

### 3.2 MÓDULO DE INTERFACE DE COMUNICACIONES SERIALES ASÍNCRONAS – SCI COMUNICACIÓN “HALF DÚPLEX” CON “HYPERTERMINAL” (SOLO GP32)

Preparado por: Rangel Alvarado

Estudiante Graduando de Lic. en Ing. Electromecánica

Universidad Tecnológica de Panamá

Panamá, Panamá

“e-mail”: [issaiass@cwpanama.net](mailto:issaiass@cwpanama.net)

“web site”: <http://www.geocities.com/issaiass/>

#### ÍNDICE

3.2.1	<i>Introducción</i>	335
3.2.2	<i>Formato de Datos Caracteres y Errores</i>	336
3.2.3	<i>Registros de Estado y Control del Módulo de Comunicaciones Seriales</i>	337
3.2.4	<i>Diagrama de Flujo</i>	340
3.2.5	<i>Código</i>	341
3.2.6	<i>Aplicación</i>	347
3.2.7	<i>Conclusión</i>	349
3.2.8	<i>Referencias</i>	349
3.2.9	<i>Problemas Propuestos</i>	349

#### 3.2.1 Introducción

---

El módulo de Interface de Comunicación Serial (SCI) permite tener velocidades de transmisión altas ya sea entre unidades microcontroladoras o bien utilizando la PC como intermediario. Como la comunicación es asíncrona, la recepción puede darse en cualquier momento.

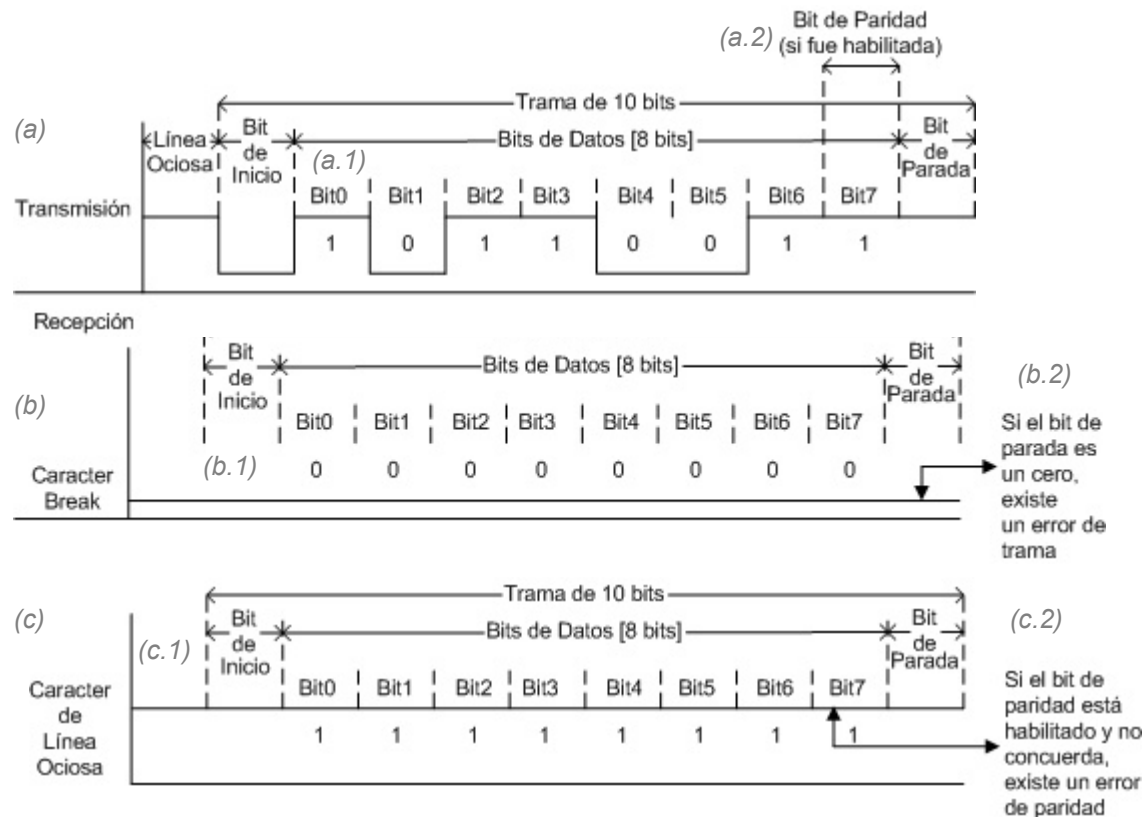
A menudo se necesita una interfaz tipo gráfica que despliegue un menú y ejecutar acciones dependientes de los caracteres teclados, esta aplicación explota ese recurso. Sin embargo, la mayoría de los desarrolladores utilizan entornos visuales avanzados como Visual Basic, Labview o Visual C++ para una aplicación, pues se puede armar una interface más cómoda para su uso.

El documento informativo abarca aspectos básicos de comunicaciones seriales y es meramente informativo, este consta de:

- Formato de Transmisión: Narra generalidades sobre el protocolo de transmisión y errores comunes en los paquetes de información.
- Configuración del Módulo SCI: Aprender a configurar el módulo para comunicaciones seriales en modos de transmisor y receptor.
- “HYPERTERMINAL”: Configurar el modulo terminal de la PC para comunicarnos y encender/apagar un LED.

### 3.2.2 Formato de Datos, Caracteres y Errores

La transmisión de datos seriales (si son 8 bits de datos) consta de una trama de diez (10) bits, de los cuales el primero es el bit de inicio (en bajo) y el último es el bit de parada (en alto). El ancho o duración de un bit, se denomina baudios, p.e., 9600 baudios (bits/seg) corresponden a un tiempo aproximado de  $1/9600 = 0.104$  milisegundos.



*Figura 187. Transmisión de Datos Seriales y Errores. (a) Transmisión. (a.1) Bits de Datos. La información llega, en este caso, en paquetes de ocho bits. (a.2) Bit de Paridad. Si el SCI habilita la función de paridad, el último bit representa el resultado de la paridad. (b) Carácter "Break". (b.1) Ruptura. No posee ni bit de inicio, datos o paridad, siempre mantiene la línea en bajo. (b.2) Error de Trama. Si el bit de parada es en bajo, existe un error de trama. (c) Caracter de línea osciosa. (c.1) Línea osciosa. La línea en espera es representada por una agrupación de unos y no posee inicio, data o bit de parada. (c.2) Error de Paridad. Asumiendo una paridad impar (ver nota inferior), existe un error de paridad y la data es corrupta.*

*Nota:* Los métodos de verificación de errores sirven para detectar si el paquete de información es corrupto, o válido. El método de paridad consiste en:

**Paridad Par:** La suma de los bits de datos más el de paridad, es un número par

**Paridad Impar:** La suma de los bits de datos más el de paridad, es un número impar.

Las comunicaciones seriales pueden ser "**simplex**", si la información viaja en una dirección; "**half-dúplex**" si la información se transmite y luego se recibe, "**full-dúplex**" si la información viaja en ambas direcciones al mismo tiempo.

Finalmente, otros errores de comunicación son el error de sobremanejo ("overrun") que ocurre si la data no es leída a tiempo; y el error de ruido ("noise") el cual sucede, p.e., si el mismo bit al ser leído tres (3) veces, alguna de sus lecturas es diferente.

Un ejemplo común de comunicaciones como 9600,8,N,1 representa: 9600 baudios, 8 bits de datos, N o sin bit de paridad y 1 bit de parada.

### 3.2.3 Registros de Estado y Control del Módulo de Comunicaciones Seriales

#### 3.2.3.1 Registros de Control del SCI

En la comunicación serial los pines **PTE0/TxD** y **PTE1/RxD** son utilizados como **transmisor** y **receptor** respectivamente.

**Nota:** Los bits enmarcados son los que serán utilizados en la aplicación

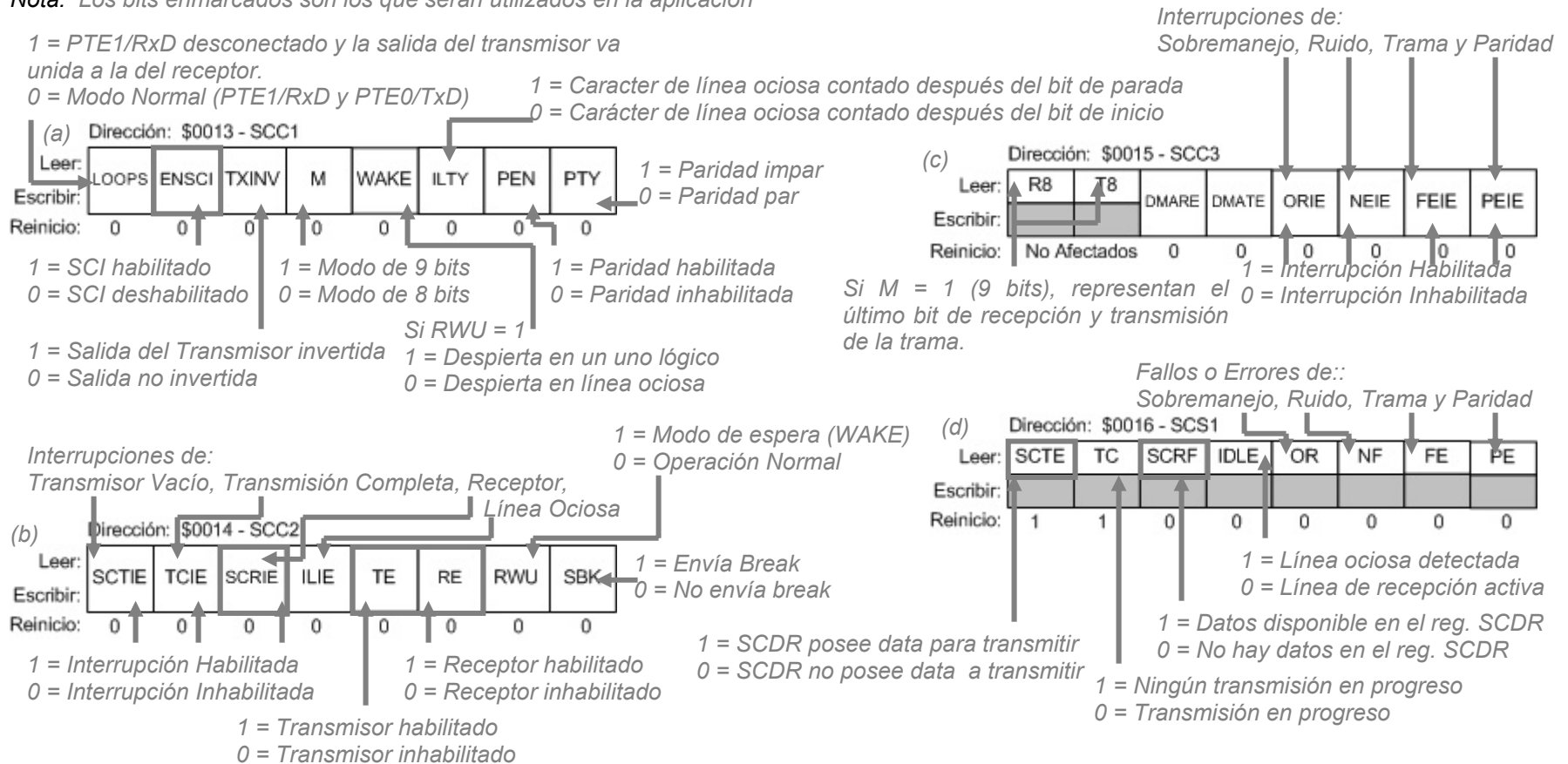
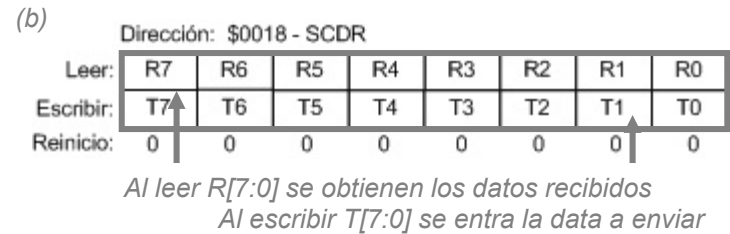
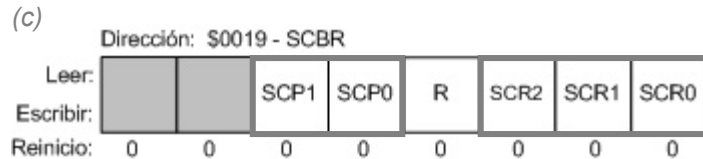


Figura 188. Registros de Control y Estado del SCI. (a) Registro de Control 1 del SCI. (b) Registro de Control 2 del SCI. (c) Registro de Control 3 del SCI. (d) Registro de Estado 1 del SCI.



Ver tabla 72, y ecuación 17

Tabla 72. Prescalar para el Cálculo del Baudio



SCP[1:0]	Divisor Prescalar (PD)
00	1
01	3
10	4
11	13

Ecuación 17. Cálculo del Baudio

$$baudios = \frac{\text{Reloj del SCI}}{PD \times 2^{6+SCR[2:0]_{10}}}$$

Figura 189. Registros de Control y Estado del SCI (continuación).  
 (a) Registro de Estado 2 del SCI. (b) Registro de Datos del SCI.  
 (c) Registro de Taza del Baudio del SCI. **Nota: SCR de 0 a 7.**

Reloj del SCI = fbus o CGMCLK, ver NT0022, registro de configuraciones 2.  
 PD = Prescalar Divisor para cálculo de la rata del baudio, ver Tabla 72.  
 SCR[2:0] = Selector de la rata del baudio; varía desde 0 hasta 7.

### 3.2.3.2 Vectores de Interrupción del Módulo SCI

El módulo SCI genera tres tipos de interrupciones diferentes; interrupciones de transmisión, recepción y errores.

Tabla 73. Interrupciones que procesa el Módulo SCI

Bandera	Máscara	Dirección	Vector (Dirección)
SCTE   TC	SCTIE   TCIE	\$FFE2	Vector de Transmisión (Alto)
		\$FFE3	Vector de Transmisión (Bajo)
SCRIF   IDLE	SCRIF   ILIE	\$FFE4	Vector de Recepción (Alto)
		\$FFE5	Vector de Recepción (Bajo)
OR   NE   FE   PE	ORIE   NEIE   FEIE   PEIE	\$FFE6	Vector de Error (Alto)
		\$FFE7	Vector de Error (Bajo)

**Nota:** Para generar interrupciones, cada bandera y su respectiva máscara deben estar habilitadas.

### 3.2.3.3 Configuración del Módulo SCI

Configure el módulo SCI para generar una tasa de 9600 baudios, 8 bits de datos, sin paridad y que la frecuencia del reloj provenga del CGMXCLK; habilite el transmisor y las interrupciones del receptor. Nota: para cálculo del baudio, ver ecuación 17.

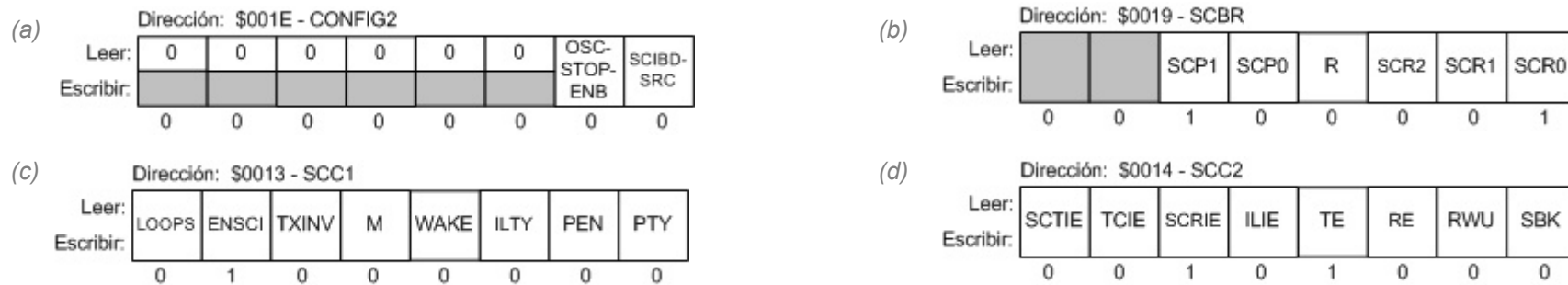


Figura 190. Configuración del Módulo SCI. (a) Registro de Configuraciones 2. SCIBDSRC = 0, CGMXCLK es el reloj del SCI. (b) Registro de Taza del Baudio. Configura el SCI a 9600 baudios. (c) Registro de Control 1 del SCI. Habilita el SCI, sin paridad. (d) Registro de Control 2 del SCI. Habilita las interrupciones del receptor y habilita el transmisor.

**Nota:** En la siguiente sección la aplicación requiere de un circuito o integrado que permitan la comunicación serial como MAX232 ó puede obtenerlo mediante la conexión del pin Ser/Com, que se puede extraer de las tarjetas de desarrollo JK3/JL3 según la tabla 59 de la NT0107.



### 3.2.5 Código

---

```
=====
; FILE:      NT1002 - SCI - 28 07 04.asm
; PURPOSE:   Configura el microcontrolador GP32 para generar una comunica-
;            ción bidireccional estilo half-dúplex y comunicarse con la PC
;            vía HYPERTERMINAL.  Despliega un menú de elección:
;            1 - Para encender el LED y 2 - Para apagar el LED
;
;
;Nota:
;            1 - Observar la rutina de inicialización del SCI.
;            2 - Inicializar la configuración en HYPERTERMINAL para
;            9600,8,n,1.
;Referencias:
;            MC68HC908GP32, MC68HC08GP32 Technical Data
;            http://www.freescale.com/files/microcontrollers/doc/
;            data_sheet/MC68HC908GP32.pdf
;            Págs. 238 @ 276 - Módulo de Interface de Comunicación Serial
;
; LANGUAGE   : IN-LINE ASSEMBLER
;-----
; HISTORY
; DD MM YY
; 28 07 04 Created.
; 28 07 04 Last Modification.
;-----
;Pasos para iniciar el SCI.
;            1 - Configurar el reloj del SCI (CONFIG2_SCIBDSRC = ?)
;            2 - Seleccionar la velocidad en baudios (SCBR)
;            3 - Habilitar el SCI (ENSCI = 1)
;            4 - Habilitar interrupciones del TxD, RxD o errores
;            5 - Habilitar el Transmisor, Receptor o ambos.
;Nota: Antes de cargar datos en el SCI, se debe leer el registro SCS1 y
;            luego escribir la data, repetir la misma acción para cada byte a
;            enviar o recibir.
;-----
```

```

;=====
; Igualdades Numéricas, números de bits y
; configuración de registros
;=====
PTA6      equ 6T      ; Puerto A, Bit 6
DDRA6     equ 6T      ; Registro de Direccionamiento A, Bit 6
LVI5OR3   equ %00001000 ; CONFIG1, Bit de Operación del LVI, LVI @
; 5V
COPD      equ %00000001 ; CONFIG1, Bit de Inhabilitación de Watchdog,
; BIT 0 ON
SCP1      equ %00100000 ; SCBR, Rata Prescalar del Baudio del SCI
SCP1_0    equ SCP1    ; Rata Prescalar del Baudio del SCI, SCP = 4
SCR0      equ %00000001 ; SCBR, Selección del Baudio del SCI, Bit 0
; ON
SCR2_0    equ SCR0    ; Selección del Baudio del SCI, SCR = 2
ENSCI     equ %01000000 ; SCC1, Bit Habilitador del SCI, Bit 6 ON
SCTE      equ 7T      ; SCC1, Bit Habilitador de Interrupciones del
; Transmisor, Bit 6 ON
NULL      equ 0T      ; Caracter nulo o Cero
SCRIE     equ %00100000 ; SCC2, Bit Habilitador de Interrupciones del
; Receptor, Bit 7 ON
TE        equ %00001000 ; SCC2, Bit Habilitador del Transmisor, Bit 3
; ON
RE        equ %00000100 ; SCC2, Bit Habilitador del Receptor, Bit 2 ON
HT        equ $09      ; Tabulado
LF        equ $0A      ; Enter
CR        equ $0D      ; Retorno de Carro

;=====
;
; I/O Registers
;=====
PORTA     equ $0000    ; Dirección, Puerto 0
DDRA      equ $0004    ; Registro de Direccionamiento, Puerto 0

;=====
;
; Serial Communications Interface
;=====
SCC1      equ $0013    ; Registro de Control 1 del SCI
SCC2      equ $0014    ; Registro de Control 2 del SCI
SCC3      equ $0015    ; Registro de Control 3 del SCI
SCS1      equ $0016    ; Registro de Estado 1 del SCI
SCS2      equ $0017    ; Registro de Estado 2 del SCI
SCDR      equ $0018    ; Registro de Datos del SCI
SCBR      equ $0019    ; Registro de Rata de Baudio

;=====
;
; Configuration
;=====
CONFIG1   equ $001F    ; Vector de configuración 1

```



```

;=====
;                               Flash Memory
;=====
FLASH_START equ $8000          ; Puntero - Mem.FLASH

;=====
;                               Vectores de Usuario
;=====
SCIRXH      equ $FFE4          ; Puntero del Vector de Recepción del SCI
;                               ; (Alto)
RESET_VEC   equ $FFFE          ; Puntero del RESET

;=====
; OBJETIVO: Inicio de Codif. del Ensamblador
;                               en Memoria FLASH.
;=====
org FLASH_START                ; Inicio Mem. FLASH

;=====
; OBJETIVO: Inicializar el SCI para comunicar
;                               el microcontrolador por la PC via
;                               HYPERTERMINAL y desplegar un menú
;                               que permita entradas para encener
;                               o apagar un LED.
;=====
START
    rsp                ; Inic.Stack = $00ff
    clra               ; Borra A
    clrx               ; Borra X
    jsr INIT_CONFIG    ; Inicializa el módulo de Configuración
    jsr INIT_PORTS     ; Inicializa Puertos
    jsr INIT_SCI       ; Inicializa Módulo de Comunicación Serial
    cli                ; Habilita interrupciones globales
    ldhx #MESSAGE1     ; Puntero del Primer Mensaje
    jsr PUTLINE        ; Envía Mensaje
    ldhx #MESSAGE2     ; Puntero del Segundo Mensaje
    jsr PUTLINE        ; Envía Mensaje
    ldhx #MESSAGE3     ; Puntero del Tercer Mensaje
    jsr PUTLINE        ; Envía Mensaje
    ldhx #MESSAGE4     ; Puntero del Cuarto Mensaje
    jsr PUTLINE        ; Envía Mensaje
    mov #{SCRIE|RE}, SCC2 ; Habilita interrupciones del receptor,
;                               ; y receptor.

ESPERA
    wait               ; modo de bajo consumo
    bra ESPERA         ; Salta al modo de bajo consumo

```

```

;=====
; INIT_CONFIG: Inicializa registro de configuración.
; OBJETIVO:   LVI a 5 V
;             Watchdog inhabilitado.
; ENTRADA:   Ninguna
; SALIDA:    Ninguna
; REGISTROS AFECTADOS: CONFIG1
;=====
INIT_CONFIG
    mov #{LVI5OR3|COPD},CONFIG1           ; LVI a 5V, Watchdog inhabilitado
                                           ; Nota: CONFIG2_SCIBRSC = 0
                                           ; Reloj del SCI es el cristal
    rts                                   ; retorna

;=====
; INIT_PORTS: Inicializa variables y registros.
; OBJETIVO:   Inicializa los registros de direccionamiento.
;             PORTA6 = LOW
;             PORTA6 = OUTPUT
; ENTRADA:   Ninguna
; SALIDA:    Ninguna
; REGISTROS AFECTADOS: DDRA, PTA
;=====
INIT_PORTS
    bclr PTA6,PORTA                        ; Puerto A 6 en bajo
    bset DDRA6,DDRA                       ; Puerto A 6 es salida
    rts                                   ; retorna

;=====
; INIT_SCI:   Inicializa el módulo de comunicaciones
;             seriales asíncrona
; OBJETIVO:   Inicializa a 9600 baudios, 8
;             bits de datos, (N) sin bit de
;             paridad, 1 bit de parada.
; ENTRADA:   Ninguna
; SALIDA:    Ninguna
; REGISTROS AFECTADOS:  SCBR, SCC1, SCC2, SCC3
;                 SCS1.
;=====
INIT_SCI
    mov #{SCP1_0|SCR2_0},SCBR             ; Configura SCI a 9600,8,N,1
    mov #ENSCI,SCC1                       ; Habilita el SCI
    mov #{SCRIE|TE},SCC2                 ; Habilita interrupciones del receptor,
                                           ; habilita transmisor.
    mov #NULL,SCC3                       ; Borra bit 8 (no utilizado)
    lda SCS1                             ; Borra las banderas SCTE y TC del
                                           ; registro SCS1
    rts                                   ; Retorna

```

```
=====
;
; PUTC:      Impone un Caracter por serial
; OBJETIVO:  Impone un caracter solo si se
;            envió el caracter anterior
; ENTRADA:   ACCA posee la data a enviar
; SALIDA:    ACCA posee la data a enviar
; REGISTROS AFECTADOS: SCS1, SCDR
;
=====
PUTCH
    brclr SCTE,SCS1,*           ; Espera a que se envíe el caracter anterior
    sta SCDR                   ; Almacena en el registro serial
    rts                        ; Retorna
```

```
=====
;
; PUTLINE:   Impone una línea por serial
; OBJETIVO:  Impone una línea de caracte-
;            res por serial.
; ENTRADA:   H:X, Dirección del Mensaje
;            a enviar.
; SALIDA:    H:X, Última Dirección del Ca-
;            racter del mensaje enviado.
; REGISTROS AFECTADOS: SCS1, SCDR
;
=====
PUTLINE
    lda ,x                      ; Carga primer caracter del mensaje
    cmp #NULL                   ; ¿Es cero?
    beq LINE_END               ; SI, terminó el mensaje
    jsr PUTC                    ; NO, Verifica si se puede imponer caracter
    aix #1                     ; SI, incrementa puntero de mensaje
    bra PUTLINE                ; NO, impone otro caracter
LINE_END
    rts                        ; Retorna
```

```

=====
; SCIRXL: Interrupción de recepción
; OBJETIVO: Recibe el caracter y decide
;           si enciende o apaga el LED
; ENTRADA: Ninguna
; SALIDA: Mensajes por serial enviados,
;          enciende/apaga el LED.
; REGISTROS AFECTADOS: ACCA, SCC2, H:X, PORTA,
=====
SCIRXL
    lda SCS1                ; Reconoce la interrupción
    lda SCDR                ; Lee el caracter enviado
    mov #{SCRIE|TE},SCC2    ; Habilita interrupciones del receptor,
                          ; habilita transmisor.

    cmp #'1'                ; ¿Es 1?
    bne NEXT                ; NO, salta a ver si es 2
    ldhx #MESSAGE5          ; SI, Carga Puntero del Mensaje 5
    jsr PUTLINE             ; Envía
    bset PTA6,PORTA         ; Enciende
    bra RX_OUT              ; Salir
NEXT  cmp #'2'                ; ¿Es 2?
    bne RX_OUT              ; NO, salir
    ldhx #MESSAGE6          ; SI, Carga Puntero del Mensaje 6
    jsr PUTLINE             ; Envía
    bclr PTA6,PORTA        ; Apaga
RX_OUT
    mov #{SCRIE|RE},SCC2    ; Habilita interrupciones del receptor,
                          ; habilita receptor.
    rti                    ; retorna

```

```

=====
; MESSAGEN: Tabla de Mensajes a Enviar
; OBJETIVO: La etiqueta contiene la direc
;           ción inicial del mensaje, HT
;           representa el tabulado, LF el
;           alimentador de línea (Enter)
;           CR el retorno de carro y 0 es
;           el final de cada línea.
; ENTRADA: Ninguna
; SALIDA: Ninguna
; Nota:
;   Los caracteres enviados por seriales
;   se envían en ASCII, para mayor informa-
;   ción de la tabla ASCII, ver las referen-
;   cias de la nota técnica.
=====
MESSAGE1 db 'Programa de Comunicaciones Seriales',LF,CR,0
MESSAGE2 db 'Presione:',LF,CR,0
MESSAGE3 db HT,'1 - Para encender el LED',LF,CR,0
MESSAGE4 db HT,'2 - Para apagar el LED',LF,CR,0
MESSAGE5 db CR,' - LED Encendido ',0
MESSAGE6 db CR,' - LED Apagado ',0

```

```

;=====
; OBJETIVO: Inicializa el Vector de Reset
;           Arranque del programa en la memo-
;           ria Flash y búsqueda de la inte-
;           rrupción del módulo SCI-receptor
;=====
;==== Vector del SCI =====
;           org SCIRXH           ; Origen de las Interrupciones del SCI
;           dw SCIRXL           ; Interrupción del Receptor
;==== Vector de Reinicio de Sistema =====
;           org RESET_VEC      ; Puntero Vec - RESET
;           dw START           ; al darse reset salta a Start
    
```

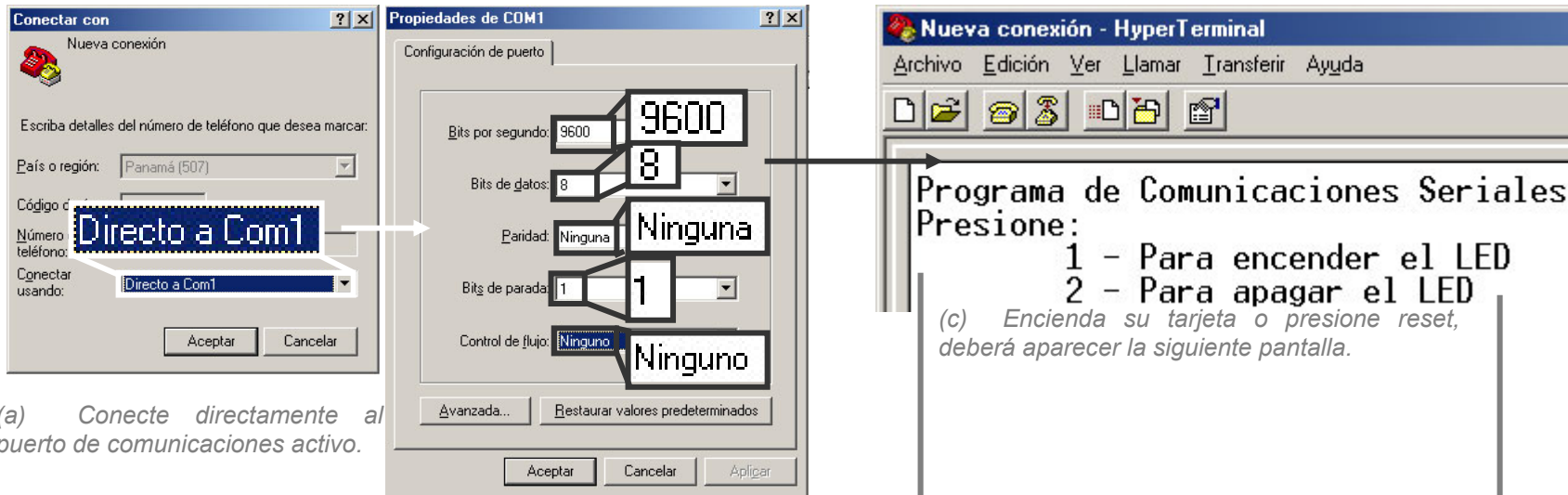
Listado 22. NT1002 – SCI. El programa principal inicializa el Módulo de Comunicaciones Seriales a 9600 baudios y despliega un menú en el cual se puede decidir si prender o apagar el LED.

### 3.2.6 Aplicación

Para la aplicación necesita tener instalado “hyperterminal”, si no lo posee, favor seguir los siguientes pasos para completar su instalación.



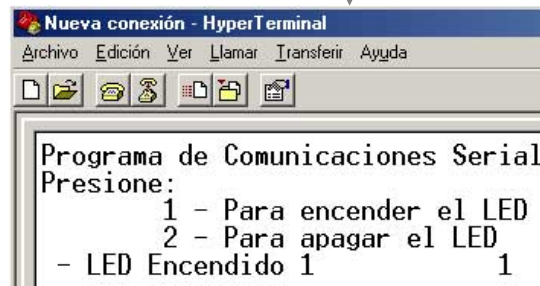
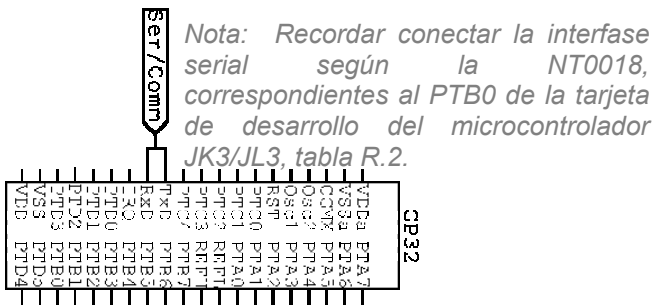
Figura 192. Instalación de “HYPERTERMINAL”. (a) Agregar o Quitar Programas. Inicie en Panel de Control el icono de agregar programas. (b) Instalación de Windows. Elija la pestaña de instalación. (c) Comunicaciones. En el menú, seleccione la opción comunicaciones. (d) “HyperTerminal”. Entre todas las selecciones, elija “hyperterminal” y presione aceptar, espere a que “windows” instale la aplicación. (e) Búsqueda de la aplicación. Navegue desde la barra de inicio y cargue “hyperterminal”. (f) Conexión. Utilice un nombre cualquiera como “mi conexión” para iniciar una conexión.



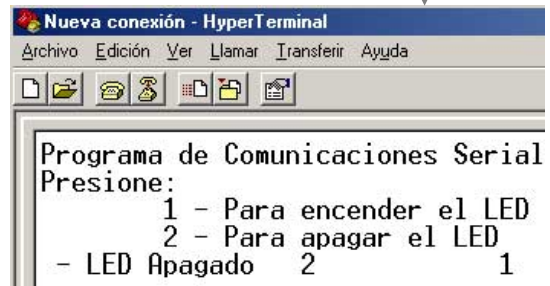
(a) Conecte directamente al puerto de comunicaciones activo.

(b) Seleccione las propiedades del puerto de comunicaciones, para 9600, 8, n, 1.

Programa de Comunicaciones Seriales  
 Presione:  
 1 - Para encender el LED  
 2 - Para apagar el LED  
 (c) Encienda su tarjeta o presione reset, deberá aparecer la siguiente pantalla.



(d) Si presiona el número 1, deberá aparecer el mensaje y encender el LED.



(e) Si presiona el número 2, deberá aparecer el mensaje y apagar el LED.

Figura 193. Instalación de "HYPERTERMINAL" (continuación). (a) Conexión al Puerto. Para iniciar la aplicación necesita definir el puerto al que tiene conectado el cable serial. (b) Propiedades de Comunicación. El canal de comunicación es configurado a 9600, 8, n, 1 y sin control de flujo. (c) Pantalla de la aplicación. Si se reinicia la tarjeta, desplegará el menú de la aplicación. (d) Opción 1. Al presionar, envía el mensaje del LED encendido y ejecuta la acción. (e) Opción 2. Al presionar, envía el mensaje del LED apagado y ejecuta la acción.

### **3.2.7 Conclusión**

---

El módulo de comunicación serial permite transmitir a altas velocidades de transmisión y recepción para comunicar microcontroladores o utilizar la PC para desplegar mensajes en pantalla y así formar un menú que permita al usuario interactuar de una manera amigable.

A manera de prueba, el módulo de comunicaciones “HyperTerminal” es utilizado para enviar y recibir caracteres, pero también se pueden utilizar otros ambientes más desarrollados como Visual C++, Visual Basic o Labview, para obtener una interfaz más amigable.

Del mismo modo ud. puede comunicar unidades microcontroladores entre maestro y esclavos para comunicaciones multipunto, tener múltiples maestros (uno a la vez) comandados por un identificador o supervisar vía serial un evento.

### **3.2.8 Referencias**

---

#### **3.2.8.1 Data Técnica del MC68HC908GP32, MC68HC08GP32**

(a) [http://www.freescale.com/files/microcontrollers/doc/data\\_sheet/MC68HC908GP32.pdf](http://www.freescale.com/files/microcontrollers/doc/data_sheet/MC68HC908GP32.pdf)

*Pág. 243 – Formato de Comunicación Serial, Sección 18.5.1*

*Pág. 244 – Diagrama de Bloques del Transmisor, figura 18-4*

*Pág. 245 – Configuración del Módulo SCI, Sección 18.5.2.2*

*Pág. 246 – Caracteres de Línea Ociosa y de Ruptura*

*Pág. 248 – Interrupciones del Transmisor, Sección 18.5.2.6*

*Pág. 249 – Diagrama de Bloques del Receptor, figura 18-5*

*Pág. 256 – Interrupciones del Receptor y de Errores*

*Págs. 260 a 275 – Información General de Registros*

#### **3.2.8.2 Manual de Referencia del CPU**

(a) [http://www.freescale.com/files/microcontrollers/doc/ref\\_manual/CPU08RM.pdf](http://www.freescale.com/files/microcontrollers/doc/ref_manual/CPU08RM.pdf)

#### **3.2.8.3 Información General del Código ASCII y Tabla**

(a) <http://www.abcdatos.com/utiles/ascii.html>

(b) <http://www.asciitable.com/>

#### **3.2.8.4 Página “web” sobre esta Nota Técnica**

(a) <http://www.geocities.com/issaiass/index.htm>

### **3.2.9 Problemas Propuestos**

---

3.2.9.1 Configure el SCI para comunicar a 19200 baudios, modo de 9 bits, y método de paridad par.

3.2.9.2 Realice un menú el cual despliegue en pantalla 3 opciones diferentes para manejar un LED y realizarle control; 1 – Encender, 2 – Apagar, 3 – Conmutar y 4 – Parpadear Indefinidamente.