EMBEDDED SYSTEM DESIGN

IV YEAR ECE

UNIT-I

INTRODUCTION TO EMBEDDED SYSTEMS

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY
Permanently Affliated to JNTUH and Approved by AICTE, New Delhi

1. Introduction to Embedded Systems

What is Embedded System? (DEC2016, March-2017.)

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 etc…

Embedded Systems are:

- ❑ Unique in character and behavior
- ❑ With specialized hardware and software

Embedded Systems Vs General Computing Systems: (March-2017)

General Purpose Computing System	Embedded System
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 OS for executing a specific set of applications
Contain a General Purpose Operating System (GPOS)	May or may not contain an operating system for functioning
Applications are alterable (programmable) by user (It is possible for the end user to re-install the Operating System, and add or remove user applications)	The firmware of the embedded system is pre-programmed and it is non-alterable by end-user
Performance is the key deciding factor on the selection of the system. Always „Faster is Better"	Application specific requirements (like performance, power requirements, memory usage etc) are the key deciding factors
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
Response requirements are not time critical	For certain category of embedded systems like mission critical systems, the response time requirement is highly critical
Need not be deterministic in execution behavior	Execution behavior is deterministic for certain type of embedded systems like „Hard Real Time" systems

History of Embedded Systems:

- First Recognized Modern Embedded System: Apollo Guidance Computer (AGC) developed by [Charles Stark Draper](#) at the MIT Instrumentation Laboratory.
  - It has two modules
  - 1.Command module(CM) 2.Lunar Excursion module(LEM)
  - RAM size 256 , 1K ,2K words
  - ROM size 4K,10K,36K words
  - Clock frequency is 1.024MHz
  - 5000 ,3-input RTL NOR gates are used
  - User interface is DSKY(display/Keyboard)

- First Mass Produced Embedded System: *Autonetics* ***D-17*** *Guidance computer for Minuteman-I missile*

Classification of Embedded Systems:

(**March-2017**)

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

1. Embedded Systems - Classification based on Generation

- **First Generation:** The early embedded systems built around 8-bit microprocessors like 8085 and Z80 and 4-bit microcontrollers

  EX. stepper motor control units, Digital Telephone Keypads etc.

  Second Generation: Embedded Systems built around 16-bit microprocessors and 8 or 16-bit microcontrollers, following the first generation embedded systems

  EX.SCADA, Data Acquisition Systems etc.

  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.

  EX. Robotics, industrial process control, networking etc.

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

EX Smart phone devices, MIDs etc.

2. Embedded Systems - Classification based on Complexity & Performance

- **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.

- **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.

- **Large Scale/Complex:** Embedded Systems built around high performance 32 or 64 bit RISC processors/controllers, RSoC or multi-core processors and PLD. It requires complex hardware and software. These system may contain multiple processors/controllers and co-units/hardware accelerators for offloading the processing requirements from the main processor. It contains RTOS for scheduling, prioritization and management.

3. Embedded Systems - Classification Based on deterministic behavior: It is applicable for Real Time systems. The application/task execution behavior for an embedded system can be either deterministic or non-deterministic

These are classified in to two types

1. Soft Real time Systems: Missing a deadline may not be critical and can be tolerated to a certain degree

2. Hard Real time systems: Missing a program/task execution time deadline can have catastrophic consequences (financial, human loss of life, etc.)

4. Embedded Systems - Classification Based on Triggering:

These are classified into two types

1. **Event Triggered :** Activities within the system (e.g., task run-times) are dynamic and depend upon occurrence of different events **.**

2. **Time triggered:** Activities within the system follow a statically computed schedule (i.e., they are allocated time slots during which they can take place) and thus by nature are predictable.

Major Application Areas of Embedded Systems:

- **Consumer Electronics:** Camcorders, Cameras etc.
- **Household Appliances:** Television, DVD players, washing machine, Fridge, Microwave Oven etc.
- **Home Automation and Security Systems:** Air conditioners, sprinklers, Intruder detection alarms, Closed Circuit Television Cameras, Fire alarms etc.
- **Automotive Industry:** Anti-lock breaking systems (ABS), Engine Control, Ignition Systems, Automatic Navigation Systems etc.
- **Telecom:** Cellular Telephones, Telephone switches, Handset Multimedia Applications 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 CROs, Logic Analyzers PLC systems etc.
- **Banking & Retail:** Automatic Teller Machines (ATM) and Currency counters, Point of Sales (POS)
- **Card Readers:** Barcode, Smart Card Readers, Hand held Devices etc.

Purpose of Embedded Systems: **(DEC2016)**

Each Embedded Systems is designed to serve the purpose of any one or a combination of the following tasks.

- Data Collection/Storage/Representation
- Data Communication
- Data (Signal) Processing
- Monitoring
- Control
- Application Specific User Interface

1. **Data Collection/Storage/Representation:-**

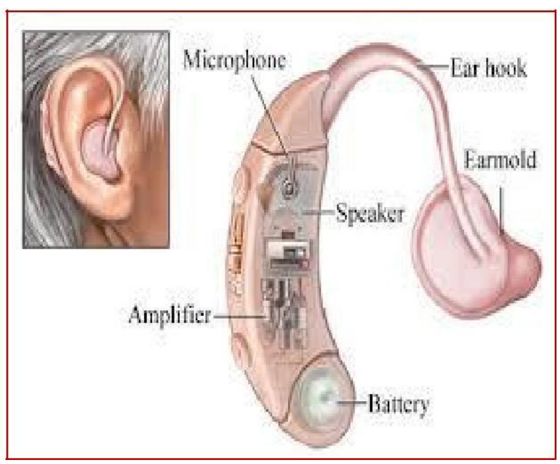
- Performs acquisition of data from the external world.
- The collected data can be either analog or digital
- Data collection is usually done for storage, analysis, manipulation and transmission
- The collected data may be stored directly in the system or may be transmitted to some other systems or it may be processed by the system or it may be deleted instantly after giving a meaningful representation

2. **Data Communication:-**

- Embedded Data communication systems are deployed in applications ranging from complex satellite communication systems to simple home networking systems
- Embedded Data communication systems are dedicated for data communication
- The data communication can happen through a wired interface (like Ethernet, RS-232C/USB/IEEE1394 etc) or wireless interface (like Wi-Fi, GSM,/GPRS, Bluetooth, ZigBee etc)

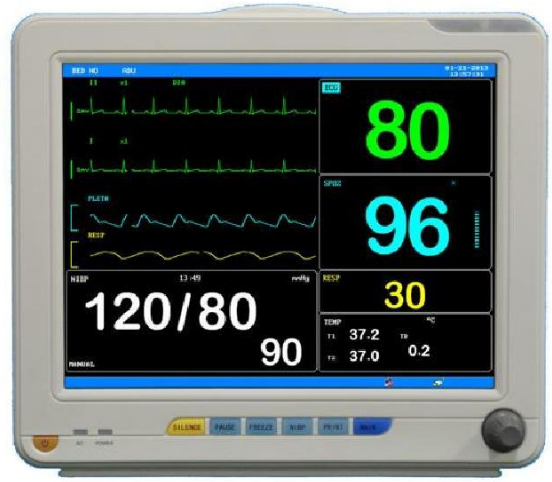
- Network hubs, Routers, switches, Modems etc are typical examples for dedicated data transmission embedded systems

3. **Data (Signal) Processing:-**



- Embedded systems with Signal processing functionalities are employed in applications demanding signal processing like Speech coding, synthesis, audio video codec, transmission applications etc
- Computational intensive systems
- Employs Digital Signal Processors (DSPs)

4. **Monitoring:-**



- Embedded systems coming under this category are specifically designed for monitoring purpose
- They are used for determining the state of some variables using input sensors
- They cannot impose control over variables.
- Electro Cardiogram (ECG) machine for monitoring the heart beat of a patient is a typical example for this
- The sensors used in ECG are the different Electrodes connected to the patient"s body
- Measuring instruments like Digital CRO, Digital Multi meter, Logic Analyzer etc used in Control & Instrumentation applications are also examples of embedded systems for monitoring purpose

5. **Control:-**

- Embedded systems with control functionalities are used for imposing control over some variables according to the changes in input variables
- Embedded system with control functionality contains both sensors and actuators
- Sensors are connected to the input port for capturing the changes in environmental variable or measuring variable
- The actuators connected to the output port are controlled according to the changes in input variable to put an impact on the controlling variable to bring the controlled variable to the specified range

Air conditioner for controlling room temperature is a typical example for embedded system with „Control" functionality

Air conditioner contains a room temperature sensing element (sensor) which may be a thermistor and a handheld unit for setting up (feeding) the desired temperature

The air compressor unit acts as the actuator. The compressor is controlled according to the current room temperature and the desired temperature set by the end user.

6. Application Specific User Interface:-

- Embedded systems which are designed for a specific application
- Contains Application Specific User interface (rather than general standard UI) like key board, Display units etc
- Aimed at a specific target group of users
- Mobile handsets, Control units in industrial applications etc are examples

EMBEDDED SYSTEM DESIGN PROCESS:

This section provides an overview of the embedded system design process aimed at two objectives. First, it will give us an introduction to the various steps in embedded system design before we delve into them in more detail. Second, it will allow us to consider the design methodology itself. A design methodology is important for three reasons. First, it allows us to keep a scorecard on a design to ensure that we have done everything we need to do, such as optimizing performance or performing functional tests. Second, it allows us to develop computer-aided design tools. Developing a single program that takes in a concept for an embedded system and emits a completed design would be a daunting task, but by first breaking the process into manageable steps, we can work on automating (or at least semi automating) the steps one at a time. Third, a design methodology makes it much easier for members of a design team to communicate.

The below Figure summarizes the major steps in the embedded system design process. In this top–down view, we start with the system requirements.

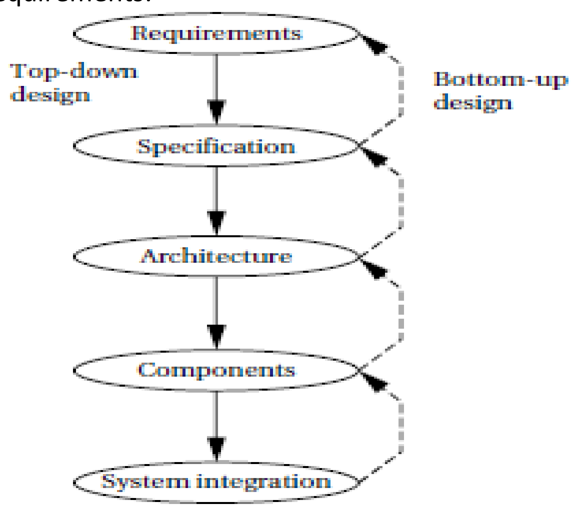


Fig: Major levels of abstraction in the design process

Requirements:

Clearly, before we design a system, we must know what we are designing. The initial stages of the design process capture this information for use in creating the architecture and components. We generally proceed in two phases: First, we gather an informal description from the customers known as requirements, and we refine the requirements into a specification that contains enough information to begin designing the system architecture.

Requirements may be ***functional*** or ***nonfunctional***. We must of course capture the basic functions of the embedded system, but functional description is often not sufficient. Typical nonfunctional requirements include:

■ *Performance:* The speed of the system is often a major consideration both for the usability of the system and for its ultimate cost. As we have noted, performance may be a combination of soft performance metrics such as approximate time to perform a user-level function and hard deadlines by which a particular operation must be completed.

■ *Cost:* The target cost or purchase price for the system is almost always a consideration. Cost typically has two major components: ***manufacturing cost*** includes the cost of components and assembly; ***nonrecurring engineering*** **(NRE)** costs include the personnel and other costs of designing the system.

■ *Physical size and weight:* The physical aspects of the final system can vary greatly depending upon the application. An industrial control system for an assembly line may be designed to fit into a standard-size rack with no strict limitations on weight. A handheld device typically has tight requirements on both size and weight that can ripple through the entire system design.

■ *Power consumption:* Power, of course, is important in battery-powered systems and is often important in other applications as well. Power can be specified in the requirements stage in terms of battery life—the customer is unlikely to be able to describe the allowable wattage.

A sample ***requirements form*** that can be filled out at the start of the project. We can use the form as a checklist in considering the basic characteristics of the system. Let's consider the entries in the form:

■ *Name:* This is simple but helpful. Giving a name to the project not only simplifies talking about it to other people but can also crystallize the purpose of the machine.

■ *Purpose:* This should be a brief one- or two-line description of what the system is supposed to do. If you can't describe the essence of your system in one or two lines, chances are that you don't understand it well enough.

■ *Inputs and outputs:* These two entries are more complex than they seem. The inputs and outputs to the system encompass a wealth of detail:

— *Types of data:* Analog electronic signals? Digital data? Mechanical inputs?

— *Data characteristics:* Periodically arriving data, such as digital audio
samples? Occasional user inputs? How many bits per data element?

— *Types of I/O devices:* Buttons? Analog/digital converters? Video displays?

■ *Functions:* This is a more detailed description of what the system does. A good way to approach this is to work from the inputs to the outputs: When the system receives an input, what does it do? How do user interface inputs affect these functions? How do different functions interact?

Performance: Many embedded computing systems spend at least some time controlling physical devices or processing data coming from the physical world. In most of these cases, the computations must be performed within a certain time frame. It is essential that the performance requirements be identified early
since they must be carefully measured during implementation to ensure that the system works properly.

■ *Manufacturing cost:* This includes primarily the cost of the hardware components. Even if you don't know exactly how much you can afford to spend on system components, you should have some idea of the eventual cost range. Cost has a substantial influence on architecture: A machine that is meant to
sell at $10 most likely has a very different internal structure than a $100 system.

■ *Power:* Similarly, you may have only a rough idea of how much power the system can consume, but a little information can go a long way. Typically, the most important decision is whether the machine will be battery powered or plugged into the wall. Battery-powered machines must be much more careful about how they spend energy.

■ *Physical size and weight:* You should give some indication of the physical size of the system to help guide certain architectural decisions. A desktop machine has much more flexibility in the components used than, for example, a lapel mounted voice recorder.

GPS MODULE:

REQUIREMENTS FORM OF GPS MOVING MAP MODULE:

Name : GPS moving map
Purpose: Consumer-grade moving map for driving use
Inputs : Power button, two control buttons
Outputs : Back-lit LCD display 400 _ 600
Functions : Uses 5-receiver GPS system; three user-selectable resolutions; always displays current latitude and longitude
Performance: Updates screen within 0.25 seconds upon movement
Manufacturing cost:$30
Power: 100mW
Physical size and weight: No more than 2" _ 6, " 12 ounces

Specification

The specification is more precise—it serves as the contract between the customer and the architects. As such, the specification must be carefully written so that it accurately reflects the customer's requirements and does so in a way that can be clearly followed during design.
The specification should be understandable enough so that someone can verify that it meets system requirements and overall expectations of the customer.
A specification of the GPS system would include several components:

- Data received from the GPS satellite constellation.
- Map data.
- User interface.
- Operations that must be performed to satisfy customer requests.
- Background actions required to keep the system running, such as operating the GPS receiver.

Architecture Design

The specification does not say how the system does things, only what the system does. Describing how the system implements those functions is the purpose of the architecture. The architecture is a plan for the overall structure of the system that will be used later to design the components that make up the architecture. The creation of the architecture is the first phase of what many designers think of as design.

This block diagram is still quite abstract—we have not yet specified which operations will be performed by software running on a CPU, what will be done by special-purpose hardware, and so on. The diagram does, however, go a long way toward describing how to implement the functions described in the specification. We clearly see, for example, that we need to search the topographic database and to render (i.e., draw) the results for the display. We have chosen to separate those functions so that we can potentially do them in parallel—performing rendering separately from searching the database may help us update the screen more fluidly.

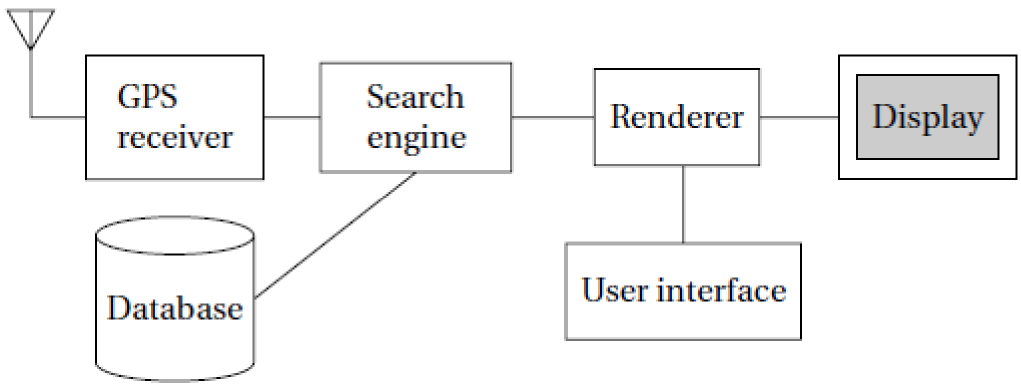


FIG: BLOCK DIAGRAM FOR THE MOVING MAP

The hardware block diagram clearly shows that we have one central CPU surrounded by memory and I/O devices. In particular, we have chosen to use two memories: a frame buffer for the pixels to be displayed and a separate program/data memory for general use by the CPU. The software block diagram fairly closely follows the system block diagram, but we have added a timer to control when we read the buttons on the user interface and render data onto the screen. To have a truly complete architectural description, we require more detail, such as where units in the software block diagram will be executed in the hardware block diagram and when operations will be performed in time.

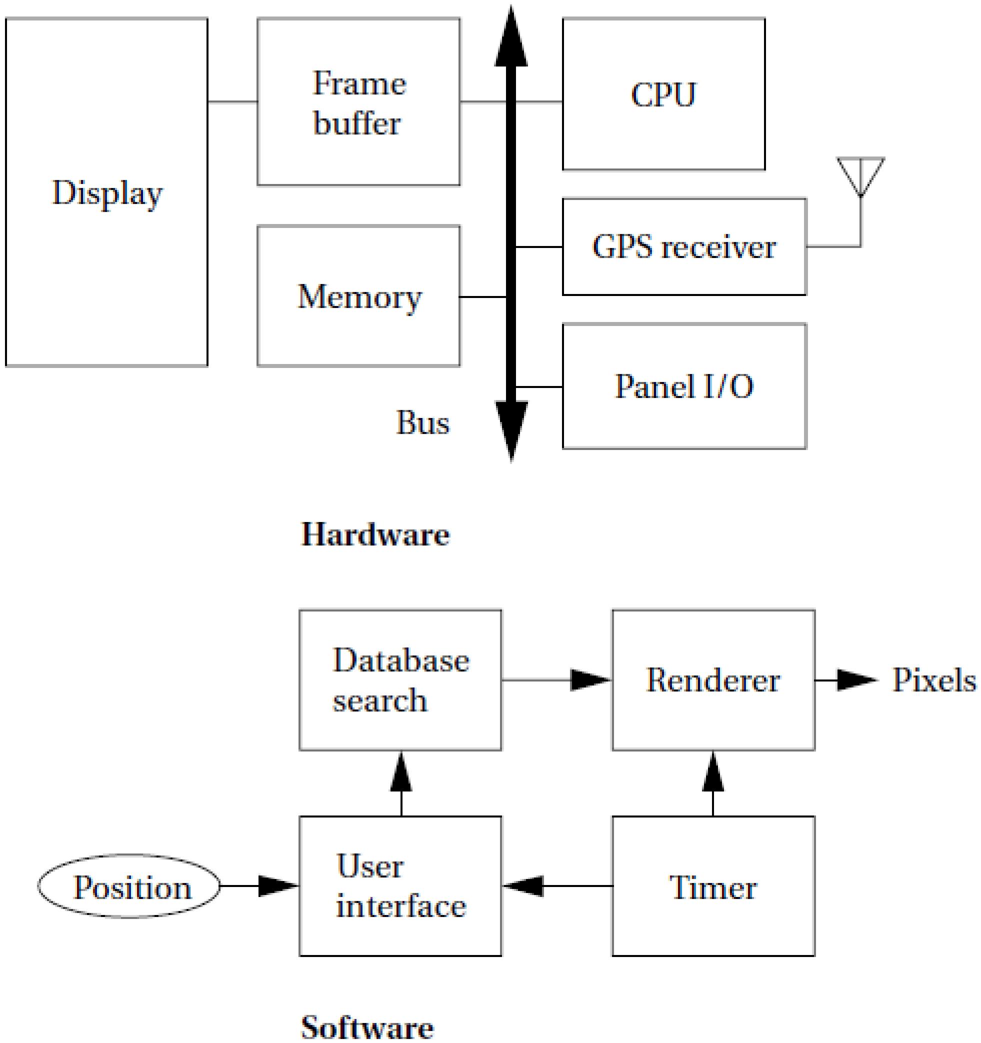


Fig : Hardware and software architectures for the moving map.

The architectural description tells us what components we need. The component design effort builds those components in conformance to the architecture and specification. The components will in general include both hardware—FPGAs, boards, and so on—and software modules. Some of the components will be ready-made. The CPU, for example, will be a standard component in almost all cases, as will memory chips and many other components .In the moving map, the GPS receiver is a good example of a specialized component that will nonetheless be a predesigned, standard component. We can also make use of standard software modules.

System Integration:

Only after the components are built do we have the satisfaction of putting them together and seeing a working system. Of course, this phase usually consists of a lot more than just plugging everything together and standing back. Bugs are typically found during system integration, and good planning can help us find the bugs quickly. By building up the system in phases and running properly chosen tests, we can often find bugs more easily. If we debug only a few modules at a time, we are more likely to uncover the simple bugs and able to easily recognize them. Only by fixing the simple bugs early will we be able to uncover the more complex or obscure bugs that can be identified only by giving the system a hard workout

Characteristics of Embedded systems: (DEC2016, March-2017)

Embedded systems possess certain specific characteristics and these are unique to each Embedded system.

1. Application and domain specific
2. Reactive and Real Time
3. Operates in harsh environments
4. Distributed
5. Small Size and weight
6. Power concerns
7. Single-functioned
8. Complex functionality
9. Tightly-constrained
10. Safety-critical

1. Application and Domain Specific:-

- Each E.S has certain functions to perform and they are developed in such a manner to do the intended functions only.
- They cannot be used for any other purpose.
- Ex – The embedded control units of the microwave oven cannot be replaced with AC‟S embedded control unit because the embedded control units of microwave oven and AC are specifically designed to perform certain specific tasks.

2. Reactive and Real Time:-

- E.S are in constant interaction with the real world through sensors and user-defined input devices which are connected to the input port of the system.
- Any changes in the real world are captured by the sensors or input devices in real time and the control algorithm running inside the unit reacts in a designed manner to bring the controlled output variables to the desired level.
- E.S produce changes in output in response to the changes in the input, so they are referred as reactive systems.
- Real Time system operation means the timing behavior of the system should be deterministic ie the system should respond to requests in a known amount of time.
- Example – E.S which are mission critical like flight control systems, Antilock Brake Systems (**ABS**) etc are Real Time systems.

3. Operates in Harsh Environment :–

- The design of E.S should take care of the operating conditions of the area where the system is going to implement.
- Ex – If the system needs to be deployed in a high temperature zone, then all the components used in the system should be of high temperature grade.
- Also proper shock absorption techniques should be provided to systems which are going to be commissioned in places subject to high shock.

4. Distributed: –

- It means that embedded systems may be a part of a larger system.
- Many numbers of such distributed embedded systems form a single large embedded control unit.
- Ex – Automatic vending machine. It contains a card reader, a vending unit etc. Each of them are independent embedded units but they work together to perform the overall vending function.

5. Small Size and Weight:-

- Product aesthetics (size, weight, shape, style, etc) is an important factor in choosing a product.
- It is convenient to handle a compact device than a bulky product.

6. **Power Concerns:-**

- Power management is another important factor that needs to be considered in designing embedded systems.
- E.S should be designed in such a way as to minimize the heat dissipation by the system.

7. **Single-functioned:-** Dedicated to perform a single function

8. **Complex functionality: -** We have to run sophisticated algorithms or multiple algorithms in some applications.

9. **Tightly-constrained:-**

   - Low cost, low power, small, fast, etc

10. **Safety-critical:-**

    - Must not endanger human life and the environment

Quality Attributes of Embedded System: Quality attributes are the non-functional requirements that need to be documented properly in any system design. **(DEC16, March-2017)**

Quality attributes can be classified as

I. Operational quality attributes

II. Non-operational quality attributes.

I. Operational Quality Attributes: The operational quality attributes represent the relevant quality attributes related to the embedded system when it is in the operational mode or online mode.

Operational Quality Attributes are:

1. **Response :-**

   - It is the measure of quickness of the system.

   It tells how fast the system is tracking the changes in input variables.

   Most of the E.S demands fast response which should be almost real time.

Ex – Flight control application.

2. Throughput :-

It deals with the efficiency of a system.

It can be defined as the rate of production or operation of a defined process over a stated period of time.

The rates can be expressed in terms of products, batches produced or any other meaningful measurements.

- Ex – In case of card reader throughput means how many transactions the reader can perform in a minute or in an hour or in a day.
- Throughput is generally measured in terms of "Benchmark".
- A Benchmark is a reference point by which something can be measured

3. Reliability :-

- It is a measure of how much we can rely upon the proper functioning of the system.
- Mean Time Between Failure (MTBF) and Mean Time To Repair (MTTR) are the terms used in determining system reliability.
- MTBF gives the frequency of failures in hours/weeks/months.
- MTTR specifies how long the system is allowed to be out of order following a failure.
- For embedded system with critical application need, it should be of the order of minutes.

4. Maintainability:-

- It deals with support and maintenance to the end user or client in case of technical issues and product failure or on the basis of a routine system checkup.
- Reliability and maintainability are complementary to each other.
- A more reliable system means a system with less corrective maintainability requirements and vice versa.
- Maintainability can be broadly classified into two categories

1. Scheduled or Periodic maintenance (Preventive maintenance)

2. Corrective maintenance to unexpected failures

5. **Security:-**

- Confidentiality, Integrity and availability are the three major measures of information security.
- Confidentiality deals with protection of data and application from unauthorized disclosure.
- Integrity deals with the protection of data and application from unauthorized modification.
- Availability deals with protection of data and application from unauthorized users.

6. **Safety :-**

- Safety deals with the possible damages that can happen to the operator, public and the environment due to the breakdown of an Embedded System.
- The breakdown of an embedded system may occur due to a hardware failure or a firmware failure.
- Safety analysis is a must in product engineering to evaluate the anticipated damages and determine the best course of action to bring down the consequences of damage to an acceptable level.

II. Non-Operational Quality Attributes: The quality attributes that needs to be addressed for the product not on the basis of operational aspects are grouped under this category.

1. **Testability and Debug-ability:-**

- Testability deals with how easily one can test the design, application and by which means it can be done.
- For an E.S testability is applicable to both the embedded hardware and firmware.
- Embedded hardware testing ensures that the peripherals and total hardware functions in the desired manner, whereas firmware testing ensures that the firmware is functioning in the expected way.
- Debug-ability is a means of debugging the product from unexpected behavior in the system
- Debug-ability is two level process
- 1.Hardware level 2.software level
- **1. Hardware level:** It is used for finding the issues created by hardware problems.
- **2. Software level:** It is employed for finding the errors created by the flaws in the software.

2. Evolvability :-

- It is a term which is closely related to Biology.
- It is referred as the non-heritable variation.
- For an embedded system evolvability refers to the ease with which the embedded product can be modified to take advantage of new firmware or hardware technologies.

3. Portability:-

- It is the measure of system independence.
- An embedded product is said to be portable if the product is capable of functioning in various environments, target processors and embedded operating systems.
- „Porting" represents the migration of embedded firmware written for one target processor to a different target processor.

4. Time-to-Prototype and Market:-

- It is the time elapsed between the conceptualization of a product and the time at which the product is ready for selling.
- The commercial embedded product market is highly competitive and time to market the product is critical factor in the success of commercial embedded product.
- There may be multiple players in embedded industry who develop products of the same category (like mobile phone).

5. Per Unit Cost and Revenue:-

- Cost is a factor which is closely monitored by both end user and product manufacturer.
- Cost is highly sensitive factor for commercial products
- Any failure to position the cost of a commercial product at a nominal rate may lead to the failure of the product in the market.
- Proper market study and cost benefit analysis should be carried out before taking a decision on the per-unit cost of the embedded product.
- The ultimate aim of the product is to generate marginal profit so the budget and total cost should be properly balanced to provide a marginal profit.

SUMMARY

1. An embedded system is an electronic/electromechanical system designed to perform a specific function and is a combination of both hardware and firmware (software).
2. A general purpose computing system is a combination of generic hardware and general purpose operating system for executing a variety of applications, whereas an embedded
3. System is a combination of special purpose hardware and embedded OS/firmware for executing a specific set of applications.
4. Apollo Guidance Computer (AGC) is the first recognized modern embedded system and Autonetics D-17, the guidance computer for the Minuteman-I missile, was the first mass produced embedded system.
5. Based on the complexity and performance requirements, embedded systems are classified into small-scale, medium-scale and large-scale/complex.
6. The presences of embedded system vary from simple electronic system toys to complex flight and missile control systems.
7. Embedded systems are designed to serve the purpose of any one or combination of data collection/storage/representation, data processing, monitoring, control or application specific user interface.
8. Wearable devices refer to embedded systems which are incorporated into accessories and apparels. It envisions the bonding of embedded technology in our day to day lives.

OBJECTIVE QUESTIONS

1. Embedded systems are

   (a) General Purpose (b) Special Purpose
2. Embedded system is

   (a) An electronic system (b) A pure mechanical system

   (c)An electro-mechanical system (d) (a) or (c)
3. Which of the following is not true about embedded systems?

   (a)Built around specialized hardware (b) Always contain an operating system

   (c)Execution behavior may be deterministic (d) All of these (e) none of these
4. Which of the following is not an example of small scale embedded system?

   (a) Electronic Barbie doll (b) Simple calculator

   (c) Cell Phone (d) Electronic toy car

5. The first recognized modern embedded system is

(a) Apple computer (b) Apollo Guidance Computer

(c) Calculator (d) Radio navigation system

6. The first mass produced embedded system is

(a) Minuteman-I (b) Minuteman-II

(c) Autonetics D17 (d) Apollo Guidance Computer

7. Which of the following is (are) an intended purpose of embedded systems?

(a) Data collection (b) Data processing

(c) Data communication (d) All of these (e) None of these

8. Which of the following is an example of an embedded system for data communication?

(a) USB mass storage device (b) Network router (c) Digital camera

(d)Music player (e) All of these (f) None of these

9. A digital multimeter is an example of embedded system for

(a) Data communication (b) Monitoring (c) Control

(d) All of these (e) None of these

10. Which of the following is an example of an embedded system for signal processing?

(a) Apple iPOD (b) Sandisk USB mass storage device

(c) both a and b (d) None of these

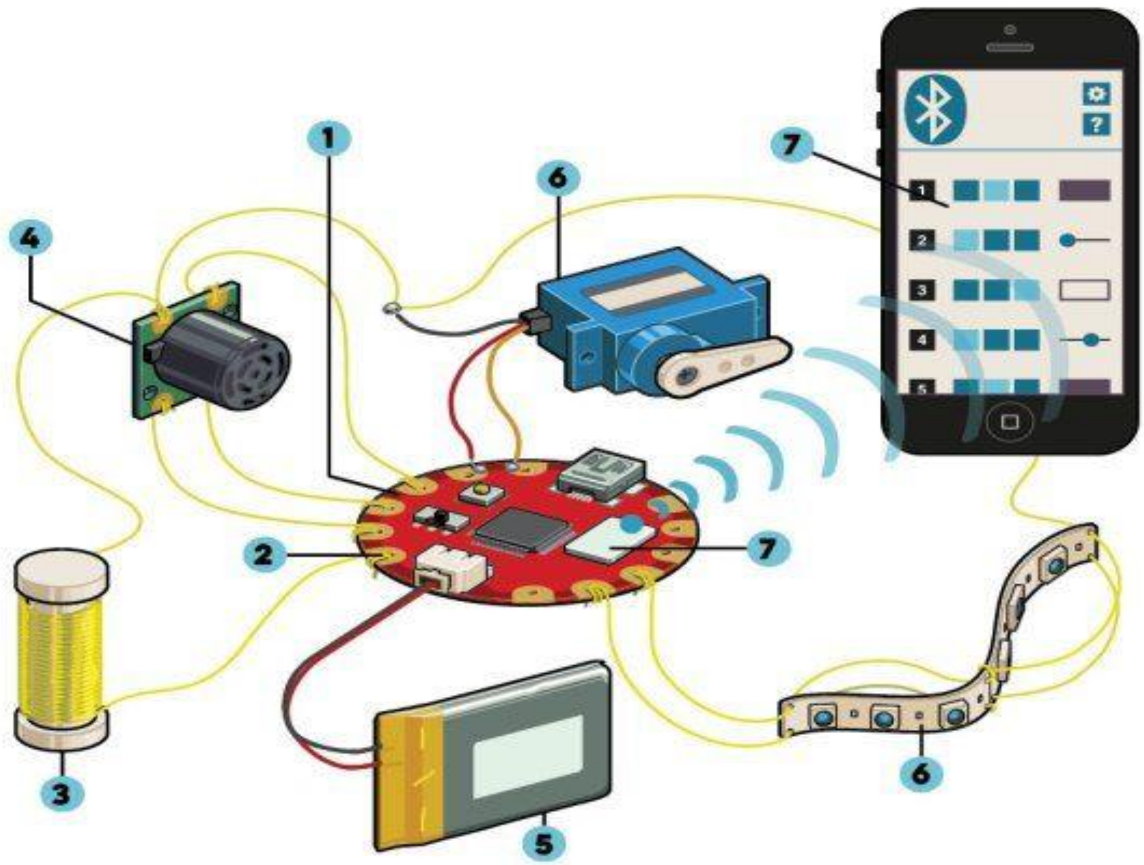
Reference Text Books:-

1. Introduction to Embedded Systems – Shibu K.V Mc Graw Hill

2. Computers as Components –Wayne Wolf-morgan Kaufmann publications

IV ECE

EMBEDDED SYSTEM DESIGN
UNIT-II
TYPICAL EMBEDDED SYSTEM



T.VINAYSIMHA REDDY
DEPT OF ECE
MRCET

ELEMENTS OF EMBEDDED SYSTEMS:

As defined earlier, an embedded system is a combination of 3 things, Hardware Software Mechanical Components and it is supposed to do one specific task only. Diagrammatically an embedded system can be represented as follows:

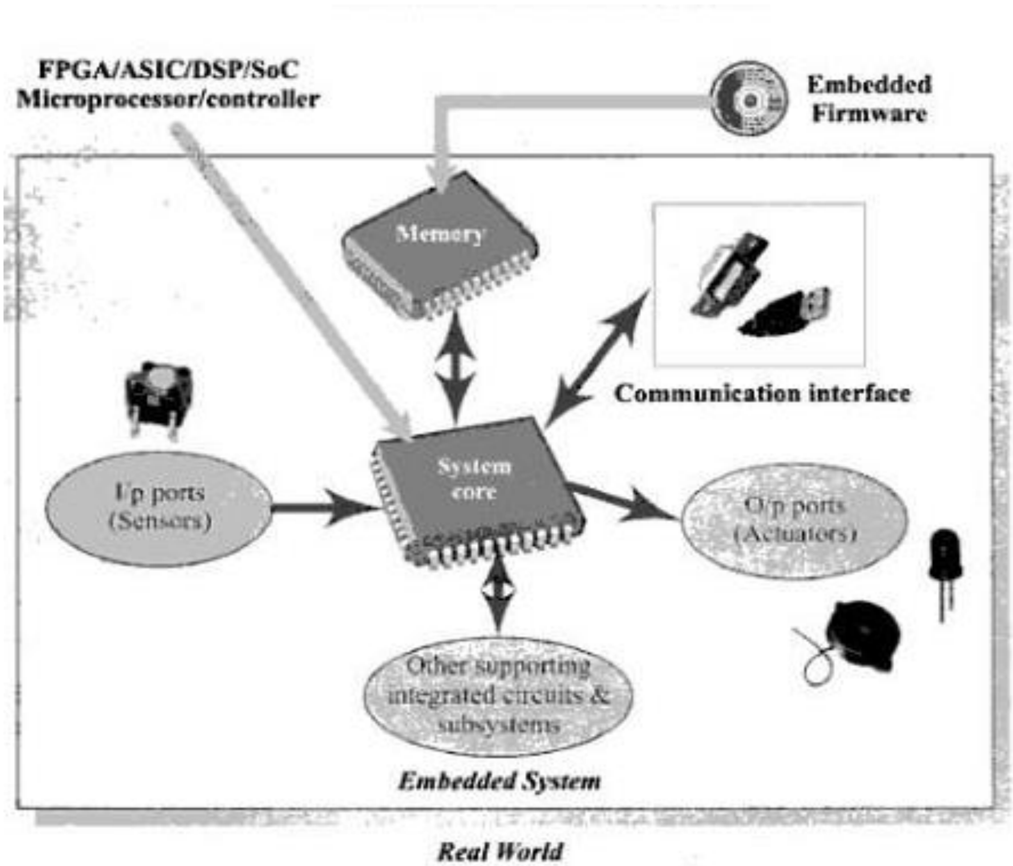


Fig: Elements of an Embedded System

Embedded systems are basically designed to regulate a physical variable (such 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.

Examples of common user interface input devices are keyboards, push button, switches, etc.

The memory of the system is responsible for holding the code (control algorithm and other important configuration details).An embedded system without code (i.e. the control algorithm) implemented memory has all the peripherals but is not capable of making decisions depending on the situational as well as real world changes. 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 embedded system, the controller may contain internal memory for storing code Such controllers are called Micro-controllers with on-chip ROM, eg. Atmel AT89C51.

CORE OF EMBEDDED SYSTEM:

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

- General purpose and Domain Specific Processors
- Microprocessors
- Microcontrollers
- Digital Signal Processors
- Application Specific Integrated Circuits. (ASIC)
- Programmable logic devices(PLD's)
- Commercial off-the-shelf components (COTs)

GENERAL PURPOSE AND DOMAIN SPECIFIC PROCESSOR:

- 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.

(a) MICROPROCESSOR:

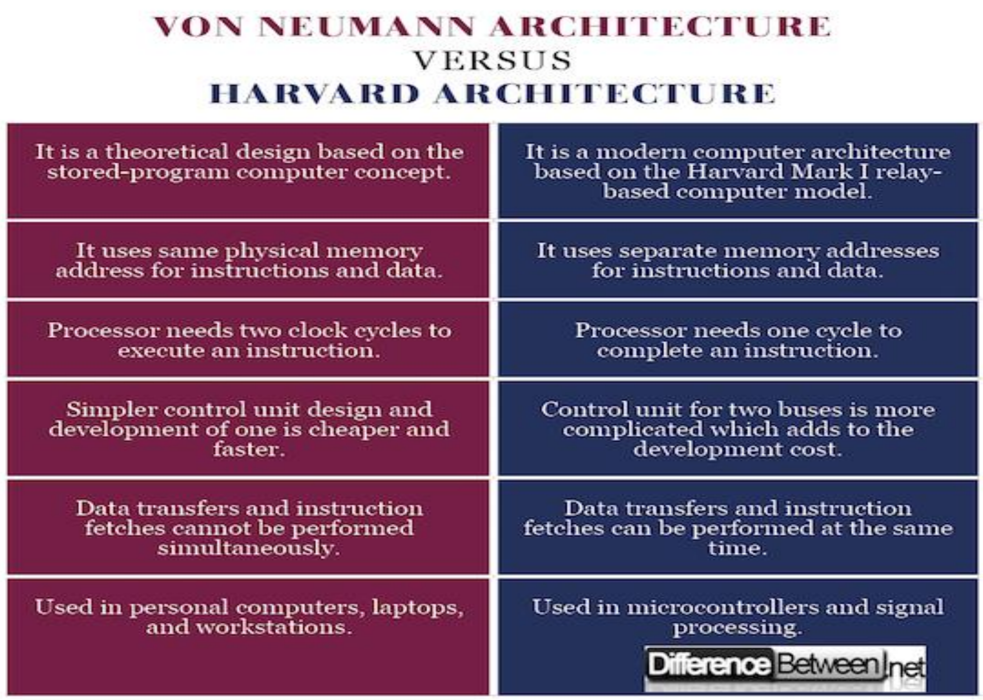
A **microprocessor** is a computer processor that incorporates the functions of a central processing unit on a single integrated circuit (IC),[1] or at most a few integrated circuits.[2]The microprocessor is a multipurpose, clock driven, register based, digital-integrated circuit that accepts binary data as input, processes it according to instructions stored in its memory, and provides results as output.

· Developers of microprocessors.

- Intel – Intel 4004 – November 1971(4-bit)
- Intel – Intel 4040.
- Intel – Intel 8008 – April 1972.
- Intel – Intel 8080 – April 1974(8-bit).
- Motorola – Motorola 6800.
- Intel – Intel 8085 – 1976.

  Zilog - Z80 – July 1976.

Architectures used for processor design are Harvard or Von-Neumann.


VON NEUMANN ARCHITECTURE VERSUS HARVARD ARCHITECTURE

It is a theoretical design based on the stored-program computer concept.	It is a modern computer architecture based on the Harvard Mark I relay-based computer model.
It uses same physical memory address for instructions and data.	It uses separate memory addresses for instructions and data.
Processor needs two clock cycles to execute an instruction.	Processor needs one cycle to complete an instruction.
Simpler control unit design and development of one is cheaper and faster.	Control unit for two buses is more complicated which adds to the development cost.
Data transfers and instruction fetches cannot be performed simultaneously.	Data transfers and instruction fetches can be performed at the same time.
Used in personal computers, laptops, and workstations.	Used in microcontrollers and signal processing.

DifferenceBetween.net

RISC and CISC are the two common Instruction Set Architectures (ISA) available for processor design.

CISC	RISC
1) CISC architecture gives more importance to hardware	1) RISC architecture gives more importance to Software
2) Complex instructions.	2) Reduced instructions.
3) It access memory directly	3) It requires registers.
4) Coding in CISC processor is simple.	4) Coding in RISC processor requires more number of lines.
5) As it consists of complex instructions, it take multiple cycles to execute.	5) It consists of simple instructions that take single cycle to execute.
6) Complexity lies in microporgram	6) Complexity lies in compiler.

Endiannes:

Endianness specifies the order which the data is stored in the memory by processor operations in a multi byte system.

Based on Endiannes processors can be of two types:

- Little Endian Processors
- Big Endian Processors

Little Endian Processors:

Little-endian means lower order data byte is stored in memory at the lowest address and the higher order data byte at the highest address. For e.g, 4 byte long integer Byte3, Byte2, Byte1, Byte0 will be store in the memory as follows:

Base address+0	Byte 0	Byte 0	(Base address)
Base address+1	Byte 1	Byte 1	(Base address+1)
Base address+2	Byte 2	Byte 2	(Base address+2)
Base address+3	Byte 3	Byte 3	(Base address+3)

Big Endian Processors:

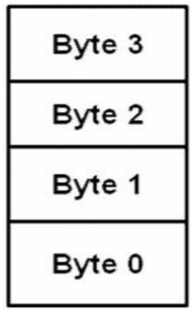
Big-endian means the higher order data byte is stored in memory at the lowest and the lower order data byte at the highest address. For e.g. a 4 byte integer Byte3, Byte2, Byte1, Byte0 will be stored in the memory as follows:

Base address+0 Byte 3 (Base address)

Base address+1 Byte 2 (Base address+1)

Base address+2 Byte 1 (Base address+2)

Base address+3 Byte 0 (Base address+3)



(b).MICROCONTROLLER:

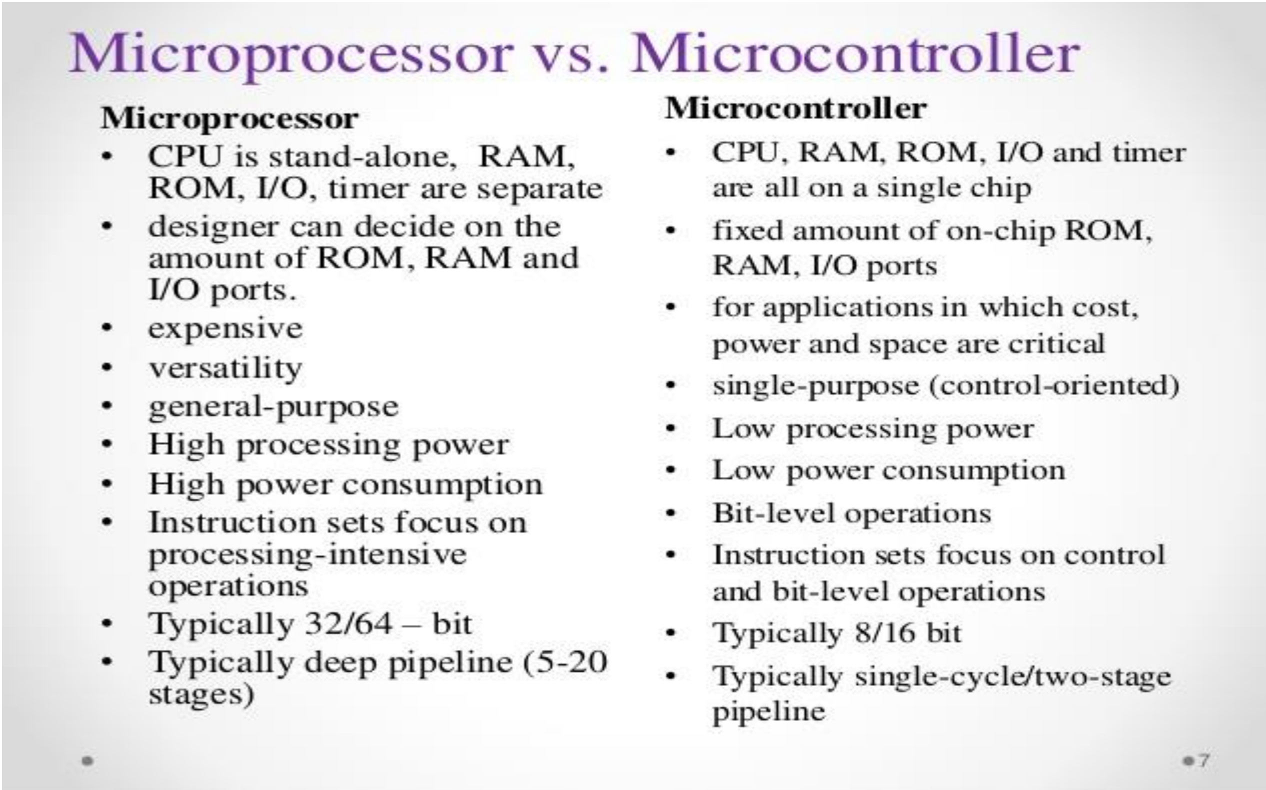
A microcontroller is a highly integrated chip that contains a CPU, scratch pad RAM, special and general purpose register arrays, on chip ROM/FLASH memory for program storage , timer and interrupt control units and dedicated I/O ports.

Texas Instrument's TMS 1000 Is considered as the world's first microcontroller.

Some embedded system application require only 8 bit controllers whereas some requiring superior performance and computational needs demand 16/32 bit controllers.

The instruction set of a microcontroller can be RISC or CISC.

Microcontrollers are designed for either general purpose application requirement or domain specific application requirement.



Digital Signal Processors:

- DSP are powerful special purpose 8/16/32 bit microprocessor designed to meet the computational demands and power constraints of today's embedded audio, video and communication applications.

- DSP are 2 to 3 times faster than general purpose microprocessors in signal processing applications. This is because of the architectural difference between DSP and general purpose microprocessors.
- DSPs implement algorithms in hardware which speeds up the execution Whereas general purpose processor implement the algorithm in software and the speed of execution depends primarily on the clock for the Processors.

DSP includes following key units:

Program memory: It is a memory for storing the program required by DSP to process the data.

Data memory: It is a working memory for storing temporary variables and data/signal to be processed.

Computational engine: It performs the signal processing in accordance with the stored program memory computational engine incorporated many specialized arithmetic units and each of them operates simultaneously to increase the execution speed. It also includes multiple hardware shifters for shifting operands and saves execution time.

I/O unit: It acts as an interface between the outside world and DSP. It is responsible for capturing signals to be processed and delivering the processed signals.

Examples: Audio video signal processing, telecommunication and multimedia applications.

SOP(Sum of Products) calculation, convolution, FFT(Fast Fourier Transform), DFT(Discrete Fourier Transform), etc are some of the operation performed by DSP.

Application Specific Integrated Circuits(ASIC):

1.ASICs is a microchip design to perform a specific and unique applications.

2.Because of using single chip for integrates several functions there by reduces the system development cost.

3.Most of the ASICs are proprietary (which having some trade name) products, it is referred as Application Specific Standard Products(ASSP).

4.As a single chip ASIC consumes a very small area in the Total system.
5.Thereby helps in the design of smaller system with high capabilities or functionalities.
6.The developers of such chips may not be interested in revealing the internal detail of it .

Programmable logic devices(PLD's):

- A PLD is an electronic component. It used to build digital circuits which are reconfigurable.
- A logic gate has a fixed function but a PLD does not have a defined function at the time of manufacture.
- PLDs offer customers a wide range of logic capacity, features, speed, voltage characteristics.
- PLDs can be reconfigured to perform any number of functions at any time.

- A variety of tools are available for the designers of PLDs which are inexpensive and help to develop, simulate and test the designs.

PLDs having following two major types.

1) CPLD(Complex Programmable Logic Device):
CPLDs offer much smaller amount of logic up to 1000 gates.

2) FPGAs(Field Programmable Gate Arrays):
It offers highest amount of performance as well as highest logic density, the most features.

Advantages of PLDs :-

- PLDs offer customer much more flexibility during the design cycle.
- PLDs do not require long lead times for prototypes or production parts because PLDs are already on a distributors shelf and ready for shipment.
- PLDs can be reprogrammed even after a piece of equipment is shipped to a customer

Commercial off-the-shelf components(COTs):

A Commercial off the Shelf product is one which is used 'as-is'.

The COTS components itself may be develop around a general purpose or domain specific processor or an ASICs or a PLDs.

The major advantage of using COTS is that they are readily available in the market, are chip and a developer can cut down his/her development time to a great extent

The major drawback of using COTS components in embedded design is that the manufacturer of the COTS component may withdraw the product or discontinue the production of the COTS at any time if rapid change in technology occurs.

Advantages of COTS:

- ❖ Ready to use
- ❖ Easy to integrate
- ❖ Reduces development time

Disadvantages of COTS:

- No operational or manufacturing standard (all proprietary)
- Vendor or manufacturer may discontinue production of a particular COTS product
- The developers of such chips may not be interested in revealing the internal detail of it .

SENSORS & ACTUATORS:

Sensor: A sensor is an object whose purpose is to detect events or changes in its environment, and then provide a corresponding output. A sensor is a type of transducer; sensors may provide various types of output, but typically use electrical or optical signals. For example, a thermocouple generates a known voltage (the output) in response to its temperature (the environment).

- A Sensor is used for taking Input
- It is a transducer that converts energy from one form to another for any measurement or control purpose

Ex. A Temperature sensor

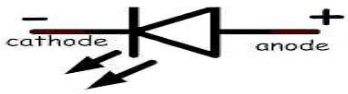
Actuator: An actuator is a component of machines that is responsible for moving or controlling a mechanism or system. An actuator requires a control signal and source of energy. The control signal is relatively low energy and may be electric voltage or current, pneumatic or hydraulic pressure, or even human power

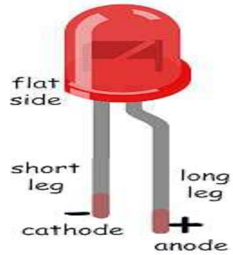
Actuator is used for output. It is a transducer that may be either mechanical or electrical which converts signals to corresponding physical actions.

I/O subsystem: The I/O sub system of the embedded system facilitates the interaction of embedded system with the external world

LED (Light Emitting Diode): LED is an important output device for visual indication in any embedded system.LED used as an indicator for the status of various signals or situations

LED is a p-n junction diode and contains a **CATHODE** and **ANODE.** For functioning the anode is connected to +ve end of power supply and cathode is connected to –ve end of power supply. The maximum current flowing through the LED is limited by connecting a RESISTOR in series between the power supply and LED as shown in the figure below





There are two ways to interface an LED to a microprocessor/microcontroller:

The Anode of LED is connected to the port pin and cathode to Ground : In this approach the port pin sources the current to the LED when it is at logic high(ie. 1).

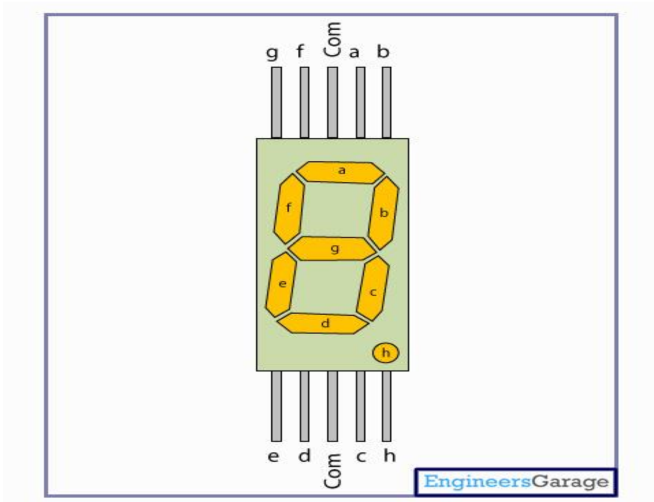
The Cathode of LED is connected to the port pin and Anode to Vcc : In this approach the port pin sources the current to the LED when it is at logic high (ie. 1). Here the port pin sinks the current and the LED is turned ON when the port pin is at Logic low (ie. 0)

7- Segment LED display:

- **It** is the most basic electronic display device that can display digits from 0-9.
- They find wide application in devices that display numeric information like digital clocks, radio, microwave ovens, electronic meters etc.
- The most common configuration has an array of eight LEDs arranged in a special pattern to display these digits.
- They are laid out as a squared-off figure '8'. Every LED is assigned a name from 'a' to 'h' and is identified by its name.
- Seven LEDs 'a' to 'g' are used to display the numerals while eighth LED 'h' is used to display the dot/decimal.

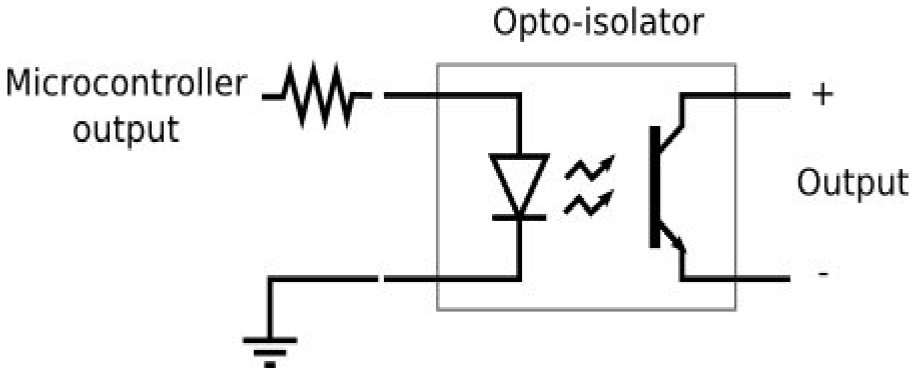
A seven segment is generally available in ten pin package. While eight pins correspond to the eight LEDs, the remaining two pins (at middle) are common and internally shorted. These segments come in two configurations, namely, Common cathode (CC) and Common anode (CA). In CC configuration, the negative terminals of all LEDs are connected to the common pins. The common is connected to ground and a particular LED glows when its corresponding pin is given high. In CA arrangement, the common pin is given a high logic and the LED pins are given low to display a number. Find out more information about a seven segment display and its working.

Pin Diagram:

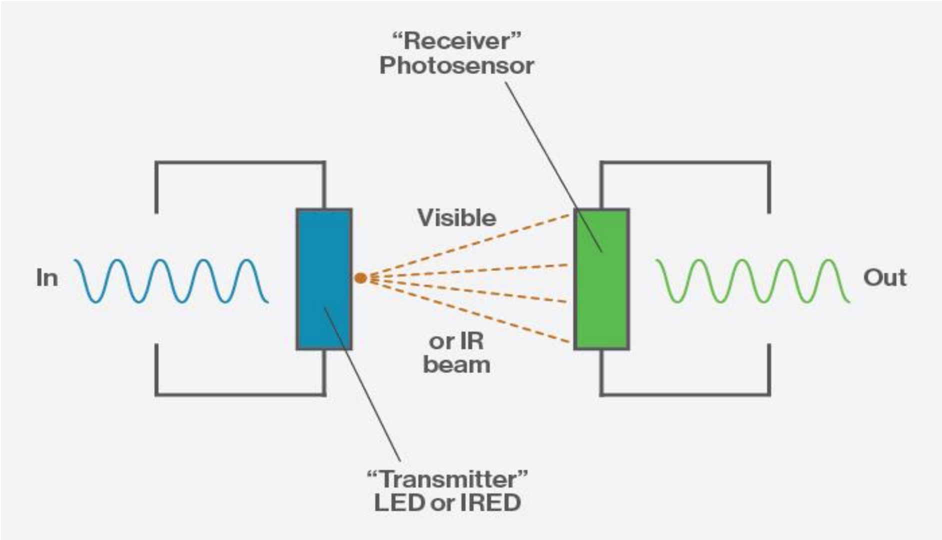


Optocoupler: Optocoupler is , also called photocoupler, or optical isolator, is a component that transfers electrical signals between two isolated circuits by using light. Optocoupler prevent high voltages from affecting the system receiving the signal. Commercially available opto-isolators withstand input-to-output voltages up to 10 kV and voltage transients with speeds up to 10 kV/μs.

An optocoupler consists of an LED and a phototransistor in the same opaque package.

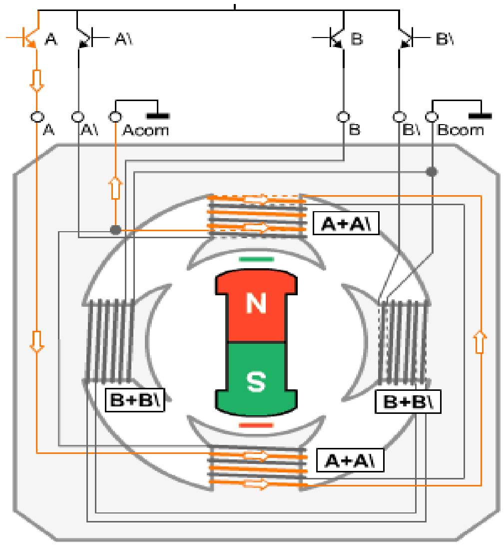


when an electrical current is applied to the LED, infrared light is produced and passes through the material inside the optoisolator. The beam travels across a transparent gap and is picked up by the receiver, which converts the modulated light or IR back into an electrical signal. In the absence of light, the input and output circuits are electrically isolated from each other.



Stepper motor :

Introduction: Stepper motors are special versions of the synchronous machine, in which the rotor is a permanent magnet, while the stator consists of a coil package. In contrast to synchronous motors, stepper motors have a large number of pole pairs. Motor operation requires a control unit, which energizes the individual motor windings based on a certain pulse sequence. Safe positioning is therefore only guaranteed within the performance limits. If the motor is operated within its load limits, positioning without feedback of the rotor position can be achieved by linking individual steps. This operating mode (open loop control) and the durability of the stepper motor enable it to be used as a positioning drive in price-sensitive applications.



Basic function principles of a stepper motor Like most electric motors, a stepper motor consists of a stator (fixed external winding) and a rotor (rotating shaft with magnets). The rotation of the motor shaft (rotor) is generated by rapid energizing the electromagnetic field of the stator, causing the shaft to turn by the step angle a. In a minimum control configuration, the stepper motor is moved from pole to pole, or from step to step. A full turn of the motor shaft is therefore made up of individual steps. Energizing of the motor windings results in a magnetic field in the motor from north to south (or south to north if the power supply has negative polarity and the winding is arranged accordingly). The movable stator with its permanent magnets aligns itself according to the direction of the external magnetic field of the stator.

Advantages of stepper motors compared with other motors:

In contrast to other motors stepper motors have a high holding torque even at low speed, and even on standstill. Another advantage is the simple control of stepper motors: Alternate energizing of the individual coils causes the motor to move step by step. The fixed number of steps per revolution always enables the current position to be determined if the steps are counted and the motor is operated within its performance limits. No encoder is required for simple positioning tasks within the performance limits. Stepper motors are therefore ideally suited as cost-effective solutions for simple positioning tasks.

Relays: A relay is a simple electromechanical switch made up of an electromagnet and a set of contacts. Relays are found hidden in all sorts of devices.

OTHER SYSTEM COMPONENTS:

Reset Circuit:

- The Reset circuit is essential to ensure that the device is not operating at a voltage level where the device is not guaranteed to operate, during system power ON.
- The Reset signal brings the internal registers and the different hardware systems of the processor/controller to a known state and starts the firmware execution from the

reset vector (Normally from vector address 0x0000 for conventional processors/controllers

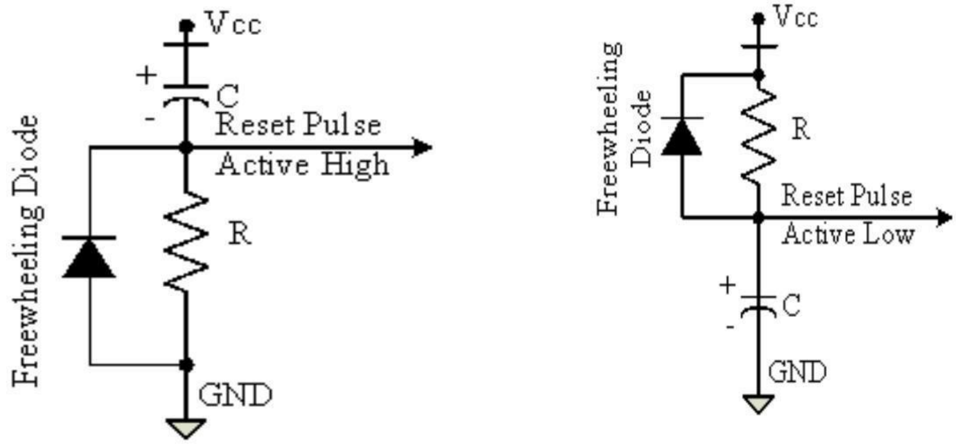


Figure : RC based Reset circuit

- The reset vector can be relocated to an address for processors/controllers supporting bootloader.

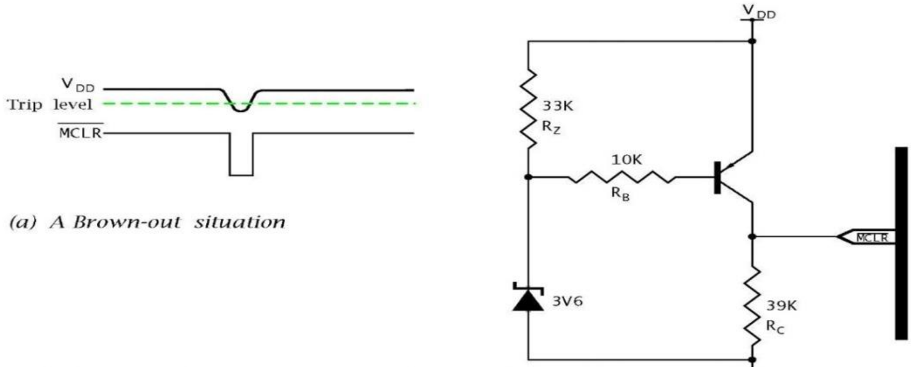
- The reset signal can be either active high (The processor undergoes reset when the reset pin of the processor is at logic high) or active low (The processor undergoes reset when the reset pin of the processor is at logic low).
- The reset signal to the processor can be applied at power ON through an external passive reset circuit comprising a capacitor and resistor or through a standard rest IC like MAX 810 from maxim Dallas.
- Some microprocessors contain inbuilt reset circuitry and they don't need external reset circuitry.

Brown-out Protection Circuit:

- Brown-out protection circuit prevents the processor/controller from unexpected program execution behavior when the supply voltage to the processor/controller falls below a specified voltage

- The processor behavior may not be predictable if the supply voltage falls below the recommended operating voltage. It may lead to situations like data corruption

- A brown-out protection circuit holds the processor/controller in reset state, when the operating voltage falls below the threshold, until it rises above the threshold voltage



(b) An external Brown-out protection circuit

- Certain processors/controllers support built in brown-out protection circuit which monitors the supply voltage internally

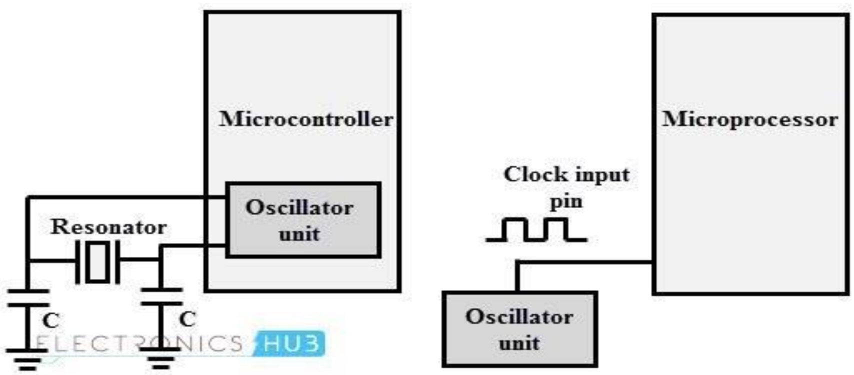
- If the processor/controller doesn't integrate a built-in brown-out protection circuit, the same can be implemented using external passive circuits or supervisor ICs
- The zener diode D and transistor Q forms the heart of the circuit. The transistor conducts always when the supply voltage VCC is greater than that of the sum of VBE and VZ.
- The transistor stops conducting when the supply voltage falls below the sum of VBE and VZ. The values of the resistors can be selected based on the electrical characteristics.

Oscillator Unit:

- A microprocessor/microcontroller is a digital device made up of digital combinational and sequential circuits

- The instruction execution of a microprocessor/controller occurs in sync with a clock signal

- The oscillator unit of the embedded system is responsible for generating the precise clock for the processor



- Certain processor/controller chips may not contain a built-in oscillator unit and require the clock pulses to be generated and supplied externally

  - Quartz crystal Oscillators are example for clock pulse generating devices.
  - The total system power consumption is directly proportional to the clock frequency. The power consumption increase with the increase in the clock frequency .
  - The accuracy of the program execution depends on the accuracy of the clock signal. The accuracy of the clock frequency of the crystal oscillator is normally expressed in terms of parts per million.

Real Time Clock (RTC):

- The system component responsible for keeping track of time. RTC holds information like current time (In hour, minutes and seconds) in 12 hour /24 hour format, date, month, year, day of the week etc and supplies timing reference to the system

- RTC is intended to function even in the absence of power. RTCs are available in the form of Integrated Circuits from different semiconductor manufacturers like Maxim/Dallas, ST Microelectronics etc

- The RTC chip contains a microchip for holding the time and date related information and backup battery cell for functioning in the absence of power, in a single IC package

- The RTC chip is interfaced to the processor or controller of the embedded system

- For Operating System based embedded devices, a timing reference is essential for synchronizing the operations of the OS kernel. The RTC can interrupt the OS kernel by asserting the interrupt line of the processor/controller to which the RTC interrupt line is connected

- The OS kernel identifies the interrupt in terms of the Interrupt Request (IRQ) number generated by an interrupt controller

- One IRQ can be assigned to the RTC interrupt and the kernel can perform necessary operations like system date time updation, managing software timers etc when an RTC timer tick interrupt occurs

Watch Dog Timer (WDT):

- A timer unit for monitoring the firmware execution

- Depending on the internal implementation, the watchdog timer increments or decrements a free running counter with each clock pulse and generates a reset signal to reset the processor if the count reaches zero for a down counting watchdog, or the highest count value for an up counting watchdog

- If the watchdog counter is in the enabled state, the firmware can write a zero (for up counting watchdog implementation) to it before starting the execution of a piece of code (subroutine or portion of code which is susceptible to execution hang up) and the watchdog will start counting. If the firmware execution doesn't complete due to malfunctioning, within the time required by the watchdog to reach the maximum count, the counter will generate a reset pulse and this will reset the processor

- If the firmware execution completes before the expiration of the watchdog timer the WDT can be stopped from action.

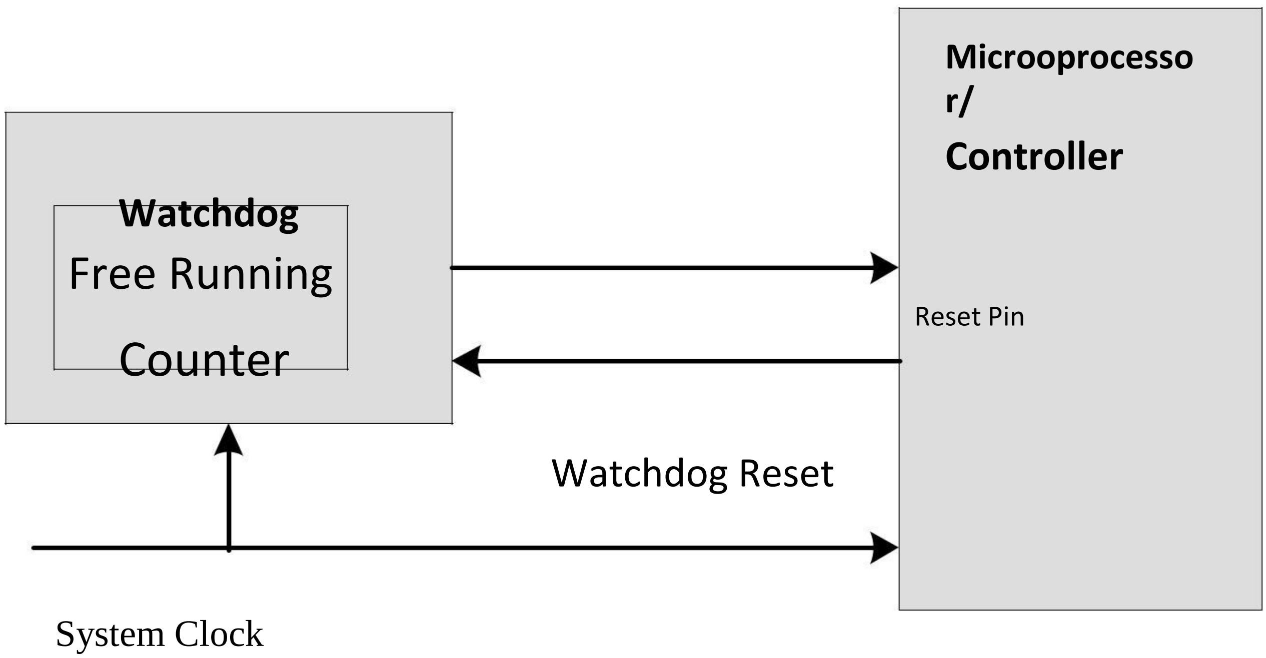


Figure : External Watch Dog Timer Unit Interfacing with Processor

- Most of the processors implement watchdog as a built-in component and provides status register to control the watchdog timer (like enabling and disabling watchdog functioning) and watchdog timer register for writing the count value. If the processor/controller doesn't contain a built in watchdog timer, the same can be implemented using an external watchdog timer IC circuit.

EMBEDDED SYSTEM DESIGN

IV YEAR ECE (R15)

UNIT-III COMMUNICATION INTERFACE

T. Vinaysimha Reddy, Dept of ECE, MRCET
D. Maheswari, Dept of ECE, MRCET

COMMUNICATION INTERFACE:

Communication interface is essential for communicating with various subsystems of the embedded system. For an embedded product, the communication interface can be viewed in two different perspectives namely

1. Device/board level(on board communication interface)
2. Product level communication interface(External Communication Interface)

ONBOARD COMMUNICATION INTERFACE:

Onboard communication interface refers to the different communication channels/buses for connecting the various integrated circuits and other peripherals within the system. Various interfaces are

1. I2C
2. SPI
3. UART
4. 1-WIRE INTERFACE

I2C: INTER INTEGRATED CIRCUIT:

I²C (**Inter-Integrated Circuit**), pronounced *I-squared-C* or *I-two-C*, is a multi-master, multi-slave, packet switched, single-ended, serial computer businvented by Philips Semiconductor (now NXP Semiconductors). It is typically used for attaching lower-speed peripheral ICs to processors and microcontrollers in short-distance, intra- board communication.

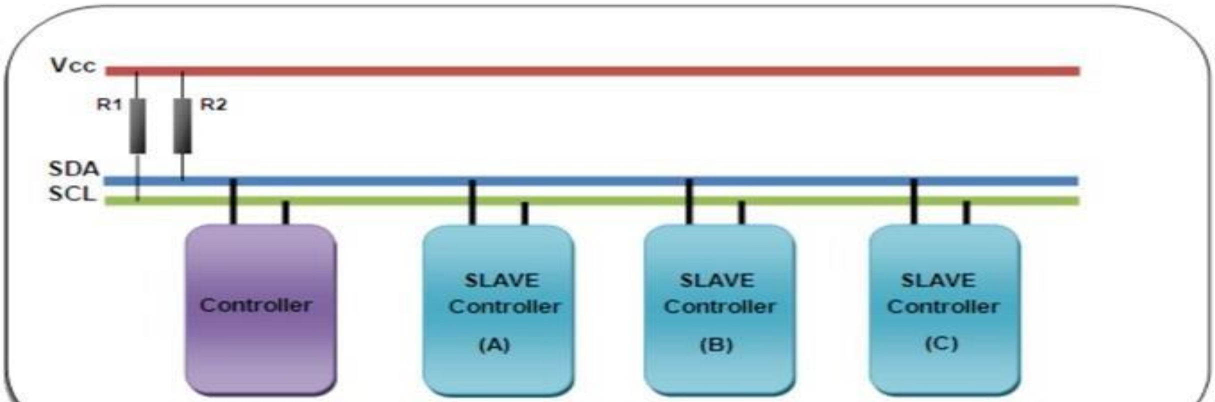
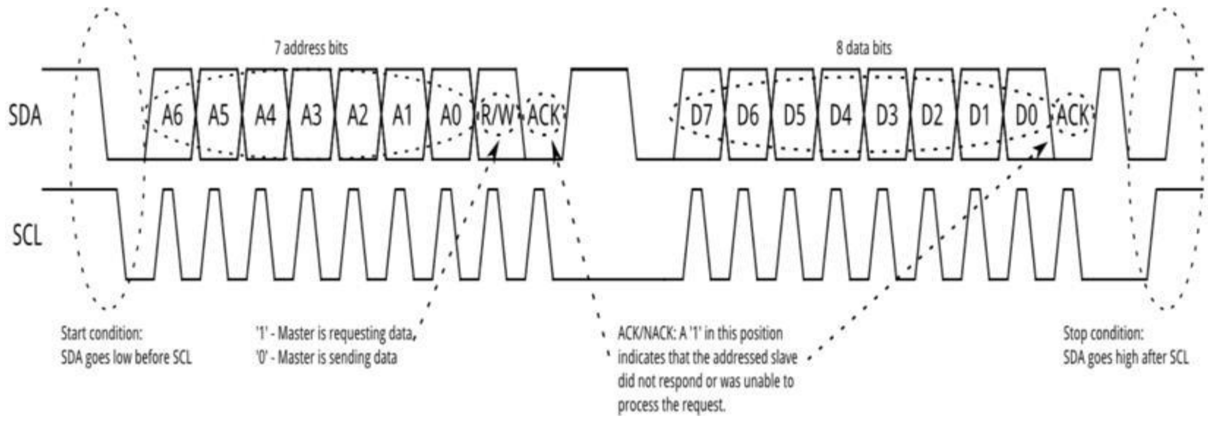


Fig 1: Block diagram

Each I^2C bus consists of two signals: SCL and SDA. SCL is the clock signal, and SDA is the data signal. The clock signal is always generated by the current bus master; some slave devices may force the clock low at times to delay the master sending more data (or to require more time to prepare data before the master attempts to clock it out).This is called “clock stretching” .

Messages are broken up into two types of frame: an address frame, where the master indicates the slave to which the message is being sent, and one or more data frames, which are 8-bit data messages passed from master to slave or vice versa. Data is placed on the SDA line after SCL goes low, and is sampled after the SCL line goes high. The time between clock edge and data read/write is defined by the devices on the bus and will vary from chip to chip.



Messages are broken up into two types of frame: an address frame, where the master indicates the slave to which the message is being sent, and one or more data frames, which are 8-bit data messages passed from master to slave or vice versa. Data is placed on the SDA line after SCL goes low, and is sampled after the SCL line goes high. The time between clock edge and data read/write is defined by the devices on the bus and will vary from chip to chip.

Start Condition:

To initiate the address frame, the master device leaves SCL high and pulls SDA low. This puts all slave devices on notice that a transmission is about to start. If two master devices wish to take ownership of the bus at one time, whichever device pulls SDA low first wins the race and gains control of the bus.

Address Frame:

The address frame is always first in any new communication sequence. For a 7-bit address, the address is clocked out most significant bit (MSB) first, followed by a R/W bit indicating whether this is a read
(1) or write (0) operation.

The 9th bit of the frame is the NACK/ACK bit. This is the case for all frames (data or address). Once the first 8 bits of the frame are sent, the receiving device is given control over SDA. If the receiving device does not pull the SDA line low before the 9th clock pulse, it can be inferred that the receiving device either did not receive the data or did not know how to parse the message.

Data Frames:

After the address frame has been sent, data can begin being transmitted. The master will simply continue generating clock pulses at a regular interval, and the data will be placed on SDA by either the master or the slave, depending on whether the R/W bit indicated a read or write operation.

Stop condition:

Once all the data frames have been sent, the master will generate a stop condition. Stop conditions are defined by a 0->1 (low to high) transition on SCL, with SCL remaining high.

Data rates:

The speed grades (standard mode: 100 kbit/s, full speed: 400 kbit/s, fast mode: 1 mbit/s, high speed: 3,2 Mbit/s) are maximum ratings.

SERIAL PERIPHERAL INTERFACE(SPI)

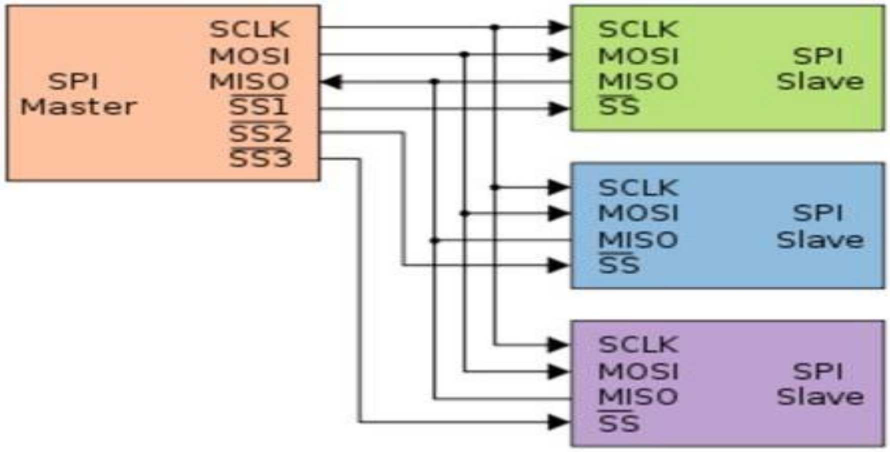
The **Serial Peripheral Interface bus** (**SPI**) is a synchronous serial communication interface specification used for short distance communication, primarily in embedded systems. Interface was developed by Motorola in the late 1980s and has become a *de facto* standard. Typical applications include Secure Digital cards and liquid crystal displays.

SPI devices communicate in full duplex mode using a master-slave architecture with a single master. The master device originates the frame for reading and writing. Multiple slave devices are supported through selection with individual slave select (SS) lines.

Sometimes SPI is called a *four-wire* serial bus.

SPI devices communicate in full duplex mode using a master-slave architecture with a single master. The master device originates the frame for reading and writing. Multiple slave devices are supported through selection with individual slave select (SS) lines.

Sometimes SPI is called a *four-wire* serial bus.



The SPI bus specifies five logic signals:

SCLK: Serial Clock (output from master).

MOSI: Master Output Slave Input, or Master Out Slave In (data output from master).

MISO: Master Input Slave Output, or Master In Slave Out (data output from slave).

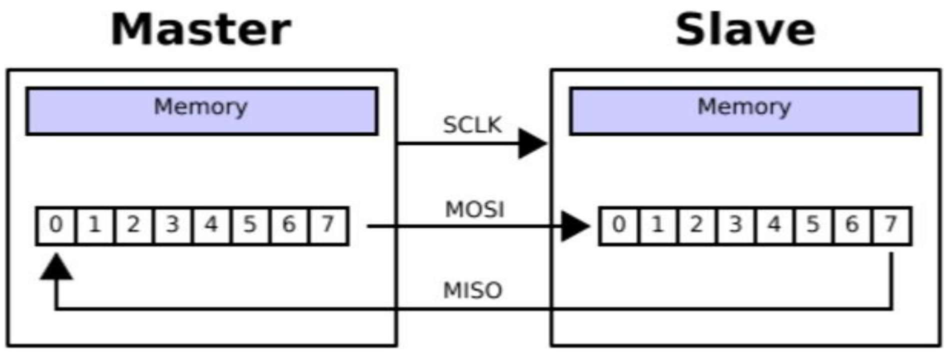
SDIO: Serial Data I/O (bidirectional I/O)

SS: Slave Select (often active low, output from master).

Data transmission: To begin communication, the bus master configures the clock, using a frequency supported by the slave device, typically up to a few Mhz. The master then selects the slave device with a logic level 0 on the select line.

The master sends a bit on the MOSI line and the slave reads it, while the slave sends a bit on the MISO line and the master reads it.

Transmissions normally involve two shift registers of some given word size, such as eight bits, one in the master and one in the slave; they are connected in a virtual ring topology. Data is usually shifted out with the most-significant bit first, while shifting a new least- significant bit into the same register. At the same time, Data from the counterpart is shifted into the least- significant bit register.



I2C V/S SPI:

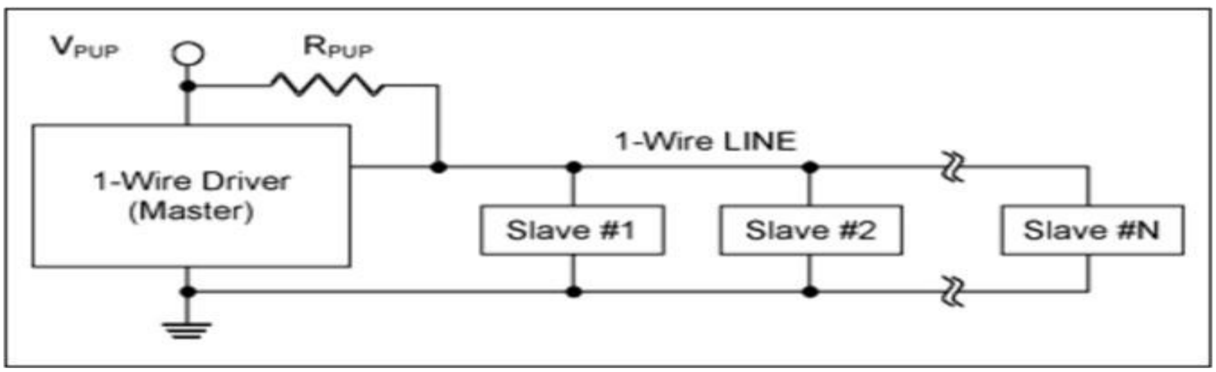
I2C	SPI
Speed limit varies from 100kbps, 400kbps, 1mbps, 3.4mbps depending on i2c version.	More than 1mbps, 10mbps till 100mbps can be achieved.
Half duplex synchronous protocol	Full Duplex synchronous protocol
Support Multi master configuration	Multi master configuration is not possible
Acknowledgement at each transfer	No Acknowledgement
Require Two Pins only SDA, SCL	Require separate MISO, MOSI, CLK & CS signal for each slave.
Addition of new device on the bus is easy	Addition of new device on the bus is not much easy a I2C
More Overhead (due to acknowledgement, start, stop)	Less Overhead
Noise sensitivity is high	Less noise sensitivity

1-wire interface (protocol)

1-Wire is a device communications bus system designed by Dallas Semiconductor Corp. that provides low-speed data, signaling, and power over a single conductor.

1-Wire is similar in concept to I²C, but with lower data rates and longer range. It is typically used to communicate with small inexpensive devices such as digital thermometers and weather instruments.

One distinctive feature of the bus is the possibility of using only two wires: data and ground. To accomplish this, 1-Wire devices include an 800 pF capacitor to store charge, and to power the device during periods when the data line is active.



There is always one master in overall charge, which may be a PC or a microcontroller. The master initiates activity on the bus, simplifying the avoidance of collisions on the bus. Protocols are built into the software to detect collisions. After a collision, the master retries the required communication.

Many devices can share the same bus. Each device on the bus has a unique 64-bit serial number. The least significant byte of the serial number is an 8-bit number that tells the type of the device. The most significant byte is a standard (for the 1-wire bus) 8-bit CRC.

The master starts a transmission with a *reset* pulse, which pulls the wire to 0 volts for at least 480 µs. This resets every slave device on the bus. After that, any slave device, if present, shows that it exists with a "presence" pulse: it holds the bus low for at least 60 µs after the master releases the bus.

To send a "1", the bus master sends a very brief (1– 15 µs) low pulse. To send a "0", the master sends a 60 µs low pulse.

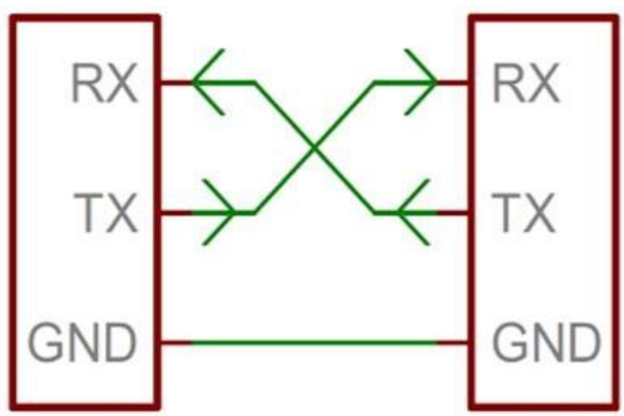
When receiving data, the master start sends a 1–15-μs 0-volt pulse to slave each bit. If the transmitting does unit wants to send a "1", it to the nothing, and the bus goes transmitting pulled-up voltage. If the a "0", it pulls slave wants to send the data line to ground for 60 μs.

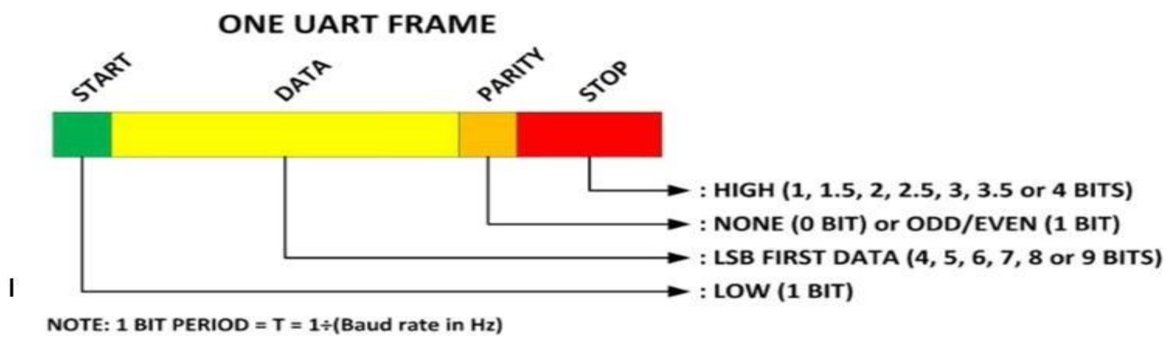
UART:

A universal asynchronous receiver/transmitter (UART)is computer hardware device for asynchronous serial communication which the data format and transmission speeds are configurable.

The universal asynchronous receiver/transmitter (UART) takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART re-assembles the bits into complete bytes. Each UART contains a shift register, which is the fundamental method of conversion between serial and parallel forms.

Communication may be *simplex* (in one direction only, with no provision for the receiving device to send information back to the transmitting device), *full duplex* (both devices send and receive at the same time) or *half duplex* (devices take turns transmitting and receiving).





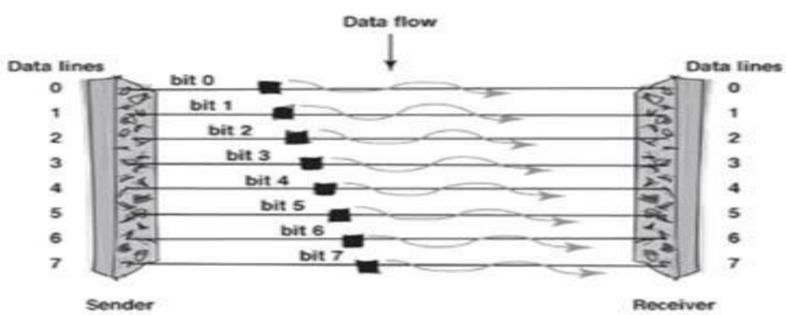
PARALLEL COMMUNICATION:

In data transmission, **parallel communication** is a method of conveying multiple binary digits (bits) simultaneously. It contrasts with serial communication.

The basic difference between a parallel and a serial communication channel is the number of electrical conductors used at the physical layer to convey bits.

Parallel communication implies more than one such conductor. For example, an 8-bit parallel channel will convey eight bits (or a byte) simultaneously, whereas a serial channel would convey those same bits sequentially, one at a time.

Parallel communication is and always has been widely used within integrated circuits, in peripheral buses, and in memory devices such as RAM.



SERIAL VS PARALLEL

	SERIAL	PARALLEL
COST	Cheap	Expensive
SPEED	Slow	Fast
TRANSMISSION AMOUNT	Single bit	8 bits (8 data lines) Transmitter & Receiver
TRANSMISSION LINES	One line to transmit one to receive	8 lines for simultaneous transmission
TRANSMISSION DISTANCE	Long	Short distance (synchronization)

EXTERNAL COMMUNICATION INTERFACE:

External communication interface refers to the different communication channels used by embedded system to communicate with the external world.

Various external communication interfaces are

1.RS232 AND RS485

2.USB (UNIVERSAL SERIAL BUS)

3.INFRARED

4.Bluetooth

5.Wi-Fi

6.Zigbee

7.GPRS

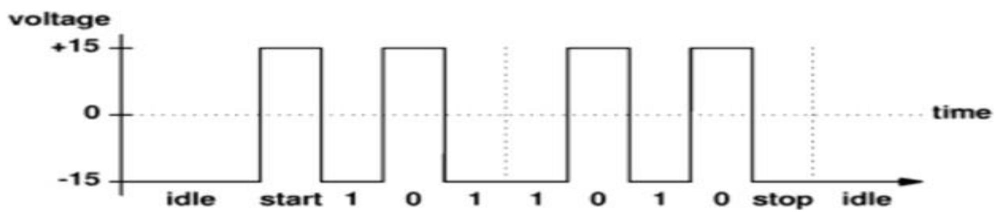
RS232 AND RS485:

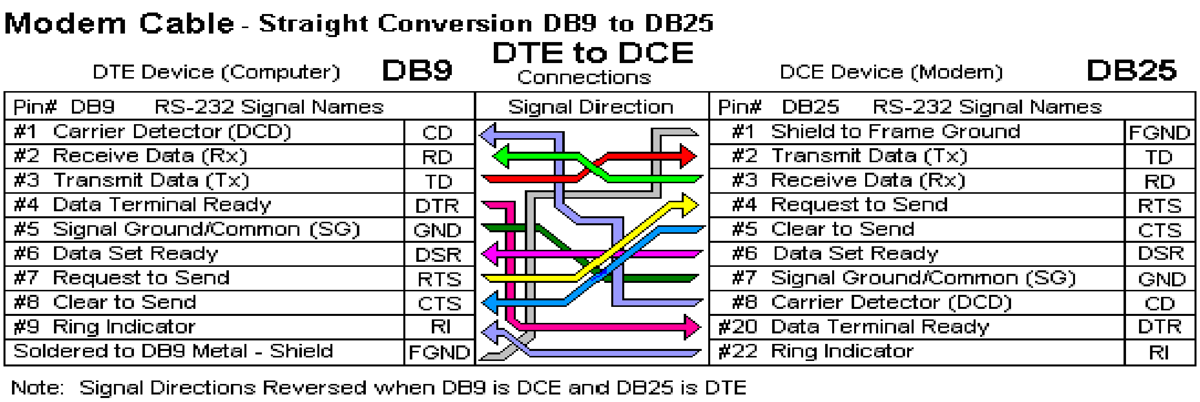
RS-232 is a standard by which two serial devices communicate:
The connection must be no longer than 50 feet.

- Transmission voltages are –15V and +15V.
- It is designed around transmission of *characters* (of 7 bits of length).
- One important aspect of RS-232 is that it is an asynchronous form of communication.
- Asynchronous communication is important because it is efficient; if no data needs to be sent, the connection is “idle.” No additional CPU overhead is

required for an idle serial line

- Logical 1 is –15VDC.
- Logical 0 is +15VDC.
- When the connection is idle, the hardware ties the connection to logical 1.
- RS-232 communication is dependent on a set timing speed at which both pieces of hardware communicate. In other words, the hardware knows how long a bit should be high or low.
- RS-232 also specifies the use of “start” and “stop” bits.





RS485:

RS-485, also known as **TIA-485(-A)**, **EIA-485**, is a standard defining the electrical characteristics of drivers and receivers for use in serial communications systems. Electrical signaling is balanced, and multipoint systems are supported.

- RS485 supports inexpensive localnetworks and multidrop communications links, using the same differential balanced line over twisted pair.
- The RS-485 standard specifies differential signaling on two lines rather than single-ended with a voltage referenced to ground. A logic 1 is a level greater than
  –200 mV, and a logic 0 is a level greater than +200 mV.
- The standard transmission medium is twisted-pair cable of either #22 or

#24 AWG solid wire.

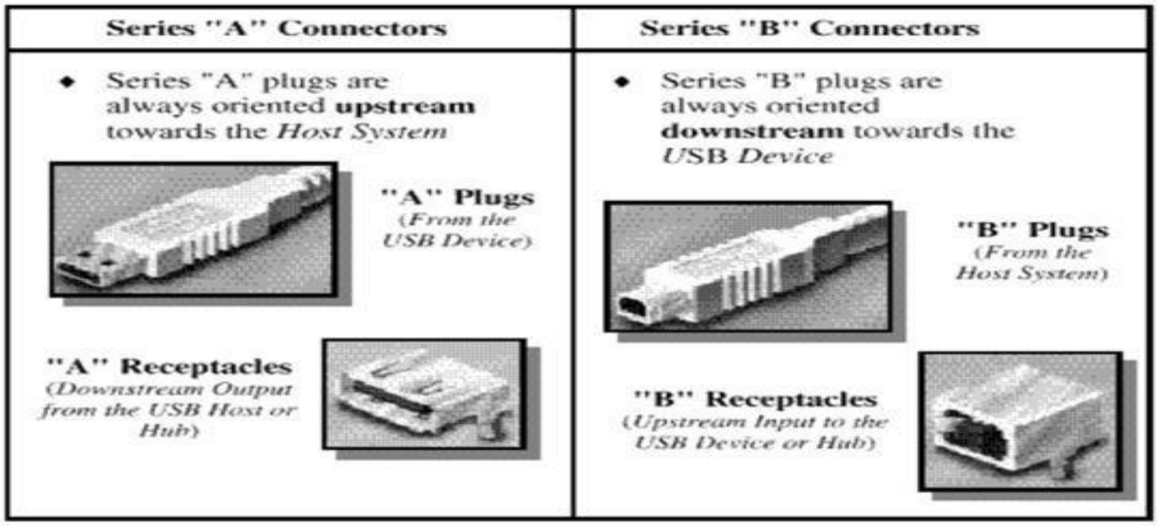
- Cable length defines the upper data rate. But because of the lower logic voltage levels and the differential connection, data rates can exceed 10 Mbits/s depending on cable length.
- Maximum cable length is commonly defined as 1200 meters or about 4000 feet. The typical maximum data rate at 4000 feet is 100 kbits/s.

- The RS-485 interface can be used in simplex of half- duplex modes with a single-pair cable.

USB (UNIVERSAL SERIAL BUS):

- External Bus Standard.
- Allows connection of peripheral devices.
- Connects Devices such as keyboards, mice, scanners, printers, joysticks, audio devices, disks.
- Facilitates transfers of data at 480 (USB 2.0 only), 12 or 1.5 Mb/s (mega-bits/second).
- Developed by a Special Interest Group including Intel, Microsoft, Compact, DEC, IBM, Northern Telecom and NEC originally in 1994.
- Low-Speed: 10 – 100 kb/s
- 1.5 Mb/s signaling bit rate
- Full-Speed: 500 kb/s – 10 Mb/s 12 Mb/s signaling bit rate
- High-Speed: 400 Mb/s
- 480 Mb/s signaling bit rate
- NRZI with bit stuffing used
- SYNC field present for every packet

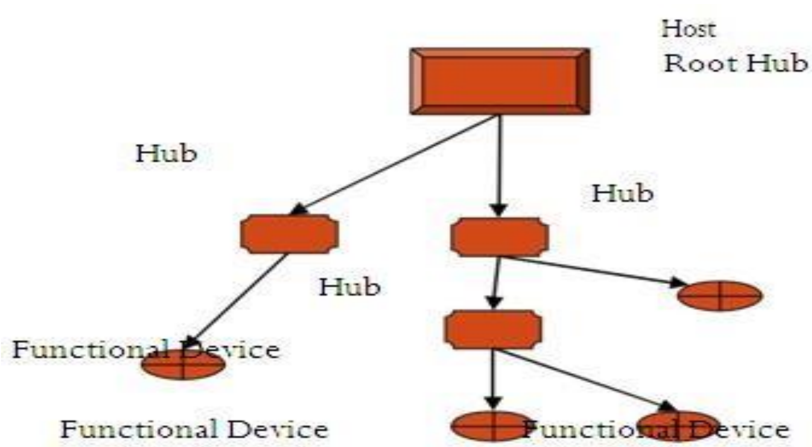
- There exist two pre-defined connectors in any USB
- system - Series "A" and Series "B" Connectors.
- Series "A" cable: Connects USB devices to a hub port.
- Series "B" cable: Connects detachable devices (hot- swappable)



Bus Topology:

- Connects computer to peripheral devices.

- Ultimately intended to replace parallel and serial ports
- Tiered Star Topology
- All devices are linked to a common point referred to as the root hub.
- Specification allows for up to 127 (2^7 -1) different devices.
- Four wire cable serves as interconnect of system - power, ground and two differential signaling lines.
- USB is a polled bus-all transactions are initiated by host.



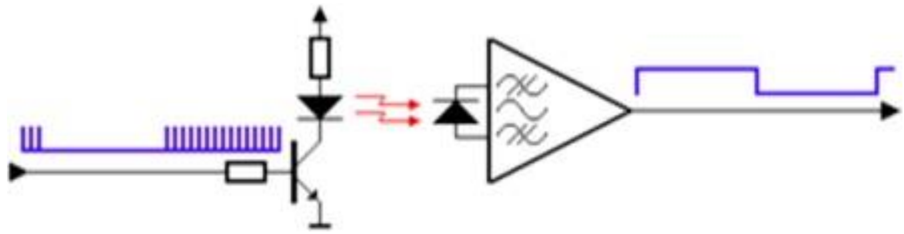
USB HOST: Device that controls entire system usually a PC of some form. Processes data arriving to and from the USB port.

USB HUB: Tests for new devices and maintains status information of child devices. Serve as repeaters, boosting strength of up and downstream signals.
Electrically isolates devices from one another - allowing an expanded number of devices.

INFRARED:

- Infrared is a certain region in the light spectrum
- Ranges from .7µ to 1000µ or .1mm
- Broken into near, mid, and far infrared
- One step up on the light spectrum from visible light
- Measure of heat

- Most of the thermal radiation emitted by objects near room temperature is infrared.
- Infrared radiation is used in industrial, scientific, and medical applications. Night-vision devices using active near-infrared illumination allow people or animals to be observed without the observer being detected.



IR transmission:

- The transmitter of an IR LED inside its circuit, which emits infrared light for every electric pulse given to it. This pulse is generated as a button on the remote is pressed, thus completing the circuit, providing bias to the LED.
- The LED on being biased emits light of the wavelength of 940nm as a series of pulses, corresponding to the button pressed. However since along with the IR LED many other sources of infrared light such as us human beings, light bulbs, sun, etc, the transmitted information can be interfered. A solution to this problem is by modulation. The transmitted signal is modulated using a carrier frequency of 38 KHz (or any other frequency between 36 to 46 KHz). The IR LED is made to oscillate at this frequency for the time duration of the pulse. The information or the light signals are pulse width modulated and are contained in the 38 KHz frequency.
- IR supports data rates ranging from 9600bits/second to 16Mbps
- Serial infrared: 9600bps to 115.2 kbps
- Medium infrared: 0.576Mbps to 1.152 Mbps
- Fast infrared: 4Mbps

BLUETOOTH:

- **Bluetooth** is a wireless technology standard for short distances (using short-wavelength UHF band from 2.4 to 2.485 GHz)for exchanging data over radio waves in the ISM and mobile devices, and building personal area networks (PANs).
- Invented by telecom vendor Ericsson in 1994, it was originally conceived as a wireless alternative to RS- 232 data cables.
- Bluetooth uses a radio technology called frequency- hopping spread spectrum. Bluetooth divides transmitted data into packets, and

transmits each packet on one of 79 designated Bluetooth channels. Each channel has a bandwidth of 1 MHz. It usually performs 800 hops per second, with Adaptive Frequency-Hopping (AFH) enabled

- Originally, Gaussian frequency-shift keying (GFSK) modulation was the only modulation scheme available. Since the introduction of Bluetooth 2.0+EDR, π/4-DQPSK (Differential Quadrature Phase Shift Keying) and 8DPSK modulation may also be used between compatible devices.

- Bluetooth is a packet-based protocol with a master- slave structure. One master may communicate with up to seven slaves in a piconet. All devices share the master's clock. Packet exchange is based on the basic clock, defined by the master, which ticks at

  312.5 µs intervals.
- A master BR/EDR Bluetooth device can communicate with a maximum of seven devices in a piconet (an ad-hoc computer network using Bluetooth technology), though not all devices reach this maximum. The devices can switch roles, by agreement, and the slave can become the master (for example, a headset initiating a connection to a phone necessarily begins as master—as initiator of the connection—but may subsequently operate as slave).

WiFi:

- Wi-Fi is trademarked name for popular wireless technology that uses radio waves to provide high-speed Internet and network connections.
- The governing body that owns the term Wi- Fi, the Wi-Fi Alliance, defines it as any WLAN (wireless area network) products that are based on the Institute of Electrical and Electronics Engineers' (IEEE) 802.11 standards.
- The way Wi-Fi works is through the use of radio signals like in phones. The wireless adapter card that is found inside of computers then uses the data that is being sent to change it into a radio signal to then be transmitted by the antenna. A router then receives these signals and decodes them in order to send the information contained within to the Internetvia a Local Area Network or a wired Ethernet connection like a cable network connection.
- Formed in 1999

FoundingCompanies:

- 3Com
- Aironet
- Harris Semi Conductor
- Lucent
- Symbol Technologies
- Nokia

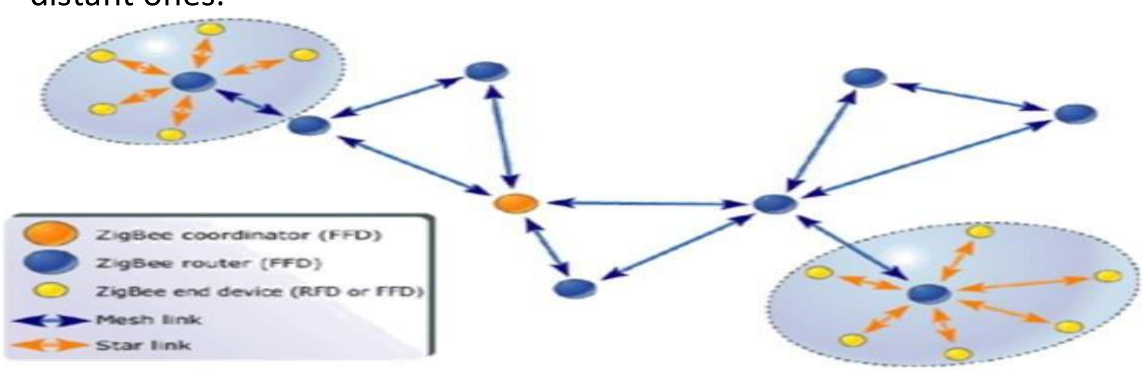
Specification:

- 802.11a
- 802.11b
- 802.11g
- 802.11n
- Devices that can use Wi-Fi technology include personal computers, video-game consoles, smartphones, digital cameras, tablet computers, smart TVs, digital audio players and modern printers.
- Wi-Fi compatible devices can connect to the Internet via a WLAN and a wireless access point. Such an access point (or hotspot) has a range of about 20 meters (66 feet) indoors and a greater range outdoors.
- Wi-Fi most commonly uses the 2.4 gigahertz (12 cm) UHF and 5 gigahertz (6 cm) SHF ISM radio bands.
- Having no physical connections, it is more vulnerable to attack than wired connections, such as Ethernet.
- The IEEE 802.11 standard is a set of media access control (MAC) and physical layer(PHY) specifications for implementing wireless local area network (WLAN) computer communication in the 2.4, 3.6, 5, and 60 GHz frequency bands. They are created and maintained by the IEEE LAN/MAN Standards Committee (IEEE802).
- The Wi-Fi Alliance approved Wi-Fi Protected Access (WPA) which uses TKIP. WPA was specifically designed to work with older equipment usually through a firmware upgrade. Though more secure than WEP, WPA has known vulnerabilities.
- The more secure WPA2 using Advanced Encryption Standard was introduced in 2004 and is supported by most new Wi-Fi devices. WPA2 is fully compatible with WPA .
- Wi-Fi supports data rates ranging from 1Mbps to 150Mbps depending on the standards and access/modulation method.

- Depending on the type of antenna and usage location. Wi-Fi offers a range of 100 to 300 feet.

ZIGBEE:

- **Zigbee** is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection.
- Hence, zigbee is a low-power, low data rate, and close proximity (i.e., personal area) wireless ad hoc network.
- The technology defined by the zigbee specification is intended to be simpler and less expensive than other wireless personal area networks (WPANs), such as Bluetooth or Wi-Fi .
- Applications include wireless light switches, electrical meters with in-home-displays, traffic management systems, and other consumer and industrial equipment that requires short-range low- rate wireless data transfer.
- Its low power consumption limits transmission distances to 10– 100 meters line-of-sight, depending on power output and environmental characteristics.
- Zigbee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to reach more distant ones.



Zigbee Coordinator: The zigbee coordinator acts as the root of the zigbee network. The ZC is responsible for initiating the Zigbee network and it has the capability to store information about the network.

Zigbee Router: Responsible for passing information from device to another device or to another ZR.
Zigbee end device:End device containing zigbee functionality for data communication. It can talk only with a ZR or ZC and doesn't have the capability to act as a mediator for transferring data from one device to another.
Zigbee supports an operating distance of up to 100 metres an data rate of 20 to 250 Kbps.

General Packet Radio Service(GPRS):

- **General Packet Radio Service** (**GPRS**) is a packet oriented mobile data service on the 2G and 3G cellular communication system's global system for mobile communications (GSM).
- GPRS was originally standardized by European Telecommunications Standards Institute (ETSI)
- GPRS usage is typically charged based on volume of data transferred, contrasting with circuit switched data, which is usually billed per minute of connection time. Sometimes billing time is broken down to every third of a minute. Usage above the bundle cap is charged per megabyte, speed limited, or disallowed.

Services offered:

- GPRS extends the GSM Packet circuit switched data capabilities and makes the following services possible:
- SMS messaging and broadcasting
- "Always on" internet access
- Multimedia messaging service (MMS)
- Push-to-talk over cellular (PoC)
- Instant messaging and presence-wireless village Internet applications for smart devices through wireless application protocol (WAP).
- Point-to-point (P2P) service: inter-networking with the Internet (IP).
- Point-to-multipoint (P2M) service[*citation needed*]: point- to-multipoint multicast and point-to-multipoint group calls.

Protocols supported:

- GPRS supports the following protocols:
- InternetProtocol(IP).Inpractic,built-inmobile browsers use IPv4 since IPv6 was not yet popular.
- Point-to-Point Protocol (PPP). In this mode PPP is often not supported by the mobile phone operator but if the mobile is used as a modem to the connected computer, PPP is used to tunnel IP to the phone. This allows an IP address to be assigned dynamically (IPCP not DHCP) to the mobile equipment.
- X.25 connections. This is typically used for applications like wireless payment terminals, although it has been removed from the standard.
  X.25 can still be supported over PPP, or even over IP, but doing this requires either a network-based router to perform encapsulation or intelligence built into the end-device/terminal; e.g., user equipment (UE).