MODULE III

Embedded Systems: Definition, Embedded systems vs. General computing systems, Classification of Embedded Systems, Major application areas of Embedded Systems, Elements of an Embedded System, Core of the Embedded System, Microprocessor vs. Microcontroller, RISC vs. CISC, and Harvard vs. Von- Neumann.

Sensors and Interfacing: Instrumentation and control systems, Transducers, Sensors.

Actuators: LED, 7-Segment LED Display, Stepper Motor, Relay, Piezo Buzzer, Push Button Switch, Keyboard.

Communication Interface: UART, Parallel Interface, USB, Bluetooth, Wi-Fi, GPRS.

3.1 Embedded Systems

- An Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software).
- E.g. Electronic Toys, Mobile Handsets, Washing Machines, Air Conditioners, Automotive Control Units, Set Top Box, DVD Player.

3.1.1 Embedded Systems vs. General Computing Systems

No.	General Purpose Computing System	Embedded System (ES)
1.	A system which is a combination of generic hardware and General Purpose Operating System for executing a variety of applications.	A system which is a combination of special purpose hardware and embedded Operating System for executing a specific set of applications.
2.	Contain a General Purpose Operating System (GPOS).	May or may not contain an operating system for functioning.
3.	Applications are alterable (programmable) by user. (The end user can re-install the OS, and add or remove user applications).	The firmware of the embedded system is pre-programmed and it is non-alterable by end-user.
4.	Performance is the key deciding factor on the selection of the system. Always "Faster is Better".	Application specific requirements (like performance, power requirements, memory etc) are the key deciding factors.
5.	Less/not at all tailored towards reduced operating power requirements, options for different levels of power management.	Highly tailored to take advantage of the power saving modes supported by hardware and Operating System.

6.	Response requirements are not time critical.	For certain category of ES, the response time requirement is highly critical.
7.	Need not be deterministic in execution behavior.	Execution behavior is deterministic for few ES like "Hard Real Time" systems.

3.1.2 Classification of Embedded Systems:

- The criteria used in the classification of embedded systems are:
  1. Based on generation.
  2. Complexity and performance requirements.
  3. Based on deterministic behaviour.
  4. Based on triggering.

1. Classification Based on Generation

- **First Generation:** The early embedded systems were built around 8-bit microprocessors like 8085 and 280, and 4-bit microcontrollers. Simple in hardware circuits with firmware developed in Assembly code.
  Eg: Digital telephone keypads, stepper motor control units etc.
- **Second Generation:** Embedded systems built around 16-bit microprocessors and 8 or 16-bit microcontrollers. The instruction set became much more complex and powerful than the first generation processors/controllers.
  Eg: Data Acquisition Systems, SCADA systems etc.
- **Third Generation:** ES built around powerful 32-bit processors and 16-bit microcontrollers. Application and domain specific processors /controllers like Digital Signal Processors (DSP) and Application Specific Integrated Circuits (ASICs) came into the picture. The instruction set is more complex and powerful.
  Eg: Robotics, media, industrial process control, networking, etc.
- **Fourth Generation:** Embedded system built around System on Chips (SOC), reconfigurable processors and multicore processors. The fourth generation embedded systems are making use of high performance real time embedded operating systems for their functioning.
  Eg: Smart phone devices, mobile internet devices (MIDs), etc.

2. Classification Based on Complexity and Performance

- **Small-Scale Embedded Systems:** Small-scale ES are usually built around low performance and low cost 8 or 16 bit microprocessors / microcontrollers. These

ES are suitable for simple applications and where performance is not time critical. It may or may not contain an operating system (OS) for functioning.
Eg: Electronic toy.

- **Medium-Scale Embedded Systems:** Embedded systems built around medium performance, low cost 16 or 32 bit microprocessors/microcontrollers or DSPs. These embedded systems which are slightly complex in hardware and firmware (software) requirements. It usually contains an Embedded OS for functioning.
- **Large-Scale Embedded Systems/Complex Systems:** ES built around 32 or 64 bit RISC processors/controllers or Reconfigurable System on Chip (RSC) or multi-core processors and programmable logic devices. These embedded systems involve highly complex hardware and firmware. They may contain multiple processors/controllers and co-units/hardware accelerators. Complex embedded systems usually contain a high performance Real Time Operating System (RTOS).

3. Classification based on deterministic system behaviour

- It is applicable for Real Time systems. The application/task execution behavior for an embedded system can be either deterministic or non-deterministic. Classified into:
  - **Soft Real Time Systems:** Missing a deadline may not be critical and can be tolerated to a certain degree. Eg: ATM.
  - **Hard Real Time Systems:** Missing any deadline may produce disastrous results (financial, human loss of life, etc.). Eg: ABS, Air bags etc.

4. Classification based on triggering

- Embedded Systems which are 'Reactive' in nature can be classified based on the trigger. Reactive systems can be classified as **event triggered** or **time triggered**.

3.1.3 Major Application Areas of Embedded Systems

- Consumer Electronics: Camcorders, Cameras etc.
- Household Appliances: Television, Washing Machine, Fridge, Microwave Oven etc.
- Home Automation and Security Systems: Air conditioners, CCTVs, Fire alarms etc.
- Automotive Industry: Anti-lock Breaking Systems (ABS), Engine Control, Automatic Navigation Systems etc.
- Telecom: Cellular Telephones, Telephone switches, Handset etc.
- Computer Peripherals: Printers, Scanners, Fax machines etc.
- Computer Networking Systems: Network Routers, Switches, Hubs, Firewalls etc.

- Health Care: Different Kinds of Scanners, EEG, ECG Machines etc.
- Measurement & Instrumentation: Digital multi meters, Digital CRO, PLC systems etc.
- Banking & Retail: Automatic Teller Machines (ATM), Currency counters etc.
- Card Readers: Barcode, Smart Card Readers, Hand held Devices etc.

3.1.4 Elements of Embedded Systems:

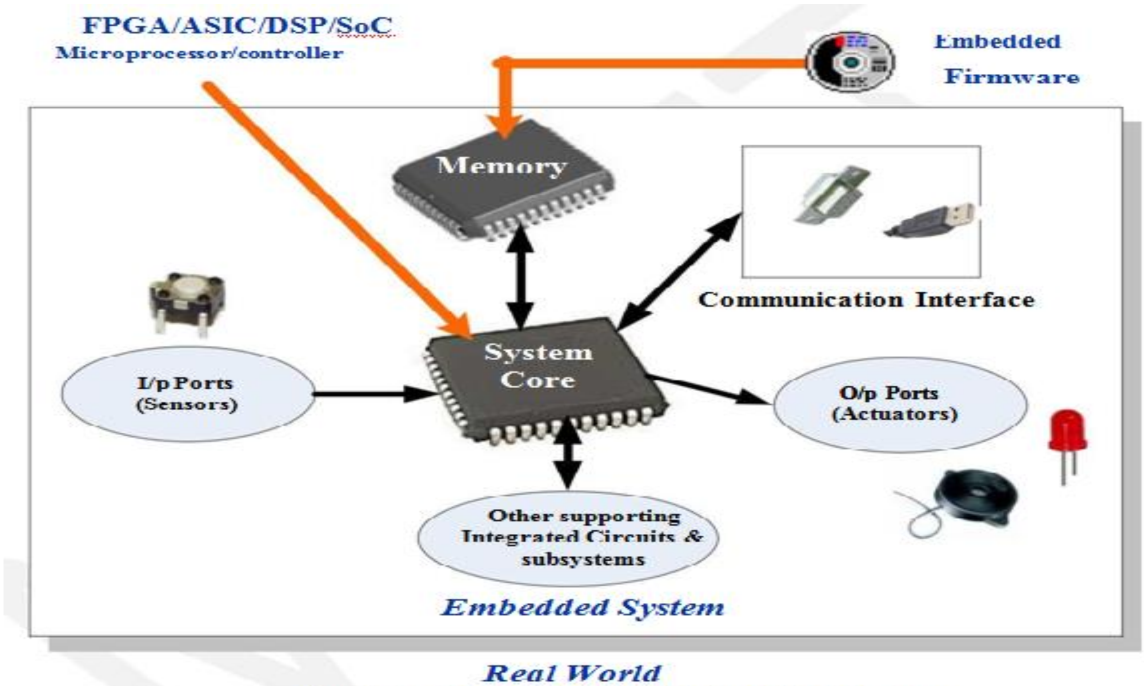


Fig. 3.1: Elements of an Embedded System

- An embedded system is a combination of 3 things: Hardware, Software, Mechanical Components and it is supposed to do one specific task only.
- 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 3.1.
- ESs are designed to regulate a physical variable or to manipulate the state of some devices by sending signals to the actuators or devices connected to the output system, 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.
- The control is achieved by processing the information coming from the sensors and user interfaces, and controlling some actuators that regulate the physical variable.
- Keyboards, push button, switches, etc. are examples of input devices and LEDs, LCDs, Piezoelectric buzzers, etc. are examples for output devices for a typical ES.

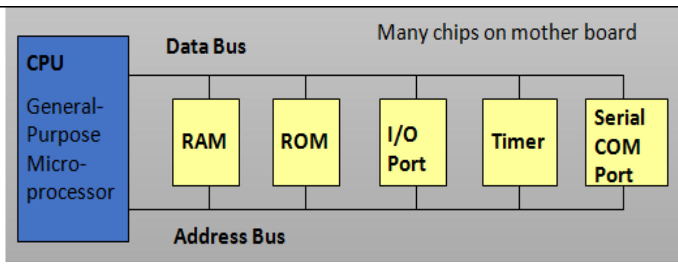
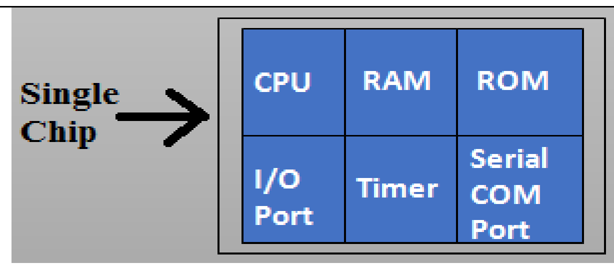
- Some ES can automatically sense input parameters from real world through sensors. Sensor information is passed to the processor after signal conditioning and digitization.
- The core of the system performs some predefined operations on input data with the help of embedded firmware and sends some actuating signals to the actuator.
- The memory of the system is responsible for holding the code. There are two types:
  - Fixed memory (ROM) is used for storing code or program. The user cannot do any modifications in this type of memory. Common types used are OTP, PROM, EPROM, EEPROM & Flash memory.
  - Temporary memory (RAM) is used for performing arithmetic operations or control algorithm executions. Common types used are SRAM, DRAM and NVRAM.
- Memory for implementing the code may be present on the processor or may be implemented as a separate chip interfacing the processor.
- In a controller based ES, the controller may contain internal memory for storing code and such controllers are called Micro-controllers with on-chip ROM, eg. Atmel AT89C51.

3.1.5 Core of the Embedded System:

- The core of the embedded system falls into any oneof the following categories.
  1. General Purpose and Domain Specific Processors
     - Microprocessors
     - Microcontrollers
     - Digital Signal Processors
  2. Programmable Logic Devices (PLDs)
  3. Application Specific Integrated Circuits (ASICs)
  4. Commercial off-the-shelf Components (COTS)

3.1.6 Microprocessors vs. Microcontrollers

No.	Microprocessors	Microcontrollers
1.	A silicon chip representing a Central Processing Unit(CPU), which is capable of performing arithmetic as well as logical operations according to a pre-defined set of Instructions.	A microcontroller is a highly integrated chip that contains a CPU, 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

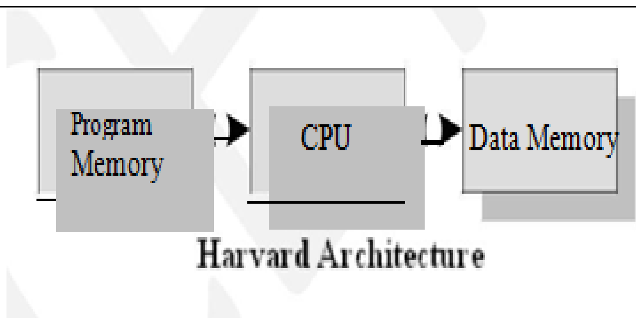
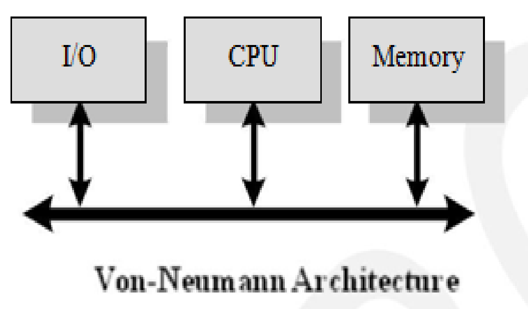
2.	It is a dependent unit. It requires the combination of other chips like Timers, Program and data memory chips, Interrupt controllers etc. for functioning.	It is a self contained unit and doesn't require external Interrupt Controller, Timer, and UART etc. for its functioning.
3.	Most of the time general purpose in design and operation.	Mostly application oriented or domain specific.
4.	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.	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 or as individual port pins.
5.	Targeted for high end market where performance is important.	Targeted for embedded market where performance is not so critical.
6.	Limited power saving options.	Includes lot of power saving features
7.		

3.1.7 RISC V/s CISC Processors/Controllers:

No.	RISC Processors/Controllers	CISC Processors/Controllers
1.	Lesser no. of instructions.	Greater no. of Instructions.
2.	Instruction Pipelining and increased execution speed.	Generally no instruction pipelining feature.
3.	Orthogonal Instruction Set.	Non Orthogonal Instruction Set.
4.	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
5.	Large number of registers are available	Limited no. of general purpose registers

6.	Programmer needs to write more code to execute a task since the instructions are simpler ones.	A programmer can achieve the desired functionality with a single instruction.
7.	Single, Fixed length Instructions.	Variable length Instructions.
8.	Less Silicon usage and pin count.	More silicon usage.
9.	With Harvard Architecture	Harvard or Von-Neumann Architecture

3.1.8 Harvard V/s Von-Neumann Processor/Controller Architecture

No.	Harvard Architecture	Von-Neumann Architecture
1.	Microprocessors/controllers based on the **Harvard** architecture will have separate data bus and instruction bus. This allows the data transfer and program fetching to occur simultaneously on both buses.	Microprocessors/controllers based on the **Von-Neumann** architecture shares a single bus for fetching both instructions and data. Program instructions & data are stored in a common main memory.
2.	Separate buses for Instruction and Data fetching.	Single shared bus for Instruction and Data fetching.
3.	Easier to Pipeline, so high performance can be achieved.	Low performance Compared to Harvard Architecture.
4.	Comparatively high cost.	Cheaper.
5.	No memory alignment problems.	Allows self modifying codes.
6.		

3.2 Sensors & Interfacing:

3.2.1 Instrumentation and control systems:

- An instrument is a device that measures or manipulates, processes physical variables such as flow, temperature, level, or pressure etc.
- Figure 3.2(a) 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.
- The output produced by the sensor may be small or may suffer from the presence of noise (i.e. unwanted signals), hence signal conditioning will be required to make it in an acceptable form for signal processing, display and recording.

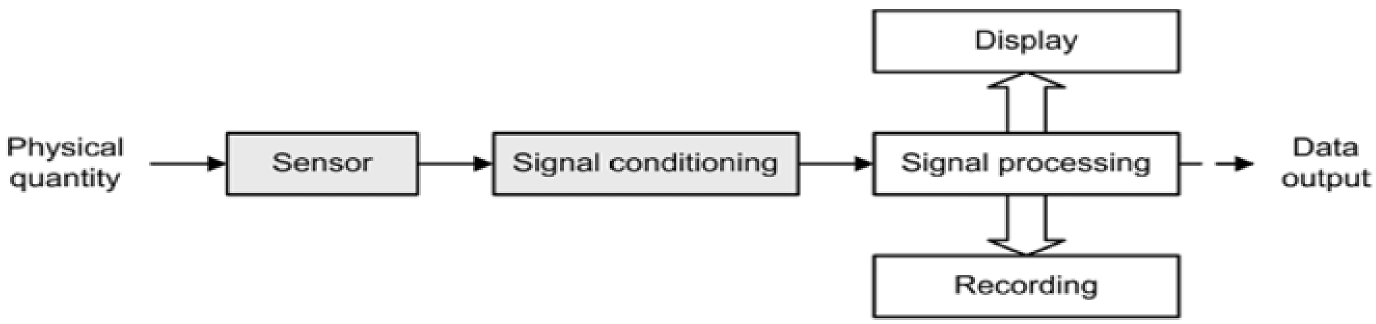


Fig. 3.2 (a) An instrumentation system

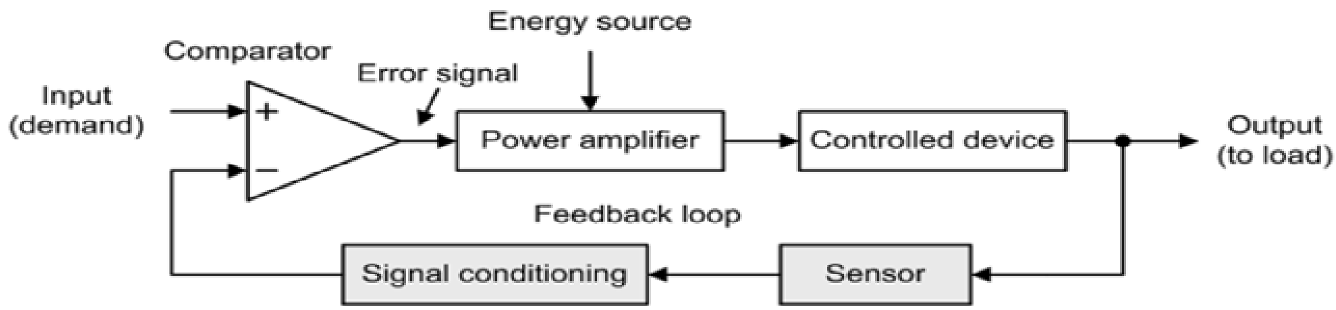


Fig. 3.2 (b) A control system

- Fig. 3.2(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 and leave the system to regulate itself by comparing it with a signal derived from the output (via a sensor and appropriate 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.
- 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.).

3.2.2 Transducers

- Transducers are devices that convert energy in the form of sound, light, heat, etc., into an equivalent electrical signal, or vice versa.
- Transducers may be used both as inputs to electronic circuits and outputs from them.
  - Output Transducer: Eg: A loudspeaker is a transducer that converts low frequency electric current into audible sounds.
  - Input Transducer: Eg: A microphone is a transducer that converts sound pressure variations into voltage or current.
- Some examples of input transducers and output transducers are given in table 1.1 and table 1.2 respectively.

Table 1.1: Some examples of input transducers

Physical quantity	Input transducer	Notes
Sound (pressure change)	Dynamic microphone	Diaphragm attached to a coil is suspended in a magnetic field. Movement of the diaphragm causes current to be induced in the coil.
Temperature	Thermocouple (see	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 1.2: Some examples of output transducers

Physical quantity	Output transducer	Notes
Sound (pressure change)	Loudspeaker	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 per revolution) and 1.8° (200 steps per revolution).

3.2.3 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, light level, etc.
- The signals returned from a sensor, together with control inputs will 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: **active** or **passive**.
  - **Active Sensors**: This requires an external excitation signal or a power signal.
  - **Passive Sensors** do not require any external power signal and directly generates output response.
- 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 output of an analogue sensor can take any one of an infinite number of voltage or current levels. It is thus said to be continuously variable.
- Sensors act as an input device.

3.3 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.

3.3.1 Light emitted Diode (LED)

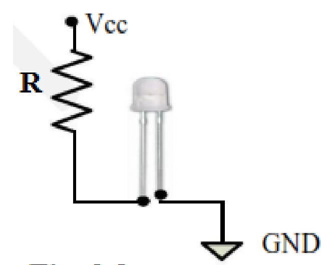


Fig. 3.3: LED interfacing circuit

- Light Emitting Diode (LED) is an output device for visual indication in any embedded system.
- Used as an indicator for the status of various signals or situations.
- LED is a p-n junction diode and it contains an anode & 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.
- A resister is used in series between the power supply and the resistor to limit the current through the LED.
- The ideal LED interfacing circuit is shown in figure 3.3.

3.3.2 7-Segment LED Display

- The 7 – segment LED display is an output device for displaying alpha numeric characters & contains 8 LED segments arranged in a special form.
- Out of the 8 LED segments, 7 are used for displaying alpha numeric characters.
- LED segments are named A to G and the decimal point LED segment is named as DP.
- The 7 – segment LED displays are available in two different configurations:
    - **Common anode configuration:** Anodes of 8 segments are connected commonly.
    - **Common cathode configuration:** 8 LED segments share a common cathode line.
- Figure 3.4 shows the common anode & cathode configurations of a 7 segment LED.
- The current flow through each of the LED segments should be limited to the maximum value supported by the LED display unit by connecting a current limiting resistor.

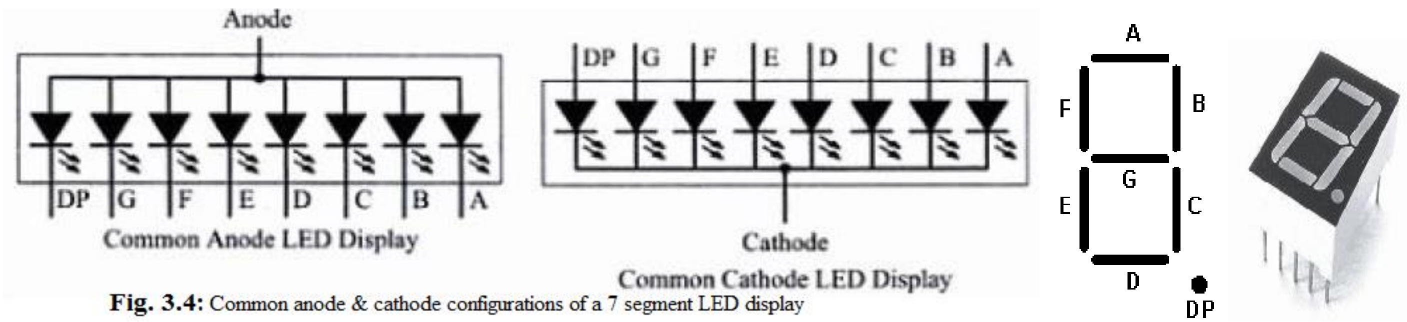


Fig. 3.4: Common anode & cathode configurations of a 7 segment LED display

3.3.3 Stepper Motor

- 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.

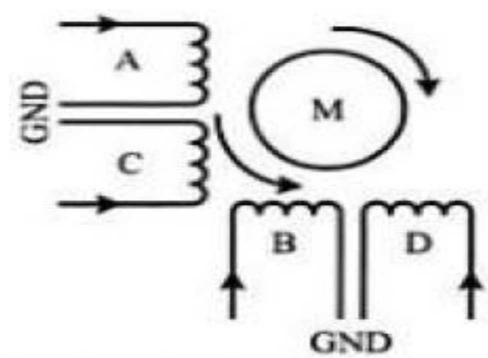


Fig. 3.5: 2 Phase Uni Polar Stepper Motor

- Stepper motors are widely used in industrial embedded applications, consumer electronic products and robotics control systems.
- Eg: Paper feed mechanism of printer/fax makes use of stepper motor for its functioning.
- Based on the coil winding arrangements, a two phase stepper motor is classified into:
- **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
- **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.

3.3.4 Relay

- A relay is an electro mechanical device which 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 an armature with one or more contacts.
- 'Relay' works on electromagnet 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.
- The Relay is normally controlled using a relay driver circuit connected to the port pin of the processor/controller.
- Figure 3.6 shows widely used relay configurations for embedded applications.

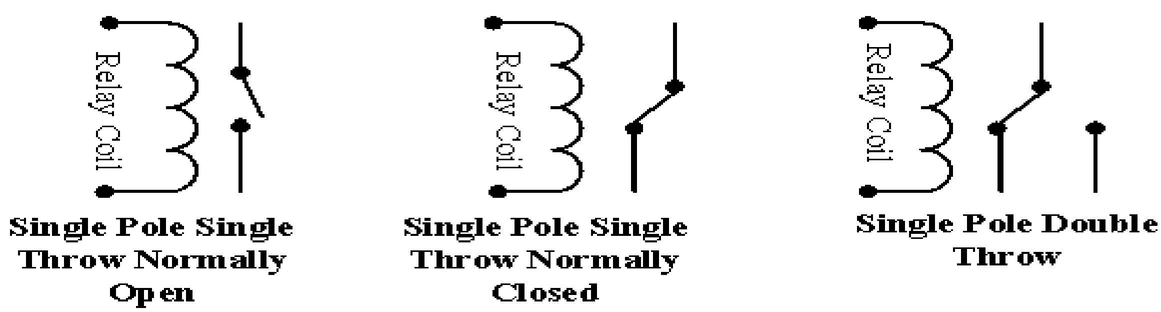


Fig. 3.6: Relay Configurations

3.3.5 Piezo Buzzer

- It is a piezoelectric device for generating audio indications in embedded applications.
- A Piezo 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'
    - Self-driving circuits 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/Controller.

3.3.6 Push button switch

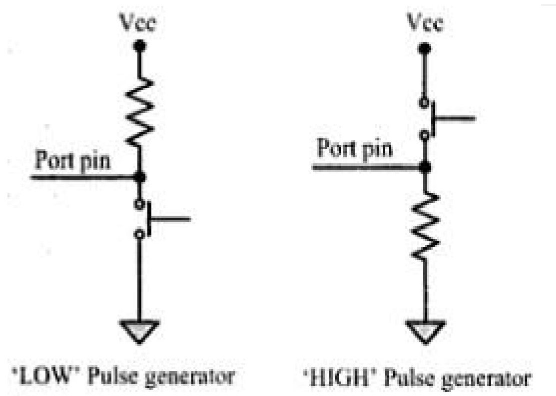


Fig. 3.7: Push Button Switch Configuration

- Push Button switch is an input device.
- This switch comes in two configurations, namely: "Push to Make" and "Push to Break".
- The switch is normally in the open state and it makes a circuit contact when it is pushed or pressed in the "Push to Make" configuration.
- In the "Push to Break" configuration, the switch is normally in the closed state & it breaks the circuit contact when it is pushed or pressed.
- The push button stays in the "closed" (For Push to Make type) or "open" (For Push to Break type) state as long as it is kept in the pushed state and it breaks/makes the circuit connection when it is released.
- Push button is used for generating a momentary pulse.
- In embedded application push button is used as reset, start switch & pulse generator.

3.3.7 Keyboard

- Keyboard is an input device for user interface.
- If the number of keys required is very less, push button switches can be used.
- If a large number of keys are required, then Matrix keyboard is used.
- Figure 3.8 illustrates the connection of keys in a matrix keyboard.
- In a matrix keyboard, the keys are arranged in matrix fashion.
- For detecting a key press, each row of the matrix is pulled low & the columns are read.
- After reading the status of each columns 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 the row is pressed, reading the

column to which the key is connected will give logic 0.

- Pull-up resistors are connected to the column lines to limit the current that flows to the row line on a key press.

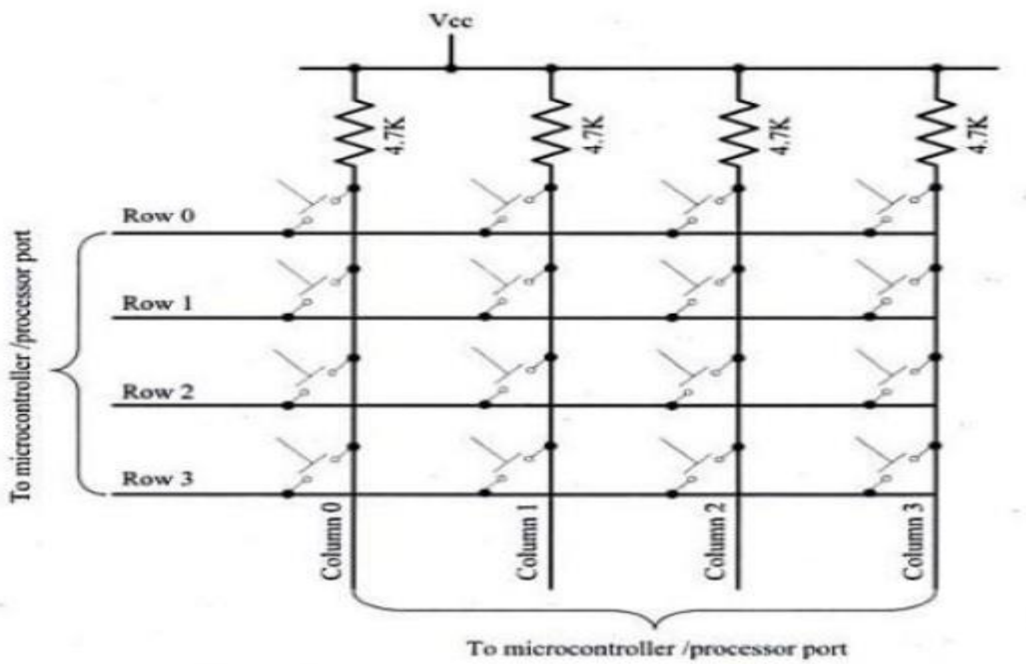


Fig. 3.8: Matrix Keyboard Interfacing

3.4 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):** The communication channel which interconnects the various components within an embedded product is referred as Device/board level communication interface.

    Eg: Serial interfaces like I2C, SPI, UART, 1-Wire etc and Parallel bus interface

    - **Product level communication interface (External Communication Interface):** The Product level communication interface is responsible for data transfer between the embedded system and other devices or modules. The external communication interface can be either wired media or wireless media and it can be a serial or parallel interface.

    Eg: Infrared (IR), Bluetooth (BT), Wireless LAN (Wi-Fi), Radio Frequency waves (RF), GPRS etc. (wireless) and RS-232, USB, Parallel port etc. (wired).

3.4.1 Universal Asynchronous Receiver Transmitter (UART)

- UART based data transmission is an asynchronous form of serial data transmission.
- It 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.
- Serial communication settings for both transmitter & receiver should be set as identical.
- Start & stop of communication is indicated through inserting special bits in data stream
- The 'start' bit informs the receiver that a data byte is about to arrive.
- The receiver device starts polling it's 'receive line'.
- 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.
- For proper communication, the Transmit line (TX) of the sending device should be connected to the Receive line (RX) of the receiving device.
- Figure 3.9 illustrates UART interfacing.

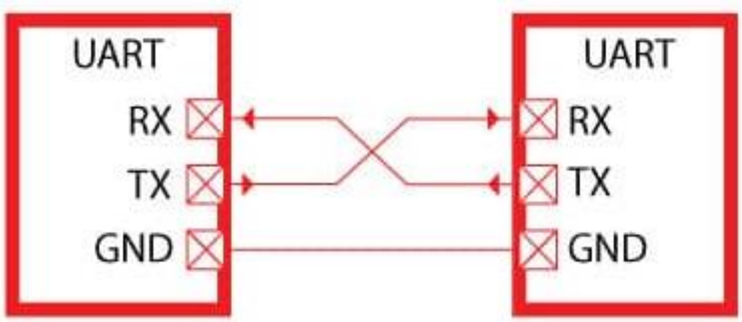


Fig. 3.9: UART Interfacing

3.4.2 Parallel Interface

- 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 are read or write signals and device select signals. The device becomes active by selecting host processor.
- Direction of data transfer is controlled through control signal lines for 'read' and 'write'.

- An address decoder circuit is used for generating chip select signal for the device. When the address selected is in the range, chip select line is activated by decoder circuit.
- If device wants to start communication, it can inform the same to processor through interrupts.
- The width of the parallel interface is determined by the data bus width of the host processor (4bit, 8bit, 16bit, 32bit or 64bit).Parallel data communication offers highest speed for data transfer.
- Figure 3.10 illustrates the interfacing of devices through parallel interface.

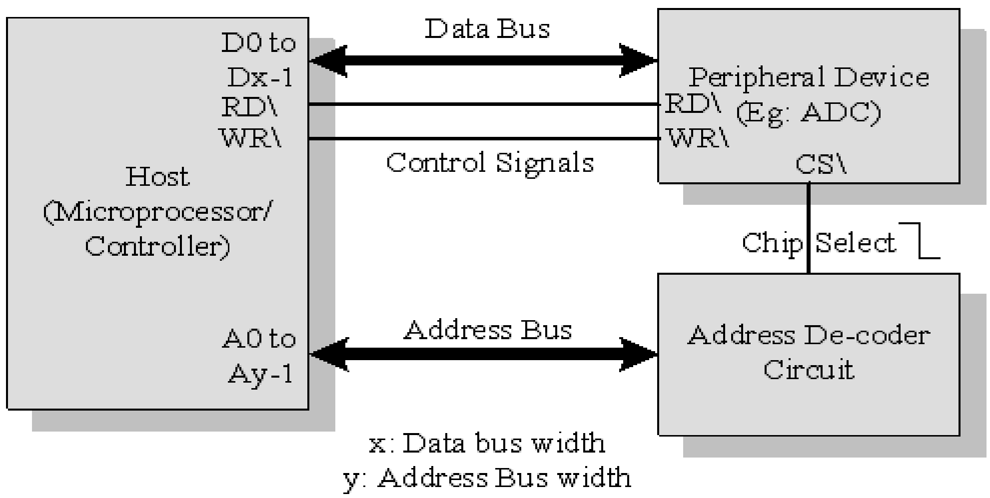


Fig. 3.10: Parallel Interface Bus

3.4.3 Universal Serial Bus (USB)

- 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 centre and one or more USB peripheral devices/USB hosts connected to it.
- Physical connection between a USB peripheral device and master device is established with a USB cable.
- USB cable supports communication distance of up to 5 meters.
- A USB host can support connections up to 127 devices.
- USB transmits data in packet format. Each data packet has a standard format.
- Figure 3.11 illustrates the star topology for USB device connection.
- USB host contains a host controller responsible for controlling the data communication.
- The USB standard uses two different types of connectors namely:
    - Type A' connector: used for upstream connection (connection with host).
    - 'Type B' connector: used for downstream connection (connection with slave device)

- Both Type A and Type B connectors contain 4 pins for communication. (Table)

Pin No:	Pin Name	Description
1	V_{BUS}	Carries power (5V)
2	D-	Differential data carrier line
3	D+	Differential data carrier line
4	GND	Ground signal line

- Each USB device contains a Product ID (PID) and a Vendor ID (VID).
- The PID and VID are embedded into the USB chip by the USB device manufacturer.
- The VID for a device is supplied by the USB standards forum.
- USB supports four different types of data transfers:
    - **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 (Printer).
    - **Isochronous** data transfer is used for real-time data communication (Audio).
    - **Interrupt** transfer is used for transferring small amount of data (Keyboard).
- USB supports four different data transfers: Low Speed (USB 1.0: 1.5 Mbps), Full Speed (USB 1.1: 12 Mbps),High Speed(USB 2.0: 480 Mbps) & Super Speed(USB 3.0: 4.8 Gbps).

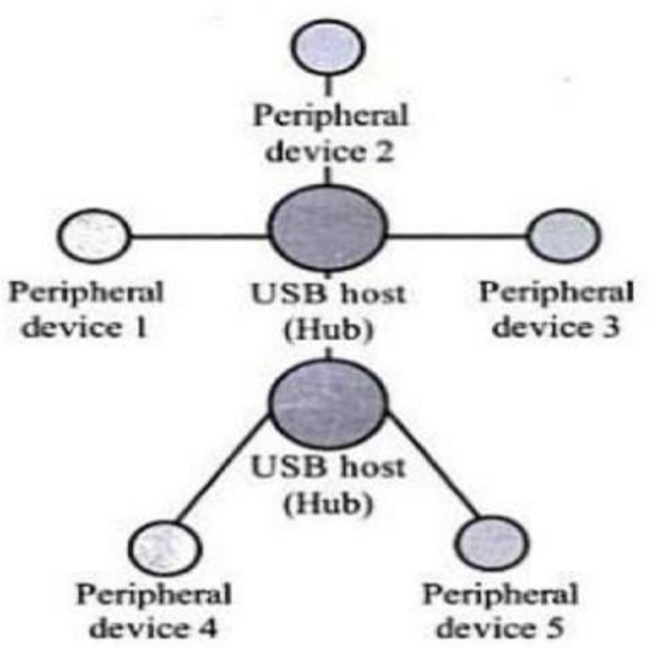
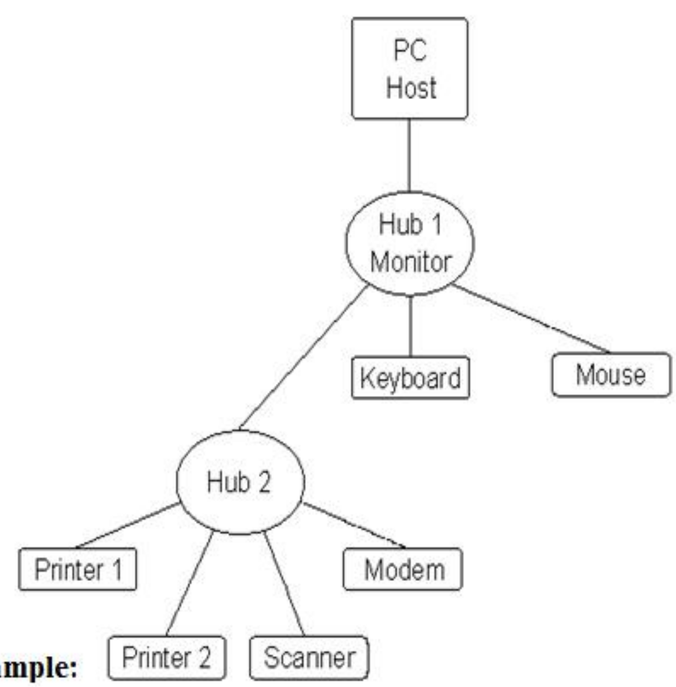


Fig. 3.11: USB Device Connection topology

Example:



3.4.4 Wireless Fidelity (Wi-Fi)

- Wi-Fi is the popular wireless communication technique for networked communication of devices.
- Wi-Fi follows the IEEE 802.11 standard.
- Wi-Fi supports Internet Protocol (IP) based communication.
- In an IP based communication each device is identified by an IP address, which is unique to each device on the network.
- Wi-Fi based communications require an intermediate agent called Wi-Fi router/ Wireless Access point to manage the communications.

Fig. 3.12: Wi-Fi network

- 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 Radio).
- Wi-Fi operates at 2.4GHz or 5GHz of radio spectrum.
- Figure 3.12 illustrates the typical interfacing of devices in a Wi-Fi network.
- For communicating, device when its Wi-Fi radio is turned ON, searches available Wi-Fi network in its vicinity & lists out the Service Set Identifier (SSID) of available networks.
- If network is security enabled, password may be required to connect to particular SSID.
- Wi-Fi supports data rates ranging from 1Mbps to 150Mbps depending on the standards (802.11a/b/g/n).
- Depending on the type of antenna and usage location (indoor/outdoor), Wi-Fi offers a range of 100 to 300 feet.

3.4.5 General Packet Radio Service (GPRS)

- GPRS is a communication technique for transferring data over a mobile communication network like GSM.
- 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 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.
- GPRS is an old technology and it is being replaced by new generation data communication techniques (EDGE).

Important Questions

1. Define an Embedded System (ES). Discuss the major application areas of ES.
2. Briefly explain the different criteria used in the classification of embedded system.
3. With block diagram, explain the elements of an embedded system.
4. Define the following with their types and examples:
   i) Transducers ii) Actuators iii) Sensors.
5. With block diagram, explain an Instrumentation and Control system.
6. With diagram, explain the following output devices:
   i) LED ii) 7 Segment LED Display iii) Relay.
7. With diagram, explain the following input devices:
   i) Keyboard ii) Push button switch iii) Relay.
8. Write a short note on:
   i) Stepper Motor ii) Piezo Buzzer iii) GPRS
9. What is Communication Interface? How they are classified? Give examples.
10. With diagram, explain the following Device/board level communication interface:
    i) UART ii) Parallel Interface.
11. With diagram, explain the following Product level communication interface:
    i) USB ii) Wi-Fi.
12. Differentiate between:
    i) General Computing system & Embedded System
    ii) Microprocessor & microcontroller
    iii) RISC & CISC Processor iv) Harvard & Von Neumann Architecture