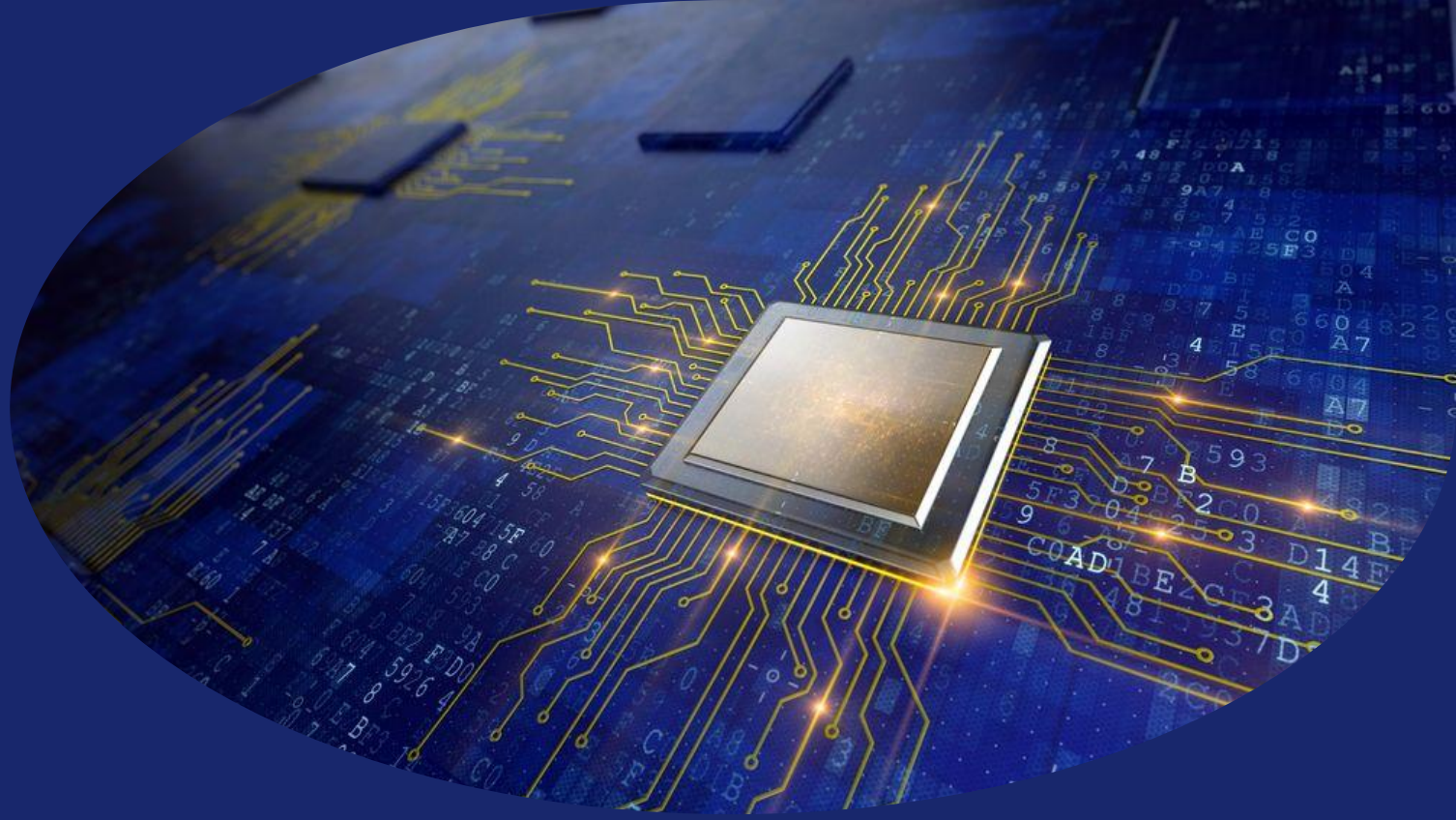
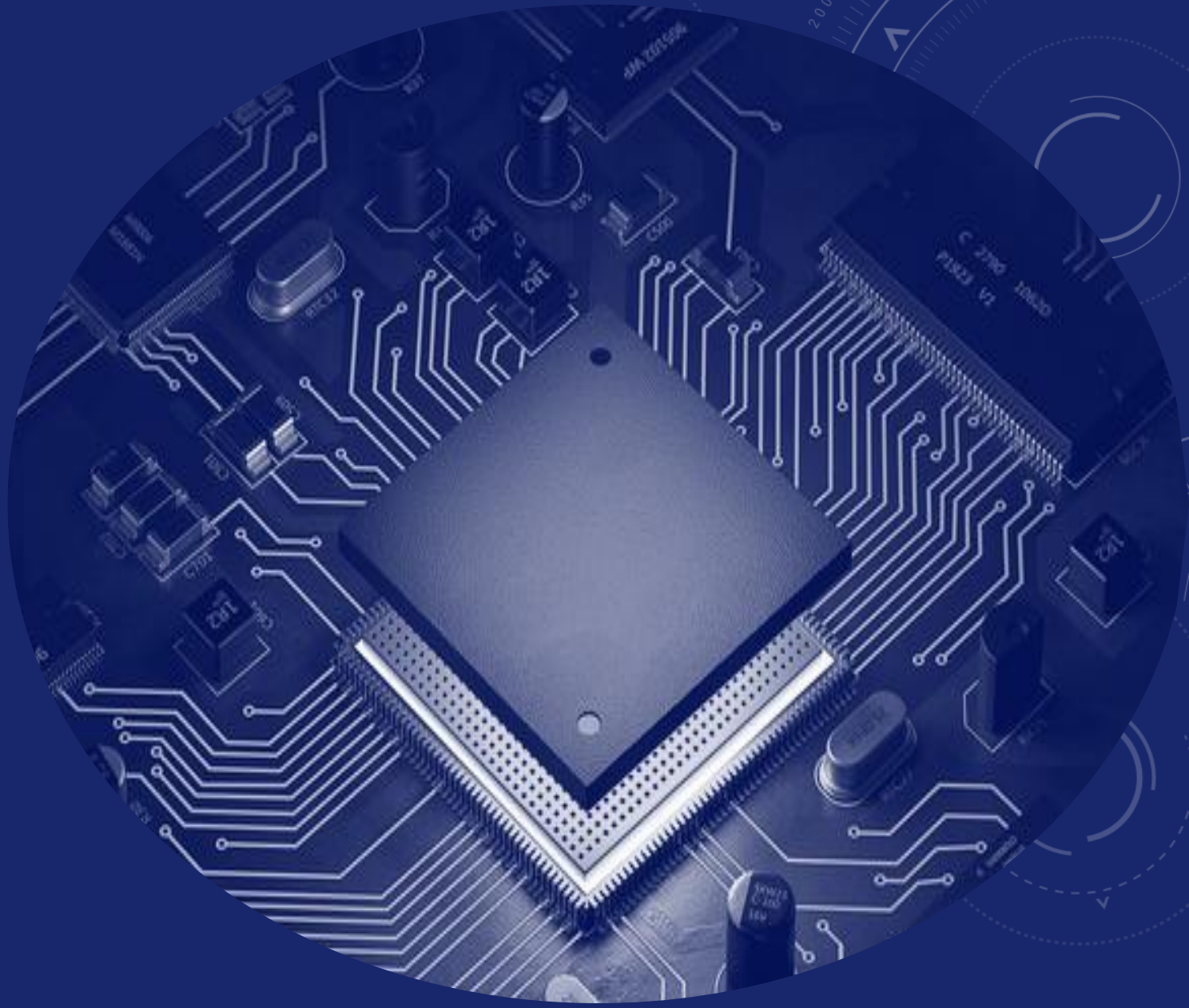


EMBEDDED SYSTEMS

ASSOCIATED PROF. WAFAA SHALASH

COURSE OVERVIEW:

- Introduction to microcontroller
- Sensors and Actuators
- C programming
- AVR microcontroller



WHAT IS EMBEDDED SYSTEMS?

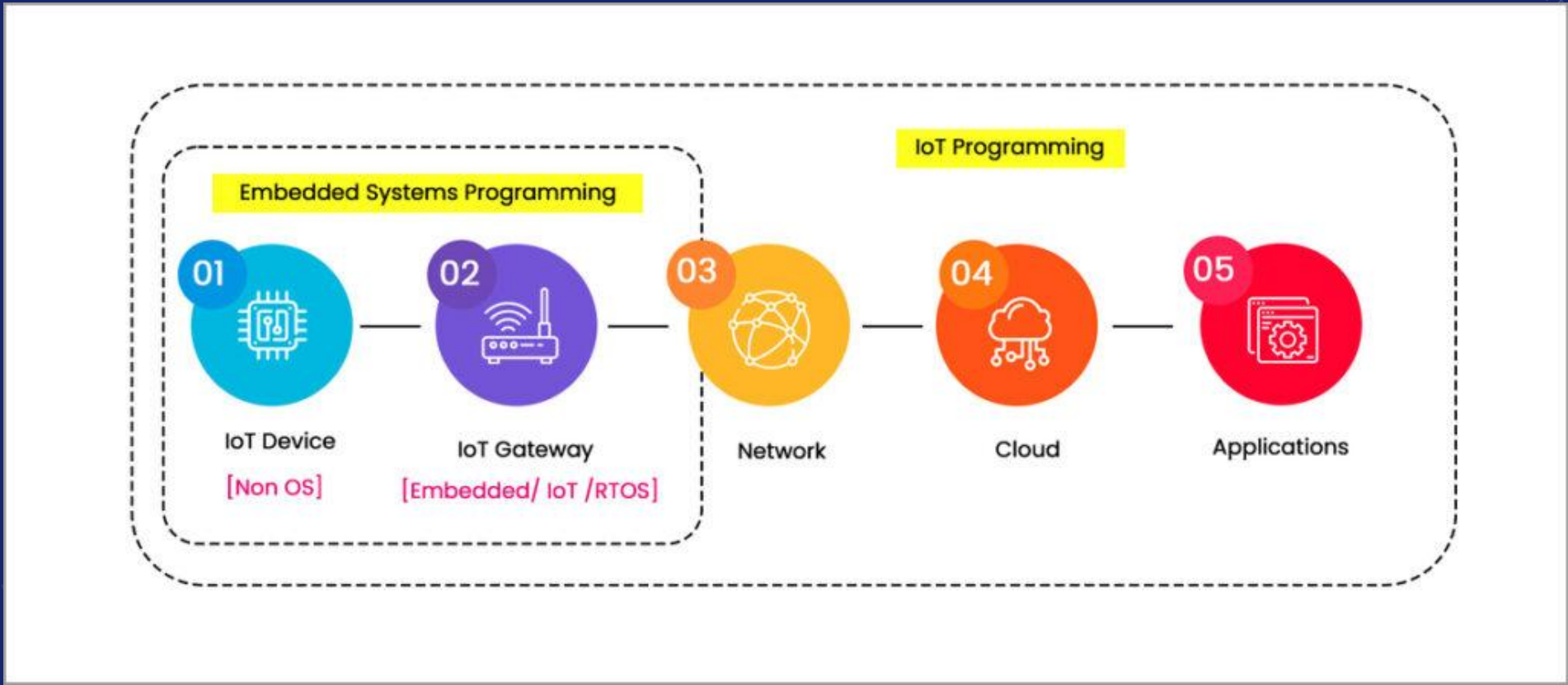
- An **embedded system** is a **specialized computer system** designed to perform a **specific task** within a larger system. Unlike general-purpose computers (like laptops or smartphones), embedded systems are **optimized for efficiency, reliability, and real-time performance**.

KEY FEATURES OF EMBEDDED SYSTEMS:

- **Dedicated Function** → Designed for a **specific task** (e.g., controlling a washing machine, monitoring sensors in IoT).
- **Real-Time Operation** → Many embedded systems process data **in real time** (e.g., airbags in cars).
- **Low Power & Compact Size** → Often run on **low-power hardware** with **minimal resources**.
- **Microcontroller or Microprocessor Based** → Uses components like **ESP32, Raspberry Pi, STM32, or Arduino**.
- **Limited User Interface** → May have **no screen** or a simple display with buttons.

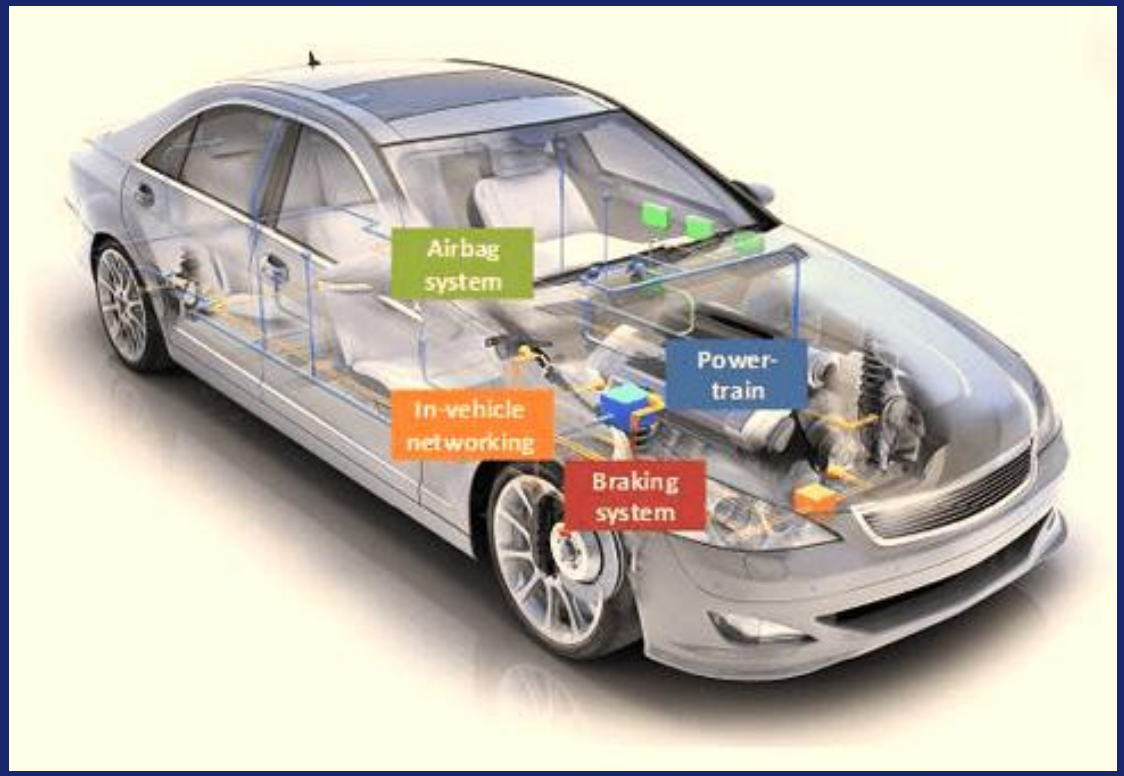
Examples of Embedded Systems:

- **IoT Devices** → Smart home systems (e.g., smart bulbs, thermostats).



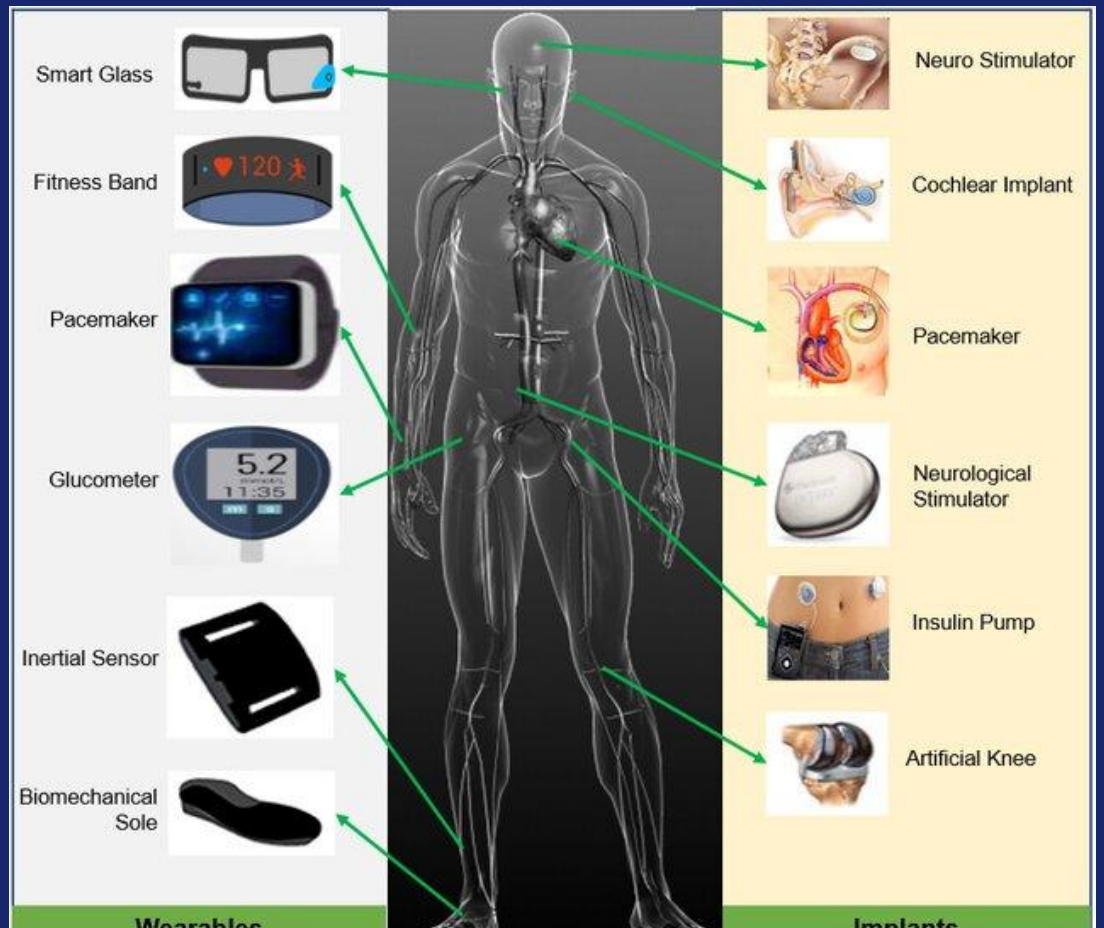
Examples of Embedded Systems:

- **Automotive Systems** → ABS brakes, airbag controllers, and engine management units.



Examples of Embedded Systems:

- **Medical Devices** → Pacemakers, MRI scanners, glucose monitors.



Examples of Embedded Systems:

- **Consumer Electronics** → Smart TVs, washing machines, and digital cameras.

Examples of Embedded Systems:

- **Industrial Control** → Robotics, factory automation, and CNC machines.

Embedded System vs. General-Purpose System

Feature	Embedded System	General-Purpose Computer
Purpose	Specific Task	Multiple Tasks
Processing Power	Low to Medium	High
User Interface	Minimal/None	Full UI (GUI, Keyboard)
Real-Time Support	Yes (Often)	No (Not Always)
Power Efficiency	High (Low Power)	Medium to High (Depends)
Examples	IoT, Cars, Robots	Laptops, Smartphones

WHY ARE EMBEDDED SYSTEMS IMPORTANT?

- **Everyday Use** → They are in almost **every device** we use.
- **Efficiency** → Designed to be **fast and power-efficient**.
- **IoT & Automation** → Powering **smart homes, industries, and healthcare**.

EMBEDDED SYSTEMS VS. IOT SYSTEMS

WHILE **EMBEDDED SYSTEMS** AND **IOT SYSTEMS** ARE CLOSELY RELATED, THEY HAVE **KEY DIFFERENCES** IN DESIGN, COMMUNICATION, AND APPLICATION.

What’s the Difference?

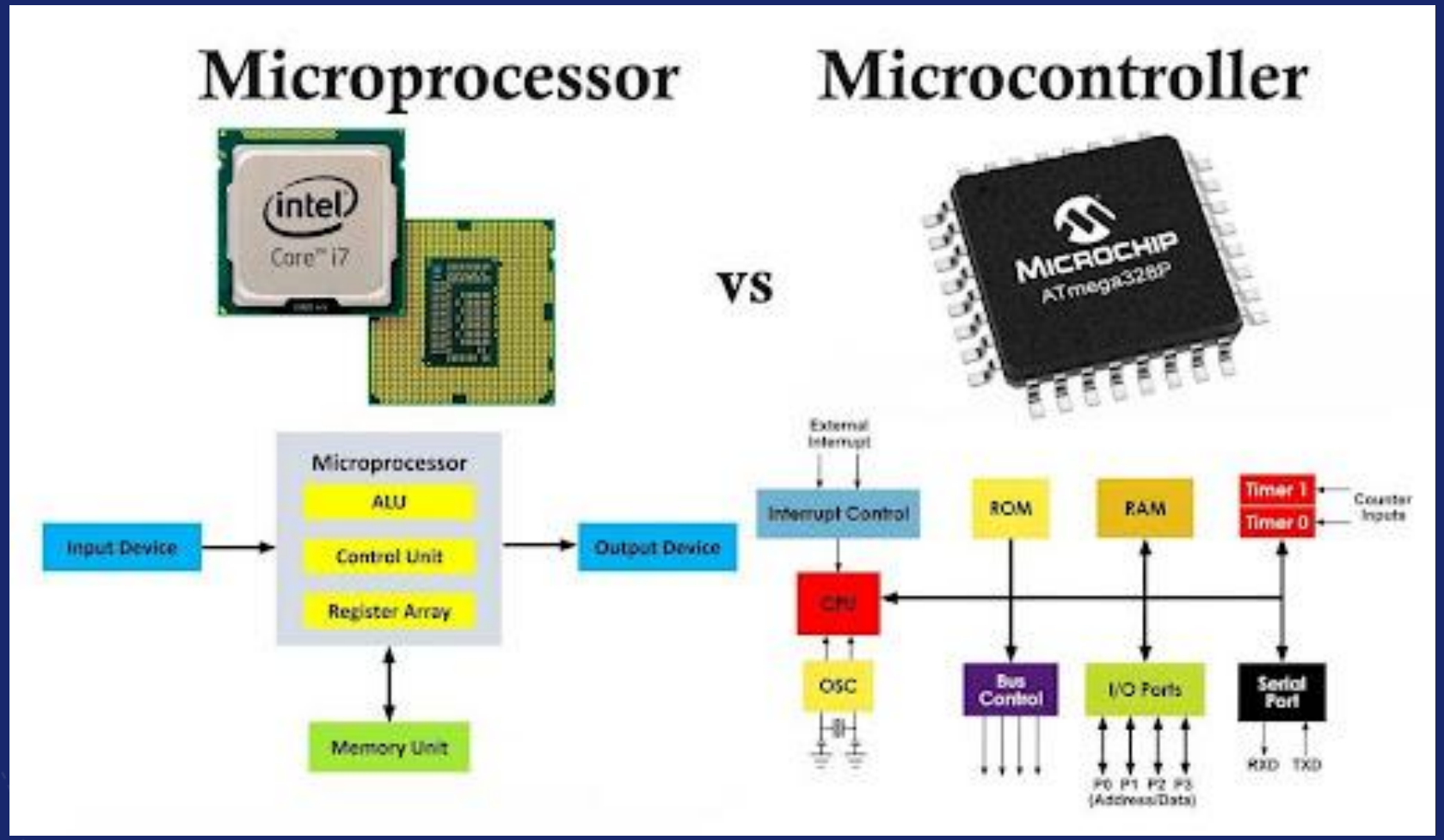
Feature	**Embedded System** 🖥️	**IoT System** 🌍
Definition	A self-contained computing system designed for a specific task.	A network of interconnected embedded systems that communicate over the internet.
Connectivity	Usually **standalone** (limited/no internet connection).	**Connected to the internet** for remote monitoring and control.
Example Devices	Washing machines, digital cameras, automotive ECU.	Smart thermostats, wearable health monitors, smart agriculture.
Data Exchange	Limited to internal processing.	Sends/receives data via **cloud services & networks**.
User Interaction	Minimal (buttons, LED indicators).	Often controlled via **mobile apps, dashboards**.
Scalability	**Fixed functionality** with limited expansion.	**Highly scalable**, supporting multiple connected devices.

EXAMPLE SCENARIOS

- 📟 **Example 1: Embedded System (Non-IoT)**
- A **microwave oven** has an embedded system to control heating time, power level, and the turntable.
- No internet connectivity.
- Uses a **microcontroller (like an STM32)** to handle inputs (buttons) and outputs (display, buzzer).
- 🌍 **Example 2: IoT-Enabled Embedded System**
- A **smart thermostat (e.g., Nest)** adjusts room temperature based on user settings and weather data.
- Uses **WiFi + MQTT** to send temperature data to the cloud.
- Controlled via a **mobile app** or voice assistants (Google Assistant, Alexa).

EXAMPLE SCENARIOS

MICROCONTROLLER VS. MICROPROCESSOR



MICROCONTROLLER VS. MICROPROCESSOR

	Microcontroller	Microprocessor
Design complexity	Low	High
Clock speed	Slow	Fast
Operating system	No	Yes
Processing speed	Low	High
Power consumption	Low	High
Memory	Small / Internal	Large / External
I/O pins	Yes	No
Number of bits	8-32 bits	32-64 bits
Cost	Low	High

NEXT TIME .....

MORE ONE **MICROCONTROLLER VS. MICROPROCESSOR**