

A computer screen with white text

AI-generated content may be incorrect.

**Basic Classification of Development Boards**

| Feature | Single Board Microcontroller (SBM) | Single Board Computer (SBC) |
| --- | --- | --- |
| Core Comp. | Based on Microcontroller (MCU) | Based on Microprocessor (CPU) |
| Includes | CPU (inside MCU), Flash memory, Clock, USB/programmer interface, GPIO/I/O headers | CPU + RAM + storage (onboard/external), Ports (HDMI, USB, Ethernet, WiFi, etc.) |
| Program Execution | Runs one program at a time (bare-metal or lightweight runtime, no full OS) | Runs a full Operating System (Linux, Windows IoT, etc.) |
| Memory & Storage | Limited Flash & SRAM inside MCU | External RAM & storage (microSD, eMMC, SSD) |
| Connectivity | GPIO pins, UART, I2C, SPI, PWM, ADC/DAC | HDMI, USB, Ethernet, WiFi, Bluetooth, GPIO (limited) |
| Use Cases | Real-time control, robotics, sensors, IoT devices | AI, multimedia, multitasking, networking, edge computing |
| Examples | Arduino Uno (ATmega328P), ESP32 DevKit, Teensy 4.1 | Raspberry Pi, BeagleBone Black, NVIDIA Jetson Nano |

**A screenshot of a computer program

AI-generated content may be incorrect.**

So in context to Arduino we have two classification views:

1. By architecture → SBM vs SBC.
2. By ecosystem → Official, Compatible, Non-Arduino.

| Category | Examples | Notes |
| --- | --- | --- |
| Official Arduino Boards | Uno, Mega, Nano, GIGA, Portenta | Made & supported by Arduino. Strong IDE support. |
| Arduino-Compatible Boards | Teensy, ESP32, STM32, Seeeduino | Not official, but IDE & library compatible. Often more powerful/cheaper. |
| Non-Arduino Boards | Raspberry Pi, BeagleBone, Jetson Nano, FPGA boards | Different ecosystem (Linux, Python, HDL, etc.). Not programmed with Arduino IDE. |
|  |  |  |
| Category | SBM (Single Board Microcontroller) | SBC (Single Board Computer) |
| Official Arduino Boards | Arduino Uno, Nano, Mega, Due, GIGA R1 WiFi, Portenta H7 | Portenta X8 (Linux-based SBC) |
| Arduino-Compatible Boards | Teensy 4.1, ESP32 DevKit, STM32 Blue Pill, Seeeduino | (Rare, most 3rd party boards are SBM, not SBC) |
| Non-Arduino Boards | Few exist (e.g., some STM32 boards not Arduino IDE friendly) | Raspberry Pi, BeagleBone Black, NVIDIA Jetson Nano, FPGA SoC boards |

| Board | Manufacturer | Microcontroller (MCU) | CPU Type | Clock Speed | Category |
| --- | --- | --- | --- | --- | --- |
| Arduino Uno R3 | Arduino (official) | ATmega328P (8-bit AVR) | 1× 8-bit AVR core | 16 MHz | Official Arduino |
| Teensy 4.1 | PJRC (3rd-party) | ARM Cortex-M7 (32-bit) | 1× 32-bit ARM core | 600 MHz | Arduino-compatible |
| ESP32 Dev Board | Espressif / 3rd-party | Tensilica Xtensa LX6 (32-bit) | 2× 32-bit cores | 240 MHz | Arduino-compatible |

Teensy → Third-party, works with Arduino IDE (via Teensyduino), extremely powerful (600 MHz).

**ARDUINO-(SBM)**

Arduino is an Italian open-source hardware and software company, project, and user community known for designing and manufacturing single-board microcontrollers and microcontroller kits.

Arduino is an open-source microcontroller development board (like Arduino Uno R3) that combines hardware and software to make electronics projects simple and interactive.

Arduino also makes simpler the working process of microcontroller, but it gives some advantages over other systems for teachers, students, and beginners.

- Inexpensive  
- Cross-platform  
- The simple, clear programming environment  
- Open source and extensible software  
- Open source and extensible hardware

**Different Types Of Arduino Boards**

The list of Arduino boards includes the following such as

* Arduino Uno (R3)
* Arduino Nano
* Arduino Micro
* Arduino Due
* LilyPad Arduino Board
* Arduino Bluetooth
* RedBoard Arduino Board
* Arduino Mega (R3) Board
* Arduino Leonardo Board

MOST POWERFUL SINGLE BOARD MICROCONTROLLER (SBM) IS TEENSY (NON-ARDUINO):

**Teensy vs Arduino High-End Boards**

**1. Teensy 4.1 (most powerful hobbyist microcontroller board)**

* Processor: ARM Cortex-M7 @ 600 MHz (overclockable up to ~1 GHz)
* RAM: 1 MB tightly coupled + 512 KB general-purpose
* Flash: 8 MB external
* I/O: 55 digital I/O, multiple UART/SPI/I²C, Ethernet, USB host/device, SD card, audio, etc.
* Price: Approximately ₹3,500–₹5,000
* Notes: Extremely fast relative to price, often considered the highest-performance hobbyist MCU board.

**2. Arduino GIGA R1 WiFi**

* Processor: Dual-core STM32H7 (Cortex-M7 @ 480 MHz + Cortex-M4 @ 240 MHz)
* RAM: 1 MB
* Flash: 2 MB
* Features: Large number of GPIO, built-in Wi-Fi and Bluetooth, strong Arduino ecosystem support.

**3. Arduino Portenta H7**

* Processor: Dual-core STM32H7 (Cortex-M7 @ 480 MHz + Cortex-M4 @ 240 MHz)
* RAM: Up to 8 MB external SDRAM
* Flash: 16 MB external
* Features: Suitable for AI, industrial IoT, camera/display, and advanced applications.

**Why Teensy 4.1 is Considered Most Powerful in Practice**

* Higher clock speed (600 MHz vs Arduino’s 480 MHz).
* Highly optimized libraries (Teensyduino supports math, DSP, audio efficiently).
* Better cost-to-performance ratio compared to Portenta and GIGA.
* Supports safe overclocking up to 816 MHz – 1 GHz, not available on Arduino boards.

**Trade-offs**

* **Arduino boards**: Better beginner support, ecosystem, shields, and tutorials.
* **Teensy boards**: Higher raw performance, flexibility, lower cost, but require more low-level knowledge.

**Conclusion**

* Fastest official Arduino boards: Portenta H7 and GIGA R1 WiFi (480 MHz).
* Fastest hobbyist-friendly board in the Arduino ecosystem: Teensy 4.1 (600+ MHz, overclockable).

TEENSY v/s RASPBERRY Pi:

| **Feature** | **Teensy 4.1** | **Raspberry Pi 4** | **Raspberry Pi 5** |
| --- | --- | --- | --- |
| **Processor** | ARM Cortex-M7 (MCU core) | Broadcom BCM2711, Quad-core ARM Cortex-A72 | Broadcom BCM2712, Quad-core ARM Cortex-A76 |
| **Clock Speed** | 600 MHz (overclockable ~1 GHz) | 1.5 GHz (overclockable ~2.0 GHz) | 2.4 GHz (with boost, higher w/ cooling) |
| **Cores** | 1 core | 4 cores | 4 cores |
| **RAM** | 1 MB tightly coupled + 512 KB general-purpose | 2–8 GB LPDDR4 | 4–8 GB LPDDR4X (faster, lower latency) |
| **Storage** | External Flash (8 MB onboard) | microSD card (OS + data) | microSD card (OS + data) + NVMe SSD support |
| **Operating System** | None (bare metal / Arduino) | Linux (Raspberry Pi OS, Ubuntu, etc.) | Linux (Raspberry Pi OS, Ubuntu, etc.) |
| **Best For** | Real-time control, low-latency I/O, signal processing, robotics | General computing, servers, AI at small scale, multimedia | High-end edge AI, faster servers, heavier multitasking, graphics |
| **Latency** | Extremely low (µs range) | Higher (OS scheduling overhead) | Higher (but better than Pi 4 due to faster CPU) |
| **Price (approx.)** | ₹3,500–₹5,000 | ₹4,500–₹8,000 (depending on RAM) | ₹6,500–₹12,000+ (depending on RAM) |

| **Feature** | **Microprocessor** | **Microcontroller** |
| --- | --- | --- |
| **Definition** | A **chip containing only a CPU** (sometimes cache, FPU, etc.). Needs external support. | A **chip containing CPU + memory + I/O** all in one. |
| **Memory & I/O** | External RAM, ROM, and I/O chips must be connected. | Internal Flash, SRAM, EEPROM, and I/O ports are built-in. |
| **System cost** | Expensive (needs many external components). | Cheaper (single-chip solution). |
| **Power consumption** | Higher (multi-GHz, multi-core, runs complex OS). | Low (MHz range, optimized for embedded tasks). |
| **Speed** | Very high (GHz, used in PCs & smartphones). | Moderate (kHz–MHz, used in control systems). |
| **Applications** | General-purpose computing: laptops, desktops, servers. | Embedded systems: Arduino Uno R3, washing machines, cars, IoT. |
| **Examples** | Intel 8085, Intel Core i7, AMD Ryzen, ARM Cortex-A53. | ATmega328P (Arduino Uno R3), PIC, STM32, ESP32. |

A screenshot of a computer program

AI-generated content may be incorrect.

**🔹 Key Point**

* The **CPU core inside a microcontroller (like ATmega328P in Arduino Uno R3)** performs the **same fundamental job as a microprocessor** → fetching, decoding, and executing instructions.
* BUT it is **not sold separately** as a standalone chip. It comes bundled with **memory (Flash, SRAM, EEPROM) and peripherals (GPIO, ADC, timers, UART, SPI, etc.)** in the same microcontroller package.

**🔹 Why we don’t say “microcontrollers contain microprocessors”**

* In electronics terminology, **microprocessor** usually refers to a **standalone CPU chip** (like Intel i7, ARM Cortex-A53, or 8085) that requires external RAM, ROM, and I/O to function.
* A **microcontroller** already has a **CPU core built-in**, but since it’s **integrated and not separate**, we call it a **CPU inside a microcontroller**, not a microprocessor.

**🔹 Simple Analogy**

* **Microprocessor (standalone CPU):** Just the **brain chip** 🧠, sold separately. Needs memory + I/O chips to form a complete computer.
* **CPU inside a microcontroller:** A brain 🧠 that’s already **part of a whole body** 🤖 (with memory, senses, and hands built-in).

👉 **Final Statement (to remember):**

* The **CPU inside a microcontroller** works like a microprocessor, but since it’s not standalone, we call it a **CPU core**, not a microprocessor.
* Therefore: **Microcontrollers contain a CPU, not a microprocessor.**

**Q1: If ATmega328P is a microcontroller, why is Arduino also called a microcontroller?**

* **Arduino Uno R3** is **not itself a microcontroller**.
* It is a **development board** built around the **ATmega328P microcontroller**.

👉 That means:

* The **actual microcontroller = ATmega328P**
* The **Arduino board** = ATmega328P + supporting hardware (USB-to-serial chip, voltage regulator, oscillator, connectors, PCB, etc.)

So technically:

* **Arduino Uno R3 = Microcontroller development board**
* People sometimes casually say “Arduino is a microcontroller” because the **main component is the ATmega328P microcontroller**.

✅ Correct wording: *Arduino Uno R3 is a microcontroller-based board.*

**Q2: Can the CPU inside ATmega328P be called a microprocessor?**

Yes, but with nuance 👇

* The **CPU core inside ATmega328P** executes instructions, just like a **microprocessor**.
* It is an **8-bit RISC CPU core** (often simply called the AVR core).
* In textbooks, the **CPU inside a microcontroller is sometimes referred to as a microprocessor**, since it performs the same kind of computation (fetch–decode–execute cycle).

BUT 🚫:

* When we say **“microprocessor” in electronics**, we usually mean a **standalone chip like Intel i7 or ARM Cortex-A**, not the CPU inside a microcontroller.
* That’s why to avoid confusion, we say:
  + **ATmega328P = Microcontroller**
  + **Inside it = AVR CPU core (which functions like a microprocessor)**

| **Feature** | **Specification** | **Explanation** |
| --- | --- | --- |
| **Microcontroller** | ATmega328P (8-bit AVR) | The “brain” of the board – executes instructions. |
| **Operating Voltage** | 5V | Logic HIGH = 5V, LOW = 0V. Directly connects with 5V sensors. |
| **Input Voltage (recommended)** | 7–12V (via barrel jack or VIN) | Safe range for onboard regulator. Less than 7V → unstable; more than 12V → overheating. |
| **Digital I/O Pins** | 14 (D0–D13) | Used to read/write digital signals (HIGH/LOW). |
| **PWM Pins** | 6 (D3, D5, D6, D9, D10, D11) | Can output “fake analog” signals for dimming LEDs, motors, etc. |
| **Analog Input Pins** | 6 (A0–A5) | Used to read analog voltages (0–5V) → converted into values 0–1023. |
| **Flash Memory** | 32 KB (0.5 KB used by bootloader) | Stores your program (sketch). |
| **SRAM** | 2 KB | Stores variables during program execution (temporary memory). |
| **EEPROM** | 1 KB | Permanent memory to save data even if power is off. |
| **Clock Speed** | 16 MHz | Defines speed of instruction execution. |
| **USB Connection** | ATmega16U2 chip handles USB-to-serial | Lets you upload code + communicate with PC (Serial Monitor). |
| **Communication** | UART, SPI, I²C | For talking with sensors/modules. |
| **Current per I/O pin** | 40 mA max | Limit per pin to avoid damage. |