UNIT V

Microcontroller (MCU)

A **Microcontroller** is a **compact integrated circuit** designed to govern a specific task or device.

✅ Key Features:

- Contains **CPU + RAM + ROM + I/O ports** on one chip
- Optimized for **embedded systems**
- Low power, low cost

🔍 Example:

- ATmega328P (used in Arduino UNO), PIC, 8051, STM32

🧠 Architecture:

- **All-in-one system** (CPU + memory + I/O)
- No need for external components for basic tasks

📘 Example Use:

- Washing machines
- Microwave ovens
- Arduino projects
- Automotive control systems

🔬 3. Microprocessor vs Microcontroller

Feature	Microprocessor (MPU)	Microcontroller (MCU)
Purpose	General computing tasks	Specific control tasks (embedded)
Components	CPU only	CPU + RAM + ROM + I/O on-chip
Speed	High (GHz range)	Moderate (MHz range)
Power Consumption	High	Low
Memory	External RAM/ROM required	Built-in memory
Cost	More expensive	Cheaper

Feature	Microprocessor (MPU)	Microcontroller (MCU)
Example	Intel i7, AMD Ryzen, 8086	ATmega328, PIC, 8051
Applications	PCs, servers	Embedded devices, smart gadgets

⚙️ 4. Real-Life Analogy

- **Microprocessor** is like a **laptop CPU**: powerful, but needs RAM, storage, monitor, etc.
- **Microcontroller** is like a **calculator**: everything built in for one job.

✅ Summary

Concept	Microprocessor	Microcontroller
Focus	Processing power	Control and automation
Integration	CPU only	CPU + memory + peripherals
Applications	Computers, servers	Appliances, robots, IoT

Arduino is an open-source electronics platform based on **easy-to-use hardware and software**. It’s primarily used to build digital devices and interactive objects that can sense and control the physical world. Here's a breakdown of the main **Arduino concepts**:

🔌 1. What is Arduino?

- **Microcontroller-based board** (like Arduino UNO, Mega, Nano, etc.)
- Programmed using the **Arduino IDE** with a simplified version of C/C++
- Used in **prototyping, IoT, automation, robotics**, etc.

🧠 2. Basic Components

- **Arduino Board**: Main hardware (UNO is the most popular)
- **Microcontroller**: The brain of the board (e.g., ATmega328P)
- **Digital Pins**: Input/Output pins (0–13 on UNO)
- **Analog Pins**: For reading analog sensors (A0–A5)
- **Power Pins**: 3.3V, 5V, GND
- **USB Port**: For uploading code and serial communication

🖥️ 3. Arduino IDE

- Free software to write and upload code
- Code written in **.ino files**
- Two main functions in every sketch:

```
cpp
CopyEdit
void setup() {
```

```
  // runs once
}

void loop() {
  // runs repeatedly
}
```

🗒️ 4. Basic Code Concepts

- **Digital I/O**:

```
pinMode(13, OUTPUT);   // Set pin 13 as output
digitalWrite(13, HIGH); // Turn on LED
```

- **Analog Input**:

```
int value = analogRead(A0); // Read analog value from pin A0
```

- **Serial Communication**:

```
cpp
CopyEdit
Serial.begin(9600);       // Start serial at 9600 bps
Serial.println(value);    // Print value to serial monitor
```

🧪 5. Common Projects

- Blinking an LED
- Reading temperature from a sensor
- Controlling a servo motor
- Line-following robots
- IoT devices with WiFi (e.g., ESP8266)

📚 6. Popular Arduino Boards

Board	Microcontroller	Features
UNO	ATmega328P	Basic, beginner-friendly
Nano	ATmega328P	Smaller version of UNO
Mega 2560	ATmega2560	More I/O pins, memory
Leonardo	ATmega32u4	Can emulate USB devices
ESP8266	ESP8266	Built-in WiFi, IoT applications

🔗 7. Key Arduino Libraries

- `Servo.h` – control servo motors
- `LiquidCrystal.h` – LCD display
- `Wire.h` – I2C communication
- `SPI.h` – SPI communication
- `WiFi.h` / `ESP8266WiFi.h` – networking



ARDUINO UNO DIAGRAM

✅ Summary

Concept	Purpose
Microcontroller	Brain of the board
Digital I/O	Control LEDs, buttons, etc.
Analog I/O	Read sensors like potentiometers
Serial Communication	Debug and interact with PC
Sketch (.ino)	Code uploaded to Arduino

The **Raspberry Pi** is a **low-cost, credit-card-sized computer** that plugs into a monitor or TV and uses a keyboard and mouse. It enables people of all ages to explore computing, learn to program, and build electronics projects.

🍓 1. What is Raspberry Pi?

- A **single-board computer** developed by the Raspberry Pi Foundation.
- Runs a **full Linux operating system** (usually Raspberry Pi OS).
- Can be used for **programming, robotics, IoT, media centers**, and even as a **desktop PC**.

💡 2. Key Features

Feature	Description
CPU	ARM-based processor
RAM	Ranges from 512MB to 8GB depending on model
Storage	Uses a microSD card (OS + files)
GPIO Pins	General Purpose Input/Output for hardware projects
USB ports	Connect mouse, keyboard, etc.
HDMI port	For video output to monitors
WiFi & Bluetooth	Built-in in newer models
Camera & Display Ports	For Raspberry Pi Camera & touchscreen display

🧠 3. Popular Raspberry Pi Models

Model	RAM Options	Special Features
Raspberry Pi 4	2GB–8GB	Fast, dual display, USB 3.0
Raspberry Pi 3	1GB	WiFi + Bluetooth built-in
Raspberry Pi Zero	512MB	Tiny, low power, very cheap
Raspberry Pi 5	4GB/8GB	Much faster, PCIe support

🖥️ 4. Operating System

- Most common OS: **Raspberry Pi OS** (based on Debian Linux)
- Others: Ubuntu, Kali Linux, RetroPie, Home Assistant OS
- Supports full desktop, terminal access, Python, and more

🔌 5. GPIO (General Purpose Input/Output)

- 40-pin header used to **connect sensors, motors, LEDs**, etc.
- Can be controlled using **Python (RPi.GPIO or gpiozero libraries)**

Example Python code:

```
import RPi.GPIO as GPIO
import time

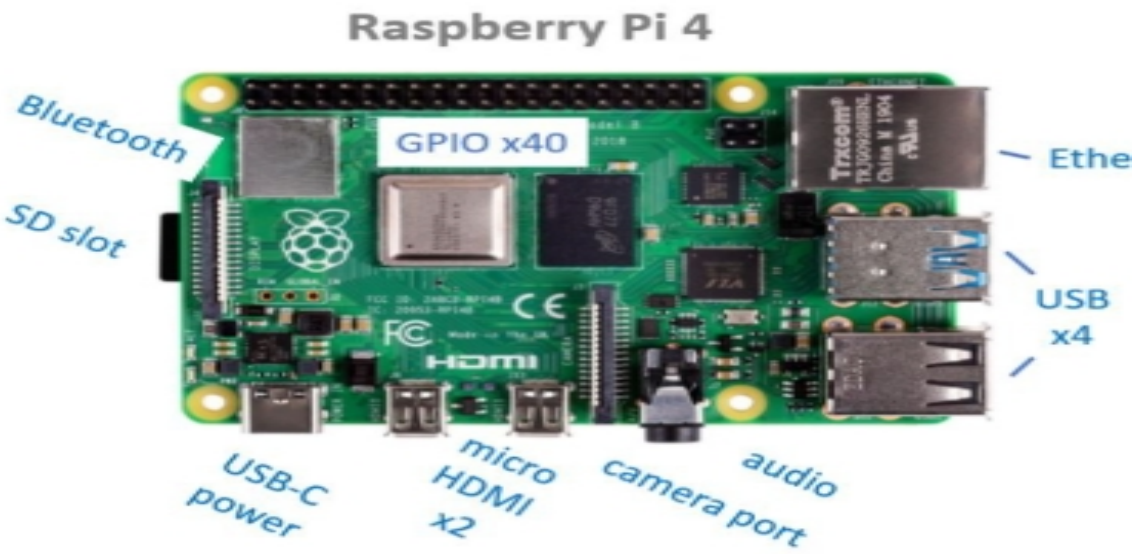
GPIO.setmode(GPIO.BCM)
GPIO.setup(18, GPIO.OUT)

GPIO.output(18, GPIO.HIGH)
time.sleep(1)
GPIO.output(18, GPIO.LOW)

GPIO.cleanup()
```

🧰 6. Common Uses & Projects

- Home automation (e.g., smart lights)
- Media center (e.g., using Kodi/OSMC)
- Weather station
- Retro gaming console (RetroPie)
- Personal web server
- IoT devices with sensors and actuators
- AI and Computer Vision with a camera module



RASPBERRY PI DIAGRAM.

🔌 7. Raspberry Pi vs Arduino

Feature	Raspberry Pi	Arduino
Type	Full computer (OS, multitasking)	Microcontroller board (single task)
Programming	Python, C++, Java, etc.	C/C++ in Arduino IDE
Power	5V via USB-C or micro-USB	5V via USB or external supply
Use Case	Complex software projects	Real-time control, simple I/O
GPIO	Yes	Yes
Networking	WiFi, Ethernet (built-in)	Needs external module

✅ Summary

Concept	Description
Linux OS	Lets you run full software environments
GPIO	Connect and control physical devices
Python Support	Built-in for scripting and automation
Versatile Use	Desktop, IoT, media, AI, and more