

Share on your Social  
Media



# Embedded Tutorial

Published On: September 14, 2024

## Embedded Tutorial

An embedded systems profession can be a fantastic fit for you if you have a strong interest in technology and enjoy finding solutions to challenging issues. Let's explore this in detail in this embedded tutorial.

[Download Embedded Tutorial PDF](#)

## Introduction to Embedded Systems

Combining computer hardware and software with a specific purpose creates an embedded system. We cover the following in this embedded tutorial:

- Overview of Embedded Systems
- Embedded System Architecture
- Designing of an Embedded System
- Processes of Embedded Systems

## Featured Articles



Want to know  
more about  
becoming an  
expert in IT?

[Click Here to Get Started](#)

100%  
Placement  
Assurance

AUTHORISED  
CERTIFICATION  
PARTNER

IBI



Quick Enquiry

## Related Courses at SLA

- ➔ **Embedded Online Training**
- ➔ **Embedded Training in OMR**
- ➔ **Embedded Training in Chennai**

## Related Posts



- Tools and Peripherals of Embedded Systems
- Coprocessor

## Overview of Embedded Systems

An embedded system is a type of specialized computer system that is integrated into a larger mechanical or electronic system for a specific purpose. It is made up of an input/output peripheral device, computer memory, and a CPU.

**Examples of embedded systems** are video cameras, MP3 players, smart watches, temperature monitoring equipment, washing machines, and other items.

## Major Components of Embedded Systems

**Hardware:** It is a physically used part that is physically connected to an embedded system.

**Application Software:** By altering the embedded system's installed code, **application software** enables the user to run a variety of applications on an embedded system.

**Real-Time Operating System (RTOS):** an embedded system's operation is managed by the RTO. It serves as an interface between the hardware and the application software, supervising the latter and offering a scheduling mechanism that allows the processor to operate while limiting the impact of latency.

## Embedded Interview Questions and Answers

## Features of Embedded Systems

In terms of real-time system computation, let's

### Tableau Developer Salary in Chennai

Published On: October 12, 2024

Introduction A Tableau Developer designs, develops, and maintains dashboards and visualizations using Tableau software. Key...

### VMware Tutorial for Cloud Computing Aspirants

Published On: October 12, 2024

VMware Tutorial for Cloud Computing Aspirants VMware software allows you to run a virtual machine...

### Embedded Project Ideas

Published On: October 12, 2024

Introduction An Embedded Professional focuses on the design, development, and maintenance of embedded systems that...

examine the general characteristics of an embedded system:

- An embedded system is a computer **hardware and software** combination that produces a multi-application system.
- Embedded systems are utilized for specialized tasks that yield real-time results based on their many attributes.
- Through a configuration known as hardware-software intermixing, a smaller part of a larger device called an embedded system, may be used to serve a more specialized application and perform a variety of activities.
- It offers fast computing speed and great dependability.

## Advantages of Embedded Systems

- Many applications can use the same hardware.
- Reduced power use
- Reduced system operating costs
- Deliver excellent performance and efficiency.

## Limitations of Embedded Systems

- Time-taken because of the intricacy of the functions.
- Expert engineers are needed, as even one error could mean the project's destruction.

[Embedded Syllabus PDF](#)

## Embedded System Architecture

According to the Basic Design of a Computer, an ALU may execute a program and produce two different types of outputs by accepting two different types of inputs (Input Operand, Function Code).

## Types of Architecture

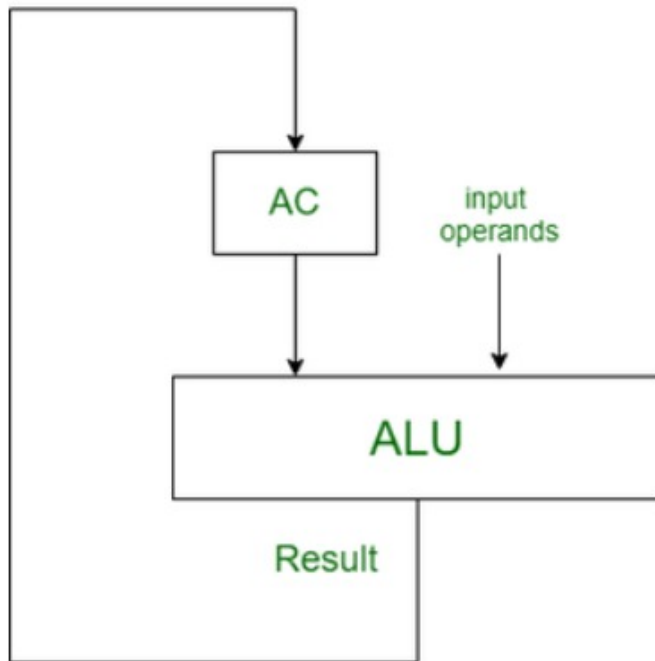
**Accumulator-Based Architecture:** In this architecture type, the register, memory, or any other

## VBA Macros Tutorial for Beginners

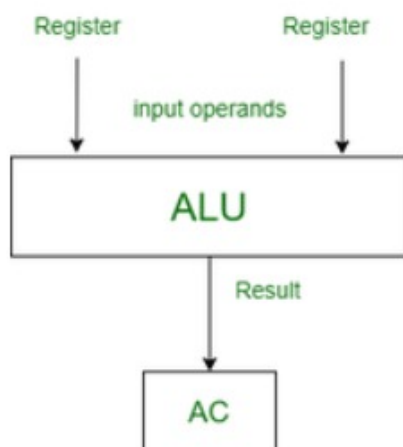
Published On: October 10, 2024

VBA Macros Tutorial for Beginners VBA macros are programs that automate repetitive operations in Microsoft...

source can provide one of the input operands for the ALU, while the other operand comes from the Accumulator (AC). Following operand receipt, the ALU operates on the operands, storing the outcome in the accumulator (AC).



**Register-Based Architecture:** In this architectural type, an ALU retrieves both operands from a register. It then operates on the operands, storing the outcome in an AC (Accumulator Capacitor). Assume that your input operands are first stored in memory and that you subsequently move them to one of the registers for execution.



**Register-Memory Architecture:** One of the input operands for the ALU in this sort of architecture comes from the register, but the other input


operand is not limited to coming from the register; it can also come from memory (i.e., it is not mandatory to take input from memory; it can be taken from either the register or memory). Following operand receipt, the ALU operates on the operands, storing the outcome in the Accumulator (AC).

**Complex System Architecture:** There is no restriction on where input operands go to the ALU in this kind of architecture. Both of the input operands can be retrieved from any location, including memory and registers. Following the operand's receipt, the ALU operates on the operands, and the outcome is stored in the Accumulator (AC).

**Stack-Based Architecture:** This kind of design involves the ALU retrieving both of its operands off the stack by simply popping the top two components. Once the operands are obtained, the ALU operates on them, and the outcome is stored in the Accumulator (AC).

## Designing of an Embedded System

Let's explore the elements, that depict an embedded system's fundamental architecture.

- **Sensor:** This type of sensor detects changes in the surrounding conditions and produces an electric signal in response to such changes. As a result, it is also known as transducers since they provide an electric input signal based on changes in the surrounding conditions.
  - **A-D Converter:** An analog-to-digital converter, or A-D converter, is a device that transforms analog electric input signals into their digital equivalents so that embedded systems can process them further.
  - **ASIC:** An integrated circuit (ASIC) that is specifically intended to carry out a single task inside an embedded system is called an application-specific integrated circuit.
  - **Processors:** The processor processes data and signals to quickly and efficiently carry out a predetermined set of instructions.
  - **Digital-to-analog converters:** D-A converters are devices that translate digital electric input signals into their analog counterparts so that embedded systems can process them further.
  - **Actuators:** Actuators are a type of comparator that provides error-free output from the system by comparing the level of the analog input signal to the desired output signal.
- 

### Design Steps of Embedded Systems

The steps involved in designing an embedded system differ from those involved in designing another type of electronic system.

Here is a flowchart that shows the stages involved in **designing an embedded system**:

### Processes of Embedded Systems

In embedded systems that process digital input from sensors and produce output in a real-time processing environment, microprocessors are essential components.

A system's processors are made up of two primary parts:

**Control Unit:** This unit in processors manages the program flow control function of the embedded system. The control unit acts as a fetching unit to retrieve the set of instructions stored in memory.

**Execution Unit:** The execution unit is responsible for carrying out the many tasks that are carried out by processors. It is primarily made up of an arithmetic and logical unit (ALU) and a circuit that runs the instruction sets that processors utilize to carry out program control operations.

### Types of Processors

An embedded system's processors fall into one of the following groups:

- **Application-Specific System Processors:**  
ASSPs are system processors that are dependent on applications and are used to process embedded system signals. Therefore, a distinct set of system processors is needed for each task that a different application performs.
- **Application-Specific Instruction Processors:**

ASIPs are processors that depend on the application. It is employed in an embedded system's combinational circuit to process the different sets of instructions.

- **General Purpose Processor (GPP):** In an embedded system, the GPP regulates the system bus, address bus, and data bus to process signals from input to output. General-purpose processor types include:
  - Microprocessor
  - Microcontroller
  - Digital Signal Processor
  - Analog Signal Processor

## Microprocessor

A microprocessor is a single VLSI chip that has a central processing unit integrated into it. It is a general-purpose gadget that can function as a microcomputer by adding more external hardware.

**ALU:** The arithmetic and logic operation is carried out by the Arithmetic and Logic Unit (ALU), a microprocessor's internal ALU. It applies the logic operation to the information kept in a register.

**Accumulator:** This is a register that stores data related to intermediate arithmetic and logical operations.

**Registers:** These are a type of storage device that a microprocessor uses to store data at various address locations.

**Program counter:** The number of programs that are run inside a microprocessor is counted using the program counter.

**Stack pointer:** The stack pointer serves as a pointer to a certain address. It is a register that holds the address of the most recent program request the processor made when a stack is in use.

**Clock circuit:** This circuit generates the clock pulse that the CPU needs as a reference signal.



**Interrupt circuit:** This circuit is used to generate an interrupt signal when a microprocessor's priority dictates that a higher-priority process must be handled first.

## Microcontroller

A single-chip microcomputer is constructed from a single semiconductor chip. Single-chip microcomputers are sometimes known as microcontrollers since they are typically employed in control applications.

## Elements of Microcontroller

A microcontroller consists of three main components: memory, I/O peripherals, and the central processor unit (CPU).

**I/O peripherals:** The processor's interface to the outside world is provided by the I/O devices. The CPU receives information from the input ports and sends it as binary data. After the processor has received the data, it instructs the output devices, which carry out duties that are not related to the microcontroller, on what to do.

**Memory:** The information that the processor receives and utilizes to react to commands that it is programmed to execute is stored in a microcontroller's memory. Two primary forms of memory exist in microcontrollers:

- **Program Memory:** This keeps track of long-term information regarding the commands that the CPU executes. Program memory, also known as non-volatile memory, holds data over time without the need for a power source.
- **Data Memory:** While the instructions are being carried out, this temporary data storage is used. The data stored in volatile data memory is only temporary since it is only retained while the device is powered on.

**Central Processing Unit:** A CPU, sometimes referred

to as a processor, serves as the device's brain. The microcontroller's operation is guided by a variety of instructions that it processes and reacts to. This entails executing fundamental I/O, logic, and arithmetic operations. Additionally, it carries out data transfer activities, which relay commands to other embedded system parts.

The following are other microcontroller-supporting components:

- **Digital to analog converter:** A circuit that transforms analog signals into digital signals is called an ADC. It enables communication between the microcontroller's central CPU and external analog devices, such as sensors.
- **Analog to digital converter:** By carrying out an ADC's opposite operation, a DAC enables the microcontroller's processor to send outgoing signals to external analog components.
- **System Bus:** The connecting wire that joins all of the microcontroller's parts is called the system bus.
- **Serial Port:** One type of I/O port that lets the microcontroller interface with external parts is the serial port. It works similarly to a parallel port or USB, except it exchanges bits in a different method.

### Challenges of Embedded

## Types of Microcontrollers:

Microcontrollers can be categorized based on their architecture and data size. Typical varieties consist of the following:

**8-bit Microcontroller:** Only eight bits of data can be transmitted at once by these MCUs. Still, when compared to greater data sizes, they use less electricity.

**16-bit Microcontroller:** Compared to 8-bit microcontrollers, these microcontrollers have faster

clock speeds and larger memory. Compared to 8-bit microcontrollers, they are two times faster.

**32-bit Microcontroller:** Compared to 16-bit microcontrollers, these high-speed microcontrollers are quicker and have more processing power. Their power usage is noticeably higher, though.

**In microcontrollers, two architectures are used:**

- **Von Neumann:** Microcontrollers featuring von Neumann designs handle memory and data through a single internal bus, allowing for one action to be performed at a time.
- **Harvard Architecture Microcontrollers:** They outperform Von Neumann's ones in terms of performance. They have several buses for processing commands and transporting data, which explains why.

## Applications of Microcontrollers

Microcontrollers are employed in the following basic areas:

- **Digital signal processors (DSPs):** To transform an incoming noisy analog signal into a consistent outgoing digital output, we can employ their ADC and DAC.
- **Household appliances:** for the operation of electromechanical devices, including televisions, video game consoles, toasters, mobile phones, and ovens.
- **Office equipment** includes smart meters, ATMs, security systems, photocopiers, scanners, fax machines, and printers.
- **More advanced uses for devices** such as **robotics**, spaceships, airplanes, and oceangoing vessels.
- **Uses in medicine** to control the functions of artificial kidneys, hearts, or other organs.

## Tools and Peripherals of Embedded Systems

Following are the tools and peripherals used in embedded systems:

## Compilers and Assemblers

**Compiler:** Computer software known as a compiler is designed to convert source code written in one programming language into binary format.

The creation of executable software is the most frequent cause of conversion. The term “compiler” generally refers to software that converts source code written in a high-level programming language into a low-level language (such as machine code or assembly).

- **Cross-Compiler:** A compiler is referred to as a cross-compiler if the program it has produced may run on a computer with a different CPU or operating system from the computer it was used to compile it.
- **Decompiler:** It is software that converts programs between low-level and high-level languages.
- **Language Converter:** A language translator, source-to-source translator, or language converter is a program that translates programs written in various high-level languages.

The following tasks are likely to be carried out by a compiler:

- Preprocessing
- Parsing
- Semantic Analysis
- Code generation
- Code optimization

**Assemblers:** Basic computer instructions, often known as assembly language, are sent into an assembler program, which then translates them into a bit pattern that the computer’s processor may utilize to carry out basic tasks.

Symbolic names are resolved to memory locations by an assembler, which also converts assembly instruction mnemonics into opcodes of object code. Every machine operation at the low level (opcode) in assembly language is represented by a mnemonic.

## Embedded Training

### Debugging Tools in an Embedded System

To make a computer program or a piece of electronic hardware function as intended, debugging is the systematic process of identifying and minimizing defects in these systems.

Tightly connected subsystems make debugging challenging since even minor changes to one subsystem might introduce bugs into another.

The debugging features and development times of the various debugging tools used in embedded systems vary significantly.

The following debugging tools will be covered here:

- Simulators
- Microcontroller starter kits
- Emulator

**Simulators:** By mimicking the MCU or system on the host computer used for code development, code is tested for it. Simulators attempt to create a software model of the entire microcontroller's activity.

### Functions of Simulators

The following tasks are carried out by a simulator:

- Identifies the target system's processor or processing device family and all of its variants.
- Keeps track of the specific details of a source code segment using labels and symbolic parameters during each step of the execution

process.

- Gives the target system's simulated ports and RAM status for every single step execution.
- Tracks the reaction of the system and calculates throughput.
- Provide a trace of the program counter's output with the processor registers.
- Explains the current command's meaning in detail.
- Keeps track of the specifics of the simulator commands as they are chosen from a menu or input via the keyboard.
- Supports unconditional breakpoints and conditions (up to 8, 16, or 32 conditions).
- Gives access to breakpoints and the trace, two crucial testing and debugging tools.
- Helps to synchronize internal delays and peripherals.

**Microcontroller Starter Kit:** A microcontroller starter kit includes:

- Hardware board or Evaluation board)
- In-system programmer
- Conversion, assembling, linking, and other software tools.
- An IDE and code size may restrict the compiler's evaluation version.

This kit's ability to operate in real-time makes input/output functionality verification simple, which is a major advantage over simulators. However, starter kits are the most affordable and fully functional choice for creating basic microcontroller projects.

**Emulators:** An emulator is a piece of hardware, software, or both that mimics the operations of one computer system (the guest) in another computer system (the host) so that the behavior of the simulated system is almost the same as that of the real system (the guest).

- A computer program's capacity to replicate, or

mimic, another program or device is referred to as emulation.

- Replicating the original computer environment is the main goal of emulation.
- Emulators can sustain a stronger link with the genuineness of the digital asset.
- Working on any kind of program or operating system on a platform in a manner comparable to how the software operates in its native environment is made easier for the user by an emulator.

## Peripheral Devices of Embedded Systems

Through their peripherals, embedded systems can interact with the outside world by following &min;

- **Interfaces for serial communication**, such as RS-232, RS-422, RS-485, etc.
- **Universal Serial Bus (USB)**: I2C, SPI, SSC, and ESSI are examples of synchronous serial communication interfaces.
- **Multimedia Cards** (SD Cards, Compact Flash, etc.)
- **Networks** such as LonWorks, Ethernet, etc.
- **Field buses**: PROFIBUS, LIN-Bus, CAN-Bus, and so forth.
- **Components** such as time processing units, capture/compare, and PLL(s).
- **Discrete IO** refers to General Purpose Input/Output (GPIO) and includes JTAG, ISP, ICSP, BDM Port, BITP, and DP9 ports for analog to digital/digital to analog (ADC/DAC) debugging.

When selecting a microcontroller, consider the following factors:

- **Speed**: What is the maximum speed that the microcontroller is capable of handling?
- **Packaging**: Is it a quad flat package (QFP) or a 40-pin DIP (dual-inline package)? This matters for the final product's space, assembly, and prototype.

- **Power Consumption:** This is a crucial factor for items that run on batteries.
- The chip's RAM and ROM capacities.
- The chip's I/O pin and timing count.
- **Cost per Unit:** This factor has a significant impact on the end product cost, which is where the microcontroller will be employed.

Additionally, confirm that the microcontroller comes with tools like assemblers, debuggers, and compilers. Above all, make sure you buy a microcontroller from a reputable supplier.

### **Embedded Project Ideas**

## **Coprocessor**

If a microprocessor chip has extra circuitry, it can be added specifically to carry out certain tasks or to manipulate numbers to relieve the main CPU of some of its workload. The CPU can then operate more quickly.

**Coprocessor:** It is also referred to as a math processor. The core processor is relieved of this computation and its time is saved as the coprocessor completes standard mathematical operations.

- Coprocessors free up the main microprocessor to perform at a faster pace by taking on specialized processing jobs from the core CPU.
- Specialized operations like intricate mathematical calculations and graphical display processing can be carried out by a coprocessor.
- They work more quickly on certain tasks than a core CPU. The system's overall computer speed rises as a result.
- We can attach the coprocessor to an ARM processor. When a coprocessor is added, the core CPU's instruction set must be expanded, or configurable registers must be added, to boost processing capability.



- A handful of coprocessors can be attached to the ARM CPU via the coprocessor interface.

Coprocessors help the system operate more smoothly by delegating some of the CPU's workload. They can be:

**Independent Type:** It operates in tandem with the CPU asynchronously. When it executes any kind of task or computation, it is not in sync with the CPU. Here, the co-processor is capable of acting independently.

This frees the CPU from having to worry about the work of its coprocessors. The CPU that initiated the request is freed up to work on another task rather than waiting synchronously.

**Direct Control Type:** They are floating-point units and managed by coprocessor instructions, which are a subset of the CPU instruction set. The CPU that sent the request waits or verifies that the co-processor has finished its work.

Here, the coprocessor is managed or under the direct authority of the primary CPU and is not autonomous. Here, the CPU and coprocessor operate in synchronization.

## Properties of Coprocessor

- A primary microprocessor is necessary for the coprocessor to operate.
- The primary processor must recognize and separate instructions in a program that requires a lot of calculation.
- Coprocessors carry out instructions involving large amounts of computation.
- Every other task is managed by the primary processor.

## Functionalities of Coprocessor

- Coprocessors cannot perform I/O operations, manage memory, execute instructions (flow

control kinds) directly, and retrieve instructions from memory.

- To retrieve coprocessor instructions and handle all other tasks unrelated to the coprocessor, the coprocessor depends on the host, or main, processor.
- A coprocessor is not the system's primary processor.
- The coprocessor is capable of carrying out various tasks like signal, string, graphical processing, encryption/decryption, and floating point arithmetic operations like addition, subtraction, multiplication, and calculating the square root of a given number or estimating its logarithmic value.
- Coprocessors allow for computer customization, so if a client decides they don't need the extra performance, they won't have to pay.

## Conclusion

We have covered the basics of embedded system concepts in this embedded system tutorial. There are many more concepts to learn to become an embedded developer. Join us to explore them comprehensively with hands-on exposure in our [\*\*embedded system training in Chennai.\*\*](#)

Share on your Social  
Media



## Navigation

---

About Us  
Blog Posts  
Careers  
Contact  
Placement Training

# Softlogic Systems

## KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai  
– 600 078.

**Landmark:** Karnataka Bank Building

**Phone:** [+91 86818 84318](tel:+918681884318)

**Email:** [enquiry@softlogicsys.in](mailto:enquiry@softlogicsys.in)

**Map:** [Google Maps Link](#)

## OMR

No. E1-A10, RTS Food Street  
92, Rajiv Gandhi Salai (OMR),  
Navalur, Chennai – 600 130.

**Landmark:** Adj. to AGS Cinemas

**Phone:** [+91 89256 88858](tel:+918925688858)

**Email:** [info@softlogicsys.in](mailto:info@softlogicsys.in)

**Map:** [Google Maps Link](#)

## Courses

Python

Software Testing

Full Stack Developer

Java

Power BI

Clinical SAS

Data Science

Embedded

Cloud Computing

Hardware and Networking

VBA Macros

Mobile App Development

DevOps

Corporate Training

Hire With Us

Job Seekers

SLA's Recently Placed Students

Reviews

Sitemap

## Important Links

Disclaimer

Privacy Policy

Terms and Conditions

## Social Media Links



## Review Sources

Google

Trustpilot

Glassdoor

Mouthshut

Sulekha

Justdial

Ambitionbox

Indeed

Software Suggest

Sitejabber