IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

An Embedded System

Sakshi Ramavatar Yadav, Nikhil Deepak Kalokar, Rohit Dashrath Ade, Amit Prakash Kasambe
Vineet Shivkumar Patale, Anjali Parmeshwar Yewale
Jawaharlal Darda Institute of Engineering and Technology, Yavatmal, India
yadavsakshi65008@gmail.com, nikhi.kalokar123@gmail.com, rohitade9712@gmail.com
amitkasambe4848@gmail.com, vineetpatale@gmail.com, yewaleanjali03@gmail.com

Abstract: *An embedded system is a programmed controlling and operating system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. Embedded systems control many devices in common use today.[3] Ninety-eight percent of all microprocessors are manufactured as components of embedded systems. Examples of properties of typical embedded computers when compared with general-purpose counterparts are low power consumption, small size, rugged operating ranges, and low per-unit cost. This comes at the price of limited processing resources, which make them significantly more difficult to program and to interact with. However, by building intelligence mechanisms on top of the hardware, taking advantage of possible existing sensors and the existence of a network of embedded units, one can both optimally manage available resources at the unit and network levels as well as provide augmented functions, well beyond those available. For example, intelligent techniques can be designed to manage power consumption of embedded systems. Modern embedded systems are often based on microcontrollers (i.e. CPUs with integrated memory or peripheral interfaces),[7] but ordinary microprocessors (using external chips for memory and peripheral interface circuits) are also common, especially in more-complex systems. In either case, the processor(s) used may be types ranging from general purpose to those specialized in certain class of computations, or even custom designed for the application at hand. A common standard class of dedicated processors is the digital signal processor (DSP).*

Keywords: embedded, digital signal processor

I. INTRODUCTION

Embedded System is an integrated system that is formed as a combination of computer hardware and software for a specific function. It can be said as a dedicated computer system has been developed for some particular reason. But it is not our traditional computer system or general-purpose computers, these are the Embedded systems that may work independently or attached to a larger system to work on a few specific functions. These embedded systems can work without human intervention or with little human intervention.

Three main components of Embedded systems are:

1. Hardware
2. Software
3. Firmware

Some examples of embedded systems:

- Digital watches
- Washing Machine
- Toys
- Televisions
- Digital phones
- Laser Printer
- Cameras
- Industrial machines

- Electronic Calculators
- Automobiles
- Medical Equipment

Application areas of Embedded System:

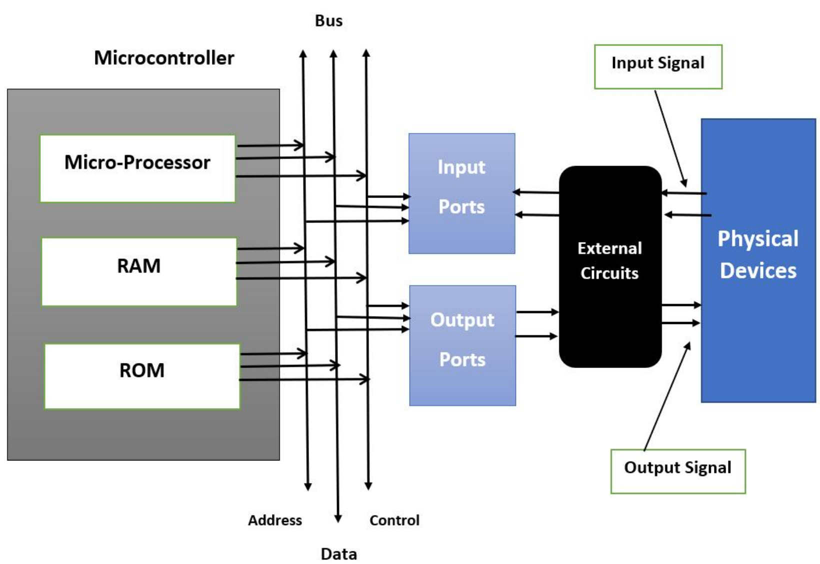
Mostly Embedded systems are present everywhere. We use it in our everyday life unknowingly as in most cases it is integrated into the larger systems. So, here are some of the application areas of Embedded systems:

- Home appliances
- Transportation
- Health care
- Business sector & offices
- Defense sector
- Aerospace
- Agricultural Sector

Important Characteristics of an Embedded System:

- Performs specific task: Embedded systems perform some specific function or tasks.
- Low Cost: The price of an embedded system is not so expensive.
- Time Specific: It performs the tasks within a certain time frame.
- Low Power: Embedded Systems don't require much power to operate.
- High Efficiency: The efficiency level of embedded systems is so high.
- Minimal User interface: These systems require less user interface and are easy to use.
- Less Human intervention: Embedded systems require no human intervention or very less human intervention.
- Highly Stable: Embedded systems do not change frequently mostly fixed maintaining stability.
- High Reliability: Embedded systems are reliable they perform tasks consistently well.
- Use microprocessors or microcontrollers: Embedded systems use microprocessors or microcontrollers to design and use limited memory.
- Manufacturable: The majority of embedded systems are compact and affordable to manufacture. They are based on the size and low complexity of the hardware.

Block Structure Diagram of Embedded System:



Embedded System

Advantages of Embedded System:

- Small size.
- Enhanced real-time performance.
- Easily customizable for a specific application.

Disadvantages of Embedded System:

- High development cost.
- Time-consuming design process.
- As it is application-specific less market available.

Top Embedded Programming Languages: Embedded systems can be programmed using different programming languages like Embedded C, Embedded C++, Embedded Java, and Embedded Python. However, it entirely depends on the developer to use which programming language for the development of the embedded systems

While embedded systems are computing systems, they can range from having no user interface (UI) -- for example, on devices designed to perform a single task -- to complex graphical user interfaces (GUIs), such as in mobile devices. User interfaces can include buttons, LEDs (light-emitting diodes) and touchscreen sensing. Some systems use remote user interfaces as well.

Markets and Markets, a business-to-business (B2B) research firm, predicted that the embedded market will be worth $116.2 billion by 2025. Chip manufacturers for embedded systems include many well-known technology companies, such as Apple, IBM, Intel and Texas Instruments. The expected growth is partially due to the continued investment in artificial intelligence (AI), mobile computing and the need for chips designed for high-level processing.

Examples of embedded systems

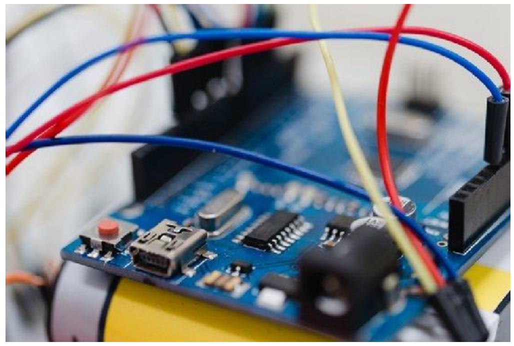
Embedded systems are used in a wide range of technologies across an array of industries. Some examples include:

- Automobiles. Modern cars commonly consist of many computers (sometimes as many as 100), or embedded systems, designed to perform different tasks within the vehicle. Some of these systems perform basic utility functions and others provide entertainment or user-facing functions. Some embedded systems in consumer vehicles include cruise control, backup sensors, suspension control, navigation systems and airbag systems.
- Mobile phones. These consist of many embedded systems, including GUI software and hardware, operating systems (OSes), cameras, microphones, and USB (Universal Serial Bus) I/O (input/output) modules.
- Industrial machines. They can contain embedded systems, like sensors, and can be embedded systems themselves. Industrial machines often have embedded automation systems that perform specific monitoring and control functions.
- Medical equipment. These may contain embedded systems like sensors and control mechanisms. Medical equipment, such as industrial machines, also must be very user-friendly so that human health isn't jeopardized by preventable machine mistakes. This means they'll often include a more complex OS and GUI designed for an appropriate UI.

How does an embedded system work?

Embedded systems always function as part of a complete device -- that's what's meant by the term *embedded*. They are low-cost, low-power-consuming, small computers that are embedded in other mechanical or electrical systems. Generally, they comprise a processor, power supply, and memory and communication ports. Embedded systems use the communication ports to transmit data between the processor and peripheral devices -- often, other embedded systems -- using a communication protocol. The processor interprets this data with the help of minimal software stored on the memory. The software is usually highly specific to the function that the embedded system serves.

IJARSCT ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53 Volume 4, Issue 2, February 2024



YASINEMIR/STOCK.ADOBE.COM

A photo of an embedded system motherboard with attached cables.

The processor may be a microprocessor or microcontroller. Microcontrollers are simply microprocessors with peripheral interfaces and integrated memory included. Microprocessors use separate integrated circuits for memory and peripherals instead of including them on the chip. Both can be used, but microprocessors typically require more support circuitry than microcontrollers because there is less integrated into the microprocessor. The term *system on a chip (SoC)* is often used. SoCs include multiple processors and interfaces on a single chip. They are often used for high-volume embedded systems. Some example SoC types are the application-specific integrated circuit (ASIC) and the field-programmable gate array (FPGA).

Often, embedded systems are used in real-time operating environments and use a real-time operating system (RTOS) to communicate with the hardware. Near-real-time approaches are suitable at higher levels of chip capability, defined by designers who have increasingly decided the systems are generally fast enough and the tasks tolerant of slight variations in reaction. In these instances, stripped-down versions of the Linux operating system are commonly deployed, although other OSes have been pared down to run on embedded systems, including Embedded Java and Windows IoT (formerly Windows Embedded).

Characteristics of embedded systems

The main characteristic of embedded systems is that they are task-specific.

Additionally, embedded systems can include the following characteristics:

- typically, consist of hardware, software and firmware;
- can be embedded in a larger system to perform a specific function, as they are built for specialized tasks within the system, not various tasks;
- can be either microprocessor-based or microcontroller-based -- both are integrated circuits that give the system compute power;
- are often used for sensing and real-time computing in internet of things (IoT) devices, which are devices that are internet-connected and do not require a user to operate;
- can vary in complexity and in function, which affects the type of software, firmware and hardware they use; and
- are often required to perform their function under a time constraint to keep the larger system functioning properly.

Structure of embedded systems

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

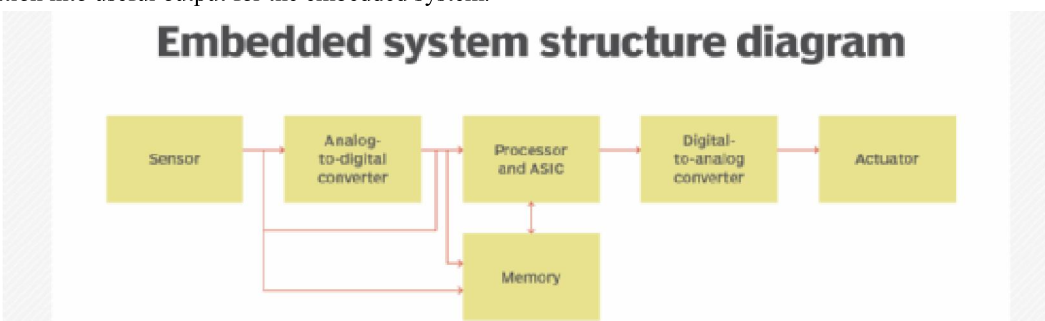
- Hardware. The hardware of embedded systems is based around microprocessors and microcontrollers. Microprocessors are very similar to microcontrollers and, typically, refer to a CPU (central processing unit) that is integrated with other basic computing components such as memory chips and digital signal processors (DSPs). Microcontrollers have those components built into one chip.

- Software and firmware. Software for embedded systems can vary in complexity. However, industrial-grade microcontrollers and embedded IoT systems usually run very simple software that requires little memory.
- Real-time operating system. These are not always included in embedded systems, especially smaller-scale systems. RTOSes define how the system works by supervising the software and setting rules during program execution.

In terms of hardware, a basic embedded system would consist of the following elements:

- Sensors convert physical sense data into an electrical signal.
- Analog-to-digital (A-D) converters change an analog electrical signal into a digital one.
- Processors process digital signals and store them in memory.
- Digital-to-analog (D-A) converters change the digital data from the processor into analog data.
- Actuators compare actual output to memory-stored output and choose the correct one.

The sensor reads external inputs, the converters make that input readable to the processor, and the processor turns that information into useful output for the embedded system.



A diagram of the basic structure and flow of information in embedded systems.

Types of embedded systems

There are a few basic embedded system types, which differ in their functional requirements. They are:

- Mobile embedded systems are small-sized systems that are designed to be portable. Digital cameras are an example of this.
- Networked embedded systems are connected to a network to provide output to other systems. Examples include home security systems and point of sale (POS) systems.
- Standalone embedded systems are not reliant on a host system. Like any embedded system, they perform a specialized task. However, they do not necessarily belong to a host system, unlike other embedded systems. A calculator or MP3 player is an example of this.
- Real-time embedded systems give the required output in a defined time interval. They are often used in medical, industrial and military sectors because they are responsible for time-critical tasks. A traffic control system is an example of this.

Embedded systems can also be categorized by their performance requirements:

- Small-scale embedded systems often use no more than an 8-bit microcontroller.
- Medium-scale embedded systems use a larger microcontroller (16-32 bit) and often link microcontrollers together.
- Sophisticated-scale embedded systems often use several algorithms that result in software and hardware complexities and may require more complex software, a configurable processor and/or a programmable logic array.

There are several common embedded system software architectures, which become necessary as embedded systems grow and become more complex in scale. These include:

- Simple control loops call subroutines, which manage a specific part of the hardware or embedded programming.

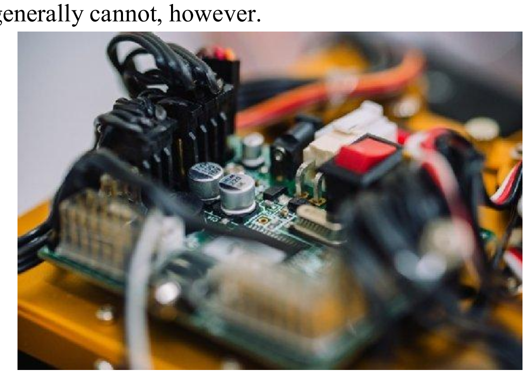
- Interrupt controlled systems have two loops: a main one and a secondary one. Interruptions in the loops trigger tasks.
- Cooperative multitasking is essentially a simple control loop located in an application programming interface (API).
- Preemptive multitasking or multithreading is often used with an RTOS and features synchronization and task switching strategies.

Very large-scale integration, or *VLSI*, is a term that describes the complexity of an integrated circuit (IC). VLSI is the process of embedding hundreds of thousands of transistors into a chip, whereas LSI (large-scale integration) microchips contain thousands of transistors, MSI (medium-scale integration) contains hundreds of transistors, and SSI (small-scale integration) contains tens of transistors. ULSI, or ultra-large-scale integration, refers to placing millions of transistors on a chip.

VLSI circuits are common features of embedded systems. Many ICs in embedded systems are VLSIs, and the use of the VLSI acronym has largely fallen out of favor.

Debugging embedded systems

One area where embedded systems part ways with the operating systems and development environments of other larger-scale computers is in the area of debugging. Usually, developers working with desktop computer environments have systems that can run both the code being developed and separate debugger applications that can monitor the embedded system programmers generally cannot, however.



YASINEMIR/STOCK.ADOBE.COM

Macro photo of a little embedded system motherboard with attached cables.

Some programming languages run on microcontrollers with enough efficiency that rudimentary interactive debugging is available directly on the chip. Additionally, processors often have CPU debuggers that can be controlled -- and, thus, control program execution -- via a JTAG or similar debugging port.

In many instances, however, programmers need tools that attach a separate debugging system to the target system via a serial or other port. In this scenario, the programmer can see the source code on the screen of a general-purpose computer, just as would be the case in the debugging of software on a desktop computer. A separate, frequently used approach is to run software on a PC that emulates the physical chip in software. This is essentially making it possible to debug the performance of the software as if it were running on an actual physical chip.

Broadly speaking, embedded systems have received more attention to testing and debugging because a great number of devices using embedded controls are designed for use, especially in situations where safety and reliability are top priorities.

History of embedded systems

Embedded systems date back to the 1960s. Charles Stark Draper developed an integrated circuit in 1961 to reduce the size and weight of the Apollo Guidance Computer, the digital system installed on the Apollo Command Module and Lunar Module. The first computer to use ICs, it helped astronauts collect real-time flight data.

In 1965, Autonetics, now a part of Boeing, developed the D-17B, the computer used in the Minuteman I missile guidance system. It is widely recognized as the first mass-produced embedded system. When the Minuteman II went into production in 1966, the D-17B was replaced with the NS-17 missile guidance system, known for its high-volume use of integrated circuits. In 1968, the first embedded system for a vehicle was released; the Volkswagen 1600 used a microprocessor to control its electronic fuel injection system.

By the late 1960s and early 1970s, the price of integrated circuits dropped and usage surged. The first microcontroller was developed by Texas Instruments in 1971. The TMS1000 series, which became commercially available in 1974, contained a 4-bit processor, read-only memory (ROM) and random-access memory (RAM), and it cost around $2 apiece in bulk orders.

Also, in 1971, Intel released what is widely recognized as the first commercially available processor, the 4004. The 4-bit microprocessor was designed for use in calculators and small electronics, though it required eternal memory and support chips. The 8-bit Intel 8008, released in 1972, had 16 KB of memory; the Intel 8080 followed in 1974 with 64 KB of memory. The 8080's successor, the x86 series, was released in 1978 and is still largely in use today.

In 1987, the first embedded operating system, the real-time VxWorks, was released by Wind River, followed by Microsoft's Windows Embedded CE in 1996. By the late 1990s, the first embedded Linux products began to appear. Today, Linux is used in almost all embedded devices.

Embedded system trends

While some embedded systems can be relatively simple, they are becoming more complex, and more and more of them are now able to either supplant human decision-making or offer capabilities beyond what a human could provide. For instance, some aviation systems, including those used in drones, are able to integrate sensor data and act upon that information faster than a human could, permitting new kinds of operating features.

The embedded system is expected to continue growing rapidly, driven in large part by the internet of things. Expanding IoT applications, such as wearables, drones, smart homes, smart buildings, video surveillance, 3D printers and smart transportation, are expected to fuel embedded system growth.

REFERENCES

[1]. https://en.wikipedia.org/wiki/Embedded_system
[2]. https://www.elprocus.com/basics-of-embedded-system-and- applications
[3]. https://www.quora.com/What-is-an-embedded-system