entri

Serial Peripheral Interface - SPI

Dr.Sreejith Rajan

entri

SPI

- SPI (Serial Peripheral Interface) is a synchronous serial communication protocol used to connect a microcontroller (Master) with one or more peripheral devices (Slaves) such as sensors, displays, and memory chips.

- It was developed by Motorola and is widely used in embedded systems because it is fast, simple, and full-duplex (data can be sent and received at the same time).

entri

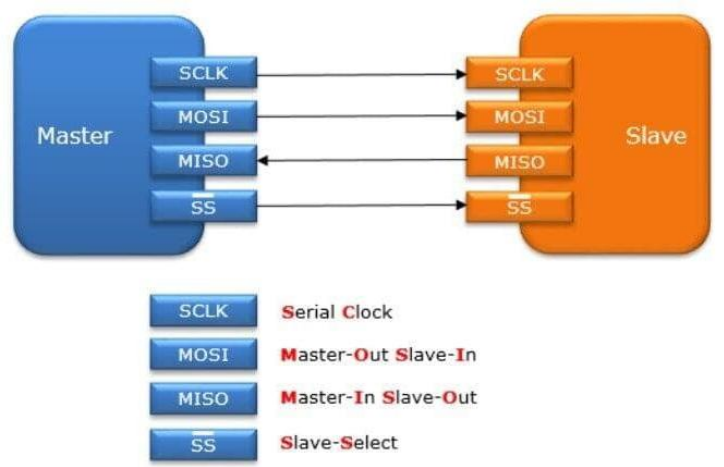
SPI – Common Usage

Common peripherals that use SPI:

- **OLED and TFT displays**
- **Temperature and pressure sensors (BMP280, MAX6675, etc.)**
- **Memory chips (EEPROM, Flash)**
- **ADC and DAC Ics**
- **SD cards**

entri

SPI Protocol



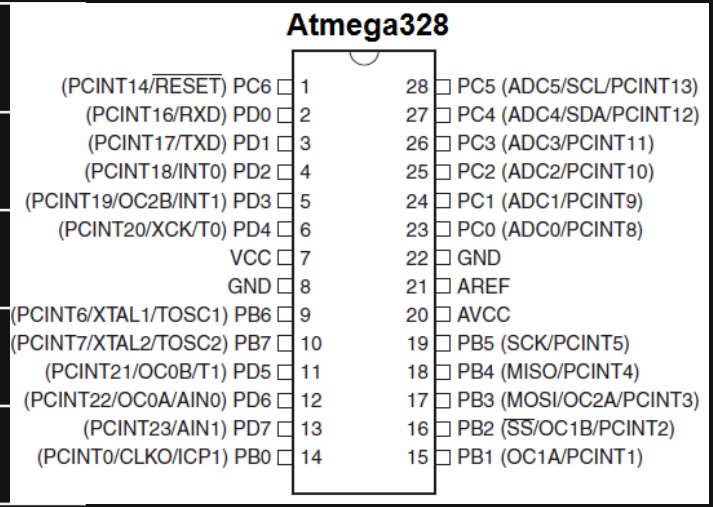
1

Line	Direction	Description
MOSI	Master → Slave	Master Out, Slave In (data from master to slave)
MISO	Slave → Master	Master In, Slave Out (data from slave to master)
SCK	Master → Slave	Serial Clock (generated by master)
SS (CS)	Master → Slave	Slave Select (active LOW, selects the slave)

entri

ATMEGA 328 Pin mapping

SPI Line	AVR Pin	Arduino Pin
MOSI	PB3	D11
MISO	PB4	D12
SCK	PB5	D13
SS	PB2	D10



entri

Useful Registers

Register Name	Address	Purpose
SPCR	0x2C	SPI Control Register — used to configure SPI mode and enable it
SPSR	0x2D	SPI Status Register — shows SPI transfer status and flags
SPDR	0x2E	SPI Data Register — holds data to transmit or receive

entri

SPCR – SPI Control Register

Bit	Name	Description
7	**SPIE**	SPI Interrupt Enable (1 = enable interrupt on transfer complete)
6	**SPE**	SPI Enable (1 = enables SPI peripheral)
5	**DORD**	Data Order (0 = MSB first, 1 = LSB first)
4	**MSTR**	Master/Slave Select (1 = Master mode)
3	**CPOL**	Clock Polarity (0 = SCK idle low, 1 = SCK idle high)
2	**CPHA**	Clock Phase (0 = sample on leading edge, 1 = sample on trailing edge)
1	**SPR1**	SPI Clock Rate Select bit 1
0	**SPR0**	SPI Clock Rate Select bit 0

entri

SPI Clock rate Table

SPR1	SPR0	SPI2X (in SPSR)	SCK Frequency
0	0	0	fosc / 4
0	0	1	fosc / 2
0	1	0	fosc / 16
0	1	1	fosc / 8
1	0	0	fosc / 64
1	0	1	fosc / 32
1	1	0	fosc / 128
1	1	1	fosc / 64

SPSR – SPI Status Register

Bit	Name	Description
7	**SPIF**	SPI Interrupt Flag (set when transfer complete)
6	**WCOL**	Write Collision Flag (set if SPDR written before previous transfer ended)
5–1	—	Reserved
0	**SPI2X**	Double SPI Speed Bit (1 = double speed)

entri

Steps to configure SPI in ATMega 328 (Master) – Step1: Understand the pins

Signal	AVR Pin	Arduino Pin	Direction (Master)	Function
MOSI	PB3	D11	Output	Master Out Slave In
MISO	PB4	D12	Input	Master In Slave Out
SCK	PB5	D13	Output	Serial Clock
SS	PB2	D10	Output	Slave Select (active low)

entri

Step 2: Set the Pin directions

In master mode

- MOSI, SCK, and SS must be outputs.
- MISO must be input.

entri

Step3 : Enable SPI and configure as master

Bit	Field	Meaning
SPE	SPI Enable	Must be set to 1 to enable SPI
MSTR	Master Select	1 for Master mode
SPR1, SPR0	Clock Rate Select	Choose SPI clock speed
CPOL, CPHA	Clock Polarity & Phase	Define SPI mode (0–3)

entri

Step4 : (Optional) Double the SPI Clock Speed

If you want to increase the SPI clock, set the **SPI2X** bit in the **SPSR register**.

Step 5 : Transmit and Receive Data

Use the **SPDR register** (SPI Data Register).

➔Writing to SPDR starts transmission.

➔Reading SPDR gives received data.

➔Wait for the transfer to complete by checking **SPIF** flag in **SPSR**.

entri

Step 6 : Handle the SS (Slave Select) Pin

Before sending data to a specific device:

➔ **Pull SS low** → select slave.
➔ **Transfer data.**
➔ **Pull SS high** → deselect slave

THANK YOU