Embedded Systems (CMPE 30247 Module 1)

Mr. Adelino T. Racusa

Objectives:

At the end of this lesson, students will be able to :

- Understand the fundamentals or basic concepts of embedded systems.
- Be familiar in the different components of embedded systems.
- Discuss the different applications of an embedded system.

What is an Embedded System?

Which statement best defines an embedded system?

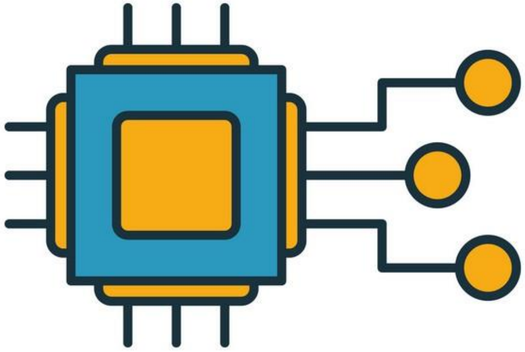
a) A device used for general-purpose computing.
b) A dedicated system designed to perform a specific task within a larger device.
c) A computer with high processing power for complex tasks.
d) A system that can run multiple applications simultaneously.

Which statement best defines an embedded system?

a) A device used for general-purpose computing.
b) A dedicated system designed to perform a specific task within a larger device.
c) A computer with high processing power for complex tasks.
d) A system that can run multiple applications simultaneously.

Answer : B

Embedded Systems



Embedded system is a computer system designed to perform specific functions within a larger mechanical or electrical system.

It is embedded, or integrated, into a larger device or product, and its primary purpose is to control and monitor the operation of that device or system.

Combination of hardware and software designed to do specific function or functions

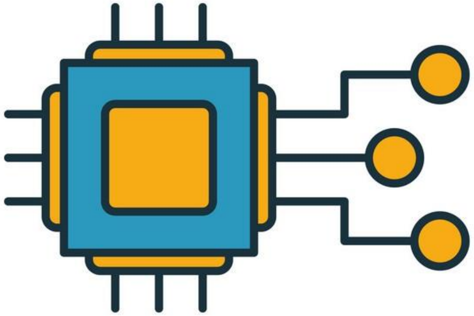
Typically small, low-cost, and low-power.
Used in real-time applications

Which of the following is an application of Embedded Systems?
Pick one and tell us why?

L/N	Device	YES	NO
1	ASUS Laptop		✓
2	Digital Camera	✓	
3	Smart Phones	✓	
4	Smart Television	✓	
5	Washing Machine	✓	
6	Desktop Computers		✓
7	Microwave Oven	✓	
8	Garmin GPS	✓	
9	Electrocardiogram (ECG) Monitor	✓	
10	Pacemakers	✓	

L/N	Device	YES	NO
1	ASUS Laptop		✓
2	Digital Camera	✓	
3	Smart Phones	✓	
4	Smart Television	✓	
5	Washing Machine	✓	
6	Desktop Computers		✓
7	Microwave Oven	✓	
8	Garmin GPS	✓	
9	Electrocardiogram (ECG) Monitor	✓	
10	Pacemakers	✓	

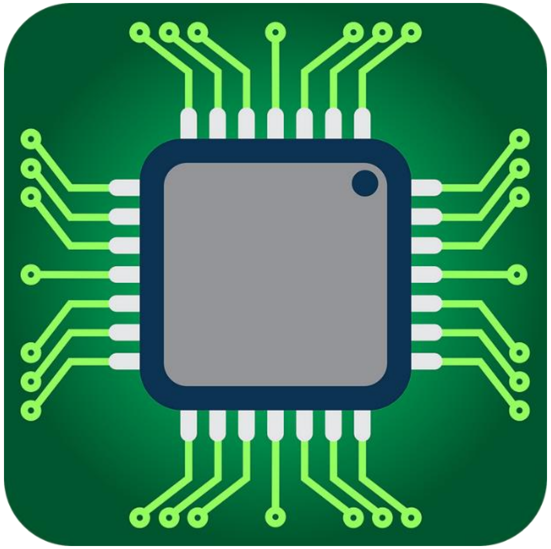
Microcontroller



- A microcontroller is a small computer on a single integrated circuit (IC) that is designed to perform specific tasks and control electronic systems. It consists of a central processing unit (CPU), memory, and input/output peripherals, all integrated onto a single chip.

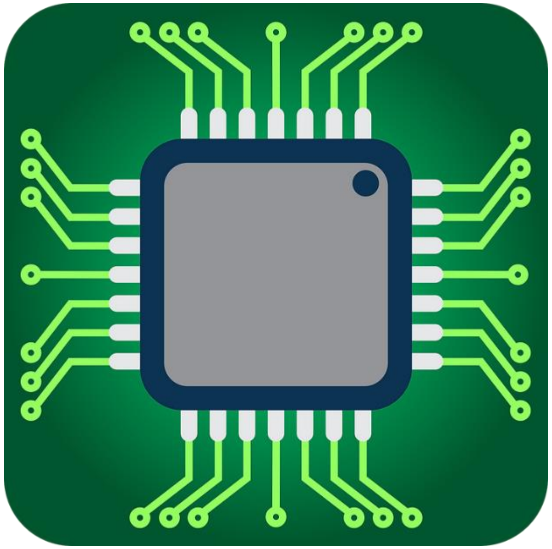
- Microcontrollers are commonly used in embedded systems, which are computer systems designed for specific applications or tasks. They are found in a wide range of devices, including consumer electronics, automotive systems, medical devices, industrial control systems, and more.

What do you think is the difference between microcontroller and microprocessor?



The main difference between a **microcontroller** and a **microprocessor** is that

a **microcontroller** has all the essential components of a computer system integrated onto a single chip,



while a **microprocessor** is a central processing unit (CPU) that requires external components such as memory and I/O devices to function.

Feature	Microcontroller (MCU)	Microprocessor (MPU)
Integration	CPU, memory, and peripherals are integrated into a single chip	Only CPU on the chip; requires external memory and I/O
Functionality	Designed for specific control tasks in embedded systems	Designed for general-purpose computing
Complexity	Simple, compact, and cost-effective	More complex and requires additional components
Power Consumption	Low power consumption, ideal for battery-powered devices	Higher power consumption due to external components
Speed and Performance	Lower speed, optimized for real-time control	Higher speed, optimized for complex computations

Provide examples of specific scenarios where one might be preferred over the other.
Explain the reasons behind this preferences.

Scenarios

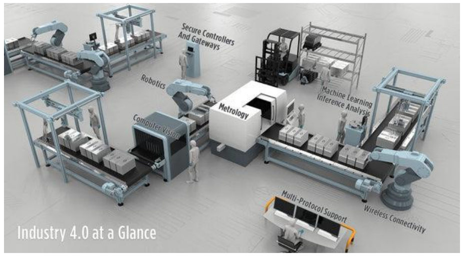
Application	Preferred Type	Reason for Preference
Home Automation System	MCU	Compact, low power, dedicated functionality
Personal Computer	MPU	Versatile, high performance, supports complex tasks
Wearable Health Monitor	MCU	Long battery life, real-time control, compact design
Server Infrastructure	MPU	High processing power, multitasking capability
Automotive Control Systems	MCU	Reliable, low power, designed for specific tasks
Gaming Console	MPU	High speed, complex computations, large memory support
Industrial Automation	MCU	Cost-effective, simple design for specific applications
Cloud Computing	MPU	Scalability, external memory support, multitasking

Feature	Microcontroller (MCU)	Microprocessor (MPU)
Integration	CPU, memory, and peripherals are integrated into a single chip	Only CPU on the chip; requires external memory and I/O
Functionality	Designed for specific control tasks in embedded systems	Designed for general-purpose computing
Complexity	Simple, compact, and cost-effective	More complex and requires additional components
Power Consumption	Low power consumption, ideal for battery-powered devices	Higher power consumption due to external components
Speed and Performance	Lower speed, optimized for real-time control	Higher speed, optimized for complex computations
Memory	Limited internal RAM, ROM/Flash	Requires external RAM, ROM, and storage
Input/Output (I/O)	Built-in GPIO, ADC, timers, and communication interfaces	Requires external I/O devices
Programming	Programmed to perform a specific task	Can run various applications and operating systems
Cost	Low cost due to integrated components	Higher cost due to external components
Applications	Used in embedded systems: washing machines, cameras, IoT	Used in computers, smartphones, servers, and tablets

Applications of Embedded Systems

Traffic Management

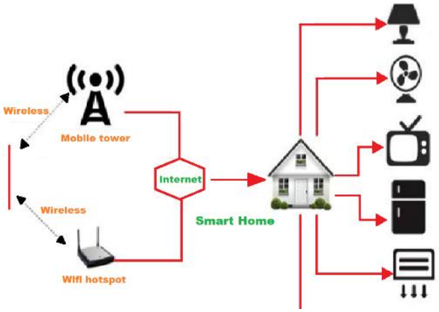
Airplane Traffic Controller



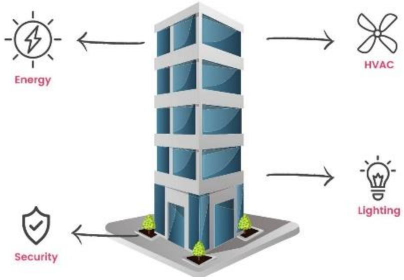
Industrial Automation

Location Mapping

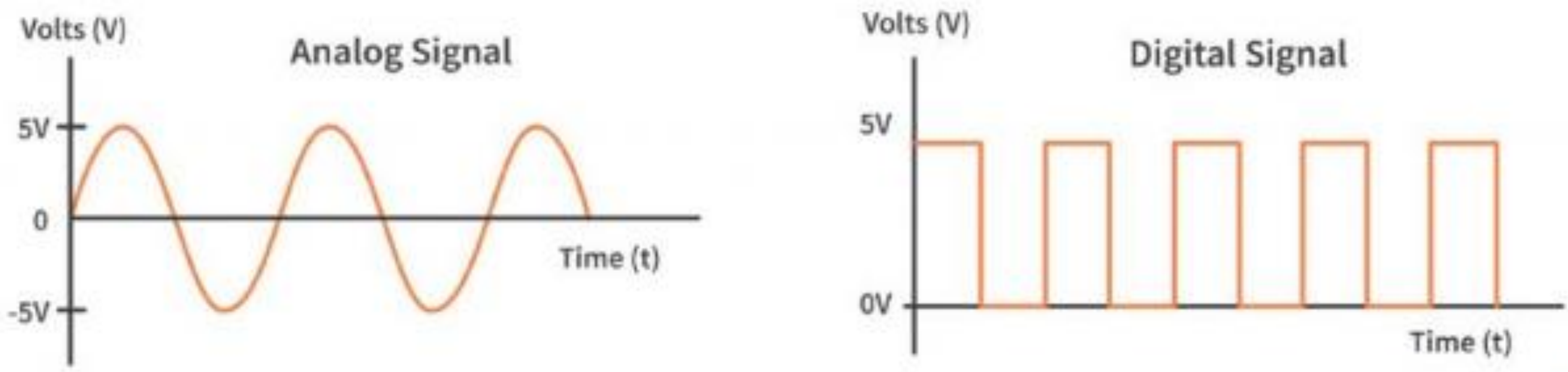
Medical Care



Consumer Electronics



Building Management System


Volts (V)
Analog Signal
5V
0
-5V
Time (t)
Volts (V)
Digital Signal
5V
0V
Time (t)

What is the relationship of analog and digital signals to embedded systems?

What are the implications of the conversion of signals to the signal integrity and system performance in real time applications?

Implications of Conversion on Signal Integrity

Noise and Distortion

Analog signals can be affected by noise and distortion during transmission. If not adequately filtered or conditioned before conversion, these issues can degrade the quality of the digital signal.

Quantization Error

The process of quantization introduces errors, as the continuous analog signal is approximated to discrete levels. This quantization noise can affect the accuracy and integrity of the digital representation of the signal

Implications on System Performance in Real-Time Applications

Processing Speed

The efficiency of an embedded system in converting and processing signals can significantly impact its performance. High-speed ADCs and optimized algorithms are necessary for real-time applications, such as robotics or automotive systems.

Latency

The conversion process can introduce latency. In applications requiring immediate feedback (e.g., control systems), delays can lead to performance degradation or system instability.

Implications on System Performance in Real-Time Applications

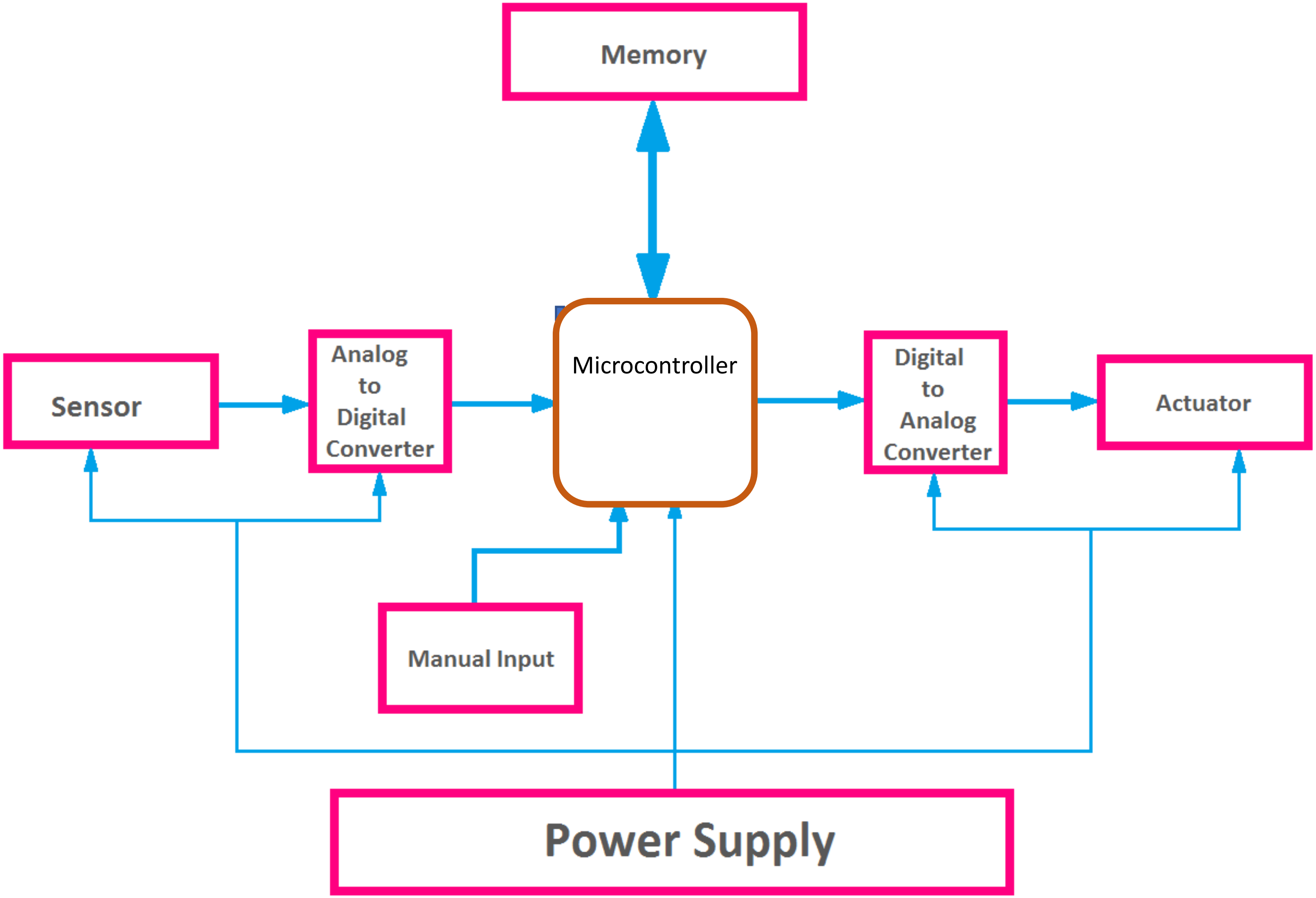
Resource Constraints

Embedded systems often operate under constraints like limited processing power, memory, and energy resources. Efficient algorithms and hardware designs are crucial to ensure that the system performs optimally without exceeding these limits.

Reliability

In critical applications (e.g., medical devices), the integrity of the signal used in decision-making is paramount. Poor signal quality can lead to incorrect readings and potentially dangerous outcomes.

Block Diagram of Embedded Systems



Block Diagram of Embedded Systems Flow

Sensors: The input to the embedded system is typically obtained from sensors.

Sensors detect and measure physical quantities such as temperature, pressure, light intensity, or motion.

Block Diagram of Embedded Systems Flow

Analog-to-Digital Converter (ADC): The electrical signals generated by the sensors are often in analog form.

To process these signals in a digital system, they need to be converted into digital format.

The analog-to-digital converter (ADC) performs this conversion, transforming the continuous analog signals into discrete digital values that can be processed by the microcontroller.

Block Diagram of Embedded Systems Flow

Microcontroller: The microcontroller processes the digital data received from the ADC and performs the necessary computations, decision making, and control algorithms based on the programmed instructions.

Block Diagram of Embedded Systems Flow

Digital-to-Analog Converter (DAC): In certain applications, the output of the microcontroller needs to be converted back to analog form to drive analog devices or actuators.

The digital-to-analog converter (DAC) converts the digital control signals from the microcontroller into corresponding analog signals that can be understood by the actuators.

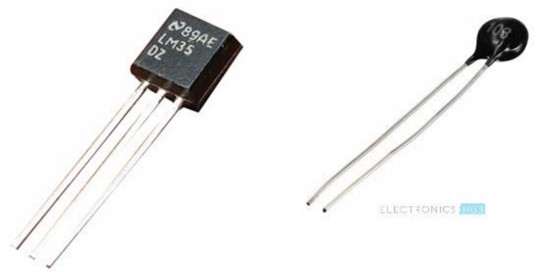
Block Diagram of Embedded Systems Flow

Actuators: Actuators are the output devices that generate physical action or movement based on the control signals received from the microcontroller. Actuators can include motors, solenoids, valves, or other devices that convert electrical signals into mechanical motion.

They perform the desired actions, such as moving a robotic arm, controlling a valve, or adjusting the position of a device.

Sensor

A sensor is defined as a device or a module that helps to detect any changes in physical quantity like pressure, force or electrical quantity like current or any other form of energy.

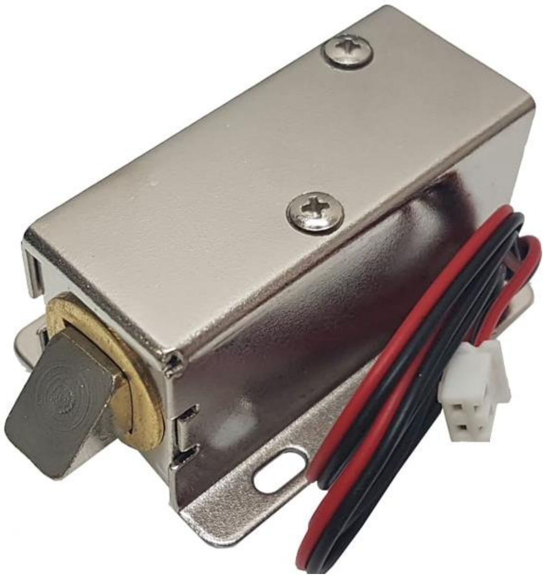


LM35 - Temperature Sensor IC

10KΩ NTC Thermistor

Solenoids

- Are actuators that use an electromagnet to create a force.
- Are typically used to open or close valves, or to move small objects.
- Are often used in conjunction with electric motors to provide precise control of motion.



Key milestones and developments in the history of embedded systems

Key milestones and developments in the history of embedded systems

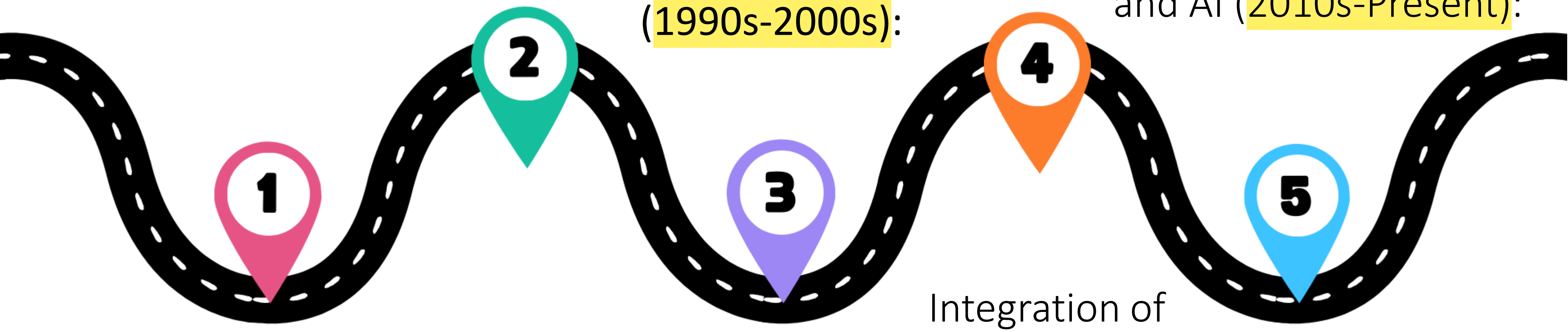
1. Early Development (1940s-1960s):

Microcontrollers and the 8-bit Era (1970s-1980s):

Rise of 16-bit and 32-bit Microcontrollers (1990s-2000s):

Integration of Connectivity and Networking (2000s-Present):

Advancements in Processing Power and AI (2010s-Present):



Key milestones and developments in the history of embedded systems

1. Early Development (1940s-1960s):

The term "embedded system" was coined in the late 1960s by Jack Ganssle while working at IBM.

In the 1940s and 1950s, computers were large and expensive, mainly used for scientific and military purposes. Early embedded systems were developed to control specific tasks, such as radar systems during World War II.

The first electronic digital computer, ENIAC (Electronic Numerical Integrator and Computer), built during World War II, can be considered an early embedded system due to its dedicated control systems.

Key milestones and developments in the history of embedded systems

2. Microcontrollers and the 8-bit Era (1970s-1980s):

The introduction of microcontrollers in the 1970s revolutionized embedded systems.

Microcontrollers integrated a microprocessor, memory, and I/O peripherals on a single chip, making them smaller, cheaper, and more power-efficient.

Intel's 4004, released in 1971, was one of the earliest microprocessors, paving the way for future developments. In the 1980s, **8-bit microcontrollers**, such as the **Intel 8051** and **Motorola 68HC11**, gained popularity, enabling the development of a wide range of embedded systems, including home appliances, automotive systems, and consumer electronics.

Key milestones and developments in the history of embedded systems

3. Rise of 16-bit and 32-bit Microcontrollers (1990s-2000s):

The 1990s witnessed the rise of 16-bit and 32-bit microcontrollers, offering more computational power and enhanced capabilities.

Rapid advancements in semiconductor technology, lower costs, and increased memory sizes allowed for more complex embedded systems to be developed.
The adoption of real-time operating systems (RTOS) became more prevalent, enabling better multitasking, resource management, and improved system reliability.

Key milestones and developments in the history of embedded systems

4. Integration of Connectivity and Networking (2000s-Present):
With the proliferation of the internet and advances in wireless communication, embedded systems began incorporating connectivity and networking features.

The emergence of protocols like Ethernet, Wi-Fi, and Bluetooth enabled seamless connectivity between embedded systems and other devices or networks.

The concept of the Internet of Things (IoT) gained traction, where embedded systems could connect, communicate, and interact with each other and the internet, leading to a wide range of applications in smart homes, industrial automation, healthcare, and more.

Key milestones and developments in the history of embedded systems

Advancements in Processing Power and AI (2010s-Present):
Embedded systems have continued to benefit from advancements in processing power, with the integration of multicore processors and powerful GPUs.

The application of artificial intelligence (AI) and machine learning (ML) techniques in embedded systems has become more prevalent, enabling intelligent decision-making, computer vision, and natural language processing in real-time applications.

Thank you for listening. End of Module 1.