Using up to 5 external interrupts on 80C51 family microcontrollers

AN420

80C51 family microcontrollers are equipped with up to two inputs which may be used as general-purpose interrupts. A typical device provides a total of 5 interrupt sources. Timer 0 and Timer 1 generate vectored interrupts, as does the Serial Port. Applications that require more than two externally signaled vectored interrupts, and do not use one or more of the counters or the serial port, can be configured to use these facilities for additional external interrupt inputs.

This note describes a method to configure the timer/counters and the serial port for use as interrupt inputs (see Figure 1). Minimum response time is a goal for this configuration.

Another popular method to implement extra interrupt inputs is to poll under software control a port pin configured as an input. This method is necessary when the on-chip peripherals are in use. Applications where this approach is recommended are ones in which the processor spends more than half of the time executing a "wait loop," or a short code sequence which jumps or branches back on itself without performing any functions. In this case, the instructions that will check the state of input used as an interrupt source are inserted into this sequence. Consequently, this input is ignored when other routines are being executed. This input may have to be latched externally, or the processor may miss the signal while executing other routines.

Dedicated interrupt inputs that vector the processor to individual service routines (as the two general-purpose interrupt inputs work) do not have the drawbacks of the method described above.

COUNTER/TIMER CONFIGURATION

Timers 0 and 1 are placed in mode 2, which configures the timer/register as an 8-bit counter with automatic reload. The counter and reload register are loaded with FF hexadecimal which is stored in TH1 and TL1 or TH0 and TL0.

To prepare one of the timers for this kind of operation, a number of control bits have to be set up. The following is a list of these bits and their values:

In TMOD:	**In TCON:**	**In IE:**
GATE = 0	TRi = 1	ETi = 1
C/T = 1		EA = 1
M1 = 1		
M0 = 0		

Where "i" is the timer number being used as the external interrupt. The TMOD value would be 66 hexadecimal if both timers are being used as external interrupt sources, x6 hex for timer 0, and 6x hex for timer 1. The interrupt priority may also be set in the IP register.

A falling edge on the corresponding Timer 0 or Timer 1 input (T0 or T1) will cause the counter to overflow and generate a timer interrupt. The counter will be automatically loaded with another FF from the reload register, so the interrupt can occur again as soon as the interrupt service routine completes. Counter/Timer operation is described in detail elsewhere in this manual.

SERIAL PORT CONFIGURATION

The serial port can be placed in mode 2, which is a 9-bit UART with the baud rate derived from the oscillator. The external interrupt is signaled through this port on the RxD receive data pin. Reception is initiated by a detected 1-to-0 transition at RxD. The signal must stay at 0 for at least five-eighths of a bit period for this level to be recognized. Refer to the description of baud rates to determine the length of a bit period at the oscillator frequency selected for the application. The input signal should remain low for at least one bit period and for not more than 9 bit periods.

To prepare the serial port for use as an external interrupt, the following bits must be set up:

In SCON:

SM0 = 1
SM1 = 0
SM2 = 0
REN = 1

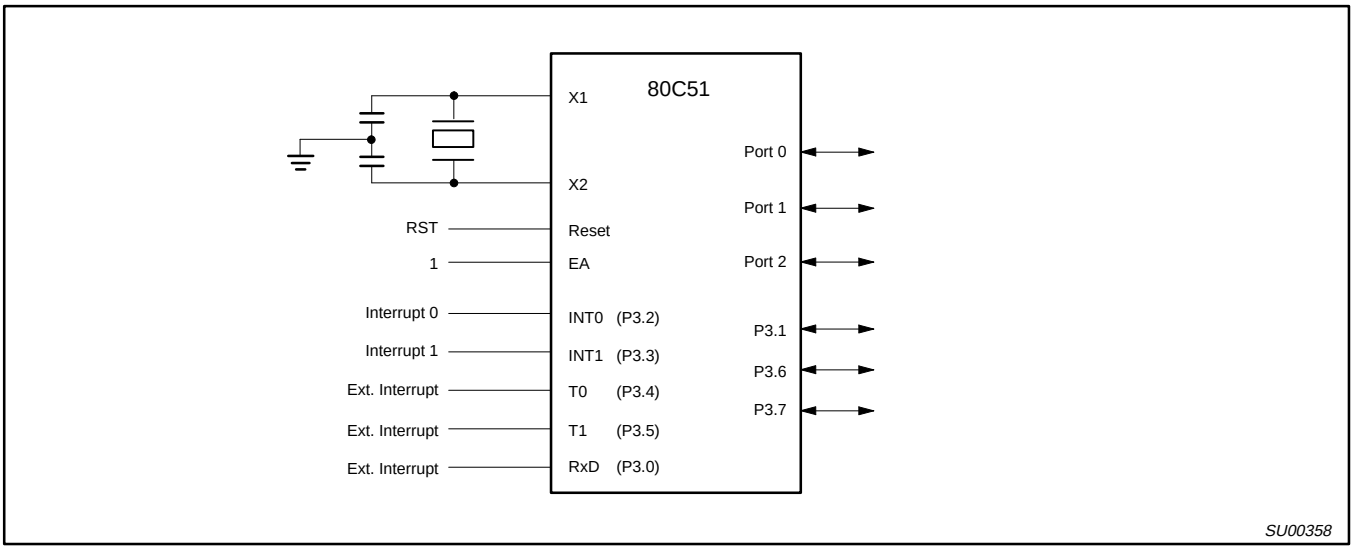

Figure 1. 80C51 Five Interrupt Configuration

The Serial Port Interrupt is then used as a general-purpose interrupt. The contents of receive buffer should be ignored, and will subsequently be overwritten during the next interrupt.

Note that the response time for this input will be slower than for the Counter/Timer inputs. This is due to the fact that the RI is generated after the eighth serial data bit time after the falling edge on RxD.

```
;   Demonstration program for five external interrupts.

$MOD51
$TITLE (Five Vectored External Interrupts)

;   Interrupt Jump Table
            ORG     OH              ;Reset
            AJMP    Setup

            ORG     3H              ;External interrupt 0.
            RETI                    ;(not implemented in this demo)

            ORG     0BH             ;Timer 0 interrupt.
            AJMP    Tim0

            ORG     13H             ;External interrupt 1.
            RETI                    ;(not implemented in this demo)

            ORG     1BH             ;Timer 1 interrupt.
            AJMP    Tim1

            ORG     23H             ;Serial port interrupt.
            AJMP    Serial

;   Begin setup code
Setup       MOV     SP,#7FH         ;Initialize the stack pointer.

;   Configure both timers
            MOV     TMOD,#66H       ;Put both counters into mode 2.
            MOV     A,#0FFH
            MOV     TL0,A           ;Load FF hex into both counters
            MOV     TH0,A
            MOV     TL1,A
            MOV     TH1,A
            SETB    ET0             ;Enable Timer 0 interrupt.
            SETB    ET1             ;Enable Timer 1 interrupt.
            SETB    TR0             ;Enable Timer 0 to run.
            SETB    TR1             ;Enable Timer 1 to run.

;   Configure the serial port
            SETB    ES              ;Enable serial port interrupt.
            MOV     SCON,#90H       ;Put the serial port in mode 2.
            SETB    EA              ;Enable interrupt system.

Wait:       NOP                     ;Wait for an interrupt.
            JMP     Wait

Serial:     NOP                     ;Serial interrupt service routine.
            CLR     RI              ;Clear receiver interrupt flag.
            RETI

Tim0:       NOP                     ;Timer 0 interrupt service routine.
            RETI

Tim1:       NOP                     ;Timer 0 interrupt service routine.
            RETI

            END
```