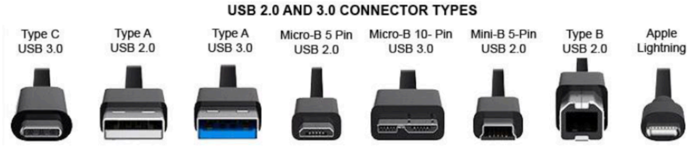
- Before USB, computers used serial ports
    - can still connect over a COM port
- Plug & Play protocol

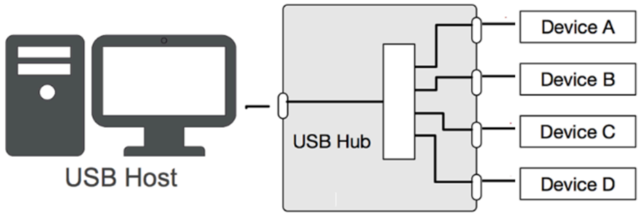
USB A 2.0 && USB A 3.0



Property	USB-A 2.0	USB-A 3.0
Speed	480Mbps	5 Gbps
Physical	black/white	blue
Power	500mA (2.5W)	900mA (4.5W)
Backward Compatibility	USB-A 3.0 ports accept 2.0 but at 2.0 speeds	accepts 3.0, at 2.0 speed

USB Workings

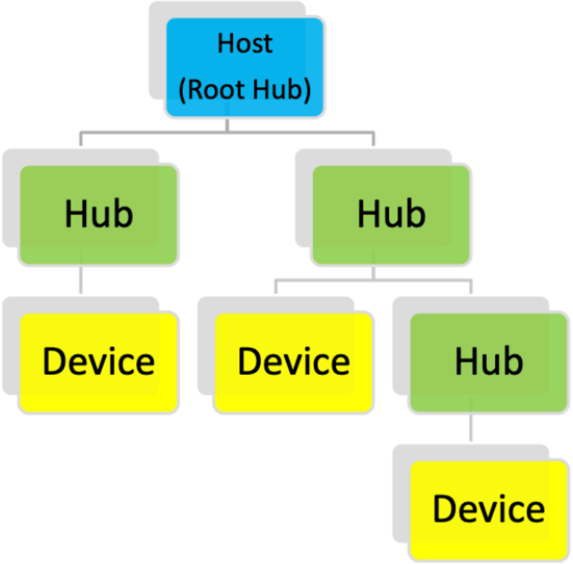
- **host** to **device** protocol
- host controls all communication
    - USB 1.1 - only host OR a device can be communicating at a time (half duplex)
        - one dev at a time can transmit to host
    - USB 2.0 - full duplex



USB 1.1

USB topology

- USB bus is setup to work like a tree
  - Supports up to 127 USB devices using ‘hubs’.
  - Host packets are sent to every device, downward via each hub.
  - Only the device with the correct address accepts the data.
- From device to host, each hub repeats data from a ‘downstream’ device.
- Still only the host or ONE device transmitting data at a time.



Connecting USB to Device

When a device is attached

I. Host resets device

II. Assigns an address to device

III. Enumerates device

1. Determines USB version
2. Retrieves device descriptor
   - ‘Human readable information’
   - E.g., USB version number, Vendor ID, product ID, Serial number
3. Retrieves configuration descriptor
   - Hardware specific configurations
   - E.g., maximum power supported
4. Loads the corresponding device driver

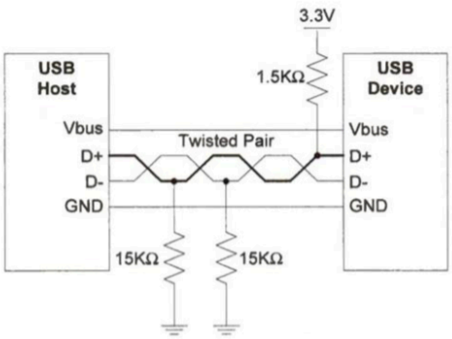
USB - Physical Layer

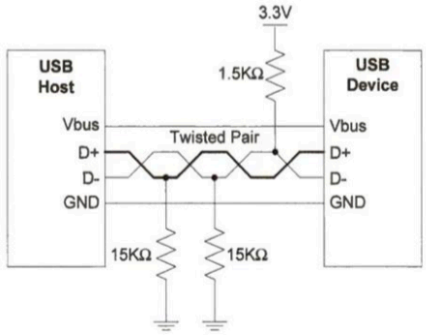
USB – physical layer

- Uses differential signals to send data and !data
  - Labelled as D+ and D-
  - But !data is not always inverse of data!

- Logic 1 is 2.8V to 3.3V and Logic 0 is 0V to 0.3V.
- 4 wires, twisted to prevent EM interference.

- On the host, both D+ and D- use 15kΩ pull-down resistors.
- On device, 1.5k pull-up on either D+ or D-
  - D+ indicates full-speed and D- indicates low-speed



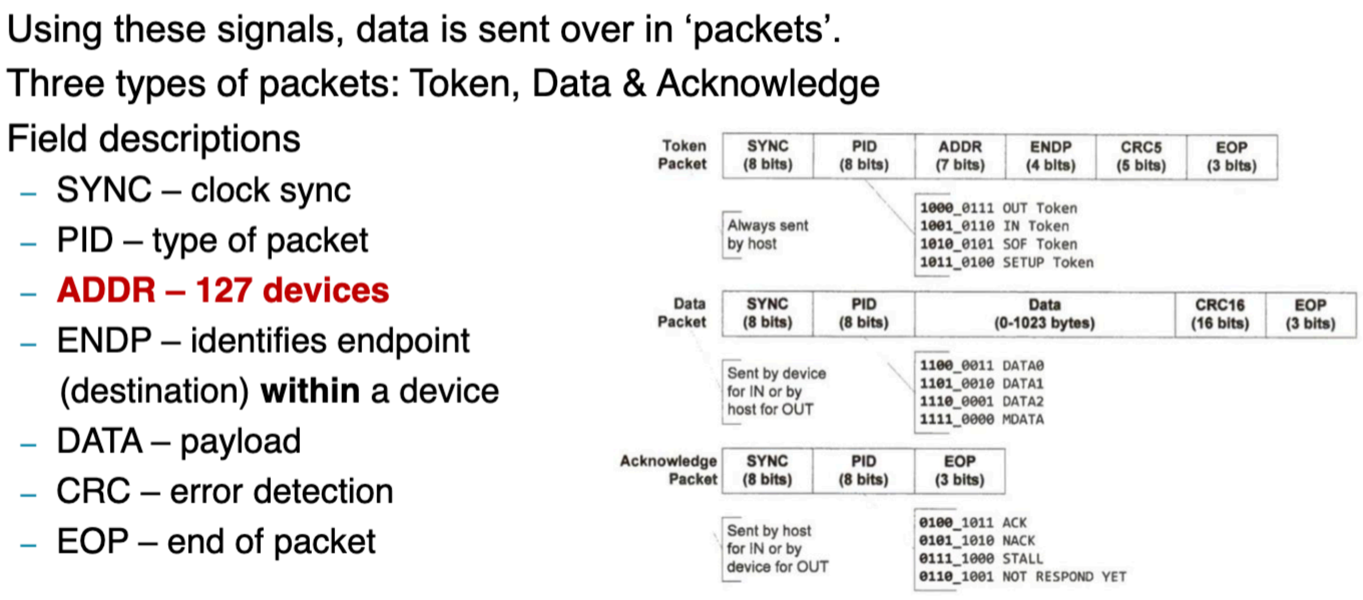


- Using just D+ and D-, USB creates different signals.
- This changes for ‘low-speed’ and ‘full-speed’ so we use these named states.

	Low-speed	Full-speed
J state	Differential ‘0’	Differential ‘1’
K state	Differential ‘1’	Differential ‘0’
Idle state	D- high, D+ low	D+ high, D- low

Signal type	Bus line behavior
Differential '1'	D+ high, D- low
Differential '0'	D- high, D+ low
Single Ended Zero (SE0)	D+ and D- low
Single Ended One (SE1)	D+ and D- high
Resume State	Data K state
Start of Packet (SOP)	Switch from idle to K state
End of Packet (EOP)	SE0 for 2-bit times followed by J state for 1 bit time
Disconnect	SE0 for >= 2us
Connect	Idle for 2.5us
Reset	SE0 for >= 2.5 us

USB Packets



Using these signals, data is sent over in 'packets'.

Three types of packets: Token, Data & Acknowledge

Field descriptions

- SYNC – clock sync
- PID – type of packet
- **ADDR – 127 devices**
- ENDP – identifies endpoint (destination) **within** a device
- DATA – payload
- CRC – error detection
- EOP – end of packet

Token Packet	SYNC (8 bits)	PID (8 bits)	ADDR (7 bits)	ENDP (4 bits)	CRC5 (5 bits)	EOP (3 bits)

Always sent by host

```
1000_0111 OUT Token
1001_0110 IN Token
1010_0101 SOF Token
1011_0100 SETUP Token
```

Data Packet	SYNC (8 bits)	PID (8 bits)	Data (0-1023 bytes)	CRC16 (16 bits)	EOP (3 bits)

Sent by device for IN or by host for OUT

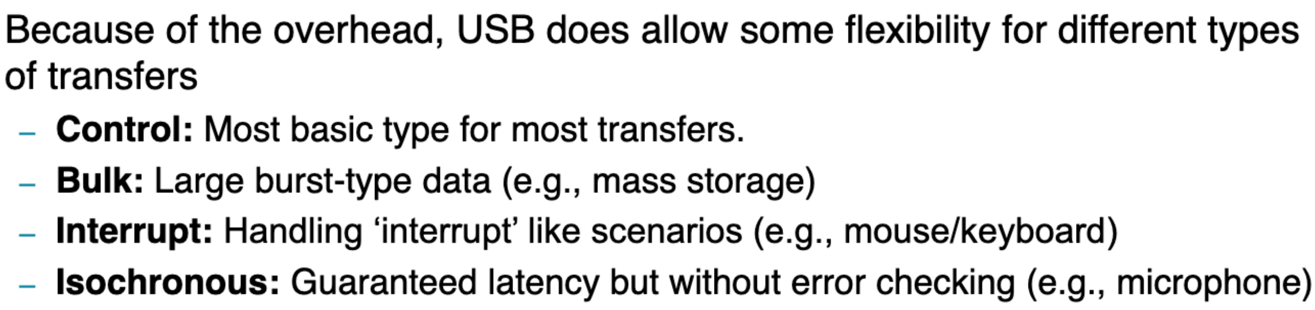
```
1100_0011 DATA0
1101_0010 DATA1
1110_0001 DATA2
1111_0000 MDATA
```

Acknowledge Packet	SYNC (8 bits)	PID (8 bits)	EOP (3 bits)

Sent by host for IN or by device for OUT

```
0100_1011 ACK
0101_1010 NACK
0111_1000 STALL
0110_1001 NOT RESPOND YET
```

USB - Transfer Types



Because of the overhead, USB does allow some flexibility for different types of transfers

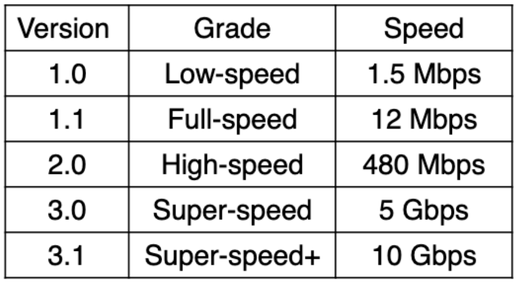
- **Control:** Most basic type for most transfers.
- **Bulk:** Large burst-type data (e.g., mass storage)
- **Interrupt:** Handling 'interrupt' like scenarios (e.g., mouse/keyboard)
- **Isochronous:** Guaranteed latency but without error checking (e.g., microphone)

CRC (Cyclic Redundancy Check)

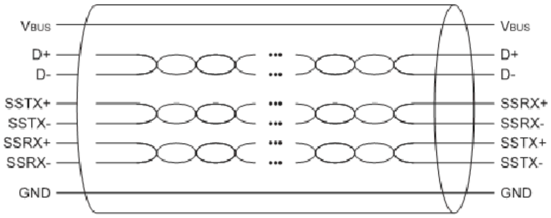
- Robust form of error checking
  - USB uses CRC to ensure data received correctly
- CRC is specified using the number of bits used.
  - More CRC bits means more errors can be detected.
  - USB uses CRC5 and CRC16
- Parity bit in UART is CRC1

USB 2.0 +

- USB 2.0 adds high-speed
  - higher speed configured via packets
- USB 3.0 uses different cable
  - adds 2 more pairs of signals
  - supports full duplex
  - switches to 8b/10b encoding



Version	Grade	Speed
1.0	Low-speed	1.5 Mbps
1.1	Full-speed	12 Mbps
2.0	High-speed	480 Mbps
3.0	Super-speed	5 Gbps
3.1	Super-speed+	10 Gbps



8b/10b Encoding

USB 3.0 uses ‘clock recovery’, where the clock is inferred from the data.

- Based on data switching 0 <-> 1.
- But if there are long strings of 0s or 1s, the clock can get out of sync.
- USB 3.0 uses 8b/10b encoding to avoid this.

Special type of encoding used to balance number of 0s and 1s

- Encodes 8-bit data using 10-bits.
- Long strings of 0s or 1s are represented using a ‘balanced’ string of 0s and 1s.

Also used in SATA, PCIe, Gigabit Ethernet, display port etc.