# Comparison Table

|  |  |  |
| --- | --- | --- |
| Feature | Microcontroller (e.g., Arduino) | Single-Board Computer (e.g., Raspberry Pi) |
| Main Use | Specific, real-time control tasks | General-purpose computing tasks |
| Operating System | No OS (runs bare-metal code) | Runs full OS (Linux, Windows IoT, etc.) |
| Power Consumption | Very low | Higher |
| Boot Time | Instant (milliseconds) | Slow (takes seconds to boot OS) |
| Hardware Resources | Limited (small memory, no GPU, low clock speed) | Powerful (multi-core CPU, GPU, large RAM, storage) |
| Connectivity | Usually minimal (basic USB, I2C, UART) | Full connectivity (USB, Wi-Fi, Bluetooth, HDMI, Ethernet) |
| Cost | Generally cheaper ($2–$20) | More expensive ($30 and up) |
| Programming | Typically in C/C++ (e.g., Arduino IDE) | Can use Python, Java, C++, etc. |

# Reasons to Use Each

## Why Use a Microcontroller

1. Power Efficiency: Ideal for battery-powered or energy-sensitive applications (e.g., temperature sensors, wearables).

2. Real-Time Control: Handles time-critical tasks like motor control, sensor polling, and embedded systems with minimal latency.

## Why Use a Single-Board Computer

1. Complex Tasks: Can run advanced software like databases, image processing, or even host a web server.

2. Multimedia and Interfaces: Great for projects that need GUI, HDMI display output, internet access, or camera processing (e.g., smart home hubs, surveillance systems).