

# WHAT IS A MICROCONTROLLER?

*A microcontroller is a whole small computer on one chip: a CPU, memory, and built-in peripherals that run one program to sense and control the world.*

---

ONE THOUSAND DRONES ENGINEERING TEAM · VERIFIED 2026-07

A microcontroller is a whole small computer on a single chip: a processor, its memory, and a set of built-in peripherals, all on one piece of silicon. It runs one program, on its own, to sense and control the physical world. The ESP32 on a One Thousand Drones board is a microcontroller, and this whole library is about getting the most out of one.

## WHAT IS ON THE CHIP?

Four things share the die. The CPU executes your program. RAM is fast working memory that holds variables while the chip runs, and forgets everything at power-off. Flash is non-volatile memory that stores the firmware, so the program survives a power cycle. And the peripherals are dedicated hardware blocks for talking to the world: GPIO pins, an ADC, timers, and serial buses. A desktop computer spreads those across many parts; a microcontroller puts them on one chip.

## HOW IS IT DIFFERENT FROM A PC PROCESSOR?

A PC processor is one part of a large system, with separate RAM sticks, a separate drive, and an operating system sitting between your program and the hardware. A microcontroller integrates its memory and peripherals on-die, runs a single program directly on the metal, starts in milliseconds, and sips power measured in milliamps. It trades raw speed for integration, real-time control, and low power, which is exactly what a small connected device needs.

## BARE-METAL OR AN RTOS

Firmware runs one of two ways. Bare-metal means your code is the only thing running, usually one endless loop. Or it runs on a small real-time operating system (an RTOS) that schedules several tasks and juggles them. The ESP32's default framework ships [FreeRTOS](#), so even a simple program runs as a task under it, which is worth knowing the first time you meet a task or a delay call.

**DEEP DIVE · WHY THE ESP32-S3 SPECIFICALLY**

The ESP32-S3 packs a lot onto one part: two Xtensa CPU cores, a Wi-Fi and Bluetooth Low Energy radio, native USB, and the usual GPIO, ADC, timers, and serial peripherals. That is why one chip can be the entire computer, radio, and input-output of a small connected device, with almost nothing else on the board but power and connectors. The concepts in this library are generic to microcontrollers; the ESP32-S3 is just the specific part we work them on. (Espressif ESP32-S3 datasheet)

- [Espressif. ESP32-S3 Series Datasheet \(system overview, on-chip CPU, memory, and peripherals\).](#) [espressif.com](#)

ONE CHIP: A CPU, ITS MEMORY, AND THE PERIPHERALS THAT CONNECT IT TO THE WORLD.

On a One Thousand Drones L1.01 board the ESP32-S3 module is the whole computer. Everything else on the board just feeds it power and breaks its pins out to headers you can wire to.

**CHECKPOINT****1. What makes a microcontroller different from a typical PC processor?**

- It always runs faster than a desktop CPU
- Its memory and peripherals are built onto the same chip**
- It needs a separate operating-system disk to start

ANSWER · B

*A microcontroller integrates RAM, flash, and peripherals on one die; a PC CPU relies on separate parts and an OS.*

**2. Which of these is NOT normally built into a microcontroller?**

- Flash memory that holds its program
- Timers and GPIO pins
- A removable hard drive**

ANSWER · C

*Storage and peripherals are on-chip; there is no removable drive on a microcontroller.*

**3. A microcontroller is best described as what?**

- A whole small computer on one chip**
- A processor that needs external RAM and storage chips to run
- A memory chip that stores data but cannot run a program

ANSWER · A

*A microcontroller integrates the processor, memory, and peripherals on one chip; a bare microprocessor needs external RAM and storage, and a memory chip has no processor.*

- Next: [GPIO, reading and driving pins](#)