UNIT III – IOT PHYSICAL DEVICES AND ENDPOINTS

Introduction to Arduino and Raspberry Pi- Installation, Interfaces (serial, SPI, I2C), Programming – Python program with Raspberry PI with focus on interfacing external gadgets, controlling output, and reading input from pins.

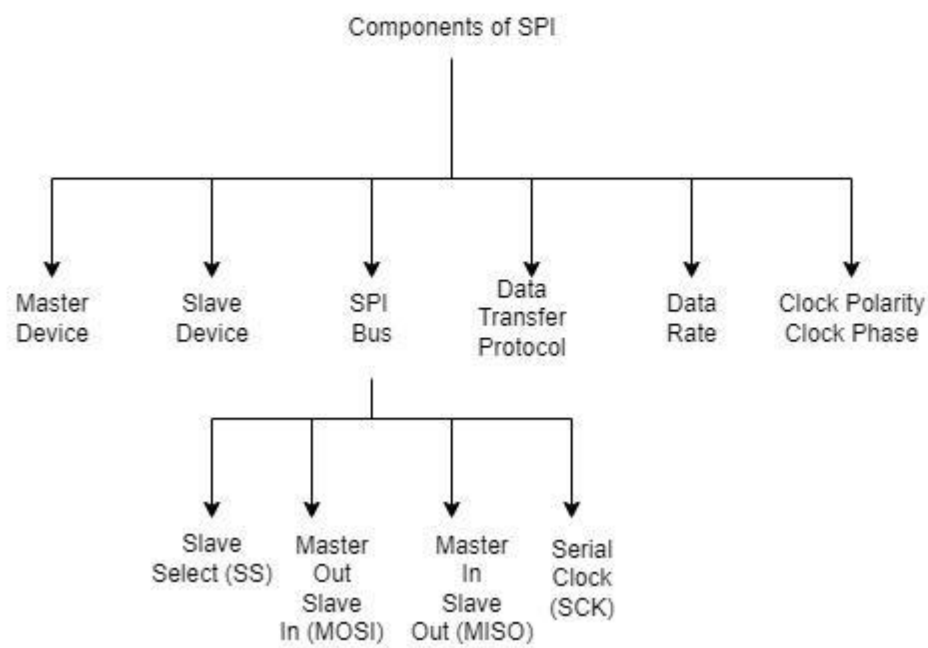
INTERFACES (SERIAL, SPI, I2C)

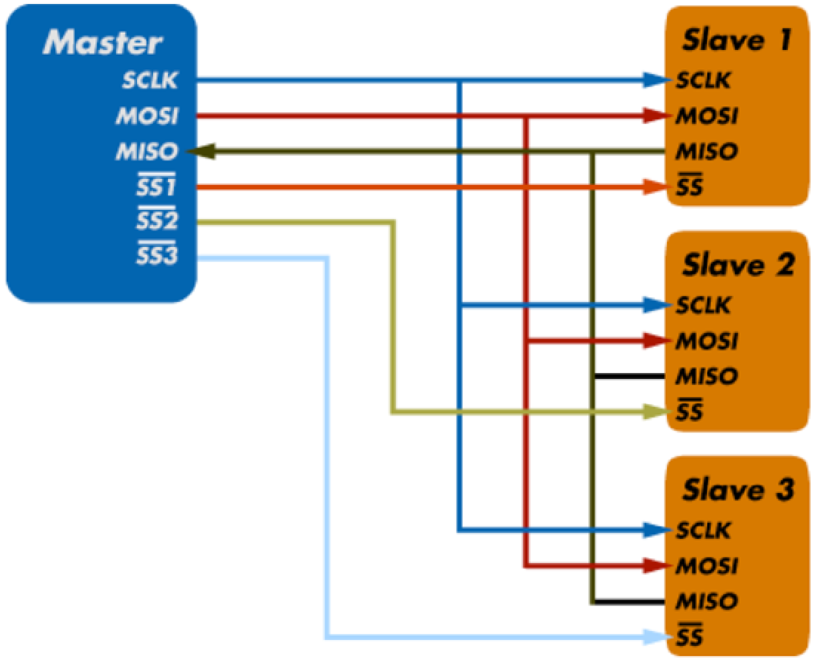
A Serial Peripheral Interface (SPI) facilitates short-distance communication between peripheral integrated circuits and microcontrollers.

SPI stands for Serial Peripheral Interface. It is a protocol that is synchronous serial communication. It is used to communicate between the peripheral devices i.e. input and output devices and microcontrollers. It is allowed to transfer high-speed data. It is popular with digital communication applications and embedded systems. SPI can transfer the data and receive data from one device to another device at a time.

Components of SPI

Serial Peripheral Interface (SPI) is the process of synchronous serial communication protocol. It is mainly used for connecting the microcontrollers to peripheral devices like sensors, displays, and memory chips. It facilitates the full-duplex, synchronous serial communication between one or more slave devices and a microcontroller.





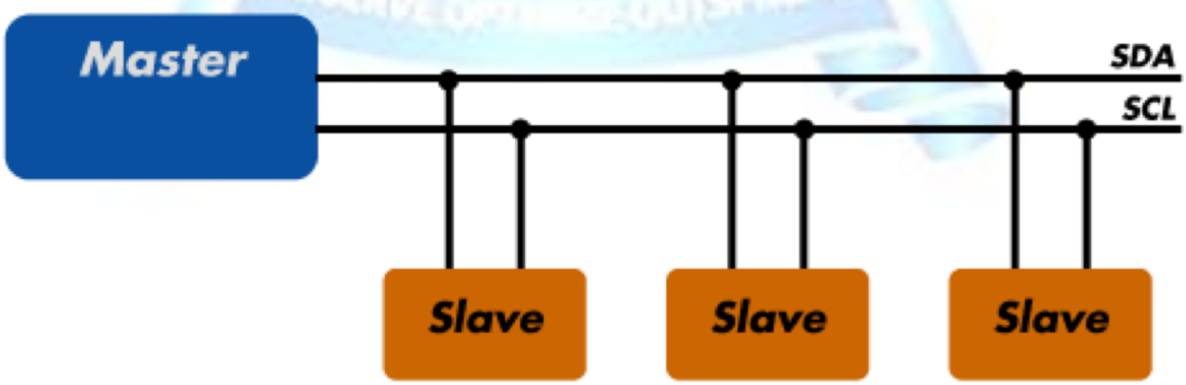
- **Master Device:** The master device is nothing but it controls the process of transformation of data on the SPI bus. It controls the data flow and it generates the clock signal. In most of the applications, the master device is the microcontroller or specialized SPI controller.
- **Slave Device:** Slave devices are peripheral devices that are connected to the SPI bus and controlled by master devices. Every slave device has a different slave select (SS) line, allowing the master to select which device it wants to communicate with.
- **SPI Bus:** SPI bus is a physical connection over the data transferring between the slave devices and the master. It contains four signal lines as below.
    - **Slave Select (SS):** In Slave Select, each slave device contains a dedicated SS pin. If the master will communicate with the specific slave. Multiple slave devices can be shared with the as same as MOSI, MISO, and SCK lines but it must have separated SS lines.
    - **Master Out Slave In (MOSI):** In Master Out Slave In, MOSI can share the data or information from the master to other slave devices.
    - **Master In Slave Out (MISO):** In Master In Slave Out, MISO can share the data or information from the slave device with the master.
    - **Serial Clock (SCK):** In Serial Clock, this clock signal is used by the master and the slave devices for coordinating the data transfer timings.

- **Data Transfer Protocol:** SPI is used as a synchronous serial communication for simple transferring of data. The data is transferred and received at the same time in full duplex mode. By generating the clock pulses, the master-slave will initialize the transfer of the data. In every clock cycle, one bit of data will be transmitted both from master to slave and from slave to master directions.
- **Data Rate:** The SPI bus can support the different data transferring rates depending upon the master capabilities of the slave devices and the transmission line's length. The data rate is specified in bits per megahertz (MHz) or second (bps).
- **Clock Polarity (CPOL) and Clock Phase (CPHA):** These are used to defined the relationship between the data signals and the clock signals. The data signals are nothing but, MOSI and MISO are called as the data signals. The SCK is called as the clock signal. There are available in four different possible combinations of CPHA settings and CPOL, they are allowing flexible to configuring to the SPI interface for work with the different devices.

I2C COMMUNICATION PROTOCOL

I2C: Inter-Integrated Circuit (also abbreviated I2C or IIC) – synchronous master/slave protocol developed in 1982 by Philips Semiconductors (now NXP) for communication between processors and peripheral ICs.

I2C is a two-wire communication protocol that is commonly used to connect low-speed devices like microcontrollers, I/O interfaces, A/D and D/A converters, EEPROMs, and other peripherals in embedded systems. One of these wires, known as SCL (Serial Clock) carries the clock signal, while the other wire, known as SDA (Serial Data) allows master and slave devices on the bus to send and receive data. The I2C protocol allows for multiple slave devices to be connected to a single master device, or for multiple masters controlling one or more slave devices.



Steps of I2C Data Transmission?

Here are the steps of I2C (Inter-Integrated Circuit) data transmission

- **Start Condition**: The master device sends a start condition by pulling the SDA line low while the SCL line is high. This signals that a transmission is about to begin.
- **Addressing the Slave:** The master sends the 7-bit address of the slave device it wants to communicate with, followed by a read/write bit. The read/write bit indicates whether it wants to read from or write to the slave.
- **Acknowledge Bit (ACK):** The addressed slave device responds by pulling the SDA line low during the next clock pulse (SCL). This confirms that the slave is ready to communicate.
- **Data Transmission:** The master or slave (depending on the read/write operation) sends data in 8-bit chunks. After each byte, an ACK is sent to confirm that the data has been received successfully.
- **Stop Condition:** When the transmission is complete, the master sends a stop condition by releasing the SDA line to high while the SCL line is high. This signals that the communication session has ended.

Features of I2C Communication Protocol

- **Half-duplex Communication Protocol -**
  Bi-directional communication is possible but not simultaneously.
- **Synchronous Communication** -
  The data is transferred in the form of frames or blocks.
- Can be configured in a multi-master configuration.
- **Clock Stretching -**
  The clock is stretched when the slave device is not ready to accept more data by holding the SCL line low, hence disabling the master to raise the clock line. Master will not be able to raise the clock line because the wires are AND wired and wait until the slave releases the SCL line to show it is ready to transfer next bit.
- **Arbitration** -
  I2C protocol supports multi-master bus system but more than one bus can not be used simultaneously. The SDA and SCL are monitored by the masters. If the SDA is found high when it was supposed to be low it will be inferred that another master is active and hence it stops the transfer of data.

- **Serial transmission** -
  I2C uses serial transmission for transmission of data.
- Used for low-speed communication.

Comparison between I2C and SPI Communication Protocols

Features	I2C Communication Protocol	SPI Communication Protocol
Number of wires	2 (SDA and SCL)	4 (MOSI, MISO, SCK, and SS)
Communication type	Half-duplex	Full-duplex
Maximum number of devices	Limited by addressing scheme	Limited by number of chip select (SS) lines
Data transfer speed	Slower	Faster
Error handling	Improved due to ACK/NACK feature	Not as robust
Cost	Cost-efficient due to fewer wires	More expensive due to additional wires
Complexity	Simpler due to fewer wires	More complex due to additional wires
Multi-master configuration	Yes	No
Synchronous communication	Yes	Yes
Clock stretching	Yes	No

Arbitration	Yes	No