MODULE 3- EMBEDDED SYSEM

What is an Embedded System?

An embedded system is an Electronic/ Electromechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software). Every embedded system is unique, and the hardware as well as firmware is highly specialized to the application domain. Embedded systems are designed for a specific application.

An embedded system must perform the operations at a high speed so that it can be readily used for real time applications and its power consumption must be very low and the size of the system should be as small as possible and the readings must be accurate with minimum error.

E.g., Electronic Toys, Mobile Handsets, Washing Machines, Air Conditioners, Automotive Control Units, Set Top Box, DVD Player etc.

Embedded Systems vs. General Computing Systems

The Embedded System and the General-purpose computer are at two extremes. The embedded system is designed to perform a specific task whereas as per definition the general-purpose computer is meant for general use. It can be used for playing games, watching movies, creating software, work on documents or spreadsheets etc.

Following are certain specific points of difference between embedded systems and general-purpose computers:

Criteria	**General Computer Purpose**	**Embedded system**
Contents	It is combination of generic hardware and a general-purpose OS for executing a variety of	It is combination of special purpose hardware and embedded OS for executing specific set of applications
Operating System	It contains general purpose operating system	It may or may not contain operating system.
Alterations	Applications are alterable by the user.	Applications are non-alterable by the user.
Key factor	Performance is key factor.	Application specific requirements are key factors.
Power Consumption	More	Less
Response Time	Not Critical	Critical for some applications

Classification of Embedded Systems

Some of the criteria used in the classification of embedded systems are as follows:

- Based on Generation
- Based on Complexity & Performance Requirements
- Based on deterministic behavior
- Based on Triggering

Classification based on generation

This classification is based on the order in which the embedded processing system evolved from the first version to where they are today.

i. **First Generation**

The early embedded systems built around 8-bit microprocessors like 8085 and Z80 and 4-bit microcontrollers
E.g.: stepper motor control units, Digital Telephone Keypads etc.

ii. **Second Generation**

Embedded Systems built around 16-bit microprocessors and 8 or 16-bit microcontrollers, following the first-generation embedded systems.
E.g.: SCADA, Data Acquisition Systems etc.

iii. **Third Generation**

Embedded Systems built around high performance 16/32-bit Microprocessors /controllers, Application Specific Instruction set processors like Digital Signal Processors (DSPs), and Application Specific Integrated Circuits (ASICs). The instruction set is complex and powerful.
E.g.: Robotics, industrial process control, networking etc.

iv. **Fourth Generation**

Embedded Systems built around System on Chips (SoC's), Re- configurable processors and multicore processors. It brings high performance, tight integration and miniaturization into the embedded device market
E.g.: Smart phone devices, MIDs etc.

Classification based on Complexity and Performance

i. **Small Scale:** The embedded systems built around low performance and low cost 8- or 16-bit microprocessors/ microcontrollers. It is suitable for simple applications and where performance is not time critical. It may or may not contain OS.

ii. **Medium Scale:** Embedded Systems built around medium performance, low cost 16- or 32-bit microprocessors / microcontrollers or DSPs. These are slightly complex in hardware and firmware. It may contain GPOS/RTOS.

iii. **Large Scale/Complex:** Embedded Systems built around high performance 32- or 64-bit RISC

Major Application Areas of Embedded Systems

We are living in a world where embedded systems play a vital role in our day We are living in a world where embedded systems play a vital role in our day people find their job for a livelihood.

A few of the important domains and products are listed below:

(1) Consumer electronics: Camcorders, cameras, etc.

(2) **Household appliances:** Television, DVD players, washing machine, fridge, microwave oven, etc.

(3) Home automation and security systems: Air conditioners, sprinklers, intruder detection alarms, circuit television cameras, fi re alarms, etc.

(4) Automotive industry: Anti-lock braking systems (ABS), engine control, ignition systems, automatic navigation systems, etc.

(5) Telecom: Cellular telephones, telephone switches, handset multimedia applications, etc.

(6) Computer peripherals: Printers, scanners, fax machines, etc.

(7) Computer networking systems: Network routers, switches, hubs, firewalls, etc.

(8) Healthcare: Different kinds of scanners, EEG, ECG machines etc.

(9) Measurement & Instrumentation: Digital multimeters, digital CROs, logic analyzers PLC systems,

(10) Banking & Retail: Automatic teller machines (ATM) and currency counters, point of sales (POS)

(11) Card Readers: Barcode, smart card readers, hand held devices, etc.

(12) Wearable Devices: Health and Fitness Trackers, Smartphone Screen extension for notifications, etc.

(13) Cloud Computing and Internet of Things (IOT)

Elements of an Embedded System

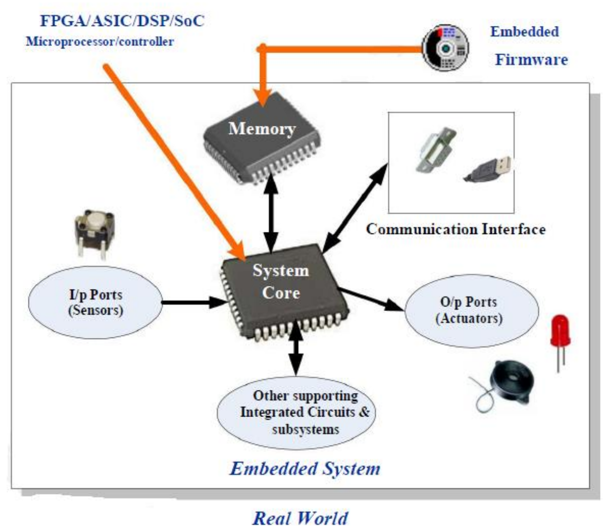

Figure 3.6.1: Elements of an embedded system

- A typical embedded system contains a single chip controller which acts as the master brain of the system. Diagrammatically an embedded system can be represented as shown in figure 3.6.1.
- Embedded systems are basically designed to regulate a physical variable (such as Microwave Oven) or to manipulate the state of some devices by sending some signals to the actuators or devices connected to the output port system (such as temperature in Air Conditioner)
- In response to the input signal provided by the end users or sensors which are connected to the input ports. Hence the embedded systems can be viewed as a reactive system.
- Keyboards, push button, switches, etc. are examples of common user interface input devices
- LEDs, LCDs, Piezoelectric buzzers, etc. are examples for common user interface output devices for a typical embedded system.
- Some embedded systems do not require any manual intervention for their operation. They automatically sense the input parameters from real world through sensors which are connected at input port.
- The sensor information is passed to the processor after signal conditioning and **digitization**.
- The core of the system performs some predefined operations on input data with the help of embedded firmware in the system and sends some actuating signals to the actuator connect connected to the output port of the system.
- The memory of the system is responsible for holding the code (control algorithm and other important configuration details).
- There are two types of memories are used in any embedded system.
- Fixed memory (ROM) is used for storing code or program. The user cannot change the firmware in this type of memory.
- The most common types of memories used in embedded systems for control algorithm storage are OTP, PROM, UVEPROM, EEPROM and FLASH.

Core of the Embedded Systems

Embedded systems are domain and application specific and are built around a central core. The core of the embedded system falls into any one of the following categories:

- General Purpose and Domain Specific Processors
  - Microprocessors
  - Microcontrollers
  - Digital Signal Processors
- Programmable Logic Devices (PLDs)
- Application Specific Integrated Circuits (ASICs)

- Commercial off the shelf Components (COTS)

General Purpose and Domain Specific Processors

Almost 80% of the embedded systems are processor/ controller based. The processor may be microprocessor or a microcontroller or digital signal processor, depending on the domain and application.

Microprocessor:

- It is a silicon chip representing a Central Processing Unit (CPU), which can perform **arithmetic as well as logical operations**
- In general, the CPU contains the Arithmetic and Logic Unit (ALU), Control Unit and Working registers.
- Microprocessor is a dependent unit and it requires the combination of other hardware like Memory, Timer Unit, and Interrupt Controller etc. for proper functioning.
- Intel claims the credit for developing the first Microprocessor unit Intel 4004, a 4 bit processor which was released in Nov 1971.

Microcontroller:

- It is a highly integrated silicon chip containing a CPU, scratch pad RAM, Special and General-purpose Register Arrays, On Chip ROM/FLASH memory for program storage, Timer and Interrupt control units and dedicated I/O ports.
- Microcontrollers can be considered as a super set of Microprocessors.
- Microcontroller can be general purpose (like Intel 8051, designed for generic applications and domains) or application specific (Like Automotive AVR from Atmel Corporation. Designed specifically for automotive applications).
- Since a microcontroller contains all the necessary functional blocks for independent working, they found greater place in the embedded domain in place of microprocessors.
- Microcontrollers are cheap, cost effective and are readily available in the market.
- Texas Instruments TMS 1000 is considered as the world's first microcontroller.

Microprocessor vs Microcontroller

The following table summarizes the differences between a microcontroller and microprocessor.

MICROPROCESSOR	MICROCONTROLLER
a silicon chip representing a central processing unit (CPU), which is capable of performing arithmetic as well as logical operations according to a predefined set of instructions it is a dependent unit. it requires the combination of other chips like timers, program and data memory chips, interrupt controllers, etc. for functioning	A microcontroller is a highly integrated chip that contains a CPU, scratchpad RAM, special and general-purpose register arrays, on chip ROM/FLASH memory for program storage, timer and interrupt control units and dedicated I/O ports It is a self-contained unit and it doesn't require external interrupt controller, timer, UART, etc. for its functioning
most of the time general purpose in design and operation	Mostly application-oriented or domain-specific
doesn't contain a built in I/O port. the i/o port functionality needs to be implemented with the help of external programmable peripheral interface chips like **8255**	Most of the processors contain multiple built-in I/O ports. Which can be operated as a single 8 or 16- or 32-bit port as individual port pins.

targeted for high end market where performance is important	Targeted for embedded market where performance is not so critical (At present this demarcation is invalid)
limited power saving options compared to microcontrollers	includes lot of power saving features

RISC vs CISC Processors/Controllers

The term RISC stands for Reduced Instruction Set Computing and CISC stands for Complex Instruction Computing. Some of the important criteria are listed below:

RISC	CISC
Lesser number of instructions	Greater number of Instructions
Instruction pipelining and increased execution speed	Generally, **no instruction pipelining feature**
Orthogonal instruction set	**Non-orthogonal** instruction set
Operations are performed on registers only; the only memory operations are load and store.	Operations are performed **on registers or memory** depending on the instruction.
Many registers are available.	**Limited** number of general-purpose registers.
Programmer needs to write more code to execute a task since the instructions are simpler ones.	Instructions are like macros in **C language**. A programmer can achieve the desired functionality with a single instruction which in turn provides the effect of using **more simpler single instructions** in RISC.
Single, fixed length instructions	**Variable length instructions**
Less silicon usage and pin count	**More** silicon usage since more additional decoder logic is required to implement the complex instruction decoding.
With Harvard Architecture	**Harvard or Von-Neumann Architecture**
Lesser number of instructions	Greater number of Instructions

Harvard V/s Von-Neumann Processor/Controller Architecture

The terms Harvard and Von-Neumann refers to the processor architecture design. Figure 3.7.1 explains the Harvard and Von-Neumann architecture concept.

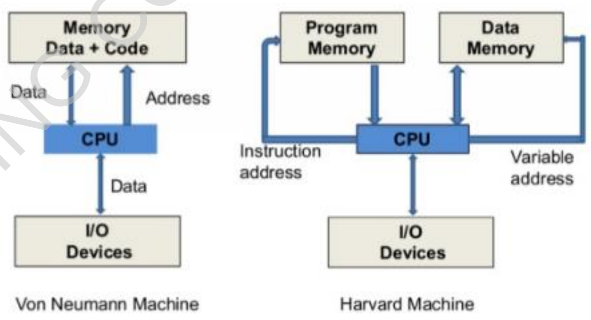

Figure 3.7.1 Harvard vs Von-Neumann architecture

The following table highlights the difference between Harvard and Von-Neumann architecture.

Harvard	Von-Neumann
Separate buses for instruction and data fetching	Single shard bus for instruction and data fetching
Easier to pipeline, so high performance can be achieved	Comparatively, low performance
Comparatively high cost	Cheaper
No chances for accidental corruption of program memory	Memory in same chip, chances of accidental corruption of program memory.

Chapter- Sensors and Interfacing

Instrumentation and controlsystems

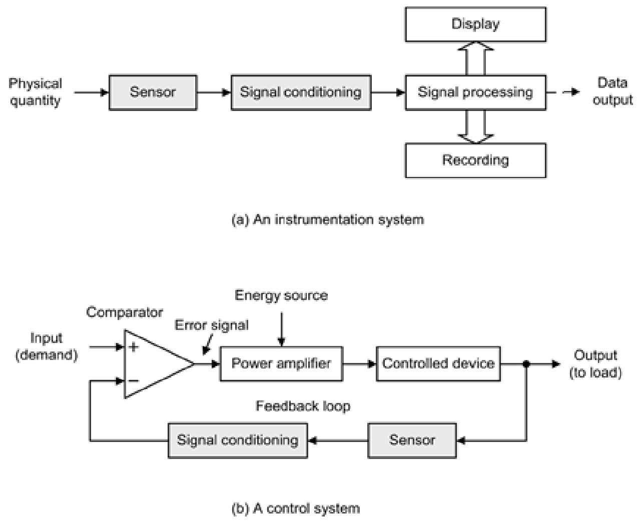

Figure 15.1 Instrumentation and control systems

- Fig. 15.1 shows the arrangement of an instrumentation system. The physical quantity to be measured (e.g., temperature) acts upon a sensor that produces an electrical output signal.
- This signal is an electrical analogue of the physical input but note that there may not be a linear relationship between the physical quantity and its electrical equivalent.
- Because of this and since the output produced by the sensor may be small or may suffer from the presenceof noise (i.e., unwanted signals)
- Further signal conditioning will be required before the signal will be at an acceptable level and in an acceptable form for signal processing
- Display and recording. Furthermore, because the signal processing may use digital rather than analogue signalsan additional stage of analogue-to-analogueconversion may be required.
- Fig. 15.1(b) shows the arrangement of a control system. This uses **negative feedback** in order to regulate and stabilize the output.
- It thus becomes possible to set the input or **demand** (i.e., what we desire the output to be) and leave the system to regulate itself by comparing it with a signal derived from the output (via a sensor andappropriate signal conditioning).
- A **comparator** is used to sense the difference in these two signals and where any discrepancy is detected the input to the power amplifier is adjusted accordingly.
- This signal is referred to as an **error signal** (it should be zero when the outputexactly matches the demand).
- The input (demand) is often derived from a simple potentiometer connected across a stable d.c. voltage source while the controlled device can take many forms (e.g., a d.c. motor, linear actuator, heater, etc.).

Transducers

- Transducers are devices that convert energy in the form of sound, light, heat, etc., into an equivalent electrical signal, or vice versa.
- Before we go further, let's consider a couple of examples that you will already be familiar with. A loudspeaker is a transducer that converts low-frequency electric current into audible sounds.
- A microphone, on the other hand, is a transducer that performs the reverse function, i.e., that of converting sound pressure variations into voltageor current.
- Loudspeakers and microphones can thus be considered as complementary transducers.
- Transducers may be used both as inputs to electronic circuits and outputs from them.
- From the two previous examples, it should be obvious that a loudspeaker is an **output transducer** designed for use in conjunction with an audio system. A microphone is an **input transducer** designed for use with a recording or sound reinforcing system.

There are many different types of transducer and Tables 15.1 and 15.2 provide some examples of transducers that can be used to input and output three important physical quantities; sound, temperature and angular position.

Table 15.1 Some examples of input transducers

Physical quantity	Input transducer	Notes
Sound (pressure change)	Dynamic microphone (see Fig. 15.3)	Diaphragm attached to a coil is suspended in a magnetic field. Movement of the diaphragm causes current to be induced in the coil.
Temperature	Thermo couple (see Fig. 15.2)	Small e.m.f. generated at the junction between two dissimilar metals (e.g. copper and constantan). Requires reference junction and compensated cables for accurate measurement.
Angular position	Rotary potentiometer	Fine wire resistive element is wound around a circular former. Slider attached to the control shaft makes contact with the resistive element. A stable d.c. voltage source is connected across the ends of the potentiometer. Voltage appearing at the slider will then be proportional to angular position.

Table 15.2 Some examples of output transducers

Physical quantity	Output transducer	Notes
Sound (pressure change)	Loudspeaker (see Fig. 15.3)	Diaphragm attached to a coil is suspended in a magnetic field. Current in the coil causes movement of the diaphragm which alternately compresses and rarefies the air mass in front of it.
Temperature	Heating element (resistor)	Metallic conductor is wound onto a ceramic or mica former. Current flowing in the conductor produces heat.
Angular position	Rotary potentiometer	Multi-phase motor provides precise rotation in discrete steps of 15° (24 steps per revolution), 7.5° (48 steps perrevolution) and 1.8° (200 steps per revolution).

Sensors

- A *sensor* is a special kind of transducer that is used to generate an input signal to a measurement, instrumentation or control system.
- The signal produced by a sensor is an **electrical analogy** of a physical quantity, such as distance, velocity, acceleration, temperature, pressure, lightlevel, etc.
- The signals returned from a sensor, together with control inputs from the user or controller (as appropriate) will subsequently be used to determine the output from the system.
- The choice of sensor is governed by a number of factors including accuracy, resolution, cost and physical size.

- Sensors can be categorized as either **active** or **passive**.
- An active sensor *generates* a current or voltage output.
- A passive transducer *requires a source of current or voltage* and it modifies this in some way. The result may still be a voltage or current *but it is not generated by the sensor on its own.*

- Sensors can also be classed as either **digital** or **analogue**.

- The output of a digital sensor can exist in only two discrete states, either 'on' or 'off', 'low' or 'high', 'logic 1' or 'logic 0', etc.

- The outputof an analogue sensor can take any one of an infinite number of voltage or current levels. It is thus said to be *continuously variable*.
- Table 15.3 provides details of some common types of sensors.

Table 15.3 Some examples of input transducers

Physical quantity	Input transducer	Notes
Angular position	Resistive rotary position sensor (see Fig. 15.5) Optical shaft encoder	Rotary track potentiometer with linear law produces analogue voltageproportional to angular position. Encoded disk interposed between optical transmitter and receiver (infra-red LED and photodiode or photo-transistor).
Angular velocity	Tach generator Toothed rotor tachometer	Small d.c. generator with linear output characteristic. Analogue outputvoltage proportional to shaft speed. Magnetic pick-up responds to the movement of a toothed ferrous disk. The pulse repetition frequency of the output is proportional to the angularvelocity.
Flow	Rotating vane flow sensor (see Fig. 15.9)	Turbine rotor driven by fluid. Turbine interrupts infra-red beam. Pulse repetition frequency of output is proportional to flow rate.
Linear position	Resistive linear position sensor Linear variable differential transformer(LVDT) Magnetic linear position sensor	Linear track potentiometer with linear law produces analogue voltageproportional to linear position. Limited linear range. Miniature transformer with split secondary windings and moving coreattached to a plunger. Requires a.c. excitation and phase-sensitive detector. Magnetic pick-up responds to movement of a toothed ferrous track.Pulses are counted as the sensor moves along the track.
Light level	Photocell Light-dependent resistor (LDR) (see Fig. 15.8) Photodiode (see Fig. 15.8) Phototransistor (see Fig. 15.8)	Voltage-generating device. The analogue output voltage produced isproportional to light level. An analogue output voltage results from a change of resistance within a cadmium sulphide (CdS) sensing element. Usually connected as part of apotential divider or bridge. Two-terminal device connected as a current source. An analogue outputvoltage is developed across a series resistor of appropriate value. Three-terminal device connected as a current source. An analogue outputvoltage is developed across a series resistor of appropriate value.
Liquid level	Float switch (see Fig. 15.7) Capacitive proximityswitch Diffuse scan proximityswitch	Simple switch element which operates when a particular level isdetected. Switching device which operates when a particular level is detected. Ineffective with some liquids. Switching device which operates when a particular level is detected. Ineffective with some liquids.
Pressure	Microswitch pressuresensor (see Fig. 15.4) Differential pressure vacuum switch Piezo-resistive pressure sensor	Microswitch fitted with actuator mechanism and range-setting springs. Suitable for high-pressure applications. Microswitch with actuator driven by a diaphragm. May be used to sensedifferential pressure. Alternatively, one chamber may be evacuated and the sensed pressure applied to a second input. Pressure exerted on diaphragm causes changes of resistance in attached piezo-resistive transducers. Transducers are usually arranged in the form of a four active element bridge which produces an analogue output voltage.
Proximity	Reed switch (see Fig. 15.4) Inductive proximityswitch Capacitive proximityswitch Optical proximity switch (see Fig. 15.4)	Reed switch and permanent magnet actuator. Only effective over shortdistances. Target object modifies magnetic field generated by the sensor. Only suitable for metals (non-ferrous metals with reduced sensitivity). Target object modifies electric field generated by the sensor. Suitable for metals, plastics, wood and some liquids and powders. Available in diffuse and through scan types. Diffuse scan types require reflective targets. Both types employ optical transmitters and receivers (usually infra-red emitting LEDs and photo-diodes or photo-transistors). Digital input port required.
Strain	Resistive strain gauge Semiconductor strain gauge	Foil type resistive element with polyester backing for attachment to body under stress. Normally connected in full bridge configuration with temperature-compensating gauges to provide an analogue outputvoltage. Piezo-resistive elements provide greater outputs than comparableresistive foil types. More prone to temperature changes and also inherently non-linear.

Temperature	Thermocouple (seeFig. 15.2)	Small e.m.f. generated by a junction between two dissimilar metals. For accurate measurement, requires compensated connecting cables and specialized interface.
	Thermistor (see Fig. 15.6)	Usually connected as part of a potential divider or bridge. An analogueoutput voltage results from resistance changes within the sensing element.
	Semiconductor temperature sensor (see Fig. 15.6)	Two-terminal device connected as a current source. An analogue output voltage is developed across a series resistor of appropriate value.
Weight	Load cell	Usually comprises four strain gauges attached to a metal frame. This assembly is then loaded and the analogue output voltage produced is proportional to the weight of the load.
Vibration	Electromagnetic vibration sensor	Permanent magnet seismic mass suspended by springs within a cylindrical coil. The frequency and amplitude of the analogue output voltage are respectively proportional to the frequency and amplitude ofvibration.

Figure 15.2 A selection of thermocouple probes

Figure 15.3 A selection of audible transducer

Figure 15.4 Various switch sensors

Figure 15.5 Resistive linear position sensor

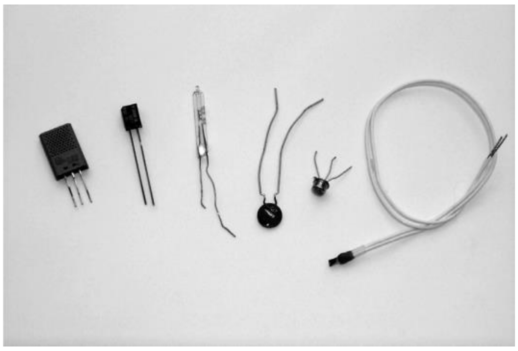

Figure 15.6 Various temperature and gas sensors

Figure 15.7 Liquid level float switch

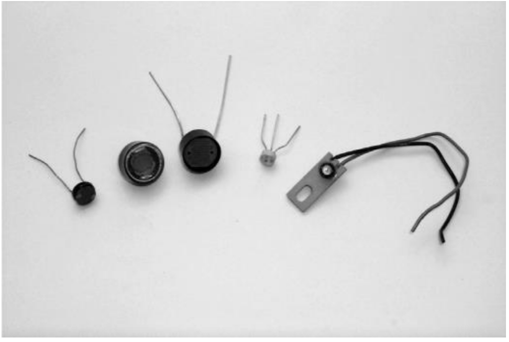

Figure 15.8 Various optical and light sensors

Figure 15.9 Liquid flow sensor (digital output)

Figure 15.10 Contactless joystick

Chapter-Actuators

- Actuator is a form of transducer device (mechanical or electrical) which converts signals to corresponding physical action (motion). Actuator acts as an output device.
- We can see that certain smartwatches use Ambient Light Sensor to detect the surrounding light intensity and uses an electrical/ electronic actuator circuit to adjust the screen brightness for better readability.

Light Emitting Diode (LED)

- Light Emitting Diode (LED) is an important output device for visual indication in any embedded system.
- LED can be used as an indicator for the status of various signals or situations.
- Typical examples are indicating the presence of power conditions like ‘Device ON’, ‘Battery low’ or ‘Charging of battery’ for a battery-operated handheld embedded device.
- Light Emitting Diode is a *p-n* junction diode and it contains an anode and a cathode.
- For proper functioning of the LED, the anode of it should be connected to +ve terminal of the supply voltage and cathode to the –ve terminal of supply voltage.
- The current flowing through the LED must be limited to a value below the maximum current that it can conduct.
- A resister is used in series between the power supply and the LED to limit the current through the LED. The ideal LED interfacing circuit is shown in Fig. 2.13.

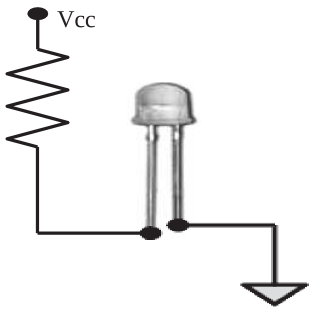

Fig: 2.13 L E D interfacing

- LEDs can be interfaced to the port pin of a processor/controller in two ways.
- In the first method, the anode is directly connected to the port pin and the port pin drives the LED. In this approach the port pin ‘sources’ current to the LED when the port pin is at logic High (Logic ‘1’).
- In the second method, the cathode of the LED is connected to the port pin of the processor/controller and the anode to the supply voltage through a current limiting

resistor. The LED is turned on when the port pin is at logic Low (Logic ‘0’).

7-Segment LED Display

- The 7-segment LED display is an output device for displaying alpha numeric characters. It contains 8 light-emitting diode (LED) segments arranged in a special form.
- Out of the 8 LED segments, 7 are used for displaying alpha numeric characters and 1 is used for representing ‘decimal point’ in decimal number display.

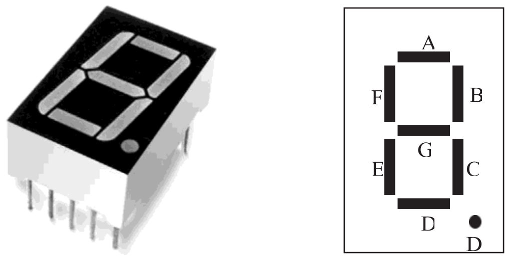

Fig 2.14: -Segment LED Display

- Figure 2.14 explains the arrangement of LED segments in a 7-segment LED display The LED segments are named A to G and the decimal point LED segment is named as DP.
- The LED segments A to G and DP should be lit accordingly to display numbers and characters.
- For example, for displaying the number 4, the segments F, G, B and C are lit. For displaying 3, the segments A, B, C, D, G and DP are lit.
- For displaying the character ‘d’, the segments B, C, D, E and G are lit.
- All these 8 LED segments need to be connected to one port of the processor/controller for displaying alpha numeric digits.
- The 7-segment LED displays are available in two different configurations, namely; **Common Anode and Common Cathode**.
- In the common anode configuration, the anodes of the 8 segments are connected commonly whereas in the common cathode configuration, the 8 LED segments share a common cathode line.
- Figure 2.15 illustrates the Common Anode and Cathode configurations.

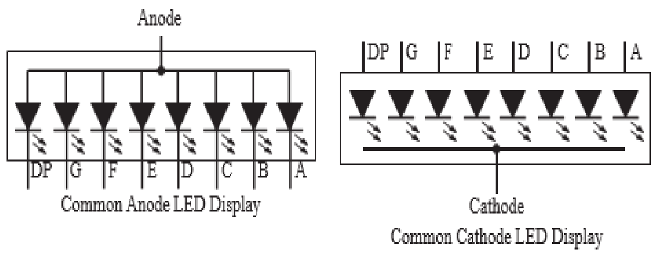

Fig. 2.15 Common anode and cathode configurations of a 7-segment LED Display

- Based on the configuration of the 7-segment LED unit, the LED segment’s anode or cathode is connected to the port of the processor/controller in the order ‘A’ segment to the least significant port pin and DP segment to the most significant port pin.
- The current flow through each of the LED segments should be limited to the maximum value supported by the LED display unit.
- The typical value for the current falls within the range of 20mA. The current through each segment can be limited by connecting a current limiting resistor to the anode or cathode of each segment.
- The value for the current limiting resistors can be calculated using the current value from the electrical parameter listing of the LED display.
- For common cathode configurations, the anode of each LED segment is connected to the port pins of the port to which the display is interfaced.
- The anode of the common anode LED display is connected to the 5V supply voltage through a current limiting resistor and the cathode of each LED segment is connected to the respective port pin lines.
- For an LED segment to lit in the Common anode LED configuration, the port pin to which

the cathode of the LED segment is connected should be set at logic 0.

Stepper Motor

- A stepper motor is an electro-mechanical device which generates discrete displacement (motion) in response to dc electrical signals.
- It differs from the normal dc motor in its operation. The dc motor produces continuous rotation on applying dc voltage whereas a stepper motor produces discrete rotation in response to the dc voltage applied to it.
- Stepper motors are widely used in industrial embedded applications, consumer electronic products and robotics control systems.
- The paper feed mechanism of a printer/fax makes use of stepper motors for its functioning.

Based on the coil winding arrangements, a two-phase stepper motor is classified into two. They are:

(1) Unipolar

(2) Bipolar

- ***Unipolar*** A unipolar stepper motor contains two windings per phase.
- The direction of rotation (clockwise or anticlockwise) of a stepper motor is controlled by changing the direction of current flow.
- Current in one direction flows through one coil and in the opposite direction flows through the other coil.
- It is easy to shift the direction of rotation by just switching the terminals to which the coils are connected.
- Figure 2.18 illustrates the working of a two-phase unipolarstepper motor

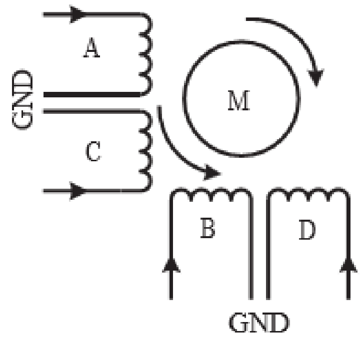

Fig. 2.18 2-Phase unipolar stepper motor

- The coils are represented as A, B, C and D.
- Coils A and C carry current in opposite directions for phase 1 (only one of them will be carrying current at a time).
- Similarly, B and D carry current in opposite directions for phase 2 (only one of them will be carrying current at a time).

- ***Bipolar*** A bipolar stepper motor contains single winding per phase. For reversing the motor rotation, the current flow through the windings is reversed dynamically.
- It requires complex circuitry for current flow reversal. The stator winding details for a two-phase unipolar stepper motor is shown in Fig. 2.19.

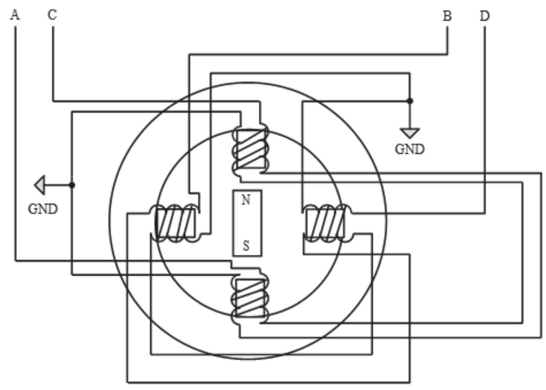

Fig. 2.19 Stator Winding details for a 2 Phase unipolar stepper motor

- The stepping of stepper motor can be implemented in different ways by changing the sequence of activation of the stator windings.
- The different stepping modes supported by stepper motor are explained below.

- **Full Step** In the full step mode both the phases are energized simultaneously. The coils A, B, C and D are energized in the following order:

Step	Coil A	Coil B	Coil C	Coil D
1	H	H	L	L
2	L	H	H	L
3	L	L	H	H
4	H	L	L	H

- **Wave Step** In the wave step mode only one phase is energized at a time and each coil of the phase is energized alternatively.
- The coils A, B, C, and D are energized in the following order:

Step	Coil A	Coil B	Coil C	Coil D
1	H	L	L	L
2	L	H	L	L
3	L	L	H	L
4	L	L	L	H

- **Half Step** It uses the combination of wave and full step. It has the highest torque and stability. The coil energizing sequence for half step is given below

Step	Coil A	Coil B	Coil C	Coil D
1	H	L	L	L
2	H	H	L	L
3	L	H	L	L
4	L	H	H	L
5	L	L	H	L
6	L	L	H	H
7	L	L	L	H
8	H	L	L	H

- The rotation of the stepper motor can be reversed by reversing the order in which the coil is energized.
- The current requirement for stepper motor is little high and hence the port pins of a microcontroller/processor may not be able to drive them directly.
- Also, the supply voltage required to operate stepper motor varies normally in the range 5V to 24 V.
- Depending on the current and voltage requirements, special driving circuits are required to interface the stepper motor with microcontroller/processors.
- Commercial off-the-shelf stepper motor driver ICs are available in the market and they can be directly interfaced to the microcontroller port.
- ULN2803 is an octal peripheral driver array available from Texas Instruments and ST microelectronics for driving a 5V stepper motor.
- Simple driving circuit can also be built using transistors
- The following circuit diagram (Fig. 2.20) illustrates the interfacing of a stepper motor through a driver circuit connected to the port pins of a microcontroller/processor.

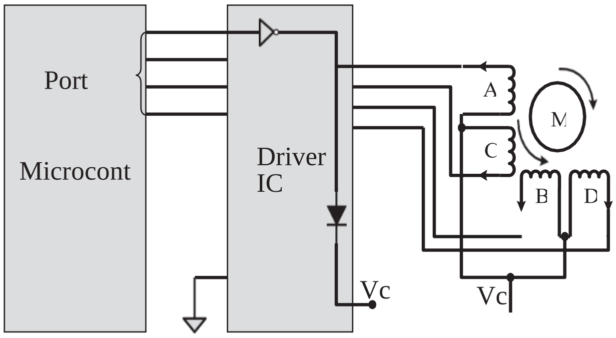

Fig. 2.20 Interfacing of stepper motor through driver circuit

Relay

- Relay is an electro-mechanical device. In embedded application
- The 'Relay' unit acts as dynamic path selectors for signals and power. The 'Relay' unit contains a relay coil made up of insulated wire on a metal core and a metal armature with one or more contacts.
- 'Relay' works on electromagnetic principle. When a voltage is applied to the relay coil, current flows through the coil, which in turn generates a magnetic field.
- The magnetic field attracts the armature core and moves the contact point.
- The movement of the contact point changes the power/signal flow path. 'Relays' are available in different configurations.
- Figure 2.21 given below illustrates the widely used relay configurations for embedded applications

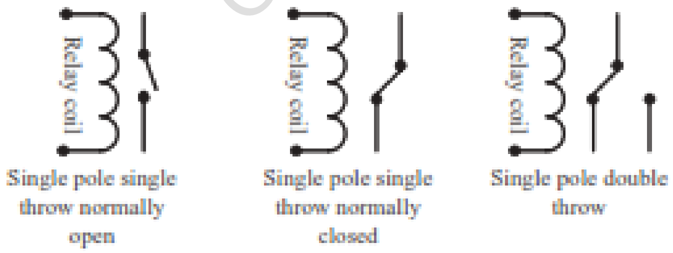

Fig. 2.21 Relay configurations

- The Single Pole Single Throw configuration has only one path for information flow. The path is either open or closed in normal condition.
- For normally Open Single Pole Single Throw relay, the circuit is normally open and it becomes closed when the relay is energized.
- For normally closed Single Pole Single Throw configuration, the circuit is normally closed and it becomes open when the relay is energized.
- For Single Pole Double Throw Relay, there are two paths for information flow and they are selected by energizing or de-energizing the relay.
- The Relay is normally controlled using a relay driver circuit connected to the port pin of the processor/ controller. A transistor is used for building the relay driver circuit. Figure 2.22 illustrates the same.

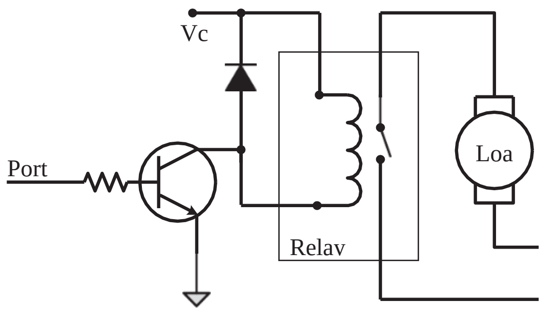

Fig. 2.22 Transistor based Relay driving circuit

- A free-wheeling diode is used for free-wheeling the voltage produced in the opposite direction when the relay coil is de-energized.
- The freewheeling diode is essential for protecting the relay and the transistor.

Piezo Buzzer

- Piezo buzzer is a piezoelectric device for generating audio indications in embedded application.
- A piezoelectric buzzer contains a piezoelectric diaphragm which produces audible sound in response to the voltage applied to it.
- Piezoelectric buzzers are available in two types. **'Self-driving' and 'External driving'**.

- The 'Self-driving' circuit contains all the necessary components to generate sound at a predefined tone. It will generate a tone on applying the voltage.
- External driving piezo buzzers supports the generation of different tones. The tone can be varied by applying a variable pulse train to the piezoelectric buzzer.
- A piezo buzzer can be directly interfaced to the port pin of the processor/control.
- Depending on the driving current requirements, the piezo buzzer can also be interfaced using a transistor-based driver circuit as in the case of a 'Relay'.

Push Button Switch

- It is an input device. Push button switch comes in two configurations, namely **'Push to Make' and 'Push to Break'**.
- In the 'Push to Make' configuration, the switch is normally in the open state and it makes a circuit contact when it is pushed or pressed.
- In the 'Push to Break' configuration, the switch is normally in the closed state and it breaks the circuit contact when it is pushed or pressed.

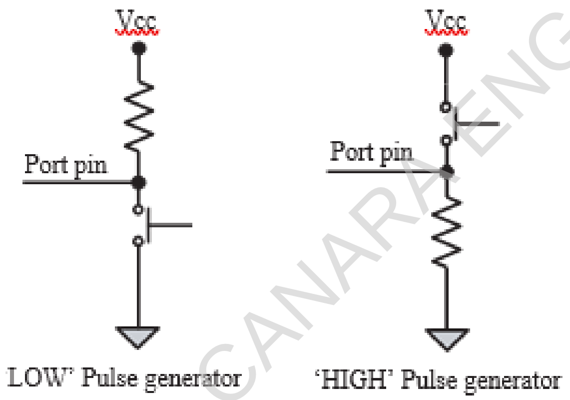

Fig. 2.23 Push button switch configurations

- Push button is used for generating a momentary pulse. In embedded application push button is generally used as reset and start switch and pulse generator.
- The Push button is normally connected to the port pin of the host processor/ controller. Depending on the way in which the push button interfaced to the controller, it can generate either a 'HIGH' pulse or a 'LOW' pulse. Figure 2.23 illustrates how the push button can be used for generating 'LOW' and 'HIGH' pulses.

Keyboard

- Keyboard is an input device for user interfacing. If the number of keys required is very limited, push-button switches can be used and they can be directly interfaced to the port pins for reading.
- However, there may be situations demanding a large number of keys for user input (e.g., PDA device with alpha-numeric keypad for user data entry).
- In such situations it may not be possible to interface each key to a port pin due to the limitation in the number of general-purpose port pins available for the processor/ controller in use and moreover it is wastage of port pins.
- Matrix keyboard is an optimum solution for handling large key requirements. It greatly reduces the number of interface connections.
- For example, for interfacing 16 keys, in the direct interfacing technique 16 port pins are required, whereas in the matrix keyboard only 8 lines are required. The 16 keys are arranged in a 4 column × 4 Row matrix.
- Figure 2.24 illustrates the connection of keys in a matrix keyboard

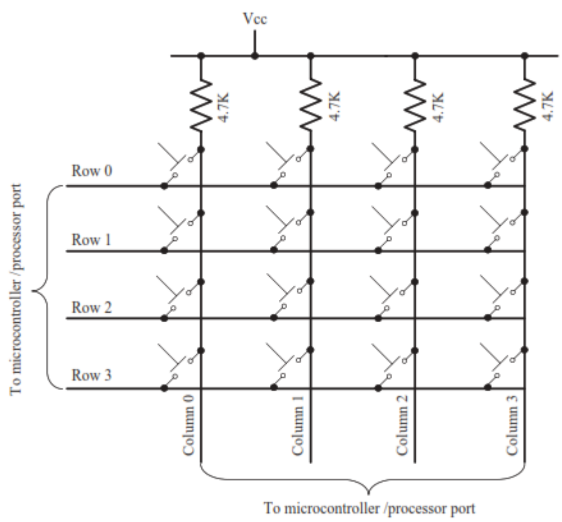

Fig. 2.24 Matrix keyboard Interfacing

In a matrix keyboard, the keys are arranged in matrix fashion (i.e., they are connected in a row and columnstyle). For detecting a key press,

- The keyboard uses the scanning technique, where each row of the matrix is pulled low and the columns are read.
- After reading the status of each column corresponding to a row,the row is pulled high and the next row is pulled low and the status of the columns are read.
- This process is repeated until the scanning for all rows are completed. When a row is pulled low and if a key connected to therow is pressed, reading the column to which the key is connected will give logic 0.
- Since keys are mechanical devices, there is a possibility for **de-bounce** issues, which may give multiple key press effect for a singlekey press.
- To prevent this, a proper key **de-bouncing technique** should be applied. **Hardware key de-bouncer** circuits and **software key de-bounce techniques** are the key de-bouncing techniques available.
- The softwarekey de-bouncing technique doesn't require any additional hardware and is easy to implement.
- In the software de-bouncing technique, on detecting a key-press, the key is read again after a de-bounce delay.\
- If the key press is a genuine one, the state of the key will remain as 'pressed' on the second read also. Pull-up resistorsare connected to the column lines to limit the current that flows to the Row line on a key press.

Communication Interface

Communication interface is essential for communicating with various subsystems of the embedded system and with the external world. The communication interface can be viewed in two different perspectives; namely;

- Device/board level communication interface (Onboard Communication Interface)
- Product level communication interface (External Communication Interface)

Universal Asynchronous Receiver Transmitter (UART)

- Universal Asynchronous Receiver Transmitter (UART) based data transmission is an asynchronous form of serial data transmission.
- UART based serial data transmission doesn't require a clock signal to synchronize the transmitting end and receiving end for transmission.
- Instead, it relies upon the pre-defined agreement between the transmitting device and receiving device.
- The serial communication settings (Baud rate, number of bits per byte, parity, number of start bits and stop bit and flow control) for both transmitter and receiver should be set as identical.
- The start and stop of communication is indicated through inserting special bits in the data stream.
- While sending a byte of data, a start bit is added first and a stop bit is added at the end of the bit stream.
- The least significant bit of the data byte follows the 'start' bit.
- The 'start' bit informs the receiver that a data byte is about to arrive. The receiver device starts polling its 'receive line' as per the baud rate settings.
- If the baud rate is 'x' bits per second, the time slot available for one bit is 1/x seconds. The receiver unit polls the receiver line at exactly half of the timeslot available for the bit.
- If parity is enabled for communication, the UART of the transmitting device adds a parity bit (bit value is 1 for odd number of 1s in the transmitted bit stream and 0 for even number of 1s).
- The UART of the receiving device calculates the parity of the bits received and compares it with the received parity bit for error checking.
- The UART of the receiving device discards the 'Start', 'Stop' and 'Parity' bit from the received bit stream and converts the received serial bit data to a word (In the case of 8 bits/byte, the byte is formed with the received 8 bits with the first received bit as the LSB and last received data bit as MSB).

In addition to the serial data transmission function, UART provides hardware handshaking signal support for controlling the serial data flow. UART Figure 2.28 illustrates the same.

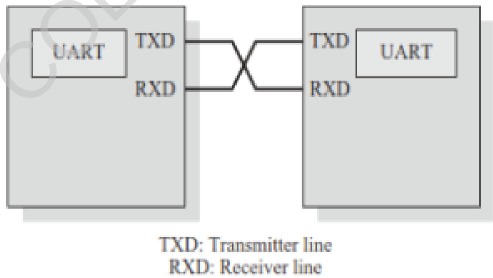

Fig. 2.28 UART Interfacing

Parallel Interface

The on-board parallel interface is normally used for communicating with peripheral devices which are memory mapped to the host of the system. The host processor/controller of the embedded system contains a parallel bus and the device which supports parallel bus can directly connect to this bus system. The communication through the parallel bus is controlled by the control signal interface between the device and the host. The 'Control Signals' for communication includes 'Read/Write' signal and device select signal. The device normally contains a device select line and the device becomes active only when this line is asserted by the host processor. The direction of data transfer (Host to Device or Device to Host) can be controlled through the control signal lines for 'Read' and 'Write'. Only the host processor has control over the 'Read' and 'Write' control signals. The device is normally memory mapped to the host processor and a

Mr. Manjunatha I R Assistant professor ECE, CEC range of address is assigned to it. An address decoder circuit is used for generating the chip select signal for the device. When the address selected by the processor is within the range assigned for the device, the decoder circuit activates the chip select line and thereby the device becomes active. The processor then can read or write from or to the device by asserting the corresponding control line (RD\ and WR\ respectively). Strict timing characteristics are followed for parallel communication. As mentioned earlier, parallel communication is host processor initiated. If a device wants to initiate the communication, it can inform the same to the processor through interrupts. For this, the interrupt line of the device is connected to the interrupt line of the processor and the corresponding interrupt is enabled in the host processor. The width of the parallel interface is determined by the data bus width of the host processor. It can be 4bit, 8bit, 16bit, 32bit or 64bit etc. The bus width supported by the device should be same as that of the host processor. The bus interface diagram shown in Fig. 2.30 illustrates the interfacing of devices through parallel interface.

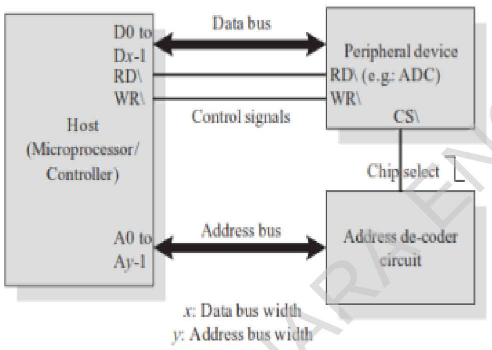

Fig. 2.30 Parallel Interface Bus

Parallel data communication offers the highest speed for data transfer.

Universal Serial Bus (USB)

Universal Serial Bus (USB) is a wired high-speed serial bus for data communication. The USB communication system follows a star topology with a USB host at the center and one or more USB peripheral devices/USB hosts connected to it. A USB 2.0 host can support connections up to 127, including slave peripheral devices and other USB hosts. Figure 2.32 illustrates the star topology for USB device connection.

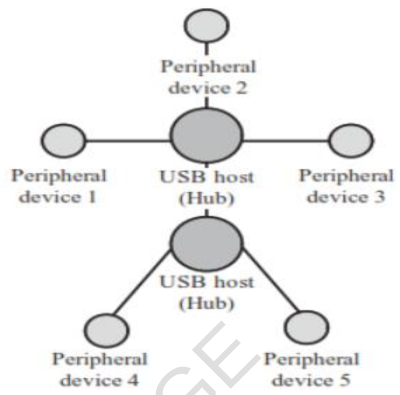

Fig. 2.32 USB Device Connection topology

- USB transmits data in packet format. Each data packet has a standard format. The USB communication is a host initiated one.
- The USB host contains a host controller which is responsible for controlling the data communicate-on, including establishing connectivity with USB slave devices, packetizing and formatting the data packet.
- There are different standards for implementing the USB Host Control interface; namely Open Host Control Interface (OHCI) and Universal Host Control Interface (UHCI).
- The physical connection between a USB peripheral device and master device is established with a USB cable.
- The USB cable in USB 2.0 specification supports communication distance of up to 5 meters. The USB 2.0 standard uses two different types of connectors at the ends of the USB cable for connecting the USB peripheral device and host device.
- ‘Type A’ connector is used for upstream connection (Connection with host) and Type B OR Mini/Micro USB connector is used for downstream connection (Connection with slave device).
- The USB 2.0 connector seen at desktop PCs or laptops are examples for ‘Type A’ USB connector. Both Type A and Type B connectors contain 4 pins for communication.
- The Pin details for the USB 2.0 Type A & B connectors are listed in the table given below.

Pin no.	Pin no.	Pin no.
1	V_{BUS}	Carries power (5V)
2	D–	Differential data carrier line
3	D+	Differential data carrier line
4	GND	Ground signal line

- USB uses differential signals for data transmission. It improves the noise immunity. USB interface has the ability to supply power to the connecting devices. Two connection lines (Ground and Power) of the USB interface are dedicated for carrying power.
- A Standard Downstream USB 2.0 Port (SDP) can supply power up to 500 mA at 5 V, whereas a Charging Downstream USB 2.0 Port (CDP) can supply power up to 1500 mA at 5 V. It is sufficient to operate low power devices.
- Mini and Micro USB connectors are available for small form factor devices like portable media players and Smartphones.
- USB supports four different types of data transfers, namely; Control, Bulk, Isochronous and Interrupt. Control transfer is used by USB system software to query, configure and issue commands to the USB device.
- Bulk transfer is used for sending a block of data to a device. Bulk transfer supports error checking and correction. Transferring data to a printer is an example for bulk transfer.
- Isochronous data transfer is used for real-time data communication. In Isochronous transfer, data is transmitted as streams in real-time.
- Isochronous transfer doesn't support error checking and re-transmission of data in case of any transmission loss.
- All streaming devices like audio devices and medical equipment for data collection make use of the isochronous transfer.
- Interrupt transfer is used for transferring small amount of data. Interrupt transfer mechanism makes use of polling technique to see whether the USB device has any data to send.
- Devices like Mouse and Keyboard, which transmits fewer amounts of data, makes use of Interrupt transfer.
- USB 3.x is the latest version of the USB standard for peripheral connectivity. USB 3.x brings a number of improvements over USB 2.0, and adds a new transfer mode called SuperSpeed (SS), capable of transferring data at speeds up to 4.8 Gbps, which is more than ten times as fast as the 480 Mbps high speed of USB 2.0.

Wi-Fi

- It is the popular wireless communication technique for networked communication of devices.
- Wi-Fi follows the IEEE 802.11 standard.
- Wi-Fi is intended for network communication and it supports Internet Protocol (IP) based communication.
- Wi-Fi based communications require an intermediate agent called Wi-Fi router/Wireless Access point to manage the communications
- The Wi-Fi router is responsible for restricting the access to a network, assigning IP address to devices on the network, routing data packets to the intended devices on the network.
- Wi-Fi enabled devices contain a wireless adaptor for transmitting and receiving data in the form of radio signals through an antenna.
- Wi-Fi operates at 2.4GHZ or 5GHZ of radio spectrum and they co-exist with other ISM band devices like Bluetooth.
- A Wi-Fi network is identified with a Service Set Identifier (SSID). A Wi-Fi device can connect to a network by selecting the SSID of the network and by providing the credentials if the network is security enabled.
- Wi-Fi networks implements different security mechanisms for authentication and data transfer.

- Wireless Equivalency Protocol (WEP), Wireless Protected Access (WPA) etc are some of the security mechanisms supported by Wi-Fi networks in data communication.
- It supports a data rate of up to 24 Mbps and a range of approximately 100 feets for data communication.

General Packet Radio Service (GPRS), 3G, 4G, LTE

- General Packet Radio Service (GPRS), 3G, 4G and LTE are cellular communication technique for transferring data over a mobile communication network like GSM and CDMA.
- Data is sent as packets in GPRS communication. The transmitting device splits the data into several related packets.
- At the receiving end the data is re-constructed by combining the received data packets.
- GPRS supports a theoretical maximum transfer rate of 171.2kbps. In GPRS communication, the radio channel is concurrently shared between several users instead of dedicating a radio channel to a cell phone user.
- The GPRS communication divides the channel into 8 timeslots and transmits data over the available channel.
- GPRS supports Internet Protocol (IP), Point to Point Protocol (PPP) and X.25 protocols for communication. GPRS is mainly used by mobile enabled embedded devices for data communication.
- The device should support the necessary GPRS hardware like GPRS modem and GPRS radio. To accomplish GPRS based communication, the carrier network also should have support for GPRS communication.
- cellular data communication techniques like 3G (3rd Generation), High Speed Downlink Packet Access (HSDPA), 4G (4th Generation), LTE (Long Term Evolution) etc. which offers higher bandwidths for communication. 3G offers data rates ranging from 144Kbps to 2Mbps or higher, whereas 4G gives a practical data throughput of 2 to 100+ Mbps depending on the network and underlying technology.