1. Serial Peripheral Interface (SPI)

1.1 Overview

SPI is a synchronous serial communication protocol commonly used for fast peripheral communication in embedded systems. It typically uses four lines:

- SCK: Serial Clock
- MOSI: Master Out Slave In
- MISO: Master In Slave Out
- SS/CS: Slave Select / Chip Select

SPI is full-duplex, meaning data can be transmitted and received at the same time.

1.2 Communication Flow

1. The master pulls SS low to select a slave.
2. The master generates clock pulses on SCK.
3. Data shifts out on MOSI and shifts in on MISO.
4. The transfer ends when the expected bits/bytes are completed.
5. The master releases SS high.

1.3 SPI Clock Modes

- Mode 0: CPOL = 0, CPHA = 0
- Mode 1: CPOL = 0, CPHA = 1
- Mode 2: CPOL = 1, CPHA = 0
- Mode 3: CPOL = 1, CPHA = 1

1.4 SPI Registers (Detailed Bits)

SPDR - SPI Data Register

Function:

- Stores data to transmit and received data.

Bit	Name	Description
7	SPD7	Data bit 7
6	SPD6	Data bit 6
5	SPD5	Data bit 5
4	SPD4	Data bit 4
3	SPD3	Data bit 3
2	SPD2	Data bit 2
1	SPD1	Data bit 1
0	SPD0	Data bit 0

SPSR - SPI Status Register

Function:

- Indicates transfer completion and collision state.

Bit	Name	Description
7	SPIF	SPI interrupt/transfer complete flag
6	WCOL	Write collision flag
5	Reserved	Reserved
4	Reserved	Reserved
3	Reserved	Reserved
2	Reserved	Reserved
1	Reserved	Reserved
0	SPI2X	Double SPI speed (master mode)

SPCR - SPI Control Register

Function:

- Controls SPI enable state, mode, clock settings, and interrupts.

Bit	Name	Description
7	SPIE	Enable SPI interrupt
6	SPE	Enable SPI peripheral
5	DORD	Data order (1=LSB first, 0=MSB first)
4	MSTR	Master select (1=master, 0=slave)

Bit	Name	Description
3	CPOL	Clock polarity
2	CPHA	Clock phase
1	SPR1	Clock rate select bit 1
0	SPR0	Clock rate select bit 0

Clock-rate note:

- SPI clock rate in master mode is determined by SPR1, SPR0, and SPI2X.

1.5 SPI Assembly Examples

SPI Master

```
;------------------------
; Assembly Code SPI Master
;------------------------
#define __SFR_OFFSET 0x00
#include "avr/io.h"
;------------------------
.global main
;================================================================
main:
.equ  SCK, 5
.equ  MOSI, 3
.equ  SS, 2
;----------------------------------------------------------------
      LDI   R17, (1<<MOSI)|(1<<SCK)|(1<<SS)
      OUT   DDRB, R17       ;set MOSI, SCK, SS as o/p
      ;-------------------------------------------------------
      LDI   R17, (1<<SPE)|(1<<MSTR)|(1<<SPR0)
      OUT   SPCR, R17       ;enable SPI as master, fsck=fosc/16
      ;-------------------------------------------------------
      LDI   R17, 0xAA       ;byte to be transmitted
      ;-------------------------------------------------------
again:CBI   PORTB, SS       ;enable slave device
      OUT   SPDR, R17       ;transmit byte to slave
      ;-------------------------------------------------------
loop: IN    R18, SPSR
```

```
      SBRS  R18, SPIF       ;wait for byte transmission
      RJMP  loop            ;to complete
      ;-----------------------------------------------------
      SBI   PORTB, SS       ;disable slave device
      ;-----------------------------------------------------
      RCALL my_delay        ;delay
      COM   R17             ;1's compliment of byte
      RJMP  again           ;repeat transmission
;=============================================================
my_delay:                   ;delay in ms
    LDI   R20, 255
l6: LDI   R21, 255
l7: LDI   R22, 40
l8: DEC   R22
    BRNE  l8
    DEC   R21
    BRNE  l7
    DEC   R20
    BRNE  l6
    RET
```

SPI Slave

```
;------------------------
; Assembly Code SPI Slave
;------------------------
#define __SFR_OFFSET 0x00
#include "avr/io.h"
;------------------------
.global main
;=============================================================
main:
;-------------------------------------------------------------
      LDI   R17, 0xFF
      OUT   DDRD, R17       ;set PORTD for o/p
      ;-----------------------------------------------------
      LDI   R17, (1<<SPE)
      OUT   SPCR, R17       ;enable SPI as slave
      ;-----------------------------------------------------
agn:  IN    R18, SPSR
      SBRS  R18, SPIF       ;wait for byte reception
```

```
RJMP  agn
;-----------------------------------------------------
IN    R18, SPDR       ;i/p byte from data register
OUT   PORTD, R18      ;and o/p to PORTD
;-----------------------------------------------------
RJMP  agn             ;repeat reception
```

Revision #1
Created 2026-04-18 14:40:01 UTC by AX
Updated 2026-04-18 15:10:10 UTC by AX