หน่วยที่ 8 พอร์ตอนุกรม

อดิศักดิ์ ชิณะวงศ์

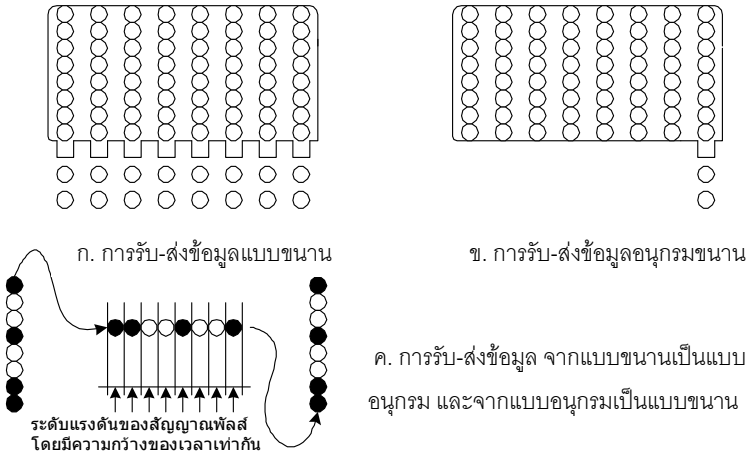
เอกสารประกอบการเรียนวิชาไมโครคอนโทรลเลอร์

เผยแพร่ที่ www.Adisak51.com

1. การรับส่งข้อมูลแบบอนุกรม

การติดต่อข้อมูลในไอซี MCS-51 โดยตรงจากพอร์ต ที่เป็นข้อมูลขนาด 1 ไบต์ หรือ 8 บิต โดยปกติการรับส่งข้อมูลไปพร้อมๆ กันทั้งหมด เรียกว่าการรับส่งข้อมูลแบบขนาน แสดงดังภาพที่ 8.1 ก. เป็นการรับส่งข้อมูลขนาด 8 บิตไปพร้อมๆ กัน วิธีการแบบนี้รับส่งข้อมูลได้รวดเร็ว แต่ต้องมีจำนวนสายสัญญาณพอดีกับจำนวนของบิตที่ต้องการรับส่ง

การสื่อสารข้อมูลแบบอนุกรมเป็นอีกวิธีหนึ่ง ที่สามารถลดจำนวนของสายสัญญาณโดยใช้เพียงสายส่ง (TxD) 1 เส้น สายรับ (RxD) 1 เส้น และสายกราวด์ (Ground) 1 เส้น แสดงดังภาพที่ 8.1ข หากต้องการส่งข้อมูลขนาด 8 บิต ต้องส่งข้อมูลออกไปทีละบิตอย่างเป็นลำดับ จนกว่าครบทั้ง 8 บิตแสดงดังภาพที่ 8.1 ค.แสดงการส่งข้อมูลแบบขนานให้เป็นแบบอนุกรม ข้อมูลถูกส่งไปตามสายสัญญาณทีละบิตตามจังหวะเวลาที่กำหนด โดยจังหวะเวลาต้องมีมาตรฐานของฝ่ายส่ง และฝ่ายรับ การรับสัญญาณที่ส่งทีละบิตโดยตรวจสอบระดับแรงดันของสัญญาณที่เข้ามา และแปลงเป็นลอจิก “1” หรือ “0” เมื่อรับข้อมูลจนครบตามมาตรฐานที่กำหนดไว้แล้ว ถูกเปลี่ยนให้อยู่ในรูปแบบของข้อมูลแบบขนานเหมือนเดิม



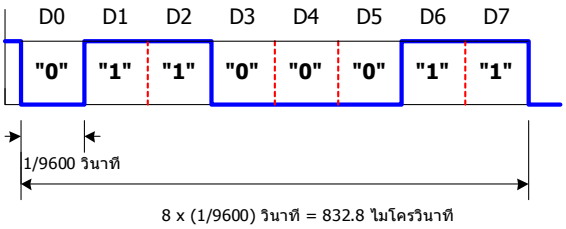
ภาพที่ 8.1 แสดงการรับส่งข้อมูลแบบขนาน และแบบอนุกรม

1.1 จังหวะเวลาของการสื่อสารข้อมูลอนุกรม

การสื่อสารข้อมูลแบบอนุกรมเพื่อรับ หรือส่งข้อมูล เป็นลักษณะของกลุ่มข้อมูล ดังนั้นอัตราความเร็วระหว่างการรับ และส่งต้องมีค่าเท่ากัน โดยทั่วไปการระบุความเร็วของจำนวนบิตในการรับ และส่งข้อมูลใน 1 วินาที โดยเรียกความเร็วในการส่งข้อมูลว่า อัตราบอด (Baud Rate) ซึ่งมีหน่วย เป็นบิตต่อวินาที เช่น 300, 1,200, 2,400, 4,800 และ 9,600 บิตต่อวินาที

ภาพที่ 8.2 แสดงการส่งข้อมูลด้วยความเร็ว 9600 บิตต่อวินาที โดยใช้เวลาในการรับส่งข้อมูลหนึ่งบิตมีค่าเท่ากับ 1/9600 หรือ 104.1 ไมโครวินาที และเวลาในการรับส่งข้อมูลทั้ง 8 บิต มีค่าเท่ากับ 8 x 104.1 หรือ 832.8 ไมโครวินาที

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	0	0	1	1	0



ภาพที่ 8.2 แสดงการส่งข้อมูลแบบอนุกรมด้วยความเร็ว 9600 บิตต่อวินาที

1.2 รูปแบบการสื่อสารข้อมูลอนุกรม

การสื่อสารแบบอนุกรม สามารถแบ่งประเภทของการสื่อสารตามลักษณะสัญญาณ ในการสื่อสารได้ 2 แบบ แบบแรกเป็นการสื่อสารแบบซิงโครนัส (Synchronous) ใช้สัญญาณนาฬิกาควบคุมการรับ และส่งสัญญาณ เช่น สายคีย์บอร์ดคอมพิวเตอร์ แบบที่สองเป็นการสื่อสารแบบอะซิงโครนัส (Asynchronous) เป็นการรับและข้อมูลโดยอาศัยความพร้อมของทางด้านรับ และด้านส่ง มีบิตตรวจสอบการเริ่มต้น การสิ้นสุด และบิตพาริตี่ส่งรวมไปด้วย ซึ่งส่วนที่ใช้ในการควบคุมการรับส่งข้อมูลที่เรียกว่า UART (Universal Asynchronous Receiver Transmitters) การสื่อสารข้อมูลอนุกรมแบบอะซิงโคนัส (Asynchronous) เป็นวิธีการรับ และส่งข้อมูลโดยไม่ต้องอาศัยสัญญาณนาฬิการ่วมไปด้วย แต่ใช้อัตราความเร็วในการส่งจำนวนข้อมูลต่อวินาที โดยเพิ่มบิตข้อมูลบางอย่างร่วมไปกับการส่งข้อมูลจริง เพื่อตรวจสอบข้อมูลได้ถูกต้องมากยิ่งขึ้นแสดงดังภาพที่ 8.3 ประกอบด้วยบิตข้อมูล 4 ส่วนคือ

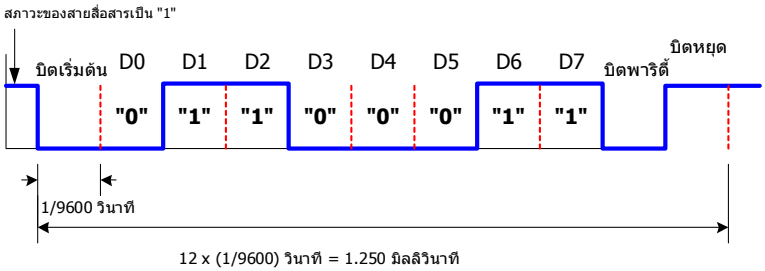
บิตเริ่มต้น (Start Bit) มีขนาด 1 บิต เป็นระดับลอจิกตรงกันข้ามกับระดับลอจิกของสภาวะสายสื่อสาร ขณะที่ยังไม่มีการส่งข้อมูล

บิตข้อมูล (Data Bit) เริ่มจากบิตที่มีนัยสำคัญต่ำสุดก่อนหรือ บิต LSB โดยข้อมูลที่ส่งอาจมีขนาด 5, 6, 7 หรือ 8 บิตได้

บิตแสดงสภาวะเลขคู่หรือเลขคี่ (Parity Bit) มีขนาด 1 บิตโดยนำไปต่อท้ายกับบิตข้อมูล ค่าของบิตนี้ขึ้นอยู่กับจำนวนข้อมูลที่เป็น “1” โดยเลือกการส่งข้อมูลเป็นแบบ พาริตี่คู่ หรือ พาริตี่คี่ ตัวอย่างเช่น ถ้ากำหนดให้มีการส่งข้อมูลแบบพาริตี่คู่ แต่ข้อมูลทั้งหมดมีเลข 1 เป็นจำนวนคี่ ให้บิตพาริตี่นี้เป็น “1” เพื่อได้จำนวนเลข “1” ที่ส่งทั้งหมดเป็นจำนวนคู่นั้นเอง และทำนองเดียวกัน ทางด้านรับต้องมีการตรวจสอบจำนวนข้อมูลที่รับเข้ามาเป็น “1” รวมทั้งบิตพาริตี่ 1 บิต ถ้ามีค่า “1” เป็นจำนวนคู่ แสดงว่าข้อมูลที่รับเข้ามาถูกต้อง สามารถกำหนดการรับและส่งเป็นแบบ NONE โดยไม่ต้องมีการตรวจสอบพาริตี่บิตได้

บิตสุดท้ายหรือบิตหยุด (Stop Bit) เป็นการระบุถึงขอบเขตการสิ้นสุดของข้อมูล โดยทำให้ขาข้อมูลมีสถานะลอจิกเป็น “1” ซึ่งอาจมีจำนวนมากกว่าหนึ่งบิตได้ เช่น 1 บิต 1.5 บิต หรือ 2 บิต

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	0	0	1	1	0



ภาพที่ 8.3 แสดงการรับส่งข้อมูลอนุกรมขนาด 8 บิต แบบอะซิงโคนัส (Asynchronous)

2. รีจิสเตอร์ที่ใช้ควบคุมการรับส่งข้อมูลของพอร์ตอนุกรม

วงจรภายในของไอซี MCS-51 ส่วนติดต่อสื่อสารอนุกรมแบบ UART และมีโครงสร้างแบบฟลูดูเพล็ก (Full Duplex) สามารถรับ และส่งข้อมูลได้ในเวลาเดียวกันได้ พอร์ตอนุกรมของไอซี MCS-51 ใช้ขาที่ชื่อ TxD ตำแหน่งพอร์ต P3.1 และขาชื่อ RxD ตำแหน่งพอร์ต P3.2 โดยรีจิสเตอร์ใช้งานติดต่อสื่อสารทางพอร์ตอนุกรม ประกอบด้วย รีจิสเตอร์ SCON ทำหน้าที่ควบคุมการเลือกโหมดการทำงาน รีจิสเตอร์ SBUF ใช้เก็บข้อมูลการรับหรือการส่ง และรีจิสเตอร์ PCON ทำหน้าที่ กำหนดอัตรารับส่ง โดยรีจิสเตอร์ในแต่ละตัวมีรายละเอียดการทำงานดังต่อไปนี้

2.1 รีจิสเตอร์ SCON (Serial Port Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ในส่วนของรีจิสเตอร์พิเศษตำแหน่งแอดเดรสที่ 98H และสามารถเข้าถึงข้อมูลแบบไบต์ และบิตได้ แสดงดังภาพที่ 8.4 ทำหน้าที่ควบคุมการเลือกโหมดการทำงาน และเก็บข้อมูลในบิตที่ 9 (โดยปกติข้อมูลมีขนาด 8 บิต อยู่ในรีจิสเตอร์ SBUF) ของการรับข้อมูล (RB8) และส่งข้อมูล (TB8) รายละเอียดของแต่ละบิตมีดังต่อไปนี้

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
SM0	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**

ภาพที่ 8.4 รีจิสเตอร์ควบคุมการเลือกโหมดการทำงานของพอร์ตอนุกรม (SCON)

SM0, SM1 (Serial Port Mode Bit 0-1) เป็นบิตที่ใช้ในการกำหนดโหมดการทำงานของพอร์ตอนุกรมจำนวน 4 โหมด ดังตารางที่ 8.1

ตารางที่ 8.1 แสดงการกำหนดบิต SM0 และ SM1 เพื่อเลือกโหมดการทำงาน

SM0	SM1	โหมด	การทำงาน	อัตรารับส่ง
0	0	0	Shift Register	f osc /12
0	1	1	8 bit UART	Variable
1	0	2	9 bit UART	fosc/32 หรือ fosc/64
1	1	3	9 bit UART	Variable

SM2 (Serial Port Mode 2) ทำหน้าที่ควบคุมการทำงาน และเลือกลักษณะการเชื่อมต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์แบบ Single Processor System หรือ Multi Processors System โดยกำหนดให้ SM2 = “1” แบบ Multi Processors System เป็นการสื่อสารแบบใช้ซีพียู หลายๆ ตัวร่วมกันทำงาน ใช้งานในโหมด 2 หรือโหมด 3 SM2 = “0” แบบ Single Processor System โดยสามารถใช้ได้กับทุกโหมด (การใช้งานโหมด 0 ต้องกำหนดให้ SM2 = “0”) กรณีเลือกให้ SM2 = “1” ถ้าข้อมูลรับเข้ามาบิตที่ 9 (อยู่ในบิต RB8) มีค่าเป็น “1” แฟลกอินเตอร์รัปต์ทางด้านรับถูกเซตให้เป็น “1” (RI = 1) ถ้าข้อมูลในบิตที่ 9 มีค่าเป็น “0” แฟลกอินเตอร์รัปต์ทางด้านรับถูกเคลียร์เป็น 0 (RI = 0) การทำงานในโหมด 1 ถ้าให้ SM2 = 1 แฟลกอินเตอร์รัปต์ทางด้านรับ (แฟลก RI) ไม่ถูกเซตหากข้อมูลที่รับเข้ามาไม่มีบิตหยุด (Stop Bit)

REN (Enable Serial Reception) เป็นบิตควบคุมการรับข้อมูลของพอร์ตอนุกรม กำหนดสถานะของบิตได้โดยซอฟต์แวร์ กำหนดให้บิต REN มีสถานะลอจิก “1” ให้มีการรับข้อมูล กำหนดให้บิต REN มีสถานะลอจิก “0” ไม่ให้มีการรับข้อมูล

TB8 (Transmit Bit D8) เป็นบิตข้อมูลของบิตที่ 9 ในการส่งข้อมูลใช้งานโหมด 2 และโหมด 3 กำหนดสถานะของบิตได้โดยซอฟต์แวร์

RB8 (Receive Bit D8) เป็นบิตข้อมูลของบิตที่ 9 ในการรับข้อมูลใช้งานโหมด 2 และโหมด 3 หากใช้งานในโหมด 1 ถ้ากำหนดให้ SM2 = 0 บิตนี้เป็นค่าของ Stop Bit ที่รับเข้ามา สำหรับโหมด 0 ไม่ใช้งานในบิตนี้

TI (Transmit Interrupt Flag) เป็นบิตอินเตอร์รัปต์ด้านส่งข้อมูล และถูกเซตทางฮาร์ดแวร์เมื่อมีการส่งข้อมูลเสร็จสิ้นลง ในบิตที่ 8 ของโหมด 0 (Shift Register) หรือเมื่อเริ่มต้นส่งบิตหยุดในโหมด 1 โหมด 2 หรือโหมด 3 และต้องเคลียร์บิตนี้ด้วยซอฟต์แวร์ทุกครั้ง หลังการส่งข้อมูลเรียบร้อยแล้ว

RI (Receive Interrupt Flag) เป็นบิตอินเตอร์รัปต์ทางด้านรับข้อมูล และถูกเซตทางฮาร์ดแวร์เมื่อมีการรับข้อมูลเสร็จสิ้นลงในบิตที่ 8 ของโหมด 0 และต้องเคลียร์บิตนี้ ด้วยซอฟต์แวร์ทุกครั้ง หลังการรับข้อมูลเสร็จเรียบร้อยแล้ว หรือ กล่าวได้ว่าถ้าบิต RI ถูกเซตเมื่อใด หมายถึงข้อมูลได้เก็บไว้ที่รีจิสเตอร์ SBUF จนครบทั้ง 8 บิตแล้ว และสามารถอ่านข้อมูลจากรีจิสเตอร์ SBUF ได้

2.2 รีจิสเตอร์ SBUF (Serial Data Buffer Register)

เป็นรีจิสเตอร์ขนาด 8 บิตหรือ 1 ไบต์มีแอดเดรสอยู่ตำแหน่งที่ 99H และเข้าถึงข้อมูลแบบไบต์ได้อย่างเดียว ทำหน้าที่รับและส่งข้อมูลไปยังพอร์ตอนุกรมของไอซี MCS-51 การอ่านค่าข้อมูลจากภายนอกโดยอ่านค่าจากรีจิสเตอร์ SBUF ซึ่งทำหน้าที่เป็นบัปเฟอร์ (Buffer) ในการเก็บข้อมูลที่รับเข้ามาจากภายนอก และในทำนองเดียวกันขณะที่ต้องการส่งข้อมูลออก นำค่าข้อมูลเก็บไว้ในรีจิสเตอร์ SBUF ก่อน หลังจากนั้นจึงส่งข้อมูลออกไป โดยใช้คำสั่งโอนย้ายข้อมูลแบบไบต์เช่น MOV SBUF,#20H หรือ MOV SBUF,@R1 การรับข้อมูลในโหมด 0 เริ่มต้นรับเมื่อค่าของบิต RI = “0” และ REN = “1” ส่วนโหมดอื่นๆ การรับข้อมูลเริ่มต้นเมื่อกำหนดบิต REN = “1” และมีบิตเริ่มต้น เข้ามาที่ขา RxD

2.3 รีจิสเตอร์ PCON (Power Control)

เป็นรีจิสเตอร์ขนาด 1 ไบต์มีแอดเดรสอยู่ตำแหน่งที่ 87H เข้าถึงข้อมูลได้แบบไบต์อย่างเดียว รีจิสเตอร์ PCON ไม่สามารถอ้างตำแหน่งแบบบิตได้ แต่ใช้คำสั่งการ OR เช่น ORL PCON, #80H เป็นการเซตบิตที่ 7 และใช้การ AND เช่น ANL PCON, #01111111B เป็นการเคลียร์บิตที่ 7 แสดงดังภาพที่ 8.5 ประกอบด้วยบิตต่างๆ ดังต่อไปนี้

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMOD	-	-	-	GF1	GF0	PD	IDL

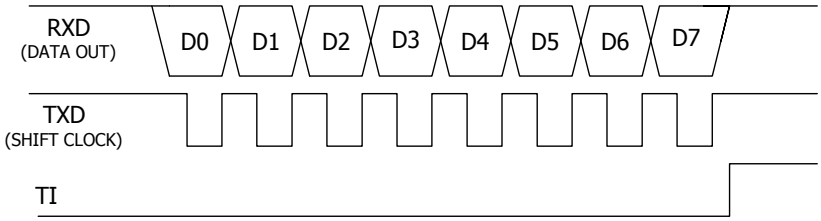
ภาพที่ 8.5 แสดงบิตต่างๆ ของรีจิสเตอร์ควบคุมพลังงาน (PCON)

PC0N.7 SMOD: ในกรณีใช้ไทม์เมอร์ 1 กำหนดอัตรารับส่ง (Baud Rate) ถ้าบิตนี้มีค่าเป็น“0” การใช้งานกับพอร์ตสื่อสารอนุกรมโหมด 1, โหมด 2 และโหมด 3 มีค่าอัตรารับส่ง เพิ่มขึ้นเป็นสองเท่า

3. โหมดการติดต่อทางพอร์ตอนุกรมของไอซี MCS-51

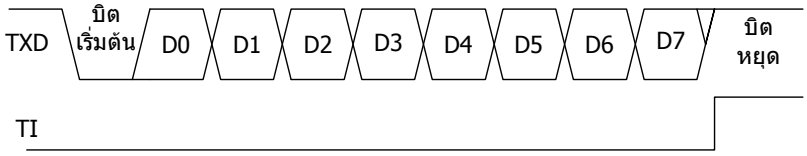
การสื่อสารอนุกรมของไอซี MCS-51 แบ่งออกได้เป็น 4 โหมด ในแต่ละโหมดสรุปหน้าที่ได้ดังนี้

โหมด 0 เป็นการรับส่งข้อมูลขนาด 8 บิตแบบอนุกรมแสดงดังภาพที่ 8.6 การส่งข้อมูล เลื่อนออกไปทีละบิตโดยใช้ RxD เพียงขาเดียว และไม่มีการส่งบิตเริ่มต้นส่วน TxD ใช้เป็นขาสัญญาณนาฬิกาให้จังหวะการเลื่อนข้อมูล กับวงจรภายนอก (Shift Clock) โดยอัตราการรับส่งข้อมูลเป็น 1/12 เท่าของสัญญาณนาฬิกา การรับส่งข้อมูลเริ่มจากบิตต่ำ (LSB) ก่อน ใช้สำหรับเป็นชิพท์รีจิสเตอร์ (Shift Register) มีการทำงานเป็นไดอะแกรมแสดงดังภาพที่ 8.9 จุดประสงค์เพื่อขยายพอร์ตอินพุต หรือพอร์ตเอาต์พุตให้มีจำนวนมากขึ้นแต่ในโหมด 0 มัก ไม่นิยมนำมาใช้งาน เพราะไอซี MCS-51 ในปัจจุบัน มีจำนวนพอร์ตที่มากพอและมีไอซีเบอร์อื่นๆ ในตระกูลเดียวกันที่มีจำนวนพอร์ตให้เลือกใช้งาน



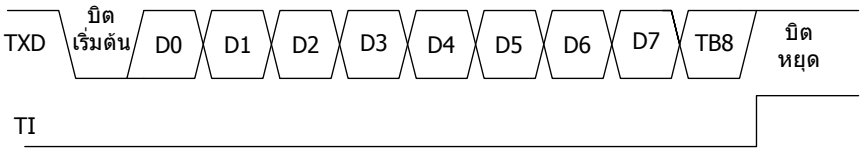
ภาพที่ 8.6 แสดงสัญญาณการรับและส่งข้อมูลในโหมด 0

โหมด 1 เป็นการรับและส่งข้อมูลขนาด 10 บิตแบบ UART สามารถติดต่อสื่อสารอนุกรมกับมาตรฐาน RS-232C ของไมโครคอมพิวเตอร์ได้ แสดงดังภาพที่ 8.7 ข้อมูลอนุกรมขนาด 10 บิต เข้ามาทางขา RXD และส่งข้อมูลออกแบบอนุกรมทางขา TXD ประกอบด้วย 1 บิตแรกเป็นบิตเริ่มต้น (Start Bit ค่า 0) 8 บิตต่อมาเป็นบิตของข้อมูล การรับและส่ง เริ่มจากบิตต่ำก่อน และบิตหยุด 1 บิต (Stop Bit ค่า 1) ส่วนทางด้านรับข้อมูลนำค่าบิตหยุดที่รับเข้ามาได้ เก็บไว้ในบิต RB8 ในรีจิสเตอร์ SCON และความเร็ว ของการส่งข้อมูลในโหมด 1 ขึ้นอยู่กับบิต SMOD ในรีจิสเตอร์ PCON และอัตราค่านับเกินของไทม์เมอร์ 1 ซึ่งอัตราการรับส่งข้อมูลในโหมดนี้สามารถกำหนดได้ตามต้องการ การทำงานเป็นไดอะแกรมแสดงดังภาพที่ 8.10



ภาพที่ 8.7 แสดงสัญญาณการส่งข้อมูลในโหมด 1

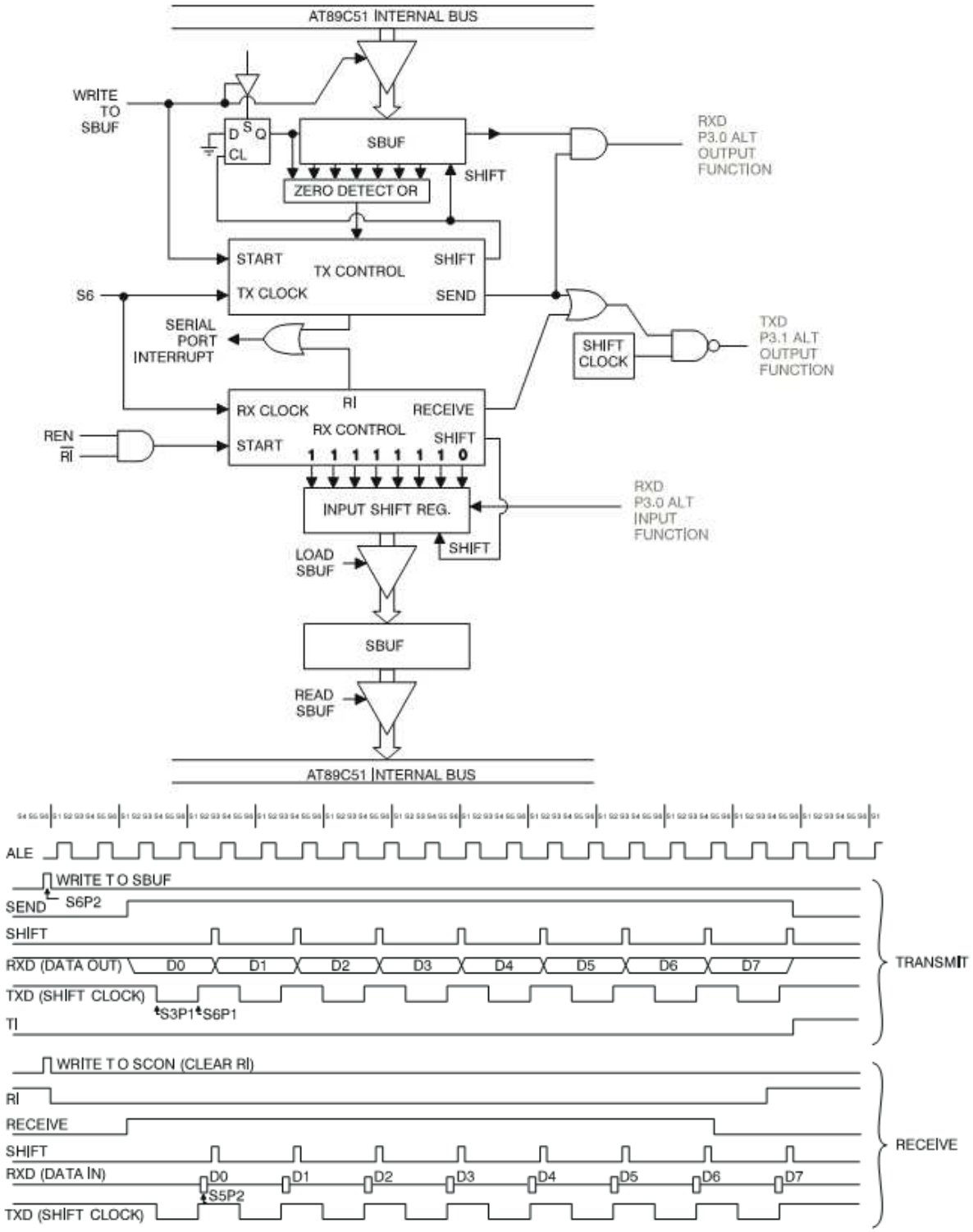
โหมด 2 เป็นการรับและส่งข้อมูลขนาด 11 บิตแบบ UART ข้อมูลแบบอนุกรมถูกรับเข้ามาทางขา RXD และส่งข้อมูลออกไปทางขา TXD แสดงดังภาพที่ 8.8



ภาพที่ 8.8 แสดงสัญญาณการส่งข้อมูลในโหมด 2

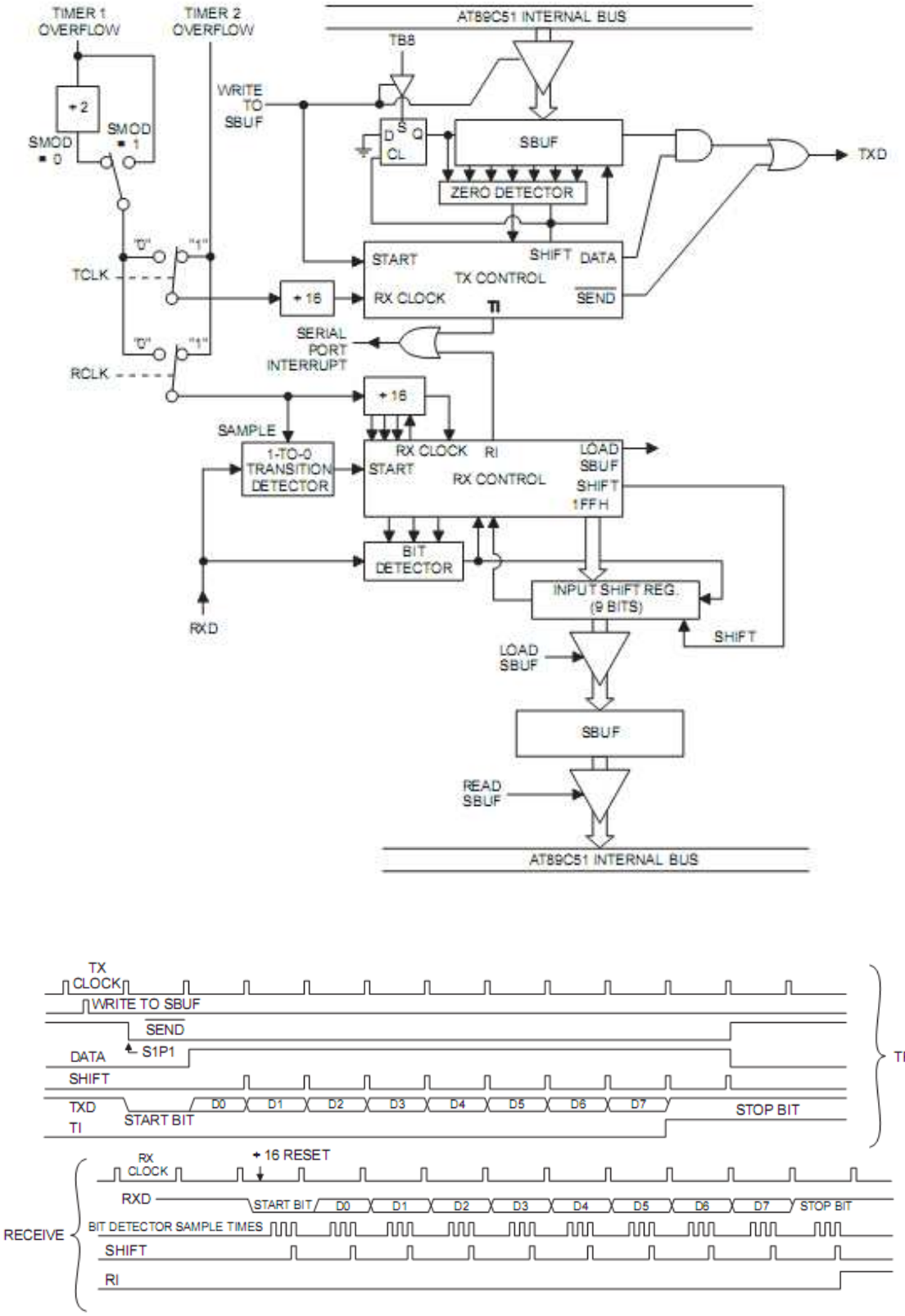
ข้อมูลขนาด 11 บิต ประกอบด้วย บิตแรกเป็นบิตเริ่มต้น (Start Bit ค่า 0) 9 บิตต่อมาเป็นบิตของข้อมูล และบิตสุดท้ายเป็นบิตหยุด 1 บิต (Stop Bit ค่า 1) สำหรับข้อมูลในบิตที่ 9 กำหนดไว้ใน TB8 อยู่ใน

รีจิสเตอร์ SCON สามารถกำหนดเป็น 1 หรือ 0 ได้ นิยมนำมาใช้ในการส่งบิตเพื่อตรวจสอบการส่งข้อมูล (Parity Bit) การทำงานเป็นไดอะแกรมแสดงดังภาพที่ 8.11



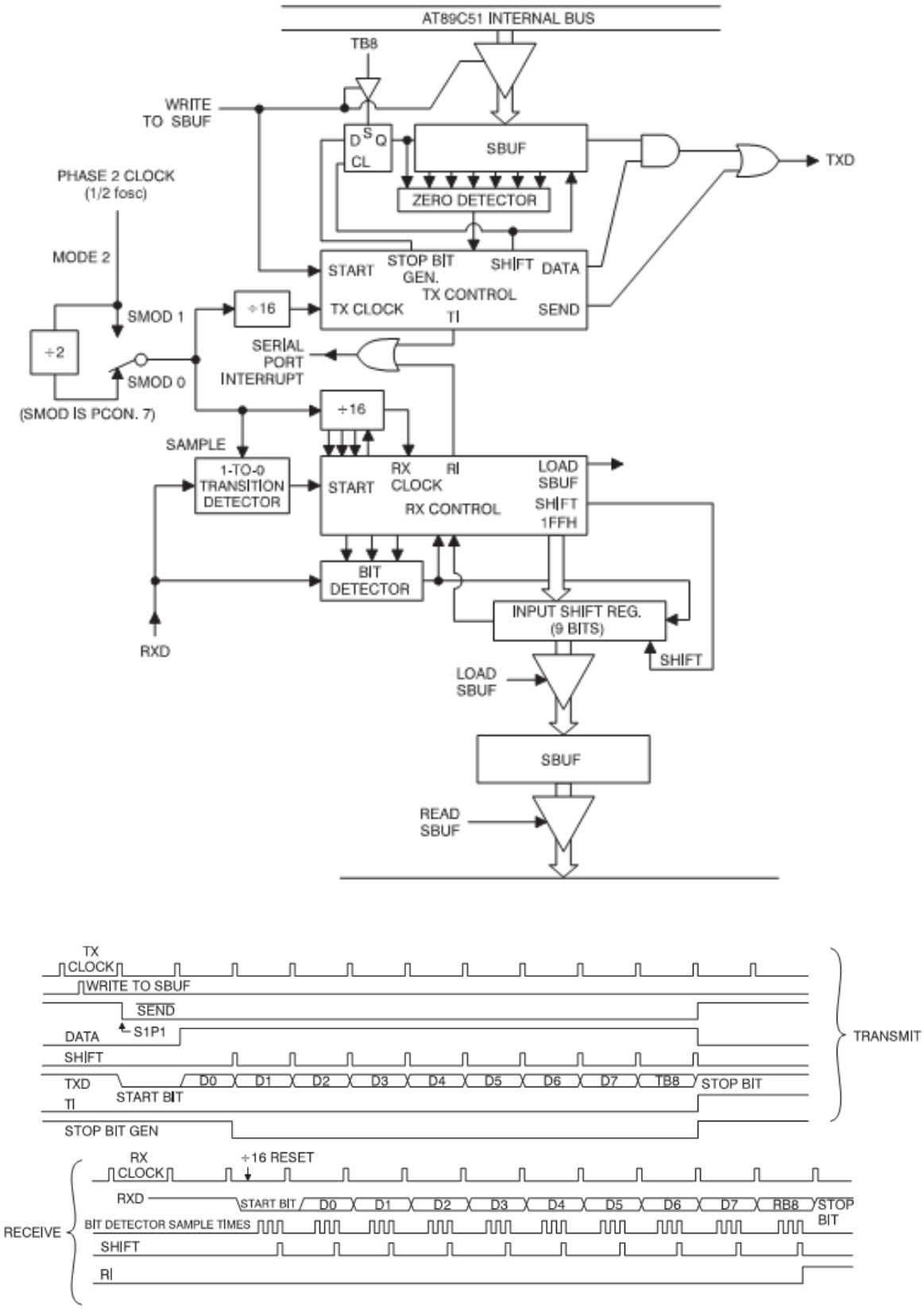
ภาพที่ 8.9 แสดงผังการทำงานของการสื่อสารอนุกรมในโหมด 0

(แหล่งอ้างอิง http://atmel.com/dyn/products/datasheets.asp?family_id=604)



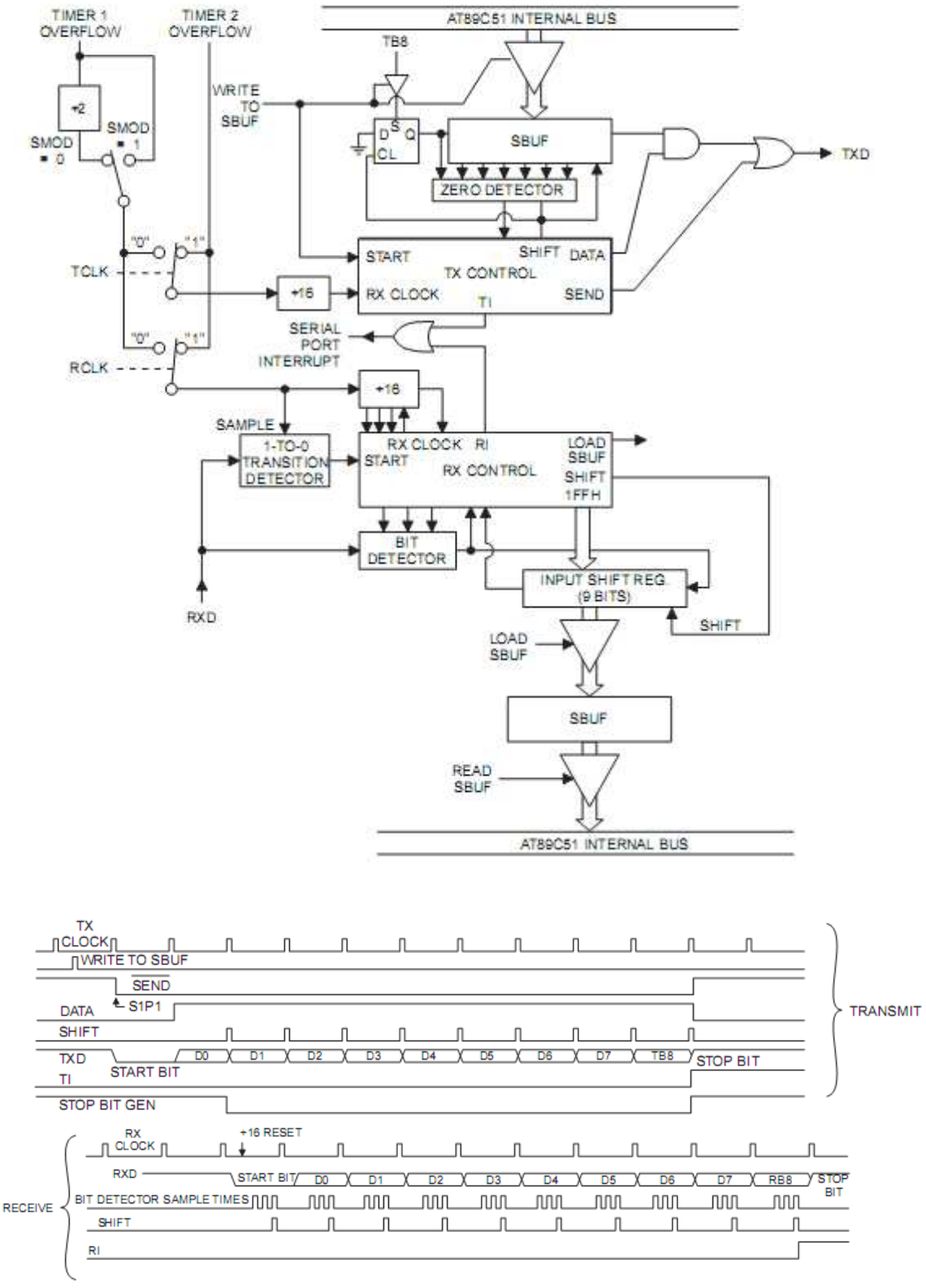
ภาพที่ 8.10 แสดงผังการทำงานของการสื่อสารอนุกรมในโหมด 1

(แหล่งอ้างอิง http://atmel.com/dyn/products/datasheets.asp?family_id=604)



ภาพที่ 8.11 แสดงผังการทำงานของการสื่อสารอนุกรมในโหมด 2

(แหล่งอ้างอิง http://atmel.com/dyn/products/datasheets.asp?family_id=604)



ภาพที่ 8.12 แสดงผังการทำงานของการสื่อสารอนุกรมในโหมด 3

(แหล่งอ้างอิง http://atmel.com/dyn/products/datasheets.asp?family_id=604)

โหมด 3 เป็นการรับส่งข้อมูล 11 บิตแบบ UART เหมือนกับโหมด 2 แต่ในโหมด 3 สามารถกำหนดอัตราความเร็วในการรับส่งข้อมูลได้ตามต้องการ การทำงานเป็นไดอะแกรมแสดงดังภาพที่ 8.12

การใช้งานในโหมด 2 และโหมด 3 นิยมนำมาใช้กับระบบการสื่อสารเพื่อเชื่อมต่อไอซี MCS-51 แบบ Multiprocessors System คือระบบการสื่อสารแบบใช้ไมโครคอนโทรลเลอร์ 1 ตัวเป็นตัวแม่ (Master) และสามารถเชื่อมต่อกับไมโครคอนโทรลเลอร์ที่เป็นตัวลูก (Slave) ได้เป็นจำนวนหลายๆ ตัว

4. การกำหนดอัตราการรับส่งข้อมูล

โหมด 0 อัตรารับส่งในโหมดนี้ไม่สามารถกำหนดอัตรารับส่งเองได้ แต่มีค่าเท่ากับความถี่สัญญาณนาฬิกาของไอซี MCS-51 หารด้วย 12 หรืออาจกล่าวได้ว่าขึ้นกับค่าความถี่ของคริสตอลที่นำมาต่อใช้งานแล้วหารด้วย 12 นั้นเอง

อัตรารับส่งโหมด 0 = ความถี่สัญญาณนาฬิกา/12

โหมด 2 อัตรารับส่งในโหมดนี้เลือกได้ 2 อัตราความเร็วในการรับส่งข้อมูล โดยหาได้จากความถี่ของสัญญาณนาฬิกาของไอซี MCS-51 หารด้วย 32 หรือหารด้วย 64 เรียกว่า SMOD0 และ SMOD1 ซึ่งขึ้นอยู่กับการกำหนดค่าสถานะของบิต SMOD ที่อยู่ในรีจิสเตอร์ PCON เป็นตัวเลือก

ถ้าบิต SMOD = 0 อัตรารับส่งโหมด 2 = 1/64 เท่าของความถี่สัญญาณนาฬิกา

ถ้าบิต SMOD = 1 อัตรารับส่งโหมด 2 = 1/32 เท่าของความถี่สัญญาณนาฬิกา

หลังจากทำการรีเซตระบบของไอซี MCS-51 ค่าข้อมูลในบิต SMOD เป็นสถานะ “0’ ดังนั้นสามารถเขียนเป็นสูตรสำหรับการคำนวณหาค่าอัตรารับส่งได้ดังสมการต่อไปนี้

$$\text{อัตรารับส่ง (Baud rate) โหมด 2} = \frac{[\ 2^{SMOD} \times (\text{ความถี่สัญญาณนาฬิกาที่ใช้})\ ]}{64}$$

ในกรณีที่ใช้คริสตอลค่า 11.0592 MHz ได้ค่าอัตรารับส่งสูงสุด = 345.6 K

ในกรณีที่ใช้คริสตอลค่า 12 MHz ได้ค่าอัตรารับส่งสูงสุด = 375 K

โหมด 1 และโหมด 3 มีอัตราการรับส่งข้อมูลโดยถูกกำหนดได้ตามต้องการ โดยใช้อัตราการเกิดค่านับเกินของไทม์เมอร์1 หรือ ไทม์เมอร์ 2 เป็นตัวกำหนด

การใช้ไทม์เมอร์1 กำหนดอัตรารับส่ง อัตราการรับส่งข้อมูลของพอร์ตอนุกรมในโหมด 1 หรือโหมด 3 สังเกตได้ว่าเมื่อเกิดค่านับเกินในไทม์เมอร์ตัวใด ทำให้เกิดสัญญาณอินเตอร์รัปต์เพื่อบอกให้ซีพียู

รับทราบ ดังนั้นในขณะใช้ไทม์เมอร์1 เพื่อสร้างอัตรารับและส่งข้อมูล ต้องไม่ให้มีการร้องขออินเตอร์รัพต์ที่เกิดจากไทม์เมอร์1 ในกรณีใดๆในระหว่างนั้นอีก (โดยการควบคุมที่ รีจิสเตอร์ IE) อัตราการรับและส่งข้อมูลมาจากอัตราการเกิดค่านับเกินของไทม์เมอร์1 และค่าของบิต SMOD ที่อยู่ในรีจิสเตอร์ PCON สามารถเขียนเป็นสูตรสำหรับการคำนวณหาค่าอัตรารับส่ง (Baud Rate) ได้ดังสมการต่อไปนี้

$$\text{อัตรารับส่งในโหมด 1 และ 3} = \frac{2^{SMOD}}{32} \times (\text{อัตราการเกิดค่านับเกินของไทม์เมอร์1}$$

ในการใช้งานสื่อสารข้อมูลแบบอนุกรมนั้น นิยมใช้งานไทม์เมอร์1 ในลักษณะของโหมด 2 (Auto-reload กำหนดค่าการควบคุมที่รีจิสเตอร์ TMOD = 0010XXXX) การคำนวณหาค่าอัตรารับส่งได้สมการดังต่อไปนี้

$$\text{อัตราการรับส่งของโหมด 1 และ 3} = \frac{2^{SMOD}}{32} \times \frac{\text{Oscillator}}{12 \times [256 - (TH1)]}$$

ตารางที่ 8.2 แสดงตัวอย่างอัตรารับส่งข้อมูลจากการใช้ ไทม์เมอร์1 เมื่อใช้ค่ามาตรฐานต่างๆ

Baud rate	Fosc (MHz)	บิต SMOD	Timer1		
			บิต C/T	MODE	Reload Value
62.5 K	12.00	1	0	2	FFH
19.25 K	11.059	1	0	2	FDH
9600	11.059	0	0	2	FDH
4800	11.059	0	0	2	FAH
2400	11.059	0	0	2	F4H
1200	11.059	0	0	2	E8H
137.5	11.059	0	0	2	1DH
110	6	0	0	2	72H
110	12	0	0	1	FEEBH

(แหล่งอ้างอิง http://atmel.com/dyn/products/datasheets.asp?family_id=604)

ไทม์เมอร์1 สามารถกำหนดให้มีอัตราการรับส่งต่ำๆ ได้โดยใช้ ไทม์เมอร์1 ในโหมด1 ทำงานลักษณะของตัวจับเวลาแบบ 16 บิต (ค่าของการควบคุมที่รีจิสเตอร์ TMOD = 0001XXXX) และเมื่อมีการ

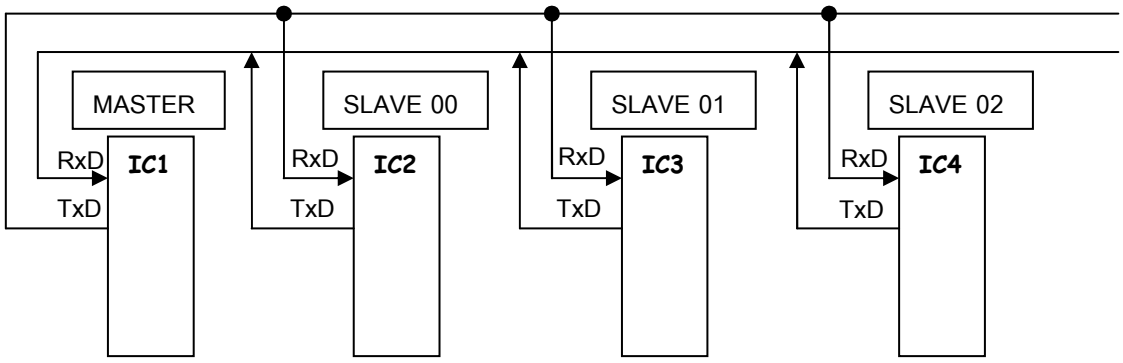
อินเตอร์รัปต์จากไทม์เมอร์1 โดยโปรแกรมตอบสนองการอินเตอร์รัปต์ของไทม์เมอร์1 แล้ว กำหนดค่าเริ่มต้นใหม่ (Reload) ให้กับตัวจับเวลา ซึ่งเป็นการทำงานแบบ 16 บิตทางซอร์ฟแวร์ (Software Reload) เนื่องจากการทำงานในโหมด 1 ของไทม์เมอร์1 ไม่สามารถโหลดค่าใหม่เองด้วยฮาร์ดแวร์ได้

จากตารางที่ 8.2 แสดงอัตรารับส่งข้อมูลที่กำหนดจากการใช้ ไทม์เมอร์1 เมื่อใช้อัตรารับส่งค่ามาตรฐานต่างๆ และการกำหนดค่าของรีจิสเตอร์ TMOD ที่บิต C/T และรีจิสเตอร์ PCON ที่บิต SMOD เพื่อใช้งาน ค่าความถี่ของคริสตอลมักนิยมเลือกใช้ค่าความถี่ 11.0592 MHz เนื่องจากค่าอัตราการรับส่งข้อมูล สามารถกำหนดค่าโหมด 1 และโหมด3 ได้ และเป็นค่ามาตรฐานเช่น 1200, 2400, 4800, 9600, 19200

5. การสื่อสารข้อมูลระบบมัลติโฟรเซสเซอร์

การสื่อสารข้อมูลระบบมัลติโฟรเซสเซอร์ (Multiprocessors System) คือการนำไอซี MCS-51 จำนวนหลายๆ ตัว มาเชื่อมต่อ เพื่อทำการสื่อสารสารข้อมูลกันนั้นต้องมีตัวหลักอยู่ 1 ตัว (Master) ส่วนตัวอื่นๆ เป็นตัวลูก (Slave) และสามารถนำตัวลูกมาต่อร่วมได้ถึง 256 ตัวแต่ต้องมีการกำหนดค่าแอดเดรสของไอซีในแต่ละตัว เพื่อให้ไอซีคอนโทรลเลอร์ตัวหลักสามารถระบุการติดต่อ กับตัวลูกตัวนั้นๆได้

การกำหนดค่าแอดเดรสของตัวลูกสามารถกำหนดได้ตั้งแต่ค่า 00H-FFH และทำการต่อขาสัญญาณ TxD ของไอซี MCS-51 ตัวหลักเข้ากับขาสัญญาณของ RxD ของตัวลูกทุกๆ ตัวด้วย แสดงดังภาพที่ 8.13



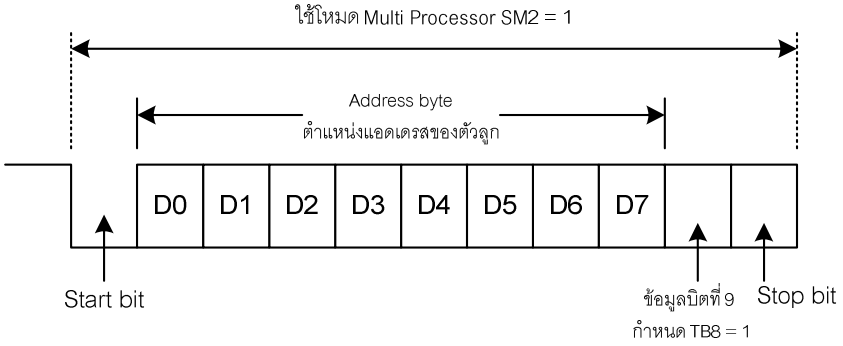
ภาพที่ 8.13 การเชื่อมต่อไอซี MCS-51หลายๆตัวเข้าด้วยกัน

สัญญาณที่ใช้สื่อสารอนุกรมแบบมัลติโพรเซสเซอร์ ในการรับส่งใช้โหมด 2 และโหมด 3 เท่านั้น โดยการกำหนดค่ารีจิสเตอร์ SCON ที่บิต SM2 = 1 วิธีการส่งข้อมูลทางพอร์ตอนุกรมส่งข้อมูลขนาด 11 บิต และมีการรับส่งข้อมูลของบิตที่ 9 หรือTB8 และ RB8 โดยกำหนดของสถานะในบิตที่ 9 ในรีจิสเตอร์ SCON ให้เป็นสถานะลอจิก 1 หรือ 0 เพื่อระบุให้เป็นสัญญาณการส่งค่าตำแหน่งแอดเดรสของตัวลูกในแต่ละตัวได้ และยังระบุให้เป็นการส่งค่าข้อมูลให้กับตัวลูกตัวนั้นๆได้เช่นเดียวกัน

5.1 การรับและส่งข้อมูลเพื่อกำหนดตำแหน่งแอดเดรสของตัวลูก

เมื่อไอซี MCS-51 ตัวหลักต้องการติดต่อกับไอซี MCS-51ตัวลูก ในลำดับแรกต้องส่งค่าข้อมูลขนาด 1 ไบต์ ซึ่งใช้เป็นข้อมูลที่กำหนดตำแหน่งแอดเดรสของตัวลูกแต่ละตัวที่ต้องการติดต่อออกไปก่อน ถ้ากำหนดทางด้านส่งให้บิตที่ 9 หรือ TB8 = 1 หมายถึงการเลือกส่งแอดเดรสไบต์ และในการส่งค่า 1ไบต์นี้ออกไป ทำให้ตัวลูกได้รับค่าเหมือนกันทั้งหมดทุกตัว แสดงดังภาพที่ 8.14

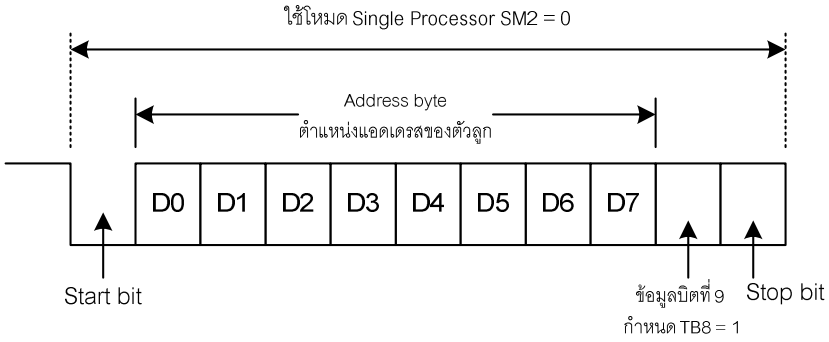
ส่วนทางด้านรับของตัวลูกทุกตัวต้องกำหนดให้ค่าที่บิต SM2 = 1 (Multiprocessors Mode) โดยหลังจากถูกขัดจังหวะการทำงาน เพื่อรับค่าข้อมูลเข้าไป หากค่าของบิตที่ 9 หรือ RB8 = 1 หมายถึงค่าข้อมูลที่รับมาได้ เป็นการระบุแอดเดรสไบต์ (Address Byte) ลำดับต่อมาไอซี MCS-51 แต่ละตัว กระทำตามโปรแกรมคำสั่ง โดยนำค่าที่ได้ไปเปรียบเทียบกับค่าประจำแอดเดรสของตัวเอง ถ้าหากมีค่าแอดเดรสตรงกับตัวลูกตัวใด ตัวนั้นทำการเคลียร์ที่บิต SM2 = 0 และเข้าสู่ขั้นตอนการรับข้อมูลในลำดับต่อไป ส่วนตัวที่มีค่าแอดเดรสไม่ตรงกับค่าที่ส่งมา กลับไปทำงานเดิมที่ค้างไว้ หลังจากสิ้นสุดการขัดจังหวะการทำงาน



ภาพที่ 8.14 การส่งข้อมูล 1 ไบต์แรกเพื่อกำหนดตำแหน่งแอดเดรสของตัวลูก

5.2 การส่งข้อมูลให้กับตัวลูกที่ถูกเลือก

การกำหนดข้อมูลให้กับตัวลูกที่ถูกเลือก ถ้าหากทางด้านส่งให้ค่าข้อมูลในบิตที่ 9 หรือ TB8 = 0หมายถึงเป็นการส่งไบต์ข้อมูล (Data Bytes) ส่วนทางด้านรับ ถ้ารับข้อมูลในบิตที่ 9 หรือ RB8 = 0 หมายถึงค่าที่รับเป็นค่าของไบต์ข้อมูล ซึ่งแสดงดังภาพที่ 8.15 หลังจากนั้นต้องกำหนดค่าตัวลูกที่ถูกเลือกให้บิต SM2 = 0 เพื่ออยู่ใน Single Processer Mode ดังนั้นจึงมีแต่ตัวลูกที่ถูกเลือกเท่านั้น สามารถทำการรับค่าข้อมูลจากตัวหลักได้ หลังจากที่ตัวลูกได้รับค่าข้อมูลจนเสร็จสิ้นแล้ว ต้องกำหนดให้ที่บิต SM2 = 1 (Multiprocessors Mode) เพื่อเริ่มต้นรับค่าตำแหน่งแอดเดรส เมื่อมีการขัดจังหวะใหม่ในครั้งต่อไป

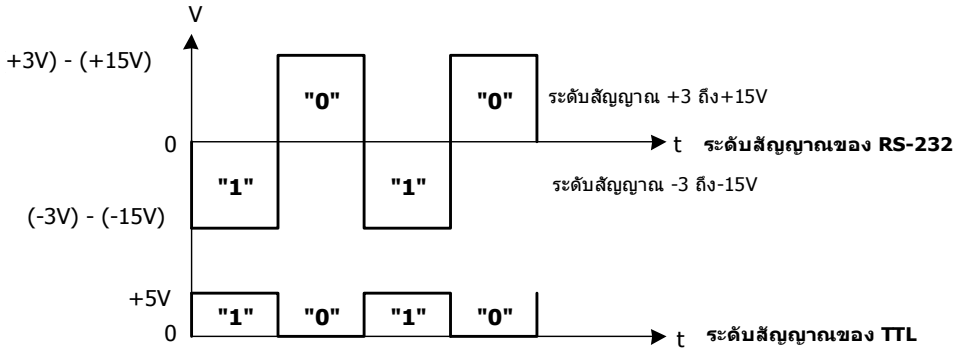


ภาพที่ 8.15 การส่งไบต์ข้อมูลให้กับตัวลูก

6. มาตรฐานอนุกรม RS-232

6.1 มาตรฐานการเชื่อมต่อแบบ RS-232

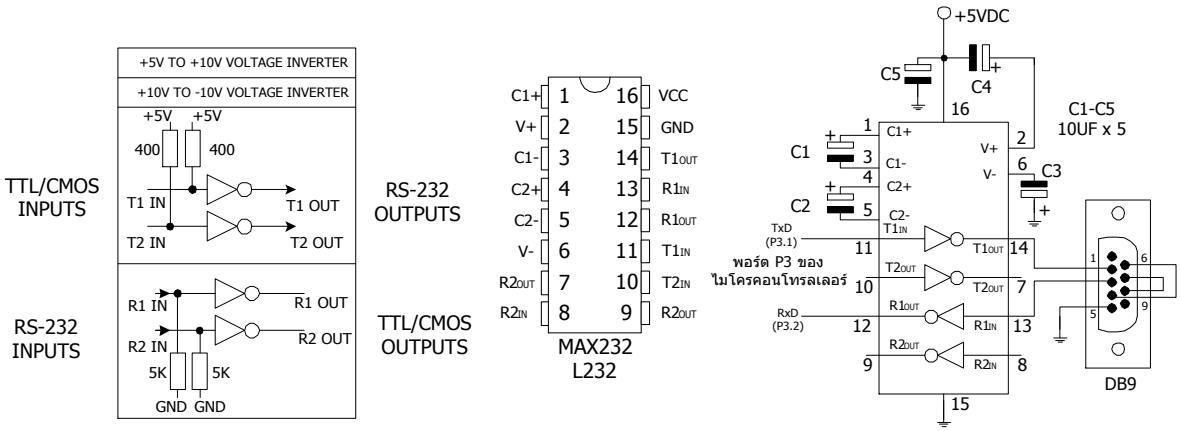
เป็นมาตรฐานอุตสาหกรรม ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง เพื่อให้มีการใช้งานในการเชื่อมต่อ ที่สอดคล้องกัน ระหว่างอุปกรณ์คอมพิวเตอร์ต่างๆ การรับส่งสัญญาณกำหนดความยาวสูงสุดไว้ที่ไม่เกิน 50 ฟุต โดยมีระดับสัญญาณตั้งแต่ 3 โวลต์ ถึง 15 โวลต์ สำหรับลอจิก “0” และมีระดับแรงดันที่ -3 โวลต์ ถึง -15 โวลต์ สำหรับลอจิก “1” ดังแสดงในภาพที่ 8.16 สังเกตได้ว่ามีระดับแรงดันที่ใช้ในสถานะลอจิก “0” และ ลอจิก “1” แตกต่างออกไปจากระบบไอซีดิจิตอลทั่วไป การต่อใช้งานจึงต้องมีอุปกรณ์ที่ทำหน้าที่เปลี่ยนระดับแรงดันจาก 0 – 5 โวลต์ จากไอซี MCS-51 ให้เป็นระดับแรงดันที่สูงกว่า +3 หรือต่ำกว่า - 3 โดยมีไอซีสำเร็จรูปพร้อมใช้งาน หรืออาจต่อวงจรจากทรานซิสเตอร์ได้



ภาพที่ 8.16 แสดงระดับแรงดันสัญญาณ RS-232 กับ TTL ในสถานะลอจิก “1” และ “0”

6.2 การแปลงระดับสัญญาณ TTL เป็น RS-232

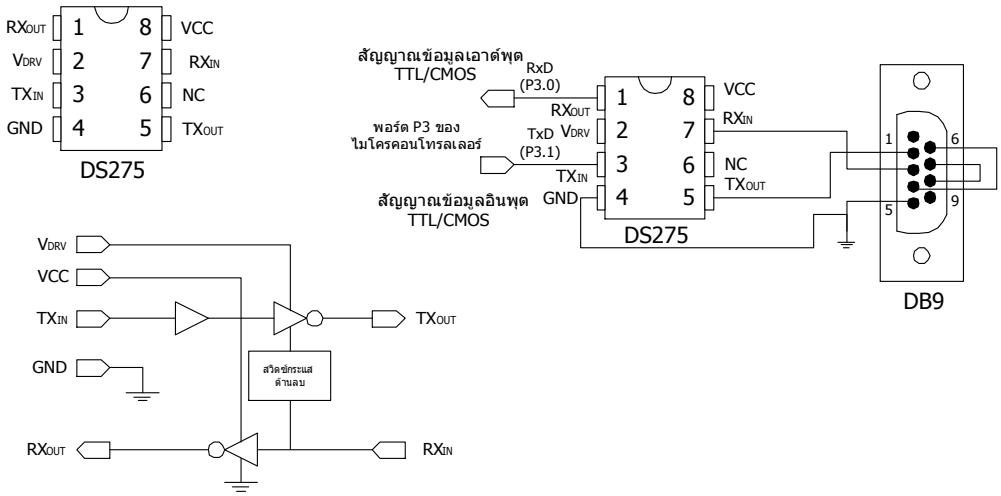
6.2.1 ไอซี MAX232 เป็นไอซีแปลงระดับสัญญาณจากระดับ TTL ไปเป็นระดับของ RS-232 แสดงดังภาพที่ 8.17



ภาพที่ 8.17 แสดงตำแหน่งขาของไอซี MAX232 และการต่อใช้งาน

(แหล่งอ้างอิง http://www.maxim-ic.com/quick_view2.cfm/qv_pk/1798)

6.2.2 ไอซี DS275 มีขนาด 8 ขา แปลงระดับสัญญาณจากระดับ TTL ไปเป็นระดับของ RS-232 การนำไปใช้งานไม่ต้องมีอุปกรณ์ต่อร่วม แสดงดังภาพที่ 8.18



ภาพที่ 8.18 แสดงตำแหน่งขาของไอซี DS275 และการต่อใช้งาน

(แหล่งอ้างอิง http://www.maxim-ic.com/quick_view2.cfm/qv_pk/2929)

รายละเอียดของขาใช้งาน

RXOUT (ขา 1) : สัญญาณเอาต์พุตของด้านรับ RS-232

VDRV (ขา 2) : ขารับแรงดัน +V ของด้านส่ง

TXIN (ขา 3) : ขารับสัญญาณอินพุตด้านส่ง RS-232

GND (ขา 4) : กราวด์

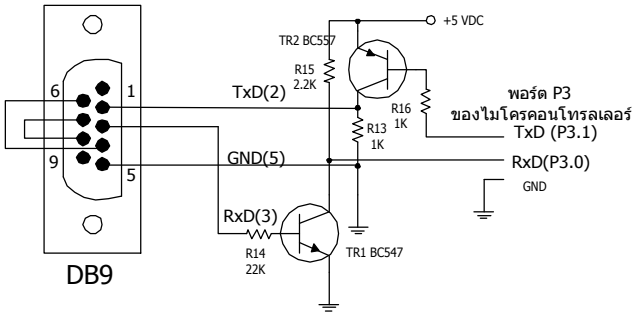
VCC (ขา 5) : ขารับไฟเลี้ยง +5 โวลต์

RXIN (ขา 6) : ขารับสัญญาณอินพุตด้านรับ RS-232

NC (ขา 7) : ไม่ใช้งาน

TXOUT (ขา 8) : ขารับสัญญาณเอาต์พุตด้านส่ง RS-232

6.2.3 วงจรแปลงระดับสัญญาณจาก TTL ไปเป็นระดับของ RS-232 จากวงจรทรานซิสเตอร์ แสดงดังภาพที่ 8.19



ภาพที่ 8.19 แสดงการต่อวงจรโดยใช้ทรานซิสเตอร์

(แหล่งอ้างอิง http://www.adisak51/page06_1.html)

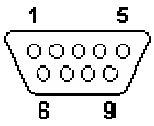
7. การเชื่อมต่อระหว่างพอร์ตอนุกรมของคอมพิวเตอร์ กับไอซี MCS-51

7.1 การสร้างสายเชื่อมต่อระหว่างไมโครคอมพิวเตอร์ กับ ไอซี MCS-51

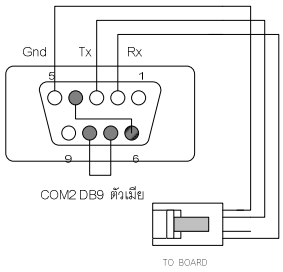
ใช้มาตรฐาน RS-232 ผ่านพอร์ตอนุกรม Com1 หรือ Com2 แบบ DB9 ของเครื่องคอมพิวเตอร์พีซี แสดงดังภาพที่ 8.20

DB9 pin

D-SUB Male

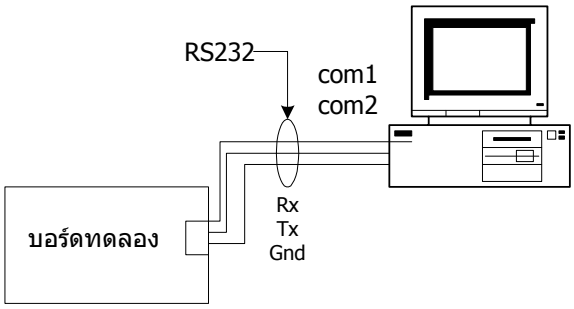


Pin	Name	Direction	Description
1	CD	«—	Carrier Detect
2	RXD	«—	Receive Data
3	TXD	—»	Transmit Data
4	DTR	—»	Data Terminal Ready
5	GND		System Ground
6	DSR	«—	Data Set Ready
7	RTS	—»	Request to Send
8	CTS	«—	Clear to Send
9	RI	«—	Ring Indicator



ภาพที่ 8.20 แสดงตำแหน่งขา DB9 และการต่อสายกับ Com Port ของคอมพิวเตอร์

ต่อสาย DB9 เพื่อติดต่อกับคอมพิวเตอร์พอร์ต Com1หรือพอร์ต Com2 เข้ากับบอร์ดไอซี MCS-51 แสดงดังภาพที่ 8.21



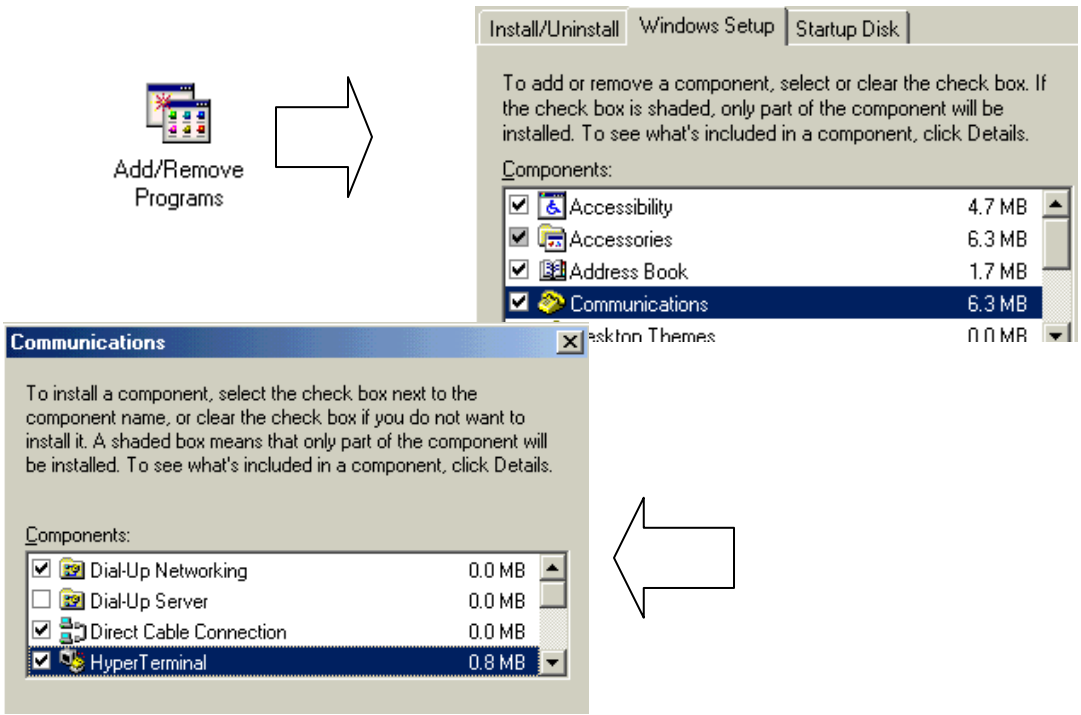
ภาพที่ 8.21 การต่อใช้งานบอร์ดไอซี MCS-51 กับคอมพิวเตอร์ตำแหน่ง Com1 (หรือ Com2)

7.2 การติดตั้ง และใช้งานโปรแกรม Hyper Terminal

7.2.1 การติดตั้งโปรแกรม Hyper Terminal

ขั้นตอนที่ 1 เริ่มต้นติดตั้งโดย คลิก **Start >> Setting >> Control Panel**

ขั้นตอนที่ 2 เลือกไอคอน **Add/Remove Program** ให้เลือกที่ **Communications** เลือกติดตั้งโปรแกรม **Hyper Terminal** แสดงดังภาพที่ 8.22



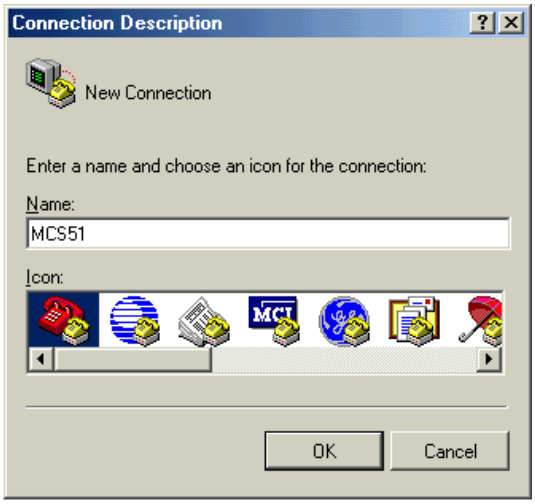
ภาพที่ 8.22 แสดงการเปิด และแสดงการติดตั้งโปรแกรม Hyper Terminal

7.2.2 การใช้งานโปรแกรม Hyper Terminal

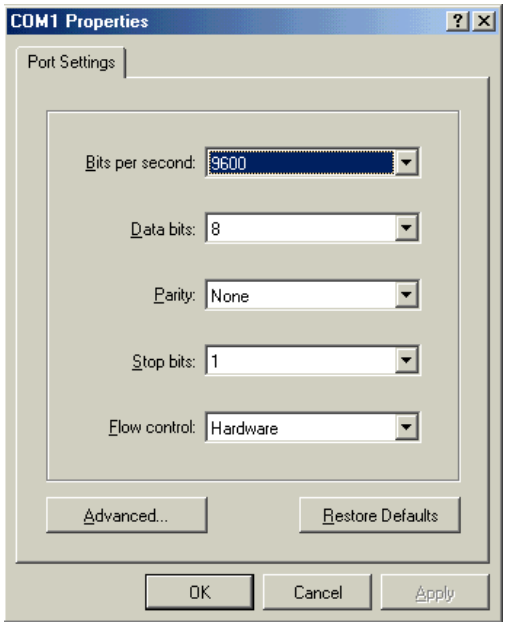
ขั้นตอนที่ 1 เปิดโปรแกรม **Hyper Terminal** โดยคลิกที่ปุ่ม **Start**

ขั้นตอนที่ 2 เลือก **Programs >> Accessories >> Communications >> Hyper Terminal**

ขั้นตอนที่ 3 ตั้งชื่อได้ตามต้องการ แสดงดังภาพที่ 8.23 ก หลังจากนั้นเลือกตำแหน่งพอร์ตที่ต้องการติดต่อกับบอร์ดแสดงดังภาพที่ 8.23 ข เลือกอัตราการรับส่งข้อมูลโดดให้เลือก 9600 บิตต่อวินาที (Bit per second) แสดงดังภาพที่ 8.23 ค



ภาพที่ 8.23 ก



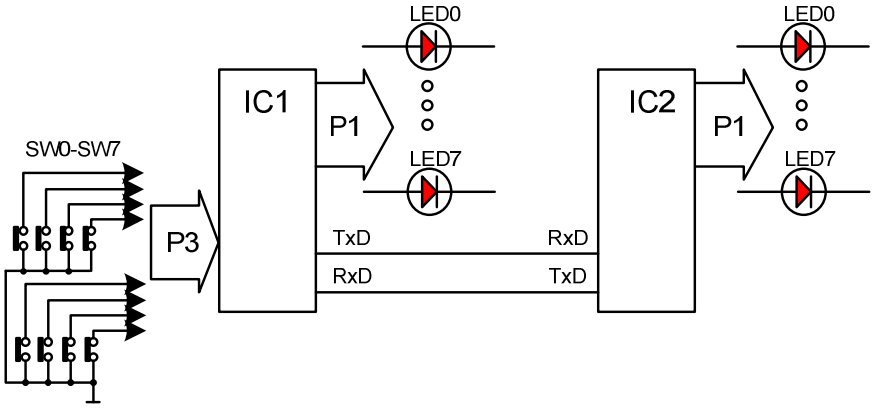
ภาพที่ 8.23 ค

ภาพที่ 8.23 ข

ภาพที่ 8.23 แสดงวิธีการเปิดโปรแกรม Hyper Terminal เพื่อติดต่อกับพอร์ตอนุกรม

8. การเขียนโปรแกรมสื่อสารข้อมูลทางพอร์ตอนุกรม

ตัวอย่างที่ 1 เขียนโปรแกรมรับส่งข้อมูลทางพอร์ตอนุกรมในโหมด 1



ภาพที่ 8.24 การเชื่อมต่อไอซี MCS-51 IC1และ IC2 เข้าด้วยกัน

เขียนโปรแกรมสื่อสารพอร์ตอนุกรมโหมด1 ส่งข้อมูลถูกกำหนดโดยพอร์ต P3 ของ IC1ไปที่ IC2 กำหนดอัตรารับส่งข้อมูลที่ 2400 บิตต่อวินาที (Bit Per Second) พร้อมแสดงค่าของข้อมูลที่ส่ง และรับที่พอร์ต P1ของ IC1 และ IC2 ด้วย แสดงดังภาพที่ 8.24 ขั้นตอนการกำหนดการเขียนโปรแกรมมีดังนี้

ขั้นตอนที่ 1 กำหนดค่ารีจิสเตอร์ SCON ให้รับส่งทางพอร์ตอนุกรมโหมด1

ขั้นตอนที่ 2 กำหนดการใช้ไทม์เมอร์ 1 ในโหมด 2 เพื่อกำหนดอัตราการรับส่งข้อมูล

ขั้นตอนที่ 3 กำหนดบิต SMOD ของรีจิสเตอร์ PCON ให้บิตมีค่าเป็น “0”

การส่งข้อมูลออกไป นำค่าข้อมูลในแอดเดรส 20H เก็บไว้ที่รีจิสเตอร์ SBUF หลังจากนั้นตรวจสอบบิต TI ถ้าไม่ถูกเซตให้วนส่งข้อมูลจนกว่าครบทุกบิต เมื่อครบแล้วบิต TI ถูกเซต หลังจากนั้นต้องเคลียร์บิต TI เพื่อเตรียมส่งข้อมูลในไบต์ต่อไป

การคำนวณหาค่าของ TH1 เพื่อกำหนดอัตราการรับส่งข้อมูลที่ 2400 BPS สามารถหาได้จาก

$$\textbf{อัตราการรับส่งของโหมด 1 และ 3} = \frac{2^{SMOD}}{32} \times \frac{\text{Oscillator}}{12 \times \left[256 - (TH1)\right]}$$

$$\textbf{แทนค่า} \qquad 2400 = \frac{2^{0}}{32} \times \frac{11.059 \times 10^{6}}{12 \times \left[256 - (TH1)\right]}$$

$$TH1 = 244$$

แปลงเป็นเลขฐานสิบหก $TH1 = F4H$

```
;***********************************************
;***  IC1:  Transmit Data Serial (Mode1) 2400 Baud Tx   ***
;***********************************************
        ORG   0000H              ; เริ่มต้นที่แอดเดรส 0050H
        MOV   P1,#00H            ;กำหนดให้พอร์ต P1 เป็น 00 แอลอีดีดับทุกดวง
        MOV   P3,#0FFH           ;กำหนดให้พอร์ต P3 เป็นสถานะสูง(อินพุตพอร์ต)
        MOV   SCON,#01000000B    ;กำหนดโหมดการรับส่งพอร์ตอนุกรมในโหมด1
        MOV   TMOD,#20H          ;กำหนดใช้ไทม์เมอร์ 1 โหมด 2 Auto Reload
        MOV   PCON,#00000000B    ;กำหนดอัตราการรับส่งข้อมูล
        MOV   TH1,#0F4H          ; กำหนดอัตราการรับส่งข้อมูล  2400 บิตต่อวินาที
        SETB  TR1                ; เริ่มทำงานของโหมดตั้งเวลาไทม์เมอร์1
LOOP:   MOV   P1,P3      ; นำข้อมูลจาก P3 ไปเก็บไว้แสดงผลที่ P1
        MOV   SBUF, P1   ; นำข้อมูลในจาก P1 ไปเก็บไว้ที่รีจิสเตอร์ SBUF
        JNB   TI,$       ; ถ้าบิต TI ไม่ถูกเซตวนส่งข้อมูลจนกว่าครบ
        CLR   TI         ; เคลียร์บิต TI เพื่อเตรียมส่งข้อมูลในไบต์ต่อไป
        SJMP  LOOP       ; กระโดดกลับไปส่งข้อมูลใหม่ที่เลเบล LOOP
        END              ; จบโปรแกรม
```

เขียนโปรแกรมรับข้อมูลทางพอร์ตอนุกรมโหมด 1 โดยรับค่ามาจาก IC1 กำหนดอัตราการรับส่งข้อมูล 2400 บิตต่อวินาที ข้อมูลที่รับได้แสดงผลที่พอร์ต P1 ของ IC2 ด้วย ขั้นตอนเขียนโปรแกรมมีดังนี้

ขั้นตอนที่ 1 กำหนดโหมดการรับส่งพอร์ตอนุกรมโหมด 1 ในรีจิสเตอร์ SCON บิต REN ให้มีการรับข้อมูล

ขั้นตอนที่ 2 กำหนดการใช้ไทม์เมอร์ 1 โหมด 2 เพื่อกำหนดอัตราการรับส่งข้อมูล

ขั้นตอนที่ 3 กำหนดบิต SMOD ของรีจิสเตอร์ PCON ให้บิตนี้มีค่าเป็น “0”

การรับข้อมูลเข้ามาตรวจสอบที่บิต RI ถ้าไม่ถูกเซต ให้วนรับข้อมูลจนครบทุกบิต หลังจากนั้นบิต RI ถูกเซต และต้องเคลียร์บิต RI นำค่าข้อมูลจากรีจิสเตอร์ SBUF ไปแสดงผลที่พอร์ต P1 วนลูปโปรแกรมเพื่อรับข้อมูลในไบต์ต่อไป

```
;***********************************************
;***  IC2:  RECEIVE Data Serial (Mode1) 2400 Baud Rx  ***
;***********************************************
ORG   0000H              ; เริ่มต้นที่แอดเดรส 0050H
MOV   P1,#00H            ; กำหนดให้พอร์ต P1 เป็น 00 แอลอีดีดับทุกดวง
MOV   SCON,#50H          ;โหมดการรับส่งพอร์ตอนุกรมในโหมด1 และกำหนดบิต REN
```

```
        MOV   TMOD,#20H          ; กำหนดใช้ไทม์เมอร์ 1 โหมด 2
        MOV   PCON,#00000000B    ; กำหนดอัตราการรับส่งข้อมูล
        MOV   TH1,#0F4H          ; กำหนดอัตราการรับส่งข้อมูล  2400 บิตต่อวินาที
        SETB  TR1                ; เริ่มทำงานของโหมดตั้งเวลาไทม์เมอร์1
LOOP:   JNB   RI,$               ; ถ้าบิต RI ไม่ถูกเซตให้วนรับข้อมูลจนครบ และเซตบิต RI
        CLR   RI                 ; เคลียร์บิต RI เพื่อเตรียมรับข้อมูลในไบต์ต่อไป
        MOV   A, SBUF            ; นำข้อมูลในรีจิสเตอร์ SBUFไปเก็บไว้ที่รีจิสเตอร์ A
        MOV   P1,A               ; นำค่าข้อมูลในรีจิสเตอร์ A ไปแสดงผลที่พอร์ต P1
        SJMP  LOOP               ; กระโดดกลับไปรับข้อมูลใหม่ที่เลเบล LOOP
        END                      ; จบโปรแกรม
```

ตัวอย่างที่ 2 เขียนโปรแกรมสื่อสารทางพอร์ตอนุกรมโหมด 1 ระหว่างไมโครคอมพิวเตอร์ PC กับไอซี MCS-51 กำหนดให้ส่งค่าข้อมูลจาก IC1 กำหนดอัตราการรับส่งข้อมูลที่อัตราการรับส่ง 9600 บิตต่อวินาที โดยข้อมูลกำหนดจากตาราง Lookup Table นำไปเก็บไว้ที่รีจิสเตอร์ SBUF ทีละไบต์ และส่งข้อมูลออกไปให้กับไมโครคอมพิวเตอร์ทีละไบต์แสดงได้ดังภาพที่ 8.25 ขั้นตอนเขียนโปรแกรมมีดังนี้

ขั้นตอนที่ 1 กำหนดโหมดการรับส่งพอร์ตอนุกรมโหมด 1 ในรีจิสเตอร์ SCON

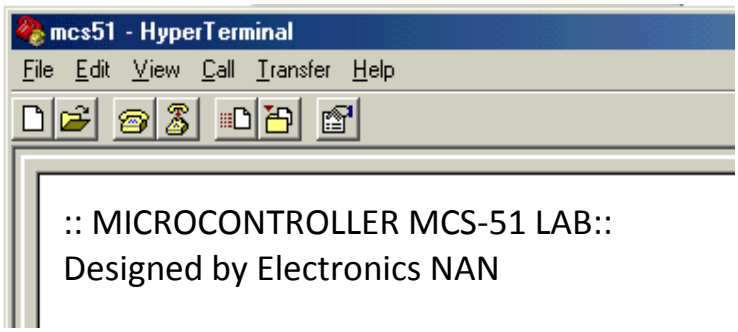
ขั้นตอนที่ 2 กำหนดการใช้ไทม์เมอร์ 1 โหมด 2 เพื่อกำหนดอัตราการรับส่งข้อมูล

ขั้นตอนที่ 3 กำหนดบิต SMOD ของรีจิสเตอร์ PCON ให้บิตนี้มีค่าเป็น “0”

ขั้นตอนที่ 4 การส่งข้อมูลออกไป นำค่าข้อมูลในแอดเดรส 20H ไปเก็บไว้ที่รีจิสเตอร์ SBUF หลังจากนั้น ตรวจสอบบิต TI ถ้าไม่ถูกเซต ให้วนส่งข้อมูลจนกว่าส่งข้อมูลครบทุกบิตแล้วบิต TI จะถูกเซต โดยต้องเคลียร์บิต TI เพื่อเตรียมส่งข้อมูลในไบต์ต่อไป

```
;***********************************************************
;***   Transmit Data Serial (Mode1) 9600 Baud To Microcomputer ***
;***********************************************************
        CR      EQU   0DH        ; Carriage return
        LF      EQU   0AH        ; Line feed
        ORG     0000H            ; เริ่มต้นที่แอดเดรส 0050H
        MOV     P1,#00H          ; กำหนดให้พอร์ต P1 เป็น 00 แอลอีดีดับทุกดวง
        MOV     P3,#0FFH         ; กำหนดให้พอร์ต P3 เป็นสถานะสูง(อินพุตพอร์ต)
        MOV     SCON,#01000000B  ; กำหนดโหมดการรับส่งพอร์ตอนุกรมในโหมด1
        MOV     TMOD,#20H        ; กำหนดใช้ไทม์เมอร์ 1 โหมด 2 Auto Reload
        MOV     PCON,#00000000B  ; กำหนดอัตราการรับส่งข้อมูล
        MOV     TL1,#0FDH        ; กำหนด Baud Rate 9600 BPS
```

```
                MOV   TH1,#0FDH          ; กำหนด Baud ret 9600 BPS
                SETB  TR1                ; เริ่มทำงานของโหมดตั้งเวลาไทเมอร์ 1
        ;****************************************
        ;***    ส่งข้อมูลในตาราง Lookup Table        ****
        ;****************************************
                MOV   DPTR,#SHOW  ; กำหนดค่าของฐานแอดเดรสให้กับรีจิสเตอร์ DPTR
                MOV   R0,#00H     ; กำหนดให้รีจิสเตอร์ R0 = 00H
SEND_SHOW:      MOV   A,R0        ; นำค่าข้อมูลในรีจิสเตอร์ R0 เก็บไว้ที่รีจิสเตอร์ A
                MOVC  A,@A+DPTR   ; เปิดตารางข้อมูล แล้วนำข้อมูลที่ได้มาเก็บไว้ที่รีจิสเตอร์ A
                MOV   SBUF,A      ; นำข้อมูลในรีจิสเตอร์ A ไปเก็บไว้ที่รีจิสเตอร์ SBUF
                JNB   TI,$        ; ถ้าบิต TI ไม่ถูกเซตวนส่งข้อมูลจนครบ หลังจากนั้นเซตบิต TI
                CLR   TI          ; เคลียร์บิต TI เพื่อเตรียมส่งข้อมูลในไบต์ต่อไป
                INC   R0          ; เพิ่มค่าข้อมูลในรีจิสเตอร์ R0
                JNZ   SEND_SHOW   ; ถ้าข้อมูลไม่เท่ากับ 00H ให้กระโดดกลับไปส่งข้อมูลใหม่
;*********************************************
;****   ข้อมูลที่ส่งให้กับไมโครคอมพิวเตอร์       *****
;*********************************************
SHOW:           DB    CR,LF," :: MICROCONTROLLER MCS-51 LAB ::  ",CR
                DB    CR,LF,"  Designed  by  Electronics NAN ",CR,LF
                DB    00H
                END
```



ภาพที่ 8.25 แสดงข้อความเมื่อมีการติดต่อระหว่างIC MCS-51 กับไมโครคอมพิวเตอร์ PC

ตัวอย่างที่ 3 เขียนโปรแกรมสื่อสารทางพอร์ตอนุกรมโหมด1 รับค่าข้อมูลจากไมโครคอมพิวเตอร์ PC กำหนดอัตราการรับส่งข้อมูลที่ 9600 บิตต่อวินาที ข้อมูลถูกนำไปเก็บไว้ที่รีจิสเตอร์ SBUF ทีละไบต์ ข้อมูลที่รับได้เป็นรหัส ASCII ให้แสดงผลที่พอร์ต P1 ตามค่าที่รับได้ ขั้นตอนเขียนโปรแกรมมีดังนี้

ขั้นตอนที่ 1 กำหนดโหมดการรับส่งพอร์ตอนุกรมโหมด 1 ในรีจิสเตอร์ SCON

ขั้นตอนที่ 2 กำหนดการใช้ไทม์เมอร์ 1 โหมด 2 เพื่อกำหนดอัตราการรับส่งข้อมูล

ขั้นตอนที่ 3 กำหนดบิต SMOD ของรีจิสเตอร์ PCON ให้บิตนี้มีค่าเป็น “0”

```
;*************************************************************
;****    การรับข้อมูลจาก Computer PC ผ่านพอร์ตอนุกรม: RS-232        ***
;*************************************************************
            ORG   0000H
MAIN:       MOV   P1, #00H            ; ให้สถานะของแอลอีดีที่ต่อกับพอร์ต P1 ให้ดับทุกดวง
            MOV   TMOD, #00100000B    ; ใช้งานพอร์ตอนุกรมในโหมด 1
            MOV   TL1, #0FDH          ; กำหนดค่าการนับในรีจิสเตอร์ TL1
            MOV   TH1, #0FDH          ; กำหนดค่าเริ่มต้นการนับในรีจิสเตอร์ TH1
            MOV   SCON, #01010000B
            SETB  TR1                 ; เริ่มต้นการทำงาน
LOOP:       ACALL RECEIVE             ; เรียกโปรแกรมย่อยรับข้อมูลจาก พอร์ต RS232
            MOV   P1,A                ; นำข้อมูลปัจจุบันในรีจิสเตอร์ A แสดงผลที่พอร์ต P1
            SJMP  LOOP                ; วนกลับไปรับค่าข้อมูลที่ถูกส่งมาใหม่
;*************************************************************
;***  โปรแกรมย่อยรับค่าข้อมูลจาก PC ผ่านทาง RS232 เข้ามาที่ MCS51     ***
;*************************************************************
RECEIVE:    JNB   RI, $               ; ตรวจสอบบิต RI รับข้อมูลไว้ใน SBUF ทุกบิตแล้วหรือไม่
            CLR   RI                  ; เคลียร์บิต RI เมื่อรับข้อมูลเก็บใน SBUF หมดทุกบิตแล้ว
            MOV   A, SBUF             ; นำค่าในรีจิสเตอร์ SBUF ไปเก็บไว้ที่รีจิสเตอร์ A
            RET
            END
```

9. การประยุกต์ใช้งานการรับส่งข้อมูลแบบอนุกรม

9.1 คุณสมบัติวงจรใช้งาน

วงจรการใช้งานทางพอร์ตอนุกรม โดยได้นำเอาคอมพิวเตอร์ PC ทำงานร่วมกับไอซี MCS-51 ผ่านทางพอร์ตอนุกรม RS-232 โดยมีขอบเขตดังนี้

9.1.1 ใช้โปรแกรมบนระบบปฏิบัติการวินโดว์เพื่อควบคุมได้

9.1.2 มีแอลอีดีแสดงผลเป็นรหัส ASCII หลังจากรับ และส่งสัญญาณทางพอร์ตอนุกรม

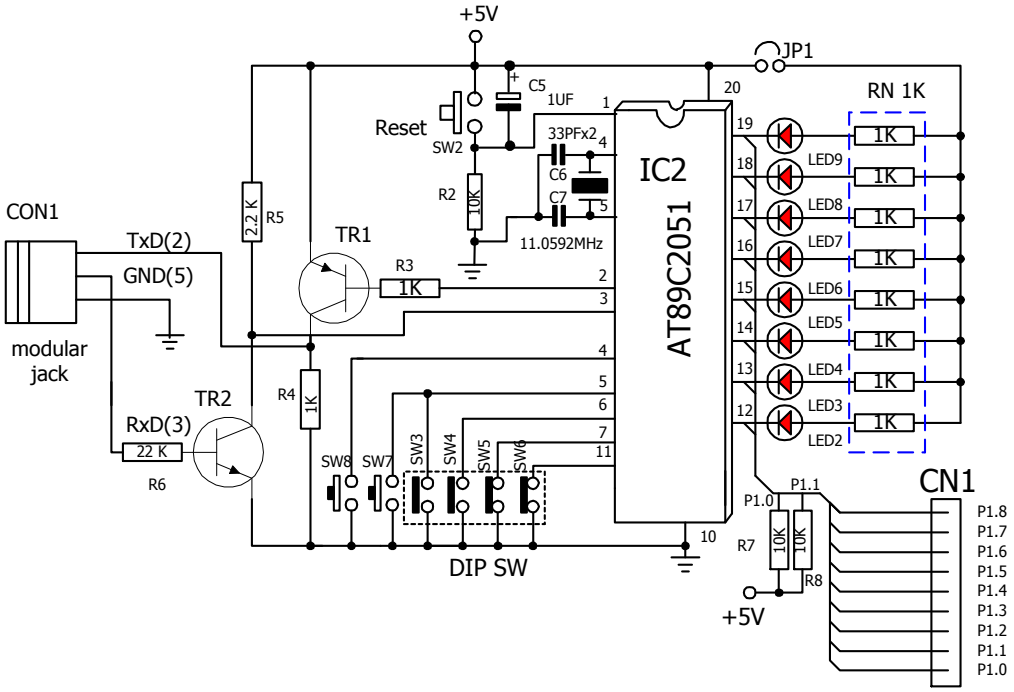
9.1.3 เก็บค่าการแสดงผลไว้ตำแหน่งหน่วยความจำข้อมูล และเรียกแสดงผลได้ทีละลำดับได้

9.1.4 รับสัญญาณอินพุตจากดิฟสวิตซ์ 4 ช่อง โดยส่งข้อมูลเป็นรหัส ASCII 0 – F แสดงผลที่แอลอีดี และบนจอมอนิเตอร์

9.1.5 ดัดแปลงเพื่อควบคุมอุปกรณ์ไฟฟ้า และรับสัญญาณอินพุตต่าง ๆ ได้

9.2 การทำงานของวงจร

การทำงานของวงจร ไอซี MCS-51 เบอร์ AT89C2051 ทำหน้าที่ติดต่อพอร์ตอนุกรมมาตรฐาน RS-232 กับไมโครคอมพิวเตอร์ และได้กำหนดอัตราความเร็วในการรับ ส่งข้อมูลไว้ที่ 9600 บิตต่อวินาที LED2 – LED9 ทำหน้าที่แสดงผลที่รับเข้ามาหรือส่งข้อมูลออกไป ทางพอร์ตอนุกรม R7, R8 ต่อเป็นวงจรพูลอัพให้กับพอร์ต P1.0 และ P1.1 คริสตอล X1, C6, C7 ทำหน้าที่ผลิตความถี่เป็นสัญญาณนาฬิกาให้กับไอซี MCS-51 แสดงดังภาพที่ 8.26

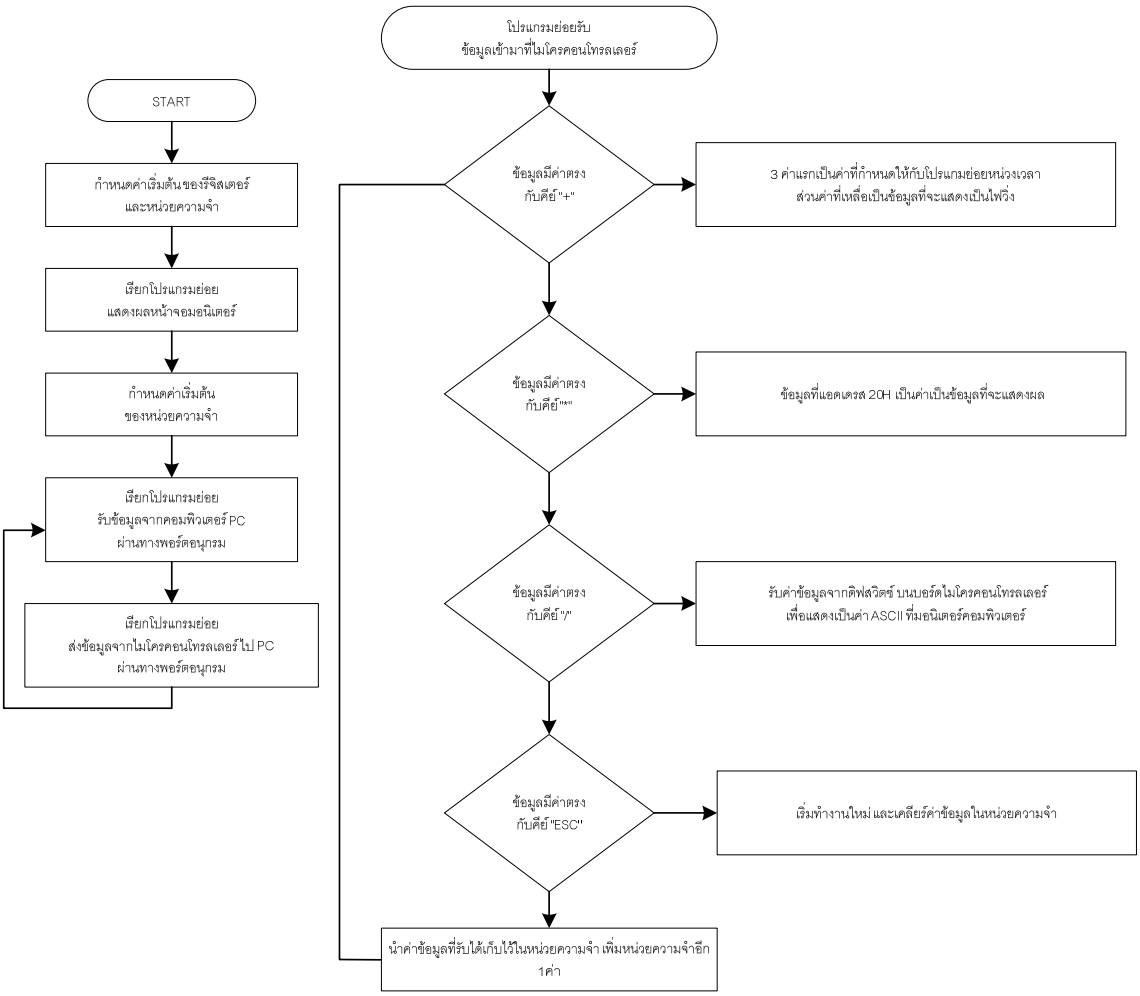


ภาพที่ 8.26 วงจรควบคุมผ่านพอร์ตอนุกรม

C5, R2, SW1 เป็นวงจรรีเซตไอซี MCS-51ส่วน TR1, TR2, R3, R4, R 5 และ R6 ต่อเป็นวงจรปรับระดับสัญญาณให้เหมาะสมกับมาตรฐานของ RS-232 ให้รับส่งข้อมูลแบบอนุกรม กับคอมพิวเตอร์ โดยเชื่อมต่อที่ขา RxD และ TxD ของพอร์ต COM1 หรือ COM2 ส่วนจัมเปอร์ JP1 เลือกในกรณีเชื่อมต่อที่จุด CN1 ใช้งานภายนอก ถ้าหากไม่ต้องการแสดงผลให้เปิดจัมเปอร์ JP1

9.3 การเขียนโปรแกรม

จากการทำงานเขียนได้เป็นผังงาน แสดงดังภาพที่ 8.27 โปรแกรมได้จองเนื้อที่ในการเก็บข้อมูลตำแหน่งหน่วยความจำเริ่มจากตำแหน่งที่ 25H โดย 3 แอดเดรสแรก (25H, 26H, 27H) จะเก็บค่าการหน่วงเวลา โดยรับข้อมูลจากโปรแกรมที่เขียนขึ้นที่โปรแกรมบนวินโดว์ หรือป้อนค่า 3 ตัวแรกจากระบบดอส ตำแหน่งแอดเดรสที่ 4 (28H) จะเริ่มเป็นข้อมูลที่ให้แสดงผล โปรแกรมจะตรวจสอบรหัสของคีย์ที่ส่งมาจากคอมพิวเตอร์ เป็นรหัส ASCII กำหนดให้คีย์ “+” เป็นการส่งข้อมูลในหน่วยความจำตั้งแต่ตำแหน่ง 25H ถึงตำแหน่งสุดท้ายที่เก็บไว้ก่อนกดคีย์ “+” ส่วนคีย์ “*” จะกำหนดให้ส่งข้อมูลเฉพาะแอดเดรส 25H เท่านั้น ส่วนคีย์ “/” จะกำหนดให้ส่งข้อมูลจากอินพุตที่ดิฟสวิตซ์ เป็นรหัส ASCII0 – F ตามคีย์สวิตซ์ และ คีย์ “Esc” จะเป็นการเริ่มต้นในการป้อนข้อมูลใหม่



ภาพที่ 8.27 ผังงานของโปรแกรมควบคุมการรับส่งข้อมูลผ่านทางพอร์ตอนุกรม

```
;************************************************
;****    Computer PC & MCS51 Control RS232 port        ***
;************************************************
        CR      EQU     0DH             ; Carriage return
        LF      EQU     0AH             ; Line feed
                ORG     25H             ; เริ่มตำแหน่งแอดเดรส 20H ในหน่วยความจำข้อมูล
DATA:           DS      50H             ; จองเนื้อที่สำหรับเก็บข้อมูลไว้ 50H ตำแหน่ง
                ORG     0000H
NEW_START:      MOV     P1,#0FFH        ; เคลียร์ค่าข้อมูลในพอร์ต  P1 = 00000000B
                MOV     SP,#07H         ; ให้ตัวชี้สแตกเริ่มที่แอดเดรส 07H
                MOV     21H,#00H        ; เคลียร์ค่าข้อมูลในแอดเดรสที่ 21H
                MOV     22H,#00H        ; เคลียร์ค่าข้อมูลในแอดเดรสที่ 22H
START:          ACALL   INIT            ; ทำการเซตค่าเริ่มต้นของรีจิสเตอร์ที่ติดต่อกับพอร์ตอนุกรม
                MOV     DPTR,#SHOW      ; กำหนดค่าในรีจิสเตอร์ DPTR = แอดเดรสของเลเบล SHOW
                ACALL   DISPLAY         ; เรียกโปรแกรมย่อยแสดงผลติดต่อที่หน้าจอ
                MOV     R1,#25H         ; รีจิสเตอร์ R1= 25H เป็นแอดเดรสเริ่มต้นหน่วยความจำข้อมูล
MAIN:           ACALL   RECEIVE         ; เรียกโปรแกรมรับข้อมูลจาก พอร์ต RS232
                ACALL   SEND            ; ส่งข้อมูลที่รับได้ไปแสดงที่หน้าจอคอมพิวเตอร์ด้วย
                SJMP    MAIN            ; วนกลับไปรับข้อมูลต่อไปอีก
;************ กำหนดค่าต่างๆ ในการติดต่อกับพอร์ตอนุกรม ************
INIT:           MOV     TMOD, #00100000B    , ใช้งานพอร์ตอนุกรมในโหมด 1
                MOV     TL1, #0FDH          ; กำหนดค่าการนับในรีจิสเตอร์ TL1
                MOV     TH1, #0FDH          ; กำหนดค่าเริ่มต้นการนับในรีจิสเตอร์ TH1
                MOV     SCON, #01010000B
                SETB    TR1                 ; เริ่มต้นการทำงาน
                RET                         ; ออกจากโปรแกรมย่อย
;***      โปรแกรมย่อยส่งค่าข้อมูลจาก MCS51 ไปที่ PC ผ่านทาง RS232 ***
SEND:           MOV     SBUF,A          ; นำค่าในรีจิสเตอร์ A ไปเก็บไว้ที่รีจิสเตอร์ SBUF
                JNB     TI,$            ; ตรวจสอบที่บิต TI ว่าได้ส่งข้อมูลใน SBUF หมดแล้วหรือไม่
                CLR     TI              ; เคลียร์บิต TI เมื่อส่งข้อมูลใน SBUF หมดแล้ว
                RET                     ; ออกจากโปรแกรมย่อย
;***   โปรแกรมย่อยรับค่าข้อมูลจาก PC ผ่านทาง RS232 เข้ามาที่ MCS51      ***
```

```
RECEIVE:    JNB   RI,$              ; ตรวจสอบ RI ว่าได้รับข้อมูลเก็บไว้ใน SBUF หมดแล้วหรือไม่
            CLR   RI                ; เคลียร์บิต RI เมื่อรับข้อมูลเก็บใน SBUF หมด
            MOV   A, SBUF           ; นำค่าในรีจิสเตอร์ SBUF ไปเก็บไว้ที่รีจิสเตอร์ A
            MOV   @R1,A             ; ข้อมูลรีจิสเตอร์ A เก็บในตำแหน่งแอดเดรสที่ชี้โดย R1
            CPL   A                 ; กลับค่าข้อมูลเพราะแอลอีดีให้ขาแคโถดต่อกับพอร์ต P1
            MOV   P1,A              ; นำข้อมูลรับได้อยู่ในรีจิสเตอร์ A ออกแสดงผลที่พอร์ต P1
            CPL   A
            CJNE  A,#0DH,END_RECE   ; เปรียบเทียบค่าที่รับเข้า คีย์ "ENTER"
            SJMP  CHEAK_KEY         ; เรียกโปรแกรมย่อยตรวจสอบค่ารหัส ASCII
END_RECE:   INC   R1                ; เพิ่ม R1 ขึ้น1ค่า (หน่วยความจำข้อมูลในลำดับถัดไป)
            RET                     ; ออกจากโปรแกรมย่อย
;***  โปรแกรมย่อยหลังจากรับค่ารหัส ASCII ของคีย์ ENTER      ***
CHEAK_KEY:  DEC   R1                ; ลดค่าข้อมูลใน R1 ในขณะนั้นลง 1 ค่า R1= R1-1
            MOV   A,@R1
            CJNE  A,#2BH,MUL_KEY    ; เปรียบเทียบกับรหัส ASCII ของคีย์ "+"
            MOV   R1,#28H           ;ถ้าเป็นคีย์ "+" ให้ R1 มีค่าข้อมูล 28H
            SJMP  OUT_DATA
MUL_KEY:    CJNE  A,#2AH,DIV_KEY    ;นำไปเปรียบเทียบกับรหัส ASCII ของคีย์ " * "
            SJMP  MUL               ; ถ้ามีค่าเท่ากับคีย์ " * " ให้กระโดดไปที่เลเบล MUL
DIV_KEY:    CJNE  A,#2FH,ESC_KEY    ; นำไปเปรียบเทียบกับรหัส ASCII ของคีย์ " / "
            SJMP  IN_PORT           ; ถ้ามีค่าเท่ากับคีย์ "/" ให้กระโดดไปที่เลเบล IN_PORT
ESC_KEY:    CJNE  A,#1BH,END_CHEAK  ; นำไปเปรียบเทียบกับรหัส ASCII ของคีย์ " ESC "
            ACALL CLEAR             ; เรียกโปรแกรมย่อยเคลียร์ค่าข้อมูล
            AJMP  NEW_START         ; กระโดดไปเริ่มต้นโปรแกรมใหม่ที่ NEW_START
END_CHEAK:  INC   R1                ; เพิ่มค่าข้อมูลใน R1 ในขณะนั้นขึ้น 1 ค่า R1= R1+1
            SJMP  END_RECE          ; กระโดดไปทำที่เลเบล END_RECE
;***  โปรแกรมแสดงผลหลังจากรับค่ารหัส ASCII ของคีย์ "+"      ***
OUT_DATA:   MOV   A,@R1             ; นำข้อมูลที่ชี้ตำแหน่งโดย R1 ไปไว้ที่รีจิสเตอร์ A
            CJNE  A,#2BH,ADD        ; ถ้าข้อมูลมีค่าเท่ากับ "+" ให้ไปที่เลเบล  ADD
            MOV   R1,#28H           ; กำหนดแอดเดรสเริ่มต้นใหม่
            SJMP  OUT_DATA          ; ให้วนทำที่เลเบล OUT_DATA ใหม่
ADD:        MOV   P1,A              ; นำข้อมูลในรีจิสเตอร์ A ออกที่พอร์ต P1
```

```
                ACALL DELAY         ; เรียกโปรแกรมย่อยหน่วงเวลา
                INC   R1            ; เพิ่มค่าข้อมูลในรีจิสเตอร์ R1 = R1+1
                ACALL CHEAK         ; ไปโปรแกรมย่อยตรวจสอบข้อมูล
                SJMP  OUT_DATA      ; กลับไปที่เลเบล OUT_DATA ใหม่
;*******        โปรแกรมย่อยตรวจสอบคีย์ ESC เพื่อเริ่มต้นโปรแกรมใหม่  ********
CHEAK:          PUSH  ACC           ; เก็บค่า ACC ไว้ในสแตก
                MOV   A, SBUF       ; นำค่าในรีจิสเตอร์ SBUF ไปเก็บไว้ที่รีจิสเตอร์ A
                CLR   RI            ; เคลียร์บิต RI เมื่อรับข้อมูลเก็บใน SBUF หมดแล้ว
                CJNE  A,#1BH,RET_CHEAK  ;นำไปเปรียบเทียบกับรหัส ASCII ของคีย์ " ESC "
                ACALL CLEAR
                AJMP  NEW_START     ; กระโดดไปเริ่มต้นโปรแกรมใหม่
RET_CHEAK:      POP   ACC           ; คืนค่าข้อมูลของรีจิสเตอร์ A จากสแตก
                RET                 ; ออกจากโปรแกรมย่อย
;***      โปรแกรมย่อยเคลียร์ค่าข้อมูลในหน่วยความจำข้อมูล        ***
CLEAR:          MOV   R5,#50H       ; กำหนดค่าจำนวนตำแหน่งที่จองไว้
                MOV   R0,#25H       ; กำหนดค่าให้กับรีจิสเตอร์ R0 =25H แอดเดรสเริ่มต้น
CLEAR_1:        MOV   @R0,#00H      ; ให้ตำแหน่งที่ถูกชี้โดยข้อมูลในรีจิสเตอร์ R0 =00H
                INC   R0            ; เพิ่มค่าในรีจิสเตอร์ R0
                DJNZ  R5, CLEAR_1   ; ลดค่าจนข้อมูลครบทั้ง 50H ตำแหน่ง
                RET                 ; ออกจากโปรแกรมย่อย
;***  โปรแกรมแสดงผลควบคุมแบบสวิตซ์หลังจากรับค่ารหัส ASCII ของคีย์ "*"      ***
MUL:            DEC   R1            ; ลดค่าข้อมูลในรีจิสเตอร์ R1 = R1 - 1
                MOV   20H, @R1
MUL_1:          MOV   P1, 20H       ; นำค่าข้อมูลในตำแหน่ง 20H ออกที่พอร์ต P1
                ACALL CHEAK         ; เรียกโปรแกรมย่อยตรวจสอบคีย์ “ESC”
                SJMP  MUL_1         ; วนกลับไปแสดงผลข้อมูลเดิมที่พอร์ต P1
;***  โปรแกรมแสดงผลควบคุมแบบสวิตซ์หลังจากรับค่ารหัส ASCII ของคีย์ "/ "      ***
IN_PORT:        MOV   21H,P3        ; นำค่าข้อมูลที่ P3 เก็บไว้ที่หน่วยความจำข้อมูล 21H
IN_PORT1:       CLR   22H.6         ; กำหนดให้บิตที่ 6 ของแอดเดรส 22H เป็น "0"
                CLR   22H.0         ; กำหนดให้บิตที่ 0 ของแอดเดรส 22H เป็น "0"
                CLR   22H.1         ; กำหนดให้บิตที่ 1 ของแอดเดรส 22H เป็น "0"
                CLR   22H.2         ; กำหนดให้บิตที่ 2 ของแอดเดรส 22H เป็น "0"
```

```
                MOV   C,21H.3          ; กำหนดค่าบิตให้แฟลกทด C <21.3H
                MOV   22H.0,C          ; กำหนดค่าบิตให้แฟลกทด C >22.3H
                MOV   C,21H.4          ; กำหนดค่าบิตให้แฟลกทด C <21.4H
                MOV   22H.1,C          ; กำหนดค่าบิตให้แฟลกทด C >22.4H
                MOV   C,21H.5          ; กำหนดค่าบิตให้แฟลกทด C <21.5H
                MOV   22H.2,C          ; กำหนดค่าบิตให้แฟลกทด C >22.5H
                MOV   C,21H.7          ; กำหนดค่าบิตให้แฟลกทด C <21.7H
                MOV   22H.3,C          ; กำหนดค่าบิตให้แฟลกทด C >22.7H
                MOV   A,22H            ; นำค่าข้อมูลในตำแหน่ง 22H ไปเก็บไว้ที่รีจิสเตอร์ A
;***  โปรแกรมแสดงผลควบคุมแบบสวิตซ์หลังจากรับค่ารหัส ASCII ของคีย์ ตั้งแต่ A-F  ***
                CJNE  A,#0AH,KEY_B     ; เปรียบเทียบกับคีย์ A ดิฟสวิตซ์ 00001010
                MOV   A,#41H           ; ให้รีจิสเตอร์ A มีค่าคงที่ #41H (ASCIIA)
                SJMP  KEY_END
KEY_B:          CJNE  A,#0BH,KEY_C     ; เปรียบเทียบกับคีย์ B ดิฟสวิตซ์ 00001100
                MOV   A,#42H           ;ให้รีจิสเตอร์ A มีค่าคงที่ #42H (ASCII B)
                SJMP  KEY_END
KEY_C:          CJNE  A,#0CH,KEY_D     ; เปรียบเทียบกับคีย์ C ดิฟสวิตซ์ 00001101
                MOV   A,#43H           ;ให้รีจิสเตอร์ A มีค่าคงที่ #43H (ASCII C)
                SJMP  KEY_END
KEY_D:          CJNE  A,#0DH,KEY_E     ; เปรียบเทียบกับคีย์ D ดิฟสวิตซ์ 00001101
                MOV   A,#44H           ;ให้รีจิสเตอร์ A มีค่าคงที่ #44H (ASCIID)
                SJMP  KEY_END
KEY_E:          CJNE  A,#0EH,KEY_F     ; เปรียบเทียบกับคีย์ A ดิฟสวิตซ์ 00001110
                MOV   A,#45H           ;ให้รีจิสเตอร์ A มีค่าคงที่ #45H (ASCII E)
                SJMP  KEY_END
KEY_F:          CJNE  A,#0FH,KEY_ASCII ; เปรียบเทียบกับคีย์ A ดิฟสวิตซ์ 00001111
                MOV   A,#46H           ;ให้รีจิสเตอร์ A มีค่าคงที่ #46H (ASCII F)
                SJMP  KEY_END
KEY_ASCII:      ADD   A,#30H           ; นำค่าที่ได้บวกกับค่า 30H ให้เป็นรหัส ASCII
KEY_END:        CPL   A
                MOV   P1,A             ; นำค่าในแอดเดรส 22H ออกพอร์ต P1
                CPL   A
```

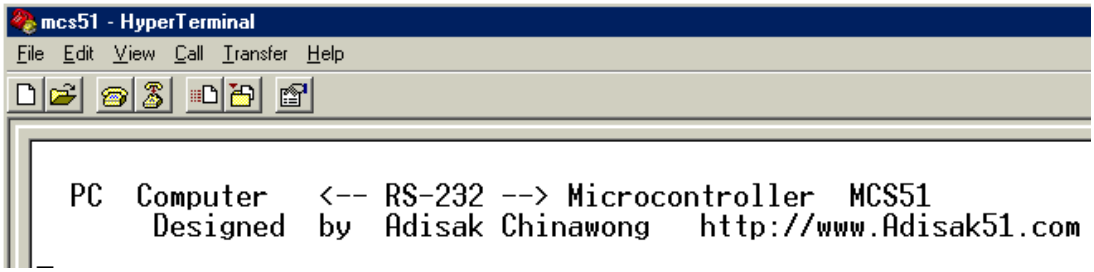
```
                ACALL SEND              ; เรียกโปรแกรมย่อยส่งข้อมูลทางพอร์ตอนุกรมไป PC
                ACALL RECEIVE           ; เรียกโปรแกรมย่อยรับค่าข้อมูลทางพอร์ตอนุกรม
                MOV   A, SBUF           ; นำค่าในรีจิสเตอร์ SBUF ไปเก็บไว้ที่รีจิสเตอร์ A
                CLR   RI                ; เคลียร์บิต RI เมื่อรับข้อมูลเก็บใน SBUF หมด
                CJNE  A,#1BH,DATA_IN    ; นำไปเปรียบเทียบกับรหัส ASCII ของคีย์ " ESC "
                ACALL CLEAR             ; เรียกโปรแกรมย่อยเคลียร์ค่าข้อมูล
                AJMP  NEW_START         ; กระโดดกลับไปเริ่มต้นโปรแกรมใหม่
DATA_IN:        SJMP  IN_PORT           ; กระโดดไปรับค่าข้อมูลใหม่
;*******        โปรแกรมย่อยแสดงผลติดต่อที่หน้าจอ      ********
DISPLAY:        MOV   A,#00             ; เคลียร์ค่าข้อมูลในรีจิสเตอร์ A = 00000000
                MOVC  A,@A+DPTR         ; กำหนดข้อมูลในรีจิสเตอร์ DPTR +A
                ACALL SEND              ; เรียกโปรแกรมย่อยส่งข้อมูลทางพอร์ตอนุกรมไป PC
                JZ    END_SHOW  ; ข้อมูลรีจิสเตอร์ A มีค่าเป็น 00 หรือไม่ ตรวจสอบที่แฟลกศูนย์
                INC   DPTR      ; เพิ่มค่าใน DPTR ชี้แอดเดรสของตัวอักษร ASCII ตัวต่อไป
                SJMP  DISPLAY   ; วนกลับไปทำที่เลเบล DISPLAY
END_SHOW:       RET             ; ออกจากโปรแกรมย่อย
;*******        โปรแกรมย่อยหน่วงเวลา********
DELAY:          MOV   R4, 25H
DELAY2:         MOV   R2, 26H
DELAY1:         MOV   R3, 27H
                DJNZ  R3, $
                DJNZ  R2, DELAY1
                DJNZ  R4, DELAY2
                RET
        ;******* ข้อมูลรหัส ASCII ที่จะแสดงผล ในขณะเริ่มติดต่อกับ PC ครั้งแรก ********
SHOW:           DB    CR, LF, “PC Computer   <-- RS-232 --> Microcontroller MCS51”, CR,LF
                DB    “Designed by Adisak Chinawong   http://www.Adisak51.com”, CR, LF
                DB    00H
                END
```

9.4 การทดสอบและการใช้งาน

9.4.1 เสียบสาย DB9 ที่จะติดต่อกับคอมพิวเตอร์ที่พอร์ต Com1 หรือ พอร์ต Com2 เข้ากับวงจร

9.4.2 เปิดโปรแกรม Hyper Terminal

9.4.3 หลังจากกดสวิตซ์รีเซตที่ไอซี MCS-51 จะปรากฏข้อความแสดงดังภาพที่ 8.28



ภาพที่ 8.28 แสดงข้อความเมื่อมีการติดต่อระหว่างไอซี MCS-51 กับไมโครคอมพิวเตอร์ PC ได้สำเร็จ

รูปแบบที่ 1 (ไฟวิ่ง) เลือกคีย์ **1 2 3 4 5 6 7 8 9 A B C B D … "+" (2BH) "Enter" (0DH)**

ผลที่เกิดขึ้น

1) ขณะที่กดคีย์สวิตซ์ใดๆบนคีย์บอร์ดคอมพิวเตอร์ แอลอีดี LED1 – LED9 จะแสดงผลเป็นเลข ASCII ตามรหัสของค่าคีย์นั้น ๆ

2) เมื่อกดที่เครื่องหมาย + แล้วตามด้วยคีย์ Enter แอลอีดีจะกระพริบตามค่าที่ **4 5 6 7 8 9 A B C D …** โดยแสดงเป็นรหัสASCII แสดงวนรอบไปตลอด

3) การกำหนดความเร็วในการกระพิมพ์ขึ้นอยู่กับ 3 ค่าแรก ให้ทดลองเปลี่ยน 3 ค่าแรก แล้วทดลองใหม่อีกครั้งสังเกตการกระพริบของแอลอีดี

4) ให้กดที่คีย์ Esc **(1BH)** หน้าจอมอนิเตอร์จะแสดงค่าเริ่มต้นใหม่

รูปแบบที่ 2 (ปิด - เปิด)เลือกที่คีย์ใดๆได้เพียงคีย์เดียว แล้วกด **"*" (2AH) "Enter" (0DH)**

ผลที่เกิดขึ้น

1) ขณะที่กดปุ่มใดๆ แอลอีดี LED1 – LED9 จะแสดงผลเป็นรหัส ASCII ของค่าคีย์นั้น

2) เมื่อกดที่คีย์ " * " แล้วตามด้วยคีย์ Enter แอลอีดีจะแสดงผลข้อมูลที่ 20H ตำแหน่งเดียว

3) ให้กดที่คีย์ Esc **(1BH)** หน้าจอมอนิเตอร์จะแสดงค่าเริ่มต้นใหม่

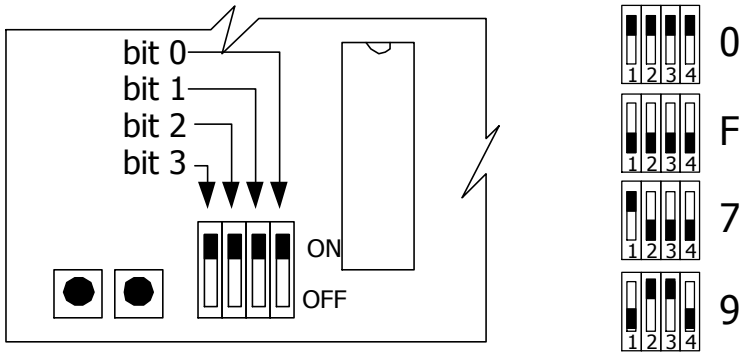
รูปแบบที่ 3 (อินพุต) เปลี่ยนค่าที่ดิฟสวิตซ์ การกำหนดตำแหน่งแสดงดังภาพที่ 8.29 จากนั้นให้กดคีย์ "/" **(2FH)** บนคีย์บอร์ดคอมพิวเตอร์ แล้วกดคีย์ **"Enter" (0DH)** ทุกครั้งที่เปลี่ยนค่าของดิฟสวิตซ์

ผลที่เกิดขึ้น

1) มอนิเตอร์แสดงผลเป็นเลข 0 1 2 3 4 5 6 7 8 9 A B C D E F ตามค่าของดิฟสวิตซ์ ที่เลือก (ONเป็นสถานะลอจิก “0”) และแอลอีดี LED1 – LED9 จะแสดงผลเป็นเลขASCII(ASCII) ตามรหัสของค่าคีย์นั้น

2) การเปลี่ยนค่าข้อมูลที่ดิฟสวิตซ์ เพื่อรับข้อมูลเข้ามาใหม่ สู่เครื่องคอมพิวเตอร์ PC ต้องกดที่คีย์ “ / “ **(2FH)** บนคีย์บอร์ดคอมพิวเตอร์ ทุกครั้งที่เปลี่ยนค่าของดิฟสวิตซ์

3) ให้กดที่คีย์ Esc **(1BH)** หน้าจอมอนิเตอร์จะแสดงค่าเริ่มต้นใหม่



ภาพที่ 8.29 แสดงตำแหน่งของดิฟสวิตซ์ และตัวอย่างการกำหนดตำแหน่งของสวิตซ์

9.5 การตรวจสอบและแก้ไข

9.5.1 ถอดไอซี AT89C2051 ยังไม่ต่อสายอินเตอร์เฟสกับพอร์ตอนุกรม

9.5.2 ให้ใช้สายไฟขนาดเล็ก (สายโทรศัพท์) เสียบที่ขา 10 (Ground) ของซ็อกเกตไอซี ให้นำปลายสายอีกข้างหนึ่งแตะกับขาที่ 12 (P1.0) ถึงขาที่ 19 (P1.7) ครบทุกขา LED 1 - LED9 จะต้องสว่าง ถ้าไม่สว่างให้ตรวจความถูกต้องของขั้ว LED

9.5.3 ใช้สายไฟขนาดเล็กเสียบที่ขา 12 (P1.0) ของซ็อกเกตไอซี นำปลายสายอีกข้างหนึ่งไปเสียบกับขาที่ 7 (P3.2) ของซ็อกเกต หลังจากนั้นให้เปลี่ยนสถานะที่ดิฟสวิตซ์แต่ละตัวที่ต่อกับขาพอร์ตนั้นๆ สังเกตที่ LED2 จะต้องดับเมื่อดิฟสวิตซ์เปิดวงจร “1” และสว่างเมื่อเป็นดิฟสวิตซ์ปิดวงจรสถานะ ”0” หลังจากนั้นให้ทดลองขาที่ 8 (P3.4) ขาที่ 9 (P3.5) และ ขาที่ 11 (P3.7) ตามลำดับ

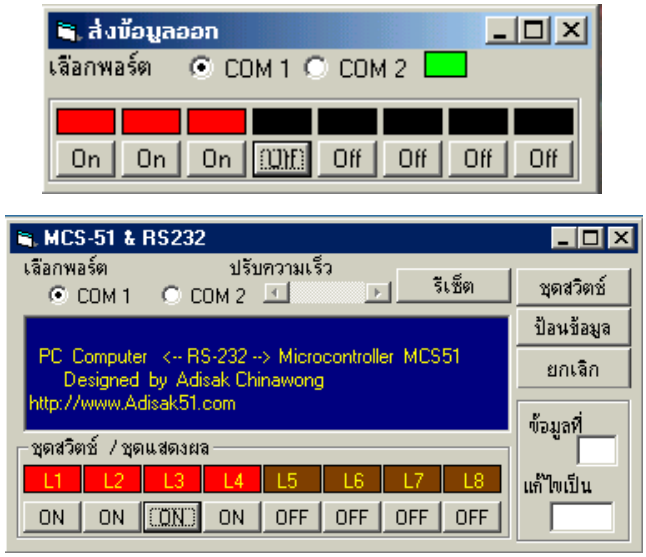
9.5.4 การทดลองในส่วนของวงจรรีเซต ให้ต่อสายระหว่างขาที่ 1 กับขาที่ 12 เมื่อทำการกดสวิตซ์ รีเซตทุกครั้งทำให้ LED 2 ดับ และเมื่อปล่อยที่คีย์สวิตซ์รีเซตทำให้ LED1 สว่าง

9.5.5 ขั้นตอนสุดท้ายให้ใส่ IC2 ลงบนซ็อกเกต แล้วต่อสายอินเตอร์เฟสระหว่างคอมพิวเตอร์กับวงจรไอซี MCS-51 หลังจากนั้นให้เปิดโปรแกรมตามขั้นตอนทดสอบการทำงานต้องได้การแสดงผลที่จอมอนิเตอร์ถ้าไม่ได้ให้ตรวจสอบการต่อสายกับคอนเน็กเตอร์ DB9 และทรานซิสเตอร์ในวงจร

9.6 ตัวอย่างการควบคุม โดยใช้โปรแกรมวิชวลเบสิก

การออกแบบโดยใช้โปรแกรมวิชวลเบสิก(Visual Basic) แสดงดังภาพที่ 8.30 ออกเป็นไฟวิ่งหลายรูปแบบ สามารถ นำไปดัดแปลงเพื่อควบคุมในการเปิดปิดอุปกรณ์ไฟฟ้า หรือหน้าจอคอมพิวเตอร์ตามจุดต่างๆได้อย่างอิสระ โดยการนำเอาค่า ASCII ของแต่ละคีย์มาเป็นเงื่อนไข ในการเขียนโปรแกรมได้

คีย์	รหัส ASCII
“ + ”	2BH
“ Enter ”	0DH
“ Esc ”	1BH
“ * ”	2AH
“ / ”	2FH



ภาพที่ 8.30 ตัวอย่างโปรแกรมควบคุมโดยใช้วิชวลเบสิก

วงจรสำเร็จแสดงดังภาพที่ 8.31 ขั้นตอนการสร้างสามารถค้นคว้าวิธีการสร้างเพิ่มเติมได้ที่ http://www.adisak51/project04.html

ภาพที่ 8.31 แสดงบอร์ดสำเร็จควบคุมผ่านพอร์ตอนุกรม

สรุป

การสื่อสารข้อมูลแบบอนุกรมสามารถลดจำนวนของสายสัญญาณโดยใช้เพียง สายส่ง (TxD) 1 เส้น สายรับ (RxD) 1 เส้น และสายกราวด์ (Ground) 1 เส้น ข้อมูลถูกส่งไปตามสายสัญญาณทีละบิตตามจังหวะเวลาที่กำหนด โดยจังหวะเวลาต้องมีมาตรฐานของฝ่ายส่ง และฝ่ายรับ การรับสัญญาณที่ส่งทีละบิต ต้องมีอัตราความเร็วระหว่างการรับ และการส่งมีค่าเท่ากัน โดยทั่วไปการระบุความเร็วของจำนวนบิตในการรับและส่งข้อมูลใน 1 วินาที โดยเรียกความเร็วในการส่งข้อมูลว่า อัตราบอด (Baud Rate) ซึ่งมีหน่วยเป็นบิตต่อวินาที เช่น 300, 1,200, 2,400, 4,800 และ 9,600 บิตต่อวินาที การสื่อสารแบบอนุกรม แบ่งประเภทของการสื่อสารตามลักษณะสัญญาณ ในการสื่อสารได้ 2 แบบ แบบแรกเป็นการสื่อสารแบบซิงโครนัส (Synchronous) แบบที่สองเป็นการสื่อสารแบบอะซิงโครนัส (Asynchronous) ส่วนที่ใช้ในการควบคุมการรับส่งข้อมูลที่เรียกว่า UART (Universal Asynchronous Receiver Transmitters)

วงจรภายในของไอซี MCS-51 ส่วนติดต่อสื่อสารอนุกรมแบบ UART และมีโครงสร้างแบบฟลูดูเพล็ก (Full Duplex) โดยสามารถรับ และส่งข้อมูลได้ในเวลาเดียวกันได้ โดยรีจิสเตอร์ใช้งานติดต่อสื่อสารทางพอร์ตอนุกรม ประกอบด้วย รีจิสเตอร์ SCON ทำหน้าที่ควบคุมการเลือกโหมดการทำงาน รีจิสเตอร์ SBUF ใช้เก็บข้อมูลการรับหรือการส่ง และรีจิสเตอร์ PCON ทำหน้าที่ กำหนดอัตรารับส่ง

การสื่อสารอนุกรมของไอซี MCS-51แบ่งออกได้เป็น 4 โหมด โหมด 0 เป็นการรับส่งข้อมูลขนาด 8 บิต การส่งข้อมูลเลื่อนออกไปทีละบิตโดยใช้ RxD เพียงขาเดียว และไม่มีการส่งบิตเริ่มต้นส่วนTxD แต่ในโหมด 0 มัก ไม่นิยมนำมาใช้งาน เพราะไอซี MCS-51ในปัจจุบัน มีจำนวนพอร์ตที่มากพอ โหมด 1 เป็นการรับและส่งข้อมูลขนาด 10 บิตแบบ UART สามารถติดต่อสื่อสารอนุกรมกับมาตรฐาน RS-232C ของไมโครคอมพิวเตอร์ได้ โหมด 2 เป็นการรับและส่งข้อมูลขนาด 11 บิตแบบ UART ข้อมูลแบบอนุกรมถูกรับเข้ามาทางขา RXD และส่งข้อมูลออกไปทางขา TXD โหมด 3 เป็นการรับส่งข้อมูลแบบ 11 บิต แบบ UART เหมือนกับโหมด 2 แต่ในโหมด 3 สามารถกำหนดอัตราความเร็วในการรับ และส่งข้อมูลได้ตามต้องการ

อัตรารับและส่งข้อมูล (Baud Rate Generator) โหมด 0 ไม่สามารถกำหนดอัตรารับส่งเองได้ แต่มีค่าเท่ากับความถี่สัญญาณนาฬิกาของไอซี MCS-51 หารด้วย 12 โหมด 2 อัตรารับส่งในโหมดนี้เลือกได้ 2 อัตราความเร็วในการรับส่งข้อมูล หาได้จากความถี่สัญญาณนาฬิกาของ ไอซี MCS-51 หารด้วย 32 หรือหารด้วย 64 เรียกว่า SMOD 0 และ SMOD 1 ซึ่งขึ้นอยู่กับการกำหนดค่าสถานะของบิต SMOD ที่อยู่ในรีจิสเตอร์ PCON เป็นตัวเลือก โหมด 1 และโหมด 3 มีอัตราการรับส่งข้อมูลโดยถูกกำหนดได้ตามต้องการโดยใช้อัตราการเกิดค่านับเกินของไทม์เมอร์1 หรือ ไทม์เมอร์ 2 เป็นตัวกำหนด

การสื่อสารข้อมูลระบบมัลติโฟรเซสเซอร์ (Multiprocessors System) เป็นวิธีการนำไอซี MCS-51 จำนวนหลายๆ ตัว มาเชื่อมต่อเพื่อสื่อสารสารข้อมูลกันนั้นต้องมีตัวหลักอยู่ 1 ตัว (Master) ส่วนตัวอื่นๆ

เป็นตัวลูก (Slave) และสามารถนำตัวลูกมาต่อร่วมได้ถึง 256 ตัวแต่ต้องมีการกำหนดค่าแอดเดรสของไอซีในแต่ละตัว เพื่อให้ไอซีคอนโทรลเลอร์ตัวหลักสามารถระบุการติดต่อกับตัวลูกตัวนั้นๆได้

มาตรฐานอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรม ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง จุดประสงค์เพื่อให้มีการใช้งานในการเชื่อมต่อที่สอดคล้องกัน ระหว่างอุปกรณ์คอมพิวเตอร์ต่างๆ การรับส่งสัญญาณกำหนดความยาวสูงสุดไว้ที่ไม่เกิน 50 ฟุต โดยมีระดับสัญญาณตั้งแต่ 3 โวลต์ ถึง 15 โวลต์ สำหรับลอจิก “0” และมีระดับแรงดันที่ -3 โวลต์ ถึง -15 โวลต์ สำหรับลอจิก “1”