Using the HC12 Serial Peripheral Interface

1. Enable SPI (SPE bit of \texttt{SP0CR1})

2. Clock phase and polarity set to match device communicating with

3. Select clock polarity - CPOL bit of \texttt{SP0CR1}
   - CPOL = 0 for clock idle low
   - CPOL = 1 for clock idle high

4. Select clock phase - CPHA bit of \texttt{SP0CR1}
   - CPHA = 0 for data valid on first clock edge
   - CPHA = 1 for data valid on second clock edge

5. Select master or slave MSTR bit of \texttt{SP0CR1}
   - Will be master when talking to devices such as D/A, A/D, clock, etc.
   - May be slave if talking to another microprocessor

6. If you want to receive interrupt after one byte transferred, enable interrupts with SPIE bit of \texttt{SP0CR1}
   - Normally master will not use interrupts - transfers are fast enough that you will normally wait for transfer to complete
   - Will often use interrupts when configured as a slave - you will get interrupt when master sends you data

7. Configure LSBF of \texttt{SP0CR1} for MSB first (LSBF = 0) or LSB first (LSBF = 1)
   - For most devices, use MSB first

8. Configure for normal mode by clearing bit SPC0 of \texttt{SP0CR2}
   - Bidirectional mode (SPC1 = 1 in \texttt{SP0CR2}) used for three-wire communication - need some protocol for selecting who is sender and who is receiver

Master Mode:
1. **Set clock rate** - *SPR2:0* bits of *SP0BR*
   - Normally select clock at highest rate compatible with slave

2. **Make MOSI, SCLK, and SS output** - bits *DDS5, DDS6, DDS7* of *DDRS*

3. **MISO automatically configured as input by choosing master mode**

4. **Configure some way to select slave(s)** - probably SS if only one slave; other I/O bits if multiple slaves

5. **Start data transfer by writing byte to *SP0DR***

6. **After transfer complete (8 clock cycles), SPIF bit of *SP0SR* set.**
   - If writing data to slave, can send next byte to *SP0DR*
   - If reading data from slave, can read data from *SP0DR*

7. **Set up SSOE of *SP0CR1***
   - *SSOE = 0* if you want to control SS yourself (to be able to send more than one byte with SS low)
   - *SSOE = 1* if you want to SS controlled automatically (SS will be active for one byte at a time)

**Slave Mode:**

1. **No need to set clock mode** - slave accepts data at rate sent by master (up to 4 MHz)

2. **Need to make MISO output** - bit *DDS4* of *DDRS*

3. **No need to Make MOSI, SCLK, and SS inputs** - this is done automatically when configuring HC12 as slave
   - If receiving data from master, wait until SPIF flag of *SP0SR* set (or until SPI interrupt received), then read data from *SP0DR*
   - If sending data to master, write data to *SP0DR* **before** master starts transfer