Chapter 4: Microcontrollers and Hardware Platforms

📘 Description

Microcontrollers serve as the "brain" of IoT devices, managing sensor data, making decisions, and controlling actuators. This chapter introduces key microcontroller platforms, their features, and how they are used to build IoT applications.

🎯 Learning Objectives

By the end of this chapter, you will be able to:

- Understand the role of microcontrollers in IoT systems.
- Identify popular hardware platforms (Arduino, ESP32, Raspberry Pi).
- Compare features and use-cases of different microcontrollers.
- Learn how to set up and program basic IoT circuits.

🧠 Section 1: What is a Microcontroller?

A **microcontroller (MCU)** is a compact integrated circuit designed to govern a specific operation in an embedded system. It includes a processor, memory, and I/O pins on a single chip.

> In IoT, microcontrollers gather sensor data, process logic, and control actuators.

⚙️ Section 2: Popular Microcontroller Platforms

1. Arduino UNO

- Based on **ATmega328P** chip
- 14 digital I/O pins, 6 analog inputs
- Ideal for beginners
- Simple C/C++ programming using Arduino IDE

2. ESP8266 / ESP32

- Built-in **Wi-Fi** and **Bluetooth** (ESP32 only)
- Low cost, high performance
- Used for smart home devices, wireless control
- Programmable via Arduino IDE or MicroPython

3. Raspberry Pi

- Single-board computer (not a microcontroller)
- Runs full Linux OS (e.g., Raspbian)
- Supports Python, GUI apps, networking, and camera input
- Ideal for advanced IoT applications like media centers, servers, and AI integration

---

🔌 Section 3: GPIO and Interfacing

- **GPIO (General Purpose Input/Output)** pins are used to connect sensors and actuators.
- Pins can be configured as **input** (read sensor values) or **output** (control devices).

Sample Pin Functions:

Pin Type	Function
Digital	ON/OFF sensors, LEDs, buzzers
Analog	Varying sensor values (e.g., light, temperature)
PWM	Motor speed control, LED brightness
I2C/SPI	Communicate with multiple sensors

💻 Section 4: Programming Environment

- **Arduino IDE**: Used for Arduino/ESP devices (based on C++)
- **MicroPython**: Lightweight Python interpreter for MCUs
- **Thonny**: Python IDE suitable for Raspberry Pi
- **GPIO libraries**: Used to control hardware in Python (e.g., `gpiozero`, `RPi.GPIO`)

Sample Code: Blink LED on Arduino

cpp
CopyEdit

```
void setup() {
  pinMode(13, OUTPUT); // Set pin 13 as output
}
void loop() {
  digitalWrite(13, HIGH); // Turn LED on
  delay(1000);            // Wait 1 second
  digitalWrite(13, LOW);  // Turn LED off
  delay(1000);            // Wait 1 second
}
```

🧪 Section 5: Project Ideas for Beginners

- **Temperature logger** using Arduino + DHT11 sensor
- **Wi-Fi controlled light** using ESP32
- **Motion-activated camera** using Raspberry Pi
- **Smart door lock** with servo motor and keypad

✅ Chapter Summary

- Microcontrollers are key to reading sensor data and controlling actuators in IoT systems.
- Popular platforms include Arduino (easy & beginner-friendly), ESP32 (IoT-focused), and Raspberry Pi (powerful Linux computer).
- GPIO pins allow you to connect various components and sensors.
- Each board supports specific programming tools and languages.