

Facultad de ingeniería

Materia: Laboratorio de Microcomputadoras

Previo 7

Título: Puerto Serie SCI (Asíncrono)

Integrantes:

- Martínez Pérez Brian Erik - 319049792
- Nuñez Rodas Abraham Enrique - 114003546
- Vicenteño Maldonado Jennifer Michel - 317207251

Profesor: Moises Melendez Reyes

Grupo: 1

Fecha de Entrega: 6 de abril de 2025

Semestre: 2025-2



1. ¿Qué es la comunicación serial síncrona y asíncrona?

Comunicación serial síncrona: Los dispositivos comparten un reloj en común, los bits se transmiten en sincronía con ese reloj. Es más rápida y precisa, pero necesita una línea extra de reloj además de los datos.

Comunicación serial asíncrona: No se comparte un reloj, se usan bits especiales (inicio, parada) para marcar el comienzo y fin de los datos. Es más simple, pero menos precisa.

2. ¿Qué es la comunicación serial full duplex?

Es un tipo de comunicación donde los datos pueden transmitirse en ambas direcciones al mismo tiempo. En UART, puedes enviar datos por TX mientras recibes por RX simultáneamente.

3. ¿A qué se refiere el término baudios y como se aplica a la comunicación serial?

Baudios o baud rate es el número de señales por segundo que se transmiten en un canal de comunicación. Si configuras un UART a 9600 baudios, estás enviando 9600 bits por segundo.

4. ¿Qué son los bits de datos, inicio, paro y paridad?

Bits de datos: Son los bits que representan la información real. Pueden ser 7 u 8 normalmente.

Bit de inicio: Indica el comienzo de una transmisión de datos en comunicación asíncrona.

Bit de paro: Indica el fin de un paquete de datos. Puede ser 1 o 2 bits en alto (lógico 1).

Bit de paridad: Sirve para detectar errores. Puede ser par o impar dependiendo de si la cantidad de unos debe ser par o impar.

5. Escriba una tabla con velocidades estandarizadas para comunicación serial asíncrona.

En la comunicación serial asíncrona, se utilizan velocidades de transmisión estandarizadas, conocidas como tasas de baudios (baudios), para asegurar la compatibilidad entre dispositivos. A continuación, se presenta una tabla con algunas de las velocidades más comunes:

Velocidad (baudios)	Tiempo para transmitir 100 bytes (en configuración 8-N-1)
300	3.33 segundos
600	1.67 segundos
1200	833.3 milisegundos
2400	416.7 milisegundos
4800	208.3 milisegundos
9600	104.2 milisegundos
19200	52.1 milisegundos
38400	26.0 milisegundos
57600	17.4 milisegundos
115200	8.7 milisegundos
230400	4.3 milisegundos
460800	2.2 milisegundos
921600	1.1 milisegundos

6. ¿Qué registros de RAM intervienen al utilizar el puerto serie asíncrono en el PIC16F877A?

En el microcontrolador PIC16F877A, la comunicación serial asíncrona se gestiona a través del módulo USART (Transmisor-Receptor Asíncrono y Síncrono Universal). Este módulo utiliza varios registros específicos para controlar y manejar la transmisión y recepción de datos. Los principales registros involucrados son:

- **TXSTA (Transmit Status and Control Register):** Este registro configura y controla el estado de la transmisión. Sus bits más relevantes incluyen:
 - **TXEN:** Habilita o deshabilita la transmisión.
 - **SYNC:** Selecciona entre modo síncrono o asíncrono.
 - **BRGH:** Selecciona alta o baja velocidad de baudios.
- **RCSTA (Receive Status and Control Register):** Controla el estado y la configuración de la recepción. Algunos de sus bits clave son:

- **TXSTA:** Configura el registro de estado y control de transmisión.

Para modo asíncrono, alta velocidad y habilitación de transmisión:

```
movlw    b'00100100' ; BRGH = 1, SYNC = 0, TXEN
= 1
```

```
movwf    TXSTA
```

- **RCSTA:** Configura el registro de estado y control de recepción.

Para habilitar el puerto serial y la recepción continua:

```
movlw    b'10010000' ; SPEN = 1, CREN = 1
```

```
movwf    RCSTA
```

2. Configuración de los pines del puerto:

Configura RC6 como salida (TX) y RC7 como entrada (RX):

```
bsf      STATUS, RP0 ; Cambia a banco de registros 1
```

```
bcf      TRISC, 6 ; RC6 como salida
```

```
bsf      TRISC, 7 ; RC7 como entrada
```

```
bcf      STATUS, RP0 ; Regresa a banco de
registros 0
```

3. Recepción de datos:

Para recibir un dato, espera a que el bit RCIF (Flag de Recepción) en el registro

PIR1 se establezca, indicando que hay un dato disponible:

`WaitForData:`

`btfss PIR1, RCIF ; ¿RCIF está en 1? (dato
recibido)`

`goto WaitForData ; No, sigue esperando`

`movf RCREG, W ; Sí, lee el dato de RCREG
al registro W ; Aquí se puede procesar el dato
recibido en W`

Este procedimiento establece la comunicación serial asíncrona entre la PC y el PIC16F877A, permitiendo la recepción de datos desde la PC al microcontrolador.

8. Describa brevemente el proceso para enviar un dato desde el microcontrolador PIC16F877A hacia una PC, en ensamblador.

Para enviar un dato desde el microcontrolador PIC16F877A hacia una PC utilizando lenguaje ensamblador, se deben seguir estos pasos:

1. Configuración de los registros del USART:

SPBRG: Establece la tasa de baudios. Por ejemplo, para una comunicación a 9600 baudios con un oscilador de 4 MHz, se carga el valor 25 en SPBRG:

`movlw d'25' ; Valor para 9600 baudios con Fosc = 4MHz
movwf SPBRG`

TXSTA: Configura el registro de estado y control de transmisión. Para modo asíncrono, alta velocidad y habilitación de transmisión:

```
movlw    b'00100100' ; BRGH = 1, SYNC = 0, TXEN = 1  
  
movwf    TXSTA
```

RCSTA: Habilita el puerto serial:

```
movlw    b'10000000' ; SPEN = 1  
  
movwf    RCSTA
```

Configuración de los pines del puerto:

Configura RC6 como salida (TX) y RC7 como entrada (RX):

```
bsf      STATUS, RP0 ; Cambia al banco de registros 1  
  
bcf      TRISC, 6    ; RC6 como salida  
  
bsf      TRISC, 7    ; RC7 como entrada  
  
bcf      STATUS, RP0 ; Regresa al banco de registros 0
```

2.

Envío de datos:

Para enviar un dato, se coloca el valor en el registro TXREG y se espera a que el bit TRMT (Transmit Shift Register Status bit) en TXSTA indique que la transmisión ha finalizado:

`movlw 'A' ; Carga el valor ASCII del carácter a enviar`

`movwf TXREG ; Lo coloca en el registro de transmisión`

WaitForTransmission:

`btfss TXSTA, TRMT ; ¿Se completó la transmisión?`

`goto WaitForTransmission ; No, sigue esperando`

Referencias

- del PIC, 2. 1-La Familia. (s/f). 2.- Descripción General del PIC16F877. Edu.ar, https://exa.unne.edu.ar/ingenieria/sistemas/public_html/Archi_pdf/HojaDatos/Microcontroladores/PIC16F877.pdf
- Manual de usuario Microchip PIC16F877A (280 páginas). (2025). Manual.cr. https://www.manual.cr/microchip/pic16f877a/manual?utm_
- Garbutt, M. (2003). *Asynchronous communications with the PICmicro® USART (Application Note AN774)*. Microchip Technology Inc. <https://www.microchip.com>
- Pelayo, R. (2017, August 15). *Serial (USART) Communication with PIC16F877A*. Microcontroller Tutorials. <https://www.teachmemicro.com/serial-usart-pic16f877a/>
- *Velocidades de Transmisión Serial (Baud Rate) Soportadas por NI-VISA*. (2024). Ni.com. https://www.ni.com/es/support/documentation/supplemental/18/serial-baud-rates-supported-by-ni-visa.html?srsId=AfmBOoofBmGl_Ci0A-sCU-efxY51VXbuhhBXg-MbRpih54ZLfwrbiVC9