Universidad Nacional de Ingeniería
Facultad de Ciencias

Arquitectura de computadores

Puerto Serie del 8051

Prof.: Lic. César Martín Cruz S.
ccruz@uni.edu.pe

2012 - II

Control del Puerto Serie (SCON)

El puerto serie contenido en la familia del MCS-51, es un puerto “FULL DUPLEX”, lo cual significa que puede transmitir y recibir datos simultáneamente. El receptor contiene un almacén “Buffer”, que le permite empezar a recibir un segundo dato sin necesidad de que el primero haya sido completamente leído del registro Buffer. Sin embargo, si el primer byte permanece sin ser leído hasta el final de la recepción del segundo dato, éste se perderá.

El dato de la recepción y la transmisión se encuentra en el registro **SBUF** del SFR (Registros de Funciones Especiales, anteriormente descrito).
El puerto serie puede ser operado en 4 modos diferentes, que son especificados mediante la escritura en los bits **SM0** y **SM1** del Registro de Control del Puerto Serie **SCON**, que se describe a continuación.

SCON

El registro de control del Puerto Serie es direccionable por bit, para activar o desactivar cada una de sus banderas.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Registro de Control Puerto Serie

SÍMBOLO	BIT	FUNCIÓN
SM0	SCON.7	Especifica el modo de operación del puerto serie.
SM1	SCON.6	Especifica el modo de operación del puerto serie.
SM2	SCON.5	Habilita la comunicación del tipo “multiprocesador” utilizado en los modos 2 y 3. En estos modos, si SM2=1, **RI** no es activado si el noveno dato recibido (RB8) es 0. En modo 1, **RI** no es activado si no se recibe un BIT de stop. En el modo 0, SM2 será 0.
REN	SCON.4	Se establece la recepción serie si REN=1 (por software), cuando REN=0 se desactiva la recepción (por software).
TB8	SCON.3	Almacena el 9no. Bit que será transmitido en los modos 2 y 3.
RB8	SCON.2	En modos 2 y 3, el 9no. BIT recibido es de datos. En el modo 1, si SM2=0, RB8 es el BIT de stop el que fue recibido. En el modo 0, RB8 no es usado.

Registro de Control Puerto…..

SÍMBOLO	BIT	FUNCIÓN
TI	SCON.1	Bandera de interrupción para transmitir. Se habilita por hardware al final del 8vo. BIT en el modo 0, ó al principio del BIT de stop en los otros modos. Debe ser deshabilitado por software.
RI	SCON.0	Bandera de interrupción para recibir. Se habilita por hardware al final del 8vo.BIT en el modo 0, ó a la mitad del camino del BIT de stop en los otros modos (ver excepción SM2). Debe ser deshabilitado por software.

Para poder seleccionar el modo en este registro, se encuentran los bits 6 y 7 (**SM1** y **SM0** respectivamente) y la configuración y descripción es como sigue:

SM0	SM1	MODO	Descripción	Baud Rate
0	0	0	Registro Desplazamiento	Fosc/12
0	1	1	8 Bit UART*	Variable
1	0	2	9 Bit UART	Fosc/64 ó Fosc/32
1	1	3	9 Bit UART	Variable

***UART: Universal Asincronic Receive – Transmit.**

El timer 1 como generador del “Baud Rate” en la comunicación serie

El **Timer 1** es usado para generar la frecuencia de Transmisión/Recepción de datos en serie, cuando el puerto es programado para trabajar en el modo 1 ó 3. La frecuencia de comunicación es obtenida mediante la ecuación siguiente:

$$\text{BAUD RATE} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{FREC. DEL OSCILADOR}}{12 \times [\,256 - (\text{TH1})\,]}$$

SMOD es el bit 7 del registro de control de potencia **PCON**.
En este caso el **Timer 1** actúa en modo 2, es decir, en modo recargable, el valor de conteo se encuentra fijo en el registro TH1, el cual se recarga cada vez que existe un overflow.

En la siguiente tabla se muestran los valores de TH1, para generar el Baud Rate, tomado en cuenta la frecuencia del oscilador o cristal.

Baud Rate	F osc	SMOD	VALOR DE TH1
19.2 Kbits/s	11.0592 MHz	1	FDH
9.6 Kbits/s	11.0592 MHz	0	FDH
4.8 Kbits/s	11.0592 MHz	0	FAH
2.4 Kbits/s	11.0592 MHz	0	F4H
1.2 Kbits/s	11.0592 MHz	0	E8H

Ejemplo 1:

Asumiendo un Fosc=11.0592 Mhz, desarrolle un programa que envíe a la PC vía comunicación serie los caracteres 1, 2, 3, 4 y 5 luego de recepcionar el carácter “B” .

Sol.

```
;Programa que envía los números 1, 2, 3, 4 y 5
        org 0000h
        lcall init          ;Inicializa el puerto serie y configura el baud rate
repite:
        lcall getchr        ;Espera a que se le envíe un carácter.
```

```
        cjne A,#42h,repite ;si A no es “B” va a repite si no continua.
        mov 30h,A
repite2:
        mov A,#31h
        lcall sndchr
        mov A,#32h
        lcall sndchr
        mov A,#33h
        lcall sndchr
        mov A,#34h
        lcall sndchr
        mov A,#35h
        lcall sndchr
        mov A,#0dh          ;retorno de línea
        lcall sndchr
        mov A,#0Ah          ;nueva fila
        lcall sndchr
        sjmp repite2
```

```
;=================================================================
;subrutina getchr
;esta subrutina lee un caracter desde el puerto serie y lo retorna
; en el acumulador
;=================================================================
    getchr:
        jnb RI,getchr       ;espera hasta que el carácter sea recibido
        mov A,SBUF          ;consigue carácter
        clr acc.7           ;el bit 7 se vuelve 0
        clr RI              ;vuelve 0 a RI
        ret

;=================================================================
;subrutina sndchr
;esta subrutina toma el caracter en el acumulador y lo envia
;por el puerto serie
;=================================================================
    sndchr:
        clr TI
```

```
        mov SBUF,A          ;inicia transmisión
    txloop:
        jnb TI,txloop       ;espera hasta que el caracter es enviado. Si TI es "1"
                            ;contínua
        ret

;=====================================================
;Subrutina init
;esta subrutina inicializa el puerto serie en modo 1 para 8
; bits de datos usando el timer 1 como el generador de baud rate.
;una frecuencia de reloj de 11.0592 Mhz es asumido
;=====================================================
    init:
        mov PCON,#80H   ;se pone "1" a SMOD
        mov TMOD,#20H   ;timer 1 en modo 2
        mov TCON,#41H   ;inicia el timer 1
        mov TH1,#0FDH   ;valor de autorecarga
        mov SCON,#50H   ;se selecciona modo 1 y se activa la recepción serie
        ret
        end
```

Ejemplo 2:

Asumiendo un Fosc=11.0592 Mhz, desarrolle un programa que envíe una cadena de caracteres a la PC vía comunicación serie.

Sol.

```
;Programa que envia una cadena de caracteres
;por el puerto serie
        org 000h
        lcall init      ;Inicializa el puerto serie y configura el baud rate
repite:
        lcall print
         db "Arquitectura de Computadores"
        db 0Dh
        db 0Ah
        db "Prof. Martín Cruz"
        db 0h
        sjmp $
```

```
;===============================================
;subrutina print
;toma la cadena inmediatamente que sigue a "lcall" y lo envia
;por el puerto serie. La cadena debe terminar con un nulo.
;Esta subrutina retornará a la instrucción
;que sigue inmediatamente a la cadena.
;===============================================
print:
        pop dph         ;coloca dirección de retorno en dptr
        pop dpl
prtstr:
        clr A
        movc A,@A+DPTR
        cjne A,#0h,mchrok
        sjmp prtdone
mchrok:
        lcall sndchr
```

```
            inc DPTR
            sjmp PRTSTR
prtdone:
            mov A,#1h
            jmp @A+DPTR

;=============================================================
;subrutina sndchr
;esta subrutina toma el caracter en el acumulador y lo envia
;por el puerto serie
;=============================================================
      sndchr:
            clr TI
            mov SBUF,A          ;inicia transmisión
      txloop:
            jnb TI,txloop       ;espera hasta que el caracter es enviado. Si TI es "1"
                                ;contínua
            ret
```

```
;===============================================
;Subrutina init
;esta subrutina inicializa el puerto serie en modo 1 para 8
;bits de datos usando el timer 1 como el generador de baud rate.
;Una frecuencia de reloj de 11.0592 Mhz es asumido
;===============================================
    init:
        mov PCON,#80H    ;se pone "1" a SMOD
        mov TMOD,#20H    ;timer 1 en modo 2
        mov TCON,#41H    ;inicia el timer 1
        mov TH1,#0FDH    ;valor de autorecarga
        mov SCON,#50H    ;se selecciona modo 1 y se activa la
                         ;recepción serie
        ret
        end
```

Ejemplo 3:

Hacer un programa que convierta el contenido del acumulador en dos números ascii hexadecimales. El resultado es retornado en el acumulador y R2.

Sol.

```
        org 0000h
        mov A,#0BCh
        lcall binasc
        sjmp $
;=========================================
;Subrutina binasc
;Toma el contenido del acumulador y convierte
;en dos números ascii hexadecimales.
;El resultado es retornado en el acumulador y R2
;=========================================
binasc:
        mov R2,A        ;se mueve a R2 para salvar el valor de A
```

```
        anl A,#0Fh      ;convierte el dígito menos significativo
        add A,#0F6h
        jnc noadj1
        add A,#07h
noadj1:
        add A,#3Ah      ;lo hace ascii
        xch A,R2        ;pone el resultado en R2
        swap A          ;convierte el digito más significativo
        anl A,#0Fh
        add A,#0F6h     ;lo ajusta
        jnc noadj2
        add A,#07h
noadj2:
        add A,#3Ah      ;lo hace ascii
        ret
```

Ejemplo 4:

Hacer un programa que envíe por comunicación serie el contenido del acumulador como dos dígitos ascii hexadecimales.

Sol.:

```
        org 0000h
        lcall init
        mov A,#0BCh
        lcall prthex
        sjmp $
;=========================================
;Subrutina prthex
;Esta rutina toma el contenido del acumulador y envía
; vía serie como 2 dígitos ascii hexadecimal.
;=========================================
prthex:
        lcall binasc
        lcall sndchr
        mov A,R2
        lcall sndchr
        ret
```