

PROGRAMMING THE SERIAL COMMUNICATION INTERRUPT

SECTION 11.4: PROGRAMMING THE SERIAL COMMUNICATION INTERRUPT

In Chapter 10 we studied the serial communication of the 8051.

Todos los ejemplos en ese capítulo utilizaron el método de sondeo. En esta sección exploramos la comunicación en serie basada en interrupciones, que permite al 8051 hacer muchas cosas, además de enviar y recibir datos desde el puerto de comunicación en serie.

RI and TI flags and interrupts

Como puede recordar del Capítulo 10, TI (transfer interrupt) se activa cuando se transfiere el último bit de los datos enmarcados, el bit de parada, lo que indica que el registro SBUF está listo para transferir el siguiente byte.

RI (received interrupt), se genera cuando se recibe todo el cuadro de datos, incluido el bit de parada.

En otras palabras, cuando el registro SBUF tiene un byte, RI se pone en alto para indicar que el byte recibido debe ser recogido antes de que se pierda (overrun) por los nuevos datos seriales entrantes. En lo que respecta a la comunicación en serie, todos los conceptos anteriores se aplican por igual cuando se utiliza un sondeo o una interrupción.

La única diferencia está en cómo se atienden las necesidades de comunicación en serie.

En el método de sondeo, esperamos que se levante la bandera (TI o RI); Mientras esperamos no podemos hacer nada más.

En el método de interrupción, se nos notifica cuando el 8051 ha recibido un byte o está listo para enviar el siguiente byte; Podemos hacer otras cosas mientras se atienden las necesidades de comunicación en serie.

En el 8051 solo se reserva una interrupción para la comunicación en serie. Esta interrupción se utiliza para enviar y recibir datos. Si el bit de interrupción en el registro IE (IE.4) **está habilitado**, cuando se pone en alto RI o TI, el 8051 se interrumpe y salta a la ubicación de dirección de memoria 0023H para ejecutar el ISR. En ese ISR debemos examinar los indicadores de TI y RI para ver cuál causó la interrupción y responder en consecuencia.

See Example 11-8.

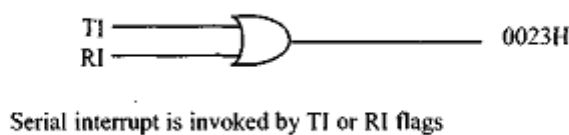


Figura 11-7. Interrupción única para TI y RI

Use of serial COM in the 8051

En la gran mayoría de las aplicaciones, la interrupción en serie se utiliza principalmente para recibir datos y nunca se utiliza para enviar datos en serie. Esto es como recibir una llamada telefónica, donde necesitamos un timbre para ser notificado. Si necesitamos hacer una llamada telefónica, hay otras formas de recordarnos y, por lo tanto, no hay necesidad de llamar.

Sin embargo, al recibir la llamada telefónica, debemos responder de inmediato sin importar lo que estemos haciendo o perderemos la llamada. Del mismo modo, utilizamos la interrupción en serie para recibir los datos entrantes para que no se pierda.

Look at Example 11-9.

Example 11-8 Escriba un programa en el que el 8051 lea los datos de P1 y los escriba en P2 de forma continua mientras le da una copia del mismo al puerto COM en serie para que se transfiera en serie.
Asuma que tiene un XTAL = 11.0592 MHz. Ajuste la velocidad en baudios a 9600.

Solution:

```

ORG 0
LJMP MAIN
ORG 23H
LJMP SERIAL      ;jump to serial interrupt ISR
ORG 30H
MAIN:  MOV  P1,#0FFH      ;make P1 an input port
      MOV  TMOD,#20H      ;timer 1, mode 2(auto-reload)
      MOV  TH1,#0FDH      ;9600 baud rate
      MOV  SCON,#50H      ;8-bit, 1 stop, REN enabled
      MOV  IE,#10010000B  ;enable serial interrupt
      SETB TR1            ;start timer 1
BACK:  MOV  A,P1           ;read data from port 1
      MOV  SBUF,A         ;give a copy to SBUF
      MOV  P2,A           ;send it to P2
      SJMP BACK           ;stay in loop indefinitely
;
;-----Serial Port ISR
ORG 100H
SERIAL: JB  TI,TRANS      ;jump if TI is high
      MOV  A,SBUF         ;otherwise due to receive
      CLR  RI             ;clear RI since CPU does not
      RETI               ;return from ISR
TRANS:  CLR  TI           ;clear TI since CPU does not
      RETI               ;return from ISR
      END

```

En el programa anterior, observe el papel de las señales TI y RI. En el momento en que se escribe un byte en SBUF, se enmarca y transfiere en serie.

Como resultado, cuando se transfiere el último bit (bit de parada), se pone en alto la TI, lo que hace que se invoque la interrupción en serie ya que el bit correspondiente en el registro de IE es alto.

En el ISR en serie, se chequea tanto TI como RI, ya que ambos podrían haber invocado la interrupción. En otras palabras, **solo hay una interrupción para transmitir y recibir**.

Clearing RI and TI before the RETI instruction

Observe en el Example 11-9 que la última instrucción antes de RETI es la eliminación de los indicadores RI o TI.

Esto es necesario ya que solo hay una interrupción para recibir y transmitir, y el 8051 no sabe quién lo generó; por lo tanto, es el trabajo del ISR borrar la bandera. En contraste a esto con las interrupciones externas y de temporizador donde está el trabajo del 8051 para borrar las banderas de interrupción.

By contrast,

Example 11-9 Escriba un programa en el que el 8051 obtenga datos de P1 y los envíe a P2 continuamente, mientras que los datos entrantes del puerto serie se envían a P0. Asuma que tiene un XTAL = 11.0592 MHz. Ajuste la velocidad en baudios a 9600.

Solution:

```
ORG 0
LJMP MAIN
ORG 23H
LJMP SERIAL ;jump to serial ISR
ORG 30H
MAIN: MOV P1,#0FFH ;make P1 an input port
      MOV TMOD,#20H ;timer 1, mode 2(auto-reload)
      MOV TH1,#0FDH ;9600 baud rate
      MOV SCON,#50H ;8-bit,1 stop, REN enabled
      MOV IE,#10010000B ;enable serial interrupt
      SETB TR1 ;start Timer 1
BACK: MOV A,P1 ;read data from port 1
      MOV P2,A ;send it to P2
      SJMP BACK ;stay in loop indefinitely
;-----SERIAL PORT ISR
ORG 100H
SERIAL: JB TI,TRANS ;jump if TI is high
        MOV A,SBUF ;otherwise due to receive
        MOV P0,A ;send incoming data to P0
        CLR RI ;clear RI since CPU doesn't
        RETI ;return from ISR
TRANS: CLR TI ;clear TI since CPU doesn't
        RETI ;return from ISR
END
```

En la comunicación en serie, la RI (o TI) deberá ser reseteado mediante instrucciones de software como "CLR TI" y "CLR RI" en el ISR. Vea el Example 11-10.

Observe que las dos últimas instrucciones del ISR están borrando la bandera, seguidas de RETI.

Table 11-2: Interrupt Flag Bits for the 8051/52

Interrupt	Flag	SFR Register Bit
External 0	IE0	TCON.1
External 1	IE1	TCON.3
Timer 0	TF0	TCON.5
Timer 1	TF1	TCON.7
Serial port	TI	SCON.1
Timer 2	TF2	T2CON.7 (AT89C52)
Timer 2	EXF2	T2CON.6 (AT89C52)

Antes de terminar esta sección, observe la lista de todas las banderas o flags de interrupción que se muestran en la Tabla 11-2. Mientras que el Registro TCON contiene cuatro de los indicadores de interrupción, en el 8051 el Registro SCON tiene los indicadores RI y TI.

Example 11-10 Escriba un programa usando interrupciones para hacer lo siguiente:

1. Recibir datos en serie y enviarlos a P0,
2. Tener el puerto P1 leído y transmitido en serie, y una copia dada al P2,
3. Haga que el temporizador 0 genere una onda cuadrada de 5 kHz de frecuencia en P0.1 (Puerto 0 bit 1). Asuma que tiene un XTAL = 11.0592 MHz. Ajuste la velocidad en baudios a 4800.

Solution:

```
ORG 0
LJMP MAIN
ORG 000BH          ;ISR for Timer 0
CPL P0.1           ;toggle P0.1
RETI               ;return from ISR
ORG 23H
LJMP SERIAL        ;jump to serial int. ISR
ORG 30H
MAIN: MOV P1,#0FFH  ;make P1 an input port
      MOV TMOD,#22H ;timer 0&1,mode 2, auto-reload
      MOV TH1,#0F6H ;4800 baud rate
      MOV SCON,#50H ;8-bit, 1 stop, REN enabled
      MOV TH0,#-92  ;for 5 KHz wave
      MOV IE,#10010010B ;enable serial, timer 0 int.
      SETB TR1      ;start timer 1
      SETB TR0      ;start timer 0
BACK: MOV A,P1      ;read data from port 1
      MOV SBUF,A    ;give a copy to SBUF
      MOV P2,A      ;write it to P2
      SJMP BACK     ;stay in loop indefinitely

;-----SERIAL PORT ISR
ORG 100H
SERIAL: JB TI,TRANS ;jump if TI is high
        MOV A,SBUF  ;otherwise due to received
        MOV P0,A    ;send serial data to P0
        CLR RI      ;clear RI since CPU does not
        RETI        ;return from ISR
TRANS:  CLR TI      ;clear TI since CPU does not
        RETI        ;return from ISR
END
```

Referencia FB-POST

<https://www.facebook.com/groups/167241170005427/permalink/952800378116165/>

<http://what-when-how.com/8051-microcontroller/programming-the-serial-communication-interrupt/>

** v0626 p04