INTRODUCTION TO EMBEDDED SYSTEMS

Embedded System

An embedded system can be thought of as a computer hardware system having software embedded in it. It can be an independent system or it can be a part of a large system. An embedded system is a microcontroller or microprocessor based system which is designed to perform a specific task. For example, a fire alarm is an embedded system; it will sense only smoke.

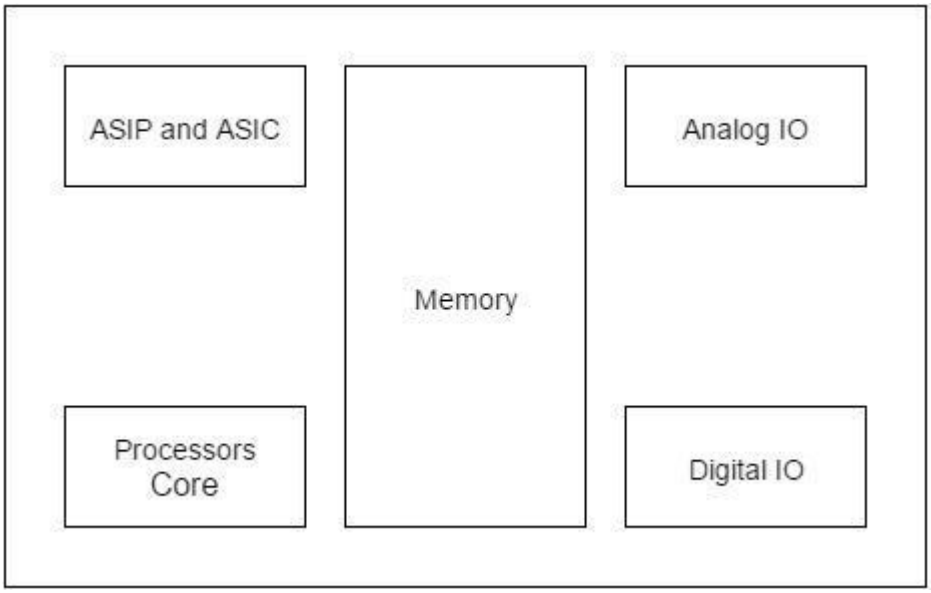
An embedded system has three components −

- It has hardware.
- It has application software.
- It has Real Time Operating System (RTOS) though a small scale embedded system may not have RTOS.

So we can define an **embedded system as a Microcontroller based, software driven, reliable, real-time control system.**

Characteristics of an Embedded System:

- **Single-functioned** − An embedded system usually repeats a specialized operation. For example: A pager always functions as a pager.
- **Tightly constrained** − All computing systems have constraints on design metrics like cost, size, power, and performance but those on an embedded system can be especially tight. Design metrics is a measure of an implementation's features. It must be of a size to fit on a single chip, must perform fast enough to process data in real time and consume minimum power to extend battery life.
- **Reactive and Real time** − Many embedded systems must continually react to changes in the system's environment and must compute certain results in real time without any delay.For example, a car cruise controller that continually monitors and reacts to speed and brake sensors. It must compute acceleration or decelerations repeatedly within a limited time; a delayed computation can result in failure to control of the car.
- **Microprocessors based** − It must be microprocessor or microcontroller based.
- **Memory** − It must have a memory, as its software usually embeds in ROM.
- **Connected** − It must have connected peripherals to connect input and output devices.
- **HW-SW systems** − Software is used for more features and flexibility. Hardware is used for performance and security.



Advantages

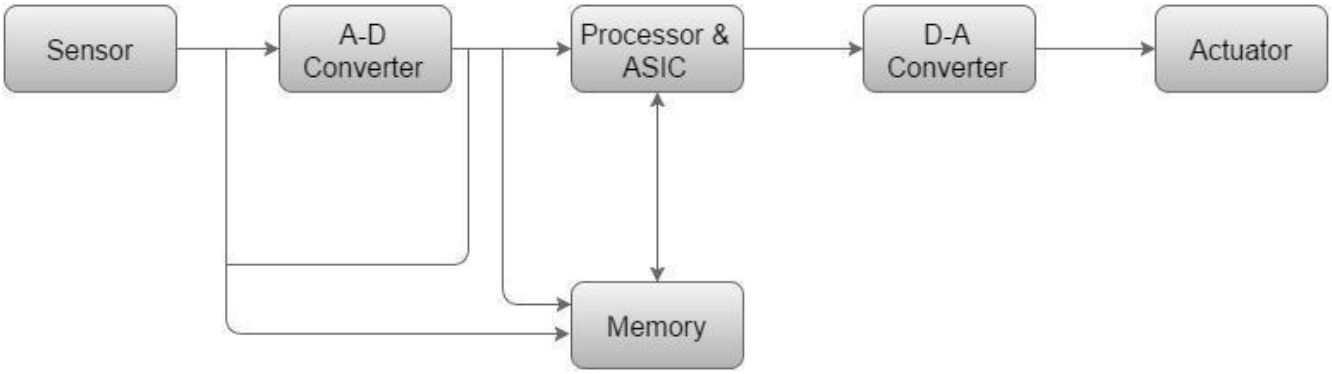
- Small size
- Low power consumption
- Low cost
- Enhanced performance
- Easily customizable

Disadvantages

- High development effort
- Larger time to market

Basic Structure of an Embedded System

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

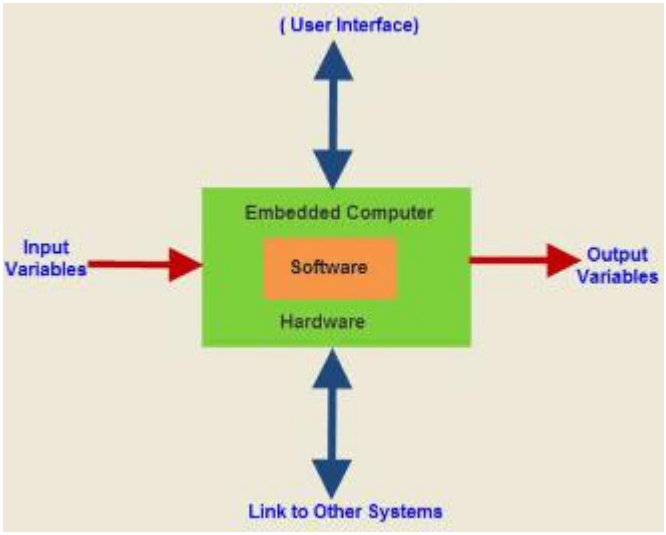


- **Sensor** − It measures the physical quantity and converts it to an electrical signal which can be read by an observer or by any electronic instrument like an A-D converter. A sensor stores the measured quantity to the memory.
- **A-D Converter** − An analog-to-digital converter converts the analog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

Difference between General purpose computer and Embedded systems:

General Purpose Computer	Embedded Systems
It is designed using a microprocessor as the main processing unit.	It is mostly designed using a microcontroller as the main processing unit.
It contains a large memory semiconductor memories like cache and RAM. It also contains secondary storage like hard disks etc.	It uses semiconductor memories but does not require secondary memories like hard disk,CD. It sometime has special memory called flash memory.
It is designed such that it can do multiple tasks as per requirement.	It is designed such that it can do a particular predefined task.
It is mostly costlier compared to the embedded systems	It is cheaper compared to a computer.
It requires huge number of peripheral devices and their controllers	It is cheaper as it requires less no of peripheral devices and their controllers are microcontroller chip itself.
The Operating system and other software for the general purpose computers, are normally complicated and occupy more memory space.	The operating system (mostly RTOS i.e Real Time Operating System) and other software occupy less memory space.

The Embedded system hardware includes elements like user interface, input/output interfaces, display and memory, etc.Generally, an embedded system comprises power supply, processor, memory, timers, serial communication ports and system application specific circuits.



Architechture of Embedded Systems:

The 8051 microcontrollers work with 8-bit data bus. So they can support external data memory up to 64K and external program memory of 64k. They can address 128k of external memory.
When data and code lie in different memory blocks, then the architecture is referred as **Harvard architecture**. In case data and code lie in the same memory block, then the architecture is referred as **Von Neumann architecture**.

Von Neumann Architecture: The Von Neumann architecture was first proposed by a computer scientist John von Neumann. In this architecture, 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 offers separate storage and signal buses for instructions and data. This architecture has data storage entirely contained within the CPU, and there is no access to the instruction storage as data. Computers have separate memory areas for program instructions and data using internal data buses, allowing simultaneous access to both instructions and data. Programs needed to be loaded by an operator; the processor could not boot itself. In a Harvard architecture, there is no need to make the two memories share properties.

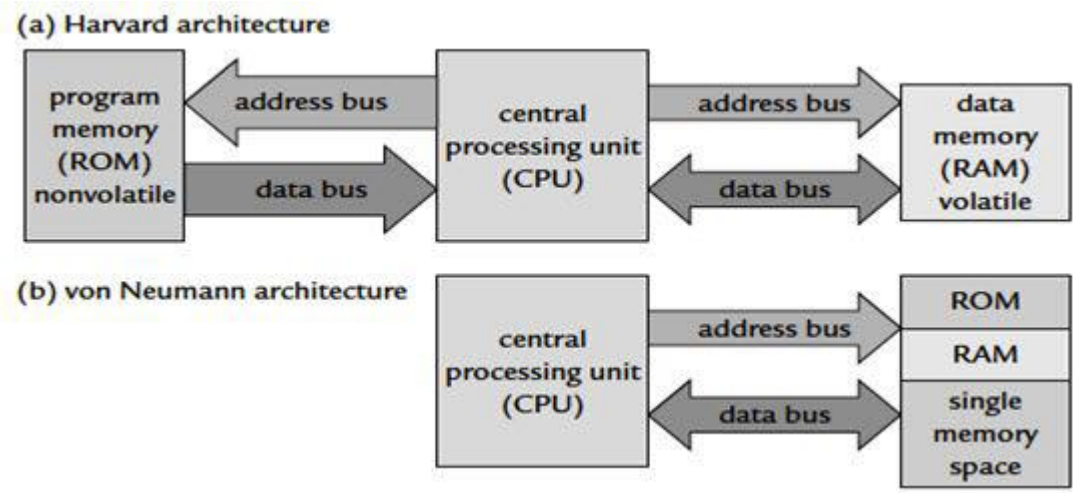


Figure: Von-Neumann and Harvard architechtures

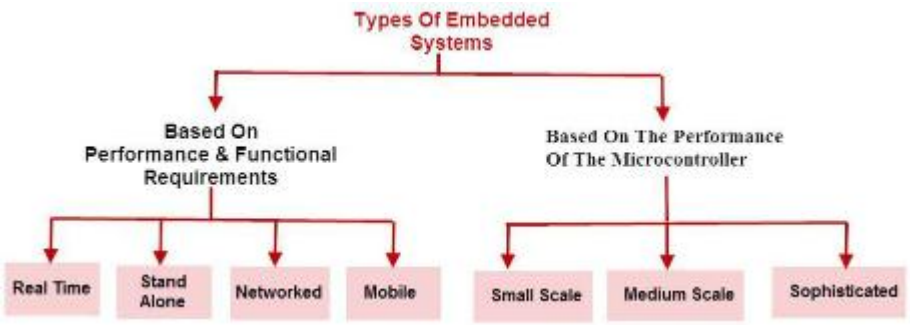
Von-Neumann Architecture vs Harvard Architecture

The following points distinguish the Von Neumann Architecture from the Harvard Architecture.

Von-Neumann Architechture	Harvard Architecture
Single memory to be shared by both code and data.	Separate memories for code and data.
Processor needs to fetch code in a separate clock cycle and data in another clock cycle. So it requires two clock cycles.	Single clock cycle is sufficient, as separate buses are used to access code and data.
Higher speed, thus less time consuming.	Slower in speed, thus more time-consuming.
Simple in design.	Complex in design.

Types of Embedded systems

Embedded systems can be classified into different types based on performance, functional requirements and performance of the microcontroller.



Embedded systems are classified into four categories based on their performance and functional requirements:

Stand alone embedded systems, Real time embedded systems ,Networked embedded systems and Mobile embedded systems.

Embedded Systems are classified into three types based on the performance of the microcontroller such as :

Small scale embedded systems, Medium scale embedded systems, Sophisticated embedded systems

Stand Alone Embedded Systems

Stand alone embedded systems do not require a host system like a computer, it works by itself. It takes the input from the input ports either analog or digital and processes, calculates and converts the data and gives the resulting data through the connected device-Which either controls, drives and displays the connected devices. Examples for the stand alone embedded systems are mp3 players, digital cameras, video game consoles, microwave ovens and temperature measurement systems.

Real Time Embedded Systems

A real time embedded system is defined as, a system which gives a required o/p in a particular time.These types of embedded systems follow the time deadlines for completion of a task. Real time embedded systems are classified into two types such as soft and hard real time systems.

Networked Embedded Systems

These types of embedded systems are related to a network to access the resources. The connected network can be LAN, WAN or the internet. The connection can be any wired or wireless. This type of embedded system is the fastest growing area in embedded system applications. The embedded web server is a type of system wherein all embedded devices are connected to a web server and accessed and controlled by a web browser. Example for the LAN networked embedded system is a home security system wherein all sensors are connected and run on the protocol TCP/IP

Mobile Embedded Systems

Mobile embedded systems are used in portable embedded devices like cell phones, mobiles, digital cameras, mp3 players and personal digital assistants, etc.The basic limitation of these devices is the other resources and limitation of memory.

Small Scale Embedded Systems

These types of embedded systems are designed with a single 8 or 16-bit microcontroller, that may even be activated by a battery. For developing embedded software for small scale embedded systems, the main programming tools are an editor, assembler, cross assembler and integrated development environment (IDE).

Medium Scale Embedded Systems

These types of embedded systems design with a single or 16 or 32 bit microcontroller, RISCs or DSPs. These types of embedded systems have both hardware and software complexities. For developing embedded software for medium scale embedded systems, the main programming tools are C, C++, JAVA, Visual C++, RTOS, debugger, source code engineering tool, simulator and IDE.

Sophisticated Embedded Systems

These types of embedded systems have enormous hardware and software complexities, that may need ASIPs, IPs, PLAs, scalable or configurable processors. They are used for cutting-edge applications that need hardware and software Co-design and components which have to assemble in the final system.

Applications of Embedded Systems:

Embedded systems are used in different applications like automobiles, telecommunications, smart cards, missiles, satellites, computer networking and digital consumer electronics.

Reference books:

1. Barrett, Embedded Systems: Design and Applications, Pearson Education.
2. Raj Kamal, Embedded Systems: Architechture, Programming and Design, Tata McGraw Hill.Mazidi,
3. Mazidi and McKinlay, The 8051 Microcontroller and Embedded Systems using assembly and C, Pearson.