EMBEDDED SYSTEMS TECH NOTE
SPI SERIAL COMMUNICATIONS

Winter 2008
Milwaukee School of Engineering

BASICS

1. SPI is an acronym for a serial peripheral interface bus originally specified by Motorola and commonly implemented in modern microcontrollers.
2. The SPI is a synchronous, serial communication interface used to transfer 8-bit data values between digital components.
    a. **Synchronous** means with (syn) + time (chronous). Synchronous communication circuits coordinate data transfer to the rising or falling edge of a clock signal.
    b. **Asynchronous** means not (a) + with (syn) + time (chronous). Asynchronous communication circuits do not use the edges of a clock signal to coordinate data transfer.
    c. **Serial** communication circuits send one (1) bit of information per clock period. An n-bit transfer will require n clock cycles.
        i. The advantage is less wiring expense.
        ii. The disadvantage is time to transfer.
    d. **Parallel** communication circuits send n bits of information per clock period. An n-bit transfer will require 1 clock cycle.
        i. The advantage is time to transfer.
        ii. The disadvantage is wiring expense.
3. SPI transfers occur between two SPI-capable digital circuits.
    a. **Master** circuits initiate the SPI data transfers.
    b. **Slave** circuits respond to master initiated data transfers.
    c. Embedded microcontroller are usually configured as the master for slave sensors and actuators.
4. SPI transfers occur as a 16-bit ring. The 8-bit master data is moved into the slave at the same time that the 8-bit slave data is moved into the master.

FIRMWARE CONTROL OF THE SPI I/O DEVICE

1. SPI input and output signals are multiplexed with general purpose port pins on microcontrollers.
    a. For example, the SPI signals are multiplexed onto Atmega32 port-b pins.

    b. For example, the SPI signals are multiplexed onto Atmega32 port-d pins.
2. SPI control and status signals are mapped to three I/O control registers called **SPCR**, **SPSR**, and **SPDR**. Successful use of the SPI device involves correct initialization of the SPCR, SPSR, and DDRD.
3. Master outputs
    a. **MOSI** = master-out/slave-in = the serial data line for bits moving from the master to the slave.
    b. **SCK** = SPI clock = the programmable, synchronizing clock signal.
4. Master inputs
    a. **SS** = slave select = a selection signal used by the master to enable/select an SPI device.
    b. Microcontrollers can be slaves to another SPI device, and thus, the SS signal exists on the Atmega32 and MC68HC11 microcontrollers, for example.
    c. **MISO** = master-in/slave-out = the serial data line for bits moving from the slave to the master.
5. Control signals are memory-mapped into the register call **SPCR**.
    a. **SPIE** = SPI interrupt enable
    b. Interrupts are appropriate for some problems but not for others. In the cases where interrupts are not appropriate, polling should be used to check a completion status flag, called SPIF, in the status register.
    c. **SPE** = SPI enable
        i. Set this bit to a 1 to enable the SPI.
        ii. If enabled, the SPI will drive MOSI and SCK in response to data transfer requests.
        iii. If disabled, the SPI will not drive MOSI and SCK allowing normal port pin functionality.
    d. **MSTR** = master selection
        i. Set this bit to 1 to select master mode.
    e. **CPOL** = clock polarity
        i. This bit determines the pretransaction value of the SCK signal.
        ii. Look at the SPCR information page in the microcontroller's reference manual. Note that CPOL = 0 forces SCK low pretransaction while CPOL = 1 forces SCK high pretransaction.
    f. **CPHA** = clock phasing
    g. **SPR1:SPR0** = SCK clock frequency control
        i. The SCK is derived from the master's clock circuits.
        ii. The SPR1:SPR0 bits select SCK as a divided frequency of the system clock. SCK is always SLOWER than the system clock.
6. Status signals are memory-mapped into a register called **SPSR**.
    a. **SPIF** = SPI complete flag
        i. This bit is set to 1 by the SPI whenever a data transfer completes.
        ii. Busy-wait on the SPIF after requesting a transfer if you need polled mode software synchronization.

7. The transferred data bytes are memory mapped into a register called **SPDR**.
8. You may hear the course instructor or others instructors refer to the SPI registers using words.
    a. SPCR pronounced SPEAKER.
    b. SPSR pronounced SPICER
    c. SPDR pronounced SPEADER.