Unit-1

Embedded System Design

- It is a system designed with the **embedding of hardware and software** together for a specific function with a larger area.
- An embedded system can be an independent system or it can be a part of a large system and it is a microcontroller or microprocessor based system.
- **Examples** home appliances, smart devices, agricultural and process industry devices, automobiles, medical equipment etc.

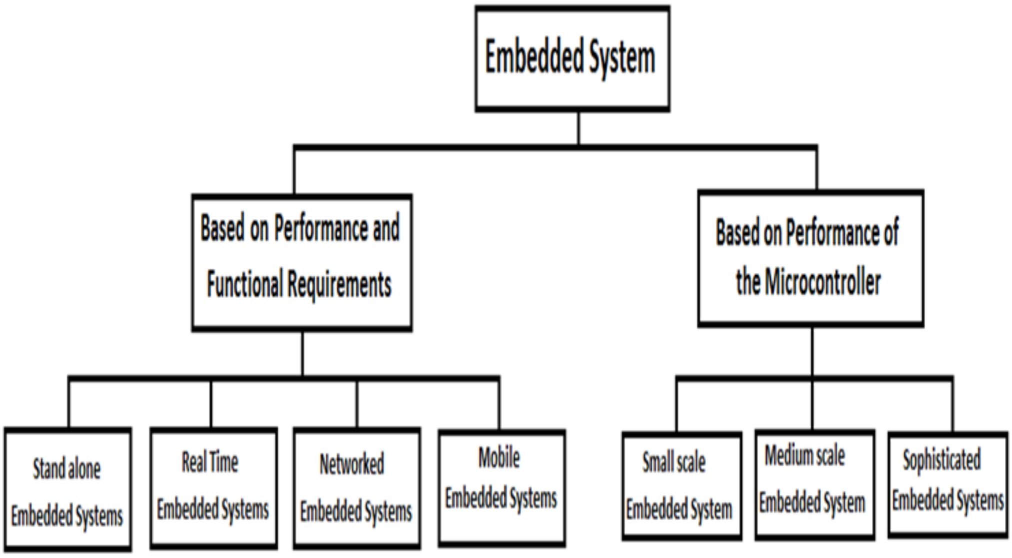
Embedded systems vary in complexity but, generally, consist of three main elements:

- **Hardware.** The hardware of embedded systems is based around microprocessors and microcontrollers. Microprocessors are very similar to microcontrollers and, typically, refer to a CPU (central processing unit) that is integrated with other basic computing components such as memory chips and digital signal processors (DSPs). Microcontrollers have those components built into one chip.
- **Software and firmware.** Software for embedded systems can vary in complexity. Application software allows the user to perform varieties of applications to be run on an embedded system by changing the code installed in an embedded system.
- **Real-time operating system.** These are not always included in embedded systems, especially smaller-scale systems. RTOSes acts as an interface between hardware and software which supervises the application software and provide mechanism to let the processor run on the basis of scheduling.

CATEGORIES OF EMBEDDED SYSTEMS

Embedded systems can be classified into two broad categories as follows;

1. Based on performance and functional requirements.
2. Based on performance of the microcontroller.



Based on performance and functional requirements of system embedded systems are classified into four categories as follows;

- Stand alone embedded systems
- Real time embedded systems
- Networked embedded systems
- Mobile embedded systems

Stand alone Embedded systems: A stand-alone embedded system works by itself. It is a self-contained device which does not require any host system like a computer.

It takes either digital or analog inputs from its input ports, calibrates, converts, and processes the data, and outputs the resulting data to its attached output device, which either displays data, or controls and drives the attached devices.

EX: Temperature measurement systems, Video game consoles, MP3 players, digital cameras, and microwave ovens are the examples for this category.

Real time embedded system: An embedded system which gives the required output in a specified time or which strictly follows the time deadlines for completion of a task is known as a Real time system. A Real Time system, in addition to functional correctness, also satisfies the time constraints. It works on the basis of real-time computing represented by a dedicated type of operating system — **RTOS.** Real-time embedded systems fall into three categories depending on the acceptability of violation of time constraints- Hard, Firm and Soft Real time systems.

Ex: car airbag control system, Satellite-based surveillance applications, manufacturing robots, Washing machine etc.

Networked embedded systems: The networked embedded systems are related to a network with network interfaces to access the resources. The connected network can be a Local Area Network (LAN) or a Wide Area Network (WAN), or the Internet. The connection can be either wired or wireless. The networked embedded system is the fastest growing area in embedded systems applications. The embedded web server is such a system where all embedded devices are connected to a web server and can be accessed and controlled by any web browser.

Ex: A home security system is an example of a LAN networked embedded system where all sensors (e.g. motion detectors, light sensors, or smoke sensors) are wired and running on the TCP/IP protocol.

Mobile embedded systems: The portable embedded devices like mobile and cellular phones, digital cameras, MP3 players, PDA (Personal Digital Assistants) are the example for mobile embedded systems. The basic limitation of these devices is the limitation of memory and other resources

Another category of embedded system based on performance of microcontroller and it is further classified into three categories as follows;

1. Small scale embedded system
2. Medium scale embedded system
3. Sophisticated Embedded Systems

Small Scale Embedded Systems: These types of embedded systems are designed with a single 8-bit or 16-bit microcontroller with on-chip RAM and ROM designed to perform simple tasks. They have tiny scaled hardware, software complexities and involve board-level design. They may even be battery operated.

Medium Scale Embedded Systems: These systems are usually designed with a single or few 16-bit or 32-bit microcontrollers or Digital Signal Processor (DSPs) with external RAM and ROM that can perform more complex operations.

Large Scale /Sophisticated Embedded Systems: Sophisticated embedded systems have massive hardware and software complexities and may require ASIPs or 32-64-bit multiple chips which can perform distributed jobs.

Major Application Areas of Embedded Systems:

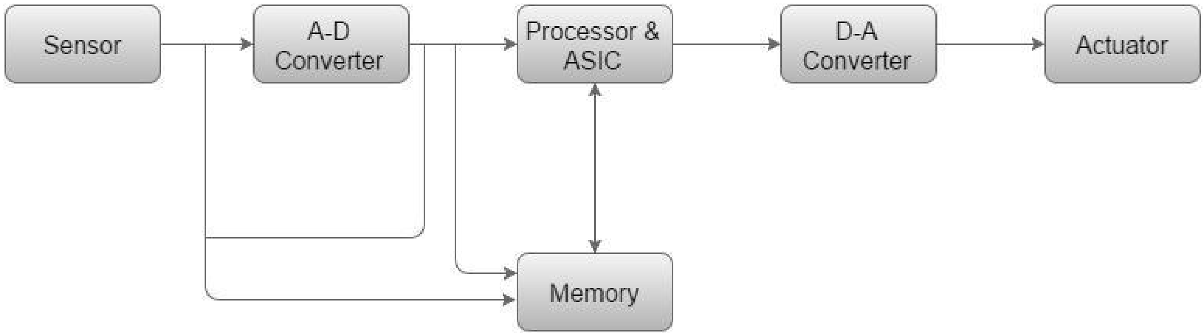
- **Consumer Electronics:** Camcorders, Cameras etc.
- **Household Appliances:** Television, DVD players, washing machine, Fridge, Microwave Oven etc.
- **Home Automation and Security Systems:** Air conditioners, sprinklers, Intruder detection alarms, Closed Circuit Television Cameras, Fire alarms etc.
- **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.

Overview of Embedded System Architecture

- Every embedded system consists of customer-built hardware components supported by a Central Processing Unit (CPU), which is the heart of a microprocessor (µP) or microcontroller (µC).
- Operating system plays an important role in most of the embedded systems. But all the embedded systems do not use the operating system.
- The systems with high end applications only use operating system. To use the operating system the embedded system should have large memory capability.
- So, this is not possible in low end applications like remote systems, digital cameras, MP3 players, robot toys etc.

- In the case of embedded systems with OS, once the application software is loaded into memory it will run the application without any host system.

The following illustration shows the basic structure of an embedded system −



- **Sensor** − It measures the physical quantity and converts it to an electrical signal which can be read by an observer or by any electronic instrument like an A2D converter. A sensor stores the measured quantity to the memory.

 A-D Converter − An analog-to-digital converter converts the analog signal sent by the sensor into a digital signal.
 Processor & ASICs − Processors process the data to measure the output and store it to the memory.
 D-A Converter − A digital-to-analog converter converts the digital data fed by the processor to analog data
 Actuator − An actuator compares the output given by the D-A Converter to the actual (expected) output stored in it and stores the approved output.

Recent Trends in Embedded Systems

- The emergence of IoT and the Industrial IoT (IIoT) has paved the way for other existing technologies to flourish under these innovations and one such technology is the embedded systems technology. Embedded systems have become a catalyst for the rapidly expanding world of smart and connected IoT ecosystems.
- And with its broad diversity in functions and flexibility embedded system technology has become the backbone for many industries and services. The popularity of this technology can also be attributed to its recent evolution in performing its functions faster, becoming more compact, and becoming cost-effective.
- With the fast developments in semiconductor industry and VLSI technology, one can find tremendous changes in the embedded system

design in terms of processor speed, power, communication interfaces including network capabilities and software developments like operating systems and programming languages etc.

Some of the emerging Trends in Embedded System Technology are:

Multicore in embedded

- With a lot functionalities being added, the need for high performance in embedded systems has become inevitable and so developers are increasingly leaning towards multicore processors in their systems design decision.
- Today most of the gaming consoles are multicore and so are smartphones, which are indeed getting 'smarter'. While this multicore paradigm offers benefits, there is also ample opportunity for the engineers to learn on this new design space – on architecture, design, programming, debugging and testing so that they are well informed and are aware about the optimal use of new power that a multicore offers.

Embedded operating systems

- Traditionally embedded systems did away with an operating system (OS), however, as the systems became complex, it was inevitable to have OS which offered low latency real-time response, low foot print both in time and space and give all traditional functionality such as memory protection, error checking/report and transparent interprocess communication,.
- OS like Linux embedded and new OS such as Android are making inroads into places where traditionally Windows CE/Vxworks etc., used to play. Today many new handhelds and smart phones are embracing Andriod. Eclipse, the open source project for building development platforms offers an environment that crosses over RTOS boundaries.

Embedded digital security and surveillance

- Digital security and surveillance is currently in the host of new applications in the embedded arena which is benefiting from multicore phenomenon.
- The applications based on computer vision and tracking offers multiple benefits in capturing, post processing and identification and alerting of security video in realtime.
- With the rise of security issues in the last few years whether it is due to the rapid increase of digital users or the change in the lifestyle of people after going remote in pandemic times, security has never been under such a great threat as it is now.

- So, the development of implementing IoT security solutions along with embedded devices like AI and Robots can be a great help in this regard.

Automation

- Automation has made its place in one of the Top most trends in Embedded System Technology due to its powerful impact in the development of the modern era with its innovative approach.
- We are seeing its impact in industries as well as in the market as all the systems and products are getting automated due to its growing popularity among people.
- Due to the impacts and popularity of computers and robots, automation can be found in every sector of development.

Automobiles

- The automotive application can be considered another case of emerging Trends in Embedded System Technology.
- Due to their various implementation uses, embedded systems are utilized in different ways like for infotainment, safety, driver awareness, and overall system control of the vehicle among a few.
- Moreover, intelligent systems control is rapidly expanding its reach in the automobile industry with the rise of hybrid electric vehicles (HEV) and electric vehicles (EV).

Artificial Intelligence

- Artificial intelligence (AI) and its practical application are steadily gaining momentum as more embedded designs are constantly been developed to be included in the concepts over time.
- AI has reached almost every type of segment you can think of ranging from retail, health care, autonomous driving, e-commerce, manufacturing, supply chain, banking, etc.
- And with the AI hardware decreasing in price as time passes, embedded designs are expected to increase AI capability to many folds.

System-on-Chip Solution

- Embedded devices are making their presence known globally by using the SOC market. Many companies are offering SoC solutions for the embedded devices.
- One such solution includes ASIC having high performance, small size, low cost, and IP protection.
- ASICs are essentially valuable for medical wearable manufacturers due to their necessary size, weight, and power requirements for their devices.

Hardware Architecture of Embedded System

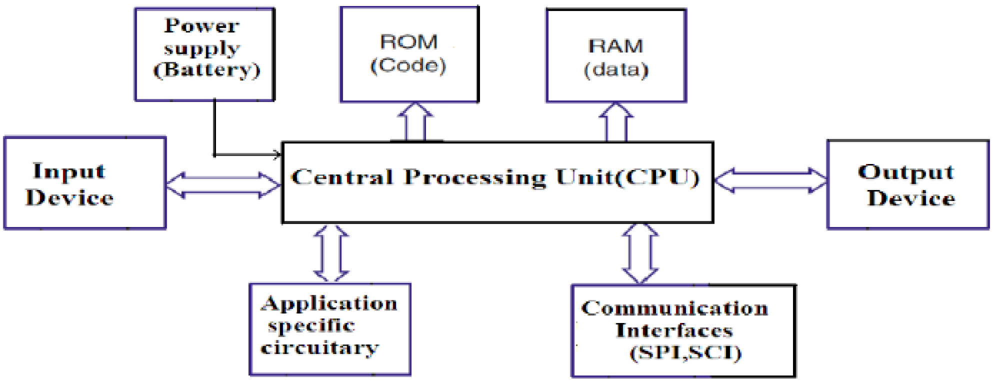


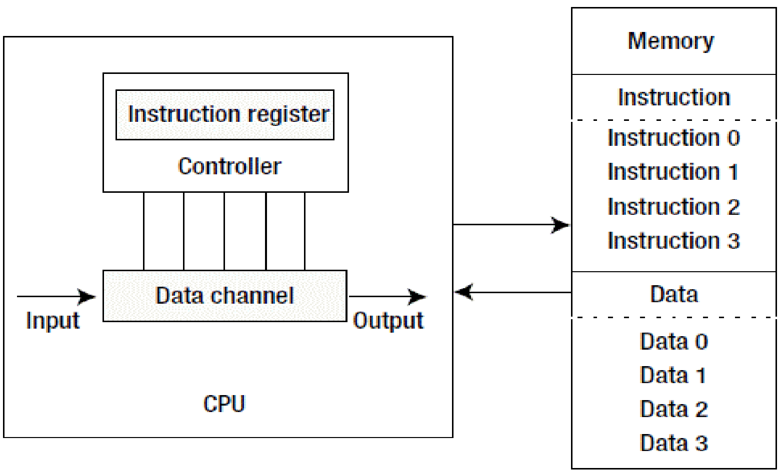
Fig: hardware architecture of embedded system

There are basically two types of architecture that apply to embedded systems:

- Von Neumann architecture
- Harvard architecture

Von Neumann Architecture:

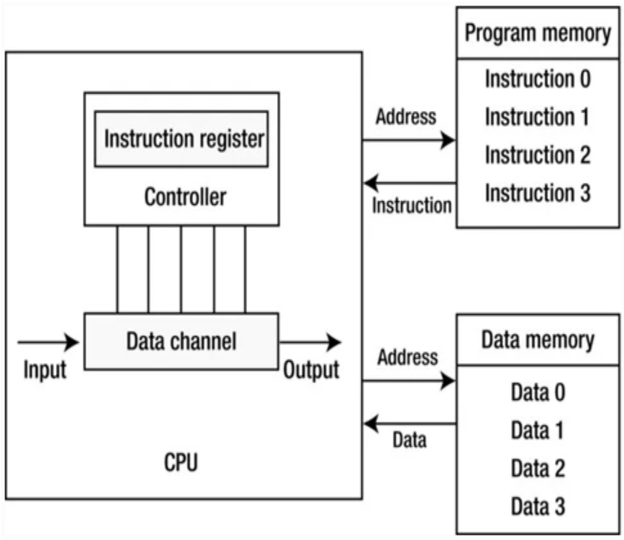
- The Von Neumann architecture was first proposed by a computer scientist John von Neumann.
- In this architecture, the same memory and bus are used to store both data and instructions that run the program. One data path or bus exists for both instruction and data.
- As a result, the CPU does one operation at a time. It either fetches an instruction from memory, or performs read/write operation on data. So an instruction fetch and a data operation cannot occur simultaneously, sharing a common bus.



- Von-Neumann architecture supports simple hardware. It allows the use of a single, sequential memory.

Harvard Architecture:

- The Harvard architecture was first named after the Harvard Mark I computer. Compared with the Von Neumann architecture, a Harvard architecture processor has two outstanding features.
- First, instructions and data are stored in two separate memory modules; instructions and data do not coexist in the same module.
- Second, two independent buses are used as dedicated communication paths between the CPU and memory; there is no connection between the two buses.
- This architecture offers separate storage and signal buses for instructions and data.



- Computers have separate memory areas for program instructions and data using internal data buses, allowing simultaneous access to both instructions and data.

Von Neumann architecture	**Harvard architecture**
Requires single bus for instructions and data	Requires separate & dedicated buses for memories for instructions and data.

Von Neumann architecture	Harvard architecture
Its design is simpler	Its design is complicated
The Von Neumann architecture uses single memory for their instructions and data	The Harvard architecture uses physically separate memories for their instructions and data
Examples of Von – Neumann Architecture: ARM 7 and Pentium Processors etc.	Examples of Harvard Architecture: 8051, ARM 9, AVR by Atmel Corporation and PIC microcontrollers by microchip Technology etc.

Central Processing Unit:

- The CPU in the embedded system may be a general purpose processor like a microcontroller or a special purpose processor like a DSP (Digital signal processor).
- But any CPU consists of of an Arithmetic Logic Unit (ALU), a Control Unit (CU), and many internal registers that are connected by buses. The ALU performs all the mathematical operations (Add, Sub, Mul, Div), logical operations (AND, OR), and shifting operations within CPU.
- The CPU works in a cycle of fetching an instruction, decoding it, and executing it, known as the fetch-decode-execute cycle.
- There are many internal registers in the CPU.
- The accumulator (A) is a special data register that stores the result of ALU operations. It can also be used as an operand. The Program Counter (PC) stores the memory location of the next instruction to be executed.
- The Instruction Register (IR) stores the current machine instruction to be decoded and executed.
- The Data Buffer Registers store the data received from the memory or the data to be sent to memory.
- The Data Buffer Registers are connected to the data bus.

- The Address Register stores the memory location of the data to be accessed (get or set). The Address Register is connected to the address bus.

Memory:

- Embedded system memory can be either on-chip or off-chip.
- On chip memory access is much fast than off-chip memory, but the size of on-chip memory is much smaller than the size of off-chip memory.
- The ROM, EPROM, and Flash memory are all read-only type memories often used to store code in an embedded system.

Communication Interfaces:

- To transfer the data or to interact with other devices, the embedded devices are provided the various communication interfaces like RS232, RS422, RS485,USB, SPI(Serial Peripheral Interface) ,SCI (Serial Communication Interface) ,Ethernet etc.

Application Specific Circuitry:

- The embedded system sometimes receives the input from a sensor or actuator. In such situations certain signal conditioning circuitry is needed. This hardware circuitry may contain ADC, Op-amps, and DAC etc. Such circuitry will interact with the embedded system to give correct output.

ADC & DAC:

- Many embedded system application need to deal with non-digital external signals such as electronic voltage, music or voice, temperature, pressures, and many other signals in the analog form.
- The digital computer does not understand these data unless they are converted to digital formats.
- The ADC is responsible for converting analog values to binary digits.
- The DAC is responsible for outputting analog signals for automation controls such as DC motor.

In addition to these peripherals, an embedded system may also have sensors like temperature sensors, light sensors, gas sensors etc and Display modules like LCD or Touch screen panels.

Address bus and data bus:

- Address bus is a part of the computer system bus that is dedicated for specifying a physical address.
- When the computer processor needs to read or write from or to the memory, it uses the address bus to specify the physical address of the individual memory block it needs to access.
- A data bus simply carries data. The same data bus is used for both read/write operations.

Real Time Embedded Systems

- The embedded systems which respond to real time situation with the help of its embedded software and hardware, within the specified time constraints are called real time embedded systems.
- It works on the basis of real-time computing represented by a dedicated type of operating system — **RTOS**.

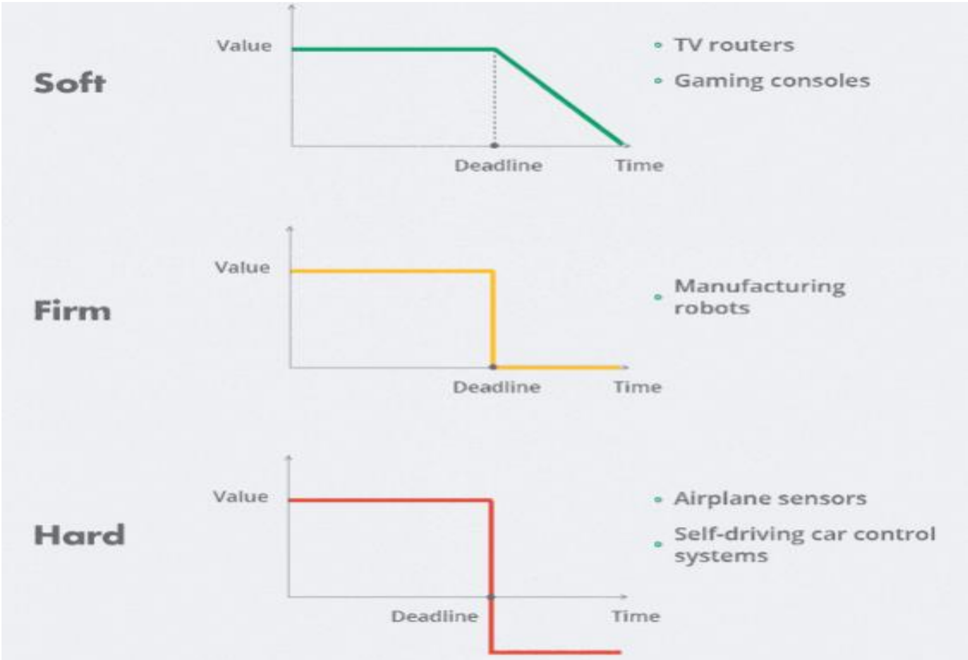
Its working principles are as follows:

- **Quick response to external factors:** an embedded system must work within fixed time constraints.
- **Predictability:** must be deterministic or predictable, meaning that no deviations are allowed.
- **The deadline is above all**: meeting the deadline is more important than other performance characteristics.
- **Operational failures may lead to catastrophe:** if a task does not meet time limits, it negatively affects users and may even lead to fatal results.

Categories of Real-Time Embedded Systems:

Real-time embedded systems fall into three categories depending on the acceptability of violation of time constraints:

- soft,
- firm and
- hard



a) **Hard real time E.S:** A Real time system in which, the violation of time constraints will cause critical failure and loss of life or property damage or catastrophe is known as a Hard Real time system.
For such built-in systems, it's crucial that the deadline is met in all cases. These systems usually interact directly with physical hardware instead of through a human being. The hardware and software of hard real-time systems must allow a worst case execution (WCET) analysis that guarantees the execution be completed within a strict deadline. The chip selection and RTOS selection become important factors for hard real-time system design.

Eg: : An air defense system that needs to detect and intercept an attacking missile within milliseconds has this type of embedded system. Other examples are airplane sensors or self-driving car control systems, Delayed alarm during a Gas leakage , car airbag control system , A delayed response in pacemakers, Failure in RADAR functioning etc.

b) **Firm Real time E.S:** A Real time system in which, the violation of time constraints don't result in harmful effects, but system loses its performance value is known as Firm Real time system. Exceeding the deadline is occasionally permitted in these systems, though it is undesirable.
Every firm real-time task is associated with some predefined deadline before which it is required to produce its results. If it does not complete

within its deadline, the system doesn't fail but the late results are merely discarded. In other words, the utility of the results computed by a firm real-time task becomes zero after the deadline.

Eg:: Manufacturing robots refer to this category, A video conferencing application, Satellite-based surveillance applications

c) **Soft Real time E.S:** A Real time system in which, the violation of time constraints will cause only the degraded quality, but the system can continue to operate is known as a Soft real time system. Exceeding the deadline is acceptable in these systems. Response failures of soft real-time systems diminish user experience, but they don't reduce the performance value at once.
If such a system fails to meet a deadline, it will either recover or gradually diminish its operation. In soft real-time systems, the design focus is to offer a guaranteed bandwidth to each real-time task and to distribute the resources to the tasks.

Eg:: TV router boxes and gaming consoles refer to soft real-time embedded systems. Time lags in their operation may happen, but they have either insignificant or no consequences, Microwave Oven, washing machine, TV remote etc.

Constraints of Real time Embedded Systems:

Physical Constraints:

These refer to hardware components and external environmental factors. Embedded engineers usually have to create a product that should satisfy the following conditions:

- Definite device size
- Spatial constraints for device installation
- Limited memory and power consumption
- Certain environmental conditions for device operation (temperature, humidity, pressure)

Timing Constraints:

- Real-time systems must respond to events within predefined time limits.

- The most severe challenge for developers is to create a system capable of meeting the deadline under any conditions.
- They must do accurate calculations and build the appropriate task scheduling system to succeed.

Task Scheduling

RTOS comprise scheduling algorithms that are responsible for managing timing constraints. There are *preemptive* and *non- preemptive* algorithms.

- **Priority scheduling.** This algorithm prioritizes all tasks and puts forward the task with the highest priority to be performed first by the processor. A preemptive version of the algorithm stops a running task if there is another one with a higher priority in the queue. A non-preemptive version doesn't stop running tasks, but a higher priority task will be the next in the line. This approach doesn't suit tasks with equal priority.
- **Round-robin scheduling.** This is a preemptive scheduling algorithm that doesn't prioritize tasks. Instead, it allocates an equal time interval (e.g., 500 ms) for each task, and the CPU processes them one by one. One task may go several rounds to be completed. This approach is relatively easy and straightforward.
- **First come, first served (FCFS).** This is a non-preemptive algorithm that puts tasks into the running state depending on the time they arrive. The process is straightforward: the task that comes first is the first to utilize the computing power. This algorithm ensures a high response time.
- **Shortest job first (SJF).** The non-preemptive version of this algorithm allocates tasks depending on their execution time: the task with the shortest execution time is run next. The preemptive variant can interrupt running tasks if a task with a shorter remaining execution period arrives.

Microprocessors and Microcontrollers

Microprocessor

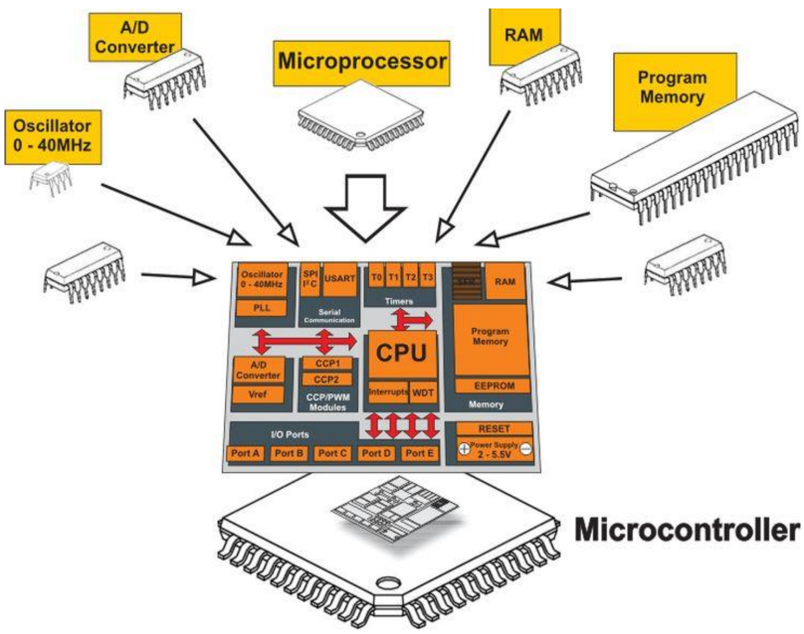
- A microprocessor is a computer processor where the data processing logic and control is included on a single integrated circuit, or a small number of integrated circuits.
- The microprocessor contains the arithmetic, logic, and control circuitry required to perform the functions of a computer's central processing unit.

- Microprocessor is used for applications that require intensive processing and is not made for a specific task.
- It is used in the development of software, games, and other applications that require high memory and where input and output are not defined. Eg: Complex home security, Home computers, Video game systems

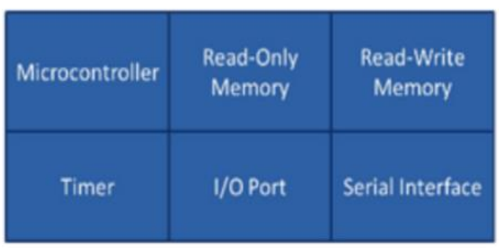
Microcontroller

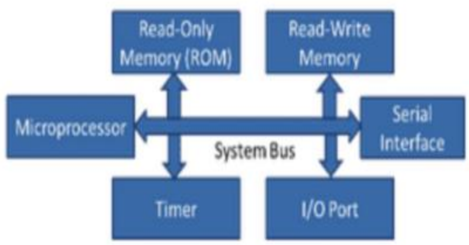
- A microcontroller is a compact integrated circuit designed to perform a specific operation in an embedded system.
- A typical microcontroller includes a processor, memory and input/output (I/O) peripherals on a single chip.
- It is specially designed circuits for embedded applications and is widely used in automatically controlled electronic devices. e.g. Calculator, Washing Machine, ATM machine, Robotic Arm, Camera, Microwave oven, Oscilloscope, Digital multimeter, ECG Machine, Printer

Difference between Microprocessor and Microcontroller



Microprocessor	Microcontroller





1.	Microprocessor is a component that is used for multiple processes; it is the heart of Computer system.	Micro Controller is an integrated circuit designed for a specific task; it is the heart of an embedded system.
2.	It is only a processor, so memory and I/O components need to be connected externally	Micro Controller has a processor along with internal memory and I/O components.
3.	Memory and I/O has to be connected externally, so the circuit becomes large.	Memory and I/O are already present, and the internal circuit is small.
4.	It cannot be used in compact systems	It is used in compact systems.
5.	Cost of the entire system is high	Cost of the entire system is low
6.	Due to external components, the total power consumption is high. Therefore, it is not ideal for the devices running on stored power like batteries.	As external components are low, total power consumption is less. So it can be used with devices running on stored power like batteries.
7.	Most of the microprocessors do not have power saving features.	Most microcontrollers have power-saving modes like idle mode and power-saving mode. This helps to reduce power consumption even further.
8.	Microprocessor has a smaller number of registers, so more operations are memory-based.	Microcontroller has more register. Hence the programs are easier to write.
9.	It uses more RAM and can vary from 512MB to 32GB.	The RAM can be either from 2KB up to 256KB. It can have a flash memory of 32KB to 2MB.
10.	Microprocessors are based on Von Neumann model	Micro controllers are based on Harvard architecture
11.	It uses an external bus to interface to RAM, ROM, and other peripherals.	It uses an internal controlling bus.
12.	It has a very high clock speed. Microprocessor-based systems can run at a very high speed of up to 1GHz.	It has low clock speed. Microcontroller based systems run up to 200MHz or more depending on the architecture.
13.	It’s complex and expensive, with a large number of instructions to	It’s simple and inexpensive with less number of instructions to process.

	process.	
14.	A microprocessor has peripheral interfaces such as USB, UART, and high-speed Ethernet.	A Microcontroller contains peripheral interfaces such as I2C, UART, and SPI.
15.	It is mainly used in personal computers.	It is used mainly in a washing machine, MP3 players, and embedded systems.

Embedded Controllers (Microcontrollers)

- A **microcontroller** is a compact integrated circuit designed to govern a specific operation in an embedded system.
- Sometimes referred to as an embedded controller or microcontroller unit (MCU), microcontrollers are found in vehicles, robots, office machines, medical devices, mobile radio transceivers, vending machines and home appliances, among other devices.
- They are essentially simple miniature personal computers (PCs) designed to control small features of a larger component, without a complex front-end operating system (OS).
- Embedded controllers are the general class of microcontrollers used to support OEM-specific implementations.
- Embedded Controllers (ECs) are often found in low power embedded reference designs, performing a range of Input/Output (I/O) and system management functions.
- An Embedded Controller can be considered to be a microcontroller, with I/O and internal features targeted to suit the typical needs of a low power platform.

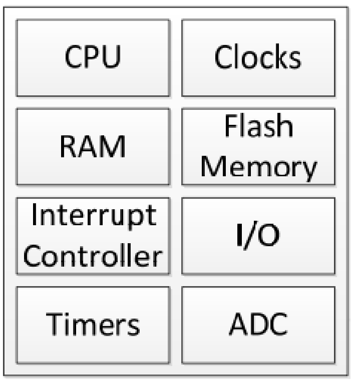


Figure 1. Generic Embedded Controller Block Diagram

- These blocks can be split into two main groups.
  1. Hard-wired
  2. Programmable
- The hard-wired group comprises items which have one hardware defined function such as the clocks, interrupt controller, I/O, Timers and Analog-toDigital Converter.
- The programmable group consists of the CPU, RAM & Flash memory blocks and the functions of those blocks are defined by the firmware (controller's software) programmed in the Flash memory.
- The hard-wired functionality is defined by the vendor of the Embedded Controller.
- The programmable functionality is generally defined by the OEM/ODM that is developing the system. This programmability allows the Embedded Controller to be customized to the specific needs of the platform and to be differentiated from the same EC used by another OEM or ODM.

Elements of a microcontroller:

The core elements of a microcontroller are:

The processor (CPU) -- A processor can be thought of as the brain of the device. This involves performing basic arithmetic, logic and I/O operations.
Memory -- A microcontroller's memory is used to store the data that the processor receives and uses to respond to instructions that it's been programmed to carry out. A microcontroller has two main memory types:

- Program memory, which stores long-term information about the instructions that the CPU carries out.
- Data memory, which is required for temporary data storage while the instructions are being executed.

I/O peripherals -- The input and output devices are the interface for the processor to the outside world. The input ports receive information and send it to the processor in the form of binary data. The processor receives that data and sends the necessary instructions to output devices that execute tasks external to the microcontroller.

There are many supporting components that can be classified as peripherals.

- **Analog to Digital Converter (ADC)** -- An ADC is a circuit that converts analog signals to digital signals. It allows the processor at the center of the microcontroller to interface with external analog devices, such as sensors.

- **Digital to Analog Converter (DAC)** -- A DAC performs the inverse function of an ADC and allows the processor at the center of the microcontroller to communicate its outgoing signals to external analog components.
- **Clock** -- The clock is used to control the clocking requirement of the CPU for executing instructions and the configuration of timers.
- **Timer** -- A timer is a specialized type of clock which is used to measure time intervals. It counts down from a specified time interval and generates a time delay.
- **Interrupt Controller** -- The interrupt controller provides a simple software interface to the interrupt system. It manages interrupts from various different peripheral devices.

An embedded controller can have the following tasks:

- Receiving and processing signals from the keyboard and the touchpad (including touchpad disable), other buttons and switches.
- Thermal measurement (CPU, GPU, Motherboard) and response including fan control, CPU and GPU throttling, and emergency shutdown in response to rising temperatures
- Controlling indicator LEDs (e.g. caps lock, scroll lock, num lock, battery, ac, power, wireless LAN, sleep)
- Managing the battery charger and the battery
- Performing software-requested CPU reset
- Controlling the watchdog timer

Robots and Robotics

- A **robot** is a type of automated machine that can execute specific tasks with little or no human intervention.
- Robot is designed to execute one or more tasks automatically with speed and precision.
- A robot can sense, plan and act according to the situation.

Robotics is the branch of technology that deals with the design and development of robots. **Robotics and Embedded Systems** are used extensively in modern technologies, including automotive, image processing, 3D printing, and healthcare devices, as well as real-time critical systems and search and rescue operations

History of Robotics

- Czech writer, Karel Capek in his drama ***R.U.R.* (*Rossum's Universal Robots*)** introduced the word robot to the world in1921.It is derived from the Czech word robota meaning "forced labourer".
- In 1941 Russian-born American science-fiction writer, Isaac Asimov, coined the word robotics and formulated the **Three Laws of Robotics.**
- In 1954 George Devol invented the first digitally operated and programmable robot, Unimate, which represents the foundation of the modern robotics industry.

- More advanced computer-controlled electric arms guided by sensors were developed in the late 1960s and 1970s where they were used with cameras in robotic hand-eye research.
- The first such design called PUMA (Programmable Universal Machine for Assembly) was used to assemble automobile subcomponents such as dash panels and lights.
- In 1989 a hexapedal robot was modelled to traverse difficult terrain. It was notable for its cheap construction and development time, which has given rise to a trend of incremental development in robotics.

- In 1999 Sony released a robotic dog, AIBO that was capable of interacting with humans. It was wildly popular as a commercial product that was able to be programmed for a variety of different functions, up to and including animatronic sports.
- In 2011 Robonaut-2 was launched to the International Space Station and became the first humanoid robot in space. It currently serves as a training tool for roboticists in space, though is currently being upgraded to help astronauts complete dangerous, out of station spacewalks.

- In 2017 the robot Sophia was granted Suadi Arabian citizenship. It became the first robot to be recognized with a gender identity and nationality.

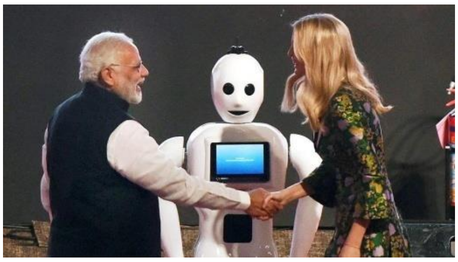
Robots made in India:

Manav – India's First 3D-Printed Humanoid Robot

Manav, India's first 3D humanoid robot was developed by the A-SET Training and Research Institute in Delhi. This two-kilogram, two-foot humanoid is primarily intended for research purposes, and it has an integrated vision and sound processing capability that allows it to walk, talk, and dance only in response to human commands.

Mitra – Humanoid Capable Of Interacting With Humans

Mitra is a five-foot humanoid robot capable of intelligent human interaction. It was developed by Invento Robotics, a Bengaluru-based robotics company, and was launched at the Global Entrepreneurship Summit (GES) in 2017 by Prime Minister Narendra Modi and Ivanka Trump. This robot can be found conversing with and greeting customers in the corridors of Canara Bank and PVR Cinemas in Bengaluru.



Robocop- World's first police officer robot

H-Bots Robotics based in Hyderabad, a start-up in artificial intelligence and machine learning, has developed Robocop, a police robot that can assist with law enforcement and traffic control. The life-size robot has cameras and a variety of sensors, including ultrasonic, proximity, and temperature sensors. The robot is designed to protect and secure offices, shopping malls, airports, signal stations, and other public places.

Vyommitra Robot

Vyommitra is a female humanoid robot being developed by the ISRO to function on-board the *Gaganyaan.* Vyommitra was first unveiled on 22 January 2020 at the *Human Spaceflight and Exploration symposium* in Bengaluru. It will accompany Indian astronauts in space missions and will also be a part of uncrewed experimental Gaganyaan missions prior to the crewed spaceflight missions.

INDRO – Tallest Robot

This is India's tallest humanoid robot. INDRO is an autonomous robot designed by researcher Santosh Vasudeo Hulawale that was built inside a house using readily available low-cost materials such as aluminum, wood, cardboard, and plastic. INDRO can be used for light tasks such as entertainment, education, and a few household chores. It has 31 motors and can lift objects weighing up to 2 kilograms with its hands.

Shalu Robot

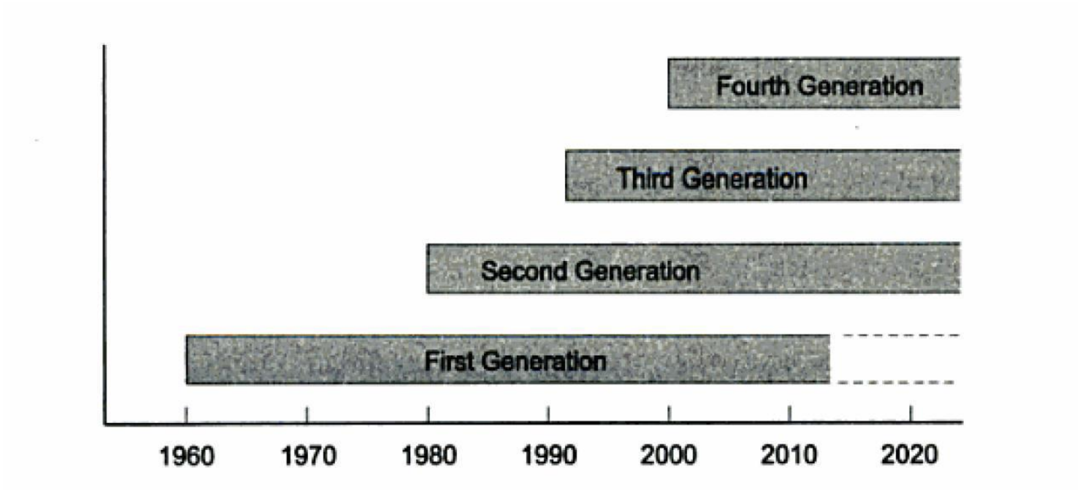
Shalu is the first female humanoid robot in India developed by Dinesh Kunwar Patel, a Kendriya Vidyalaya, Computer Science teacher from Mumbai. The robot was made at home using waste materials as well as aluminum, plastic, cardboard, wood, newspapers, and other things readily available in the local market. Shalu can speak 47 languages (9 Indian and 38 foreign). She can recognize people and remember them as well as identify many common objects. She can understand human emotions and act accordingly. It can be used as a robot-teacher in a classroom to make an interactive class with children.

Apart from the above list there many more robots developed in India and are being used for various purposes.
Eg: **KEMPA** is set up to provide flight and check-in information, as well as other flight-related information. **IRA 2.0**, introduced by HDFC Bank, is an interactive robotic assistant designed to improve the user experience for branch visitors. **Daksha** is popular for being the first anti-terror robot in India developed by the DRDO.

Laws of Robotics:

1. A robot should not injure a human being or through inaction allow a human to be harmed.
2. A robot must obey orders given by humans except when that conflicts with the first law.
3. A robot must protect its own existence unless that conflicts with the first or second law.

Generations of Robot:
First Generation: The first generation robots are repeating, non servo controlled type used for pick and place and point to point operations. The technology for these is fully developed and at present about 80% robots in use in industry is of this kind. It is predicted that these will continue to be in use for a long time.
Second Generation: The addition of sensing devices and enabling the robot to alter its movements in response to sensory feedback marked the second generation. These robots exhibit path control capabilities.
Third Generation: The third generation is marked with robots having human-like intelligence. The growth in computers led to high-speed processing of information and, thus robots also acquired artificial intelligence, self-learning and conclusion-drawing capabilities by past experiences. Online computations and control, artificial vision and active force interaction with the environment are the significant characteristics of these robots.
Fourth Generation: The fourth-generation robots are considered the robots of the future. These robots are called androids or humanoids, or automata with human features that mimic human actions and functions and may be capable of producing their own clones. This might provide for fifth and higher generation robots.



The following are things robots do better than humans:

• Automate manual or repetitive activities in corporate or industrial settings.
• Work in unpredictable or hazardous environments to spot hazards like gas leaks.
• Process and deliver reports for enterprise security.
• Fill out pharmaceutical prescriptions and prep IVs.
• Deliver online orders, room service and even food packets during emergencies.
• Assist during surgeries.
• Robots can also make music, monitor shorelines for dangerous predators, help with search and rescue and even assist with food preparation.

Different Types of robots

• There are as many different types of robots as there are tasks.

1. Androids

Androids are robots that resemble humans. They are often mobile, moving around on wheels or a track drive. These humanoid robots are used in areas such as care giving and personal assistance, search and rescue, space exploration and research, entertainment and education, public relations and healthcare, and manufacturing.

2. Telerobots

Telerobots are semi-autonomous robots controlled from a distance, chiefly using television, wireless networks (like Wi-Fi, Bluetooth, etc) connections. It is a combination of two major types, which are teleoperator and telepresence robot.

A **telemanipulator** (or **teleoperator**) is a remotely controlled device, usually encountered in research, academic and technical environments.

A telepresence robot simulates the experience -- and some capabilities -- of being physically present at a location. It combines remote monitoring and control via telemetry sent over radio, wires or optical fibers, and enables remote business consultations, healthcare, home monitoring, childcare and more.

3. Industrial robots

Industrial robots are automatically controlled, reprogrammable multipurpose manipulators programmable in three or more axes. Users can adapt these robots to different applications as well. Combining these robots with AI has helped businesses move them beyond simple automation to higher-level and more complex tasks.

In industrial settings, such robots can do the following:

- optimize process performance;
- automate production to increase productivity and efficiency;
- speed up product development;
- enhance safety; and
- lower costs.

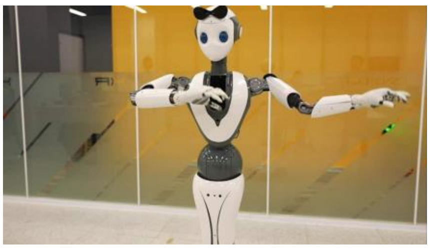
4. Swarm robot

Swarm robots (aka insect robots) work in fleets ranging from a few to thousands, all under the supervision of a single controller. These robots are analogous to insect colonies, in that they exhibit simple behaviours individually, but demonstrate behaviours that are more sophisticated with an ability to carry out complex tasks in total. They include tasks that demand miniaturization and also in search and rescue missions.

5. Smart robot

This is the most advanced kind of robot. The smart robot has a built-in AI system that learns from its environment and experiences to build knowledge and enhance capabilities to continuously improve. A smart robot can collaborate with humans and help solve problems in areas like the following:

- agricultural labor shortages;
- food waste;
- study of marine ecosystems;
- product organization in warehouses; and
- clearing of debris from disaster zones.



One type of robot commonly used in the industry is a robotic manipulator or simply a robotic arm.

- It is an open or closed kinematic chain of rigid links interconnected by movable joints.
- In some configurations, links can be considered to correspond to human anatomy as waist, upper arm, and forearm with joints at shoulder and elbow.
- At the end of the arm, a wrist joint connects an end effector to the forearm.
- The end effector may be a tool and its fixture or a gripper or any other device to do work.
- The end effector is similar to the human hand with or without fingers.