Digilent chipKIT Software SPI Library

Revision: January 18, 2012

1300 NE Henley Court, Suite 3
Pullman, WA 99163
(509) 334 6306 Voice | (509) 334 6300 Fax

Introduction

The Software SPI (SoftSPI) library allows the creation of any number of software based (bit-banged) SPI ports.

Serial Peripheral Interface (SPI) is a four wire synchronous serial interface used by many integrated circuits and electronic devices. SPI devices can operate as either master devices or as slave device. The four SPI signals are generally referred to as Slave Select (SS), Master Out, Slave In (MOSI), Master In, Slave Out (MISO), and Serial Clock (SCK). A master device generates SS, MOSI and SCK, and receives MISO. A slave device receives SS, MOSI, and SCK and transmits MISO. The SS signal is used to enable the slave device, and this signal is only significant for slave devices. A master device can use any general purpose I/O pin to generate SS to enable a slave.

The SoftSPI library only supports operation as an SPI master device.

An SPI transaction begins with the master device bringing SS low. When the slave ses SS go low it becomes enabled and waits for the master to send data. The master shifts data out on MOSI and simultaneously shifts data in on MISO. The slave device receives data from the master on its MOSI pin and simultaneously sends data to the master on its MISO pin. Each time the master sends a byte to the slave, it simultaneously receives a byte from the slave. The master generates the clock signal (SCK) that is used to control the shifting of the data in both the master and the slave.

SPI devices can operate using one of four data transfer modes, and one of two shift directions. The transfer modes specify the idle state of the clock signal (idles high or idles low) and the phase relationship between the active edge of the clock signal and the data. The modes are generally called mode 0 through mode 3. The shift direction specifies whether the data is shifted most significant bit first (shift left), or least significant bit first (shift right). The SoftSPI library header file defines symbols used to specify the transfer mode and shift direction. Refer to documentation for the SPI slave device being used to determine the transfer mode and shift direction to use. Most SPI devices use mode 0 with shift left.

The SoftSPI library defines an object class (SoftSPI) that is used to create a separate object for each desired SPI port. Each instance of the SoftSPI object class uses four specified digital pins for the SPI signals. The SoftSPI object class supports all four transfer modes, both shift directions, and setting the frequency of the clock signal.

To use the SoftSPI library, an object instance variable of the SoftSPI object class must be created. The object instance is initialized using functions to set mode, shift direction and clock speed. Data can then be transferred to a slave device by calling the various data transfer functions.

Defined Interface Symbols

The following symbols are defined in the SoftSPI header file (SoftSPI.h) and can be used with the various configuration functions:

Transfer Mode Values:

- SSPI_MODE0
- SSPI_MODE1
- SSPI_MODE2
- SSPI_MODE3

Shift Direction Values:

- SSPI_SHIFT_LEFT
- SSPI_SHIFT_RIGHT

Default Clock Speed Value:

- SSPI_SPEED_DEFAULT - select the default SPI clock speed, 500Khz.

SoftSPI Functions

The following describes the various functions defined by the SoftSPI:

Initialization and Setup Functions

The following functions are used to initialize the SoftSPI object and configure it for operation:

int begin(uint8_t pinSS, uint8_t pinMOSI, uint8_t pinMISO, uint8_t pinSCK)

Parameters:

pinSS	digital pin to use for SS
pinMOSI	digital pin to use for MOSI
pinMISO	digital pin to use for MISO
pinSCK	digital pin to use for SCK

Return value:
Returns 0 if invalid pin number specified

This function is used to specify the digital pin numbers of the four pins to use to make up the SPI port. This function must be called before any of the transfer functions can be used.

void end()

Parameters:
none

Return value:
none

This function is called when the SPI port is no longer going to be used. It releases the digital pins so that they can be used for other purposes.

void setSpeed(uint32_t spd)

Parameters:
spd desired SPI clock speed in Hz

Return value:
none

This function is used to set the frequency of the SPI clock. The default value if this function is not called will be SSPI_SPEED_DEFAULT (500Khz). The clock timing is determined by software and will not be exact. This will set the clock speed to the nearest supportable frequency to the requested frequency. The maximum frequency is dependent on CPU speed, and is ~1.65Mhz with an 80Mhz CPU clock frequency.

void setMode(int mod)

Parameters:
mod SPI transfer mode

Return value:
none

This function used to set the SPI transfer mode. The allowed values for mod are: SSPI_MODE0, SSPI_MODE1, SSPI_MODE2, SSPI_MODE3.

void setDirection(int dir)

Parameters:
mod desired shift direction

Return value:
none

This function used to set the data shift direction. The allowed values for dir are: SSPI_SHIFT_LEFT, SSPI_SHIFT_RIGHT.

void setDelay(int mod)

Parameters:
dly inter-byte delay in microseconds

Return value:
none

This is used to set an inter-byte delay when using a buffered data transfer function. Some SPI devices require that there be a time delay between successive bytes in order to receive and process the data correctly. This sets the number of microseconds to wait after sending a byte before sending the next byte of data.

void setSelect(uint8_t sel)

Parameters:

sel	state to set the SS pin

Return value:
none

This function is used to set the state of the SS pin. The allowed values for sel are HIGH or LOW. This function is called to set the SS pin LOW at the beginning of a transfer, and again to set the SS pin HIGH at the end of a transfer.

Data Transfer Functions

The following functions are used to transfer data to and/or from the SPI slave device:

uint8_t transfer(uint8_t val)

Parameters:

val	byte to send to the slave device

Return value:
Returns the byte received from the slave device.

This function is used to transfer a single byte to an SPI slave device and simultaneously receive a byte from the slave device.

void transfer(uint16_t cnt, uint8_t * snd, uint8_t * rcv)

Parameters:

cnt	number of bytes to send/receive
snd	array of bytes to send to the slave
rcv	array to hold bytes received from the slave

Return value:
none

This function is used to send an array of bytes to the slave device and simultaneously receive an array of bytes from the slave device.

void transfer(uint16_t cnt, uint8_t * snd)

Parameters:

cnt	number of bytes to send/receive
snd	array of bytes to send to the slave

Return value:
none

This function is used to send an array of bytes to the slave. Any bytes received from the slave device are ignored.

void transfer(uint16_t cnt, uint8_t pad, uint8_t * rcv)

Parameters:

cnt	number of bytes to send/receive
pad	byte sent repeatedly to slave
rcv	array to hold bytes received from the slave

Return value:
none

This function is used to read an array of bytes from the slave device. The value of pad will be sent repeatedly to cause the bytes to be sent by the slave.