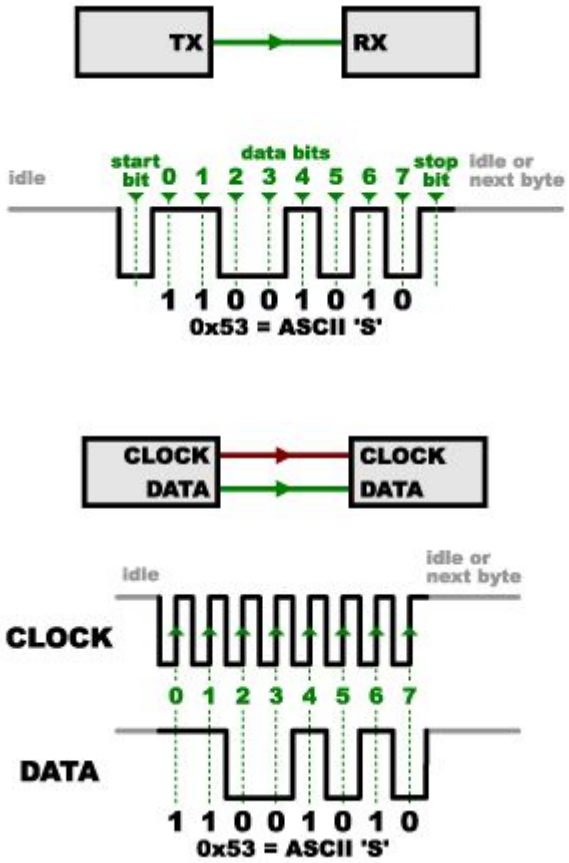
Synchronous Serial Communication

Communication Protocols

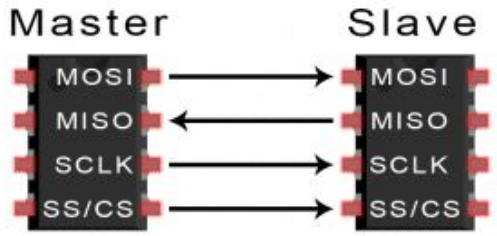
Serial Communication

- Asynchronous Communication
  - No dedicated clock
    - Both sides should agree on the baud rate
  - Handshaking
  - Slower speeds
    - E.g. in UART, up to 115200 bps
- Synchronous Communication
  - Dedicated clock
  - No Handshaking
  - Higher speeds
  - E.g. SPI, I2C and etc.



Synchronous Serial Communication (SPI)

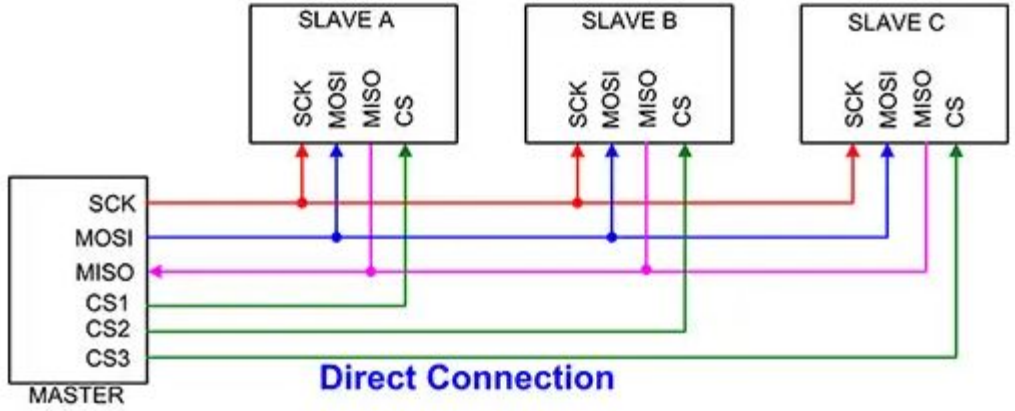
- Serial Peripheral Interface (SPI)
  - Usually used to send and receive data between microcontrollers and small peripheral devices like sensors, displays, SD cards and etc.
  - Supports full-duplex communication
  - Single master, multiple slaves
  - Normally uses 4 lines

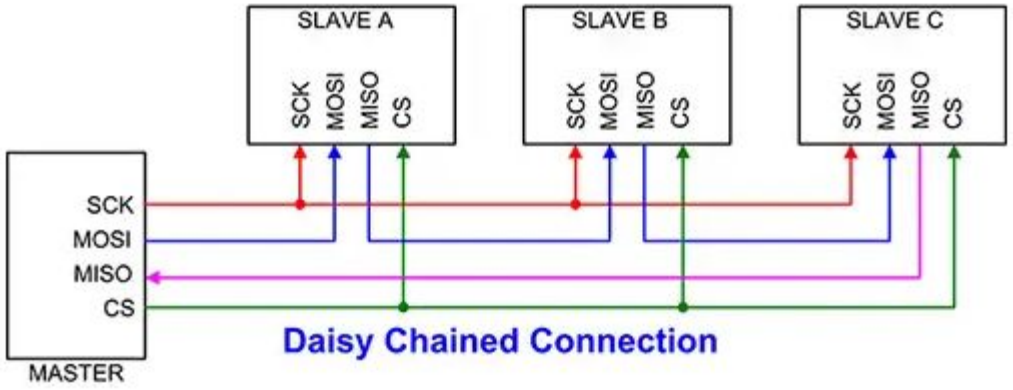


    - **MOSI** (Master Output/Slave Input): Line for the master to send data to the slave.
    - **MISO** (Master Input/Slave Output): Line for the slave to send data to the master.
    - **SCLK** (Clock) : Line for the clock signal.
      - This line is used to synchronize the communication
    - $\overline{\textbf{SS}}$ / $\overline{\textbf{CS}}$ (Slave Select/Chip Select): Line for the master to select which slave to send data to.
      - This line is active low

Synchronous Serial Communication (SPI)

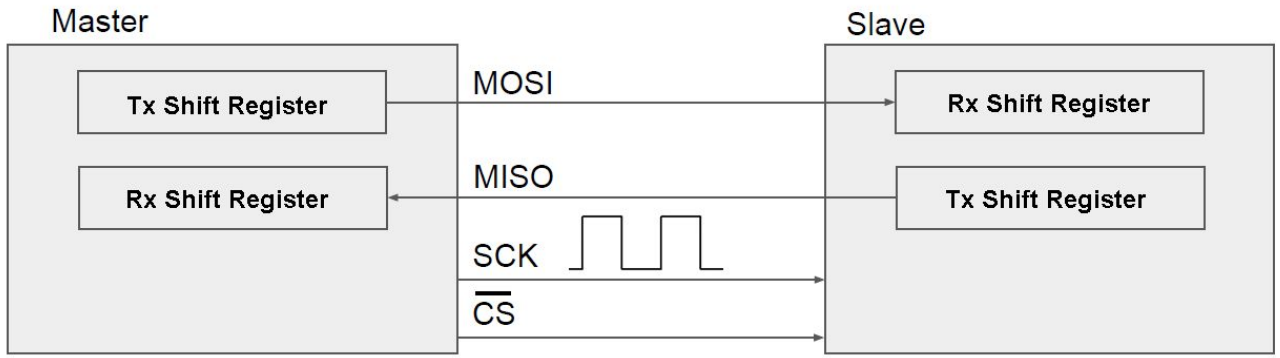
- SPI Bus Design
  - Direct Connection
    - For every slave there should be a separate chip select signal on the master
  - Daisy Chaining Connection
    - A single chip select signal on the master controls all the chip select pins of the slaves





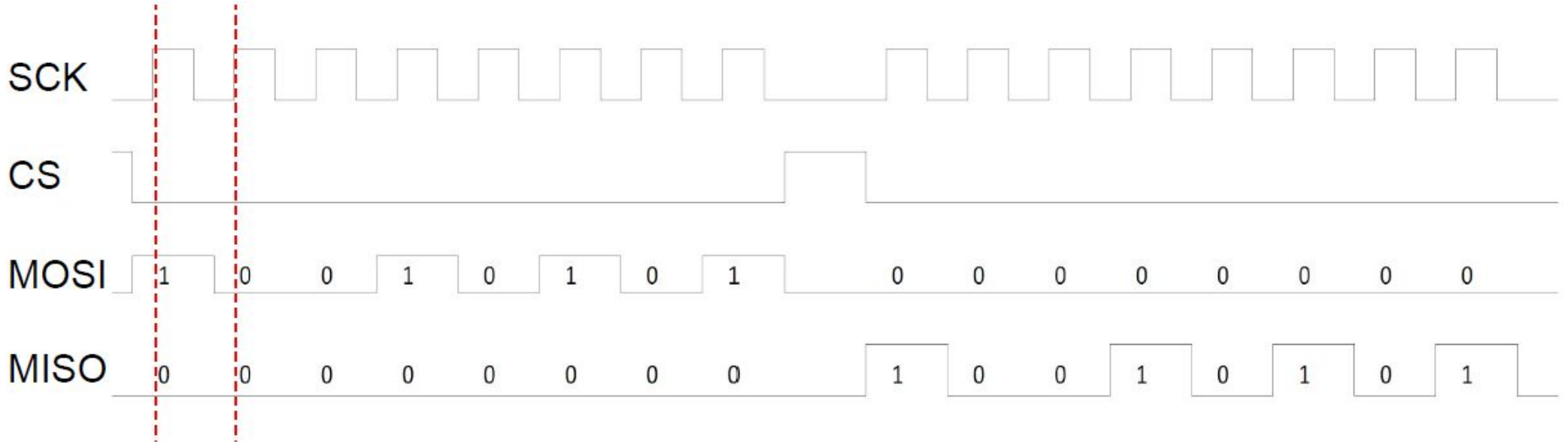
Synchronous Serial Communication (SPI)

- SPI Data
  - When data is clocked out on the master it is also clocked out on the slave
  - Any number of bits can be sent or received in a continuous stream.
  - The receiving hardware can be a simple shift register.
    - If data only needs to be received over SPI, a simple shift register like 74HC164 can be used
    - An SPI IO expander like MCP23S08 can be used to expand the GPIOs



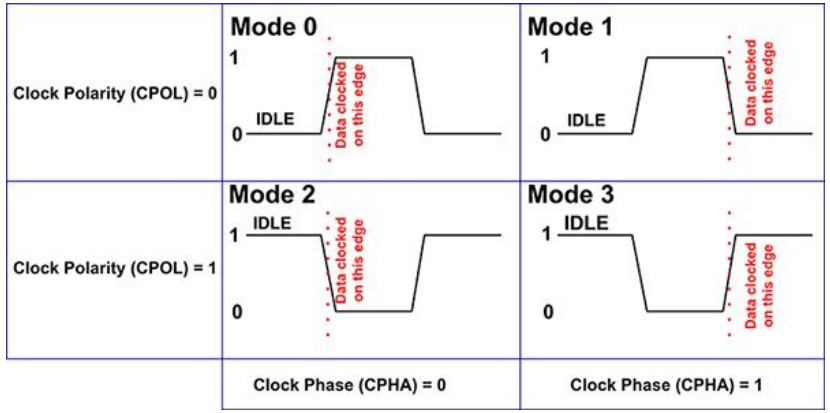
Synchronous Serial Communication (SPI)

- SPI Transaction
  - The CS normally is HIGH. Just before data is sent to the slave,CS should be set to LOW
  - Example: Bidirectional Transaction - 8bit master to slave and 8bit slave to master

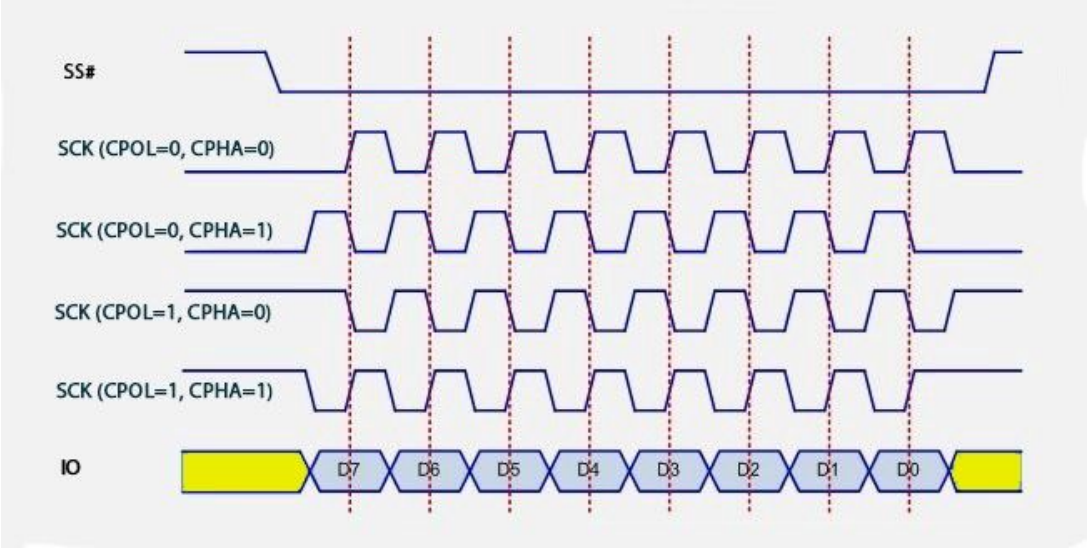


Synchronous Serial Communication (SPI)

- SPI Modes of Communication
  - CPOL - Clock Polarity: If the clock signal is inverted or not
  - CPHA - Clock Phase: If the data should be sampled on rising edge or falling edge of the clock.



Mode	Clock Polarity (CPOL)	Clock Phase (CPHA)	Data Capture
SPI_MODE0	0	0	Rising
SPI_MODE1	0	1	Falling
SPI_MODE2	1	0	Falling
SPI_MODE3	1	1	Rising



Data is written on an edge and sampled on the next edge.

Synchronous Serial Communication (SPI)

Advantages	**Disadvantages**
Full duplex and faster than I2C and UART	Requires more pins on chip than I2C and UART
Size of the message is flexible and it can be sent or received in a continuous stream without any interrupt	No error-checking, handshaking and acknowledgment (master could be transmitting without slaves)
Low power consumption, because of its simple hardware	Single master device
Slaves use master's clock, no need for precision oscillators	Unique chip select signals are required for the devices
Slaves do not need a unique address	Handles short-distance communication
Signals are unidirectional, easy for galvanic isolation	Interrupts must either be implemented with signals or be faked using periodic polling
SPI is a de facto standard interface.There are different types of SPI. E.g. four-wire, three-wire and etc.	

Synchronous Serial Communication

- Some useful links
  - Serial Peripheral Interface
  - Sparkfun SPI Tutorial
  - SPI on Arduino
  - What is SPI? Basics for beginners!
  - SPI and how to use it
  - Arduino Workshop - Using SPI
  - SPI Overview
  - Galvanic isolation