Need for Communication Interfaces

Why are communication Interfaces Required in Embedded Systems

- Need to send data to a host. The host will analyze of data and present the data through a Graphical User Interface.
- Need to communicate with another embedded system to transmit/ receive data.
- Need to be networked to share data. Network interfaces need to be provided in such a case.
- Need to be connected to the Internet so that anyone can access the embedded system.
- Mobile devices such as cell phones and palmtops need to interact with other devices such as PCs and laptops for data synchronization.
- Need of s/w upgradation after it is installed in the field. The software can be upgraded through communication interfaces.

Some Communication Interfaces

- **RS 232/UART**
- RS 422, RS 485
- **Universal Serial Bus**
- **Infrared**
- IEEE 1394 Firewire
- Ethernet
- **IEEE 802.11 wireless interface**
- **Bluetooth**

RS232/UART

- RS232 is a standard developed by Electronic Industry Association (EIA).
- One of the oldest and most widely used communication interfaces.
- The PC will have two RS232 ports designated as COM1; and COM2.
- Most of the micro-controllers have an on-chip serial interface.
- The evaluation boards of the processors are also connected to the host system using RS232.
- RS232 is used to connect a DTE (Data Terminal Equipment) to a Data Circuit Terminating Equipment(DCE).
- A DTE can be a PC, serial printer or a plotter.
- DCE can be a modem, mouse, digitizer or a scanner.
- RS232 interface specifies the physical layer interface only.
- RS232 is a standard for serial communication, i.e. the bits are transmitted serially.
- The communication between two devices is in full duplex, i.e. the data transfer can take place in both direction.

RS232/UART

- **RS232 Communication Parameters**
  - When two devices have to communicate through RS232, the sending device sends the data character by character.
  - The bits corresponding to the character are called data bits.
  - The data bits are prefixed with a bit called start bit, and suffixed with one or two bits called stop bits.
  - The receiving device decodes the data bits using the start bit and stop bits.
  - This mode of communication is called asynchronous communication because no clock signal is transmitted.
  - In addition to start bit and stop bit an additional bit called parity bit is also sent.
  - Parity bit is used for error detection at the receiving end

RS232/UART

- For two devices to communicate with each other using RS232, the communication parameters have to be set on both the systems.
- And, for a meaningful communication, these parameters have to be the same. The various communication parameters are:
  - **Data rate**: The rate at which data communication takes place. The PC supports various data rates such as 50, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200 bps. The oscillator in the RS232 circuitry operates at 1.8432 MHz and it is divided by 1600 to obtain the 115200 data rate
  - **Data bits**: Number of bits transmitted for each character. The character can have 5 or 6 or 7 or 8 bits. If you send ASCII characters, the number of bits is 7.

RS232/UART

- Start bit:- The bit that is prefixed to the data bits to identify the beginning of the character.
- Stop Bits:- These bits are appended to the data bits to identify the end of character. If the data bits are 7 or 8, one stop bit is appended. If the data bits are 5 or 6, two stop bits are appended.
- Parity:- The bit appended to the character for error checking. The parity can be even or odd. For even parity, the parity bit (1 or 0) will be added in such a way that the total number of bits will be even. For odd parity, the parity bit will make the total number of bits odd. If the parity is set to 'none', the parity bit is ignored. For example, if the data bits are 1010110, the parity bit is 0 if even parity is used ;and the parity bit is 1 if odd parity is used. At the receiving end, the device will calculate the parity bit and if the received parity bit matches with the calculated parity bit, it can be assumed that the data is without errors.

RS232/UART

- **Flow control**: If one of the devices sends data at a very fast rate and the other device cannot absorb the data at that rate, flow control is used. Flow control is a protocol to stop/resume data transmission.

  This protocol is also known as handshaking.

  Hardware handshaking in RS232 is done using two signals: Request to Send (RTS) and Clear to Send (CTS). When a device has data to send, it asserts RTS and the receiving device asserts CTS.

  S/w handshaking--a device sends a request to suspend data transmission by sending the character Control S (0x13). The signal to resume data transmission is sent using the character Control Q(0xll). This software handshaking is also known as XON/XOFF.

RS232/UART

- RS232 Connector Configuration
  - RS232 standard specifies two types of connectors: 25-pin connector and 9-pin connector. In the 25-pin configuration, only a few pins are used.

Pin number	Function (abbreviation)
1	Chassis ground
2	Transmit data (TXD)
3	Receive data (RXD)
4	Request To Send (RTS)
5	Clear To Send (CTS)
6	Data Set Ready (DSR)
7	Signal Ground (GND)
8	Carrier Detect (CD)
20	Data Terminal Ready (DTR)
22	Ring Indicator (RI)

RS232/UART

– For 9-pin connector, the pin details are given below

Pin number	Function (abbreviation)
1	Carrier Detect (CD)
2	Receive Data (RXD)
3	Transmit Data (TXD)
4	Data Terminal Ready (DTR)
5	Signal Ground (GND)
6	Data Set Ready (DSR)
7	Request to Send (RTS)
8	Clear to Send (CTS)
9	Ring Indicator (RI)

RS232/UART

- For transmission of 1's and 0's, the voltage levels are defined in the standard. The voltage levels are different for control signals and data signals. These voltage levels are:

For Data Transmission	Control Signal
-3V to -12 V=1	-3 V to -12V=0
+3V to 12=0	+3V to 12=1

RS232/UART

- Note that the voltage levels used in RS232 are different from voltage levels used in embedded systems (as most chips use 5 volts and below only).
- Another problem is that the processor gives out the data in parallel format, not in serial format. These problems are overcome through the use of UART (Universal Asynchronous Receive Transmit) chips.

RS232/UART

- UART
  - The processors process the data in parallel format, not in serial format.
  - To bridge the processor and the RS232 port, Universal Asynchronous Receive Transmit (UART) chip is used.
  - UART has two sections: receive section and transmit section.
  - Receive section receives the data in serial format, converts it into parallel format and gives it to the processor.
  - The transmit section takes the data in parallel format from the processor and converts it into serial format.
  - The UART chip also adds the start bit, stop bits and parity bit.
  - Many micro-controllers have on-chip UART. However, the necessary voltage level conversion has to be done to meet the voltage levels of RS232.
  - This is achieved using a level shifter.

RS232/UART

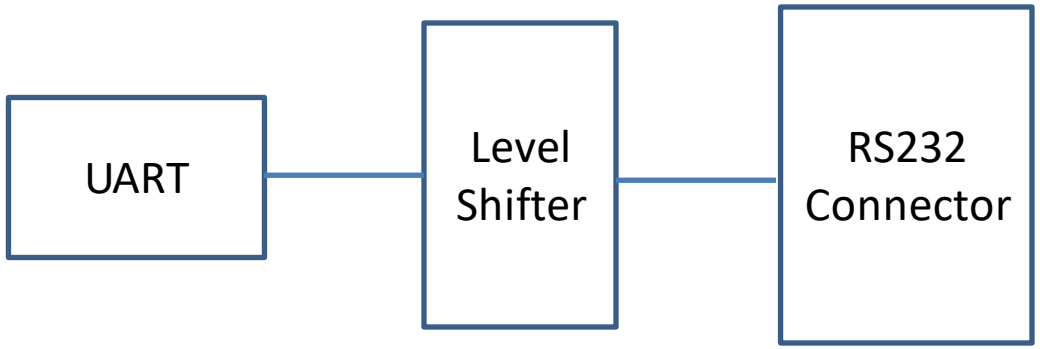

Fig1. Hardware for RS232 Interface

UART chip operates at 5 Volts. The level conversion to the desired voltages is done by the level shifter, and then the signals are passed on to the RS232 connector.

RS232 standard specifies a distance of 19.2 meters. However, you can achieve distances up to 100 meters using RS232 cables. The data rates supported will be dependent on the UART chip and the clock used.

RS232/UART

- **Null Modem Cable Connection**
  - To connect two DTEs such as two PCs, the two RS232 ports using a null modem cable are interconnected.
  - If you want to connect two DTEs such as two PCs, you need to interconnect the two RS232 ports using a null modem cable.
  - The null modem cable connections are shown the figures for 25-pin and 9-pin connectors.
  - However, the minimal connections required are for TD, RD and GND.
  - In 25 pin connectors, it is sufficient if to connect pin 2 to pin 3, pin 3 to pin 2 and pin 7 to pin 7. This is the minimal configuration.
  - To make two devices communicate with each other using RS232 interface, you need to connect the two PCs using a null modem cable and set the communication parameters on both the devices.

RS232/UART

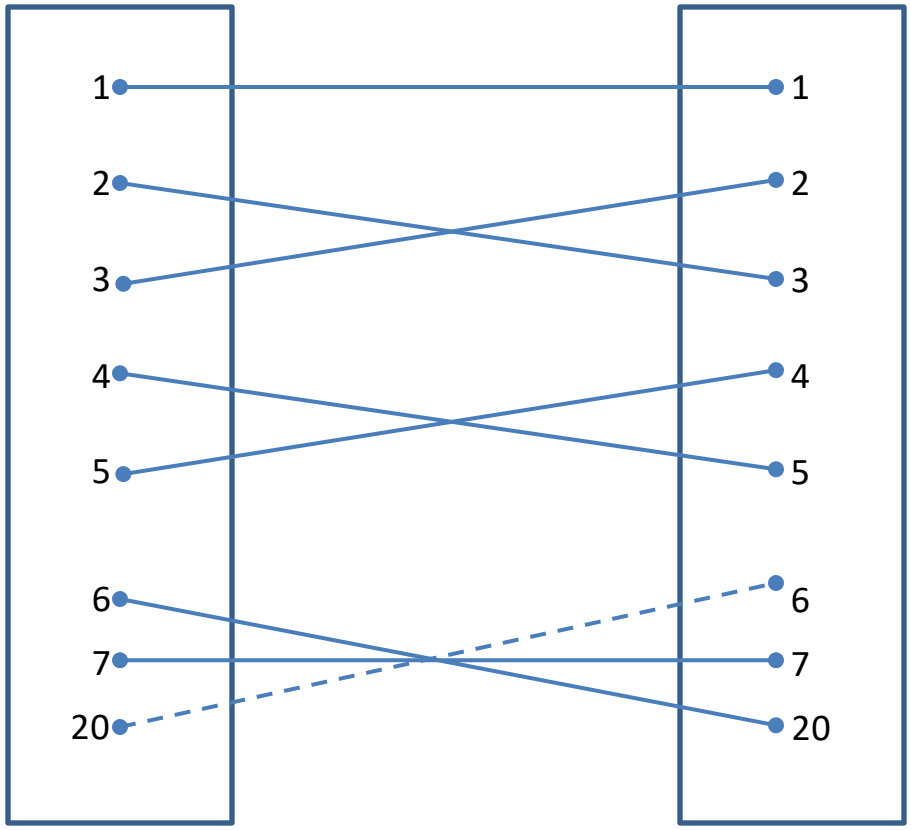

Fig2. Null Modem Cable Connections (25-pin Connectors)

RS232/UART

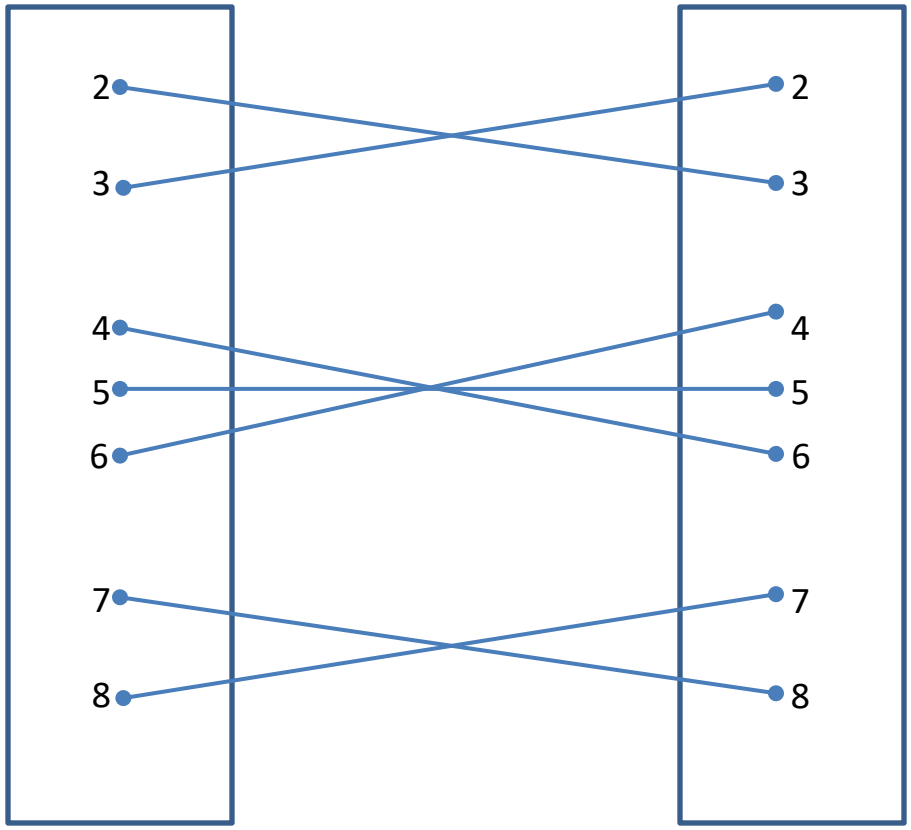

Fig3. Null Modem Cable Connections (9-pin Connectors)

RS422/RS485

- RS422 standard for serial communication is used in noisy environments.
- The distance between two devices can be up to 1200 meters.
- Twisted copper cable is used as the transmission medium.
- In RS422, voltage difference between the two copper wires represents the logic levels. Two channels are used for transmit and receive paths.
- RS422 is better suited to work in noisy environments over longer distances because of balanced transmission.
- RS485 is a variation of RS422 to connect a number of devices in a network.
- An RS485 controller chip is used on each device.
- A network using RS485 protocols operates in a Master/Slave configuration.
- Up to 512 devices can be networked.
- Using one twisted pair, half-duplex communication can be achieved and Using two twisted pairs, full-duplex communication can be achieved.

USB

- Universal Serial Bus has gained immense popularity in recent years.
- Desktops, laptops, printers, display devices, video cameras, hard disk drives, CDROM drives, audio equipment etc. are now available with USB interface.
- Using USB, a number of devices can be networked using Master/Slave architecture.
- A host, such as the PC, is designated as a master.
- On the host, such as a PC, there will be a host controller — a combination of hardware and software — to control all the USB devices.
- Devices can be connected to the host controller either directly or through a hub.
- A hub is also a USB device that extends the number of ports — from 2 to 8 — to connect other USB devices.
- A USB device can be self-powered, or powered by the bus.
- USB can supply 500 mA current to the devices.

USB

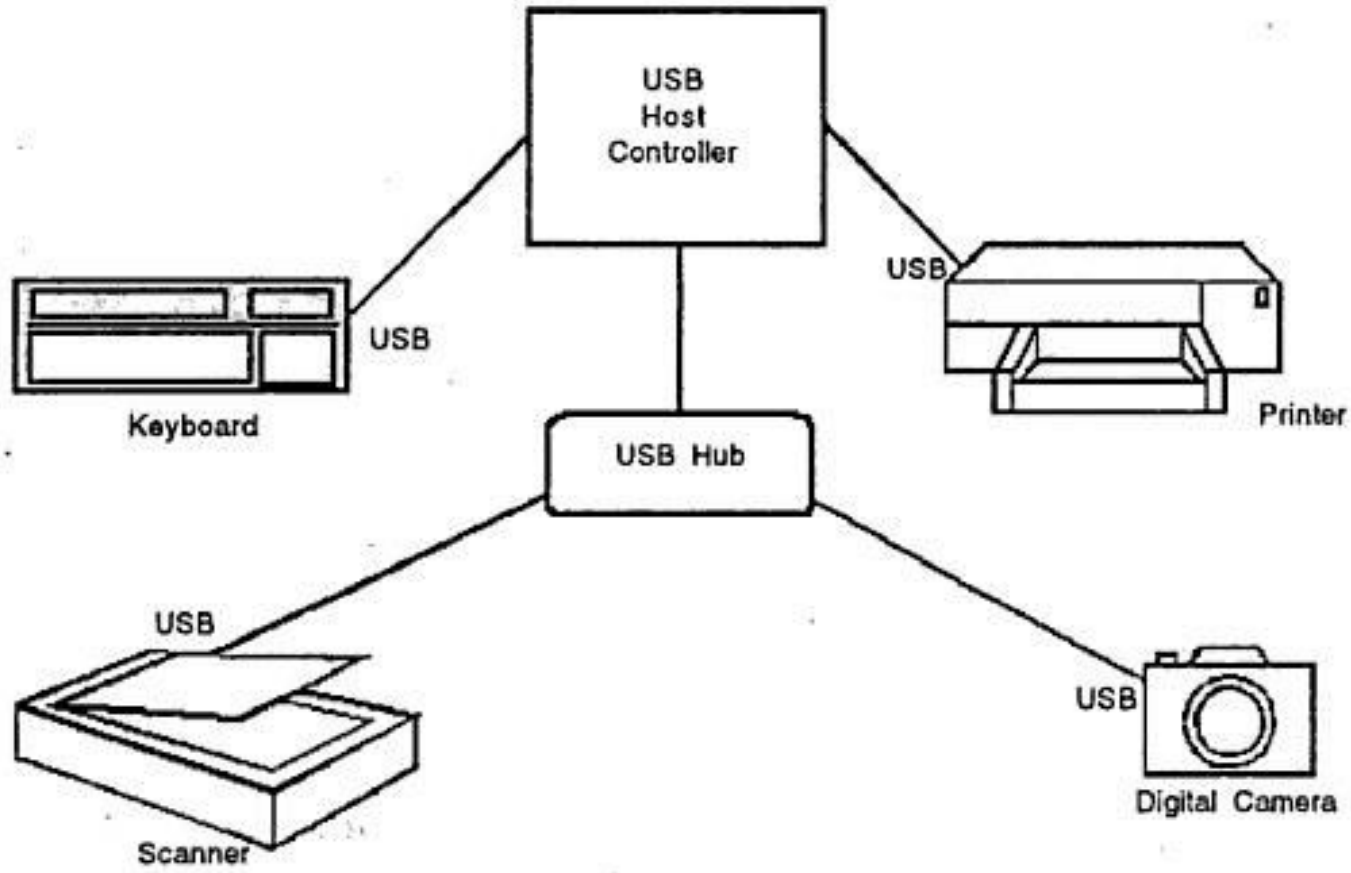

Fig 4. USB device connection hierarchy

USB

- USB Physical Interface
  - A shielded 4-wire twisted copper cable is used with the pin connections.
  - Data is transmitted over a differential twisted pair of wires labeled D+ and D-.

Pin number	Function (abbreviation)
1	+5 V Power (VBUS)
2	Differential data line (D+)
3	Differential data line (D-)
4	Power and Signal ground (GND)

USB

- **Features of USB**
  - **Data rates**: USB 1.1 supports 12 Mbps data rate, and 1.5 Mbps for slower peripherals. USB2.0 supports data rates up to 480 Mbps.
  - **Special features**: USB supports plug and play. The host will detect and identify the device by exchanging a set of packets. This is known as "Bus Enumeration". The devices are hot-pluggable.
  - **Communication protocol:** The communication between the host and the devices is in the form of packets. The host obtains the configuration and properties of the device and assigns a unique ID. When a device is removed, the hub informs the host. Short data packets are exchanged for handshaking, acknowledgements, and for informing the capabilities of the devices. Packets of the size up to 1023 bytes are exchanged for data transfer. When a device is plugged in, the host automatically gets the complete information about the device, either directly or through the hub. An ID is assigned to that device and the communication can start.

USB

- **Device Classes**: Each USB device has a unique ID (between 1 and 127) and a device descriptor provides information about the device class and its properties. The device classes are display, communication, audio, mass storage and human interface (such as keyboards, front panel knobs, control panel in VCR, data gloves etc.). USB is a powerful, versatile and simple communication interface. So, not surprisingly, many peripherals are now provided with a USB interface.