How to Get a Second Asynchronous Serial Interface on a 80C51 Microcontroller Family

Description

The 80C51 family has only one asynchronous serial interface.
However some users would like to have a low cost solution to get two in their applications.

This solution exists and is described in this application note.
The goal of this note is to present a very low cost software solution to realise this second asynchronous serial interface.

Features

- No external hardware added ;
- Full duplex ;
- Dissymetrical baud rate in reception and in transmission available ;
- 1200 bauds limitation of the internal serial interface (hardware).

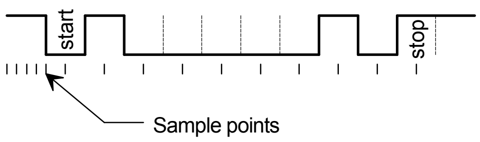
Resources used

A time reference with interrupt capability is needed and it can be TIMER 1 even if it is already used as baud rate generator for the internal serial interface. In this case a 32 time speed transmission is obtained on TIMER 1 overflow (TIMER 1 is in mode 2 : 8-bit auto-reload, and serial interface is in mode 1 : 8-bit variable baud rate).

Only two I/O pins are needed : one for RxD and one for TxD (for instance P1.0 and P1.1). Few bytes of memory are used and finally a portion of the CPU time is used to serve TIMER 1 interrupt. Three functions : initialisation, transmission and reception, are allowed to use this serial interface.

Method

Transmission of the character 01000001b 'A'



Receiver part :
On each TIMER 1 overflow interrupt, RxD input is sampled. Start of transmission is recognised by a transition of 1 to 0 on this pin. So a second sample is made half a bit later to be sure that it is a start bit. Then sampling is made in the middle of the received bits, nine times to get the 8 data bits. The stop bit must have level 1.

Transmitter part :
The operation of the transmitter is nearly the same as for the receiver : start bit is written on TxD output followed by the 8 data bits and the stop bit and so on. Time of bit writing is calculated by counting timer interrupts.

Efficiency

Number of machine cycles spent in interrupt sub-routine :

- Minimum : 10 cycles ;
- Maximum : 49 cycles (transmission and reception) ;

The measures hereafter have been done with a 11.059MHz crystal, and same baud rate in emission and in reception, and a hardware serial baud rate of 1200 bauds.

Percentage of CPU usage :

- 41.7% if there is no traffic ;
- 50% with continuous transmission or reception, and 1200 baud rate ;
- 57.4% with continuous transmission and reception, and 1200 baud rate ;
- 68.5% with continuous transmission and reception, and 9600 baud rate.

The hardware serial baud rate is limited to 1200 bauds, increasing it induces an increase of TIMER 1 interrupts frequency, and so an increase of percentage of CPU usage.

Demonstration Program

The demonstration program (listed in the following pages) allows transmission on P1.1 of all characters received on P1.0 without checking receive error.

The function TXD_S starts transmission of the character placed in accumulator when the transmitter is ready.
The function RXD_S waits for reception of a character and return it in accumulator.

Additional Information

For additional information on Microcontrollers, and Ordering Information, please refer to the following TEMIC/Matra MHS datasheets :

- 80C31/80C51
- 80C32/80C52
- 80C154/83C154
- 83C154D

The information contained herein is subject to change without notice. No responsibility is assumed by MATRA MHS SA for using this publication and/or circuits described herein : nor for any possible infringements of patents or other rights of third parties which may result

Program Listing

```
$TITLE (Software serial interface)
; Software serial interface with programmable speed

$NOMOD51
$INCLUDE (reg51.inc)

NAME      UARTSOFT

; Constant definition
RxD1      EQU   P1.0
TxD1      EQU   P1.1

; Segment definition
PROG      SEGMENT       CODE
VAR1      SEGMENT       DATA
BITVAR    SEGMENT       BIT
STACK     SEGMENT       IDATA

          RSEG  STACK
          DS    10H              ; 16 Bytes Stack

; vectors definition
          CSEG  AT 0000H         ; Reset vector
          jmp   MAIN

          CSEG  AT 001BH         ; Timer 1 vector
          jmp   ITIM1


; bits definition
          RSEG BITVAR
TXRDY:    DBIT 1            ; 1 if transmitter ready
RXRDY:    DBIT 1            ; 1 if receiver ready
RXERR     DBIT 1            ; 1 if receiver error
INCOM:    DBIT 1            ; 1 if character received


; vars definition
          RSEG VAR1
          ; Receiver
RXSPD:    DS 1  ; speed in reception
RXCH:     DS 1  ; character in reception
RXCNT:    DS 1  ; internal counter
RXSTAT:   DS 1  ; receiver status
RXCH2:    DS 1  ; last character received

          ; Transmitter
TXSPD:    DS 1  ; speed in transmission
TXCH:     DS 1  ; character in transmission
TXCNT:    DS 1  ; internal counter
TXSTAT:   DS 1  ; transmitter status
```

```
; software serial interface demonstration program
; characters received on P1.0 are transmitted on P1.1

          RSEG PROG

; Main routine
MAIN:     mov   SP,#STACK-1
          lcall SEINIT           ; interfaces init.
LOOP:     lcall RXD_S
          lcall TXD_S
          sjmp  LOOP


; Initialize serial interfaces
; desired speed is 32 for 1200 bauds, 4 for 9600 bauds
; Oscillator frequency = 11.059 MHz

SEINIT:   mov   TCON,#40H        ; Timer 1 enabled
          mov   TMOD,#20H        ; C/T = 0, mode =
2
          mov   TH1,#0E8H        ; 1200 bauds
          mov   SCON,#52H        ; serial port mode 1

          mov   A,#32            ; 1200 bauds
          mov   RXSPD,A
          mov   TXSPD,A
          setb  PT1              ; high priority It.
          setb  TXRDY            ; transmitter ready
          setb  RXRDY            ; receiver ready
          clr   RXERR            ; no error
          mov   IE,#10001000B    ; It. timer 1 enabled
          ret

; Transmission of a character on TxD1
TXD_S:    jnb   TXRDY,TXD_S
          mov   C,P
          mov   ACC.7,C          ; set parity
          mov   TXCH,A           ; character to send
          mov   A,TXSPD          ; 1 bit duration
          rr    A                ; 1/2 bit duration
          mov   TXCNT,A          ; set counter
          mov   TXSTAT,#0        ; init. status
          clr   TXRDY            ; start transmission
          ret

; Reading of the received character on RxD1
RXD_S:    jnb   INCOM,RXD_S
          mov   A,RXCH2          ; char. received
          clr   INCOM            ; char. readed
          ret
```

```
; Interrupt routine
ITIM1:   jnb    RXRDY,RX1
         ; receiver not busy
         jb     RxD1,TRANS   ; start bit ?
         clr    RXRDY
         push   ACC
         mov    A,RXSPD      ; 1 bit duration
         rr     A            ; 1/2 bit duration
         mov    RXCNT,A      ; load counter
         mov    RXSTAT,#0    ; init. status
         pop    ACC
         sjmp   TRANS
RX1:     djnz   RXCNT,TRANS ; sample point ?
         push   ACC
         push   PSW
         mov    A,RXSTAT
         jnz    RX3
         jb     RxD1,ERRFRM ; start bit OK (0) ?
RX2:     inc    RXSTAT
         mov    RXCNT,RXSPD
         sjmp   RX5
ERRFRM: setb    RXRDY
         setb   RXERR        ; receiver error
         sjmp   RX5
RX3:     cjne   A,#9,$+3     ; 8 bits + stop bit
         jnc    RX4
         mov    C,RxD1       ; bit sampling
         mov    A,RXCH
         rrc    A
         mov    RXCH,A
         sjmp   RX2
RX4:     jnb    RxD1,ERRFRM ; stop bit OK (1) ?
         mov    RXCH2,RXCH
         setb   RXRDY
         setb   INCOM        ; 1 char. received
RX5:     pop    PSW
         pop    ACC

TRANS:   ; transmission part
         jb     TXRDY,TX5
         djnz   TXCNT,TX5    ; sample point ?
         push   ACC
         push   PSW
         mov    A,TXSTAT
         jnz    TX1          ; start ?
         clr    TxD1         ; set start bit
         mov    TXCNT,TXSPD
         inc    TXSTAT
         sjmp   TX4
TX1:     cjne   A,#9,$+3     ; 8 bits + stop bit
         jnc    TX2
         mov    A,TXCH
         rrc    A            ; bit to send in carry
         mov    TXCH,A
         mov    TxD1,C       ; transmission of bit
         mov    TXCNT,TXSPD; init. counter
         inc    TXSTAT
         sjmp   TX4
TX2:     cjne   A,#10,TX3    ; end of character ?
         setb   TXRDY
         sjmp   TX4
TX3:     setb   TxD1         ; set stop bit
         mov    TXCNT,TXSPD
         inc    TXSTAT
TX4:     pop    PSW
         pop    ACC
TX5:     reti

         END
```