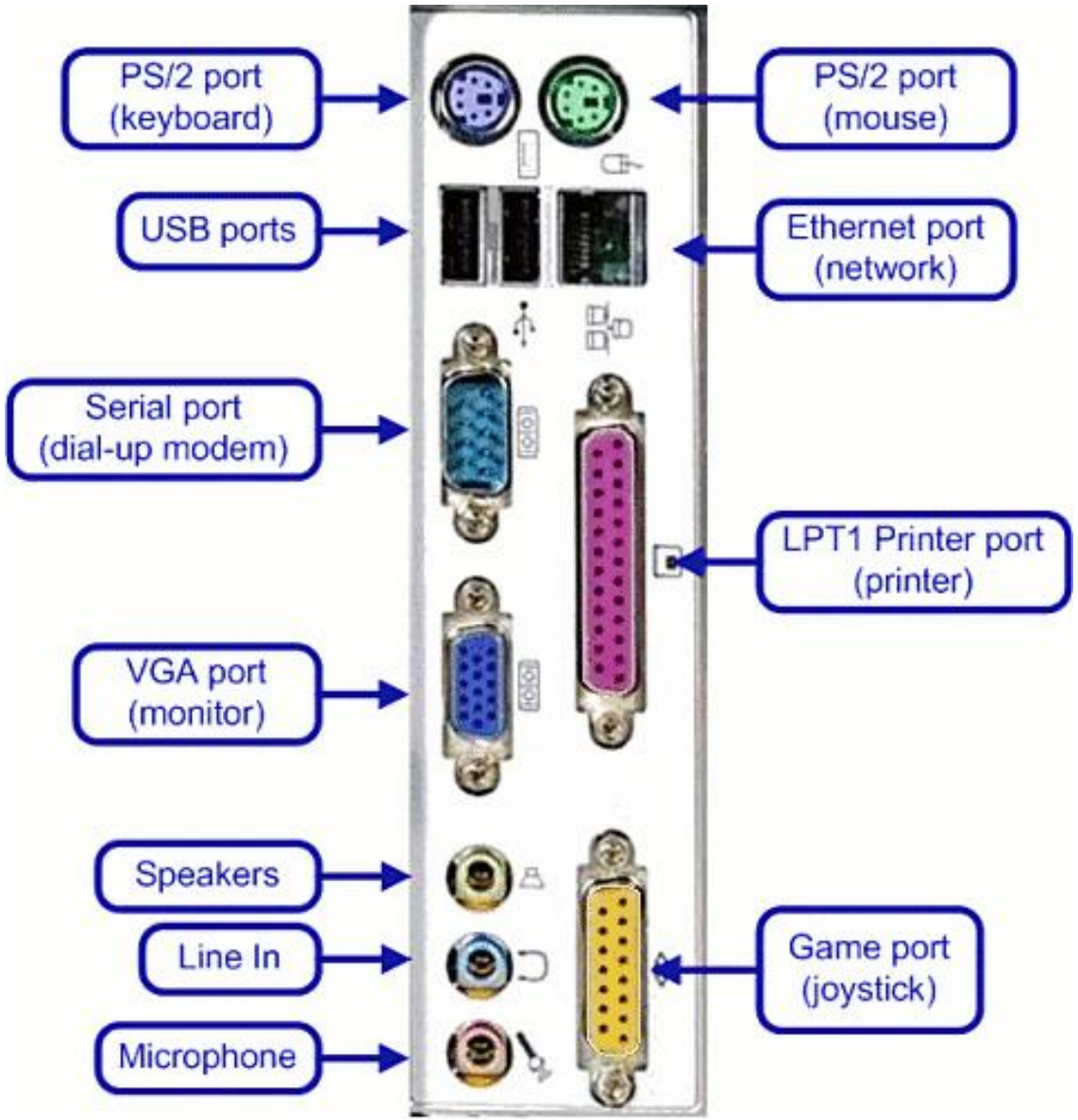
Serial Communication
RS-232

Serial Communication RS-232



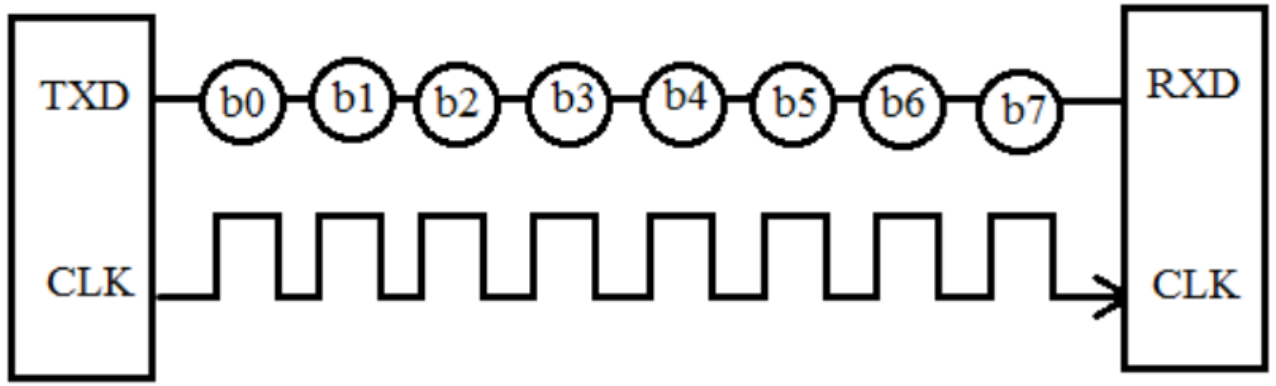
VGA cable

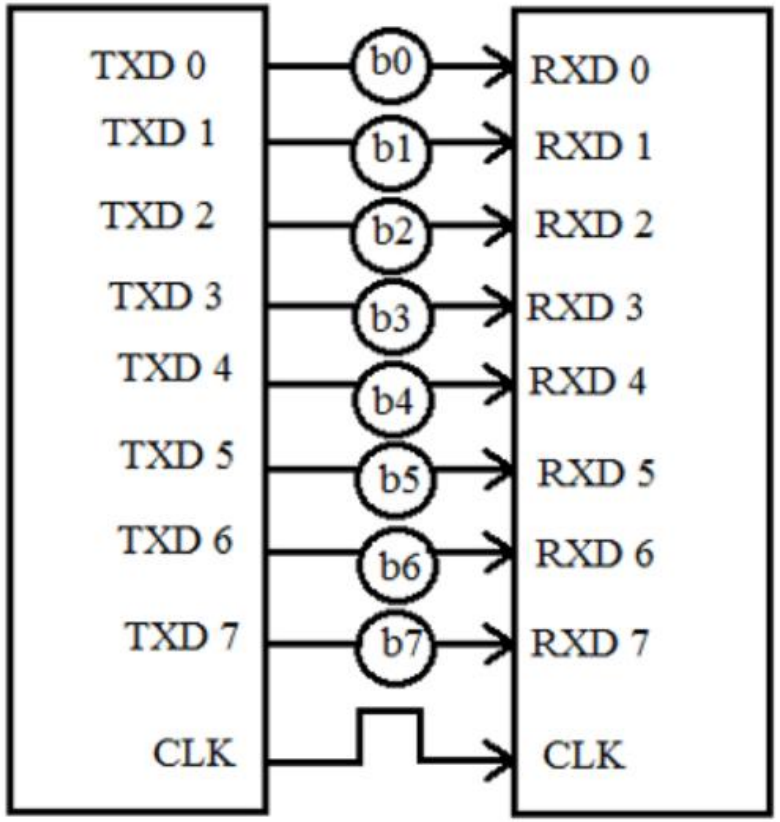
VGA connector

Serial i/o Interface

- Serial data transmission is used for digital communication between
  - Sensors and computers
  - Computers and computers
  - Computers and peripheral devices (projector, printer, stylus, mouse, ..)
  - One of the most widely used communication techniques to interface external equipment
- The process of sending data sequentially over a computer bus is called as **serial communication**, which means the data will be transmitted bit by bit.
- While in parallel communication the data is transmitted in a byte (8 bit) or character on several data lines or buses at a time.
- Serial communication is slower than parallel communication but used for long data transmission due to **lower cost** and practical reasons.

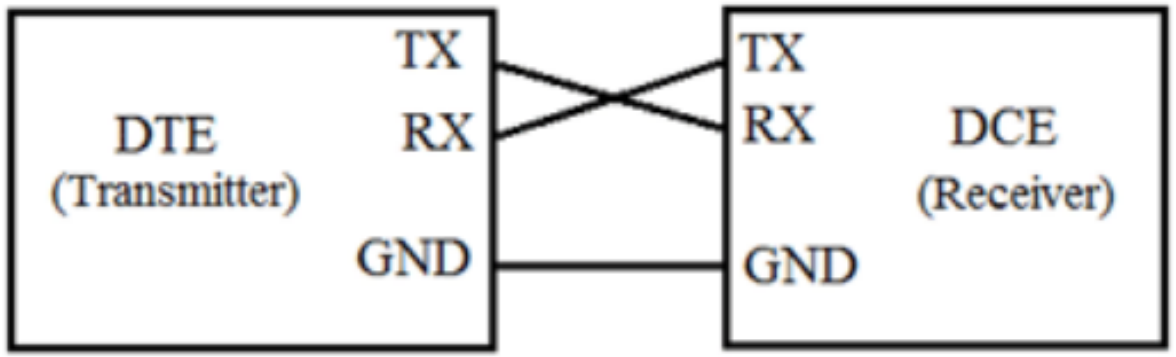
Serial i/o vs parallel i/o Interface





RS232

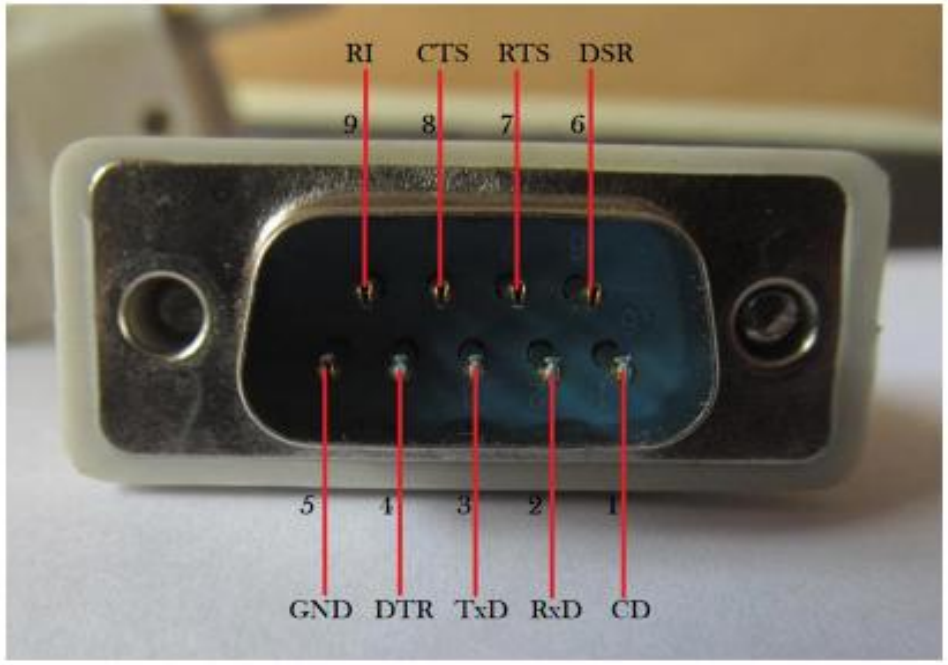
- RS232 is a standard protocol used for serial communication, used for connecting computer and its peripheral devices to allow serial data exchange between them.
- It is used in serial communication up to 50 feet with the rate of 1.492kbps.
- RS232 is used for connecting **Data Transmission Equipment (DTE)** and **Data Communication Equipment (DCE)**.

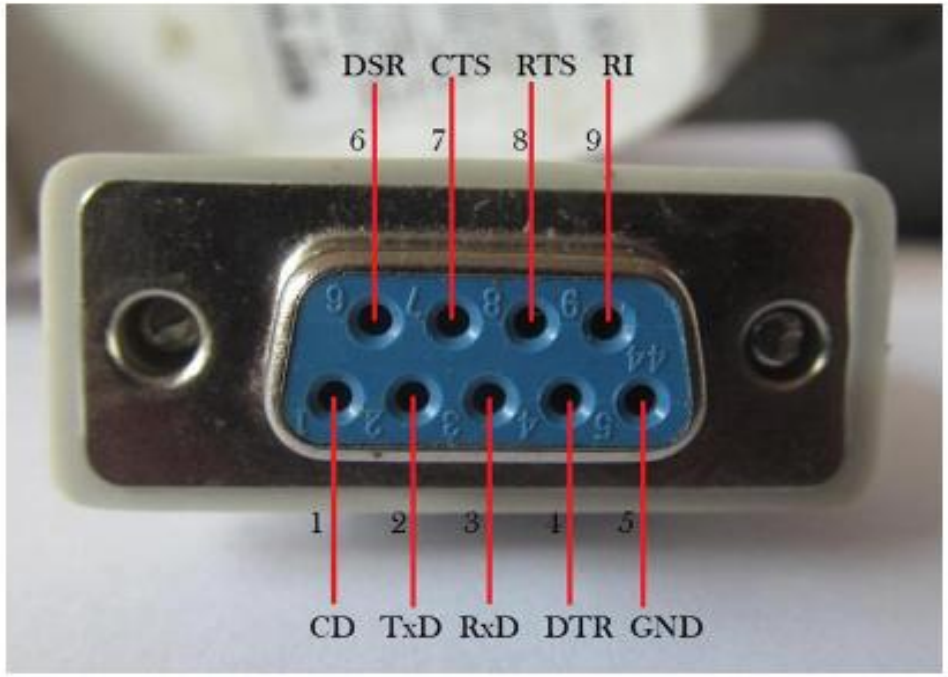


RS232 (DB9 connector)

Pin Description DB-9 Connector:

PIN No.	Pin Name	Pin Description
1	CD (Carrier Detect)	Incoming signal from DCE
2	RD (Receive Data)	Receives incoming data from DTE
3	TD (Transmit Data)	Send outgoing data to DCE
4	DTR (Data Terminal Ready)	Outgoing handshaking signal
5	GND (Signal ground)	Common reference voltage
6	DSR (Data Set Ready)	Incoming handshaking signal
7	RTS (Request to Send)	Outgoing signal for controlling flow
8	CTS (Clear to Send)	Incoming signal for controlling flow
9	RI (Ring Indicator)	Incoming signal from DCE





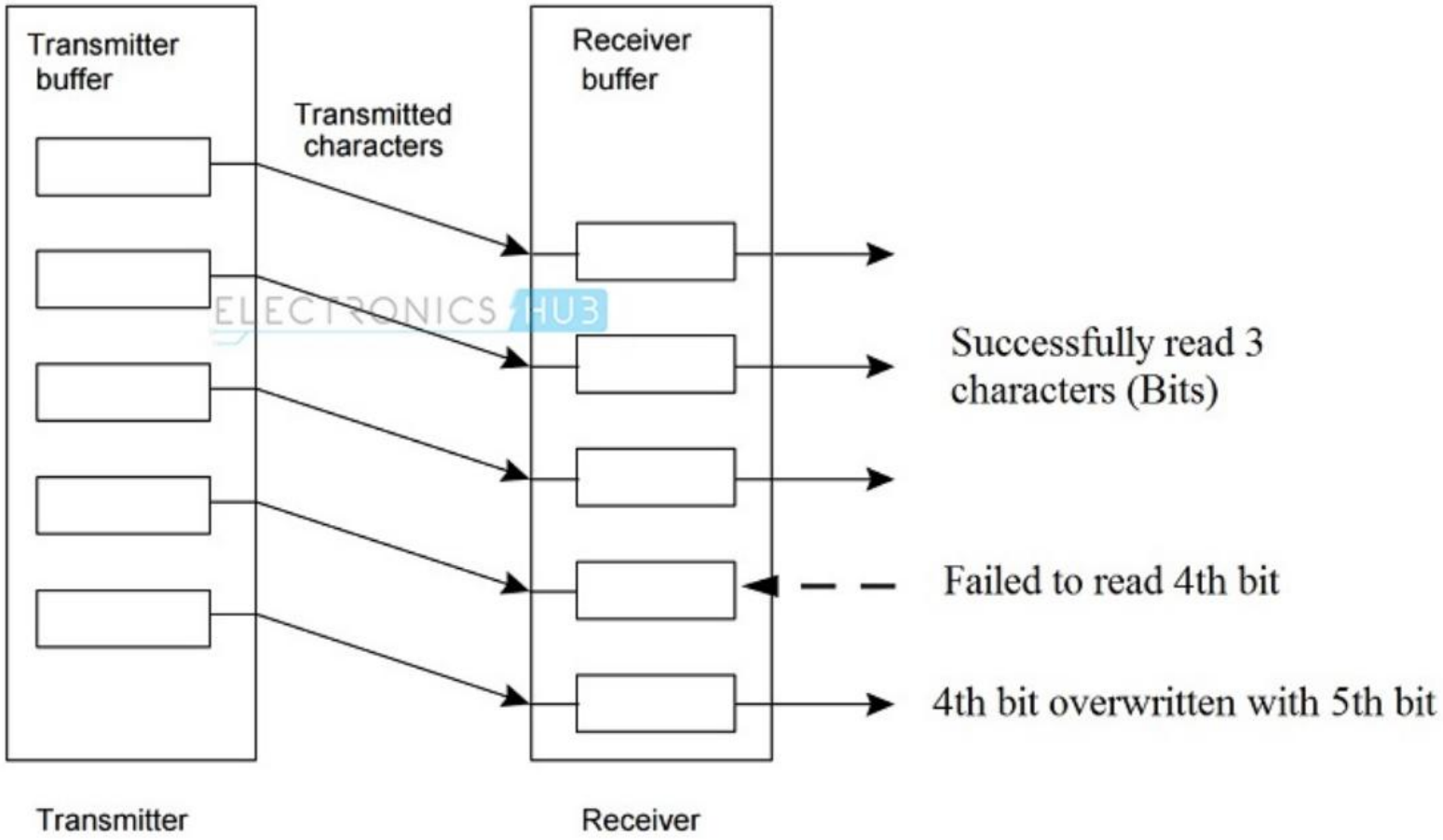
Handshaking

- Handshaking is a process of dynamically setting the parameters of a communication between the transmitter and receiver before the communication begins.
- The need for handshaking is dictated by the speed at with the transmitter (DTE) transmits the data, the speed at which the receiver (DCE) receives the data and the rate at which the data is transmitted.

NO Handshaking

- If handshaking is not used, then the receiver (DCE) must read the data that is already received by it before the transmitter (DTE) sends the next data.
- For this, the receiver uses a special memory location called Buffer and since it is used at receiver end, it is called Receiver Buffer.
- The received data is stored in the buffer before it is read by the receiver.
- The Receiver Buffer can typically store a single bit of data and this data must be cleared (read) before the next data arrives
- If it is not cleared, the existing data will be overwritten with the new data.

NO Handshaking



NO Handshaking

- To avoid situations like this, we need some sort of Handshaking mechanism (either Software or Hardware Handshaking).

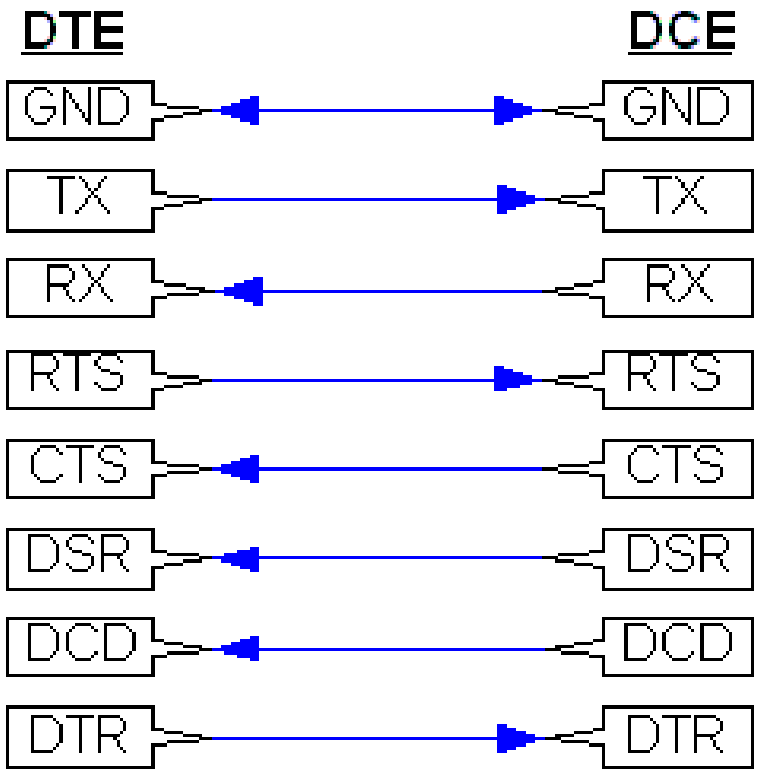
hardware Handshaking

- In Hardware Handshaking, the transmitter first asks the receiver whether it is ready to receive the data.
- The receiver then checks its buffer and if the buffer is empty, it will then tell the transmitter that it is ready to receive.
- The transmitter will transmit the data and it is loaded into the receiver buffer.
- During this time, the receiver tells the transmitter not to send any further data until the data in the buffer has been read by the receiver.

hardware Handshaking

- The RS232 Protocol defines four signals for the purpose of Handshaking:

1. Ready to Send (RTS)
2. Clear to Send (CTS)
3. Data Terminal Ready (DTR)
4. Data Set Ready (DSR)
5. Data Carrier Detect (DCD)



hardware Handshaking

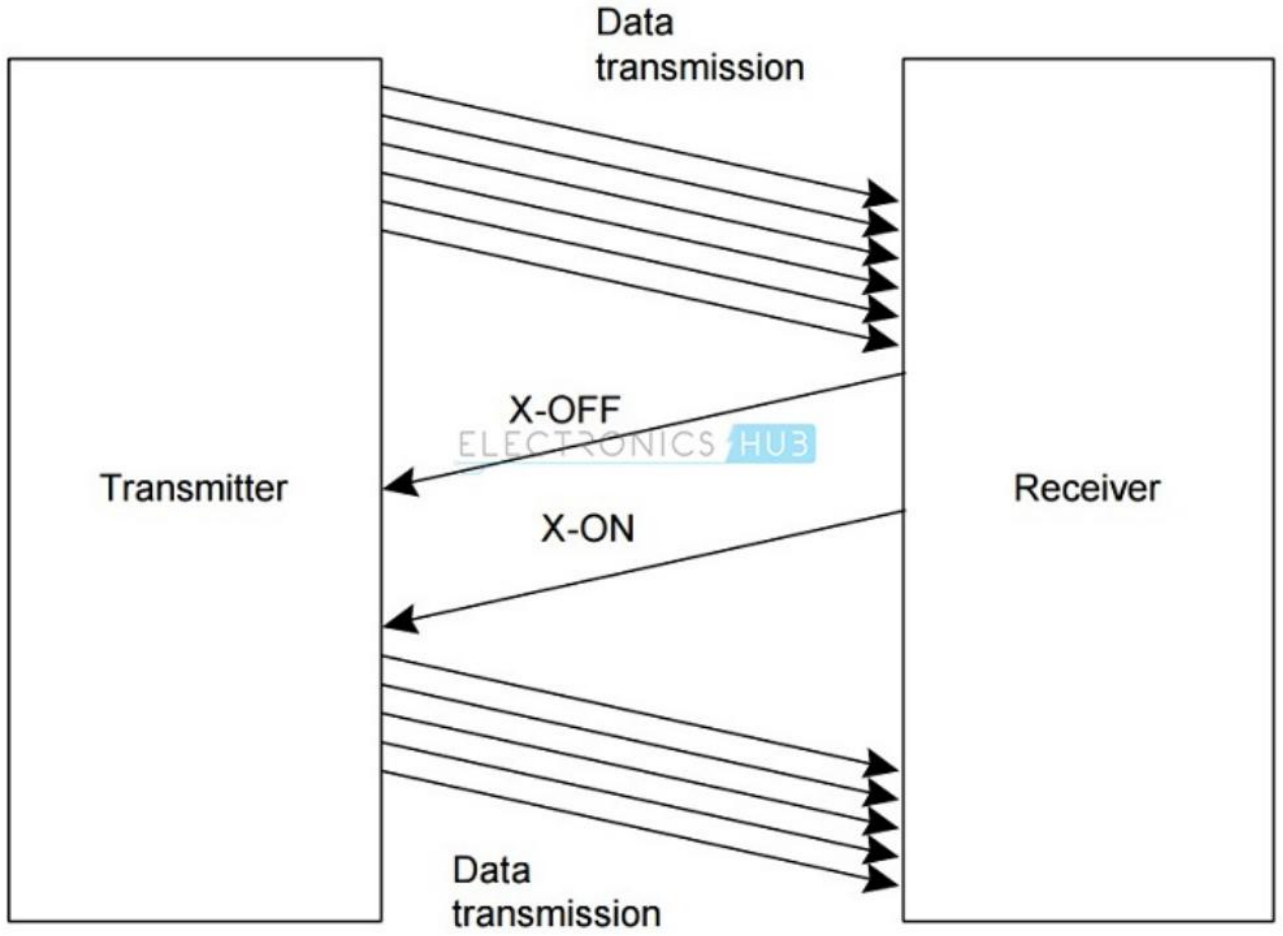
- With the help of Hardware Handshaking, the data from the transmitter is never lost or overwritten in the receiver buffer.
- When the transmitter (DTE) wants to send data, it pulls the RTS (Ready to Send) line to high.
- Then the transmitter waits for CTS (Clear to Send) to go high and hence it keeps on monitoring it.
- If the CTS line is low, it means that the receiver (DCE) is busy and not yet ready to receive data.
- When the receiver is ready, it pulls the CTS line to high.
- The transmitter then transmits the data. This method is also called as RTS/CTS Handshaking.

hardware Handshaking

- Additionally, there are two other wires used in Handshaking.
- They are DTR (Data Terminal Ready) and DSR (Data Set Ready).
- These two signals are used by the DTE and DCE to indicate their individual status.
- Often, these two signals are used in modem communication.

software Handshaking

- Software Handshaking in RS232 involves two special characters for starting and stopping the communication.
- These characters are X-ON and X-OFF (Transmitter On and Transmitter OFF).
- When the receiver sends an X-OFF signal, the transmitter stops sending the data.
- The transmitter starts sending data only after it receives the X-ON signal.



RS232: Working

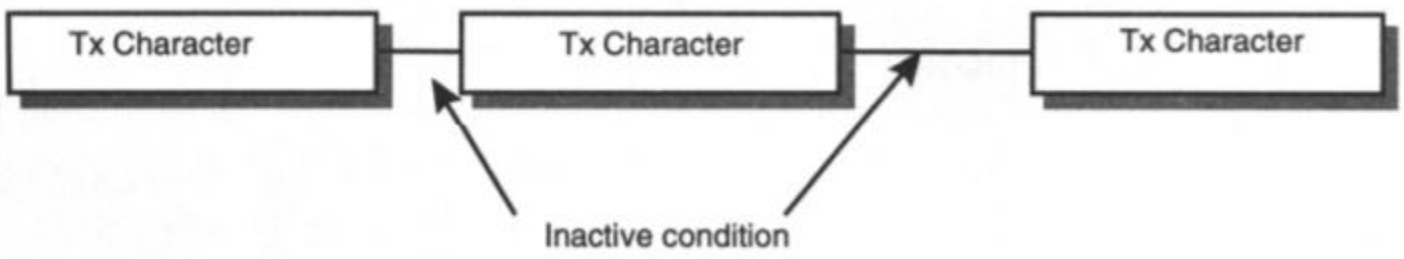
- RS232 works on the two-way communication that exchanges data to one another.
- There are two devices connected to each other, **(DTE) Data Transmission Equipment & (DCE) Data Communication Equipment** which has the pins like **TXD, RXD, and RTS & CTS**.
- Now, from **DTE** source, the **RTS** generates the *request to send* the data.
- Then from the other side **DCE**, the **CTS,** clears the path for receiving the data.
- After clearing a path, it will give a signal to **RTS** of the **DTE** source to send the signal.
- Then the bits are transmitted from **DTE** to **DCE** via the TX and RX lines.

RS232: Working

- Now again from **DCE** source, the request can be generated by **RTS** and **CTS** of **DTE** sources clears the path for receiving the data and gives a signal to send the data.
- Data Carrier Detector is
- This is the whole process through which data transmission takes place.

RS232

- RS232 takes bytes and transmits the individual bits in a sequential fashion in a frame.
- A frame is a defined structure, carrying meaningful sequence of bit or bytes of data.
- It has a start bit followed by 8 data bits, a parity bit and a stop bit.
- **Frame character** is as shown below

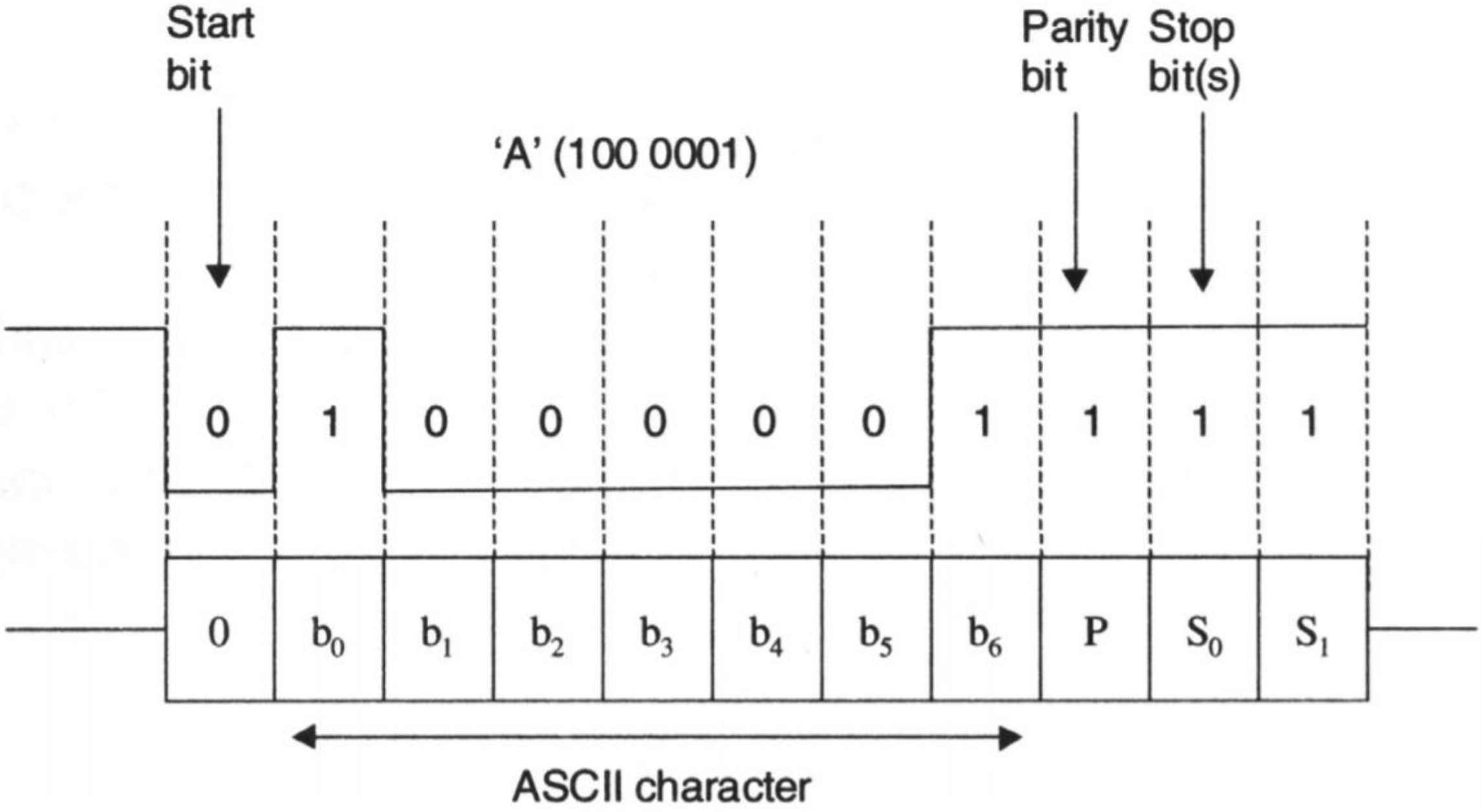


RS232

- Each bit is sent one after the other.
- This mode of transmission requires that receiver is aware when the actual data bits are arriving to synchronize itself with coming data.
- So logic 0 is sent as a start bit. The start bit in the frame signals the receiver that a new character is coming.
- Once the receiver acknowledges, the next five to eight bits are sent which represents the character.
- This is followed by parity bit used for error detection.
- Parity bit is used to specify even or odd number of one’s in the set of bits.
- The stop bit helps the receiver to identify the end of message.

RS232

- If a receiver detects a value other than mark when stop bit should be present, it knows that’s there is synchronization error.
- This causes a framing error condition during reception
- The device then tries to resynchronize on new incoming bits.


Start bit
'A' (100 0001)
Parity bit
Stop bit(s)
0 1 0 0 0 0 0 1 1 1 1
0 b0 b1 b2 b3 b4 b5 b6 P S0 S1
ASCII character

RS232

- Example
- ASCII coding, even parity, 2 stop bits:
- 1111101000001011000000111111111111110000011111110001100 1111010100110111111111
- {inactive}11111 {start bit} 0 {‘A’}1000001 {parity bit} 0 {stop bits} 11 {start bit}0 {‘p’}0000111 {parity bit} 1 {stop bits}11 {inactive}11111111 {start bit}0 {‘p’}0000111 {parity bit} 1 {stop bits}11 {inactive}11 {start bit}0 {‘L’}0011001 {parity bit} 1 {stop bits}11
- Message is ‘AppLe’

limitations

- RS232 Protocol requires a common ground between the transmitter (DTE) and receiver (DCE). Hence, the reason for shorter cables between DTE and DCE in RS232 Protocol.
- The signal in the line is highly susceptible to noise. The noise can be either internal or external.
- If there is an increase in baud rate and length of the cable, there is a chance of cross talk introduced by the capacitance between the cables.
- The voltage levels in RS232 are not compatible with modern TTL or CMOS logics. We need an external level converter.

applications

- Though RS232 is a very famous serial communication protocol, it has now been replaced with advanced protocols like USB.
- Previously they were used for serial terminals like Mouse, Modem etc.
- But, RS232 is still being used in some Servo Controllers, CNC Machines, PLC machines and some microcontroller boards.