C8051F120 SPI0 (Serial Peripheral Interface) Configuration (read Technical Summary, Ch. 20 for details)

1. SPI0 Configuration Register – SPI0CFG (0x9A, Page 0)
   - SPIBSY: Status Flag indicating a transfer is in progress
   - MSTEN: Sets the processor as the SPI master or slave
   - CKPHA: Controls the clock phase setting (see figures on p. 279 of Technical Summary)
   - CKPOL: Controls the clock polarity setting (see figures on p. 279 of Technical Summary)
   - SLVSEL: Status Flag for the NSS signal (debounced)
   - NSSIN: Actual status of NSS pin (not debounced)
   - SRMT: Status Flag indicating the Shift Register is empty
     - NOTE: only valid in slave mode, SRMT = 1 in Master Mode (MSTEN = 1)
   - RXBMT: Status Flag = 0 indicating the receive buffer has not been read
     - NOTE: only valid in slave mode, RXBMT = 1 in Master Mode (MSTEN = 1)

2. SPI0 Control Register – SPI0CN (0xF8, Page 0)
   - SPIF: SPI0 Interrupt Flag
   - WCOL: Write Collision Flag (a write to the SPI0 data register was attempted while a data transfer was in progress)
   - MODF: Mode Fault Flag (a master mode collision was detected)
   - RXOCRN: Receiver Overrun Flag (receive buffer has unread data when a new transmission is initiated
   - NSSMD1, NSSMD0: Slave Select Mode (3-wire, no NSS; 4-wire, multi-master; 4-wire, single-master) In single-master mode, NSSMD0 is used as the NSS
   - TXBMT: Transmit Buffer Empty Flag, indicates SPI is ready for next data byte
   - SPIEN: SPI0 enable (disable SPI to save power)

3. SPI0 Clock Rate – SPI0CKR (0x9D, Page 0)
   - Bits 7-0 set the SPI0 clock rate: $f_{SCK} = SYSCLK/[2 \times (SPI0CKR+1)]$

4. SPI0 Data Buffer – SPI0DAT (0x9B, Page 0)
   - Write to SFR to Transmit Character
   - Read SFR to Receive Character
     - NOTE: there are 2 separate buffers here, one is only written to and the other is only read from

NSS for Slave Selection: Slave must be selected before master can transfer data, and in the process, slave's data buffer is locked out so that slave cannot change it during transfer. Because of this, NSS must be released after completion of transfer if slave is to be able to write to data buffer. With SPI on XBAR, P0.5 is NOT NSS, NSSMD0 is.

Operational Setup

Wire hardware connections for 4-wire, single-master mode (Figure 20.4, p. 276 Tech Summary)
Determine SPI clock rate and set SPI0CKR register
Determine clock polarity, phase, master mode, and set SPI0CFG
Set NSSMD1 = 1 and use NSSMD0 in SPI0CN to set NSS

Operation – Single Character Transmission

Get character (from terminal)
Select slave (NSS - NSSMD0)
Make sure SPI is not busy
Clear SPIF
Load character into SPI data register (SPI0DAT)
Wait until transmission is complete (SPIBSY, SPIF or TXBMT: understand differences among these)
Release slave (NSS, but release not critical if no slave write-back expected)
Get received character from SPI data register (SPI0DAT)
{Discard or send to terminal}
- May need to delay ~2ms
Select slave (NSS - NSSMD0)
Make sure SPI is not busy
Clear SPIF
Load a dummy character into SPI data register (SPI0DAT)
Wait until transmission is complete (SPIBSY, . . .)
Release slave (NSS)
Get received character from SPI data register (SPI0DAT)
Output to terminal

Operation – Multiple Character String Receive (after <DEL>)

Get <DEL> (\177 or 0x7F)
Select slave (NSS - NSSMD0)
Make sure SPI is not busy
Clear SPIF
Load <DEL> into SPI data register (SPI0DAT)
Wait until transmission is complete (SPIBSY, . . .)
Release slave (NSS)
Get received character from SPI data register (SPI0DAT)
{Discard or send to terminal}
LOOP:
Delay ~2ms (to give slave chance to reload new character)
Select slave (NSS - NSSMD0)
Make sure SPI is not busy
Clear SPIF
Load a dummy character into SPI data register (SPI0DAT)
Wait until transmission is complete (SPIBSY, . . .)
Release slave (NSS)
Get received character from SPI data register (SPI0DAT)
Output to terminal
Check to see if last character = 0xFF (string terminating value)
If not, go back to LOOP: