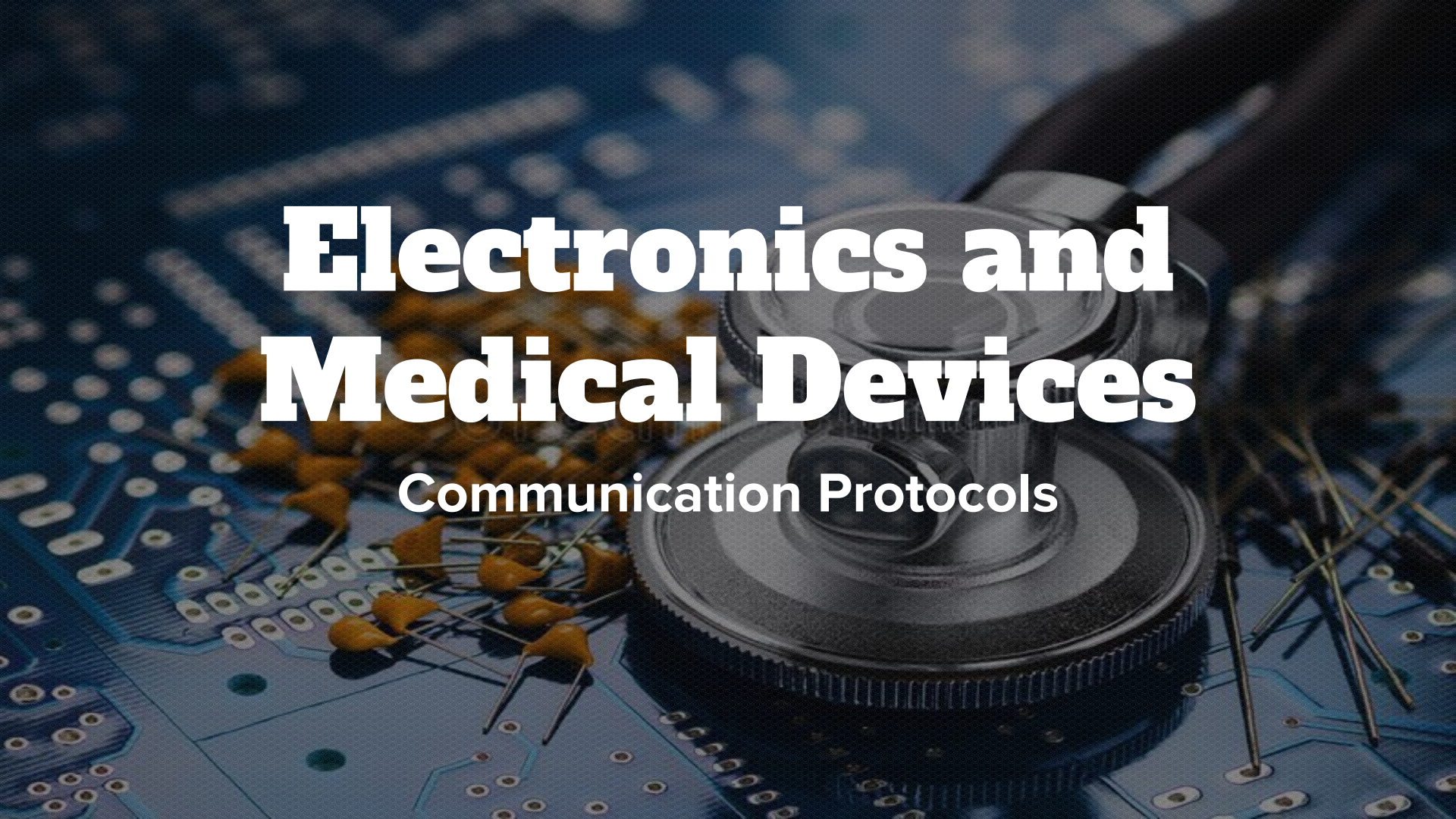


Electronics and Medical Devices

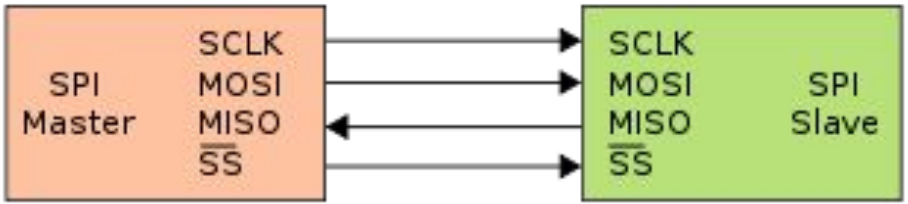
Communication Protocols

Team Members :

(Group 4)

Name	PS Number	Track
Yashwanth K	99004449	Embedded
Malavika M	99004450	Embedded
Tareni Nayak	99004462	Software
Prerna Agrawal	99004467	Software
Sajan Sethi	99004468	Software

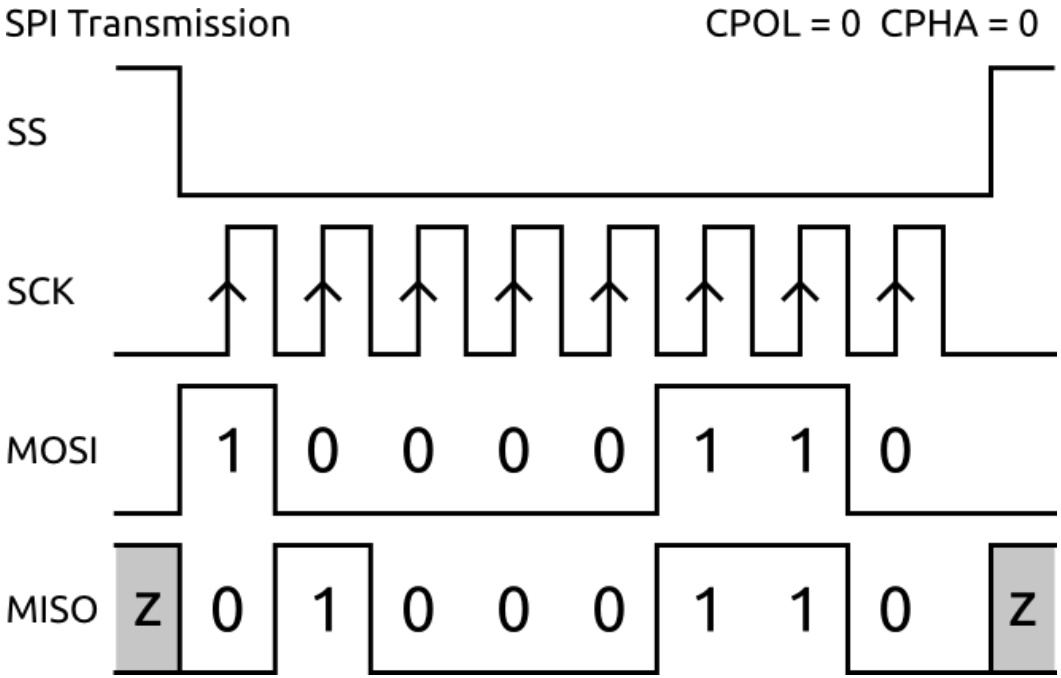
SPI Protocol

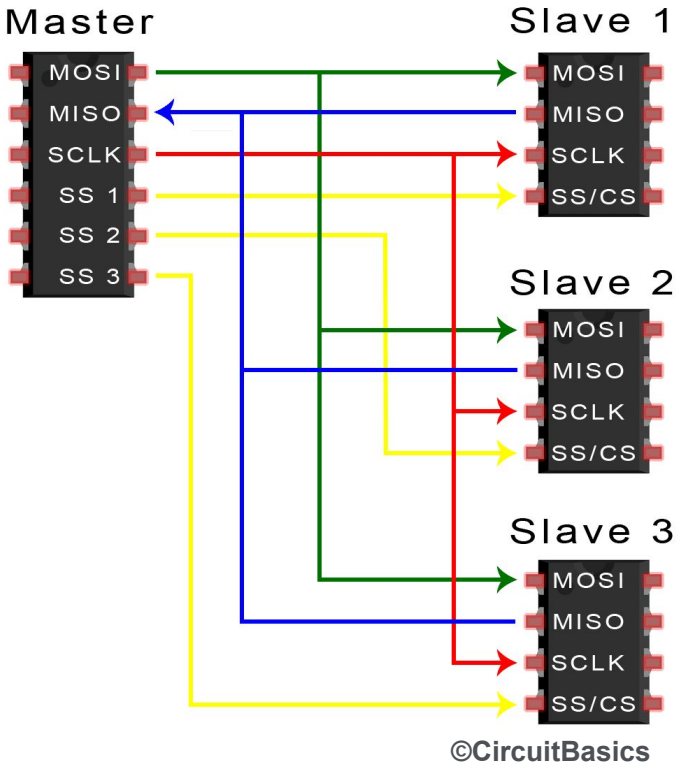


What is SPI ?

Developed by Motorola to provide synchronous serial communication between master and slave. **Serial Peripheral Interface**, or SPI, is a very common communication protocol used for two-way communication between two devices. A standard SPI bus consists of 4 signals, Master Out Slave In (MOSI), Master In Slave Out (MISO), the clock (SCK), and Slave Select (SS). Unlike an asynchronous serial interface, SPI is not symmetric. An SPI bus has one master and one or more slaves. The master can talk to any slave on the bus, but each slave can only talk to the master. Each slave on the bus must have it's own unique slave select signal. The master uses the slave select signals to select which slave it will be talking to.

SPI Communication Protocol :





Advantages and disadvantages :

Advantages	Disadvantages
• It has very simple hardware interfacing.	• It supports only one master device. No inter-slave communications.
• Not limited to any maximum clock speed, enabling potentially high speed	• No error checking protocol.
• It's faster and requires low power than I2C.. Ex: SPI 10-20Mbps, I2C 3.4Mbps	• SPI usually requires separate SS lines to each slave, which can be problematic if numerous slaves are needed.
• It supports multiple slaves	

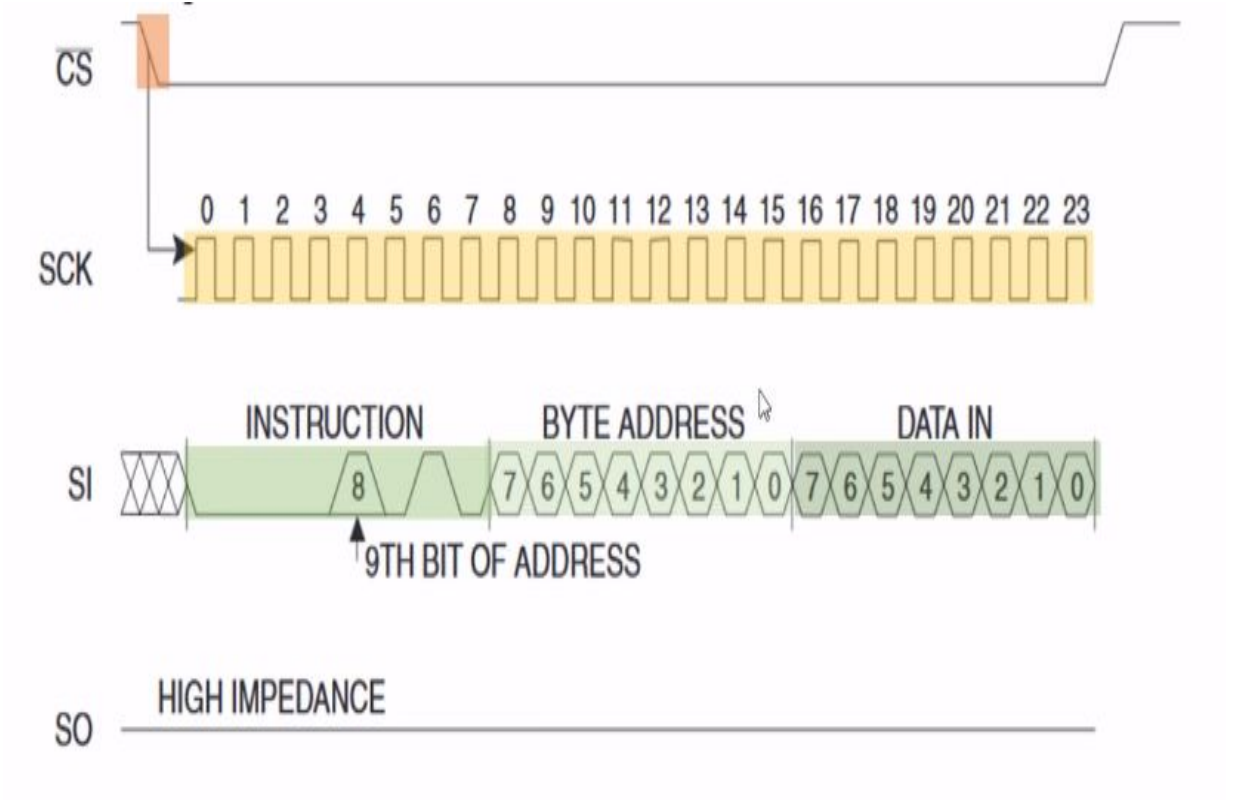
Read/write(data transmission) protocols of SPI :

Selecting slave (SS->0)---->

Clock Signal ---->

SI ---->
Instruction 10 signifies **write**
Byte Address (MSB-Left)
Data to be sent (MSB-Left)

SI ---->
Not reading anything in this case



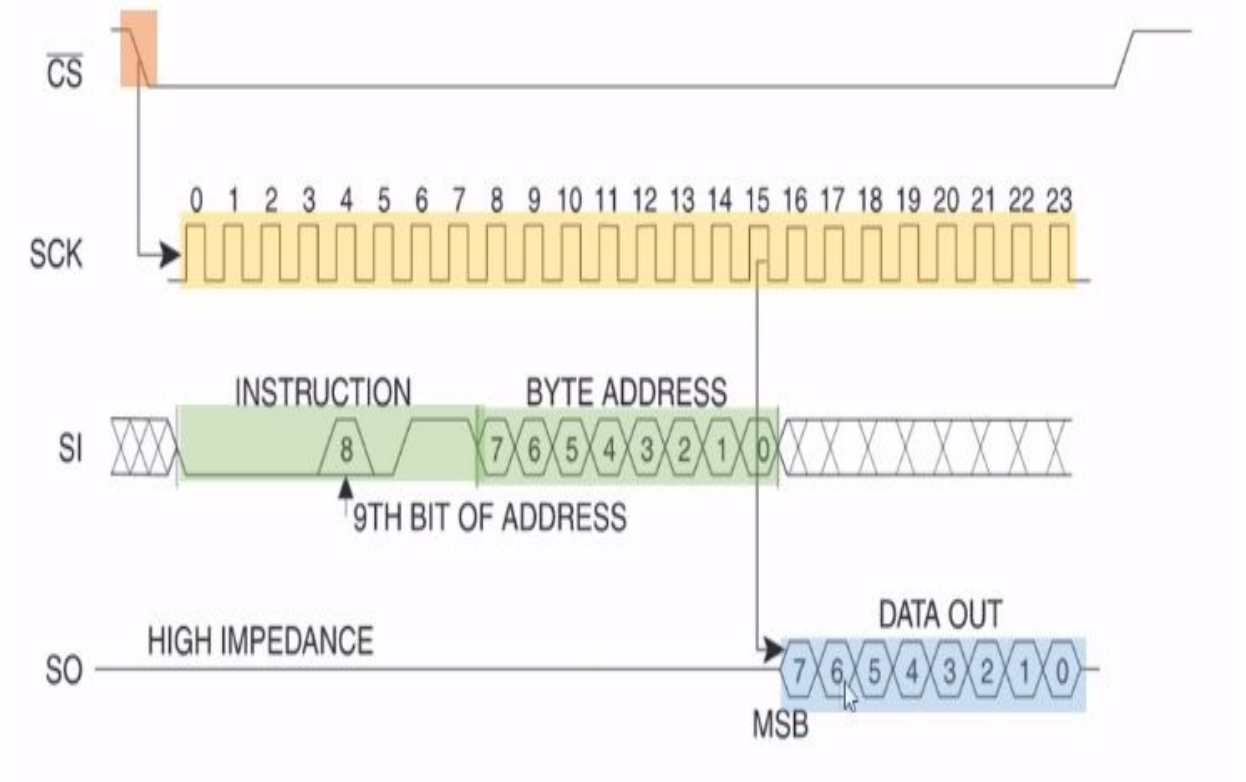
Read/write(data transmission) protocols of SPI :

Selecting slave (SS->0)---->

Clock Signal ---->

SI ---->
Instruction 11 signifies **read**
Not sending anything in this case
Byte Address (MSB-Left)
Data to be sent (MSB-Left)

SO ---->
Output data



Modes of SPI :

CPOL - Clock Polarity

CPOL - 0 (Active state = 1)

CPOL - 1 (Active state = 0)

CPHA - Clock Phase

CPHASE - 0 (Data sampling a leading edge)

CPHASE - 1 (Data sampling a trailing edge)

	Clock Starting Position	Data is received on the
SPI Mode 0	Low	Falling edge of the clock
SPI Mode 1	Low	Rising edge of the clock
SPI Mode 2	High	Falling edge of the clock
SPI Mode 3	High	Rising edge of the clock

Applications of SPI :

It is used in MMC or SD card.

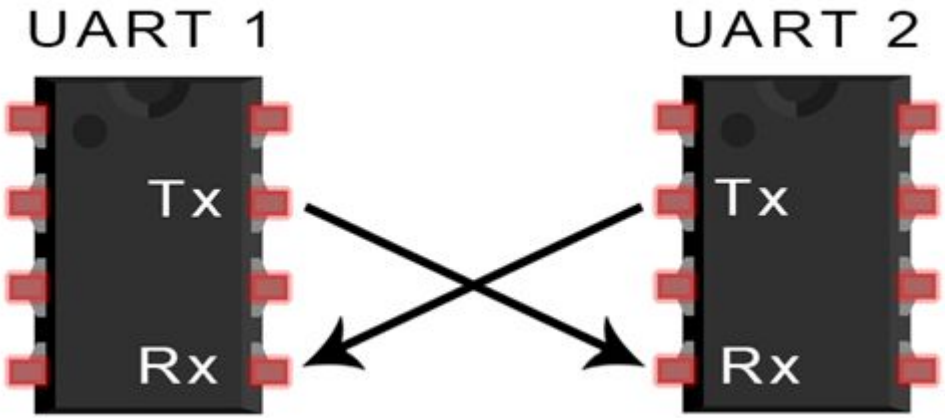
It is is used to talk to a variety of peripherals such as sensors i.e. temperature and pressure, analog to digital converter (ADC), digital to analog converter (DAC).

It is used in real time clocks.

It is used in LCD and LED.

It is also used in video games.

UART Protocol



UART Protocol

- Stands for universal asynchronous receiver-transmitter
- Device-to-device communication protocol
- Main purpose is to transmit and receive serial data
- Uses only 2 wires to transmit data between devices

Working of UART

- Here 2 UARTs communicate directly with each other
- Transmitting UART converts parallel data from a controlling device into serial form
- Transmits it in serial to receiving uart
- Converts the serial data back to parallel data
- Data flows from Tx pin of transmitting UART to Rx pin of receiving UART.

Working

- UART transmit data asynchronously which means there is no clock signal to synchronize the outputs of bits from transmitting UART
- Instead of a clock signal, transmitting UART adds start and stop bits to data packets being transferred

Wires Used	2
Maximum Speed	Any speed up to 115200 baud, usually 9600 baud
Synchronous or Asynchronous?	Asynchronous
Serial or Parallel?	Serial
Max # of Masters	1
Max # of Slaves	1

Advantages

- Only uses two wires
- No clock signal is necessary
- Has a parity bit to allow for error checking
- The structure of the data packet can be changed as long as both sides are set up for it
- Well documented and widely used method

Disadvantages

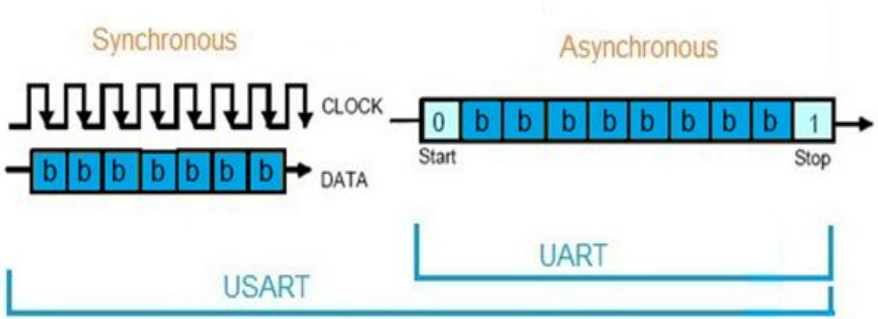
- The size of the data frame is limited to a maximum of 9 bits
- Doesn’t support multiple slave or multiple master systems
- The baud rates of each UART must be within 10% of each other

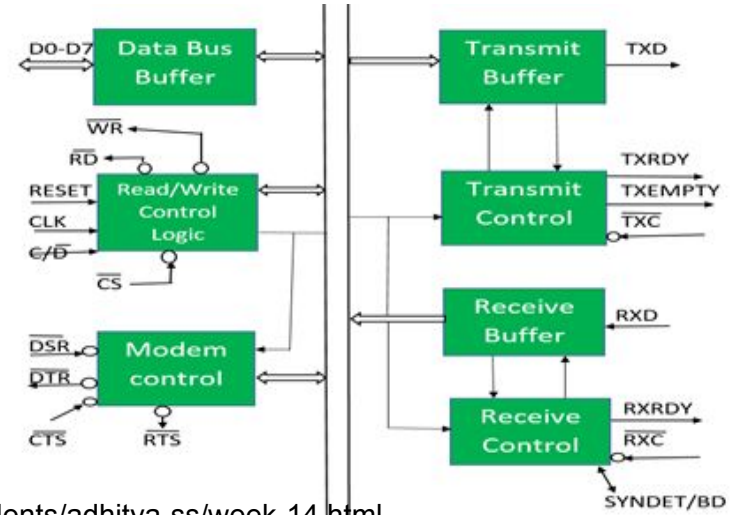
Applications of UART

- **Debugging**: Early detection of system bugs is important during development. Adding UART can help in this scenario by capturing messages from the system.
- **Manufacturing function-level tracing**: Logs are very important in manufacturing. They determine functionalities by alerting operators to what is happening on the manufacturing line.
- **Customer or client updates:** Software updates are highly important. Having complete, dynamic hardware with update-capable software is important to having a complete system.
- **Testing/verification**: Verifying products before they leave the manufacturing process helps deliver the best quality products possible to customers.

USART Protocol

- USART -Universal Synchronous Asynchronous Receiver Transmitter OR Serial Communications Interface or SCI.
- a serial communication of a two-wire protocol.
- Supports full-duplex protocol ,half duplex, simplex communication
- Application: Telecommunication





Source: http://archive.fabacademy.org/2018/labs/fablabcept/students/adhitya-ss/week-14.html

USB Protocol

- USB - Universal Serial Bus
- introduced in 1996
- Requires Drivers base on functionality
- Max cable length – 4m

- USB protocol uses below pins to function
  - VCC – For power supply
  - GND – To close the circuit
  - D+ – Data transfer differential line
  - D- – Data receive differential line

- 3 modes : 1. slow/Full Speed USB (USB 1.1) → 12 Mbps
  2. High-Speed USB (USB 2.0) → 480 Mbps
  3. SuperSpeed USB → 5 Gbps

How communication takes places over the USB bus ?

USB data packet fields include :

1. a Sync field,
2. a Packet ID (PID) field,
3. ADDR (Address) field
4. ENDP (Endpoint) field
5. CRC (cyclical redundancy check) field,
6. EOP (end of packet) field.

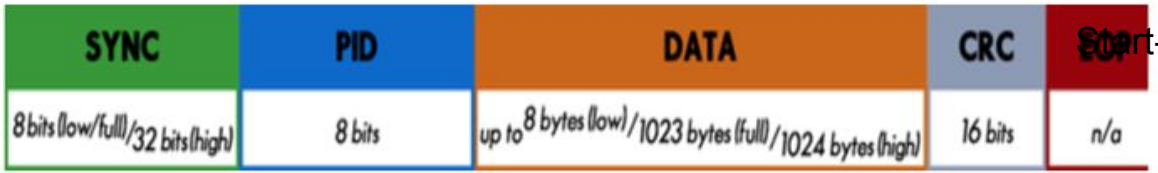
These fields are used to form data packets, which define the various transactions. There are four USB packet types including:

How communication takes places over the USB bus ?

Token Packet

Data Packet

Handshake Packet



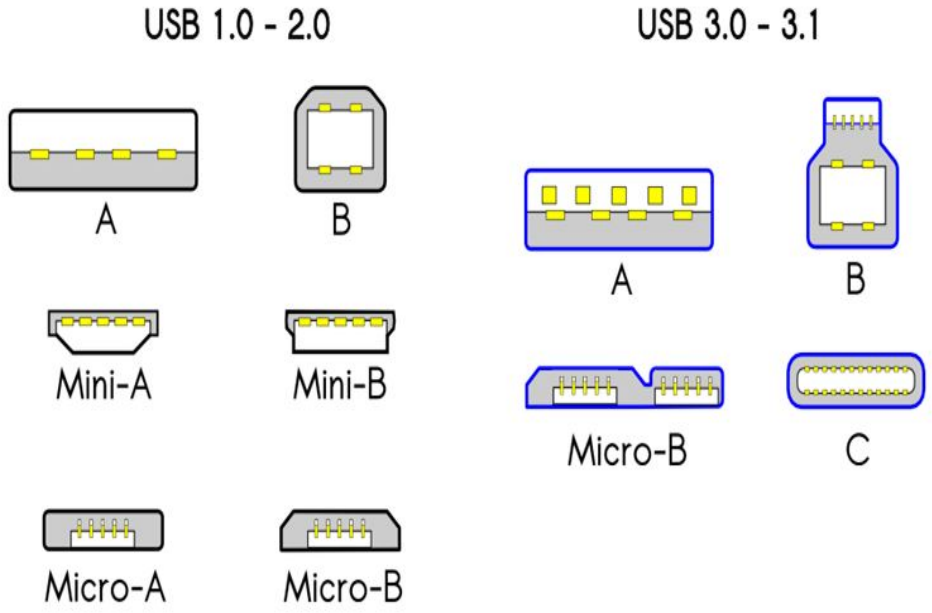
Start-of-Frame Packet

SOURCE: https://www.totalphase.com/blog/2020/07/about-the-usb-protocol-common-usb-bus-errors-and-how-to-troubleshoot-them/

How communication takes places over the USB bus ?

Types of USB Data Transfers:

1. Control
2. Isochronous
3. Interrupt
4. Bulk transfers.



SOURCE:
https://www.totalphase.com/blog/2020/07/about-the-usb-protocol-common-usb-bus-errors-and-how-to-troubleshoot-them/

Referance

https://www.totalphase.com/blog/2020/07/about-the-usb-protocol-common-usb-bus-errors-and-how-to-troubleshoot-them/

https://www.geeksforgeeks.org/microprocessor-8251-usart/

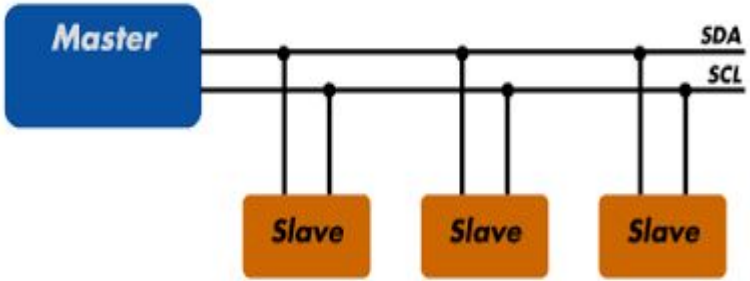
https://www.elprocus.com/communication-protocols/

http://archive.fabacademy.org/2018/labs/fablabcept/students/adhitya-ss/week-14.html

http://ww1.microchip.com/downloads/en/devicedoc/usart.pdf

https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter

I2C Protocol



I2C Communication Protocol

I2C stands for Inter-Integrated Circuit. It is a bus interface connection protocol incorporated into devices for serial communication. It is a widely used protocol for short-distance communication. It is also known as Two Wired Interface(TWI).

I2C Communication Protocol uses only 2 bi-directional open-drain lines for data communication called SDA and SCL. Both these lines are pulled high.

Serial Data (SDA) – Transfer of data takes place through this pin.

Serial Clock (SCL) – It carries the clock signal.

I2C operates in 2 modes –

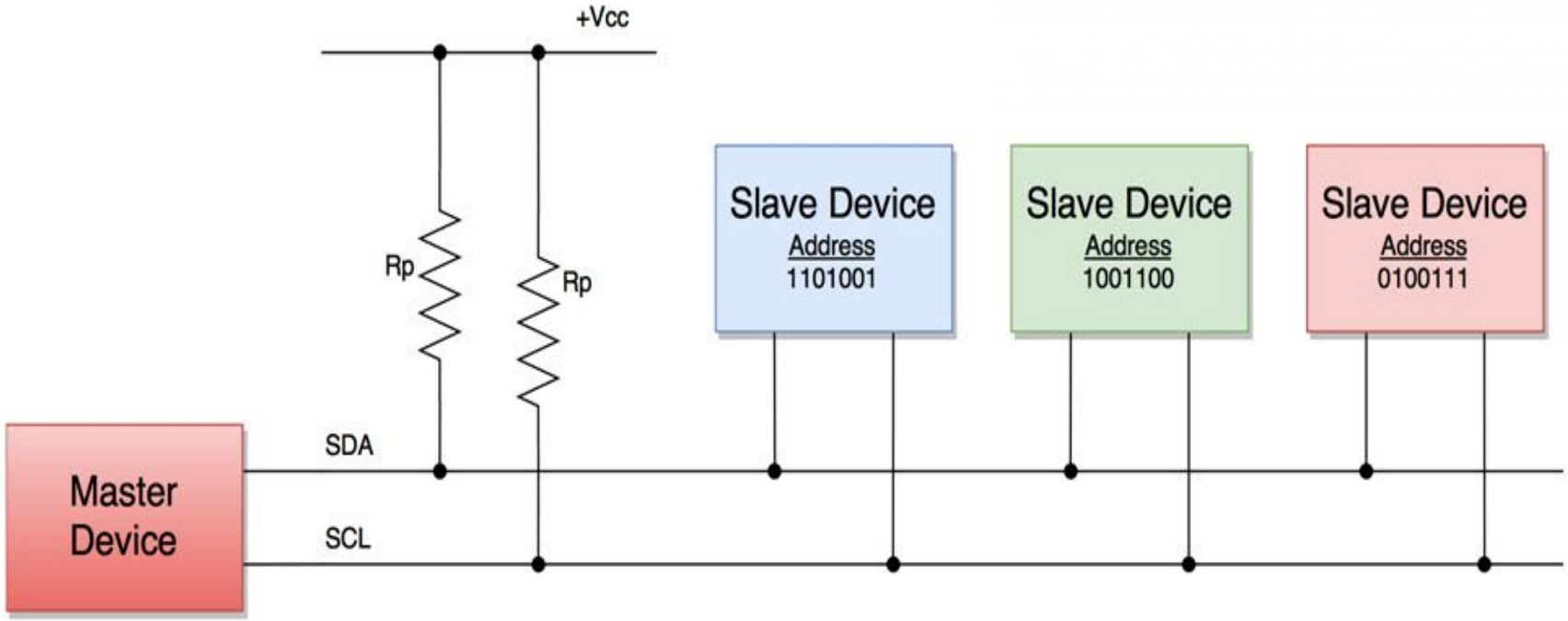
Master mode

Slave mode

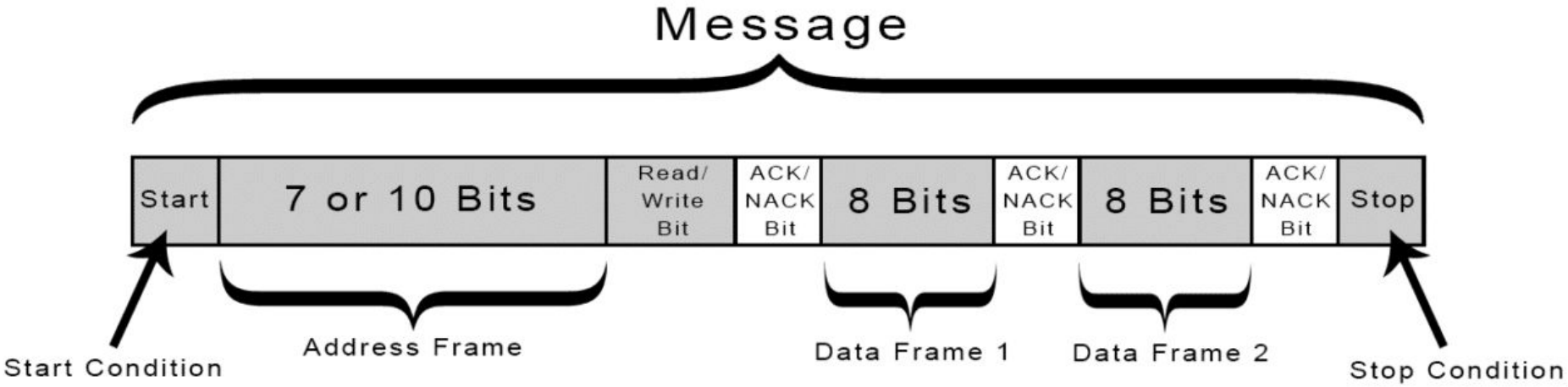
Features

The following are some of the important features of I2C communication protocol:

- Only two common bus lines (wires) are required to control any device/IC on the I2C network
- Simple mechanism for validation of data transfer
- Uses 7-bit addressing system to target a specific device/IC on the I2C bus
- I2C networks are easy to scale. New devices can simply be connected to the two common I2C bus lines



Reference : http://www.electronicshub.org



Reference : http://www.circuitbasics.com

Working Of I2C

Step 1 :

The master sends the start condition to every connected slave by switching the SDA line from a high voltage level to a low voltage level before switching the SCL line from high to low.

Step 2 :

The master sends each slave the 7 or 10 bit address of the slave it wants to communicate with, along with the read/write bit.

Step 3 : Each slave compares the address sent from the master to its own address. If the address matches, the slave returns an ACK bit by pulling the SDA line low for one bit. If the address from the master does not match the slave's own address, the slave leaves the SDA line high.

Step 4 : The master sends or receives the data frame.

Step 5 : After each data frame has been transferred, the receiving device returns another ACK bit to the sender to acknowledge successful receipt of the frame.

Step 6 : After each data frame has been transferred, the receiving device returns another ACK bit to the sender to acknowledge successful receipt of the frame.


Thank You