The SPI protocol

By: Yehia M. Abu Eita

Outlines

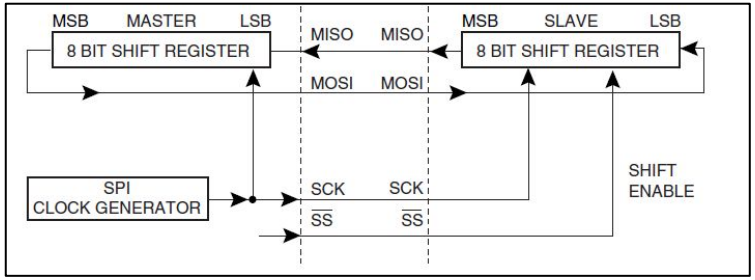
- What is SPI?
- SPI Connections
- SPI Block diagram
- SPI Modes of operation
- SPI Use Cases
- ATmega32 SPI registers
- Steps to program ATmega32 SPI

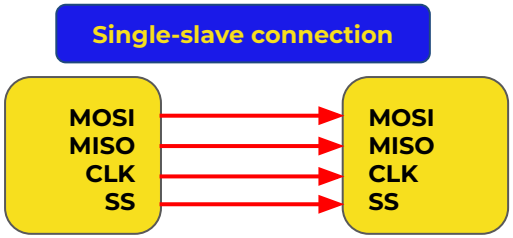
What is SPI?

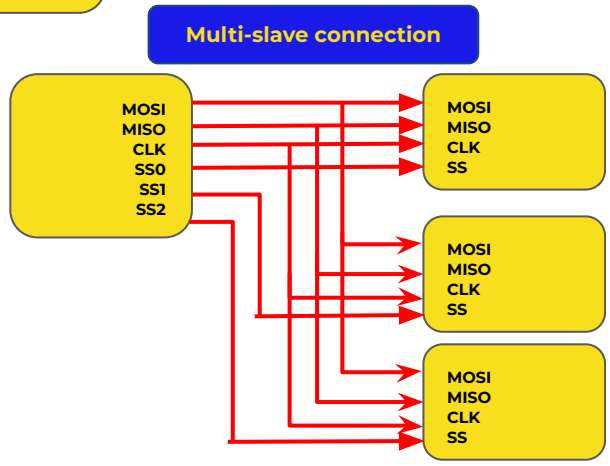
- SPI is one of the famous communications protocols used in embedded systems world.
- It is an acronym of **S**erial **P**eripheral **I**nterface.
- It is a **Multi-Slave Synchronous Full-duplex** communication protocol.
- It has a very **simple frame**, that contains **only the data** and has neither control nor error checking bytes.
- It has **data rates up to 10Mbps**.
- It is **more useful** for **short distance** communications.

SPI Connections

- **MOSI**: Master Out Slave In
- **MISO**: Master In Slave Out
- **CLK**: Clock
- **SS**: Slave Select



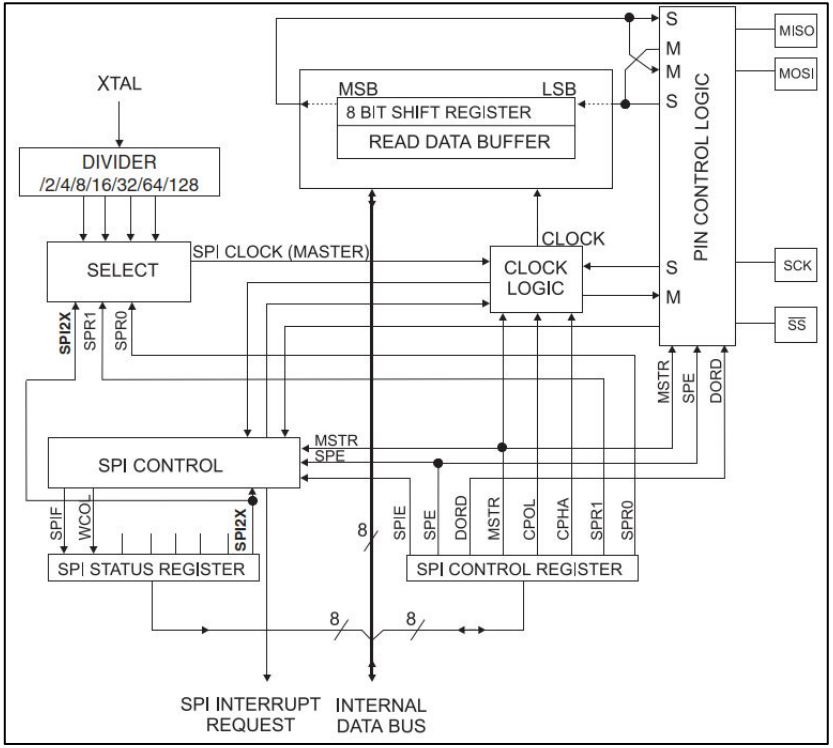




Internal connection

SPI Block diagram

- **SPI control**
- **Clock select**
- **Read data buffer**

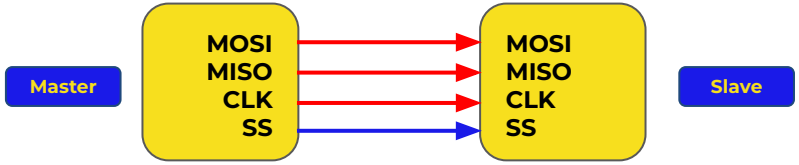


SPI Modes of operation

- **Master Mode:**
  - **Master** means it **initiates** the **communication** and **generates** the **clock**.
  - **All pins** are defined as **output except MISO** pin is defined as an **input**.
  - Master can take control of the slave-select pin to choose and make synchronization with the slave.
  - Driving the **slave-select pin low** will **start the communication** between the master and the chosen slave.
  - Driving the **slave-select pin high** will **reset the SPI logic circuitry**.
- **Slave Mode:**
  - When the SPI is configured as a slave **all pins** are configured as **input except the MISO** pin is configured as an **output**.
  - The SPI **slave will receive** the data from the master if the **slave-select pin is pulled low**.
  - When the **slave-select** pin is **pulled high** the **SPI slave logic will reset**.

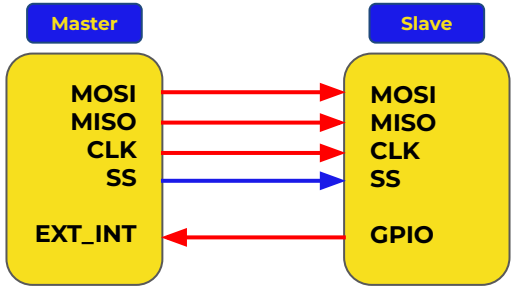
SPI Use Cases

- **Master to Slave:**
  - The master will prepare the data.
  - Then it will drive the slave-select pin low in order to start the communication.
  - The master will generate the clock.
  - The shift registers (Master and Slave Data Registers) will exchange their contents bit by bit until all the bits are transmitted.
  - Master should drive the slave-select pin high to stop the communication



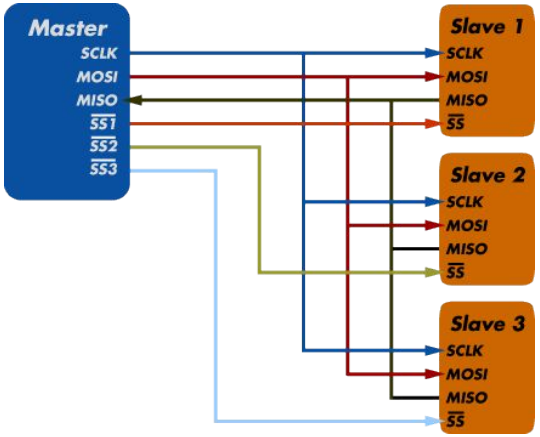
SPI Use Cases

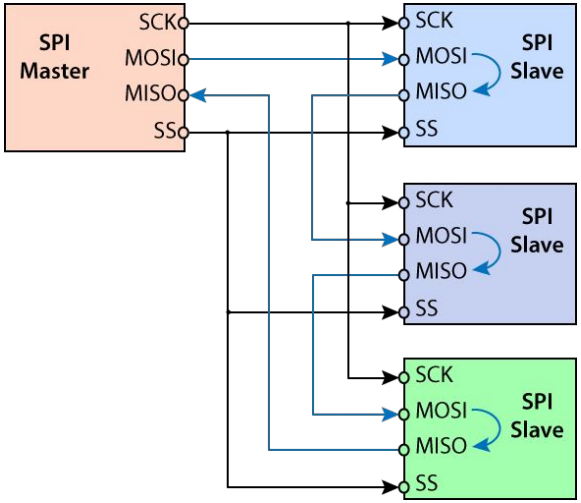
- **Slave to Master:**
  - The slave needs to tell the master that it want to start the communication.
  - The slave will trigger an external interrupt pin connected to the master.
  - The master will start the communication when the external interrupt happens.



SPI Use Cases

- **Multi-Slave SPI**
  - **Daisy chain will used if only one slave select pin is available**
  - **Normal connection**





ATmega32 SPI registers

SPCR – SPI Control Register

Bit	7	6	5	4	3	2	1	0	
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

SPSR – SPI Status Register

Bit	7	6	5	4	3	2	1	0	
	SPIF	WCOL	–	–	–	–	–	SPI2X	SPSR
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

SPDR – SPI Data Register

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	SPDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	X	X	X	X	X	X	X	X	Undefined

Steps to program ATmega32 SPI

- **Master initializing:**
  - Set MOSI, SCK, SS as an output pins
  - Set MISO pin as an input pin
  - Set SS pin to high
  - Enable SPI in master mode
  - Choose prescaler
- **Master send:**
  - Pull SS pin to low
  - Write data to SPI data register
  - Wait till transmission complete
  - Flush received data
  - Set SS pin to high
- **Master receive:**
  - Set dummy value
  - Wait till reception complete
  - Read received data

Steps to program ATmega32 SPI

- **Slave initializing:**
  - Set MOSI, SCK, SS as an input pins
  - Set MISO pin as an output pin
  - Enable SPI in slave mode
  - Choose prescaler
- **Slave send:**
  - Must notify the master using DIO pin
  - Write data to SPI data register
  - Wait till transmission complete
  - Flush received data
- **Slave receive:**
  - Set dummy value
  - Wait till reception complete
  - Read received data

Summary

- Now you are familiar with the SPI protocol
- You can make good SPI driver
- Remember, Master must drive the slave-select pin low in order to start communication with the slave
- SPI has faster throughput than UART and I2C