

LABORATORIO DIGITALES II

COMUNICACIÓN SPI

PIC 16F877A

Jefferson Camilo Grueso, David Santiago Moreno, Daniel Fernando Escamilla.

Universidad de Cundinamarca

Facultad de ingeniería

Ingeniería electrónica

Fusagasugá

2014

INTRODUCCIÓN

El **Bus SPI** es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. El bus de interfaz de periféricos serie o bus SPI es un estándar para controlar casi cualquier dispositivo electrónico digital que acepte un flujo de bits serie regulado por un reloj.

SERIAL PERIPHERAL INTERFACE

Spi es un bus de tres líneas, sobre el cual se transmiten paquetes de información de 8 bits. Cada una de estas tres líneas porta la información entre los diferentes dispositivos conectados al bus. Cada dispositivo conectado al bus puede actuar como transmisor y receptor al mismo tiempo, por lo que este tipo de comunicación serial es full duplex. Dos de estas líneas transfieren los datos (una en cada dirección) y la tercer línea es la del reloj. Algunos dispositivos solo pueden ser transmisores y otros solo receptores, generalmente un

dispositivo que tramite datos también puede recibir.

Los dispositivos conectados al bus son definidos como maestros y esclavos. Un maestro es aquel que inicia la transferencia de información sobre el bus y genera las señales de reloj y control.

Un esclavo es un dispositivo controlado por el maestro. Cada esclavo es controlado sobre el bus a través de una línea selectora llamada Chip Select o Select Slave, por lo tanto es esclavo es activado solo cuando esta línea es seleccionada. Generalmente una línea de selección es dedicada para cada esclavo.

En un tiempo determinado T1, solo podrá existir un maestro sobre el bus. Cualquier dispositivo esclavo que no esté seleccionado, debe deshabilitarse (ponerlo en alta impedancia) a través de la línea selectora (chip select).

ESPECIFICACIONES DEL BUS

Todas las líneas del bus transmiten la información sobre una sola dirección.

La señal sobre la línea de reloj (SCLK) es generada por el maestro y sincroniza la transferencia de datos.

La línea MOSI (Master Out Slave In) transporta los datos del maestro hacia el esclavo.

La línea MISO (Master In Slave Out) transporta los datos del esclavo hacia el maestro.

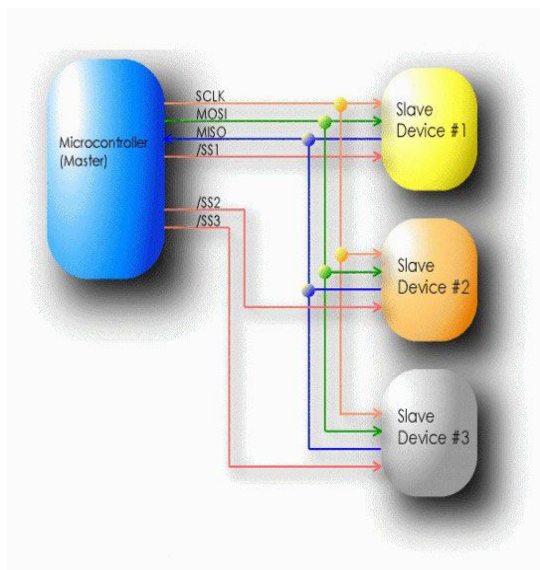


Figura 1. Especificaciones del bus.

Cada esclavo es seleccionado por un nivel lógico bajo ('0') a través de la línea (CS = Chip Select o SS Slave Select). Los datos sobre este bus pueden ser transmitidos a una razón de casi cero bits /segundo hasta 1 Mbits/ segundo. Los datos son transferidos en bloques de 8 bits, en donde el bit más significativo (MSB) se transmite primero.

OPERACIÓN

El SPI es un protocolo síncrono. La sincronización y la transmisión de datos se realizan por medio de 4 señales:

- **SCLK (Clock):** Es el pulso que marca la sincronización. Con cada pulso de este reloj, se lee o se envía un bit. También llamado TAKT (en Alemán).
- **MOSI (Master Output Slave Input):** Salida de datos del Master y entrada de datos al Slave. También llamada SIMO.
- **MISO (Master Input Slave Output):** Salida de datos del Slave y entrada al Master. También conocida por SOMI.
- **SS/Select:** Para seleccionar un Slave, o para que el Master le diga al Slave que se active. También llamada SSTE.

La Cadena de bits es enviada de manera síncrona con los pulsos del reloj, es decir con cada pulso, el Master envía un bit. Para que empiece la transmisión el Master baja la señal SSTE ó SS/Select a cero, con esto el Slave se activa y empieza la transmisión, con un pulso de reloj al mismo tiempo que el primer bit es leído. Nótese que los pulsos de reloj pueden estar programados de manera que la transmisión del bit se realice en 4 modos diferentes, a esto se llama polaridad y fase de la transmisión:

- 1. Con el flanco de subida sin retraso.
- 2. Con el flanco de subida con retraso.
- 3. Con el flanco de bajada sin retraso.
- 4. Con el flanco de bajada con retraso.

MATERIALES

- Target: micro-controlador
- Entorno de programación (IDE) o toolchain para su micro-controlador.
- Software de control de versiones (CVS o SCM).
- Biblioteca test unitarios.

PLANTEAMIENTO Y PROCEDIMIENTO

Inicialmente realizamos una investigación previa acerca del Serial Peripheral Interface bus SPI, luego de que nos documentamos acerca de Serial Peripheral Interface escogemos el micro-controlador que vamos a utilizar para nuestra practica y miramos el datasheet de dicho micro-controlador para conocer las instrucciones necesarias para dicho micro-controlador.

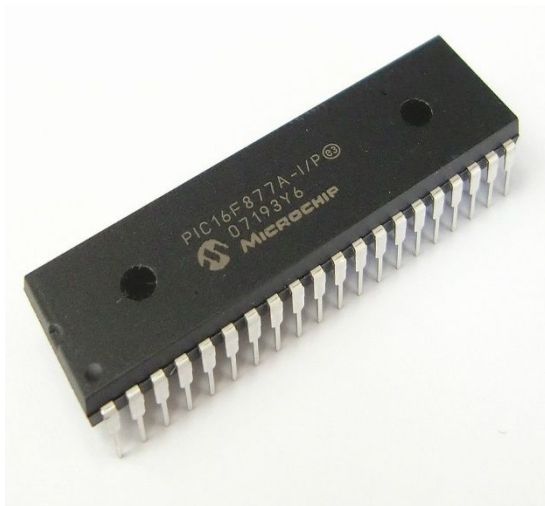


Figura 2. PIC 16f877A

40-Pin PDIP

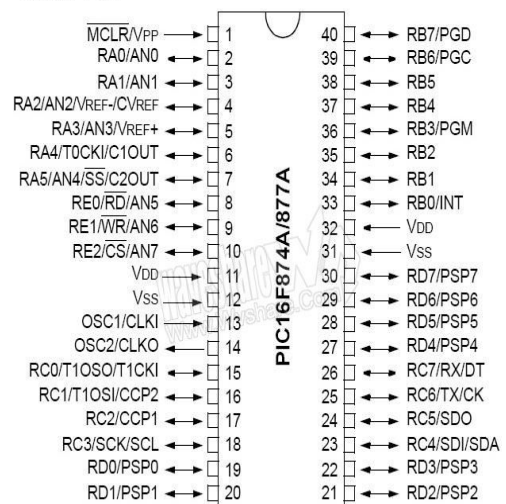


Figura 3. Datasheet pines micro-controlador

Después se realiza la implementación del Serial Peripheral Interface en C, esta se realiza con el fin de entender mejor el funcionamiento de la SPI, para lo cual se ha realizado con anterioridad una breve documentación del lenguaje de programación C, luego de tener el código en C se realizan algunos test unitarios del mismo.

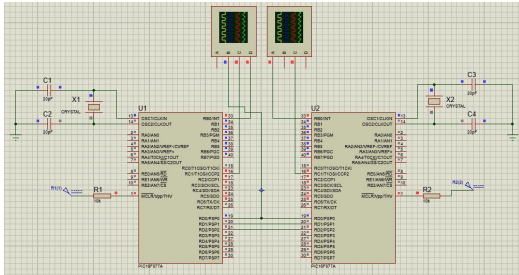
Ahora que se tiene la implementación y se entiende cómo funciona la SPI se pasa esta implementación al lenguaje de programación assembler de igual manera se ha realizado una investigación previa acerca del lenguaje de programación assembler y de su sintaxis.

CODIGO ASEMBLER

Después de realizar la implementación en assembler se realiza la simulación de la práctica para la cual se usa el software de simulación proteus, con esta simulación vamos probar el funcionamiento de la

implementación realizada en assembler y con esto se muestra también el funcionamiento de la práctica.

SIMULACION



CONCLUSIONES

- La capacidad full-duplex del protocolo de comunicaciones SPI, lo hace en la mayoría de los casos simple y eficiente para aplicaciones de un único maestro y un único esclavo también para múltiples esclavos y un solo maestro.
- Por otro lado puede ser problemático ponerlo en práctica para más de un esclavo, debido a la falta de un mejor direccionamiento y su dificultad aumenta a medida que aumenta el número de esclavos.

REFERENCIAS.

- <http://es.slideshare.net/OswST/serial-peripheral-interface-spi>
- http://wiki.webdearde.com/index.php?title=Tutorial%3A_Programaci3n_de_PICs_en_Ensamblador
- <http://www.electroensaimada.com/spi.html>

- http://es.wikipedia.org/wiki/Serial_Peripheral_Interface
- http://wikitronica.labc.usb.v.e/index.php/SPI:_Interfaz_de_Serial_Periferico_para_el_microcontrolador_QE_MC9S08QE128
- <http://prezi.com/mctrsanbtaIf/spi-serial-peripheral-interface/>