หน่วยที่ 9 อินเตอร์รัปต์

อดิศักดิ์ ชิณะวงศ์

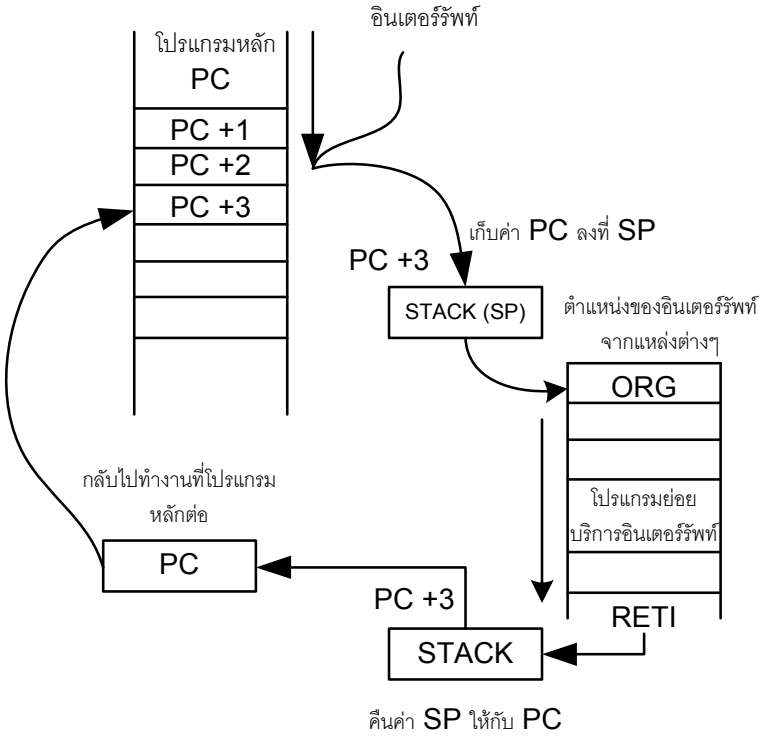
เอกสารประกอบการเรียนวิชาไมโครคอนโทรลเลอร์

เผยแพร่ที่ www.Adisak51.com

1. การอินเตอร์รัปต์

วิธีเขียนโปรแกรมแบบโพลลิ่ง (Polling) ทำให้ไอซี MCS-51 ปฏิบัติตามโปรแกรมคำสั่งโดยให้ตรวจสอบสัญญาณการเชื่อมต่อจากอุปกรณ์ภายนอก หรือสัญญาณที่เกิดขึ้นภายในตัวไอซีอยู่ตลอดเวลา และแม้ว่าไม่มีสัญญาณใด ๆ เกิดขึ้นเลย ดังนั้นคุณสมบัติภายในไอซีจึงมีส่วนใช้งานที่เรียกว่าอินเตอร์รัปต์ (Interrupt) ซึ่งหมายถึงการขัดจังหวะ โดยไอซีไม่ต้องตรวจสอบสัญญาณที่เกิดขึ้นตลอดเวลา แต่จะทำการตรวจสอบเมื่อมีสัญญาณการขัดจังหวะเข้ามา

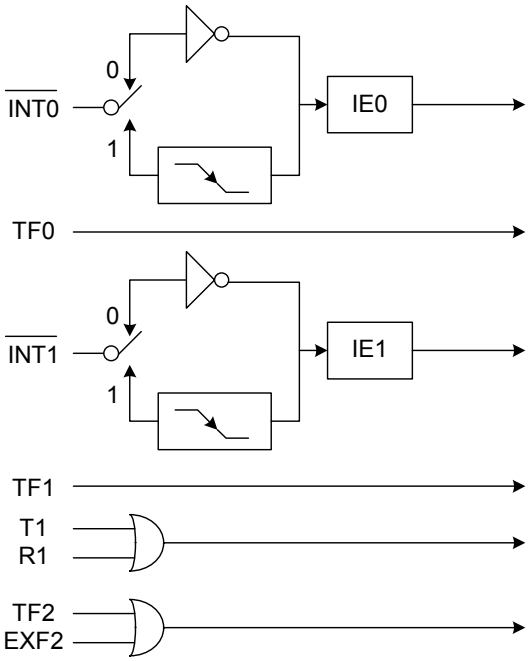
วิธีการอินเตอร์รัปต์ ใช้งานเมื่อไอซี MCS-51 กำลังปฏิบัติตามคำสั่งในโปรแกรมหลักอยู่นั้น หากมีสัญญาณการร้องขออินเตอร์รัปต์ ที่ส่งมาจากอุปกรณ์ภายนอก หรือสัญญาณจากภายในของตัว ไอซี ๆ จะหยุดการปฏิบัติตามคำสั่งของโปรแกรมหลัก และจัดเก็บตำแหน่งค่าของโปรแกรมเคาน์เตอร์ ในขณะนั้น (PC +1) ไว้ที่สแตกพอรย์เตอร์ (SP) หลังจากนั้นจึงเริ่มปฏิบัติตามโปรแกรมย่อยบริการอินเตอร์รัปต์ที่ถูกกำหนดตำแหน่งไว้ ของแต่ละแหล่ง (Interrupt Vector) เมื่อปฏิบัติตามโปรแกรมคำสั่งอินเตอร์รัปต์จนเสร็จสิ้นแล้ว ต้องใช้คำสั่ง RETI เพื่อคืนค่าในสแตกพอรย์เตอร์ ให้กับโปรแกรมเคาน์เตอร์ ไอซีจึงกลับไปทำคำสั่งที่โปรแกรมหลักต่ออีกครั้งหนึ่ง ขั้นตอนดังกล่าวแสดงดังภาพที่ 9.1



ภาพที่ 9.1 ขั้นตอนในการเกิดอินเตอร์รัปต์

2. แหล่งสัญญาณการอินเตอร์รัปต์

ไอซี MCS-51 มีแหล่งที่ทำให้เกิดสัญญาณการอินเตอร์รัปต์ จากหลายแหล่งด้วยกันทั้งภายนอก และภายในแสดงดังภาพที่ 9.2



ภาพที่ 9.2 แหล่งกำเนิดสัญญาณการอินเตอร์รัปต์ของไอซี MCS-51

การอินเตอร์รัปต์ในแต่ละแหล่ง มีการกำหนดตำแหน่งแอดเดรสของหน่วยความจำโปรแกรมสำหรับเขียนคำสั่งการบริการอินเตอร์รัปต์ แสดงดังในตารางที่ 9.1

ตารางที่ 9.1 แสดงตำแหน่งแอดเดรสเริ่มต้นของโปรแกรมบริการอินเตอร์รัปต์ในแต่ละแหล่ง

Interrupt Source	Vector Address	สัญญาณอินเตอร์รัปต์
IE0	0003H	สัญญาณอินเตอร์รัปต์ภายนอก 0
TF0	000BH	สัญญาณอินเตอร์รัปต์ไทเมอร์ 0
IE1	0013H	สัญญาณอินเตอร์รัปต์ภายนอก 1
TF1	001BH	สัญญาณอินเตอร์รัปต์ไทเมอร์ 1
RI&TI	0023H	สัญญาณอินเตอร์รัปต์พอร์ตอนุกรม
TF2&EXF2	002BH	สัญญาณอินเตอร์รัปต์ไทเมอร์ 2

2.1 สัญญาณอินเตอร์รัปต์ภายนอก (External Interrupt)

เป็นสัญญาณที่มาจากภายนอกไอซี MCS-51 มี 2 ชนิดด้วยกันคือ อินเตอร์รัปต์ภายนอก 0 โดยตรวจสอบสัญญาณที่ขา $\overline{\text{INT0}}$ ของไอซีและอินเตอร์รัปต์ภายนอก 1 ตรวจสอบสัญญาณที่ขา $\overline{\text{INT1}}$ หลังจากมีการอินเตอร์รัปต์ภายนอกเกิดขึ้นแล้ว ต้องตรวจสอบสถานะของสัญญาณให้กลับสู่สภาพเดิมด้วย หากไม่กลับสถานะเดิมทำให้ไอซีเข้าสู่โปรแกรมย่อยบริการอินเตอร์รัปต์อยู่ตลอด โดยไม่กลับไปที่โปรแกรมหลัก

2.2 สัญญาณอินเตอร์รัปต์ไทเมอร์ 0 และไทเมอร์ 1

ประกอบด้วยไทเมอร์ เคาน์เตอร์ จำนวน 2 ตัวคือไทเมอร์ 0 และไทเมอร์ 1 นำไปใช้เป็นการนับค่าหรือตั้งเวลา การใช้งานของไทเมอร์ ทั้ง 2 ตัว เก็บค่าที่กำหนดการนับ หรือตั้งเวลาไว้ในรีจิสเตอร์ TLx และ THx ใช้การนับขึ้นจากค่าข้อมูลในรีจิสเตอร์ TLx และ รีจิสเตอร์ THx มีค่าเป็น “1” ทุกบิต ทำให้บิตแสดงสถานะ TFx แสดงผลการเกิดค่านับเกิน (Overflow) ถูกเซตเป็น “1” ทำให้เกิดการอินเตอร์รัปต์ หลังจากนั้นต้องเคลียร์ TFx โดยซอร์ฟแวร์ เพื่อรับการบริการอินเตอร์รัปต์ในครั้งต่อไป

2.3 สัญญาณอินเตอร์รัปต์พอร์ตอนุกรม (Serial Port Interrupt)

เป็นสัญญาณอินเตอร์รัปต์ที่เกิดจากการสื่อสารแบบอนุกรมเพื่อรับ และส่งข้อมูล โดยตรวจสอบการเกิดอินเตอร์รัปต์ที่บิต TI หรือ RI เมื่อมีการส่ง หรือรับข้อมูลจนครบแล้ว หลังจากนั้นต้องเคลียร์บิต TI และ RI โดยซอร์ฟแวร์

2.4 สัญญาณอินเตอร์รัปต์ไทเมอร์2 (Timer2 Interrupt)

มีอยู่ในเบอร์ 8052, AT89C52, AT89SXX โดยเพิ่ม ไทเมอร์ เคาน์เตอร์อีก 1 ตัว สามารถสร้างสัญญาณการอินเตอร์รัปต์ได้ เหมือนกับไทเมอร์ 0 และไทเมอร์1 อินเตอร์รัปต์ของไทเมอร์2 เกิดขึ้นโดยการนำบิต TF2 และ EXF2 ผ่าน OR GATE โดยบิต TF2 และ EXF2 ไม่ถูกเคลียร์ด้วยฮาร์ดแวร์ ขณะที่โปรแกรมไปทำส่วนบริการอินเตอร์รัปต์ของพอร์ตสื่อสารอนุกรม ดังนั้นโปรแกรมบริการอินเตอร์รัปต์ต้องคอยตรวจสอบสถานะของบิต TF2 และ EXF2 ที่ทำให้เกิดการอินเตอร์รัปต์ขึ้น และต้องเขียนโปรแกรมเพื่อเคลียร์บิต ที่ทำให้เกิดการอินเตอร์รัปต์ด้วย ในส่วนของไทเมอร์2 ยังประกอบไปด้วยการทำงานในโหมดแคปเจอร์ (Capture) หรือตรวจจับสัญญาณที่ขา T2EX มีการเปลี่ยนแปลงของระดับลอจิก “1” เป็น “0” ถ้ามีการเปลี่ยนแปลงเกิดการเซตค่าที่บิต EXF2 เพื่อนำไปสร้างสัญญาณการอินเตอร์รัปต์ต่อไป

3. รีจิสเตอร์ และการกำหนดค่าการทำงาน

3.1 รีจิสเตอร์ IE (Interrupt Enable Register)

เป็นรีจิสเตอร์ใช้งานพิเศษอยู่ตำแหน่งแอดเดรส A8H มีขนาด 8 บิต เป็นรีจิสเตอร์ที่สามารถเข้าถึงในระดับบิตได้ ทำหน้าที่ควบคุมการตอบรับต่อสัญญาณอินเตอร์รัปต์แต่ละชนิดแสดงดังภาพที่ 9.3 ประกอบด้วยบิตต่างๆ ดังนี้

IE.7	IE.6	IE.5	IE.4	IE.3	IE.2	IE.1	IE.0
EA	-	ET2	ES	ET1	EX1	ET0	EX0

ภาพที่ 9.3 รีจิสเตอร์ควบคุมการตอบรับสัญญาณอินเตอร์รัปต์ (Interrupt Enable Register)

บิต EA หรือ IE.7 (Enable/Disable all Interrupt) ทำหน้าที่ตอบรับการอินเตอร์รัปต์ทุกแหล่ง
บิต IE.6 สำรองไว้ใช้ในไอซีเบอร์ใหม่ๆ
บิต ET2 หรือ IE.5 ทำหน้าที่ควบคุมให้ตอบรับการอินเตอร์รัปต์ TF2 เมื่อเกิดโอเวอร์โฟลว์
บิต ES หรือ IE.4 ทำหน้าที่ควบคุมการตอบรับอินเตอร์รัปต์จากพอร์ตสื่อสารอนุกรม
บิต ET1 หรือ IE.3 ทำหน้าที่ควบคุมการตอบรับอินเตอร์รัปต์ TF1 เมื่อเกิดโอเวอร์โฟลว์
บิต EX1 หรือ IE.2 ทำหน้าที่ควบคุมการตอบรับอินเตอร์รัปต์จากภายนอกที่ขา $\overline{\text{INT1}}$
บิต ET0 หรือ IE.1 ทำหน้าที่ควบคุมการตอบรับอินเตอร์รัปต์ TF0 เมื่อเกิดโอเวอร์โฟลว์
บิต EX0 หรือ IE.0 ทำหน้าที่ควบคุมการตอบรับอินเตอร์รัปต์จากภายนอก $\overline{\text{INT0}}$

* ถ้ากำหนดให้บิตเป็นสถานะลอจิก “1” เป็นการตอบรับอินเตอร์รัปต์
* ถ้ากำหนดให้บิตเป็นสถานะลอจิก “0” ไม่ตอบรับการอินเตอร์รัปต์

3.2 รีจิสเตอร์ IP (Interrupt Priority Register)

ทำหน้าที่จัดลำดับความสำคัญของการอินเตอร์รัปต์ มีตำแหน่งแอดเดรสอยู่ที่ B8H มีขนาด 8 บิต เข้าถึงข้อมูลในระดับบิตได้ แสดงดังภาพที่ 9.4 สามารถจัดลำดับความสำคัญของการอินเตอร์รัปต์ได้ 2 ระดับแตกต่างกัน ในกรณีไม่มีการจัดลำดับความสำคัญของการอินเตอร์รัปต์ หรือจัดให้มีความสำคัญในระดับเดียวกัน ไอซี MCS-51 จัดให้มีลำดับความสำคัญของสัญญาณอินเตอร์รัปต์ ตามลำดับความสำคัญจากสูงไปต่ำ เพื่อแก้ปัญหาการขออินเตอร์รัปต์ในระดับเดียวกัน และเกิดขึ้นพร้อมกัน แสดงดังตารางที่ 9.2

ตารางที่ 9.2 แสดงตามลำดับความสำคัญจากต่ำ (EXF2) ไปสูง (IE0)

IE0	External Interrupt 0
TF0	Timer 0
IE1	External Interrupt 1
TF1	Timer 1
RI หรือ TI	Serial Port
TF2 หรือ EXF2	Timer 2

IP.7	IP.6	IP.5	IP.4	IP.3	IP.2	IP.1	IP.0
-	-	PT2	PS	PT1	PX1	PT0	PX0

ภาพที่ 9.4 รีจิสเตอร์กำหนดลำดับความสำคัญของสัญญาณอินเตอร์รัปต์ (Interrupt Priority Register)

PX0 เป็นบิตกำหนดลำดับความสำคัญของสัญญาณอินเตอร์รัปต์ภายนอก 0

PT0 เป็นบิตการกำหนดลำดับความสำคัญของสัญญาณอินเตอร์รัปต์ไทเมอร์ 0

PX1 เป็นบิตกำหนดลำดับความสำคัญของสัญญาณอินเตอร์รัปต์ภายนอก 1

PT1 เป็นบิตกำหนดลำดับความสำคัญของสัญญาณอินเตอร์รัปต์ไทเมอร์ 1

PS เป็นบิตกำหนดลำดับความสำคัญของสัญญาณอินเตอร์รัปต์พอร์ตอนุกรม

PT2 เป็นบิตกำหนดลำดับความสำคัญของสัญญาณอินเตอร์รัปต์ไทเมอร์ 2

ในรีจิสเตอร์ IP บิตใดมีค่าสถานะลอจิกเป็น "1" จะมีลำดับความสำคัญสูง แต่ถ้าไม่มีการจัดลำดับความสำคัญการอินเตอร์รัปต์จาก $\overline{\text{INT0}}$ มีลำดับสูงกว่าการอินเตอร์รัปต์แบบ $\overline{\text{INT1}}$ ซึ่งเป็นค่าเริ่มต้น เมื่อมีการกำหนดลำดับความสำคัญของการอินเตอร์รัปต์แตกต่างกัน หากมีการร้องขออินเตอร์รัปต์ ที่เกิดขึ้นพร้อมกัน 2 แหล่ง มีระดับความสำคัญแตกต่างกัน ไอซี MCS-51 ตอบรับการร้องขอการอินเตอร์รัปต์จากแหล่งกำเนิดสัญญาณอินเตอร์รัปต์ที่มีระดับความสำคัญสูงกว่า และ สามารถร้องขอการอินเตอร์รัปต์ซ้อนในขณะที่ไอซี MCS-51 กำลังทำโปรแกรมตอบสนองการอินเตอร์รัปต์ของสัญญาณ ที่มีความสำคัญต่ำกว่าได้ และขณะที่ไอซี MCS-51 ทำโปรแกรมตอบสนองการอินเตอร์รัปต์อยู่ ไม่สามารถตอบรับการร้องขออินเตอร์รัปต์จากแหล่งกำเนิดระดับที่ต่ำกว่าหรือระดับเดียวกันได้อีก จนกว่าจบโปรแกรมตอบสนองการอินเตอร์รัปต์ก่อน

3.3 รีจิสเตอร์ TCON (Timer Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ตำแหน่งแอดเดรสที่ 88H เข้าถึงได้ในระดับบิต เป็นส่วนของรีจิสเตอร์ใช้งานพิเศษ แสดงดังภาพที่ 9.5 ทำหน้าที่ควบคุมการเลือกลักษณะสัญญาณอินเตอร์รัปต์จากภายนอก และสภาวะเริ่มต้นของรีจิสเตอร์ TCON หลังจากทำการรีเซตระบบกำหนดให้รีจิสเตอร์ TCON มีค่าเริ่มต้นของข้อมูลทุกบิตเป็นสถานะ “0” การกำหนดให้เป็นสัญญาณ อินเตอร์รัปต์ของไทเมอร์ รีจิสเตอร์นี้ทำหน้าที่เป็นแฟลก แสดงสถานะการทำงาน 4 บิต ที่สามารถเซต และเคลียร์ด้วยการทำงานของฮาร์ดแวร์ ส่วนที่เหลืออีก 4 บิต เป็นบิตควบคุมการทำงานของไทเมอร์ที่เซต และเคลียร์ด้วยคำสั่งทางซอฟต์แวร์จากการใช้คำสั่งการเซตบิต หรือการโอนย้ายข้อมูล

TCON.7	TCON.6	TCON.5	TCON.4	TCON.3	TCON.2	TCON.1	TCON.0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

ภาพที่ 9.5 รีจิสเตอร์ควบคุมการเลือกลักษณะสัญญาณอินเตอร์รัปต์ (Timer Control Register)

TF1: TCON.7 (Timer1 Overflow Flag) ทำหน้าที่เป็นแฟลกแสดงสถานะลอจิกเป็น “1” เมื่อเกิดค่านับเกิน ที่ไทเมอร์1 กระโดดไปทำโปรแกรมการตอบสนองการอินเตอร์รัปต์ในตำแหน่งเริ่มต้นที่แอดเดรส 001BH ของหน่วยความจำโปรแกรม และเคลียร์บิตนี้ให้เป็น "0" อัตโนมัติ โดยทางฮาร์ดแวร์

TR1: TCON.6 (Timer1 Run Rontrol Bit) เป็นบิตควบคุมการทำงานของไทเมอร์ 1 ถ้าให้เป็นสถานะลอจิก "1" ควบคุมให้ไทเมอร์ 1 เริ่มทำงานและถ้าเคลียร์ให้บิตนี้เป็นสถานะลอจิก "0" ทำการควบคุมให้ไทเมอร์ 1 หยุดทำงาน

TF0: TCON.5 (Timer1 Overflow Flag) ทำหน้าที่เป็นแฟลกแสดงสถานะลอจิกเป็น “1” เมื่อเกิดค่านับเกิน ที่ไทเมอร์0 และกระโดดไปทำโปรแกรมการตอบสนองการอินเตอร์รัปต์ซึ่งอยู่ในตำแหน่งเริ่มต้นที่แอดเดรส 000BH ของหน่วยความจำโปรแกรม และบิตถูกเคลียร์ให้เป็น "0" อัตโนมัติโดยทางฮาร์ดแวร์

TR0: TCON.4 (Timer0 Run Control Bit) เป็นบิตควบคุมการทำงานของไทเมอร์ 0 โดยถ้าเซตบิตนี้ให้เป็นสถานะลอจิก "1" เป็นการควบคุมให้ไทเมอร์ 0 เริ่มทำงาน และถ้าเคลียร์บิตนี้เป็นสถานะลอจิก "0" ควบคุมให้ไทเมอร์ 0 หยุดทำงาน

IE1: TCON.3(External Interrupt1 Edge Flag) เป็นแฟลกการร้องขออินเตอร์รัปต์ภายนอกของสัญญาณ $\overline{\text{INT1}}$ บิตถูกเซตด้วยฮาร์ดแวร์เมื่อมีสัญญาณอินเตอร์รัปต์ที่ขา $\overline{\text{INT1}}$ และบิตถูกเคลียร์อัตโนมัติ เมื่อไอซี MCS-51 กระโดดไปทำโปรแกรมตอบสนองการอินเตอร์รัปต์ของสัญญาณ $\overline{\text{INT1}}$ ซึ่งอยู่ในตำแหน่งเริ่มต้นที่แอดเดรส 0013H ของหน่วยความจำโปรแกรม

IT1: TCON.2 (Interrupt1 Type Control Bit) เป็นบิตควบคุมการเลือกรูปแบบการแอคทีฟของสัญญาณอินเตอร์รัปต์ จากภายนอกที่ขา$\overline{\text{INT1}}$ สามารถเซตหรือเคลียร์ด้วยซอฟต์แวร์ ดังนี้

1 = เลือกใช้สัญญาณอินเตอร์รัปต์ $\overline{\text{INT1}}$ แอคทีฟที่การเปลี่ยนจาก "1" ไป “0” (ขอบขาลง)

0 = เลือกใช้สัญญาณอินเตอร์รัปต์ $\overline{\text{INT1}}$ แอคทีฟที่ลอจิก “0” (Level)

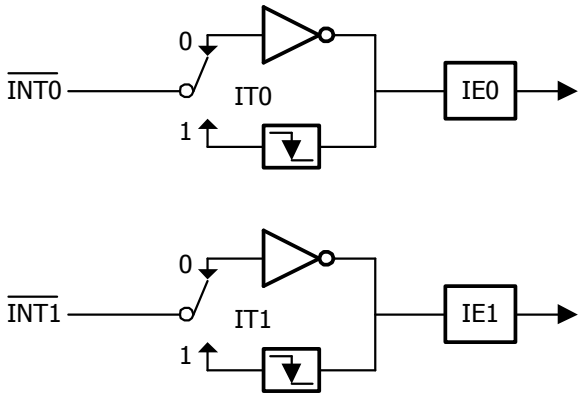
IE0: TCON.1 (External Interrupt0 Edge Flag) เป็นแฟลกร้องขออินเตอร์รัปต์จากสัญญาณ $\overline{\text{INT0}}$ บิตนี้ ถูกเซตด้วยฮาร์ดแวร์เมื่อมีสัญญาณอินเตอร์รัปต์เข้ามาที่ขา $\overline{\text{INT0}}$ และถูกเคลียร์อัตโนมัติเมื่อไอซี MCS-51 กระโดดไปทำโปรแกรมตอบสนองการอินเตอร์รัปต์ของสัญญาณ $\overline{\text{INT0}}$ ในตำแหน่งแอดเดรส 0003H ของหน่วยความจำโปรแกรม

IT0: TCON.0 (Interrupt0 Type Control Bit) เป็นบิตควบคุมการเลือกรูปแบบการแอคทีฟของสัญญาณอินเตอร์รัปต์จากภายนอกที่ขา$\overline{\text{INT0}}$ สามารถเซต หรือเคลียร์ด้วยซอฟต์แวร์

1 = เลือกใช้สัญญาณอินเตอร์รัปต์ $\overline{\text{INT0}}$ แอคทีฟที่การเปลี่ยนจาก "1" ไป "0" (ขอบขาลง)

0 = เลือกใช้สัญญาณอินเตอร์รัปต์ $\overline{\text{INT0}}$ แอคทีฟที่ลอจิก "0" (Level)

การเลือกใช้งานลักษณะสัญญาณการร้องขอการอินเตอร์รัปต์ ทำได้สองลักษณะโดยเลือกที่รีจีสเตอร์ TCON เลือกใช้สัญญาณแอคทีฟที่ระดับลอจิก "0" หรือ Low Level Triggered เลือกใช้ที่ขอบขาลงของสัญญาณที่เปลี่ยนแปลงสถานะจากลอจิก "1" ไป "0" (Falling Edge) แสดงดังภาพที่ 9.6

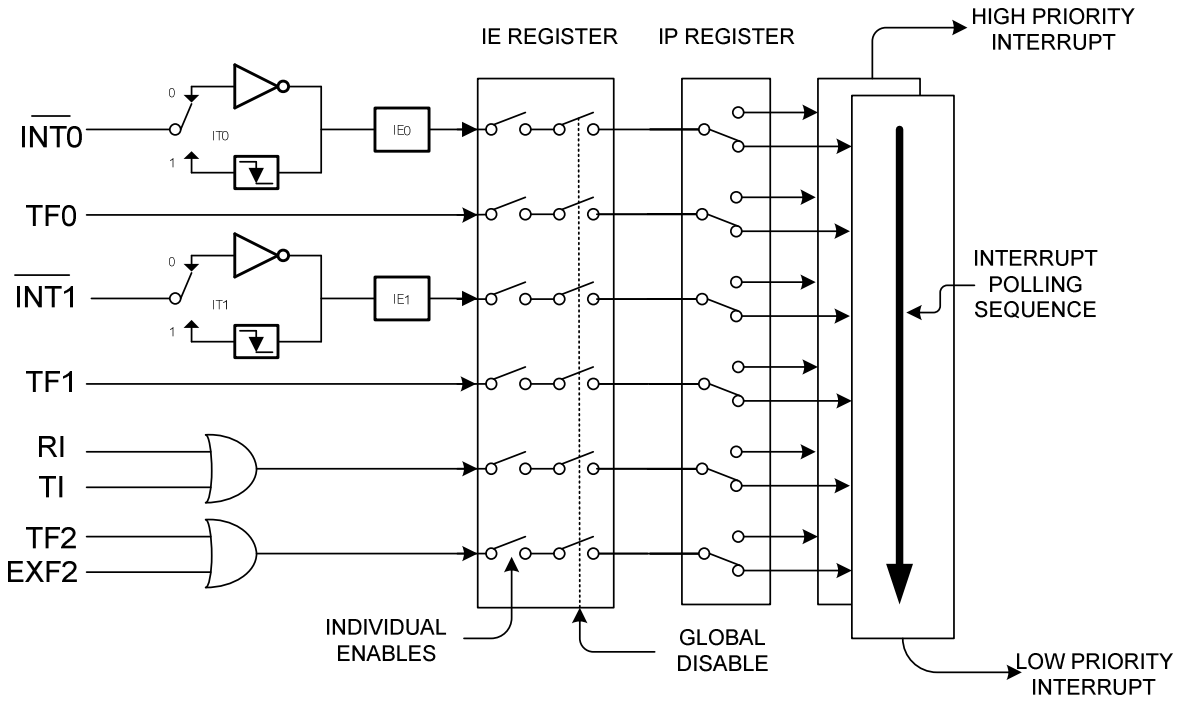


ภาพที่ 9.6 แสดงการเลือกใช้สัญญาณแอคทีฟที่ระดับลอจิก "0" หรือที่ขอบขาลงของสัญญาณ

4. โปรแกรมบริการอินเตอร์รัปต์

สัญญาณต่างๆ ของการร้องขออินเตอร์รัปต์ทั้งหมด สามารถสร้างหรือยกเลิกได้ด้วยซอฟต์แวร์ และแหล่งกำเนิดสัญญาณอินเตอร์รัปต์ทั้งหมด สามารถควบคุมให้ทำการร้องขอการอินเตอร์รัปต์ หรือไม่ก็ได้ โดยการเซตหรือเคลียร์บิตที่อยู่ในรีจิสเตอร์ IE ส่วนของการกำหนดค่าของบิตต่างๆ ในรีจิสเตอร์ เพื่อควบคุมการทำงาน แสดงดังภาพที่ 9.7

โปรแกรมตอบสนองการอินเตอร์รัปต์แต่ละแหล่ง ถูกเริ่มต้นที่ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรม โดยถูกระบุตำแหน่งไว้ภายในไอซี MCS-51 หรือเรียกว่าอินเตอร์รัปต์เวคเตอร์แสดงดังภาพที่ 9.8 ดังนั้นการเริ่มต้นเขียนคำสั่ง หลังถูกอินเตอร์รัปต์ให้ เขียนโดยมีเนื้อที่ไม่เกิน 8 ไบต์ และคำสั่งสุดท้ายต้องใช้คำสั่ง RETI เพื่อคืนค่าในสแตกพอรย์เตอร์ ให้กับโปรแกรมเคาน์เตอร์ที่เป็นค่าแอดเดรสถัดไป ก่อนเกิดการอินเตอร์รัปต์ เพื่อกลับไปทำงานในโปรแกรมหลักต่อไป และถ้าหากโปรแกรมมี ความยาวมากกว่า 8 ไบต์ ต้องเขียนโปรแกรมย่อยไว้ที่แอดเดรสตำแหน่งหลังๆ แล้วใช้คำสั่งกระโดดไปทำโปรแกรมย่อย จากภาพที่ 9.8 แบ่งคำสั่งการใช้งานการตอบสนองการอินเตอร์รัปต์ได้ดังนี้



ภาพที่ 9.7 แสดงรีจิสเตอร์ต่างๆ ที่เกี่ยวข้องกับการอินเตอร์รัปต์

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

4.1 คำสั่งที่ 1 LJMP MAIN, LJMP INT_0, LJMP INT_1

หากบันทึกข้อมูลลงในไอซี ตั้งแต่ตำแหน่งแอดเดรสที่ 0000H เป็นต้นไป และเมื่อถึงที่ตำแหน่งแอดเดรส 0003H ข้อมูลของโปรแกรมหลัก ทับกับข้อมูลในตำแหน่งแอดเดรส 0003H ซึ่งเป็นส่วนของการบริการอินเตอร์รัปต์ INT0 ถ้าหากโปรแกรมหลักใช้เนื้อที่ในการเขียนโปรแกรมยาวไปถึงแอดเดรสที่ 0013H ทับกับตำแหน่งของ INT1

เมื่อไอซีเริ่มทำงานที่แอดเดรส 0000H จึงต้องเขียนคำสั่งให้กระโดดไปเริ่มต้นตำแหน่งของเลเบล MAIN เพื่อให้ไอซีเริ่มทำงานที่โปรแกรมหลักในตำแหน่งแอดเดรสนี้ หรือใช้คำสั่ง ORG 0050H แทนคำสั่ง ORG 0000H เพื่อให้ไอซี MCS -51 เริ่มทำที่แอดเดรส 0050H เป็นตำแหน่งแรก ส่วนระยะห่างของแอดเดรสในหน่วยความจำบริการการอินเตอร์รัปต์แต่ละตัว ห่างกัน 8 ไบต์ หากเขียนโปรแกรมบริการอินเตอร์รัปต์ ที่มีความซับซ้อนมาก อาจมีเนื้อที่ไม่เพียงพอ ดังนั้นจึงต้องใช้คำสั่ง LJMP INT_0 และ LJMP INT_1 เพื่อกระโดดไปทำงานที่เลเบล INT_0 และ INT_1 ที่มีเนื้อที่ในการเขียนคำสั่งอยู่ในแอดเดรสลำดับหลังๆ

4.2 คำสั่งที่ 2 RETI

เป็นคำสั่งที่ต้องใช้หลังจากไอซี MCS-51 ทำงานในส่วนของโปรแกรมย่อยบริการอินเตอร์รัปต์เสร็จ และต้องกลับไปตำแหน่งแอดเดรสเดิมที่โปรแกรมหลัก คำสั่ง RETI เป็นการคืนค่ารีจิสเตอร์ PC ที่อยู่ในรีจิสเตอร์ SP และกลับสู่โปรแกรมหลัก ซึ่งอักษร I หมายถึง **I**nterrupt นั้นเอง

4.3 คำสั่งที่ 3 SETB EA

เป็นคำสั่งตอบรับการอินเตอร์รัปต์ทั้งหมด หรือเขียนได้เป็น SETB IE.7 โดยเซตที่ตำแหน่งบิต EA (Enable/Disable all Interrupt) ให้สถานะลอจิกเป็น “1” โดยเขียนคำสั่งในส่วนของเลเบล MAIN

```
                        ORG     0000H
คำสั่งที่ 1             LJMP    MAIN
                        ORG     0003H
                        LJMP    INT_1
                        ORG     0013H
                        LJMP    INT_2
คำสั่งที่ 3     MAIN:   SETB    EA
คำสั่งที่ 4             SETB    EX0
คำสั่งที่ 4             SETB    EX1
คำสั่งที่ 5             SETB    IT0
คำสั่งที่ 5             SETB    IT1
คำสั่งที่ 6             SETB    IP.2
                                .
                INT_0:          .
                                .
                                .
คำสั่งที่ 2             RETI
                INT_1:          .
                                .
                                .
        RETI
                        END
```

หน่วยความจำโปรแกรมของไอซี

แอดเดรส	ข้อมูลขนาด 8 บิต
0000H	ตำแหน่งเริ่มต้นเมื่อ Reset
0001H	
0003H	**เริ่มต้นโปรแกรมบริการ INT0**
0004H	
0005H	
0006H	
0007H	
0008H	
0009H	
000AH	
000BH	**เริ่มต้นโปรแกรมบริการ TIMER0**
000CH	
000DH	
000EH	
000FH	
0010H	
0011H	
0012H	
0013H	**เริ่มต้นโปรแกรมบริการ INT1**
0014H	
0015H	

ภาพที่ 9.8 ตัวอย่างการเขียนโปรแกรมบริการอินเตอร์รัปต์

4.4 คำสั่งที่ 4 SETB EX0 และ SETB EX1

เป็นการเลือกอินเตอร์รัปต์ภายนอกโดยเซตบิตที่รีจีสเตอร์ IE บิต EX0 หรือบิต EX1 เป็นการอนุญาตให้มีอินเตอร์รัปต์ที่ขา $\overline{\text{INT0}}$ หรือที่ขา $\overline{\text{INT1}}$ และสามารถเซตบิตทั้งคู่ได้

SETB EX0 ; ให้มีการบริการอินเตอร์รัปต์ที่ขา $\overline{\text{INT0}}$ โดยกระโดดไปที่แอดเดรส 0003H หรือเขียนได้เป็น SETB IE.0

SETB EX1 ; ให้มีการบริการอินเตอร์รัปต์ที่ขา $\overline{\text{INT1}}$ โดยกระโดดไปที่แอดเดรส 0013H หรือเขียนได้เป็น SETB IE.1

จากคำสั่งที่ 3 และคำสั่งที่ 4 สามารถเขียนรวมได้เป็น

```
MOV   IE, #10000101B          ; ทำการเซตบิต EA="1", EX0="1", EX1="1"
SETB  P3.2
SETB  P3.3
```

หลังจากกำหนดค่าในรีจิสเตอร์ต่าง ๆ แล้ว การใช้งานสัญญาณร้องขอการอินเตอร์รัปต์จากภายนอก $\overline{\text{INT0}}$ และ $\overline{\text{INT1}}$ (ขา P3.2 และ P3.3) ต้องเซตบิต P3.2 และ P3.3 ให้เป็น "1" เพื่อทำให้เป็นอิมพีแดนซ์สูง โดยใช้คำสั่ง SETB P3.2 และ SETB P3.3 (เป็นการกำหนดให้เป็นอินพุต)

4.5 คำสั่งที่ 5 SETB IT0 และ SETB IT1

เป็นคำสั่งเลือกลักษณะของสัญญาณการอินเตอร์รัปต์ โดยเซตบิต IT0 หรือบิต IT1 ที่รีจีสเตอร์ TCON โดยเขียนคำสั่งในส่วนของโปรแกรมในเลเบล MAIN การเปลี่ยนระดับสัญญาณเพื่อให้เป็นสถานะลอจิกต่ำหรือ "0" ต้องทำให้ขาสัญญาณนี้กลับเป็นสถานะลอจิก "1" ก่อนการบริการอินเตอร์รัปต์เสร็จสิ้น เพื่อป้องกันเกิดอินเตอร์รัปต์ซ้ำซ้อน การเปลี่ยนระดับสัญญาณจาก "1" เป็น "0" ที่ขอบขาลง ต้องรักษาสถานะของลอจิก "0" ไม่น้อยกว่า 1 แมชชีนไซเคิลโดยใช้คำสั่ง

```
SETB  IT0     ; ลักษณะของสัญญาณเป็นการทำงานที่ขอบขาลง
SETB  IT1
```

4.6 คำสั่งที่ 6 SETB IP.x และ CLR IP.x

เป็นการเลือกลำดับความสำคัญของการอินเตอร์รัปต์ โดยกำหนดรีจีสเตอร์ IP ที่ตำแหน่งบิต IPx และ IPx ตัวอย่างเช่น

```
SETB IP.2     กำหนดลำดับความสำคัญสูงของสัญญาณ INT1 (External Interrupt 1)
CLR  IP.0     กำหนดลำดับความสำคัญต่ำของสัญญาณ INT0 (External Interrupt 0)
```

หากบิตใดกำหนดค่าเป็นสถานะลอจิก "1" มีลำดับความสำคัญสูง แต่หากไม่มีการเซตบิตใดๆ ในการจัดความสำคัญ $\overline{\text{INT0}}$ มีลำดับความสำคัญสูงกว่า $\overline{\text{INT1}}$ ซึ่งเป็นค่าเริ่มต้นของไอซี MCS-51

5. การเขียนโปรแกรมรับอินเตอร์รัปต์

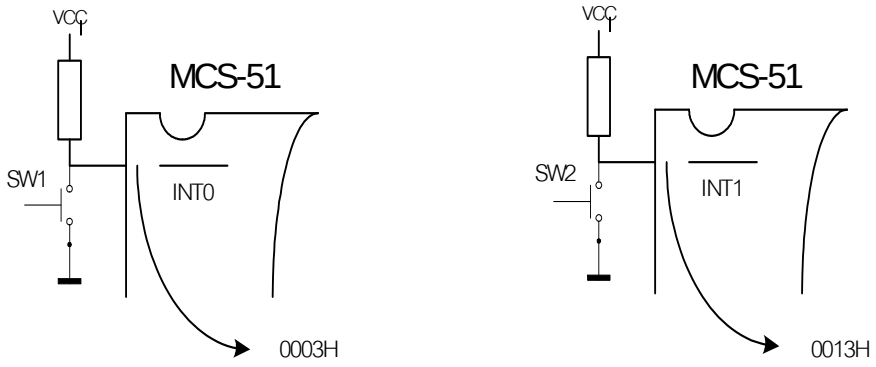
5.1 โปรแกรมรับอินเตอร์รัปต์จากภายนอก

การรับอินเตอร์รัปต์จากภายนอก เป็นการตรวจสอบสัญญาณที่ป้อนให้กับขาไอซี MCS-51 มีสัญญาณการร้องขอ และตำแหน่งขาการอินเตอร์รัปต์ แสดงดังตารางที่ 9.3 เช่นสัญญาณที่ได้จากการกดสวิตซ์ การตรวจสอบสถานะลอจิกของสัญญาณรายคาบ และสัญญาณที่มาจากวงจรต่างๆ ภายนอกไอซี การต่อกับไอซี MCS-51 กับสัญญาณภายนอก เช่นสวิตซ์แสดงได้ดังภาพที่ 9.9

ตารางที่ 9.3 แสดงสัญญาณการร้องขอ ความหมาย และตำแหน่งขาการอินเตอร์รัปต์

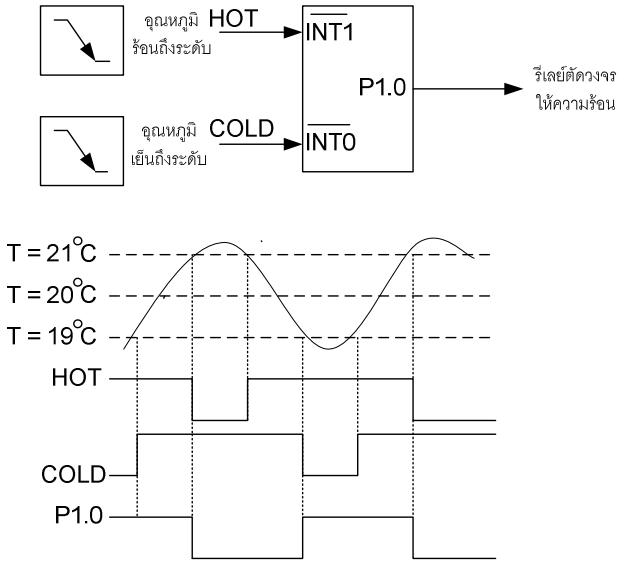
สัญญาณการขออินเตอร์รัปต์	ความหมายและตำแหน่งขา
$\overline{\text{INT0}}$	สัญญาณการร้องขอการอินเตอร์รัปต์จากภายนอกทางขา P3.2 (ขาที่ 6 ของไอซีขนาด 20 ขา และ ขาที่12 ของไอซี 40 ขา)
$\overline{\text{INT1}}$	สัญญาณการร้องขอการอินเตอร์รัปต์จากภายนอกทางขา P3.3 (ขาที่ 7ของไอซีขนาด 20 ขา และ ขาที่12 ของไอซี 40 ขา)

การอินเตอร์รัปต์สามารถเลือกใช้สัญญาณที่แอคทีฟโดยการเปลี่ยนจากสถานะลอจิก "1" ไปลอจิก "0” (ขอบขาลง) หรือเลือกใช้สัญญาณอินเตอร์รัปต์ที่แอคทีฟโดยลอจิก "0" (Level)



ภาพที่ 9.9 แสดงตำแหน่งขา และการร้องขออินเตอร์รัปต์ ของไอซี MCS-51

ตัวอย่างที่ 1 ใช้การอินเตอร์รัปต์ควบคุมตู้อบเพื่อรักษาอุณหภูมิอยู่ที่ 20 °C ±1°C ตลอดเวลา เขียนเป็นไดอะแกรม แสดงดังภาพที่ 9.10



ภาพที่ 9.10 แสดงไดอะแกรมของการควบคุมอุณหภูมิ

จากตัวอย่างที่ 1 กำหนดเงื่อนไขตามหัวข้อดังนี้

ข้อที่ 1 ตัวตรวจสอบอุณหภูมิต่ออยู่กับขา $\overline{\text{INT0}}$ และ $\overline{\text{INT1}}$ ให้สัญญาณเป็น $\overline{\text{HOT}}$ และ $\overline{\text{COLD}}$ ตามข้อกำหนดดังนี้

$\overline{\text{HOT}}$ = 0 เมื่ออุณหภูมิ T > 21°C

$\overline{\text{COLD}}$ = 0 เมื่ออุณหภูมิ T < 19°C

ข้อที่ 2 รีเลย์ต่อกับอุปกรณ์ให้ความร้อน ตามข้อกำหนดดังนี้

P1.0 = 1 รีเลย์ทำงานให้ความร้อน

P1.0 = 0 รีเลย์หยุดทำงานไม่ให้ความร้อน

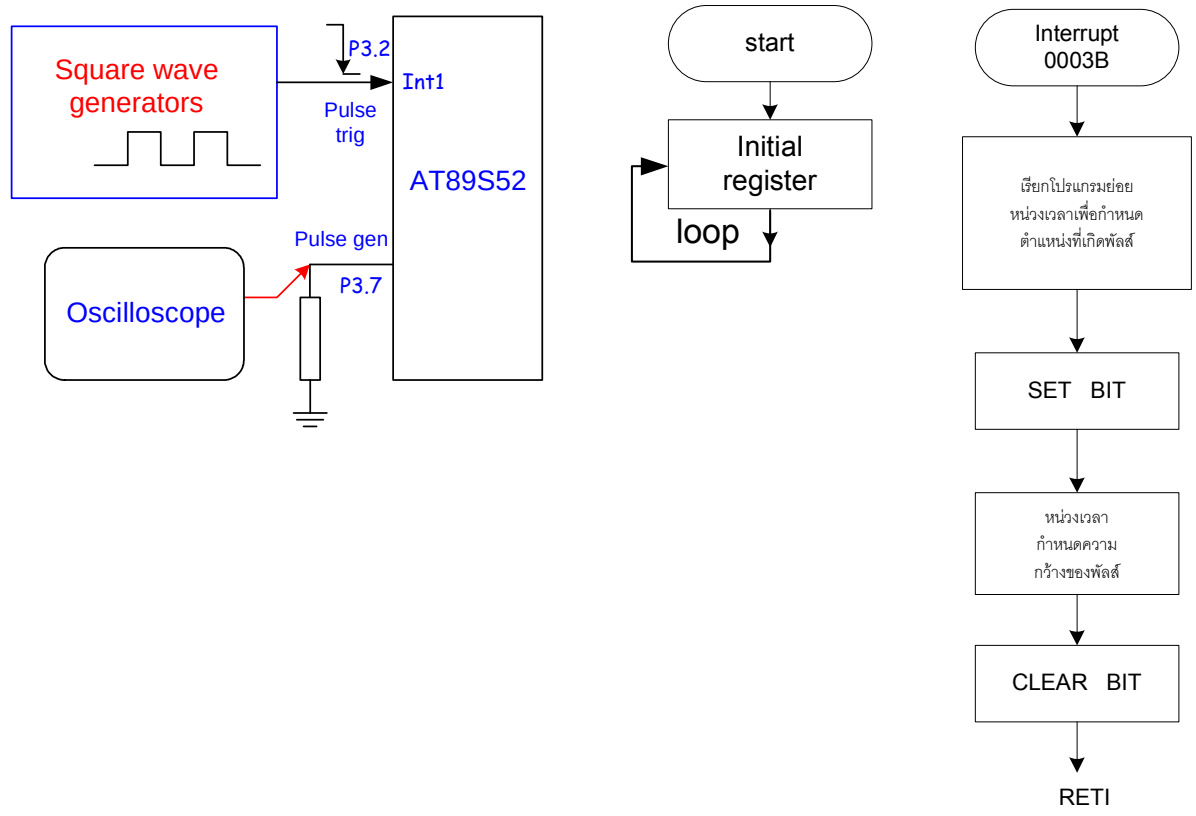
ข้อที่ 3 กำหนดเขียนโปรแกรมให้ส่วนกำหนดความร้อนทำงานเมื่ออุณหภูมิต่ำกว่า 19°C หยุดให้ความร้อนเมื่อ T > 21°C ดังโปรแกรม

```
RELAY1   BIT  P1.0     ; กำหนดชื่อ RELAY1 คือตำแหน่งบิตที่ P1.0
HOT      BIT  P3.2     ; กำหนดขา INT1 ชื่อ HOT คือตำแหน่งบิตที่ P3.2
ORG      0000H
LJMP     MAIN          ; ไปทำที่โปรแกรมหลัก
ORG      0003H
CLR      RELAY1        ; หยุดให้ความร้อน
RETI
ORG      0013H
```

```
        SETB   RELAY1       ; ให้ความร้อน
        RETI
        ORG    0050H
MAIN:   MOV    IE,#85H      ; กำหนดการอินเตอร์รัปต์
        SETB   IT0          ; เลือกสัญญาณกระตุ้นที่ขอบลบ
        SETB   IT1          ; เลือกสัญญาณกระตุ้นที่ขอบลบ
        SETB   RELAY1       ; เริ่มให้ความร้อน
        JB     HOT, SKIP    ; ให้ความร้อนจนกว่า T > 21°C
        CLR    RELAY1       ; หยุดให้ความร้อน
SKIP:   SJMP   $
        END
```

ตัวอย่างที่ 2 กำหนดเงื่อนไข ตามหัวข้อดังนี้

ข้อที่ 1 ป้อนสัญญาณ Square Wave ความถี่เอาต์พุตอยู่ในระหว่าง 50H - 10 kH5 เป็นสัญญาณอินพุตที่ขาอินเตอร์รัปต์ภายนอก $\overline{\text{INT0}}$



ภาพที่ 9.11 แสดงวงจรและผังงานโปรแกรมตัวอย่างที่ 2

ข้อที่ 2 เมื่อไอซี MCS-51 ได้รับการร้องขอ การอินเตอร์รัปต์จากขา $\overline{\text{INT0}}$ แล้ว ให้กระโดดไปทำงานที่โปรแกรมย่อยการอินเตอร์รัปต์

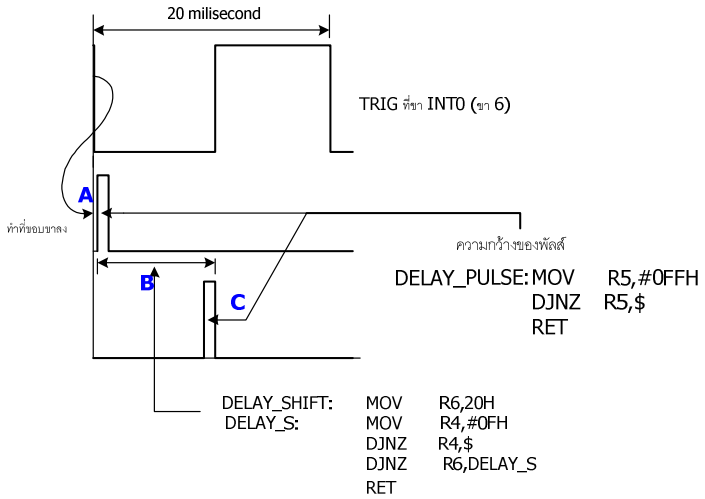
ข้อที่ 3 ส่วนโปรแกรมย่อยการอินเตอร์รัปต์ กำหนดให้สร้างสัญญาณพัลส์ ที่มีคาบเวลาในช่วงสั้นๆ 1 ค่า และก่อนทำการสร้างพัลส์ ต้องหน่วงเวลาด้วยค่าที่กำหนดไว้ล่วงหน้า ในส่วนของโปรแกรมหลักเพื่อกำหนดตำแหน่งที่ต้องการหลังจากเกิดการอินเตอร์รัปต์จากภายนอก

จากขั้นตอนดังกล่าวเขียนเป็นผังงาน และวงจรแสดงดังภาพที่ 9.11

```
; ******************************************
; **      EXTERNAL INTERRUPT INT0          **
; ******************************************
  PULSE_TRIG  EQU   P3.2    ; กำหนดค่า PULSE_TRIG คือตำแหน่งบิตที่ P3.2
  PULSE_GEN   EQU   P3.7    ; กำหนดค่า PULSE_GEN คือตำแหน่งบิตที่ P3.7
        ORG   0000H         ; เริ่มต้นที่หน่วยความจำโปรแกรมแอดเดรส 0000H
        SJMP  MAIN          ; กระโดดไปทำที่ตำแหน่งของแอดเดรสที่เลเบล MAIN
        ORG   0003H         ; แอดเดรสเริ่มต้นโปรแกรมบริการอินเตอร์รัปต์ INT0
        SJMP  PULSE         ; ให้กระโดดไปทำที่ตำแหน่งของแอดเดรสที่เลเบล PULSE
;*****  โปรแกรมหลัก กำหนดค่าในรีจิสเตอร์ที่เกี่ยวข้องกับการอินเตอร์รัปต์  *******
MAIN:   MOV   20H,#0AFH     ;กำหนดค่าเริ่มต้นของ 20H เป็นค่าของการหน่วงเวลา
        SETB  EA            ; เปิดการบริการอินเตอร์รัปต์ทั้งหมด
        SETB  EX0           ; กำหนดให้มีการอินเตอร์รัปต์ภายนอกที่ขา INT0
        SETB  IT0           ; กำหนดให้อินเตอร์รัปต์ที่ขอบขาลงของสัญญาณ
        MOV   P3,#0FFH      ; กำหนดให้พอร์ต P3 มีค่าเท่ากับ FFH (อินพุต)
        SJMP  $             ; วนรอรับสัญญาณการอินเตอร์รัปต์
;****  โปรแกรมย่อยการบริการอินเตอร์รัปต์ โดยกำหนดให้เกิดพัลส์ 1 ลูกถ้ามีการอินเตอร์รัปต์เกิดขึ้น ***
PULSE: ACALL  DELAY_SHIFT
        SETB  PULSE_GEN     ; ให้บิตที่ชื่อ PULSE_GEN เป็นสถานะลอจิก “1”
        ACALL DELAY_PULSE   ; หน่วงเวลาเพื่อกำหนดความกว้างของพัลส์
        CLR   PULSE_GEN     ; ให้บิตที่ชื่อ PULSE_GEN เป็นสถานะลอจิก “0”
        RETI                ; ออกจากโปรแกรมย่อยการอินเตอร์รัปต์
;********  โปรแกรมย่อยหน่วงเวลาเพื่อกำหนดตำแหน่งของการเกิดพัลส์ **********
DELAY_SHIFT:    MOV   R6,20H
DELAY_S:        MOV   R4,#0FH
```

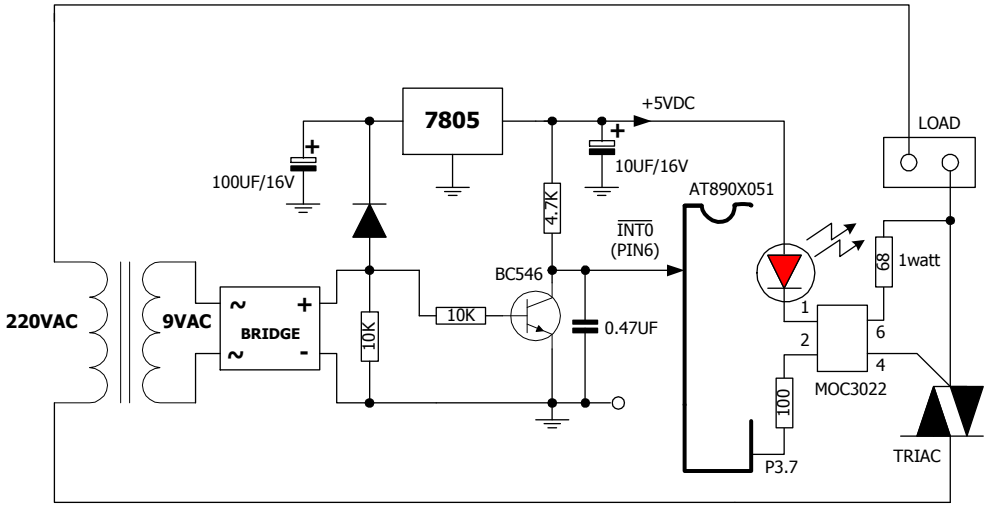
```
                DJN5    R4,$
                DJN5    R6,DELAY_S
                RET
;****   โปรแกรมย่อยหน่วงเวลา เพื่อกำหนดความกว้างของพัลส์  *****
DELAY_PULSE:    MOV     R5,#0FFH
                DJN5    R5,$
                RET
                END
```

จากโปรแกรมสรุปเป็นขั้นตอนการเกิดสัญญาณ และคำสั่งสร้างพัลส์แสดงดังภาพที่ 9.12 โดยการกำหนดให้ไอซี MCS-51 เริ่มการอินเตอร์รัปต์ที่ขอบขาลงของสัญญาณอินพุต (A) ป้อนให้ขา $\overline{\text{INT0}}$ และส่วนการเลื่อนสัญญาณเริ่มต้นเกิดสัญญาณพัลส์ตำแหน่งใดนั้น (B) ต้องกำหนดค่าข้อมูลที่แอดเดรส 20H ส่งค่าผ่าน R6 ในโปรแกรมย่อยการหน่วงเวลา ส่วนโปรแกรมย่อยการกำเนิดพัลส์ (C) กำหนดความกว้างที่โปรแกรมย่อย DELAY_PULSE



ภาพที่ 9.12 แสดงสัญญาณ และส่วนของคำสั่งที่สร้างพัลส์

จากภาพที่ 9.13 เป็นตัวอย่างวงจร สร้างสัญญาณการอินเตอร์รัปต์ภายนอกให้กับขา $\overline{\text{INT0}}$ โดยใช้สัญญาณซายน์เวฟ (Sine Wave) จากหม้อแปลงไฟขนาด 9 โวลต์ ผ่านวงจรบริดจ์เร็กติไฟร์ มีความถี่ 50 Hz เพื่อเป็นสัญญาณที่มีจังหวะเวลาตรงกัน (Sync) ส่วนเอาต์พุตของไอซี MCS-51 ขา P3.7 ต่อกับออปโตไดแอคและไตรแอค ตามลำดับ โดยเขียนโปรแกรมเพื่อกำหนดตำแหน่งการทริกมุมต่างๆของไตรแอคได้



ภาพที่ 9.13 แสดงตัวอย่างการสร้างสัญญาณเพื่อทริกให้กับขา $\overline{\text{INT0}}$

5.2 โปรแกรมรับอินเตอร์รัปต์จากไทเมอร์ เคาน์เตอร์

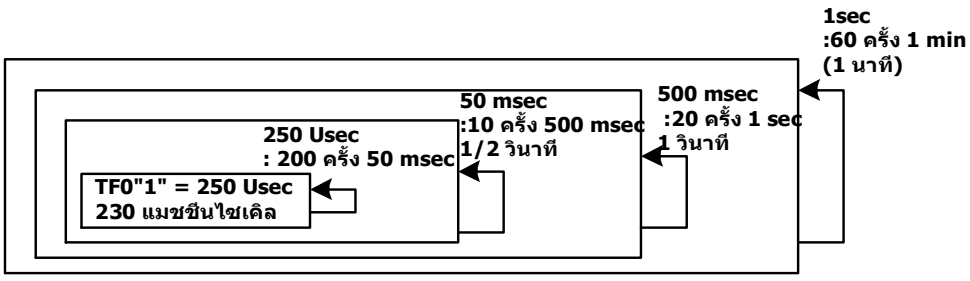
ตัวอย่างที่ 3 เขียนโปรแกรมสร้างฐานเวลาไทเมอร์โหมด 2 ใช้คริสตอลความถี่ 12MHz กำหนดค่ารีจิสเตอร์ TH0 = -250 ได้ค่าเวลา 250 µS ให้โปรแกรมวนลูป ตรวจสอบจำนวนครั้งในการร้องขออินเตอร์รัปต์ กับค่าที่ตั้งไว้

ถ้าจำนวนนับเกินถึง 200 ครั้ง หาค่าได้เท่ากับ 250 x 200 = 50,000 µS หรือ 50 msec

ถ้าวนรอบของค่า 50 msec อีก 10 รอบ ได้เท่ากับ 50 x 10 = 500 msec

ถ้าวนรอบ 500 msec อีก 2 รอบ ได้เท่ากับ 500 x 2 = 1 วินาที (second)

(การกำหนด 50 msec = 10 รอบ 500 msec = 2 รอบ เพื่อให้ได้ 1 วินาที เพราะต้องการกำหนดให้แอลอีดีกระพริบ 1 ครั้งในครึ่งวินาที ถ้าไม่ให้แอลอีดีกระพริบกำหนดให้ 50 msec = 20 รอบ ใน 1 วินาที)



ภาพที่ 9.14 แสดงรอบการทำงานโปรแกรมสร้างฐานเวลา

ถ้าทำการวนรอบ 1 วินาที อีก 60 รอบ ได้เท่ากับ 1 วินาที x 60 = 1 นาที (Min) ดังนั้นต้องทำทั้งหมด ((((250 x 200) 10)2)60) = 1 นาที รอบการทำงานแสดงดังภาพที่ 9.14

ขั้นตอนที่ 1 ทำการกำหนดชื่อของตำแหน่งบิตต่างๆ

```
LED_250µS    BIT    P1.0    ; LED_250 µS แสดงผลที่ตำแหน่งบิตที่ P1.0
LED_50msec   BIT    P1.1    ; LED_50 msec แสดงผลที่ตำแหน่งบิตที่ P1.1
LED_500msec  BIT    P1.2    ; LED_500 msec แสดงผลที่ตำแหน่งบิตที่ P1.2
LED_1sec     BIT    P1.3    ; LED_1sec แสดงผลที่ตำแหน่งบิตที่ P1.3
LED_1min     BIT    P1.4    ; LED_1 min แสดงผลที่ตำแหน่งบิตที่ P1.4
LED_FINISH   BIT    P1.7    ; LED_FINISH แสดงผลที่ตำแหน่งบิตที่ P1.7
COUNT_200    EQU    20H     ; แอดเดรส 20H กำหนดให้ชื่อ COUNT_200
COUNT_10     EQU    21H     ; แอดเดรส 21H กำหนดให้ชื่อ COUNT_10
COUNT_2      EQU    22H     ; แอดเดรส 22H กำหนดให้ชื่อ COUNT_2
SEC_60       EQU    23H     ; แอดเดรส 23H กำหนดให้ชื่อ SEC_60
 Min         EQU    24H     ;  แอดเดรส 24H กำหนดให้ชื่อ Min
```

ขั้นตอนที่ 2 กำหนดจุดเริ่มต้นของโปรแกรม และกำหนดค่าเริ่มต้นของรีจิสเตอร์ต่างๆ ที่ใช้บริการอินเตอร์รัปต์

```
        ORG   0000H              ; เริ่มต้นที่หน่วยความจำโปรแกรมแอดเดรส 0000H
        SJMP  MAIN               ; กระโดดไปทำที่ตำแหน่งของแอดเดรสที่เลเบล MAIN
        ORG   000BH              ; แอดเดรสเริ่มต้นการอินเตอร์รัปต์ INT0
        SJMP  INT_T0             ; ให้กระโดดไปทำที่เลเบล INT_T0
        ORG   0030H              ; เริ่มต้นที่หน่วยความจำโปรแกรมแอดเดรส 0030H
MAIN:   MOV   P1,#00H            ; กำหนดให้พอร์ต P1 มีค่าเท่ากับ 00H
        MOV   Min,#03            ; ให้หน่วยความจำ Min  มีค่าเท่ากับ 03 นาที
        MOV   COUNT_200,#200     ; ให้หน่วยความจำ COUNT_200  มีค่าเท่ากับ 200
        MOV   COUNT_10,#10       ; ให้หน่วยความจำ COUNT_10  มีค่าเท่ากับ 10
        MOV   COUNT_2,#2         ; ให้หน่วยความจำ COUNT_2  มีค่าเท่ากับ 2
        MOV   SEC_60,#60         ; ให้หน่วยความจำ SEC_60 มีค่าเท่ากับ 60
        MOV   TMOD,#00000010B    ; กำหนดให้มีการทำงานที่โหมด 2 เป็นไทเมอร์
        MOV   TL0,# 06H          ; กำหนดให้รีจิสเตอร์ TL0 มีค่าเท่ากับ 250µS
        MOV   TH0,# 06H          ; กำหนดให้รีจิสเตอร์ TH0 มีค่าเท่ากับ 250µS
        SETB  EA                 ; อนุญาตให้มีการอินเตอร์รัปต์ทั้งหมด
        SETB  ET0                ; อนุญาตให้มีการอินเตอร์รัปต์ที่ Timer0
```

```
    SETB   TR0                  ; เริ่มมีการอินเตอร์รัปต์ที่ INT0
    SJMP   $                    ; วนคอยการอินเตอร์รัปต์ในแต่ละครั้ง
```

ขั้นตอนที่ 3 โปรแกรมย่อยบริการอินเตอร์รัปต์

กำหนดค่าในโปรแกรมย่อยบริการอินเตอร์รัปต์ 250 µS x 200 ครั้ง = 50 millisecond เมื่อมีการอินเตอร์รัปต์ให้กลับสถานะของบิต LED_250 µS แล้วให้ตรวจสอบที่ COUNT_200 = 0 หรือไม่ ถ้าไม่เท่าให้ลดค่าลง 1 ค่า หลังจากนั้นกระโดดไปที่เลเบล END_RETI แต่ถ้ามีค่าเท่ากันให้ COUNT_200 เริ่มค่าใหม่ที่ 200 และให้กลับค่าสถานะของบิต LED_50 msec โดยแสดงผลที่แอลอีดีที่ตำแหน่งบิต P1.1

```
INT_T0:     CPL    LED_250µS
            DJNZ   COUNT_200, END_RETI
            MOV    COUNT_200, #200
            CPL    LED_50msec
```

กำหนดค่า (50 msec x 200 ครั้ง) x 10 ครั้ง = 500 millisecond แล้วให้ตรวจสอบที่ค่าข้อมูลใน COUNT_10 = 0 หรือไม่ ถ้าไม่เท่าให้ลดค่าลง 1 ค่า หลังจากนั้นกระโดดไปที่เลเบล END_RETI แต่ถ้ามีค่าเท่ากัน ให้ COUNT_10 เริ่มค่าใหม่ที่ 10 ให้กลับค่าสถานะของบิต LED_500 msec โดยแสดงผลที่แอลอีดีที่ตำแหน่งบิต P1.2

```
            DJNZ   COUNT_10, END_RETI
            MOV    COUNT_10,#10
            CPL    LED_500msec
```

กำหนดค่า ((250 msec x 200 ครั้ง) x 10 ครั้ง) x 2 ครั้ง 1 วินาที แล้วให้ตรวจสอบที่ค่าข้อมูลใน COUNT_2 = 0 หรือไม่ ถ้าไม่เท่าให้ลดค่าลง 1 ค่า หลังจากนั้นกระโดดไปที่เลเบล END_RETI แต่ถ้ามีค่าเท่ากันให้ COUNT _2 เริ่มค่าใหม่ที่ 02 ให้กลับค่าสถานะของบิต LED_1sec โดยแสดงผลที่แอลอีดีที่ตำแหน่งบิต P1.3

```
            DJNZ   COUNT_2, END_RETI
            MOV    COUNT_2, #02
            CPL    LED_1sec
```

กำหนดค่า (((250µS x 200 ครั้ง) x 10 ครั้ง) x 2 ครั้ง) x 60 ครั้ง 1 นาที แล้วให้ตรวจสอบที่ค่าข้อมูลใน SEC_60 = 0 หรือไม่ ถ้าไม่เท่าให้ลดค่าลง 1 ค่า หลังจากนั้นกระโดดไปที่เลเบล END_RETI แต่ถ้ามีค่าเท่ากันให้ SEC_60 เริ่มค่าใหม่ที่ 60 และให้กลับค่าสถานะของบิต LED_1minโดยแสดงผลที่แอลอีดีตำแหน่งบิต P1.4

```
            DJNZ   SEC_60, END_RETI
            MOV    SEC_60, #60
            CPL    LED_1min
```

กำหนดค่า ((((250μS x 200 ครั้ง) x 10 ครั้ง) x 2 ครั้ง) x 60 ครั้ง) 3 ครั้งแล้วให้ตรวจสอบที่ค่าข้อมูลใน MIN = 0 หรือไม่ ถ้าไม่เท่าให้ลดค่าลง 1 ค่าหลังจากนั้นกระโดดไปที่เลเบล END_RETI แต่ถ้ามีค่าเท่ากันให้หยุดการทำงานไทเมอร์ 0 หยุดการอินเตอร์รัปต์ และให้บิต LED_FINISH เซตเป็นสถานะลอจิก “1”

```
                DJNZ    MIN, END_RETI
                CLR     TR0
                SETB    LED_FINISH
END_RETI:       RETI
```

5.3 โปรแกรมอินเตอร์รัปต์จากพอร์ตอนุกรม

ตัวอย่างที่ 4 เขียนโปรแกรมอินเตอร์รัปต์พอร์ตอนุกรมในโหมด 1 ให้ส่งค่าข้อมูล ASCII ของอักษร A – Z ไปที่ไมโครคอมพิวเตอร์ กำหนดอัตราการรับส่งข้อมูล 9600 บิตต่อวินาที ให้แสดงผลที่โปรแกรม Hyper Terminal ตามค่าที่ส่งไป ขั้นตอนเขียนโปรแกรมมีดังนี้

ขั้นตอนที่ 1 กำหนดโหมดการรับส่งพอร์ตอนุกรมโหมด 1 ในรีจิสเตอร์ SCON

ขั้นตอนที่ 2 กำหนดการใช้ไทเมอร์ 1 โหมด 2 เพื่อกำหนดอัตราการรับส่งข้อมูล

ขั้นตอนที่ 3 กำหนดบิต SMOD ของรีจิสเตอร์ PCON ให้บิตนี้มีค่าเป็น “0”

ขั้นตอนที่ 4 กำหนดให้มีการรับอินเตอร์รัปต์ทางพอร์ตอนุกรมที่รีจิสเตอร์ IE

ขั้นตอนที่ 5 ค่าของ ASCII “ A” คือ 41H และ ASCII “ [” คือ 5BH มีลำดับต่อจาก ASCII “Z”

```
;*********************************************************************
;****    การส่งข้อมูลจากไอซี MCS-51 ผ่านพอร์ตอนุกรม: RS-232 Computer PC    ***
;*********************************************************************
                ORG     0000H
                AJMP    MAIN
                ORG     0023H
                AJMP    Tx_ISR
                ORG     0050H
MAIN:           MOV     TMOD, #00100000B    ; ใช้งานไทเมอร์ 1 โหมด 2
                MOV     TL1, #0FDH          ; กำหนดค่าการนับในรีจิสเตอร์ TL1
                MOV     TH1, #0FDH          ; กำหนดค่าเริ่มต้นการนับในรีจิสเตอร์ TH1
                SETB    TR1                 ; เริ่มต้นการทำงานไทเมอร์1
                MOV     SCON, #01000010B    ; เลือกอนุกรมโหมด 1 TI = 1
                MOV     A, #41H             ; กำหนดค่าในรีจิสเตอร์ A เป็น ASCII “A”
                MOV     PCON,#00H           ; เลือก SMOD = 0
```

```
                MOV   IE,#90H              ; อนุญาตให้มีการอินเตอร์รัปต์พอร์ตอนุกรม
                SJMP  $                    ; วนทำบรรทัดเดิม
Tx_ISR :        CLR   TI                   ; เคลียร์บิต TI เพื่อส่งไบต์ต่อไป
                CJNE  A, #5BH, UP_CHA      ; ค่าในรีจิสเตอร์ A ไม่เท่ากับ ASCII "[" ไปที่ UP_CHA
                MOV   A, #41H              ; กำหนดค่าในรีจิสเตอร์ A เป็น ASCII "A" ใหม่
UP_CHA:         MOV   SUBF, A              ; นำข้อมูลในรีจิสเตอร์ A เก็บไว้ที่รีจิสเตอร์ SUBF
                INC   A                    ; เพิ่มค่า ASCII เป็นอักษรตัวต่อไป
                RETI                       ; ออกจากโปรแกรมย่อยการอินเตอร์รัปต์
                END
```

ตัวอย่างที่ 5 เขียนโปรแกรมบริการอินเตอร์รัปต์ทางพอร์ตอนุกรมโหมด 1 โดยรับค่าข้อมูลจากไมโครคอมพิวเตอร์ กำหนดอัตราการรับส่งข้อมูลที่ 9600 บิตต่อวินาที โดยข้อมูลถูกนำไปเก็บไว้ที่รีจิสเตอร์ SBUF ทีละไบต์ ข้อมูลที่รับได้เป็นรหัส ASCII ให้แสดงผลที่พอร์ต P1 ตามค่ารับได้ ขั้นตอนเขียนโปรแกรมมีดังนี้

ขั้นตอนที่ 1 กำหนดโหมดการรับส่งพอร์ตอนุกรมโหมด 1 ในรีจิสเตอร์ SCON
ขั้นตอนที่ 2 กำหนดการใช้ไทเมอร์ 1 โหมด 2 เพื่อกำหนดอัตราการรับส่งข้อมูล
ขั้นตอนที่ 3 กำหนดบิต SMOD ของรีจิสเตอร์ PCON ให้บิตนี้มีค่าเป็น "0"
ขั้นตอนที่ 4 กำหนดให้มีการรับอินเตอร์รัปต์ทางพอร์ตอนุกรมที่รีจิสเตอร์ IE

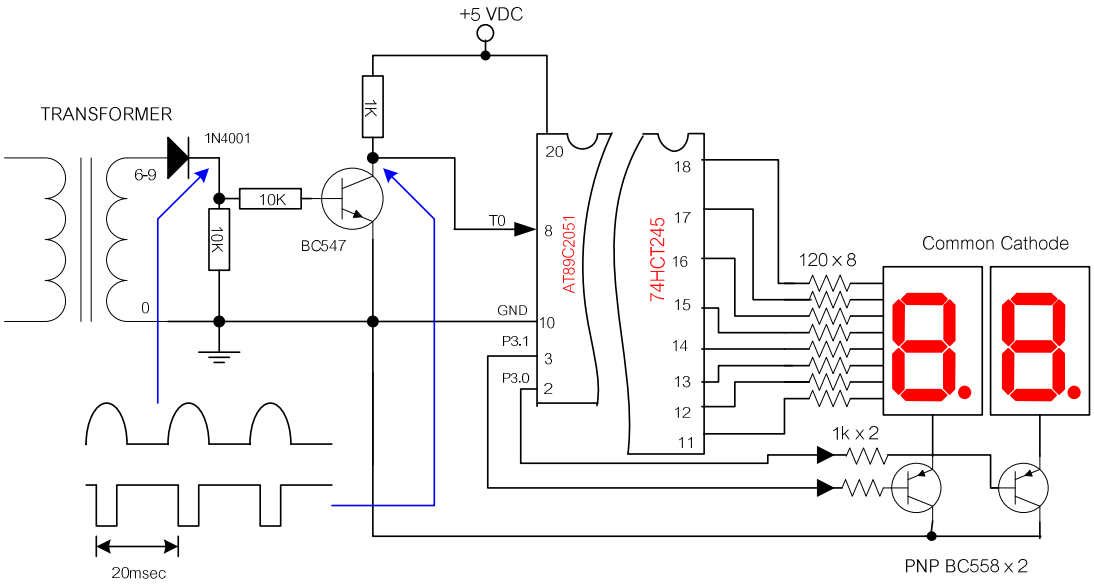
```
;*****************************************************************
;****    การรับข้อมูลจาก Computer PC ผ่านพอร์ตอนุกรม: RS-232          ***
;*****************************************************************
                ORG   0000H
                AJMP  MAIN
                ORG   0023H
                AJMP  Rx_ISR
                ORG   0050H
MAIN:           MOV   P1, #00H             ; ให้สถานะของแอลอีดีที่ต่อกับพอร์ต P1 ให้ดับทุกดวง
                MOV   TMOD, #00100000B     ; ใช้งานไทเมอร์ 1 โหมด 2
                MOV   TL1, #0FDH           ; กำหนดค่าการนับในรีจิสเตอร์ TL1
                MOV   TH1, #0FDH           ; กำหนดค่าเริ่มต้นการนับในรีจิสเตอร์ TH1
                SETB  TR1                  ; เริ่มต้นการทำงานไทเมอร์1
                MOV   SCON, #01010000B     ; เลือกอนุกรมโหมด 1 REN = 1
                MOV   PCON, #00H           ; เลือก SMOD = 0
```

```
        MOV   IE,#90H          ; อนุญาตให้มีการอินเตอร์รัปต์พอร์ตอนุกรม
        SJMP  $                ; วนทำบรรทัดเดิม
Rx_ISR: CLR   RI               ; เคลียร์บิต RI เพื่อรับไบต์ต่อไป
        MOV   P1, SUBF         ; นำข้อมูลในรีจิสเตอร์ SUBF แสดงผลที่พอร์ต P1
        RETI                   ; ออกจากโปรแกรมย่อยการอินเตอร์รัปต์
        END
```

6. การประยุกต์ใช้งานอินเตอร์รัปต์ไทเมอร์เคาน์เตอร์

6.1 การทำงานของวงจร

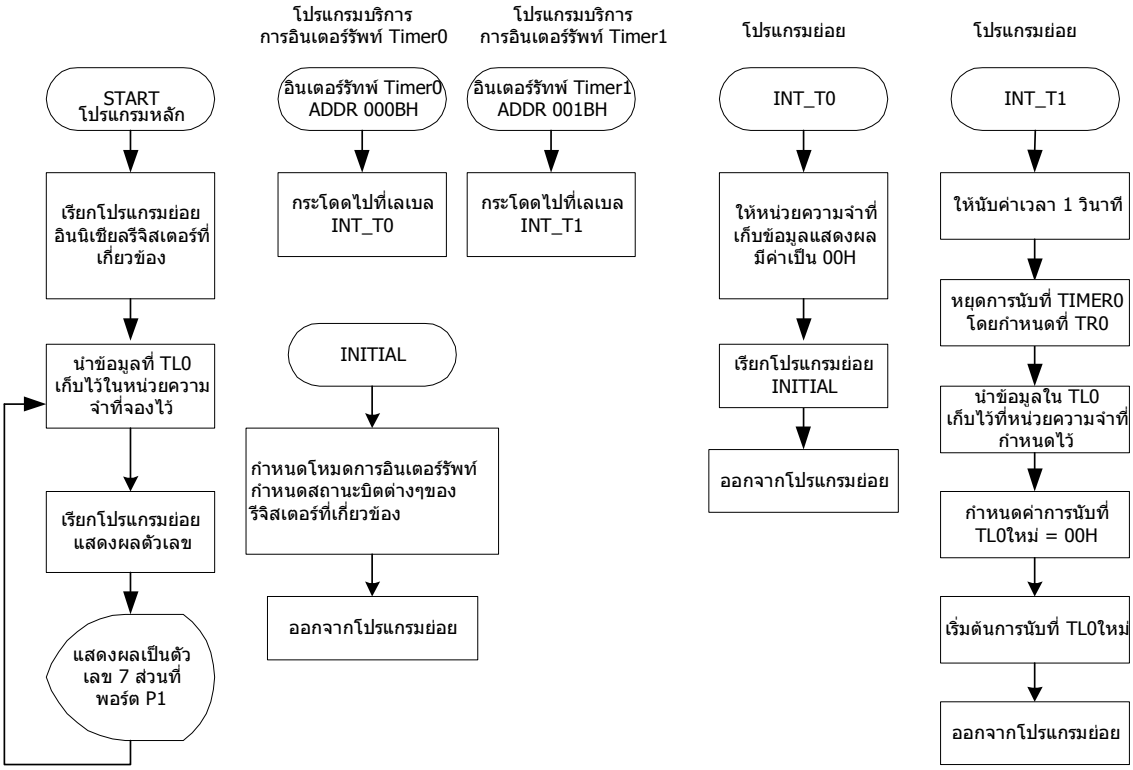
ใช้งานไทเมอร์และเคาน์เตอร์พร้อมกัน เป็นวงจรวัดค่าความถี่ของสัญญาณไฟฟ้ากระแสสลับที่ใช้ในบ้านขนาด 50 Hz โดยกำหนดให้ใช้งานไทมเมอร์ 1 เป็นตัวตั้งเวลาไว้ 1 วินาที และให้ทำงานในโหมด 2 เป็นแบบโหลดค่าใหม่อัตโนมัติ (Auto Reloads) ส่วนไทเมอร์ 0 กำหนดการใช้งานเป็นตัวนับค่า โดยให้รับสัญญาณจากภายนอก และให้ทำงานในโหมด 2 เช่นเดียวกัน วงจรใช้งานแสดงดังภาพที่ 9.15



ภาพที่ 9.15 แสดงวงจรวัดค่าความถี่ของไฟกระแสตรงในช่วงเวลา 1 วินาที

วงจรใช้สัญญาณซายน์เวฟ (Sine Wave) จากหม้อแปลง (Transformer) โดยทรานซิสเตอร์ทำหน้าที่จัดรูปแบบของสัญญาณจากซายน์เวฟให้เป็นสัญญาณแบบสี่เหลี่ยม แล้วนำมาเป็นอินพุตให้กับขา T0 (ขา 8) ของไมโครคอนโทรลเลอร์ โดยให้ Timer0 ทำหน้าที่นับค่าความถี่ที่มีค่า 50Hz หรือมีค่าเท่ากับ 20 msec (T=1/f) ส่วนการกำหนดเวลาในการนับสัญญาณรายคาบ จะกำหนดไว้ที่ 1 วินาที รีจิสเตอร์ TL0

เป็นข้อมูลของการนับ และจะนับสัญญาณรายคาบแต่ละครั้ง โดยเพิ่มค่าที่ข้อมูลในรีจิสเตอร์ TL0 ทีละ 1ค่า หลังจากครบเวลา 1 วินาที จะกำหนดให้รีจิสเตอร์ TL0 หยุดการนับ หลังจากนั้นนำข้อมูลใน รีจิสเตอร์ TL0 ที่นับได้ไปแปลงค่าเป็นเลขฐานสิบ และนำไปเปิดตารางข้อมูลโดยโปรแกรมย่อย Lookup Table ก่อนที่จะแสดงผลเป็นตัวเลข 7 ส่วนจำนวน 2 หลัก โดยขณะแสดงผลตัวเลข 7 ส่วนอยู่นั้น ในส่วนของรีจิสเตอร์ TL1 จะเริ่มทำการนับค่าเวลาใหม่ และ TL0 ก็จะเริ่มนับจำนวนสัญญาณที่ T0 ใหม่เช่นเดียวกัน พอครบ 1 วินาที จะนำข้อมูลในรีจิสเตอร์ TL0 มาแสดงผลอีก ถ้าหากมีการนับเกิน 256 ค่าหรือมีการเซตบิตแสดงสถานะค่านับเกิน TF0 (Overflow) จะกระโดดไปทำที่ตำแหน่งบริการอินเตอร์รัปต์ ดังนั้นที่ตำแหน่งแอดเดรส 000BH จึงเขียนคำสั่งป้อนค่าคงที่ 00H ให้กับหน่วยความจำตำแหน่งที่กำหนดไว้สำหรับการแสดงผลข้อมูลที่แอลอีดี 7 ส่วนเพื่อแสดงถึงการเกิดค่านับเกิน ขั้นตอนการเขียนโปรแกรมเป็นผังงานแสดงดังในภาพที่ 9.16



ภาพที่ 9.16 แสดงผังงานของโปรแกรมวัดค่าความถี่ไฟฟ้ากระแสสลับ 50 Hz

6.2 การเขียนโปรแกรม

```
; **          Frequency Counter
; **    TIMER/COUNTER: User TIMER0 = Counter Mode2 Ext (T0), TIMER1 = Timer Mode2
COUNT_200     EQU     20H
COUNT_10      EQU     21H
```

```
COUNT_2     EQU   22H
COUNT       EQU   23H
DIGIT1      BIT   P3.1
DIGIT0      BIT   P3.0
DATA:       DS    2
        ORG   0000H
        SJMP  START
        ORG   000BH         ; แอดเดรสเริ่มต้นโปรแกรมบริการการอินเตอร์รัพท์ TIMER 0
        SJMP  INT_T0        ; ให้กระโดดไปทำที่ตำแหน่งของแอดเดรสที่เลเบล INT_T0
        ORG   001BH         ; แอดเดรสเริ่มต้นโปรแกรมบริการการอินเตอร์รัพท์ TIMER 1
        SJMP  INT_T1        ; ให้กระโดดไปทำที่ตำแหน่งของแอดเดรสที่เลเบล INT_T1
        ORG   0030H         ; เริ่มต้นที่หน่วยความจำโปรแกรมแอดเดรส 0030H
;****  โปรแกรมหลัก  ******
START:      MOV   P3,#0FFH      ;กำหนดให้ P3=11111111B ทรานซิสเตอร์หยุดทำงานทั้ง 2 ตัว
            MOV   P1,#00H       ;กำหนดให้พอร์ต P1 มีค่าข้อมูลเท่ากับ 00H
MAIN:       ACALL INIT          ; เรียกโปรแกรมย่อยการกำหนดค่าเริ่มต้นให้กับรีจิสเตอร์ต่างๆ
LOOP:       ACALL DATA_DIS
            ACALL DISPLAY       ; เรียกโปรแกรมย่อยการแสดงผลข้อมูลที่ตัวเลข 7 ส่วน
            SJMP  LOOP          ; วนกลับไปทำที่เลเบล LOOP ใหม่
;****  ทำการกำหนดค่าเริ่มต้นในรีจิสเตอร์ ต่างๆ (Initial) ที่ใช้ในการบริการการอินเตอร์รัพท์
INIT:   MOV   COUNT_200,#200
        MOV   COUNT_10,#10
        MOV   COUNT_2,#2
        MOV   COUNT,#00H
        SETB  P3.2
        SETB  P3.4
        MOV   TMOD,#00101110B
        MOV   TL0,#00H
        MOV   TH0,#00H
        MOV   TL1,#-230
        MOV   TH1,#-230
        SETB  EA            ; อนุญาตให้มีการอินเตอร์รัพท์
```

```
        SETB   ET1              ; กำหนดให้บิต ET1 เป็นการอนุญาตให้มีการอินเตอร์รัพท์ที่ Timer0
        SETB   ET0              ; กำหนดให้บิต ET0 เป็นการอนุญาตให้มีการอินเตอร์รัพท์ที่ Timer0
        SETB   TR0              ; กำหนดให้บิต TR0 เป็นการเริ่มให้มีการนับที่ Timer0
        SETB   TR1              ; กำหนดให้บิต TR1 เป็นการเริ่มให้มีการนับค่าที่ Timer1
        RET
;****   โปรแกรมย่อยบริการอินเตอร์รัพท์ TIMER1 กำหนดให้เป็นตัวตั้งเวลา          ******
INT_T1:         DJNZ   COUNT_200,END_RETI
                MOV    COUNT_200,#200
                DJNZ   COUNT_10,END_RETI
                MOV    COUNT_10,#10
                DJNZ   COUNT_2,END_RETI
                MOV    COUNT_2,#02
                CLR    TR0              ; ให้ Timer0 หยุดทำการนับ
                MOV    COUNT,TL0
                MOV    TL0,#00H
                SETB   TR0              ; ให้ Timer0 เริ่มทำการนับ
END_RETI:       RETI
;****   โปรแกรมย่อยบริการอินเตอร์รัพท์ TIMER 0 หากมีการนับเกิน 256 ครั้ง        **
INT_T0:         MOV    COUNT,#00H
                RETI                    ; ออกจากโปรแกรมย่อย
;****   โปรแกรมย่อยเก็บค่าข้อมูลที่จะแสดงผลตำแหน่ง DATA และ DATA+1 ******
DATA_DIS:       MOV    DPTR,#TABLE
                MOV    B,#10
                MOV    A, COUNT
                DIV    AB
                MOVC   A,@A+DPTR
                MOV    DATA+1,A
                MOV    A,B
                MOVC   A,@A+DPTR
                MOV    DATA,A
                RET
;****   โปรแกรมย่อยโดยนำข้อมูลที่ตำแหน่ง DATA และ DATA+1 แสดงผลที่แอลอีดี 7 ส่วน ******
```

```
DISPLAY:    MOV   A,DATA+1
            MOV   P1,A
            CLR   DIGIT1
            ACALL DELAY_1
            SETB  DIGIT1
            MOV   A,DATA
            MOV   P1,A
            CLR   DIGIT0
            ACALL DELAY_1
            SETB  DIGIT0
            RET
;*** โปรแกรมย่อยการหน่วงเวลาเพื่อแสดงผลที่ตัวเลข 7 ส่วน 0-9  ***
DELAY_1:    MOV   R1,#200
            DJNZ  R1,$
            RET
TABLE:      DB          3FH, 06H, 5BH, 4FH, 66H
            DB          6DH, 7DH, 07H, 7FH, 6FH
            END
```

หลังจากกดที่สวิตซ์รีเซต แอลอีดี 7 ส่วน แสดงผลเป็นตัวเลข 50 ซึ่งเป็นค่าความถี่ของไฟบ้าน หากใช้ออสซิลโลสโคปวัดสัญญาณที่ขา T0 ของไอซี MCS-51 ได้ค่าความถี่ 50 Hz วัดคาบเวลาได้ 20 msec (T=1/f) แสดงดังภาพที่ 9.17

ภาพที่ 9.17 แสดงสัญญาณและคาบเวลาที่วัดได้โดยออสซิลโลสโคป

สรุป

การเขียนโปรแกรมคำสั่ง โดยตรวจสอบสัญญาณการเชื่อมต่อจากอุปกรณ์ภายนอก หรือสัญญาณ ที่เกิดขึ้นภายในตัวไอซีอยู่ตลอดเวลา และถึงแม้ว่าไม่มีสัญญาณใดๆเกิดขึ้น วิธีเขียนโปรแกรมแบบนี้เรียกว่า แบบโพลลิ่ง (Polling)

การอินเตอร์รัปต์ (Interrupt) หมายถึงการขัดจังหวะ ใช้งานเมื่อไอซี MCS-51 กำลังปฏิบัติตามคำสั่งในโปรแกรมหลักอยู่ หากมีสัญญาณการร้องขออินเตอร์รัปต์ ที่ส่งมาจากอุปกรณ์ภายนอก หรือสัญญาณที่มาจากภายในของตัวไอซี ๆ หยุดการปฏิบัติตามคำสั่งของโปรแกรมหลัก หลังจากนั้นจึงเริ่มปฏิบัติตามคำสั่งในโปรแกรมย่อยบริการอินเตอร์รัปต์ที่ได้เขียนกำหนดไว้ในส่วนของหน่วยความจำโปรแกรมจนเสร็จสิ้น จึงจะกลับมาทำคำสั่งที่โปรแกรมหลักต่ออีกครั้งหนึ่ง

ไอซี MCS-51 มีแหล่งสัญญาณการอินเตอร์รัปต์ ทั้งภายนอก และภายในประกอบไปด้วยสัญญาณอินเตอร์รัปต์ภายนอก 0 สัญญาณอินเตอร์รัปต์ ไทเมอร์ 0 สัญญาณอินเตอร์รัปต์ภายนอก 1 สัญญาณอินเตอร์รัปต์ไทเมอร์ 1 สัญญาณอินเตอร์รัปต์พอร์ตอนุกรม สัญญาณอินเตอร์รัปต์ ไทเมอร์ 2

รีจิสเตอร์ IE (Interrupt Enable Register) ทำหน้าที่ควบคุมการตอบรับต่อสัญญาณอินเตอร์รัปต์ โดยถ้ากำหนดให้บิตเป็นสถานะลอจิก “1” เป็นการตอบรับการอินเตอร์รัปต์ ถ้ากำหนดให้บิตเป็นสถานะลอจิก “0” ไม่ตอบรับการอินเตอร์รัปต์

รีจิสเตอร์ IP (Interrupt Priority Register) ทำหน้าที่จัดลำดับความสำคัญของการอินเตอร์รัปต์ในกรณีไม่มีการจัดลำดับความสำคัญของการอินเตอร์รัปต์ หรือจัดให้มีความสำคัญในระดับเดียวกันไอซี MCS-51 จัดให้มีลำดับความสำคัญของสัญญาณอินเตอร์รัปต์ ตามความสำคัญจากลำดับสูงไปลำดับต่ำ

รีจิสเตอร์ TCON (Timer Control Register) ทำหน้าที่ควบคุมการเลือกลักษณะสัญญาณอินเตอร์รัปต์จากภายนอก และสภาวะเริ่มต้นของรีจิสเตอร์ TCON หลังจากทำการรีเซต การเลือกใช้งานลักษณะสัญญาณการร้องขอการอินเตอร์รัปต์ ทำได้สองลักษณะโดยเลือกที่รีจีสเตอร์ TCON

โปรแกรมการตอบสนองอินเตอร์รัปต์ต้องทำคำสั่งเริ่มต้นของโปรแกรม ถ้ามีการใช้อินเตอร์รัปต์หากความยาวของโปรแกรมตอบสนองอินเตอร์รัปต์มีความยาวมากกว่า 8 ไบต์ ต้องนำไปเขียนเป็นโปรแกรมย่อยไว้ภายนอก แล้วใช้วิธีการเรียกโปรแกรมย่อยมาทำงาน เพราะระยะห่างตำแหน่งแอดเดรสของโปรแกรมตอบสนองอินเตอร์รัปต์แต่ละตัว มีระยะห่างเพียงแค่ 8 ไบต์เท่านั้น

การอินเตอร์รัปต์สามารถเลือกใช้สัญญาณที่แอคทีฟโดยการเปลี่ยนจากสถานะลอจิก "1" ไปลอจิก "0” (ขอบขาลง) หรือเลือกใช้สัญญาณอินเตอร์รัปต์ที่แอคทีฟโดยลอจิก "0" (Level)