างไร ระบบการรับข้อมูลก็จะกลับไปตรวจหาการเปลี่ยนสถานะจาก 1 เป็น 0 ที่ขา RXD ต่อ (รอรับบิตเริ่มต้นของข้อมูลไบต์ถัดไป)

สังเกตว่าค่าของบิตสิ้นสุดของข้อมูลที่ถูกรับเข้ามาจะไม่เกี่ยวข้องกับรีจิสเตอร์ SBUF บิต RB8 หรือบิต R1 แต่อย่างใดเลย

ในกรณีที่ใช้พอร์ตสื่อสารข้อมูลแบบอนุกรมเพื่อเชื่อมต่อกับวงจรภายนอกตามมาตรฐาน RS-232 จำเป็นต้องมีวงจรเพิ่มเติมเพื่อช่วยยกระดับแรงดันให้ได้มาตรฐาน ทั้งนี้เพราะการติดต่อสื่อสารแบบอนุกรมตามมาตรฐาน RS-232 ใช้ระดับแรงดัน 12 โวลต์วงจรสำหรับเชื่อมต่อพอร์ตสื่อสารข้อมูลแบบอนุกรมของ MCS-51 ตามมาตรฐาน RS-232 มีดังแสดงในรูปที่ 2.13



รูปที่ 2.13 แสดงวงจรการเชื่อมต่อพอร์ตสื่อสารข้อมูลแบบอนุกรมของ MCS-51 ตามมาตรฐาน RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 โปรโตรคอลการสื่อสารข้อมูลแบบอนุกรม

โปรโตรคอล (protocal) ของการสื่อสารข้อมูลแบบอนุกรมสามารถแบ่งออกเป็น 2 ประเภทใหญ่ ๆ คือ

1. โปรโตรคอลสำหรับการสื่อสารข้อมูลแบบซิงโครนัส (Synchronous)

2. โปรโตรคอลสำหรับการสื่อสารข้อมูลแบบอซิงโครนัส (Asynchronous)

การสื่อสารแบบซิงโครนัสข้อมูลจะถูกส่งออกไปอย่างสม่ำเสมอช่วงเวลาระหว่างบิตและระหว่างเวิร์ดจะมีค่าเท่ากันเสมอดังนั้นในการสื่อสารแบบซิงโครนัสจึงต้องมีสายสัญญาณเพื่อกำกับการส่งว่าควรจะส่งเมื่อใดและควรจะหยุดเมื่อใดระบบนี้จะมีความเร็วสูงมากแต่ก็ยังต่ำกว่าแบบขนาน

การสื่อสารแบบอซิงโครนัสช่วงเวลาระหว่างบิตจะมีค่าเท่ากันเหมือนการสื่อสารแบบซิงโครนัสแต่แตกต่างกันที่ระยะห่างระหว่างเวิร์ดซึ่งไม่มีข้อกำหนดว่าจะห่างกันเป็นระยะทางเท่าใด จึงมีการกำหนดข้อตกลงเกี่ยวกับฟอร์แมตของข้อมูลที่จะส่งให้ทางผู้รับสามารถเข้าใจว่าจุดใดเป็นจุดเริ่มต้นของเวิร์ดข้อกำหนดดังกล่าวกำหนดให้แต่ละเวิร์ดต้องขึ้นต้นด้วย บิตเริ่มต้น (start bit) ซึ่งต้องเป็นลอจิกศูนย์เสมอ จากนั้นตามด้วยบิตข้อมูลที่ต้องการส่งซึ่งมีความยาว 5 ถึง 8 บิต ถัดจากข้อมูลก็เป็นพาริตีบิตซึ่งทำหน้าที่เป็นบิตตรวจสอบความถูกต้องของข้อมูล บิตพาริตีนี้มี 2 แบบ คือ

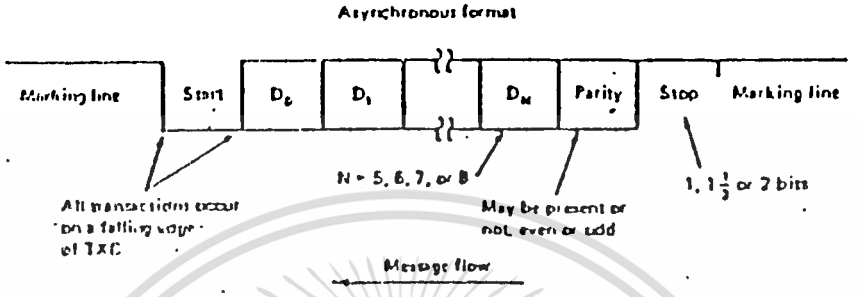
- Even parity ซึ่งจะเซตเมื่อจำนวนบิตที่เป็นลอจิก 1 ในบิตที่เป็นข้อมูลมีจำนวนเป็นคี่

- Odd parity ซึ่งจะเซตเมื่อจำนวนบิตที่เป็นลอจิก 1 ในบิตที่เป็นข้อมูลมีจำนวนเป็นคู่

ในการส่งข้อมูลที่มีความน่าเชื่อถือสูงอาจจะไม่มีการใช้บิตพาริตีก็ได้เพราะสัญญาณรบกวนต่ำมีการเพิ่มความเร็วในการสื่อสารได้ด้วย บิตสุดท้ายในฟอร์แมตก็คือ Stop bit ทำหน้าที่บอกทางผู้รับว่าขณะนี้ข้อมูลที่ทางผู้รับได้รับนั้นครบเวิร์ดแล้วขอให้เตรียมตัวรับเวิร์ดต่อไป Stop bit ถูกกำหนดให้เป็นลอจิก 1 เสมอ อาจมีความยาว 1 บิต 1.5 บิต 2บิตก็ได้ ฟอร์แมตในการสื่อสารข้อมูลดังกล่าว เช่น 5E1 (5 Data bit , Even parity, 1 Stop bit) , 7E1 (7 Data bit,Even parity, 1 Stop bit) ,8N1 (8 Data bit ,No parity ,1 Stop bit) ในการใช้งานทั่ว ๆ ไปเรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

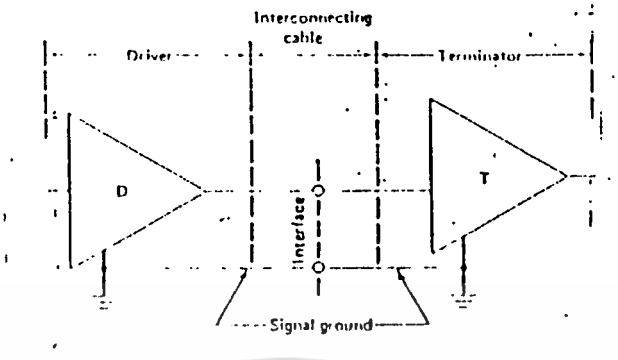
นิยมใช้อยู่เพียง 2 ฟอร์แมต คือ 7E1,8N1การเลือกใช้ฟอร์แมตแบบใดขึ้นอยู่กับสภาพสาย ส่งสัญญาณว่ามีสัญญาณรบกวนมากเพียงใด หากสายสัญญาณมีการรบกวนมากก็ควรใช้ 7E1 แต่ถ้าสายสัญญาณมีสภาพดีมีสัญญาณรบกวต่ำควรใช้ 8E1 ลักษณะของข้อมูลที่ถูกส่งออกไปจะมีลักษณะดังรูป 2.14



รูปที่ 2.14 ลักษณะข้อมูลฟอร์แมตแบบ 8N1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การสื่อสารข้อมูลมาตราฐาน-RS-232



รูปที่ 2.15 ลักษณะสมบัติทางไฟฟ้าของการอินเตอร์เฟสแบบ RS-232

ลักษณะของ RS-232

- ถูกออกแบบให้ใช้กับอุปกรณ์พวก discrete
- ใช้การอินเตอร์เฟสแบบ Unbulance
- ในแต่ละเซอร์กิตใช้ลวดตัวนำสัญญาณ 1 เส้น
- อัตราเร็วในการส่งข้อมูลมีค่า < 20 kbps
- ระยะทางสูงสุดที่ใช้ในการส่งข้อมูลมีค่า < 15 m

ตามมาตราฐาน RS-232 ที่ถูกตีพิมพ์โดย EIA ได้กล่าวถึงการสื่อสารข้อมูลระหว่าง Data Terminate (DTE) และ Data Communication Equipment (DCE) โดยที่

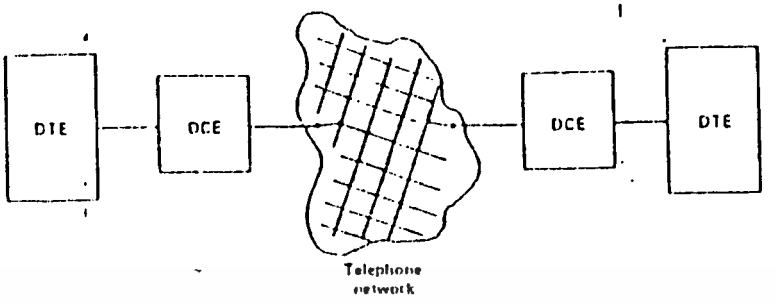
DCE : เป็นอุปกรณ์ที่มีฟังก์ชันการทำงานต่าง ๆ ที่ทำให้เกิดการเชื่อมต่อดำรงต่อไปและยุติการเชื่อมต่อ นอกจากนี้ยังใช้เปลี่ยนลักษณะของสัญญาณและสร้างรหัสสัญญาณต่าง ๆ ที่จำเป็นในการสื่อสารข้อมูลระหว่าง DTE และ Data Circuit โดย DCE อาจจะเป็นส่วนใดส่วนหนึ่งของคอมพิวเตอร์หรือไม่ก็ได้

DTE :1. เป็นอุปกรณ์ที่ประกอบไปด้วยตัวส่งข้อมูล (Data Source) หรือตัวรับข้อมูล (Data Sink) หรือเป็นทั้งตัวส่งและตัวรับก็ได้

2. เป็นอุปกรณ์ที่ประกอบด้วย Function Unit ต่อไปนี้ Control logic,Buffer store อุปกรณ์อินพุท เอ้าท์พุทหรือเครื่องคอมพิวเตอร์ DTE อาจจะ

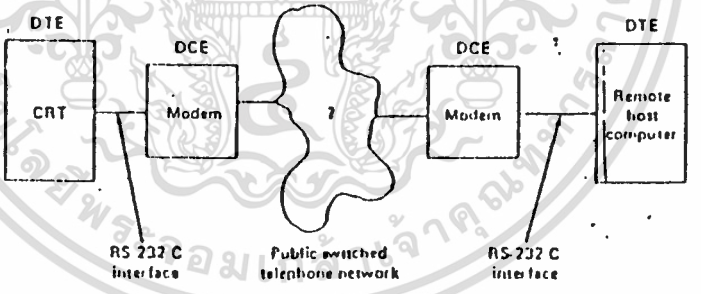
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รวมส่วน Error Control,Synchronization และความสามารถในการบ่งหรือระบุว่าต้องการเกี่ยวข้องกับอุปกรณ์ตัวใดเข้าไปด้วยก็ได้



รูปที่ 2.16 ลักษณะของ DCE และ DTE ที่ใช้ในการสื่อสารข้อมูล

ในหลักความเป็นจริง DTE มักจะแทนแหล่งกำเนิดข้อมูลแหล่งแรก/หรืออุปกรณ์ที่เป็นแหล่งข้อมูลแหล่งสุดท้าย เช่น เครื่องพิมพ์ ส่วน DCE เป็นอุปกรณ์ที่ทำให้การสื่อสารข้อมูลระหว่างแหล่งกำเนิดกับตัวรับข้อมูลที่ปลายทางทำได้สะดวกขึ้นตัวอย่างของ DCE ก็คือ MODEM



รูปที่ 2.17 ลักษณะของระบบที่ใช้แสดงเป็นตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิตที่ 9 ที่รับเข้ามาในการทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมทั้งสองโหมดจะถูกนำไปไว้ในบิต RB8 ของรีจิสเตอร์ใช้งานเฉพาะ SCON ตามด้วยบิตสิ้นสุดของข้อมูล เหมือนการรับส่งข้อมูลทั่วไปที่กล่าวมาแล้ว ถ้าบิต SM2 ถูกเซต (เลือกให้พอร์ตสื่อสารข้อมูลแบบอนุกรมใช้ติดต่อระหว่างซีพียูด้วยกันเอง) และบิตสิ้นสุดของข้อมูลถูกรับเข้ามาแล้ว หากบิต RB8 (บิตที่ 9 ที่รับเข้ามา) มีค่าเป็นหนึ่ง จะส่งผลไปกระตุ้นให้วงจรส่วนควบคุมการอินเตอร์รัปต์ของพอร์ตสื่อสารข้อมูลแบบอนุกรมเริ่มทำงาน เพื่ออินเตอร์รัปต์ซีพียูต่อไป หากบิต RB8 มีค่าเป็นศูนย์ วงจรส่วนควบคุมการอินเตอร์รัปต์ของพอร์ตสื่อสารข้อมูลแบบอนุกรมจะไม่ทำงานแต่อย่างใด รายละเอียดการทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมที่ใช้ในการติดต่อระหว่างซีพียูด้วยกันเองมีดังนี้

เมื่อไมโครคอนโทรลเลอร์ตัวแม่หรือตัวหลัก (master processor) ต้องการส่งข้อมูลจำนวนหนึ่งไปยังไมโครคอนโทรลเลอร์ตัวลูก (slave) ตัวใดตัวหนึ่งจากที่มีหลายตัวในระบบ ขั้นแรกไมโครคอนโทรลเลอร์ตัวแม่จะต้องส่งข้อมูลขนาด 1 ไบต์ ที่มีชื่อเรียก "แอดเดรสไบต์" (address byte) ค่าแอดเดรสไบต์นี้จะเป็นค่าที่ระบุหมายเลขประจำหรือตำแหน่งของไมโครคอนโทรลเลอร์ตัวลูก ในระบบที่เป็นเป้าหมายของไมโครคอนโทรลเลอร์ตัวแม่ต้องการติดต่อด้วย ค่าแอดเดรสไบต์ที่ไมโครคอนโทรลเลอร์ตัวแม่ส่งไปมีข้อแตกต่างจากข้อมูลทั่วไปรับส่งกันจริง ๆ ระหว่างไมโครคอนโทรลเลอร์ที่มีชื่อเรียกเฉพาะว่า "ดาต้าไบต์" (data byte) ดังนี้

- แอดเดรสไบต์ : บิตที่ 9 จะเป็น 1
- ดาต้าไบต์ : บิตที่ 9 จะเป็น 0

หากในไมโครโปรเซสเซอร์ตัวลูกมีการเซตบิต SM2 = 1 แล้ว (ใช้พอร์ตสื่อสารข้อมูลแบบอนุกรมใน MCS-51 ติดต่อระหว่างซีพียูด้วยกันเอง) ถ้าข้อมูลที่รับเข้ามาเป็นดาต้าไบต์จะไม่สามารถอินเตอร์รัปต์ซีพียูได้ แต่หากข้อมูลที่ได้รับเป็นแอดเดรสไบต์ไมโครคอนโทรลเลอร์ตัวลูกทุกตัวจะถูกอินเตอร์รัปต์ การทำเช่นนี้เพื่อที่ไมโครคอนโทรลเลอร์ตัวลูกทุกตัวที่เชื่อมต่อกับตัวแม่ จะสามารถตรวจสอบแอดเดรสไบต์ที่ได้รับเข้ามาว่ามีค่าตรงกับหมายเลขตำแหน่งของตัวเองหรือไม่ หากไมโครคอนโทรลเลอร์ตัวลูกตัวใดมีค่าหมายเลขตำแหน่งของตัวเองตรงกับแอดเดรสไบต์ที่ได้รับเข้ามา ก็จะเคลียร์บิต SM2 และเตรียมรับดาต้าไบต์ซึ่งจะตามเข้ามาภายหลังจากที่ได้รับแอดเดรสไบต์เรียบร้อยแล้ว สำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรับไมโครคอนโทรลเลอร์ตัวลูกอื่น ๆ ที่ตรวจสอบแอดเดรสไบต์ดูแล้ว และปรากฏว่าไม่ตรงกับหมายเลขตำแหน่งของตัวเองจะยังคงปล่อยให้บิต SM2 ถูกเซตต่อไป และกลับไปทำงานเดิมที่ค้างอยู่ก่อนได้รับการอินเตอร์รัปต์ต่อโดยไม่สนใจดาต้าไบต์ซึ่ง ตามเข้ามาหลังแอดเดรสไบต์นั่นคือหากไมโครคอนโทรลเลอร์ตัวลูกตัวใดตรวจสอบพบว่า แอดเดรสไบต์ที่ได้รับเข้ามาไม่ตรงกับค่าตำแหน่งของตัวเอง มันจะไม่สนใจดาต้าไบต์ที่ส่งเข้ามาตามหลังแอดเดรสไบต์เลย แต่ในขณะนี้ไมโครคอนโทรลเลอร์ตัวแม่ส่งแอดเดรสไบต์มายังไมโครคอนโทรลเลอร์ตัวลูก ไมโครคอนโทรลเลอร์ตัวลูกจะถูกอินเตอร์รัปต์เพื่อตรวจสอบค่าแอดเดรสไบต์ดูว่ามีค่าตรงกับตำแหน่งของตัวเองหรือไม่ทุกครั้ง

บิต SM2 จะไม่มีผลในการทำงานของพอร์ตสื่อสารข้อมูลแบบอนุกรมโหมด 0 แต่การทำงานในโหมด 1 บิต SM2 สามาถถูกใช้เพื่อตรวจสอบบิตสิ้นสุดของข้อมูล (validity of the stoo bit) โดยในการรับข้อมูลของโหมด 1 ถ้าบิต SM2 = 1 จะเป็นการตรวจสอบบิตสิ้นสุดของข้อมูลโดยหากบิตสิ้นสุดของข้อมูลที่มีค่าเป็น 1 การอินเตอร์รัปต์ของพอร์ตสื่อสารข้อมูลแบบอนุกรมจะไม่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

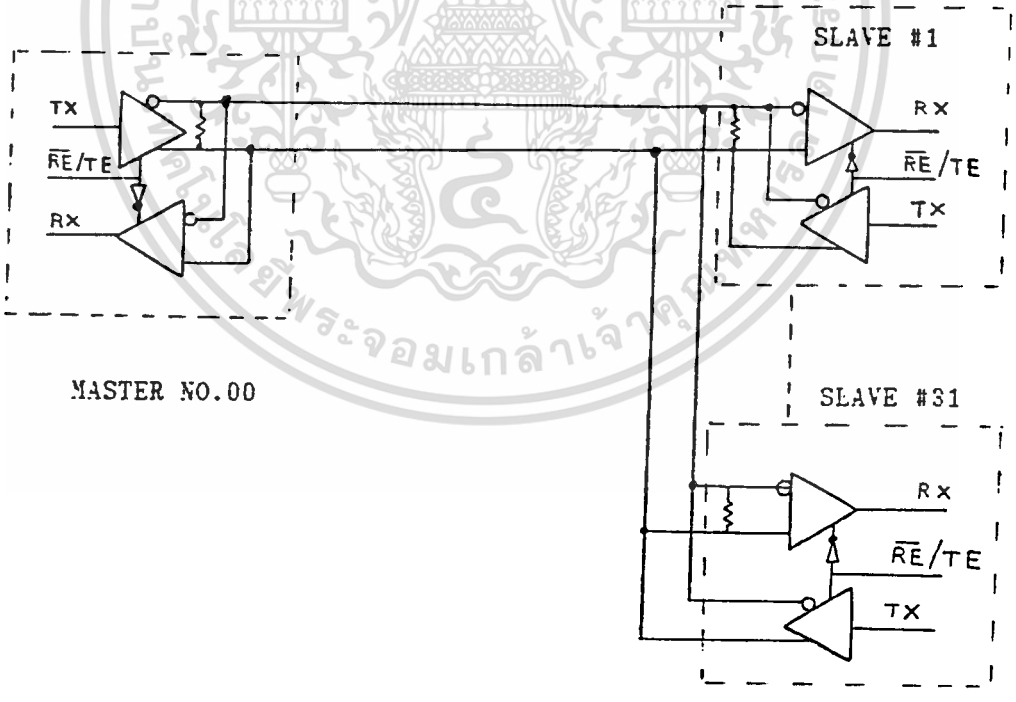
บทที่ 3

การออกแบบและการควบคุมด้วยซอฟแวร์

3.1 การออกแบบส่วนฮาร์ดแวร์

การออกแบบระบบตามมาตราฐาน RS -232 เราจะส่งข้อมูลผ่านทางพอร์ทของ 8255 และปรับระดับแรงดันด้วย IC MAX-232 ซึ่ง RS-232 จะใช้ส่งข้อมูลแบบจุดต่อจุด โดยติดต่อระหว่างบอร์ดต่อบอร์ด และ บอร์ดกับคอมพิวเตอร์

การออกแบบระบบตามมาตราฐาน RS-485 ซึ่งมีการส่งข้อมูลแบบ Half Duplex โดยที่ซอร์ฟแวร์ที่จัดการระบบจะต้องออกแบบให้มีความสัมพันธ์กัน โดยต้องทำงานกันคนละเวลา มาตราฐาน RS-485 นี้ใช้ในการสื่อสารข้อมูลในระยะทางไกล ๆ ฉะนั้นอุปกรณ์ที่ใช้ต้องมีความเที่ยงตรงสูงเพราะถ้าเกิดข้อผิดพลาดในการส่งข้อมูลใน Memory ของ Master กับ Slave จะต่างกันการเชื่อมต่อของระบบ RS-485 เป็นดังรูป 3.1



รูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนฮาร์ดแวร์ประกอบด้วย

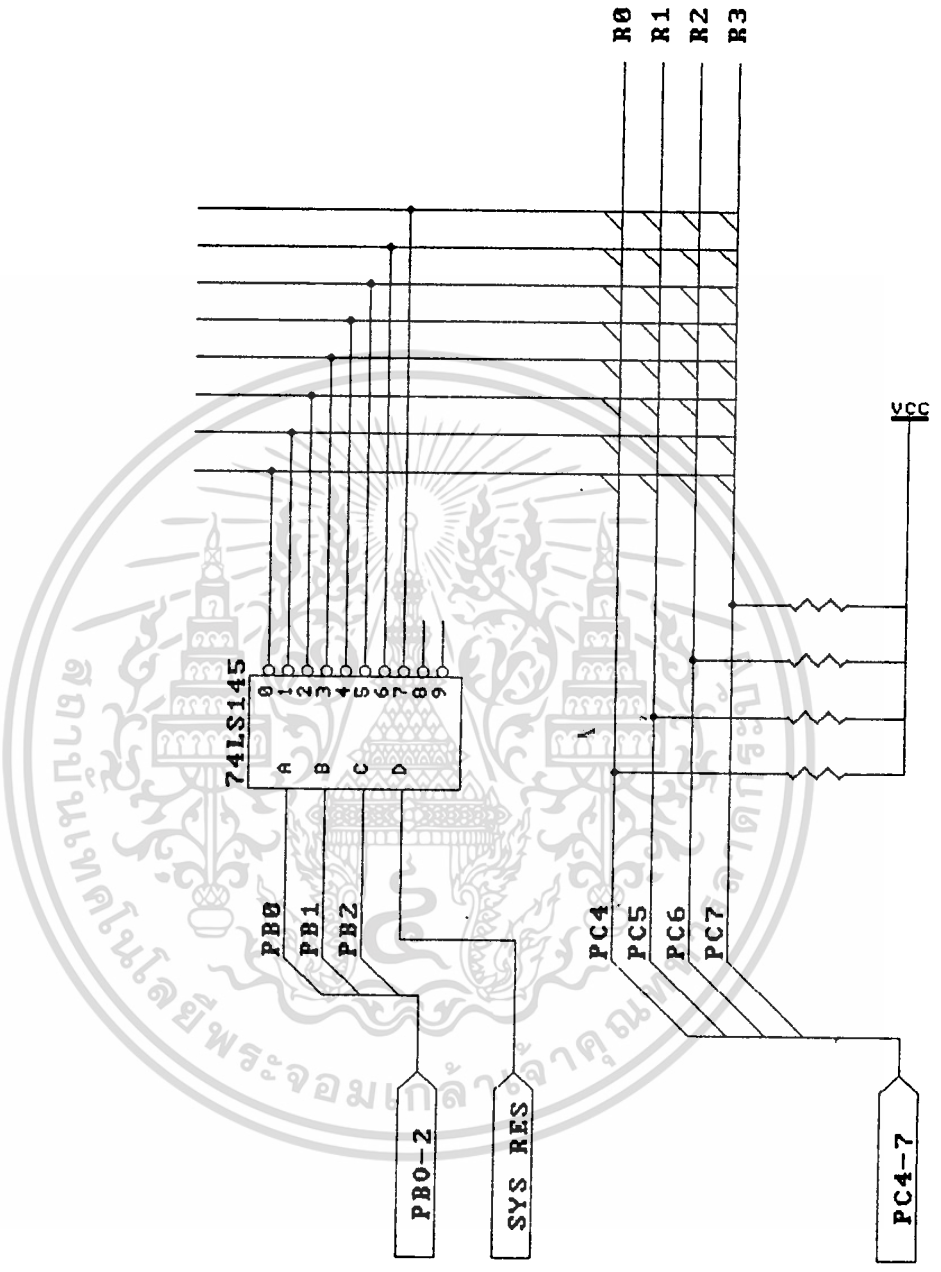
1. CPU BOARD
2. KEY BOARD
3. LCM
4. RS-232 CIRCUIT
5. RS-485 CIRCUIT
6. POWER SUPPLY
7. TERMINAL RS-232 2 ชุด
8. TERMINAL RS-485 1 ชุด

การทำงานของบอร์ดมาสเตอร์

มาสเตอร์บอร์ดจะทำหน้าที่เป็นตัวจัดการทำงานทั้งหมดของระบบเกี่ยวกับฟังก์ชันต่าง ๆ และการ รับ-ส่ง ข้อมูล ซึ่งวงจรของมาสเตอร์บอร์ดเป็นดังรูป 3.2

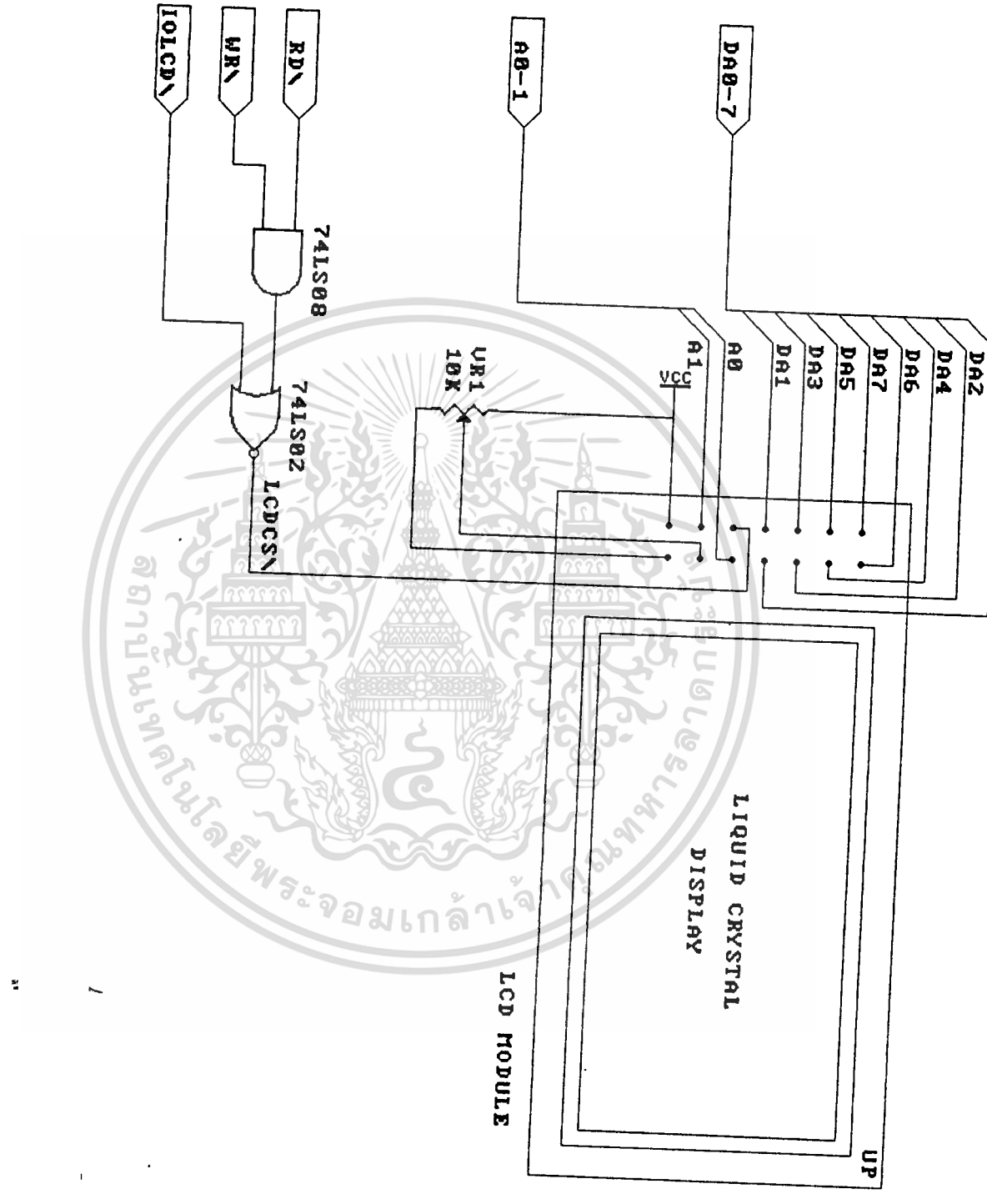
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VPP ENABLE
PMS
ALE DISABLE
VCC
D1 LED
R1 330
74LS32
U3A U3B U3C
R14 4.7K
J1 EXTIN
TO EXP2 RESET
TO SW2 BURN
SW1 RESET
R28 100
C7 10 uF
SYS RES TO 8279
R29 8.2K
X'TAL 11.0592MHz
C6 22 pF
C5 22 pF
U1 8032
ER/VP X1 X2 RESET INT0 INT1 T0 T1 RXD TXD PSEN ALE
P0.0..P0.7
P1.0..P1.7
P2.0..P2.7
P3.0..P3.7
RPACK1 10K
U2 74LS373
U3 27256 (ONLY)
U4 62256/76256
U5 62256
U10 8255
RESET
R26 4.7K
D3 1N4148
D4 1N4148
ALE
AD0..AD7
A8..A14
R18 4.7K
VBAT
(WR)
(RD)
(PSEN)
D5 1N4148
D6 1N4148
R11 4.7K
R8 10K
R10 82K
SW 3a PSM
SW 3b VPP
R9 10K
SW2
J2A J2B J2C
SEE NOTE 2
U7 74LS138
EC8OH
R25 4.7K
R3 10K
R4 10K
R5 4.7K
R6 10K
R7 10K
Q1 Q2 Q3 BC557
Q1-Q2 BC547
D2 1N4148
C1 .1uF
C2 .1uF
VPP00
I/O 34 PIN
R16 10K
R12 5.6K
R13 1K
D22 1N4148
Q6 BC 547
R31 330
R30 82K
R32 4.7K
Q7 BC547
D14 LED
R17 1.2K
A13
A15
U8 74LS138
J2A 62256
J2B 62256
J2C 6264
8000H-9FFFH
(TO DIP SW2)
R37 10K
D9-D11 1N4148
E000H (IORD)
IORD
TO EXP2
R21 4.7K
R22 4.7K
R23 1.8K
R24 4.7K
Q4 BC557
Q5 BC547
4 PIN CONNECTOR
RS 232
VCC RXD TXD GND
R19 100
R20 4.7K
D15 D16 D17
C4 10uF
R31 - R36 10K
D15 - D17 1N4148
EXP1
INT0 INT1 T0 T1 10VDC
FRONT PANEL
EXP2
CONNECTOR
RESET
IORD
NOTE 1
J2
6264 62256
* = Signal for some only
NOTE 2
SW2
OFF ON
1 62256
2 62256,27256,27128
3 62256,6264,WRITE
4 62256,27256,27128
5 27256
6 62256,27256
7 Back UP
8 Burn
UNUSED
U9D
74LS32
U11 LM7805
VI VO GND
D21 P3100R
JACK1 10 VDC
C11 0.1
C10 470uF
C9 47uF
D18 1N4148
R15 330
D20 1N4148
(TO DIP SW2)
(BACK UP)
D19 1N4148
BT1 3.6V
C8 120uF (TAN)
TO PIN28 U4,U5
NOTE THAT PIN 28 OF U4 AND U5 IS NOT DIRECTLY CONNECTED TO VCC
BY ETT.CO,LTD.
TEL 3917716,FAX 3917716
TITLE CONTROL PACK 8032 BASIC
Size Document Number REV 1
C L



รูปที่ 3.3 แสดงการต่อวงจรคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 รูปแสดงการต่อ LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การทำงานของส่วนคีย์บอร์ด

ส่วนของคีย์บอร์ดจะทำหน้าที่รับการกดคีย์ เพื่อไปทำงานตามฟังก์ชันที่กำหนดโดยใช้ 8255 พอร์ตทำงานร่วมกับ IC 74LS145 เพื่อใช้ในการสแกนคีย์ โดยเมื่อคีย์ใดถูกกดก็จะได้ KEY CODE เพื่อบอกให้ CPU ทราบแล้วกระทำตามคีย์ดังกล่าว

ตำแหน่งของฟังก์ชันคีย์บอร์ด

EDIT CH	EDIT RAM	C	D	E	F	BACK	RESET
LIST CH	LIST RAM	8 CLRDM1	9 CLRDM2	A RTEST	B LTEST	INC UNDO	BREAK
NETW	FUNC	4 RXB	5 TXB	6 RECV	7 SEND	DEC BAKD	
MON		0 BAUD	1 CHKS	2 MOVE	3 FIND	ENTER	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดคำสั่งสำหรับระบบการสื่อสารแบบ RS-485 (NETWORK)

ชุดคำสั่งในที่นี้ก็คือ กลุ่มข้อมูลที่สั่งจากเครื่อง Master ไปยังเครื่อง Slave และส่งจาก Slave กลับไปยัง Master เพื่อทำการอ่านข้อมูล โดยจะมีหลักการเหมือนกันทั้งขาไปและขากลับ ซึ่งมีรูปแบบมาตรฐานดังนี้

STX	02H	รหัสเริ่มต้นของชุดคำสั่ง (START OF TEXT)
ADDRESS	00-31	เลขที่ ADDRESS ของเครื่อง Master และเครื่อง Slave
CODE	20-22H	รหัสคำสั่ง
LENGTHH	nnH	ความยาวของข้อมูลที่จะตามมา
LENGTHL	nnH	ความยาวของข้อมูลที่จะตามมา
DATA	nnH	ข้อมูลแต่ละ Byte โดยในกรณีที่ LENGTH เท่ากับ 0 ส่วนของ DATA ก็จะไม่มี
ETX	03H	รหัสจบของชุดคำสั่ง (END OF TEXT)
CHECKSUM	nnH	ค่าสำหรับการตรวจสอบความถูกต้องของชุดคำสั่ง โดยได้มาจากค่า TWO'S COMPLEMENT ของผลข้อมูลตั้งแต่ STX จนถึง ETX

ด้วยรูปแบบของชุดคำสั่งนี้เอง จะทำให้ระบบการสื่อสารเป็นไปอย่างมีประสิทธิภาพ ซึ่งในกรณีที่มีสัญญาณรบกวนทำให้ข้อมูลผิดพลาดไป เราก็สามารถตรวจสอบได้ด้วยค่า CHECKSUM และขณะเดียวกันก็มีความยืดหยุ่นด้วยค่า LENGTH รหัสคำสั่ง (Code) ที่ใช้กับเครื่อง MBC-1 จะมีรายละเอียดดังจะได้อธิบายต่อไปนี้ ซึ่งในตัวอย่างของคำสั่ง จะสมมติให้ ADDRESS = 01H และค่า CHECKSUM ของคำสั่งจะขอเขียนเป็นเพียง XXH เอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CODE 20H

DESCRIPTION DUMMY

MASTER -> 02H,01H,20H,00H,00H,03H,XXH

MASTER <- 02H,01H,20H,00H,00H,03H,XXH

คำอธิบาย สำหรับการตรวจสอบว่ามีเครื่อง SLAVE ตามหมายเลข ADDRESS นั้น ๆ อยู่ในระบบหรือไม่ โดยดูจากการส่งข้อมูลตอบกลับมา ซึ่งถ้าไม่มีการตอบกลับมา ก็แสดงว่าไม่มีหมายเลข ADDRESS นั้น ๆ อยู่ในระบบ

CODE 21H

DESCRIPTION READ DATA

MASTER -> SLAVE 02H,01H,21H,00H,00H,03H,XXH

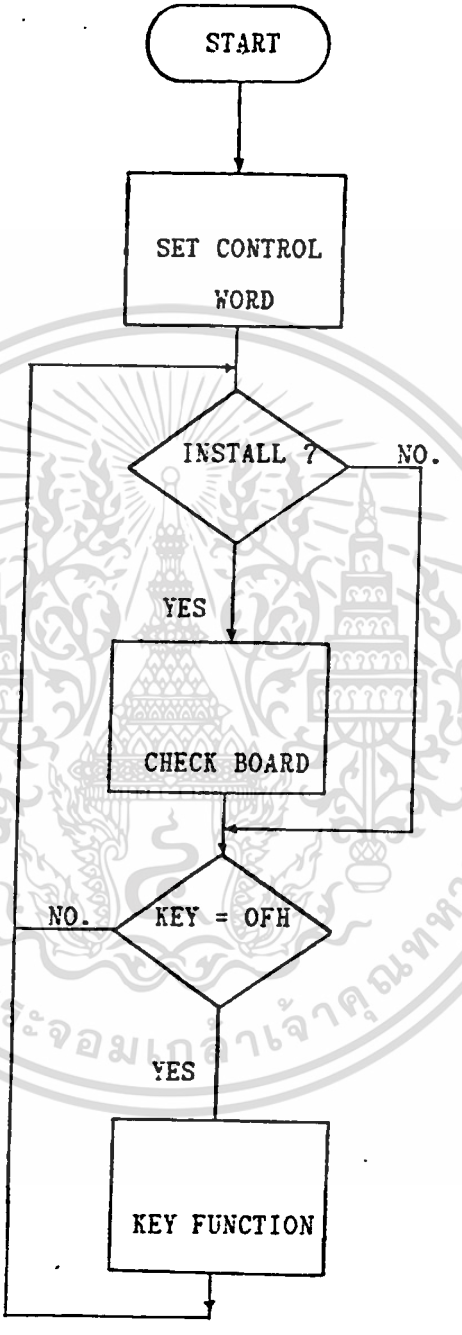
MASTER <- SLAVE 02H,01H,21H,00H,06H,nnH,nnH,nnH,nnH,nnH

nnH,03H,XXH

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

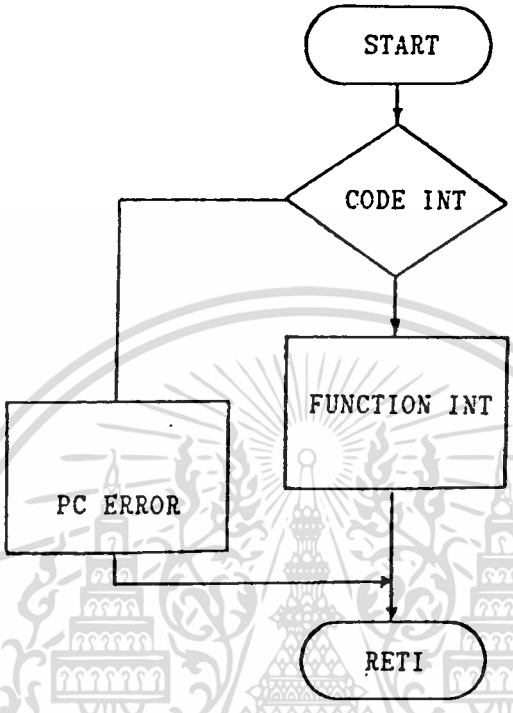
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAIN PROGRAM



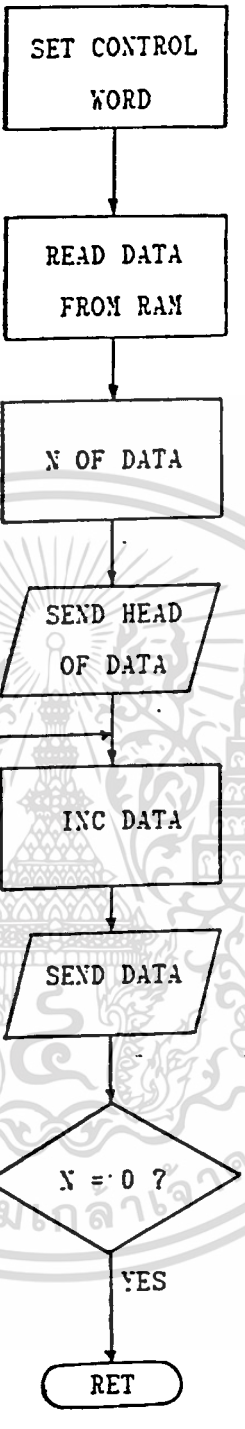
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTERUPT ROUTINE



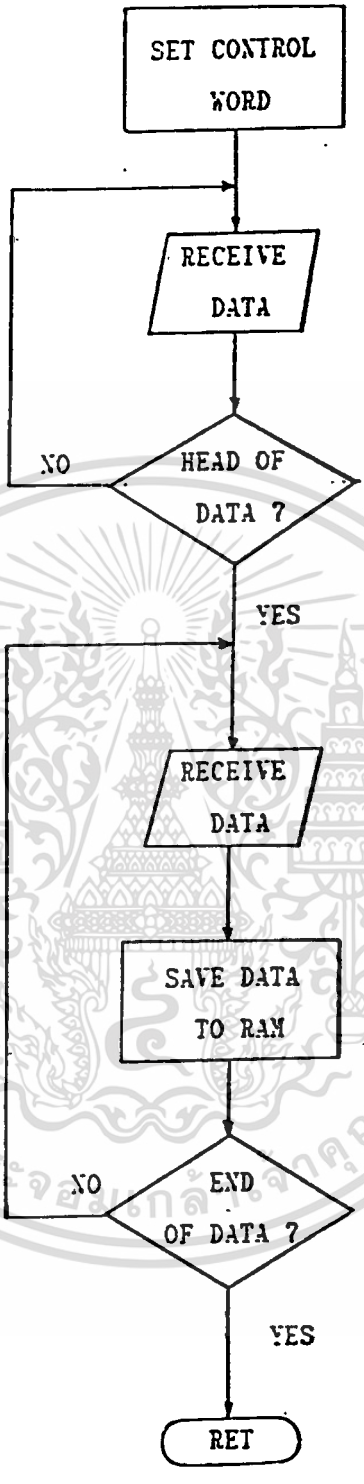
FLOW CHART INTERUPT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FLOW CHART TRANSMIT ROUTAINE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FLOW CHART RECEIVE ROUTINE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ฟังก์ชันและการใช้งาน

4.1 ตำแหน่งของคีย์บอร์ดและฟังก์ชั่นการใช้งาน

EDIT CH	EDIT RAM	C	D	E	F	BACK	RESET
LIST CH	LIST RAM	8 CLRDM1	9 CLRDM2	A RTEST	B LTEST	INC UNDO	BREAK
NETW	FUNC	4 RXB	5 TXB	6 RECV	7 SEND	DEC BAKD	
MON		0 BAUD	1 CHKS	2 MOVE	3 FINE	ENTER	

การใช้งาน FUNCTION

MICROCONTROLLER-BASED COMMUNICATION (MBC-1)

MBC-1 ประกอบด้วย 16 FUNCTION ซึ่งช่วยให้ผู้ใช้สามารถใช้งานได้สมบูรณ์แบบ FUNCTION ดังกล่าวได้แก่

FUNC 0 : BAUD RATE

FUNC 1 : CHECK SUM

FUNC 2 : MOVE DATA

FUNC 3 : FINE DATA

FUNC 4 : RXB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FUNC 5 : TXB

FUNC 6 : RECV

FUNC 7 : SEND

FUNC 8 : CLEAR DATA MEMORY 1 (U4)

FUNC 9 : CLEAR DATA MEMORY 2 (U5)

FUNC 10 : RTEST

FUNC 11 : LTEST

4.2 การทำงานและการแสดงผลของฟังก์ชั่น

FUNC 0 : BAUD RATE

ใช้ในกรณีการตั้งอัตราเร็วของการสื่อสารของ Hardware RS-232 ผู้ใช้สามารถตั้งค่า BAUD RATE ได้ 5 ค่า ดังนี้

1,200 BPS

2,400 BPS

4,800 BPS

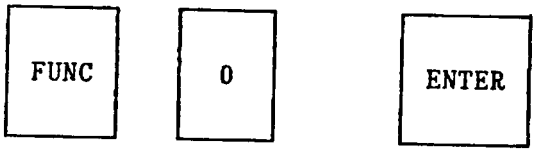
9,600 BPS

19,200 BPS

ถ้าผู้ใช้ไม่ได้ตั้งค่าอัตราเร็วใน FUNCTION "BAUD" นี้ MBC-1 จะตั้งค่าให้อัตโนมัติเมื่อเปิดเครื่องที่อัตราเร็ว 9,600 BPS เสมอ

วิธีการ

1. กดคีย์

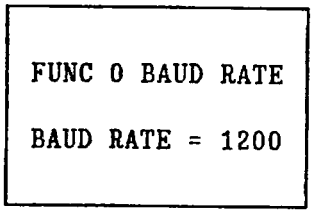


2. ใส่อัตราเร็วที่ต้องการจากคีย์หมายเลขเพื่อเปลี่ยนแปลงค่า

เช่น กำหนดให้อัตราเร็วของการสื่อสาร Hardware RS-232 เท่ากับ 1,200 BPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอปรากฏ



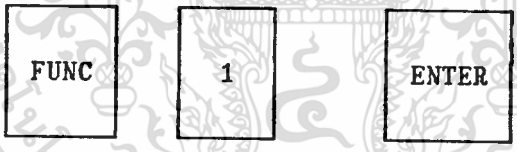
3. เมื่อกำหนดเสร็จเรียบร้อยแล้ว กดคีย์ ENTER

FUNC 1 : CHECK SUM

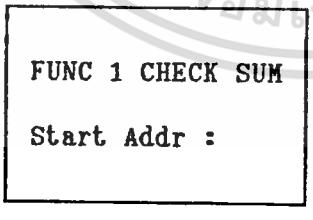
ใช้ในการตรวจสอบค่า Check sum ของข้อมูลในช่วงที่กำหนด เพื่อประโยชน์ในการตรวจสอบความถูกต้องของข้อมูลที่เก็บ

วิธีการ

1. กดคีย์



หน้าจอจะปรากฏ

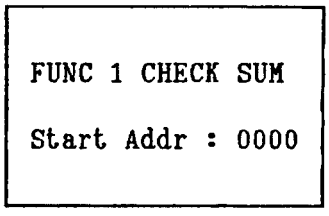


2. ใส่แอดเดรสเริ่มต้นที่ต้องการ Check sum จากคีย์หมายเลข

เช่น กำหนดให้ไปเริ่ม Check sum ที่แอดเดรส 0000H

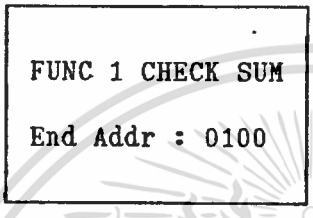
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอจะปรากฏ

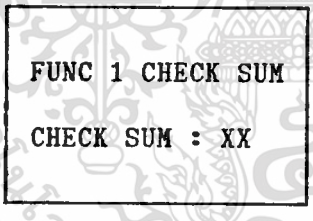


3. ใส่แอดเดรสสิ้นสุดของการ Check sum จากคีย์หมายเลข

เช่น กำหนดให้สิ้นสุดการ Check sum ที่แอดเดรส 0100 H



4. MBC-1 จะให้ค่า Check sum ของช่วงที่กำหนด ซึ่งจิงเป็นเลข 2 หลัก (ผู้ใช้สามารถใช้ค่า Check sum นี้เปรียบเทียบกับค่า Check sum ในครั้งต่อไป ถ้าค่าแตกต่างไปจากเดิม แสดงว่า ข้อมูลมีการเปลี่ยนแปลงไป)



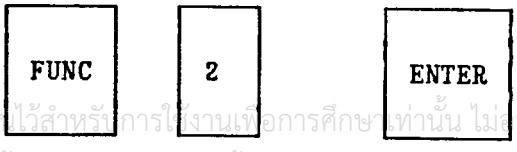
5. กดคีย์  เพื่อกลับสู่ MAIN PROGRAM

FUNC 2 : MOVE DATA

ใช้ในการย้ายข้อมูลจากช่วงของแอดเดรสที่กำหนดไปยังแอดเดรสอื่น ๆ

วิธีการ

1. กดคีย์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ใส่แอดเดรสเริ่มต้นที่ต้องการ MOVE จากคีย์หมายเลข

เช่น กำหนดให้เริ่มย้ายข้อมูลที่แอดเดรส 0000H

หน้าจอจะปรากฏ

```
FUNC 2 : MOVE DATA
Start Addr : 0000
```

3. กดคีย์ [ENTER] เมื่อใส่ค่า แอสเดรสเริ่มต้นแล้ว

4. ใส่แอดเดรสสิ้นสุดของการ MOVE จากคีย์หมายเลข

เช่น กำหนดให้สิ้นสุดการย้ายข้อมูลที่แอดเดรส 0100 H

หน้าจอจะปรากฏ

```
FUNC 2  : MOVE DATA
End Addr : 0101
```

5. กดคีย์ [ENTER] เมื่อใส่ค่า แอสเดรสสิ้นสุดแล้ว

6. หลังจากการ MOVE ข้อมูลทั้งหมดเรียบร้อยแล้ว จะกลับเข้าสู่ Main Program

FUNC 3 : FIND DATA

ใช้ในการค้นหาข้อมูล ภายในช่วงของแอดเดรสที่กำหนด

วิธีการ

1. กดคีย์

[FUNC] [3] [ENTER]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอจะปรากฏ

```
FUNC 3 FIND DATA
Start Addr :
```

2. ใส่ค่าแอดเดรสเริ่มต้นที่ต้องการค้นหา จากคีย์หมายเลข
เช่น กำหนดให้เริ่มต้นหาข้อมูลตั้งแต่แอดเดรส 0000 H
หน้าจอจะปรากฏ

```
FUNC 3 FIND DATA
Start Addr : 0000
```

3. กดคีย์ [ENTER] เมื่อใส่ค่า แอสเดรสเริ่มต้นแล้ว

4. ใส่แอดเดรสสิ้นสุดของการค้นหา จากคีย์หมายเลข
เช่น กำหนดให้สิ้นสุดการค้นหาข้อมูลที่แอดเดรส 0100 H
หน้าจอจะปรากฏ

```
FUNC 3 FIND DATA
End Addr : 0100
```

5. กดคีย์ [ENTER] เมื่อใส่ค่า แอสเดรสสิ้นสุด

6. ใส่ความยาวของค่าที่ต้องการค้นหาจากคีย์หมายเลข
เช่น กำหนดค่าที่ต้องการหามีความยาว 2 Byte

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอปรากฏ

```
FUNC 3 FIND DATA
Data Lenght = 2
```

7. กดคีย์ ENTER

8. ใส่ค่าแรกที่ต้องการค้นหา จากคีย์หมายเลข
เช่น กำหนดค่าแรกที่ต้องการค้นหาคือ EAH
หน้าจอปรากฏ

```
FUNC 3 FINC DATA
Data -1 : EA
```

9. กดคีย์ ENTER

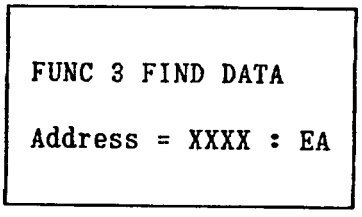
10. ใส่ค่าที่ 2 ที่ต้องการค้นหา
เช่น กำหนดค่าที่ 2 ที่ต้องการค้นหาคือ 39 H
หน้าจอปรากฏ

```
FUNC 3 FIND DATA
Data -2 = 39
```

11. กดคีย์ ENTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. MBC-1 จะหยุดแสดงแอดเดรสที่มีค่า EAH และ 39 H หลังจากนั้นจะไปยังแอดเดรสอื่น ๆ ที่มีค่าที่ต้องการค้นหาจนกว่าจะหมดช่วงที่กำหนด



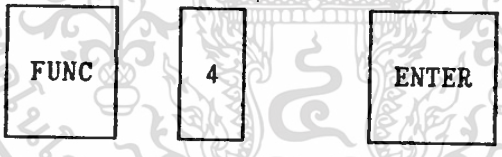
13. กดคีย์ ENTER เพื่อกลับเข้าสู่ Main Program

FUNC 4 : RECEIVE BLOCK

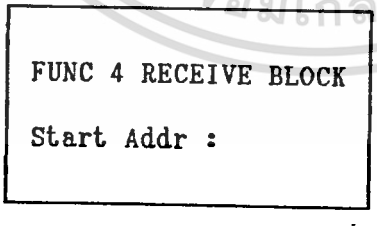
ใช้ในการรับข้อมูลจากพอร์ต Serial 1 โดยจะเก็บข้อมูลที่รับมาไว้ใน Data memory (U5)

วิธีการ

1. กดคีย์



หน้าจอจะปรากฏ

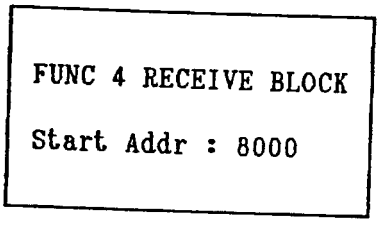


2. ใส่แอดเดรสเริ่มต้นที่ต้องการให้ข้อมูลไปอยู่จากคีย์หมายเลข

เช่น กำหนดให้ข้อมูลที่มาจาก Serial 1 ไปอยู่ที่แอดเดรส 8000 H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอจะปรากฏ



3. เครื่องจะแสดง "Wait Receive Data" แสดงว่ากำลังรอรับข้อมูลจาก Serial 1 และจะแสดงข้อมูลในระดับ bit ที่ LED

หน้าจอจะปรากฏ

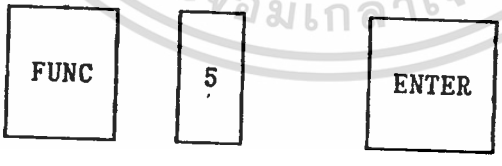
ข้อมูลที่ MBC-1 รับมานี้จะเก็บไว้ใน U5 โดยจะเรียงจากชุดเริ่มต้นที่กำหนดไปเรื่อย ๆ จนกว่าจะหมดข้อมูลที่ต้องการส่ง หลังจากนั้นถ้าผู้ใช้ต้องการออกจาก FUNCTION "RECEIVE BLOCK" ให้กดปุ่ม BREAK เครื่องจะกลับสู่ Main Program

FUNC 5 : TRANSMIT BLOCK

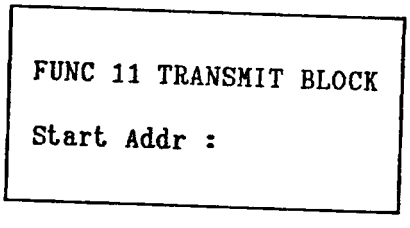
ใช้ในการส่งข้อมูลไปยังอุปกรณ์เป้าหมายผ่านทางพอร์ต Serial 1

วิธีการ

1. กดคีย์



หน้าจอจะปรากฏ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ใส่แอดเดรสเริ่มต้นของข้อมูลที่ต้องการส่ง

เช่น กำหนดให้เริ่มส่งข้อมูลที่แอดเดรส 8000 H

หน้าจอจะปรากฏ

```
FUNC 11 TRANSMIT BLOCK
Start Addr : 8000
```

3. กดคีย์ ENTER เมื่อใส่ค่า แอสเดรสเริ่มต้นแล้ว

4. ใส่แอดเดรสสิ้นสุดของข้อมูลที่ต้องการส่ง

เช่น กำหนดให้สิ้นสุดการส่งข้อมูลที่แอดเดรส 8100 H

หน้าจอจะปรากฏ

```
FUNC 11 TRANSMIT BLOCK
End Addr : 8100
```

5. กดคีย์ ENTER เมื่อใส่ค่า แอสเดรสสุดท้ายแล้ว

4. เครื่องจะแสดง "Please Wait" แสดงว่ากำลังส่งข้อมูลตามแอดเดรสที่กำหนด

หน้าจอจะปรากฏ

```
Please Wait!
```

ถ้าผู้ใช้ส่งข้อมูลหมดแล้ว ต้องการออกจาก FUNCTION "TRANSMIT BLOCK" ให้กดปุ่ม BREAK เครื่องจะกลับเข้าสู่ Main Program

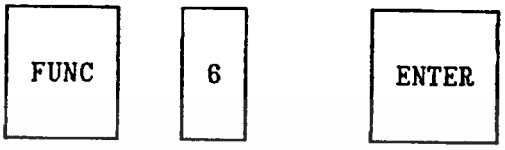
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FUNC 6 : RECEIVE HEX

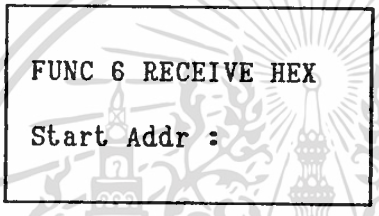
ใช้ในการรับข้อมูลจากพอร์ต Serial 1 โดยจะเก็บข้อมูลที่รับมาไว้ใน Data memory (U4)

วิธีการ

1. กดคีย์



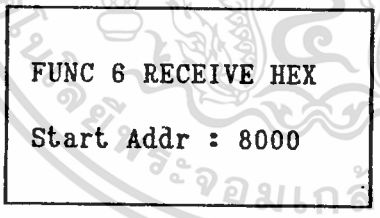
หน้าจอจะปรากฏ



2. ใส่แอดเดรสเริ่มต้นที่ต้องการให้ข้อมูลไปอยู่จากคีย์หมายเลข

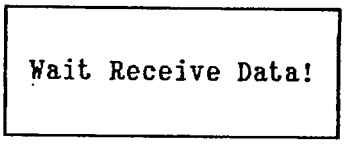
เช่น กำหนดให้ข้อมูลที่มาจาก Serial 1 ไปอยู่ที่แอดเดรส 8000 H

หน้าจอจะปรากฏ



3. เครื่องจะแสดง "Wait Receive Data" แสดงว่ากำลังรอรับข้อมูลจาก Serial 1 และจะแสดงข้อมูลในระดับ bit ที่ LED

หน้าจอจะปรากฏ



ข้อมูลที่ MBC-1 รับมานี้จะเก็บไว้ใน U5 โดยจะเรียงจากชุดเริ่มต้นที่กำหนดไปเรื่อย ๆ จนกว่าจะหมดข้อมูลที่ต้องการส่ง หลังจากนั้นถ้าผู้ใช้ต้องการออกจาก FUNCTION "RECEIVE BLOCK" ให้กดปุ่ม BREAK เครื่องจะกลับสู่ Main Program

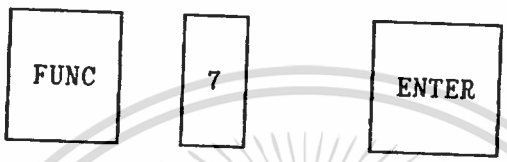
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FUNC 7 : SEND HEX

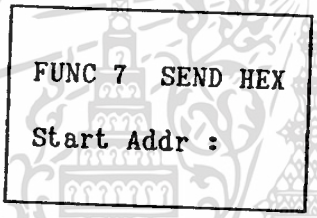
ใช้ในการส่งข้อมูลจาก Data memory (U5) ไปเครื่องคอมพิวเตอร์ PC ผ่านทางพอร์ต Serial 1 โดยจะทำการส่งชุดข้อมูลที่มีลักษณะเป็นข้อมูลมาตรฐานแบบ "INTEL HEX FILE" เท่านั้น ซึ่งจะเป็นขบวนการย้อนกลับของ FUNCTION "RECEIVE HEX"

วิธีการ

1. กดคีย์



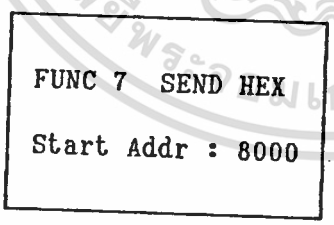
หน้าจอจะปรากฏ



2. ใส่แอดเดรสเริ่มต้นของข้อมูลที่ต้องการส่ง

เช่น กำหนดให้เริ่มส่งข้อมูลของแอดเดรส 8000 H

หน้าจอจะปรากฏ



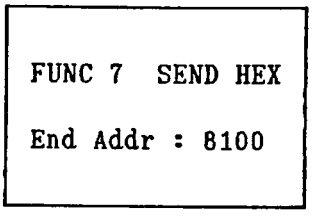
3. กดคีย์ ENTER เมื่อใส่ค่า แอสเดรสเริ่มต้นแล้ว

4. ใส่แอดเดรสสุดท้ายของข้อมูลที่ต้องการส่ง

เช่น กำหนดให้สิ้นสุดการส่งข้อมูลที่แอดเดรส 8100 H

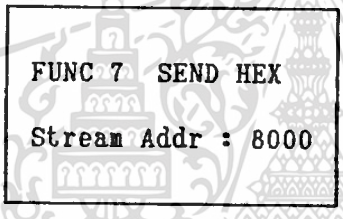
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอจะปรากฏ



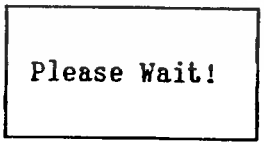
5. กดคีย์ ENTER เมื่อใส่ค่า แอสเดรสสุดท้ายแล้ว

4. ใส่แอดเดรสที่จะใส่ข้อมูลที่ส่งไปปรากฏในอุปกรณ์เป้าหมาย (โดยปกติจะกำหนดให้เท่ากับค่า Start เสมอ แต่ผู้ใช้สามารถกำหนดให้เป็นค่าแอดเดรสอื่นได้ตามต้องการ)



6. กดคีย์ ENTER เมื่อใส่ค่า แอสเดรสสุดท้ายแล้ว

7. เครื่องจะแสดง "Please Wait!" แสดงว่ากำลังส่งข้อมูลไปเครื่อง PC หน้าจอจะปรากฏ



เมื่อส่งข้อมูลทั้งหมดแล้ว จะกลับเข้าสู่ Main Program

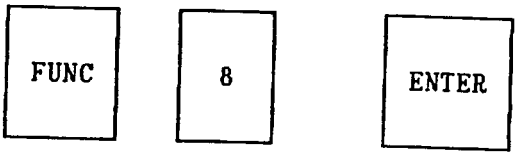
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FUNC 8 : CLEAR DATA MEMORY 1 (U4)

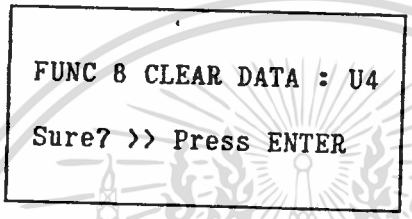
ใช้ในการลบข้อมูลทั้งหมดที่อยู่ใน Data Memory1 (U4)

วิธีการ

1. กดคีย์

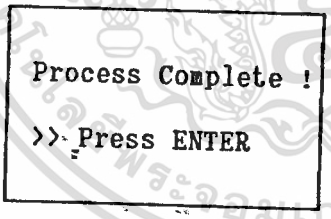


หน้าจอจะปรากฏ



2. กดคีย์ ENTER เมื่อต้องการลบข้อมูลทั้งหมดใน Data Memory1 (U4)

หน้าจอจะปรากฏ



3. กดคีย์ ENTER เพื่อกลับสู่ Main Program

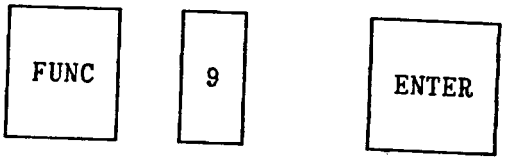
FUNC 9 : CLEAR DATA MEMORY2 (U5)

ใช้ในการลบข้อมูลภายใน Data memory (U5) ให้เป็นค่า "00H" โดยจะ CLEAR ใส่เฉพาะใน 24 Kbyte แรก ในแอดเดรส 8000H - DFFFH

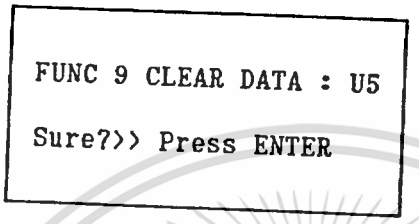
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการ

1. กดคีย์



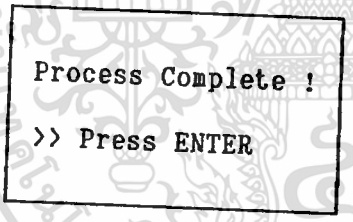
หน้าจอจะปรากฏ



2. กดคีย์ 

เมื่อต้องการลบข้อมูลทั้งหมดใน Data memory2 (U5)

หน้าจอจะปรากฏ



3. กดคีย์ 

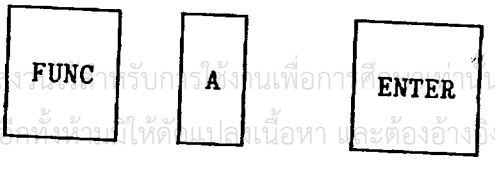
เพื่อกลับเข้าสู่ Main Program

FUNC 10 : RAM TEST

ใช้ในการทดสอบคุณภาพหน่วยความจำโดยเครื่องจะตรวจตั้งแต่ 0000 H ถึง DFFFH คือทั้ง U4 (32K) และ U5 (24K)

วิธีการ

1. กดคีย์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอจะปรากฏ

```
FUNC 10 RAM TEST
Sure?>> Press ENTER
```

2. กดคีย์ [ENTER] เพื่อต้องการทดสอบคุณภาพหน่วยจำ

ถ้าการทำงานของหน่วยความจำดี หลังจากทดสอบเสร็จแล้ว ก็จะกลับเข้าสู่ Main Program

ถ้าการทำงานของหน่วยความจำผิดพลาด เครื่องจะแสดงแอดเดรสที่มีปัญหาให้เห็น

ถ้าผู้ใช้ต้องการทดสอบต่ำให้กด [ENTER] เครื่องจะทดสอบต่อไปจนเสร็จแล้วกลับสู่ Main Program

FUNC 11 : LCD TEST

ใช้ในการทดสอบคุณภาพ LCD

วิธีการ

1. กดคีย์

[FUNC] [B] [ENTER]

หน้าจอจะปรากฏ

```
FUNC 11 LCD TEST
Sure?>> Press ENTER
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กดคีย์ | ENTER | เพื่อต้องการทดสอบคุณภาพ LCD

ถ้าการทำงานของ LCD ดี หลังจากทดสอบเสร็จแล้ว ก็จะกลับเข้าสู่ Main Program

4.3 ความหมายของคีย์ในส่วน EDIT KEY

ความหมายของคีย์ในส่วน EDIT KEY มีความหมายดังนี้

BACK สำหรับการย้อนกลับไปป้อนข้อมูลชุดที่ผ่านมา ในกรณีที่ไม่สามารถย้อนได้ก็จะไม่มีการกระทำแต่อย่างใด

UNDO สำหรับการยกเลิกข้อมูลที่กำลังทำการแก้ไข และนำค่าเดิมที่มีอยู่กลับคืนมา โดยทั้งนี้เพื่อกันความผิดพลาดจากการคีย์ที่ไม่ตั้งใจ

(INCREMENT) สำหรับการเพิ่มค่าแอดเดรสที่แสดงผลอยู่

BCKD (BACK DIGIT) สำหรับการลบถอยหลัง 1 ตัวอักษร เพื่อความสะดวกในการแก้ไขตัวเลขที่ป้อนผิดพลาด

(DECREMENT) สำหรับการลดค่าแอดเดรสที่แสดงผลอยู่

ENTER สำหรับการตกลงตามข้อมูลที่ป้อน และไปสู่การป้อนข้อมูลชุดต่อไป หรือสู่การทำงานต่าง ๆ ตาม FUNCTION ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การใช้งาน DIRECT FUNCTION

EDIT CHANNAL.....(EDIT CH)

EDIT RAM.....(EDIT RAM)

LIST CHANNAL.....(LIST CH)

LIST RAM.....(LIST RAM)

NETWORK.....(NETW)

SUB-FUNCTION.....(FUNC)

STOP.....(STOP)

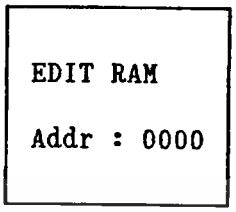
EDIT CH สำหรับแก้ไขข้อมูลโดยจะแก้ไขข้อมูลเป็น CHANNAL โดยเมื่อกดคีย์นี้เครื่องจะให้ป้อนข้อมูลดังนี้

กดคีย์	LCD แสดงผล	ความหมาย
	Direct Function Name	ให้ป้อนค่า CHANNAL และ NUMBER ที่ต้องการแก้ไข จากนั้นให้กด ENTER
EDIT CH	Edit channal CH : 0000 NO.: 00	
	Channal Value Number Value	
ENTER	0000:00: 00 01 02 03 0000:04: 04 05 06 07	เครื่องจะเข้าสู่ EDIT CH
	Channal Number Data ในแต่ละ Channal และ Number นั้น ๆ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน EDIT CH จะกระทำการอ่านและเขียนได้ และมีระบบ AUTO-INC

EDIT RAM สำหรับการแก้ไข โดยจะแก้ไขข้อมูลตามแอดเดรสของ RAM โดยเมื่อกดคีย์นี้ เครื่องจะให้ป้อนข้อมูลดังนี้

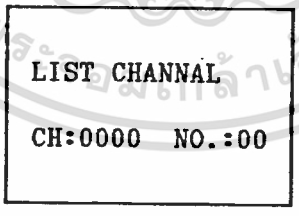


ให้ป้อนค่าแอดเดรสที่จะต้องแก้ไขจากนั้นให้กด

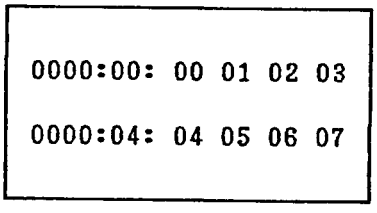
เครื่องจะเข้าสู่ EDIT RA

การใช้งาน EDIT RAM จะกระทำการอ่านและเขียนได้ และมีระบบ AUTO-INC

LIST CH สำหรับการดูข้อมูล โดยจะดูข้อมูลเป็น CHANNAL โดยเมื่อกดคีย์นี้ เครื่องจะให้ป้อนข้อมูลดังนี้



ให้ป้อนค่า CHANNAL และ NUMBER ที่ต้องการดูจากนั้นให้กด



เครื่องจะเข้าสู่ LIST CH

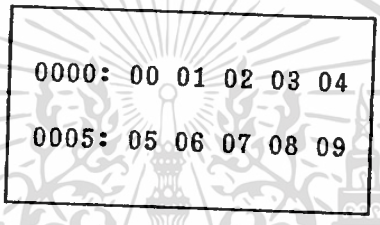
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ LIST CH จะทำการอ่านได้อย่างเดียว และมีระบบ AUTO-INC

LIST RAM สำหรับการดูข้อมูล โดยจะดูข้อมูลตามแอดเดรสของ RAM โดยเมื่อกดคีย์นี้ เครื่องจะให้ป้อนข้อมูลดังนี้

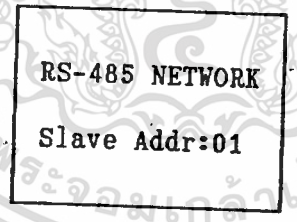
ให้ป้อนค่าแอดเดรสที่ต้องการดูข้อมูลจากนั้นให้กด

เครื่องจะเข้าสู่ LIST RAM

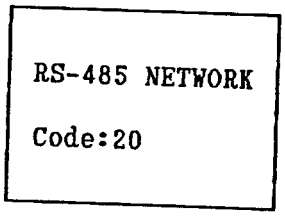


การใช้งาน LIST RAM จะทำการอ่านได้อย่างเดียว และมีระบบ AUTO-INC

NETW สำหรับระบบการสื่อสารแบบ RS-485 โดยเมื่อกดคีย์นี้ เครื่องจะให้ป้อนข้อมูลดังนี้



ให้ป้อนเลขที่แอดเดรสของ Slave ที่ต้องการจะติดต่อ หลังกด



ให้ป้อนรหัสคำสั่ง จากนั้นให้

(รหัสคำสั่งที่ใ

กับเครื่องจะมีรายละเอียดดั

จะได้อธิบายต่อไป)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ENTER

Please Wait!

เครื่องจะแสดง "Please Wait" แสดงว่าเครื่องกำลังทำงานตามรหัสคำสั่งที่ป้อน

Process Complete!
>> Press ENTER

เครื่องจะแสดง "Process Complete!" แสดงว่าเครื่องทำงานเสร็จเรียบร้อยแล้วให้กด ENTER เพื่อกลับเข้าเข้าสู่ Main Program

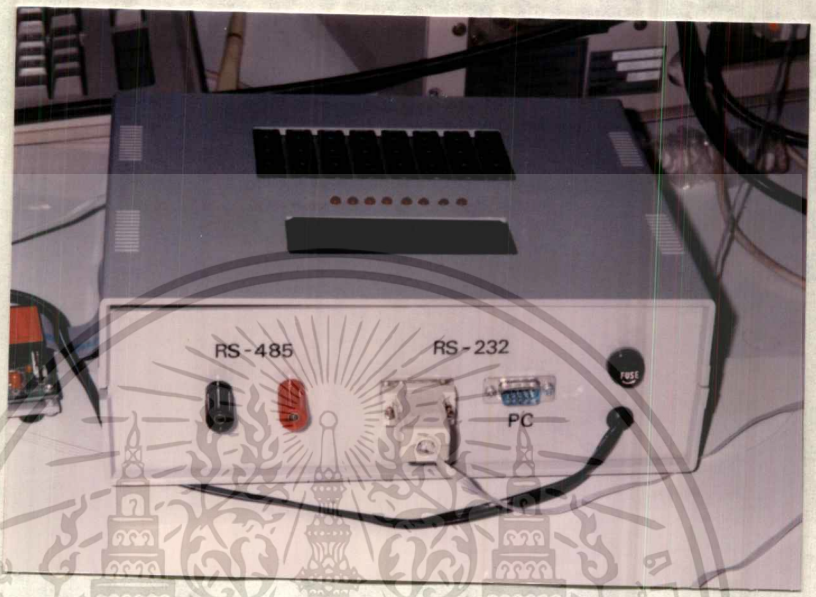
FUNC สำหรับการเรียกใช้ SUB-FUNCTION ต่าง ๆ โดยรายละเอียดการใช้งานขอให้อ่านจากหัวข้อ การใช้งาน SUB-FUNCTION

STOP สำหรับการกลับเข้าสู่ Main Program หรือสภาวะเตรียมพร้อมเพื่อจะใช้งานต่าง ๆ ต่อไป ผู้ใช้จะสามารถกดคีย์ STOP ได้ทันทีตามต้องการ ไม่ว่าจะอยู่ในการทำงานใด ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การต่อใช้งาน

MCB-1 มีจุดต่อเพื่อใช้ในการติดต่อแบบอนุกรมตามมาตราฐาน RS-232 และ RS-485 ดังรูปแผงหน้าปัม์ท



ตำแหน่งของ RS-232 จะมีอยู่ 2 ช่อง คือ PC และ SB

- PC ใช้ในการติดต่อกับเครื่องคอมพิวเตอร์แบบจุดต่อจุดตามมาตราฐาน RS-232
- SB ใช้ในการติดต่อกับ SINGLE BOARD แบบจุดต่อจุดตามมาตราฐาน RS-232

ตำแหน่งของ RS-485 จะมีอยู่ 2 ขา คือ สีแดง และ สีดำ

- สีแดง จะต่อกับสัญญาณบวก
- สีดำ จะต่อกับสัญญาณลบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและข้อแนะนำ

บทสรุป

การสื่อสารตามมาตราฐาน RS-232 เหมาะสำหรับการติดต่อระหว่างจุดต่อจุดในช่วงระยะทางใกล้ ๆ หากการใช้งานที่ต้องมีการติดต่อในระยะทางไกล ๆ RS-485 เป็นระบบที่ได้ออกแบบมาใช้ อีกทั้งยังสามารถติดต่อได้ในแบบหลายจุดคือแบบตัวมาสเตอร์ กับตัวสลาฟ ดังนั้นในโรงงานอุตสาหกรรมซึ่งมี เครื่องมือ-อุปกรณ์ควบคุมต่าง ๆ ที่มีการรับส่งข้อมูลทั้งระยะทางใกล้และไกล หากเรามีการเลือกใช้ระบบการสื่อสารข้อมูลให้เหมาะสมก็จะทำให้การรับส่งข้อมูลมีประสิทธิภาพมากขึ้นและยิ่งไปกว่านั้นหากเครื่องมือของเราสามารถติดต่อได้หลายระบบด้วยแล้วก็สามารถทำให้การทำงานกว้างขวางมากยิ่งขึ้น

ข้อแนะนำ

ซีพียูของอุปกรณ์ไมโครคอนโทรลเลอร์จะมีชุดรับ-ส่งข้อมูลเพียงชุดเดียวแต่เราสามารถเขียนโปรแกรมให้ทำงานในโหมดต่าง ๆ ของระบบรับ-ส่งข้อมูล ทั้งนี้ขึ้นอยู่กับว่าระบบการติดต่อของเรานั้น ระบบไหนสำคัญกว่ากันก็นำระบบนั้นมาต่อกับชุด TXD,RXD ของซีพียูเพราะจะทำให้การเขียนซอร์ฟแวร์และการออกแบบทางฮาร์ดแวร์ทำได้ง่ายและสะดวก หากระบบที่มีความสำคัญรองลงมากก็อาจจะรับ-ส่งผ่านทางพอร์ตอื่นแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

โปรแกรมมอนิเตอร์ประกอบด้วยโปรแกรมสำหรับจัดการระบบและโปรแกรมการใช้งานในส่วน DIRECT KEY คือ EDIT CH,EDIT RAM,LIST CH,LIST RAM, NETW,FUNC,MON,ENTER,DEC,INC,BACK และ FUNCTION KEY คือ BAUD,CHKS, MOVE,FINE,RXB,TBX,RECV,SEND,CLRDM1,CLRDM2,RTEST,LTEST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ********************************************
; PROJECT MICROCONRTOLLER-BASED COMMUNICATION
; BY  MR. SUKCHAI  WAWSUK
;     MR. SAWAI    LAMAI
; ASSOC. PROF. PIPHAT  LAUHASONGKRAM
; YEAR 2537
; ********************************************

; ********** INTERNAL RAM **********
        ORG     0000H
                                ; **** USER AREA ****
        DS      8               ;REGISTER R0-R7 BANK-0
        DS      24
                                ;**** BUFFER AREA ****
BITADD: DS      2               ;BIT ADDRESSABLE FLAG
DISBUF: DS      20              ;DISPLAY BUFFER (ASCII CODE)
HEXBUF: DS      7               ;HEX BUFFER
SCANKM: DS      1               ;SCAN  - KEY MEMORY
SCANAM: DS      1               ;SCAN  - AUTO REPEAT MEMORY
SCANAR: DS      1               ;SCAN  - AUTO REPEAT COUNT DOWN
SCANFC: DS      1               ;SCAN  - FLASH COUNT DOWN
                                ;**** GENERAL AREA ****
RAMTMM:                         ;RAMT  - 3 BYTE ERROR ADDRESS
INPBKH: DS      1               ;INPUT - DATA HIGH BACKUP
INPBKL: DS      1               ;INPUT - DATA LOW BACKUP
FINDAD: DS      1               ;FIND  - ADDRESS
TEMADD:                         ;TEMP  - RECV,SEND,RAMT
MEMADD: DS      2               ;ADDR  - DMEM
                                ;**** MEMORY AREA ****
DMMADD: DS      2               ;DATA ADDRESS
STTADD: DS      2               ;START ADDRESS
ENDADD: DS      2               ;END ADDRESS
DESADD: DS      2               ;DESTINATION ADDRESS
HEXDAT: DS      1               ;HEX DATA
POWMEM: DS      1               ;POWER UP CODE
                                ;**** VIRTUAL AREA ****
VTREG:  DS      8               ;FOR REGISTOR
                                ;**** STACK AREA ****
STACK:  DS      23
INTEND:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ********** VARIABLE SET **********
                                    ;***** PORT *****
LCMWI     EQU     0E020H            ;WRITE INSTRUCTION
LCMRB     EQU     0E021H            ;READ BUSY FLAG
LCMWD     EQU     0E022H            ;WRITE DATA
LCMRD     EQU     0E023H            ;READ DATA
CONA      EQU     0E0E0H            ;OUT SEGMENT
CONB      EQU     0E0E1H            ;OUT 0-2=SCAN 3=LED
CONC      EQU     0E0E2H            ;IN 4-7=KEYIN
CONP      EQU     0E0E3H            ;CONTROL PORT
                                    ;***** PARAMETER *****
AUTOR1    EQU     40H               ;AUTO REPEAT FIRST DELAY
AUTOR2    EQU     30H               ;AUTO REPEAT SECOND DELAY (DEFAULT)
AUTORH    EQU     AUTOR2+8          ;AUTO REPEAT SLOW (8 LEVEL ADJUST)
AUTORL    EQU     AUTOR2-8          ;AUTO REPEAT FAST
COLON     EQU     3AH               ;COLON CHARECTOR
FLASHC    EQU     13H               ;FLASH SPEED DELAY
BRAT12    EQU     0E8H              ;1200 BAUD RATE VAR.
BRAT24    EQU     0F4H              ;2400
BRAT48    EQU     0FAH              ;4800
BRAT96    EQU     0FDH              ;9600
SMOD      EQU     B.7               ;FOR BAUD FUNCTION
POWCOD    EQU     0A5H              ;POWER UP CODE
                                    ;***** BIT-ADDRESSABLE *****
SCANPF    EQU     00H               ;SCAN  - PRESS FLAG
SCANFF    EQU     01H               ;SCAN  - CURSOR FLASH FLAG
SCANFD    EQU     02H               ;SCAN  - FLASH DATA 0=DARK 1=BRIGTH
SCANBF    EQU     03H               ;SCAN  - BY PASS DIRECT KEYS FLAG
INPFIR    EQU     05H               ;INPUT - FIRST FLAG
INPBFG    EQU     06H               ;INPUT - BACK-FIELD FLAG
GETOKF    EQU     08H               ;GET2  - 2 DIGIT OK FLAG
SENDEN    EQU     09H               ;SEND  - LAST LINE FLAG
POWFAG    EQU     0AH               ;RESET - POWER UP FLAG
STMFAG    EQU     0BH               ;RECV  - STREAM FLAG
; ********** INT MAIN **********
; INTERRUPT VECTOR ADDRESS JUMP
          ORG     0000H
          LJMP    RES               ;RESET
IEOVEC:   DB      0FFH,0FFH,0FFH    ;EXT-INT-0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          DB      0FFH,0FFH,0FFH
          DB      0FFH,0FFH
TF0VEC:   DB      0FFH,0FFH,0FFH      ;TIMER-0
          DB      0FFH,0FFH,0FFH
          DB      0FFH,0FFH
IE1VEC:   RET                         ;EXT-INT-1 (FOR BREAK KEY)
          DB      0FFH,0FFH,0FFH
          DB      0FFH,0FFH,0FFH
          DB      0FFH
TF1VEC:   DB      0FFH,0FFH,0FFH      ;TIMER-1 (FOR HW-SERIAL)
          DB      0FFH,0FFH,0FFH
          DB      0FFH,0FFH
RTIVEC:   DB      0FFH,0FFH,0FFH      ;RI & TI UART
          DB      0FFH,0FFH,0FFH
          DB      0FFH,0FFH
; ********* RESET MAIN *********
; SYSTEM RESET
RES:      MOV     R2,#40H             ;POWER UP DELAY
RES1:     MOV     R3,#0
          DJNZ    R3,$
          DJNZ    R2,RES1
          MOV     SP,#STACK
          MOV     A,#89H              ;SET CONSOLE
          MOV     DPTR,#CONP
          MOVX    @DPTR,A
          CLR     A                   ;CLEAR SEGMENT
          MOV     DPTR,#CONA
          MOVX    @DPTR,A
                                      ;**** LCD INITIALIZE ****
          MOV     A,#00111000B        ;8BIT,2LINE,5x7DOT
          LCALL   LCDWI
          MOV     A,#00001000B        ;DISPLAY ON/OFF
          LCALL   LCDWI
          MOV     A,#01H              ;CLEAR DISPLAY
          LCALL   LCDWI
          MOV     A,#10H              ;BREAK=1
          MOV     DPTR,#CONB
          MOVX    @DPTR,A
          CLR     POWFAG              ;CHECK POWER UP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,POWMEM
        CJNE    A,#POWCOD,$+6
        LJMP    RES7                    ;RESET BY KEY
                                        ;**** POWER UP ****
        MOV     R0,#08H                 ;CLEAR INT-RAM (ALL)
        MOV     R2,#INTEND-08H
RES4:   MOV     @R0,#0
        INC     R0
        DJNZ    R2,RES4
        MOV     SCANAM,#AUTOR2          ;DEFAULT VAR.
        MOV     SCANFC,#FLASHC
        MOV     POWMEM,#POWCOD
        SETB    POWFAG
        MOV     DMMADD,#80H             ;DEFAULT ADDRESS
        MOV     DMMADD+1,#0
        MOV     A,#00001110B            ;DISPLAY ON/OFF
        LCALL   LCDWI
        MOV     DPTR,#DATA1             ;**MICROCONTROLLER**
        MOV     A,#93H                  ;BASED  COMUNICATION
        MOV     R2,#1H
        LCALL   LCDWDLn
        MOV     DPTR,#DATA2
        MOV     A,#0D3H
        MOV     R2,#1H
        LCALL   LCDWDLn
        MOV     R2,#2
        LCALL   DTSEC
        MOV     DPTR,#DATA1
        MOV     A,#92H
        MOV     R2,#2H
        LCALL   LCDWDLn
        MOV     DPTR,#DATA2
        MOV     A,#0D2H
        MOV     R2,#2H
        LCALL   LCDWDLn
        MOV     R2,#2
        LCALL   DTSEC
        MOV     DPTR,#DATA1
        MOV     A,#91H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV     R2,#3H
LCALL   LCDWDLn
MOV     DPTR,#DATA2
MOV     A,#0D1H
MOV     R2,#3H
LCALL   LCDWDLn
MOV     R2,#2
LCALL   DTSEC
MOV     DPTR,#DATA1
MOV     A,#90H
MOV     R2,#4H
LCALL   LCDWDLn
MOV     DPTR,#DATA2
MOV     A,#0D0H
MOV     R2,#4H
LCALL   LCDWDLn
MOV     R2,#2
LCALL   DTSEC
MOV     DPTR,#DATA1
MOV     A,#8FH
MOV     R2,#5H
LCALL   LCDWDLn
MOV     DPTR,#DATA2
MOV     A,#0CFH
MOV     R2,#5H
LCALL   LCDWDLn
MOV     R2,#2
LCALL   DTSEC
MOV     DPTR,#DATA1
MOV     A,#8EH
MOV     R2,#6H
LCALL   LCDWDLn
MOV     DPTR,#DATA2
MOV     A,#0CEH
MOV     R2,#6H
LCALL   LCDWDLn
MOV     R2,#2
LCALL   DTSEC
MOV     DPTR,#DATA1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV     A,#8DH
MOV     R2,#7H
LCALL   LCDWDLn
MOV     DPTR,#DATA2
MOV     A,#0CDH
MOV     R2,#7H
LCALL   LCDWDLn
MOV     R2,#2
LCALL   DTSEC
MOV     DPTR,#DATA1
MOV     A,#8CH
MOV     R2,#8H
LCALL   LCDWDLn
MOV     DPTR,#DATA2
MOV     A,#0CCH
MOV     R2,#8H
LCALL   LCDWDLn
MOV     R2,#2
LCALL   DTSEC
MOV     DPTR,#DATA1
MOV     A,#8BH
MOV     R2,#9H
LCALL   LCDWDLn
MOV     DPTR,#DATA2
MOV     A,#0CBH
MOV     R2,#9H
LCALL   LCDWDLn
MOV     R2,#2
LCALL   DTSEC
MOV     DPTR,#DATA1
MOV     A,#8AH
MOV     R2,#10
LCALL   LCDWDLn
MOV     DPTR,#DATA2
MOV     A,#0CAH
MOV     R2,#10
LCALL   LCDWDLn
MOV     R2,#2
LCALL   DTSEC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV     DPTR,#DATA1
MOV     A,#89H
MOV     R2,#11
LCALL   LCDWDLn
MOV     DPTR,#DATA2
MOV     A,#0C9H
MOV     R2,#11
LCALL   LCDWDLn
MOV     R2,#2
LCALL   DTSEC
MOV     DPTR,#DATA1
MOV     A,#88H
MOV     R2,#12
LCALL   LCDWDLn
MOV     DPTR,#DATA2
MOV     A,#0C8H
MOV     R2,#12
LCALL   LCDWDLn
MOV     R2,#2
LCALL   DTSEC
MOV     DPTR,#DATA1
MOV     A,#87H
MOV     R2,#13
LCALL   LCDWDLn
MOV     DPTR,#DATA2
MOV     A,#0C7H
MOV     R2,#13
LCALL   LCDWDLn
MOV     R2,#2
LCALL   DTSEC
MOV     DPTR,#DATA1
MOV     A,#86H
MOV     R2,#14
LCALL   LCDWDLn
MOV     DPTR,#DATA2
MOV     A,#0C6H
MOV     R2,#14
LCALL   LCDWDLn
MOV     R2,#2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LCALL   DTSEC
MOV     DPTR,#DATA1
MOV     A,#85H
MOV     R2,#15
LCALL   LCDWDLn
MOV     DPTR,#DATA2
MOV     A,#0C5H
MOV     R2,#15
LCALL   LCDWDLn
MOV     R2,#2
LCALL   DTSEC
MOV     DPTR,#DATA1
MOV     A,#84H
MOV     R2,#16
LCALL   LCDWDLn
MOV     DPTR,#DATA2
MOV     A,#0C4H
MOV     R2,#16
LCALL   LCDWDLn
MOV     R2,#2
LCALL   DTSEC
MOV     DPTR,#DATA1
MOV     A,#83H
MOV     R2,#17
LCALL   LCDWDLn
MOV     DPTR,#DATA2
MOV     A;#0C3H
MOV     R2,#17
LCALL   LCDWDLn
MOV     R2,#2
LCALL   DTSEC
MOV     DPTR,#DATA1
MOV     A,#82H
MOV     R2,#18
LCALL   LCDWDLn
MOV     DPTR,#DATA2
MOV     A,#0C2H
MOV     R2,#18
LCALL   LCDWDLn
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     R2,#2
        LCALL   DTSEC
        MOV     DPTR,#DATA1
        MOV     A,#81H
        MOV     R2,#19
        LCALL   LCDWDLn
        MOV     DPTR,#DATA2
        MOV     A,#0C1H
        MOV     R2,#19
        LCALL   LCDWDLn
        MOV     R2,#2
        LCALL   DTSEC
        MOV     DPTR,#DATA1
        MOV     A,#80H
        MOV     R2,#20
        LCALL   LCDWDLn
        MOV     DPTR,#DATA2
        MOV     A,#0C0H
        MOV     R2,#20
        LCALL   LCDWDLn
        MOV     R2,#2
        LCALL   DTSEC
        MOV     R2,#10
        LCALL   DTSEC
                                ;**** RESET ****
RES7:   CLR     SCANBF          ;RESET VAR.
        MOV     TMOD,#20H       ;TIMER1 MODE2
        MOV     SCON,#52H       ;SERIAL 8 BIT UART MODE1
        MOV     TH1,#BRAT96     ;9600 BPS
        SETB    TR1             ;TIMER1 ON
        JB      INT1,RES8       ;CHECK REBOOT (BREAK KEY)
                                ;REBOOT
        JNB     INT1,$          ;WAIT FOR RELEASE KEY
        MOV     POWMEM,#0
        MOV     DPTR,#RES
        PUSH    DPL
        PUSH    DPH
        RETI                    ;CLEAR INT-ROUTINE
RES8:   LJMP    MON             ;TO MON MODE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ********** MON MAIN **********
; MONITOR MODE
MON:    CLR    A                    ;CLEAR SEGMENT
        MOV    DPTR,#CONA
        MOVX   @DPTR,A
        LCALL  CLEAR
        CLR    SCANBF
        MOV    A,#00001110B         ;DISPLAY ON/OFF
        LCALL  LCDWI
        LCALL  CLRLCD               ;CLEAR DISPLAY
        MOV    DPTR,#DATA33         ;"Select Mode 7"
        LCALL  LCDWDL1
MON2:   LCALL  SCAN
        LJMP   MON2
; ********** EDCH MAIN **********
; DATA CHANNEL EDIT
EDCH:   LCALL  CLRLCD
        MOV    DPTR,#DATA3
        LCALL  LCDWDL1
        LJMP   MON2
; ********** EDIT MAIN **********
; DATA MEMORY EDIT
EDIT:   LCALL  CLEAR
        LCALL  CLRLCD
        MOV    DPTR,#DATA4
        LCALL  LCDWDL1
        MOV    DPTR,#DATA22
        LCALL  L12WDL2
EDIT0:  MOV    DPH,DMMADD
        MOV    DPL,DMMADD+1
        MOV    VTREG+3,#0CCH        ;SET DDRAM ADDRESS
        LCALL  INW
        MOV    DMMADD,DPH
        MOV    DMMADD+1,DPL
        JB     INPBFG,EDIT0
        MOV    MEMADD,DMMADD
        MOV    MEMADD+1,DMMADD+1
EDIT1:  MOV    DPH,MEMADD           ;EDIT LOOP
        MOV    DPL,MEMADD+1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     HEXBUF,DPH
        MOV     HEXBUF+1,DPL
        MOVX    A,@DPTR
        MOV     HEXBUF+2,A          ;DATA MEM
        MOV     DPTR,#CONA
        MOVX    @DPTR,A             ;DATA LED
        LCALL   HTOA
        MOV     DISBUF+4,#COLON
        LCALL   CLRLCD
EDIT2:  LCALL   SCANDL1
        LCALL   SCAN
        CJNE    A,#11H,EDIT4        ;DEC KEY
        MOV     DPH,MEMADD
        MOV     DPL,MEMADD+1
        LCALL   DPDEC
        MOV     MEMADD,DPH
        MOV     MEMADD+1,DPL
        SJMP    EDIT1
EDIT4:  CJNE    A,#12H,EDIT44       ;INC KEY
EDIT41: MOV     DPH,MEMADD
        MOV     DPL,MEMADD+1
        INC     DPTR
        MOV     MEMADD,DPH
        MOV     MEMADD+1,DPL
        SJMP    EDIT1
EDIT44: CJNE    A,#10H,EDIT46       ;ENT KEY
        SJMP    EDIT1
EDIT46: CJNE    A,#13H,EDIT5        ;BACK KEY
        SJMP    EDIT1
EDIT5:  LCALL   GET2
        JB      GETOKF,EDIT6
        SJMP    EDIT2
EDIT6:  LCALL   ATOH                ;WRITE
        MOV     DPH,MEMADD
        MOV     DPL,MEMADD+1
        MOV     A,HEXBUF+2
        MOVX    @DPTR,A
        PUSH    DPH
        PUSH    DPL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          MOV     DPTR,#CONA
          MOVX    @DPTR,A
          POP     DPL
          POP     DPH
          MOVX    A,@DPTR             ;CHECK DATA OK
          CJNE    A,HEXBUF+2,EDIT9
          LCALL   SCANDL1
          MOV     R2,#3
          LCALL   DTSEC
          SJMP    EDIT41              ;TO INC
EDIT9:    LJMP    EDIT1               ;CROSS TO EDIT1
; ********** LSCH MAIN **********
; DATA CHANNEL LIST
LSCH:     LCALL   CLRLCD
          MOV     DPTR,#DATA5
          LCALL   LCDWDL1
          LJMP    MON2
; ********** LSRM MAIN **********
; DATA MEMORY LIST
LSRM:     LCALL   CLRLCD
          MOV     DPTR,#DATA6
          LCALL   LCDWDL1
          MOV     DPTR,#DATA22
          LCALL   L12WDL2
LSRM1:    MOV     DPH,DMMADD
          MOV     DPL,DMMADD+1
          MOV     VTREG+3,#0CCH       ;SET DDRAM ADDRESS
          LCALL   INW
          MOV     DMMADD,DPH
          MOV     DMMADD+1,DPL
          JB      INPBFG,LSRM1
          MOV     MEMADD,DMMADD
          MOV     MEMADD+1,DMMADD+1
          MOV     DISBUF+4,#COLON
LSRM2:    MOV     DPH,MEMADD          ;EDIT LOOP1
          MOV     DPL,MEMADD+1
          MOV     HEXBUF,DPH
          MOV     HEXBUF+1,DPL
          MOVX    A,@DPTR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     HEXBUF+2,A          ;DATA MEM 1
        INC     DPTR
        MOVX    A,@DPTR
        MOV     HEXBUF+3,A          ;DATA MEM 2
        INC     DPTR
        MOVX    A,@DPTR
        MOV     HEXBUF+4,A          ;DATA MEM 3
        INC     DPTR
        MOVX    A,@DPTR
        MOV     HEXBUF+5,A          ;DATA MEM 4
        INC     DPTR
        MOVX    A,@DPTR
        MOV     HEXBUF+6,A          ;DATA MEM 5
        LCALL   HTOA8
LSRM21: LCALL   CLRLCD
        LCALL   SCANDL1
LSRM3:  MOV     DPH,MEMADD          ;EDIT LOOP2
        MOV     DPL,MEMADD+1
        MOV     R2,#5
        LCALL   DPADDS
        MOV     HEXBUF,DPH
        MOV     HEXBUF+1,DPL
        MOVX    A,@DPTR
        MOV     HEXBUF+2,A          ;DATA MEM 1
        INC     DPTR
        MOVX    A,@DPTR
        MOV     HEXBUF+3,A          ;DATA MEM 2
        INC     DPTR
        MOVX    A,@DPTR
        MOV     HEXBUF+4,A          ;DATA MEM 3
        INC     DPTR
        MOVX    A,@DPTR
        MOV     HEXBUF+5,A          ;DATA MEM 4
        INC     DPTR
        MOVX    A,@DPTR
        MOV     HEXBUF+6,A          ;DATA MEM 5
        LCALL   HTOA8
LSRM31: LCALL   SCANDL2
LSRM32: LCALL   SCAN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          CJNE   A,#11H,LSRM4      ;DEC KEY
          MOV    DPH,MEMADD
          MOV    DPL,MEMADD+1
          MOV    R2,#10
          LCALL  DPSUBS
          MOV    MEMADD,DPH
          MOV    MEMADD+1,DPL
          LJMP   LSRM2
LSRM4:    CJNE   A,#12H,LSRM5      ;INC KEY
          MOV    DPH,MEMADD
          MOV    DPL,MEMADD+1
          MOV    R2,#10
          LCALL  DPADDS
          MOV    MEMADD,DPH
          MOV    MEMADD+1,DPL
          LJMP   LSRM2
LSRM5:    LJMP   LSRM2
; ********** NETW MAIN **********
; RS-485 (NETWORK)
NXB:      MOV    DPTR,#DATA7
          LCALL  LCDWDL1
          LCALL  INPSE
          LCALL  SURE
          LCALL  CLRLCD
          MOV    DPTR,#DATA31
          LCALL  LCDWDL1
          MOV    DPH,STTADD
          MOV    DPL,STTADD+1
          MOV    R2,ENDADD
          MOV    R3,ENDADD+1
NXB1:     MOVX   A,@DPTR
          LCALL  SBYTE
          LCALL  DPCOM
          INC    DPTR
          JNZ    NXB1
          LJMP   MON
; ********** FUNC MAIN **********
; SUB-FUNCTION
FUNC:     LCALL  CLEAR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL   CLRLCD
        MOV     DPTR,#DATA9         ;FUNCTION
        LCALL   LCDWDL1
FUNC2:  LCALL   SCAN
        MOV     R2,A
        CLR     C
        SUBB    A,#0CH              ;COMPARE
        JNC     FUNC2               ;=>
        MOV     DPTR,#FUNCTB
        MOV     B,#2                ;JUMP TO SUB-FUNCTION
        MOV     A,R2
        MUL     AB
        MOV     R2,A                ;MEM
        MOVC    A,@A+DPTR
        MOV     R3,A                ;H
        MOV     A,R2
        INC     ACC
        MOVC    A,@A+DPTR
        PUSH    ACC                 ;PUSH L
        MOV     A,R3
        PUSH    ACC                 ;PUSH H
        RET                         ;GOTO
FUNCTB: DW      BAUD,CHKS,MOVE,FIND
        DW      RXB,TXB,RECV,SEND
        DW      CLRDM1,CLRDM2,RTEST,LTEST
; ********** BAUD FUNCTION **********
; BAUD RATE
BAUD:   MOV     DPTR,#DATA10
        LCALL   LCDWDL1
        LCALL   CONF
BAUD0:  LCALL   CLRLCD
        MOV     DPTR,#DATA23
        LCALL   L12WDL1
        MOV     B,PCON              ;SMOD BIT
        MOV     A,TH1
        CJNE    A,#BRAT96,BAUD12
        JNB     SMOD,BAUD1
        MOV     A,#19H              ;19200
        MOV     DPL,#20H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SJMP    BAUD2
BAUD1:  MOV     A,#96H          ;9600
        MOV     DPL,#0
        SJMP    BAUD2
BAUD12: CJNE    A,#BRAT48,BAUD13
        MOV     A,#48H          ;4800
        MOV     DPL,#0
        SJMP    BAUD2
BAUD13: CJNE    A,#BRAT24,BAUD14
        MOV     A,#24H          ;2400
        MOV     DPL,#0
        SJMP    BAUD2
BAUD14: MOV     A,#12H          ;1200
        MOV     DPL,#0
BAUD2:  MOV     VTREG+3,#8CH    ;SET DDRAM ADDRESS
        MOV     DPH,A
BAUD21: LCALL   INW
        JB      INPBFG,BAUD0
        MOV     A,DPH
        CJNE    A,#19H,BAUD31
        MOV     A,DPL
        CJNE    A,#20H,BAUD31
        MOV     TH1,#BRAT96     ;19200
        SETB    SMOD
        SJMP    BAUD4
BAUD31: MOV     A,DPH
        CJNE    A,#96H,BAUD32
        MOV     A,DPL
        CJNE    A,#0,BAUD32
        MOV     TH1,#BRAT96     ;9600
        CLR     SMOD
        SJMP    BAUD4
BAUD32: MOV     A,DPH
        CJNE    A,#48H,BAUD33
        MOV     A,DPL
        CJNE    A,#0,BAUD33
        MOV     TH1,#BRAT48     ;4800
        SJMP    BAUD4
BAUD33: MOV     A,DPH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CJNE    A,#24H,BAUD34
        MOV     A,DPL
        CJNE    A,#0,BAUD34
        MOV     TH1,#BRAT24         ;2400
        SJMP    BAUD4
BAUD34: MOV     A,DPH
        CJNE    A,#12H,BAUD21
        MOV     A,DPL
        CJNE    A,#0,BAUD21
        MOV     TH1,#BRAT12         ;1200
BAUD4:  MOV     PCON,B
        LJMP    MON
; ********** CHKS FUNCTION **********
; CHECK SUM
CHKS:   MOV     DPTR,#DATA11
        LCALL   LCDWDL1
        LCALL   CONF
        LCALL   CLR2L
        LCALL   INPSE
        MOV     R2,STTADD
        MOV     R3,STTADD+1
        MOV     R4,ENDADD
        MOV     R5,ENDADD+1
        LCALL   CHKSUM
        MOV     HEXBUF,A
        MOV     R0,#HEXBUF
        MOV     R1,#DISBUF
        LCALL   HTOAS
        LCALL   CLR1L
        MOV     DPTR,#DATA24
        LCALL   L12WDL1
        MOV     A,#8DH
        LCALL   LCDWI
        MOV     A,DISBUF
        LCALL   LCDWD
        MOV     A,#8EH
        LCALL   LCDWI
        MOV     A,DISBUF+1
        LCALL   LCDWD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          MOV     DPTR,#DATA29
          LCALL   LCDWDL2
CHKS1:    LCALL   SCAN
          CJNE    A,#10H,CHKS1
          LJMP    MON
; ********** MOVE FUNCTION **********
; MOVE DATA MEMORY
MOVE:     MOV     DPTR,#DATA12
          LCALL   LCDWDL1
          LCALL   CONF
          LCALL   INPSED
          LCALL   CLRLCD
          MOV     DPTR,#DATA31
          LCALL   LCDWDL1
          MOV     R2,STTADD
          MOV     R3,STTADD+1
          MOV     R4,ENDADD
          MOV     R5,ENDADD+1
          MOV     R6,DESADD
          MOV     R7,DESADD+1
          LCALL   MOVES
          LJMP    MON
; ********** FIND FUNCTION **********
; FIND DATA MEMORY
FIND:     MOV     DPTR,#DATA13
          LCALL   LCDWDL1
          LCALL   CONF
          MOV     DPTR,#DATA25
          LCALL   L12WDL2
          MOV     DPH,STTADD
          MOV     DPL,STTADD+1
          MOV     VTREG+3,#0CCH      ;SET DDRAM ADDRESS
          LCALL   INW
          MOV     STTADD,DPH
          MOV     STTADD+1,DPL
          JB      INPBFG,FIND
FIND1:    MOV     DPTR,#DATA26
          LCALL   L12WDL2
          MOV     DPH,ENDADD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     DPL,ENDADD+1
        MOV     VTREG+3,#0CCH       ;SET DDRAM ADDRESS
        LCALL   INW
        MOV     ENDADD,DPH
        MOV     ENDADD+1,DPL
        JB      INPBFG,FIND
FIND2:  LCALL   CLR2L
        MOV     DPTR,#DATA28
        LCALL   L12WDL2
        MOV     A,HEXDAT
        MOV     VTREG+3,#0CCH       ;SET DDRAM ADDRESS
        LCALL   INH
        MOV     HEXDAT,A
        JZ      FIND2
        CJNE    A,#5,$+3
        JNC     FIND2               ;=>5
        JB      INPBFG,FIND1
        MOV     B,HEXDAT
        MOV     FINDAD,#VTREG
FIND3:  MOV     A,HEXDAT
        INC     A
        CLR     C
        SUBB    A,B
        MOV     R2,A
        LCALL   HTOAX
        MOV     R2,A                ;CHAR
        MOV     R0,FINDAD
        MOV     A,@R0               ;DATA
        LCALL   INH
        MOV     R0,FINDAD
        MOV     @R0,A
        JB      INPBFG,FIND2
        INC     FINDAD              ;NEXT
        DEC     B
        MOV     A,B
        JNZ     FIND3
        MOV     DPH,STTAD
        MOV     DPL,STTADD+1
FIND5:  PUSH    DPH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          PUSH    DPL
          MOV     R2,HEXDAT          ;COMPARE STRING
          MOV     R0,#VTREG
FIND51:   MOVX    A,@DPTR
          CLR     C
          SUBB    A,@R0
          JNZ     FIND7              ;NOT EQUAL
          INC     DPTR
          INC     R0
          DJNZ    R2,FIND51
          POP     DPL                ;DISPLAY
          POP     DPH
          PUSH    DPH
          PUSH    DPL
          MOV     HEXBUF,DPH
          MOV     HEXBUF+1,DPL
          MOVX    A,@DPTR
          MOV     HEXBUF+2,A
          LCALL   HTOA
          ORL     DISBUF+3,#80H
FIND6:    LCALL   SCAN
          CJNE    A,#12H,FIND6
FIND7:    POP     DPL
          POP     DPH
          MOV     R2,ENDADD
          MOV     R3,ENDADD+1
          LCALL   DPCOM
          INC     DPTR
          JNZ     FIND5
          LJMP    MON
; ********** RXB FUNCTION ***********
; RECEIVE DATA FROM SINGLE BOARD
RXB:      MOV     DPTR,#DATA14
          LCALL   LCDWDL1
          LCALL   CONF               ;WAIT ENTER KEY
          MOV     DPTR,#DATA27
          LCALL   L12WDL2
RXB1:     MOV     DPH,STTADD         ;START ADDRESS
          MOV     DPL,STTADD+1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     VTREG+3,#0CCH       ;SET DDRAM ADDRESS
        LCALL   INW
        MOV     A,DPH
        SUBB    A,#80H
        JNC     RXB11               ;=>
        SJMP    RXB1
RXB11:  MOV     STTADD,DPH
        MOV     STTADD+1,DPL
        JB      INPBFG,RXB1
        LCALL   CLRLCD
        LCALL   WAITR
        CLR     RI                  ;CLEAR OLD DATA
RXB2:   JNB     INT1,RXB31          ;CHECK FIRST BYTE
        JNB     RI,RXB2
        MOV     DPH,STTADD          ;ADDRESS
        MOV     DPL,STTADD+1
RXB3:   JB      RI,RXB4             ;RX OK
        JB      INT1,RXB3
RXB31:  LJMP    MON
RXB4:   LCALL   RBYTE               ;READ
        MOVX    @DPTR,A             ;WRITE TO DATA MEM
        MOV     ENDADD,DPH          ;END ADDRESS MEMORY
        MOV     ENDADD+1,DPL
        INC     DPTR
        PUSH    DPH                 ;DISPLAY LED
        PUSH    DPL
        MOV     DPTR,#CONA          ;SEGMENT PORT
        MOVX    @DPTR,A             ;DISPLAY
        POP     DPL
        POP     DPH
        LCALL   SBYTE               ;ECHO
        SJMP    RXB3
; ********** TXB FUNCTION ***********
; TRANSMIT DATA TO SINGLE-BOARD
TXB:    MOV     DPTR,#DATA15
        LCALL   LCDWDL1
        LCALL   CONF                ;WAIT ENTER KEY
        LCALL   INPSE
        LCALL   SURE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL   CLRLCD
        MOV     DPTR,#DATA31
        LCALL   LCDWDL1
        MOV     DPH,STTADD
        MOV     DPL,STTADD+1
        MOV     R2,ENDADD
        MOV     R3,ENDADD+1
TXB1:   MOVX    A,@DPTR
        LCALL   SBYTE
        LCALL   DPCOM
        INC     DPTR
        JNZ     TXB1
        LJMP    MON
; ********** RECV FUNCTION **********
; RECEIVE DATA FROM COMPUTER
RECV:   MOV     DPTR,#DATA16
        LCALL   LCDWDL1
        LCALL   CONF
        LCALL   CLRLCD
        MOV     DPTR,#DATA34        ;"Wait Receive Data"
        LCALL   LCDWDL1
        CLR     RI                  ;CLEAR OLD DATA
        MOV     TEMADD,#0
        MOV     TEMADD+1,#0
        CLR     STMFAG
RECV0:  JB      INT1,RECV01
        LJMP    MON
RECV01: JNB     RI,RECV0
        LCALL   RBYTE               ;FIRST BYTE
        CJNE    A,#0DH,$+6          ;DON'T CARE 0DH
        LJMP    RECV0
        CJNE    A,#0AH,$+6          ;DON'T CARE 0AH
        LJMP    RECV0
        CJNE    A,#'O',$+6
        LJMP    RECV0
        CJNE    A,#'S',$+6
        LJMP    RECVS
        CJNE    A,#':',$+6
        LJMP    RECV11
        LJMP    RECVE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RECV1:  LCALL   RBYTE
        CJNE    A,#':',RECV1
RECV11: MOV     R7,#0           ;CLEAR CHECKSUM
        LCALL   RBYTEHC         ;BYTE COUNT
        MOV     R5,A            ;COUNTER
        LCALL   RBYTEHC
        MOV     DPH,A           ;ADDRESS HIGH
        LCALL   RBYTEHC
        MOV     DPL,A           ;ADDRESS LOW
        LCALL   RBYTEHC
        CJNE    A,#00H,RECV8    ;END OF FILE
        JNB     STMFAG,RECV15   ;CHECK FIRST STREAM-OFFSET ADDRESS
        MOV     R2,DPH
        MOV     R3,DPL
        MOV     DPH,TEMADD
        MOV     DPL,TEMADD+1
        LCALL   DPSUB
        MOV     TEMADD,DPH
        MOV     TEMADD+1,DPL
        MOV     DPH,R2
        MOV     DPL,R3
        CLR     STMFAG
RECV15: MOV     R2,TEMADD       ;OFFSET PROCESS
        MOV     R3,TEMADD+1
        LCALL   DPADD
RECV2:  LCALL   RBYTEHC         ;DATA LOOP
        MOVX    @DPTR,A
        PUSH    DPH
        PUSH    DPL
        MOV     DPTR,#CONA      ;SEGMENT PORT
        MOVX    @DPTR,A         ;DISPLAY LED
        POP     DPL
        POP     DPH
        INC     DPTR
        DJNZ    R5,RECV2
        MOV     A,R7            ;CHECKSUM
        CPL     A
        INC     A
        MOV     B,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL   RBYTEHC
        CJNE    A,B,RECVE
        LCALL   RBYTE           ;CR
        LCALL   RBYTE           ;LF
        SJMP    RECV1
RECV8:  MOV     A,R7            ;END OF FILE
        CPL     A
        INC     A
        MOV     B,A
        LCALL   RBYTEH
        CJNE    A,B,RECVE
        LCALL   RBYTE           ;CR
        LCALL   RBYTE           ;LF
        LJMP    MON
RECVE:  MOV     DPTR,#DATA35    ;"Data Error"
        LCALL   LCDWDL1
        MOV     R2,#10
        LCALL   DTSEC
        LJMP    MON
RECVS:  SETB    STMFAG          ;STREAM RECEIVE
RECVO:  LCALL   RBYTEH          ;OFFSET RECEIVE
        MOV     TEMADD,A        ;ADDRESS HIGH
        LCALL   RBYTEH
        MOV     TEMADD+1,A      ;ADDRESS LOW
        LCALL   RBYTE           ;CR
        LCALL   RBYTE           ;LF
        LJMP    RECVO
; ********** SEND FUNCTION **********
; SEND DATA TO COMPUTER
SEND:   MOV     DPTR,#DATA17
        LCALL   LCDWDL1
        LCALL   CONF            ;WAIT ENTER KEY
        LCALL   INPSE
        MOV     DPTR,#DATA27
        LCALL   L12WDL2
        MOV     DPH,STTADD
        MOV     DPL,STTADD+1
        MOV     VTREG+3,#0CCH   ;SET DDRAM ADDRESS
        LCALL   INW
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     R2,STTADD
        MOV     R3,STTADD+1
        LCALL   DPSUB
        MOV     TEMADD,DPH
        MOV     TEMADD+1,DPL
        LCALL   SURE
        LCALL   CLRLCD
        MOV     DPTR,#DATA31
        LCALL   LCDWDL1
        MOV     DPH,STTADD
        MOV     DPL,STTADD+1
        CLR     SENDEN              ;END FLAG
SEND0:  MOV     R7,#0               ;START CHECK SUM
        MOV     A,#':'              ;FIRST BYTE
        LCALL   SBYTE
        LCALL   SENDC               ;CHECK LOOP COUNTER
        MOV     A,R6
        LCALL   SBYTEHC
        PUSH    DPH                 ;-[
        PUSH    DPL
        MOV     R2,TEMADD
        MOV     R3,TEMADD+1
        LCALL   DPADD
        MOV     A,DPH
        LCALL   SBYTEHC             ;ADDRESS
        MOV     A,DPL
        LCALL   SBYTEHC
        POP     DPL
        POP     DPH                 ;-]
        CLR     A
        LCALL   SBYTEHC             ;00
SEND1:  MOVX    A,@DPTR
        LCALL   SBYTEHC             ;DATA
        INC     DPTR
        DJNZ    R6,SEND1
        MOV     A,R7                ;CHECKSUM
        CPL     A
        INC     A
        LCALL   SBYTEH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          LCALL   SLF
          JNB     SENDEN,SEND0
          MOV     DPTR,#SENDTB
          LCALL   SBLOCK
          LJMP    MON
SENDC:    MOV     A,DPH               ;CHECK LINE LOOP
          XCH     A,R2
          MOV     DPH,A
          MOV     A,DPL
          XCH     A,R3
          MOV     DPL,A
          MOV     DPH,ENDADD
          MOV     DPL,ENDADD+1
          LCALL   DPSUB
          MOV     A,DPH
          JNZ     SENDC9
          MOV     A,DPL
          CJNE    A,#0FH,$+6
          LJMP    SENDC8
          JNC     SENDC9
          SETB    SENDEN              ;LAST LINE (<10H LOOP)
          MOV     R6,DPL
          INC     R6
          MOV     DPH,R2
          MOV     DPL,R3
          RET
SENDC8:   SETB    SENDEN              ;LAST LINE (10H LOOP)
SENDC9:   MOV     DPH,R2              ;10H LOOP
          MOV     DPL,R3
          MOV     R6,#10H             ;LOOP COUNTER
          RET
SENDTB:   DB      ":00000001FF",0DH
; ********* CLRDM1 FUNCTION *********
; CLEAR RAM U4
CLRDM1:   MOV     DPTR,#DATA18
          LCALL   LCDWDL1
          LCALL   SURE
          LCALL   WAIT
          MOV     DPTR,#0             ;CLEAR PROCESS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          MOV     R2,#7FH
          MOV     R3,#0FFH
CLRDM4:   CLR     A
          MOVX    @DPTR,A
          LCALL   DPCOM
          INC     DPTR
          JNZ     CLRDM4
          LJMP    MON
; ********** CLRDM2 FUNCTION **********
; CLEAR RAM U5
CLRDM2:   MOV     DPTR,#DATA19
          LCALL   LCDWDL1
          LCALL   SURE
          LCALL   WAIT
          MOV     DPTR,#8000H       ;CLEAR PROCESS
          MOV     R2,#0DFH
          MOV     R3,#0FFH
CLRDM5:   CLR     A
          MOVX    @DPTR,A
          LCALL   DPCOM
          INC     DPTR
          JNZ     CLRDM5
          LJMP    MON
; ********** RTEST FUNCTION ***********
; RAM TEST
RTEST:    MOV     DPTR,#DATA20      ;TEST RAM U3/U4
          LCALL   LCDWDL1
          LCALL   CONF
          MOV     TEMADD,#00H
          MOV     TEMADD+1,#00H
          MOV     ENDADD,#0DFH
          MOV     ENDADD+1,#0FFH
          LCALL   CLRLCD
          MOV     A,#00001100B      ;OFF CURSOR
          LCALL   LCDWI
          LCALL   RAMT
          MOV     A,#00001110B      ;ON CURSOR
          LCALL   LCDWI
          LCALL   PROCO
          LJMP    MON
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RAMT:   MOV     HEXBUF,TEMADD
        MOV     HEXBUF+1,TEMADD+1
        LCALL   HTOA
        MOV     VTREG+2,#4
        MOV     VTREG+3,#80H
        LCALL   SCNDL24
        MOV     R2,#130
        LCALL   DMSEC
        MOV     DPH,TEMADD          ;LOAD FF-00
        MOV     DPL,TEMADD+1
        MOV     R2,ENDADD
        MOV     R3,ENDADD+1
        MOV     R7,#0
RAMT1:  MOV     A,R7
        MOVX    @DPTR,A
        LCALL   DPCOM
        JZ      RAMT2
        INC     DPTR
        DJNZ    R7,RAMT1
RAMT2:  MOV     DPH,TEMADD          ;CHECK FF-00
        MOV     DPL,TEMADD+1
        MOV     R7,#0
RAMT21: MOVX    A,@DPTR
        CJNE    A,07H,RAMT3         ;COMPARE R7
        CLR     A
        MOVX    @DPTR,A             ;CLEAR RAM
RAMT22: MOV     R2,ENDADD
        MOV     R3,ENDADD+1
        LCALL   DPCOM
        JZ      RAMT4
        INC     DPTR
        DJNZ    R7,RAMT21
        MOV     TEMADD,DPH          ;NEXT BLOCK
        MOV     TEMADD+1,DPL
        SJMP    RAMT                ;BACK LOOP
RAMT3:  MOV     RAMTMM,DPH          ;DISPLAY ERROR ADDRESS
        MOV     RAMTMM+1,DPL
        MOV     RAMTMM+2,R7
        MOV     HEXBUF,DPH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     HEXBUF+1,DPL
        MOVX    A,@DPTR
        MOV     HEXBUF+2,A
        LCALL   HTOA
        MOV     DISBUF+6,#20H
        MOV     DISBUF+7,#20H
        LCALL   CLRLCD
        LCALL   SCANDL1
RAMT31: LCALL   SCAN
        CJNE    A,#12H,RAMT31
        MOV     R7,RAMTMM+2
        MOV     DPL,RAMTMM+1
        MOV     DPH,RAMTMM
        SJMP    RAMT22
RAMT4:  RET
; ********* LTEST FUNCTION **********
; LCD TEST
LTEST:  MOV     DPTR,#DATA21
        LCALL   LCDWDL1
        LCALL   CONF
        MOV     R2,#80H
        MOV     R3,#20H
LTEST1: MOV     A,R2
        LCALL   LCDWI
        MOV     A,#0FFH
        LCALL   LCDWD
        INC     R2
        DJNZ    R3,LTEST1
        MOV     R2,#0C0H
        MOV     R3,#20H
LTEST2: MOV     A,R2
        LCALL   LCDWI
        MOV     A,#0FFH
        LCALL   LCDWD
        INC     R2
        DJNZ    R3,LTEST2
        MOV     R2,#20
        LCALL   DTSEC
        LJMP    MON
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ********** LCDWI SUB **********
; WRITE INSTRUCTION TO LCD MODULE
LCDWI:   MOV    DPTR,#LCMWI
         MOVX   @DPTR,A
         LCALL  LCDRB              ;WAIT LCD MODULE READY
         RET
; ********** LCDRD SUB **********
; READ DATA FROM LCD MODULE
LCDRD:   MOV    DPTR,#LCMRD
         MOVX   A,@DPTR
         LCALL  LCDRB              ;WAIT LCD MODULE READY
         RET
; ********** LCDRB SUB **********
; WAIT FOR LCD MODULE READY
; BY MEAN OF CHECK BUSY FLAG
LCDRB:   MOV    DPTR,#LCMRB
LCDRB1:  MOVX   A,@DPTR
         JB     ACC.7,LCDRB1       ;BUSY FLAG
         RET
; ********** LCDWD SUB **********
; WRITE ASCII TO LCD MODULE
LCDWD:   MOV    DPTR,#LCMWD
         MOVX   @DPTR,A
         LCALL  LCDRB              ;WAIT LCD MODULE READY
         RET
; ********* CLRLCD SUB **********
; CLEAR DISPLAY LCD
CLRLCD:  MOV    A,#01H
         LCALL  LCDWI
         RET
; ********** CLR1L SUB **********
; WRITE SPACE TO LCD (LINE1)
CLR1L:   MOV    DPTR,#DATA36
         MOV    A,#80H
         MOV    R2,#20
         LCALL  LCDWDLn
         RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ********** CLR2L SUB **********
; WRITE SPACE TO LCD (LINE2)
CLR2L:   MOV    DPTR,#DATA36
         MOV    A,#0C0H
         MOV    R2,#20
         LCALL  LCDWDLn
         RET
; ********* LCDWDLn SUB *********
; WRITE ASCII TO LCD MODULE (LINEn)
LCDWDLn: PUSH   DPH
         PUSH   DPL
         LCALL  LCDWI
         POP    DPL
         POP    DPH
LCDWDL5: CLR    A
         MOVC   A,@A+DPTR
         PUSH   DPH
         PUSH   DPL
         LCALL  LCDWD
         POP    DPL
         POP    DPH
         INC    DPTR
         DJNZ   R2,LCDWDL5
         RET
; ********* LCDWDL1 SUB *********
; WRITE ASCII TO LCD MODULE (LINE1)
LCDWDL1: MOV    A,#80H
         MOV    R2,#20
         LCALL  LCDWDLn
         RET
; ********* LCDWDL2 SUB *********
; WRITE ASCII TO LCD MODULE (LINE2)
LCDWDL2: MOV    A,#0C0H
         MOV    R2,#20
         LCALL  LCDWDLn
         RET
; ********* L12WDL1 SUB *********
; WRITE ASCII TO LCD MODULE (LINE1)
L12WDL1: MOV    A,#80H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          MOV     R2,#12
          LCALL   LCDWDLn
          RET
; ********* L12WDL2 SUB *********
; WRITE ASCII TO LCD MODULE (LINE2)
L12WDL2: MOV     A,#0C0H
          MOV     R2,#12
          LCALL   LCDWDLn
          RET
; ********** PROCO SUB **********
; PROCESS COMPLETE & WAIT ENTER KEY
PROCO:    MOV     DPTR,#DATA32
          LCALL   LCDWDL1
          MOV     DPTR,#DATA29
          LCALL   LCDWDL2
PROCO2:   LCALL   SCAN
          CJNE    A,#10H,PROCO2
          RET
; ********** CONF SUB ***********
; CONFIRM DISPLAY & WAIT ENTER KEY
CONF:     MOV     DPTR,#DATA29
          LCALL   LCDWDL2
CONF1:    LCALL   SCAN
          CJNE    A,#10H,CONF1
          RET
; ********** SURE SUB ***********
; SURE DISPLAY & WAIT ENTER KEY
SURE:     MOV     DPTR,#DATA30
          LCALL   LCDWDL2
SURE1:    LCALL   SCAN
          CJNE    A,#10H,SURE1
          RET
; ********** WAIT SUB ***********
; PLEASE WAIT DISPLAY
WAIT:     LCALL   CLRLCD
          MOV     DPTR,#DATA31
          LCALL   LCDWDL1
          RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ********** WAITR SUB **********
; PLEASE WAIT DISPLAY FOR RECEIVE
WAITR:   LCALL  CLRLCD
         MOV    DPTR,#DATA34
         LCALL  LCDWDL1
         RET
; ********** CLEAR SUB **********
; CLEAR HEXBUF & DISBUF
CLEAR:   MOV    HEXBUF,#0
         MOV    HEXBUF+1,#0
         MOV    HEXBUF+2,#0
         MOV    HEXBUF+3,#0
         MOV    R0,#DISBUF
         MOV    R2,#20
CLEAR1:  MOV    @R0,#20H
         INC    R0
         DJNZ   R2,CLEAR1
         RET
; ********** DMSEC SUB **********
; DELAY 1/1000 SECOND
DMSEC:   MOV    R3,#230          ;1 MSEC LOOP
DMSEC1:  NOP
         NOP
         DJNZ   R3,DMSEC1
         DJNZ   R2,DMSEC
         RET
; ********** DTSEC SUB **********
; DELAY 1/10 SECOND
DTSEC:   MOV    R3,#179
DTSEC1:  MOV    R4,#0
         DJNZ   R4,$
         NOP
         NOP
         DJNZ   R3,DTSEC1
         DJNZ   R2,DTSEC
         RET
; ********** CHKSUM SUB *********
; CHKSUM FIND
CHKSUM:  MOV    DPH,R2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          MOV     DPL,R3
          MOV     02H,R4
          MOV     03H,R5
          MOV     R0,#0
CHKSUM1:  MOVX    A,@DPTR
          ADD     A,R0
          MOV     R0,A
          LCALL   DPCOM
          INC     DPTR
          JNZ     CHKSUM1
          MOV     A,R0
          CPL     A
          INC     A
          RET
; ********** MOVES SUB **********
; MOVE DATA MEMORY
MOVES:    MOV     DPH,R6
          MOV     DPL,R7
          LCALL   DPCOM
          JC      MOVES4
          JZ      MOVES4
          MOV     DPH,R4          ;***** START < DEST *****
          MOV     DPL,R5          ;MOVE ADDRESS DOWN
          LCALL   DPSUB           ;DPTR=END-START
          MOV     A,R7            ;R6,R7=R6,R7+DPTR
          ADD     A,DPL
          MOV     R7,A
          MOV     A,R6
          ADDC    A,DPH
          MOV     R6,A
MOVES2:   MOV     DPH,R4
          MOV     DPL,R5
          MOVX    A,@DPTR
          MOV     DPH,R6
          MOV     DPL,R7
          MOVX    @DPTR,A
          MOV     DPH,R4
          MOV     DPL,R5
          LCALL   DPCOM
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          JNZ     MOVES21
          RET                          ;FINISH
MOVES21:  MOV     DPH,R4
          MOV     DPL,R5
          LCALL   DPDEC
          MOV     R4,DPH
          MOV     R5,DPL
          MOV     DPH,R6
          MOV     DPL,R7
          LCALL   DPDEC
          MOV     R6,DPH
          MOV     R7,DPL
          SJMP    MOVES2
MOVES4:   MOV     DPH,R2             ;***** START => DEST *****
          MOV     DPL,R3             ;MOVE ADDRESS UP
          MOVX    A,@DPTR
          MOV     DPH,R6
          MOV     DPL,R7
          MOVX    @DPTR,A
          MOV     DPH,R4
          MOV     DPL,R5
          LCALL   DPCOM
          JNZ     MOVES41
          RET                          ;FINISH
MOVES41:  MOV     DPH,R2             ;NEXT ADDRESS
          MOV     DPL,R3
          INC     DPTR
          MOV     R2,DPH
          MOV     R3,DPL
          MOV     DPH,R6
          MOV     DPL,R7
          INC     DPTR
          MOV     R6,DPH
          MOV     R7,DPL
          SJMP    MOVES4
; ********** DPDEC SUB **********
; DPTR = DPTR - 1
DPDEC:    XCH     A,DPL
          JNZ     $+4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          DEC     DPH
          DEC     A
          XCH     A,DPL
          RET
; ********** DPADDS SUB *********
; DPTR = DPTR + R2
DPADDS:   MOV     A,DPL
          ADD     A,R2
          MOV     DPL,A
          MOV     A,DPH
          ADDC    A,#0
          MOV     DPH,A
          RET
; ********** DPADD SUB **********
; DPTR = DPTR + R2,R3
DPADD:    MOV     A,DPL
          ADD     A,R3
          MOV     DPL,A
          MOV     A,DPH
          ADDC    A,R2
          MOV     DPH,A
          RET
; ********** DPSUBS SUB *********
; DPTR = DPTR - R2
DPSUBS:   CLR     C
          MOV     A,DPL
          SUBB    A,R2
          MOV     DPL,A
          MOV     A,DPH
          SUBB    A,#0
          MOV     DPH,A
          RET
; ********** DPSUB SUB **********
; DPTR = DPTR - R2,R3
DPSUB:    CLR     C
          MOV     A,DPL
          SUBB    A,R3
          MOV     DPL,A
          MOV     A,DPH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          SUBB    A,R2
          MOV     DPH,A
          RET
; ********** DPCOM SUB **********
; COMPARE WORD (16 BIT) DPTR <> R2,R3
DPCOM:    PUSH    DPL
          CLR     C
          MOV     A,DPL
          SUBB    A,R3
          MOV     DPL,A
          MOV     A,DPH
          SUBB    A,R2
          ORL     A,DPL
          POP     DPL
          RET
; *********** INW SUB ***********
; INPUT 4 DIGIT HEX
INW:      MOV     HEXBUF+0,DPH
          MOV     HEXBUF+1,DPL
          MOV     INPBKH,DPH          ;BACKUP
          MOV     INPBKL,DPL
          LCALL   HTOA
          SETB    INPFIR
          SETB    SCANFF
INW1:     MOV     VTREG+2,#4
INW10:    LCALL   SCNDL24
INW11:    LCALL   SCAN                ;WAIT KEY
          MOV     R2,A
          CLR     C
          SUBB    A,#10H
          MOV     A,R2
          JNC     INW5                ;>F
          MOV     R2,A
          LCALL   HTOAX               ;CHANGE TO ASCII CODE
          MOV     R2,A
          JB      INPFIR,INW3         ;FIRST DIGIT PROCESS
          MOV     R0,#DISBUF-1        ;NEXT DIGIT PROCESS
          MOV     R1,#0
INW2:     INC     R0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        INC     R1
        MOV     A,@R0
        CJNE    A,#20H,INW2
        MOV     A,R2                ;SPACE DIGIT
        MOV     @R0,A
        CJNE    R1,#4,INW21
        SETB    INPFIR
        LJMP    INW1
INW21:  MOV     A,R1
        MOV     VTREG+2,A
        LJMP    INW10
INW3:   MOV     A,#20H              ;FIRST DIGIT PROCESS
        MOV     DISBUF+0,R2
        MOV     DISBUF+1,A
        MOV     DISBUF+2,A
        MOV     DISBUF+3,A
        MOV     VTREG+2,#4
        LCALL   SCNDL24
        MOV     A,#00010000B        ;SHIF DISPLAY
        LCALL   LCDWI
        MOV     A,#00010000B        ;SHIF DISPLAY
        LCALL   LCDWI
        MOV     A,#00010000B        ;SHIF DISPLAY
        LCALL   LCDWI
        CLR     INPFIR
        LJMP    INW11
INW5:   CJNE    R2,#11H,INW6        ;CHECK DEC KEY
        MOV     R0,#DISBUF+3        ;IS DEC KEY
        MOV     R1,#4
INW51:  MOV     A,@R0
        CJNE    A,#20H,INW55
        DEC     R0
        DJNZ    R1,INW51
        MOV     VTREG+2,#1
        LCALL   SCNDL24
        MOV     A,#00010000B        ;SHIF DISPLAY
        LCALL   LCDWI
        LJMP    INW11
INW55:  MOV     @R0,#20H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,R1
        MOV     VTREG+2,A
        LCALL   SCNDL24
        MOV     A,#00010000B        ;SHIF DISPLAY
        LCALL   LCDWI
        CLR     INPFIR
        LJMP    INW11
INW6:   CJNE    R2,#12H,INW8        ;CHECK INC KEY
        MOV     DPH,INPBKH          ;IS INC KEY
        MOV     DPL,INPBKL
        LJMP    INW
INW8:   CLR     INPBFG
        CJNE    R2,#13H,INW81
        SETB    INPBFG
INW81:  LCALL   ATOH                ;EXIT
        MOV     DPH,HEXBUF+0
        MOV     DPL,HEXBUF+1
        CLR     SCANFF
        RET
; ********** INPSED SUB *********
; INPUT START,END,DEST
INPSED: MOV     DPTR,#DATA25
        LCALL   L12WDL2
        MOV     DPH,STTADD
        MOV     DPL,STTADD+1
        MOV     VTREG+3,#0CCH       ;SET DDRAM ADDRESS
        LCALL   INW
        MOV     STTADD,DPH
        MOV     STTADD+1,DPL
        JB      INPBFG,INPSED
INPSED1: MOV    DPTR,#DATA26
        LCALL   L12WDL2
        MOV     DPH,ENDADD
        MOV     DPL,ENDADD+1
        MOV     VTREG+3,#0CCH       ;SET DDRAM ADDRESS
        LCALL   INW
        MOV     ENDADD,DPH
        MOV     ENDADD+1,DPL
        JB      INPBFG,INPSED
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SCAN3:  ORL     A,R6                ;PRESS
        JNB     SCANPF,SCAN5        ;IS NEW PRESS
        MOV     R1,A                ;INKEY MEM
        MOV     A,SCANKM            ;CHECK AUTO REPEAT
        CJNE    A,#11H,$+6
        LJMP    SCAN4
        CJNE    A,#12H,$+6
        LJMP    SCAN4
```



```
        MOV     DPTR,#DATA27
        LCALL   L12WDL2
        MOV     DPH,DESADD
        MOV     DPL,DESADD+1
        MOV     VTREG+3,#0CCH       ;SET DDRAM ADDRESS
        LCALL   INW
        MOV     DESADD,DPH
        MOV     DESADD+1,DPL
        JB      INPBFG,INPSED1
        RET
; ********** SCAN SUB ***********
; SCAN DISPLAY AND WAIT FOR KEY

SCAN:   MOV     R4,#8               ;LOOP
        MOV     R5,#8               ;RELEASE KEY COUNT
        MOV     R6,#0               ;KEY-TABLE COUNT
SCAN1:  MOV     DPTR,#CONB          ;SCAN COLUMN
        MOVX    A,@DPTR
        ANL     A,#0F8H
        ORL     A,R6
        MOVX    @DPTR,A             ;OUT SCAN COLUMN
        INC     R6
        CJNE    R6,#8,$+6
        LJMP    SCAN11
        MOV     DPTR,#CONC
        MOVX    A,@DPTR             ;INKEY
        ANL     A,#0F0H
        CJNE    A,#0F0H,SCAN3       ;PRESS CHECK
SCAN11: DJNZ    R5,SCAN2
        CLR     SCANPF
SCAN2:  DJNZ    R4,SCAN1
        PUSH    2
        PUSH    3
        MOV     R2,#6               ;DELAY TIME
        LCALL   DMSEC
        POP     3
        POP     2
        LCALL   SCANFH
        MOV     A,#0FFH
        RET                         ;ONE LOOP SCAN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          SUBB    A,#14H
          JNC     SCAN8              ;=> TO DIRECT KEYS
SCAN77:   MOV     A,R2
          RET                        ;EXIT NEW PRESS
SCAN8:    JB      SCANBF,SCAN77      ;BY PASS FLAG
          MOV     A,R2               ;IS DIRECT KEYS
          MOV     SP,#STACK          ;RESET STACK
          MOV     DISBUF+6,#20H
          MOV     DISBUF+7,#20H
          CJNE    A,#15H,$+6
          LJMP    SCAN
          CJNE    A,#16H,$+6
          LJMP    NXB
          CJNE    A,#17H,$+6
          LJMP    FUNC
          CJNE    A,#18H,$+6
          LJMP    LSCH
          CJNE    A,#19H,$+6
          LJMP    LSRM
          CJNE    A,#1AH,$+6
          LJMP    EDCH
          CJNE    A,#1BH,$+6
          LJMP    EDIT
          LJMP    MON                ;14H
KEYTAB:   DB      0E2H,0E1H,0D2H,0D1H    ;1B,1A,19,18
          DB      0B2H,0B1H,072H,071H    ;17,16,15,14
          DB      0E7H,0D7H,0B7H,077H    ;13,12,11,10
          DB      0E6H,0E5H,0E4H,0E3H    ;F,E,D,C
          DB      0D6H,0D5H,0D4H,0D3H    ;B,A,9,8
          DB      0B6H,0B5H,0B4H,0B3H    ;7,6,5,4
          DB      076H,075H,074H,073H    ;3,2,1,0
; ***** SCANUP/DN XSUB *****
; UP/DOWN AUTO REPEAT SPEED
SCANUP:   MOV     A,SCANAM           ;AUTO REPEAT SPEED-UP
          CJNE    A,#AUTORL,SCANUP1
          RET
SCANUP1:  DEC     SCANAM
          DEC     SCANAM
          RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SCANDN:  MOV    A,SCANAM          ;AUTO REPEAT SPEED-DOWN
         CJNE   A,#AUTORH,SCANDN1
         RET
SCANDN1: INC    SCANAM
         INC    SCANAM
         RET
; ******* SCANFH SUB *******
; FLASH CURSOR
SCANFH:  DEC    SCANFC
         MOV    A,SCANFC
         JZ     SCANH
         RET                      ;NOTING
SCANH:   MOV    SCANFC,#FLASHC    ;DELAY FINISH
         CPL    SCANFD
         MOV    C,SCANFD
         JC     SCANH1
         MOV    A,#00001100B      ;OFF CURSOR
         LCALL  LCDWI
         RET
SCANH1:  MOV    A,#00001110B      ;ON CURSOR
         LCALL  LCDWI
         RET
; ******* SCANDLn SUB ******
; WRITE ASCII FROM DISBUF TO LCD
SCANDL1: MOV    R7,#DISBUF
         MOV    A,#80H            ;SET ADDRESS LINE 1
         LCALL  LCDWI
         LCALL  SCAND5
         RET
SCANDL2: MOV    R7,#DISBUF
         MOV    A,#0C0H           ;SET ADDRESS LINE 2
         LCALL  LCDWI
         LCALL  SCAND5
         RET
SCAND5:  MOV    R2,#20            ;20 CHAR.
         MOV    A,R7
         MOV    R0,A
SCAND51: MOV    A,@R0
         LCALL  LCDWD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          INC     R0
          DJNZ    R2,SCAND51
          RET
; ***************************
SCNDL24: MOV      R7,#DISBUF
          MOV     A,VTREG+3          ;SET DDRAM ADDRESS
          LCALL   LCDWI
          LCALL   SCNDL25
          RET
SCNDL25: MOV      A,R7
          MOV     R0,A
SCNDL26: MOV      A,@R0
          LCALL   LCDWD
          INC     R0
          DJNZ    VTREG+2,SCNDL26
          RET
; ***************************
SCNDL22: MOV      R7,#DISBUF+2
          MOV     A,VTREG+3          ;SET DDRAM ADDRESS
          LCALL   LCDWI
          LCALL   SCNDL21
          RET

SCNDL21: MOV      A,R7
          MOV     R0,A
SCNDL23: MOV      A,@R0
          LCALL   LCDWD
          INC     R0
          DJNZ    VTREG+2,SCNDL23
          RET
; ************ INH SUB ************
; INPUT 2 DIGIT HEX
INH:      MOV     HEXBUF+1,A
          MOV     INPBKH,A           ;BACKUP
          LCALL   HTOA
          SETB    INPFIR
          SETB    SCANFF
INH1:     MOV     VTREG+2,#2
INH10:    LCALL   SCNDL22
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
INH11:  LCALL   SCAN            ;WAIT KEY
        MOV     R2,A
        CLR     C
        SUBB    A,#10H
        MOV     A,R2
        JNC     INH5            ;>F
        MOV     R2,A
        LCALL   HTOAX           ;CHANGE TO ASCII CODE
        MOV     R2,A
        JB      INPFIR,INH3     ;FIRST DIGIT PROCESS
        MOV     R0,#DISBUF+1    ;NEXT DIGIT PROCESS
        MOV     R1,#0
INH2:   INC     R0
        INC     R1
        MOV     A,@R0
        CJNE    A,#20H,INH2
        MOV     A,R2            ;SPACE DIGIT
        MOV     @R0,A
        CJNE    R1,#2,INH21
        SETB    INPFIR
        LJMP    INH1
INH21:  MOV     A,R1
        MOV     VTREG+2,A
        LJMP    INH10
        MOV     R0,#DISBUF+1    ;NEXT DIGIT PROCESS
        MOV     R1,#0
INH3:   MOV     A,#20H          ;FIRST DIGIT PROCESS
        MOV     DISBUF+2,R2
        MOV     DISBUF+3,A
        MOV     VTREG+2,#2
        LCALL   SCNDL22
        MOV     A,#00010000B    ;SHIF DISPLAY
        LCALL   LCDWI
        CLR     INPFIR
        LJMP    INH11
INH5:   CJNE    R2,#11H,INH6    ;CHECK DEC KEY
        MOV     R0,#DISBUF+3    ;IS DEC KEY
        MOV     R1,#2
INH51:  MOV     A,@R0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CJNE    A,#20H,INH55
        DEC     R0
        DJNZ    R1,INH51
        MOV     VTREG+2,#1
        LCALL   SCNDL22
        MOV     A,#00010000B        ;SHIF DISPLAY
        LCALL   LCDWI
        SJMP    INH11
INH55:  MOV     @R0,#20H
        MOV     A,R1
        MOV     VTREG+2,A
        LCALL   SCNDL22
        MOV     A,#00010000B        ;SHIF DISPLAY
        LCALL   LCDWI
        CLR     INPFIR
        LJMP    INH11
INH6:   CJNE    R2,#12H,INH8        ;CHECK INC KEY
        MOV     A,INPBKH            ;IS INC KEY
        LJMP    INH
INH8:   CLR     INPBFG
        CJNE    R2,#13H,INH81
        SETB    INPBFG
INH81:  LCALL   ATOH                ;EXIT
        MOV     A,HEXBUF+1
        CLR     SCANFF
        RET
; ********** GET2 SUB **********
; GET HEX ON DIGIT 6,7
GET2:   CLR     GETOKF
        PUSH    ACC
        CLR     C
        SUBB    A,#10H              ;COMPARE
        POP     ACC
        JC      GET21               ;<=F
        RET
GET21:  MOV     R2,A
        LCALL   HTOAX
        MOV     R2,A
        MOV     A,DISBUF+7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CJNE    A,#20H,GET25
        MOV     A,R2
        MOV     DISBUF+7,A          ;LAST DIGIT
        SETB    GETOKF
        RET
GET25:  MOV     DISBUF+6,R2         ;FIRST DIGIT
        MOV     DISBUF+7,#20H
        RET
; ********** INPSE SUB **********
; INPUT START,END
INPSE:  MOV     DPTR,#DATA25
        LCALL   L12WDL2
        MOV     DPH,STTADD
        MOV     DPL,STTADD+1
        MOV     VTREG+3,#0CCH       ;SET DDRAM ADDRESS
        LCALL   INW
        MOV     STTADD,DPH
        MOV     STTADD+1,DPL
        JB      INPBFG,INPSE
INPSE1: MOV     DPTR,#DATA26
        LCALL   L12WDL2
        MOV     DPH,ENDADD
        MOV     DPL,ENDADD+1
        MOV     VTREG+3,#0CCH       ;SET DDRAM ADDRESS
        LCALL   INW
        MOV     ENDADD,DPH
        MOV     ENDADD+1,DPL
        JB      INPBFG,INPSE
        RET
; ********** INPSEH SUB *********
; INPUT START END HEX
INPSEH: MOV     DPH,STTADD
        MOV     DPL,STTADD+1
        MOV     R2,#6DH             ;S
        LCALL   INW
        MOV     STTADD,DPH
        MOV     STTADD+1,DPL
        JB      INPBFG,INPSEH
INPSEH1: MOV    DPH,ENDADD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          MOV     DPL,ENDADD+1
          MOV     R2,#79H             ;E
          LCALL   INW
          MOV     ENDADD,DPH
          MOV     ENDADD+1,DPL
          JB      INPBFG,INPSEH
          MOV     A,HEXDAT
          MOV     R2,#76H             ;H
          LCALL   INH
          MOV     HEXDAT,A
          JB      INPBFG,INPSEH1
          RET
; ********** SBYTE SUB **********
; SEND BYTE
SBYTE:    JNB     TI,$                ;WAIT FOR SEND OK
          CLR     TI
          MOV     SBUF,A
          RET
; ********** SBYTEH SUB **********
; SEND 2 BYTE HEX FROM A
SBYTEH:   LCALL   HTOAC
          MOV     A,R2
          LCALL   SBYTE
          MOV     A,R3
          LCALL   SBYTE
          RET
SBYTEHC:  XCH     A,R7                ;R7=CHECKSUM
          ADD     A,R7
          XCH     A,R7
          LCALL   SBYTEH
          RET
; ********** SBLOCK SUB *********
; SEND BLOCK
SBLOCK:   CLR     A
          MOVC    A,@A+DPTR
          CJNE    A,#0,SBLOCK1
          INC     DPTR                ;NEXT ADDRESS
          RET                         ;EXIT BY 0
SBLOCK1:  CJNE    A,#0DH,SBLOCK2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          LCALL   SLF                ;CR/LF
          INC     DPTR               ;NEXT ADDRESS
          RET                        ;EXIT BY ODH
SBLOCK2:  LCALL   SBYTE
          INC     DPTR
          SJMP    SBLOCK
; ********** SLF SUB **********
; PRINT CR/LF
SLF:      MOV     A,#0DH
          LCALL   SBYTE
          MOV     A,#0AH
          LCALL   SBYTE
          RET
; ********** RBYTE SUB **********
RBYTE:    JNB     RI,$               ;WAIT FOR RECEIVE OK
          CLR     RI
          MOV     A,SBUF
          RET
; ********** RBYTEH SUB *********
; READ HEX-ASCII FROM SERIAL (2 BYTE)

RBYTEH:   LCALL   RBYTE
          MOV     R2,A
          LCALL   RBYTE
          MOV     R3,A
          LCALL   ACTOH
          RET
RBYTEHC:  LCALL   RBYTEH             ;R7=CHECKSUM
          XCH     A,R7
          ADD     A,R7
          XCH     A,R7
          RET
; ********** HTOAX SUB **********
; ONE BYTE HEX TO ASCII CODE
HTOAX:    MOV     A,R2               ;MAKE SURE 0-F
          ANL     A,#0FH
          MOV     R2,A
          MOV     DPTR,#ASCTAB       ;TABLE
          MOV     A,DPL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ADD     A,R2
        MOV     DPL,A
        MOV     A,DPH
        ADDC    A,#0
        MOV     DPH,A
        CLR     A
        MOVC    A,@A+DPTR
        RET
ASCTAB: DB      30H,31H,32H,33H         ;0123
        DB      34H,35H,36H,37H         ;4567
        DB      38H,39H,41H,42H         ;89AB
        DB      43H,44H,45H,46H         ;CDEF
; ********** HTOA SUB ***********
; HEX TO ASCII CODE
HTOA:   MOV     R0,#HEXBUF
        MOV     R1,#DISBUF
        LCALL   HTOAS
        LCALL   HTOAS
        INC     R1
        INC     R1
        LCALL   HTOAS
        RET
HTOAS:  MOV     A,@R0                   ;DIGIT L
        PUSH    ACC
        SWAP    A
        ANL     A,#0FH
        MOV     R2,A
        LCALL   HTOAX
        MOV     @R1,A
        INC     R1
        POP     ACC                     ;DIGIT R
        ANL     A,#0FH
        MOV     R2,A
        LCALL   HTOAX
        MOV     @R1,A
        INC     R1
        INC     R0
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ********** HTOA8 SUB **********
; HEX TO ASCII CODE
HTOA8:  MOV     R0,#HEXBUF
        MOV     R1,#DISBUF
        LCALL   HTOAS
        LCALL   HTOAS
        INC     R1
        INC     R1
        LCALL   HTOAS8
        INC     R1
        LCALL   HTOAS8
        INC     R1
        LCALL   HTOAS8
        INC     R1
        LCALL   HTOAS8
        INC     R1
        LCALL   HTOAS8
        RET
HTOAS8: MOV     A,@R0                   ;DIGIT L
        PUSH    ACC
        SWAP    A
        ANL     A,#0FH
        MOV     R2,A
        LCALL   HTOAX
        MOV     @R1,A
        INC     R1
        POP     ACC                     ;DIGIT R
        ANL     A,#0FH
        MOV     R2,A
        LCALL   HTOAX
        MOV     @R1,A
        INC     R1
        INC     R0
        RET
; ********** ATOHX SUB **********
; ONE BYTE ASCII CODE TO HEX
ATOHX:  MOV     DPTR,#ASCTAB            ;TABLE
        MOV     R3,A                    ;MEMO TO R3
        MOV     R2,#0                   ;COUNTER
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ATOHX1:  CLR     A
         MOVC    A,@A+DPTR
         CLR     C
         SUBB    A,R3                ;COMPARE
         JZ      ATOHX2
         INC     DPTR
         INC     R2
         CJNE    R2,#16,ATOHX1
         MOV     R2,#0
ATOHX2:  MOV     A,R2
         RET
; ********** ATOH SUB ***********
; ASCII CODE TO HEX
ATOH:    MOV     R0,#DISBUF
         MOV     R1,#HEXBUF
         LCALL   ATOHS
         LCALL   ATOHS
         INC     R0
         INC     R0
         LCALL   ATOHS
         RET
ATOHS:   MOV     A,@R0               ;DIGIT L
         LCALL   ATOHX
         ANL     A,#0FH
         SWAP    A
         MOV     @R1,A
         INC     R0
         MOV     A,@R0               ;DIGIT R
         LCALL   ATOHX
         ANL     A,#0FH
         ORL     A,@R1
         MOV     @R1,A
         INC     R0
         INC     R1
         RET
; ********** HTOAC SUB **********
; CONVERT HEX TO ASCII
HTOAC:   PUSH    ACC
         SWAP    A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          LCALL   HTOACS
          MOV     R2,A
          POP     ACC
          LCALL   HTOACS
          MOV     R3,A
          RET
HTOACS:   ANL     A,#0FH
          CJNE    A,#0AH,$+3
          JNC     HTOACS1
          ORL     A,#30H
          RET
HTOACS1:  SUBB    A,#9
          ORL     A,#40H
          RET
; ********** ACTOH SUB **********
; ASCII TO HEX CONVERT
ACTOH:    MOV     A,R2
          LCALL   ACTOHS
          SWAP    A
          MOV     R2,A
          MOV     A,R3
          LCALL   ACTOHS
          ORL     A,R2
          RET
ACTOHS:   CJNE    A,#'A',$+3
          JC      ACTOHS1
          ADD     A,#9
ACTOHS1:  ANL     A,#0FH
          RET
; ************ DATA *************
DATA1:    DB      "**MICROCONTROLLER** "
DATA2:    DB      "BASED COMMUNICATION "
DATA3:    DB      "EDIT CHANNAL        "
DATA4:    DB      "EDIT RAM            "
DATA5:    DB      "LIST CHANNAL        "
DATA6:    DB      "LIST RAM            "
DATA7:    DB      "RS-485 NETWORK      "
DATA9:    DB      "FUNCTION            "
DATA10:   DB      "FUNC 0 BAUD RATE    "
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DATA11:  DB     "FUNC 1 CHECK SUM    "
DATA12:  DB     "FUNC 2 MOVE DATA    "
DATA13:  DB     "FUNC 3 FIND DATA    "
DATA14:  DB     "FUNC 4 RX BLOCK     "
DATA15:  DB     "FUNC 5 TX BLOCK     "
DATA16:  DB     "FUNC 6 RECEIVE HEX  "
DATA17:  DB     "FUNC 7 SEND HEX     "
DATA18:  DB     "FUNC 8 CLEAR DATA:M1"
DATA19:  DB     "FUNC 9 CLEAR DATA:M2"
DATA20:  DB     "FUNC10 RAM TEST     "
DATA21:  DB     "FUNC11 LCD TEST     "
DATA22:  DB     "   Address :"
DATA23:  DB     " Baud Rate :"
DATA24:  DB     " Check Sum ="
DATA25:  DB     "Start Addr :"
DATA26:  DB     "  End Addr :"
DATA27:  DB     "   To Addr :"
DATA28:  DB     "    Length :"
DATA29:  DB     ">>Press ENTER       "
DATA30:  DB     "Sure? >> Press ENTER"
DATA31:  DB     "Please  Wait        "
DATA32:  DB     "Process Complete    "
DATA33:  DB     "Select Mode ?       "
DATA34:  DB     "Wait Receive Data   "
DATA35:  DB     "Data Error          "
DATA36:  DB     "                    "

ENDROM:  END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลุล่วงได้ด้วยความช่วยเหลือจากหลายท่าน ผู้จัดทำขอขอบคุณ รศ.พิพัฒน์ เลาหสงคราม อาจารย์ที่ปรึกษาที่ให้คำแนะนำ ปรึกษาและให้คำปรึกษา และให้ความรู้ในการจัดทำมาโดยตลอด ขอขอบคุณพ่อและแม่ของผู้จัดทำที่คอยให้กำลังใจและทุนทรัพย์ในการจัดทำ สุดท้ายนี้ขอขอบคุณอาจารย์ทุกท่านและเพื่อนทุกคนที่มีส่วนทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี จึงขอขอบคุณมา ณ โอกาสนี้ด้วย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. รศ. พิพัฒน์ เลาหสงคราม , ไมโครคอนโทรลเลอร์ MCS-51 ภาควิชาเทคโนโลยีการวัดคุม ฯ
2. รศ. พิพัฒน์ เลาหสงคราม , การทดลองไมโครคอนโทรลเลอร์ MCS-51 ภาควิชาเทคโนโลยีการวัดคุม ฯ
3. ยืน ภู่วรวรรณ , ทฤษฎีและการประยุกต์ไมโครคอนโปรเซสเซอร์ Z-80
4. ปรเมษฐ์ ประนยานันทน์ , ปิยพงศ์ เผ่าวนิช , คู่มือและการประยุกต์ใช้งานไมโครคอนโทรลเลอร์ MCS-51
5. สุนทร วิทูสุรพจน์ , การใช้งานไมโครคอนโทรลเลอร์ ตระกูล 8051
6. ผศ. สุพรรณ กุลพาณิชย์ , พยัต ชินวิไล , วินัย เสือสีนวล , อดุลย์ เพ็ชรศิริพันธ์, อนุชิต สิทธิสินธุ์ , โครงงานการสื่อสารข้อมูล RS-485 , ปริญญานิพนธ์ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้