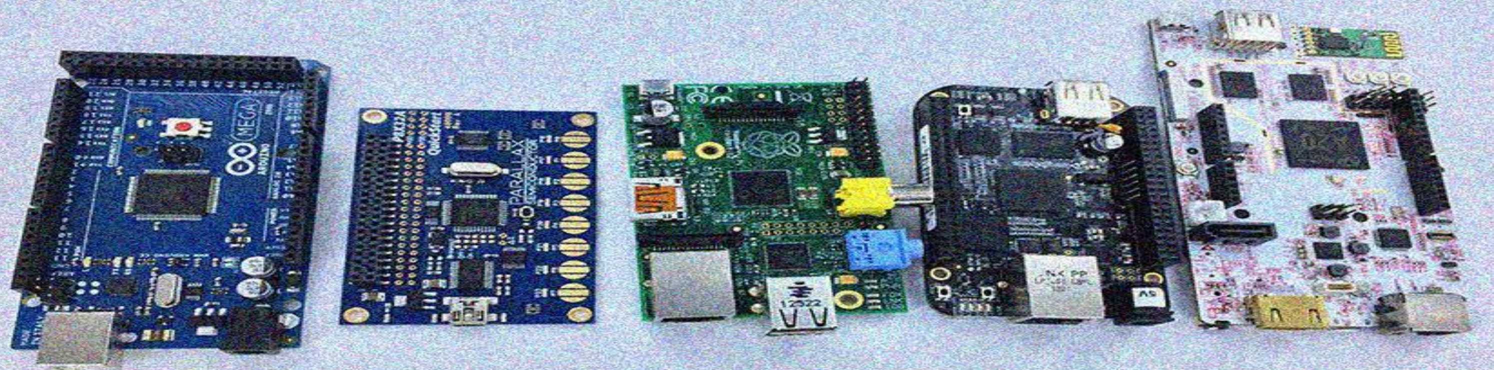


CHAPTER 15 MICROCONTROLLER & SBC

Learning Outcomes

- Introduction to Microcontroller
- SBC
- Architecture of Microcontroller/SBC
- Building Block of Microcontroller

Microcontrollers and Single Board Computers (SBCs) are the brains behind most robots and smart devices. They control how a robot senses the environment, processes information, and acts accordingly. To design and work with robots, it is important to understand how these computing units are built and how they function.

In this chapter, you will learn about the basic architecture of microcontrollers and SBCs by studying their block diagrams. Block diagrams are simple pictures that show the main parts of these computing devices and how they are connected. This helps us understand how the robot's "brain" receives inputs from sensors, processes data, and sends commands to motors or other parts.

Why Is This Important?

Knowing the architecture of microcontrollers and SBCs helps you design better robot control systems. It also makes programming easier because you understand how your code interacts with the hardware. This knowledge is the foundation for building smarter and more efficient robots. Microcontrollers are small computers built into chips no bigger than your fingernail. Despite their tiny size, they can control complex machines like drones, washing machines, and even Mars rovers!

What is a Microcontroller and SBC?

- **Microcontroller:**
  A microcontroller is a small computer built on a single integrated circuit (chip). It contains a processor, memory, and input/output (I/O) ports. Microcontrollers are designed to perform specific control tasks in embedded systems like robots, home appliances, and vehicles.
- **Single Board Computer (SBC):**
  An SBC is a complete computer built on a single circuit board. It includes a processor, memory, storage, and I/O ports. SBCs (like Raspberry Pi) are more powerful than

microcontrollers and can run full operating systems. They are often used in robotics for complex processing tasks.

Understanding the architecture of microcontrollers and SBCs helps you know how information flows inside these devices. It shows how they take inputs from sensors, process the data, and send commands to actuators (motors, LEDs, etc.) to perform actions.

Understanding Block Diagrams

A **block diagram** is a simple drawing that shows the main parts (blocks) of a system and how they connect with each other. It is a visual tool that helps understand the architecture without needing to look at complex circuit details.

Common Blocks in Microcontroller / SBC Architecture

Here are the key components you will find in most microcontrollers and SBCs:

a) Central Processing Unit (CPU)

- The CPU is the "brain" of the microcontroller or SBC.
- It performs all the instructions and calculations.
- It controls how data moves inside the system.

b) Memory

- **Program Memory (Flash Memory):**
  Stores the program or instructions that the microcontroller runs.
- **Data Memory (RAM):**
  Temporarily stores data and variables while the program is running.
- **EEPROM/ROM:**
  Stores permanent data that should not be lost when power is off.

c) Input/Output (I/O) Ports

- These ports connect the microcontroller to the outside world.
- Input ports receive data from sensors (e.g., temperature sensor, IR sensor).
- Output ports send signals to devices like motors, LEDs, or displays.

d) Clock Generator

- Provides timing signals (clock pulses) that synchronize all operations inside the microcontroller.
- Determines how fast the CPU executes instructions.

e) Analog-to-Digital Converter (ADC)

- Converts Analog signals from sensors (like temperature or light sensors) into digital data that the CPU can understand.

f) Timers and Counters

- Used to keep track of time or count events.
- Important for generating precise delays or controlling the timing of robot actions.

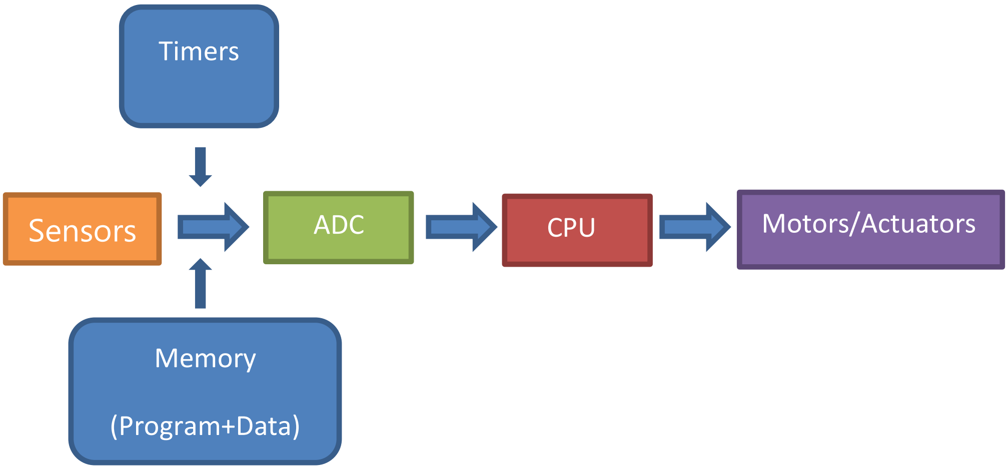
g) Communication Interfaces

- Allow the microcontroller or SBC to communicate with other devices or computers.
- Common interfaces include UART, SPI, I2C, USB, and Ethernet.

How Does It All Work Together?

1. **Sensors collect information** from the environment (like distance, temperature, light).
2. Sensors send signals to the microcontroller's **input ports**.
3. If the signal is Analog, the **ADC converts** it into digital data.
4. The **CPU processes** the data by following the program stored in memory.
5. Based on the program, the CPU sends commands through the **output ports** to control motors, LEDs, or other devices.
6. **Timers** help in performing actions at specific intervals or for specific durations.
7. Communication interfaces allow the robot to exchange data with other systems or networks.

Block Diagram of a Microcontroller-Based Robot Control System



Difference between Microcontroller and SBC Architecture

Feature	Microcontroller	Single Board Computer (SBC)
Size	Small, compact	Larger, full computer on one board
Processing Power	Lower, designed for simple tasks	Higher, can run complex OS and apps
Memory	Limited program and data memory	Larger RAM and storage options
Applications	Simple control tasks (robots, appliances)	Complex processing, AI, multimedia

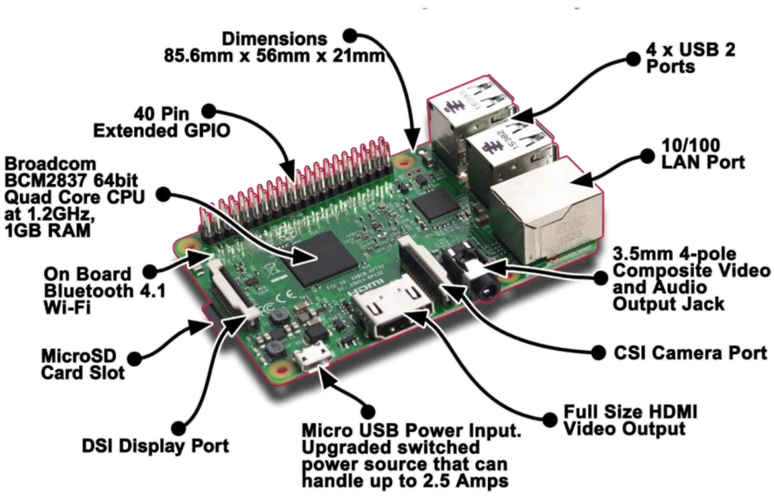
Single Board Computer (SBC)

A **Single Board Computer (SBC)** is a complete computer built on a single circuit board. It includes a **processor (CPU)**, **memory (RAM)**, **storage**, **input/output ports**, and sometimes even **networking features** like Wi-Fi and Bluetooth. Unlike traditional computers, which have separate components for different tasks, an SBC combines everything into one compact and integrated unit. This design makes SBCs small, portable, energy-efficient, and ideal for embedded systems, learning platforms, and smart devices.

SBCs are widely used in robotics, home automation, media centres, industrial control systems, and educational tools. They are especially useful in projects where space and power are limited. Many SBCs come with GPIO (General Purpose Input/Output) pins that allow users to connect sensors, motors, LEDs, and other hardware components, making them perfect for electronics and robotics prototyping.

Raspberry Pi: A Popular Example of an SBC

One of the most well-known and widely used SBCs is the **Raspberry Pi**. It was developed by the **Raspberry Pi Foundation** in the UK to promote computer science education. Today, it is used not only in classrooms but also in industries, research, and personal projects. Due to its low cost, powerful features, and strong community support, Raspberry Pi has become the go-to platform for hobbyists, students, and developers working on real-world applications.



The Raspberry Pi is a **credit card-sized computer** that can run a full operating system like a desktop or laptop. It allows users to browse the internet, run programs, play videos, and even control electronic devices. Its main advantage is that it brings the power of a computer into a tiny and affordable package, which anyone can use for learning or building projects.

Features

The Raspberry Pi board includes several important components:

- A **processor (CPU)** for handling instructions
- **RAM** for temporary memory storage
- A **micro SD card slot** for permanent storage
- **USB ports** to connect a keyboard, mouse, or USB devices
- **HDMI port** to connect to a display
- **Camera and display connectors**
- **Audio output**
- **GPIO pins** to connect with electronic components
- **Networking options** like Wi-Fi, Bluetooth, or Ethernet

This setup allows Raspberry Pi to work like a regular computer, and also interact with the physical world like a microcontroller.

Software and Programming

Raspberry Pi usually runs **Raspberry Pi OS**, a Linux-based operating system that comes with pre-installed applications such as **Python**, **Scratch**, **LibreOffice**, and a web browser. It supports other operating systems too, like **Ubuntu**, **Windows IoT Core**, and custom systems for gaming or media centres. Users can program the Pi in various languages like **Python, C++, Java, and JavaScript**, making it a versatile tool for both beginners and experts.

The Raspberry Pi is commonly used as the "brain" of a robot. It can read inputs from sensors (like ultrasonic, IR, or cameras), process the data, and send commands to motors and other output devices. For example, a robot built using a Raspberry Pi can:

- Avoid obstacles
- Follow lines or paths
- Stream live video
- Respond to voice commands
- Connect to the cloud for remote control

Because of its computing power, it can also handle advanced tasks like **image recognition**, **speech processing**, or **machine learning**, which are difficult to achieve with simple microcontrollers.

Variants and Models

There are several versions of Raspberry Pi available:

- **Raspberry Pi 4 Model B**: High-performance board with up to 8GB RAM, dual HDMI output, and USB 3.0
- **Raspberry Pi 3 Model B+**: Affordable and widely used in education
- **Raspberry Pi Zero / Zero W**: Ultra-small and cheap boards used in compact projects
- **Raspberry Pi Pico**: A microcontroller board (not an SBC) used for simple electronics tasks

Real-World Applications

Raspberry Pi is used in many real-world applications:

- **Home automation** (e.g., smart lights, security cameras)
- **Media centres** using Kodi
- **IoT devices** for data logging and control
- **AI projects** using TensorFlow or OpenCV
- **Web servers** and mini databases
- **Educational robots** and programming kits

Chapter Highlights

- **Single Board Computers (SBCs)** are full computers on one board, like Raspberry Pi, used for advanced processing.
- **Architecture** shows how CPU, memory, input/output ports, ADCs, timers, and communication systems work together.
- **Block diagrams** help visually understand how components are connected and how data flows.
- **Key components** include:
  - **CPU** (brain of the system)
  - **Memory** (Flash, RAM, EEPROM)
  - **I/O Ports** (connect sensors and actuators)
  - **Clock Generator** (timing signals)
  - **ADC** (converts Analog to digital)
  - **Timers** (manage time-based actions)
  - **Communication Interfaces** (UART, SPI, I2C, USB)
- **Microcontrollers vs SBCs**:
  - Microcontrollers: simpler, no OS, used for control tasks
  - SBCs: powerful, run OS, used for complex applications

Exercise

Multiple-Choice Questions (MCQs)

1. What is the function of a microcontroller?
   a) Play games
   b) Perform control tasks in embedded systems
   c) Display web pages
   d) Store music
2. Which of the following is a type of Single Board Computer (SBC)?
   a) Arduino Uno
   b) ESP8266
   c) Raspberry Pi
   d) 555 Timer
3. What does the CPU in a microcontroller do?
   a) Stores music
   b) Connects to the internet
   c) Executes instructions and controls data flow
   d) Displays images
4. Which memory is used to store programs in a microcontroller?
   a) RAM

    b) EEPROM
    c) Flash memory
    d) Cache memory
5. What converts analog sensor signals into digital form in microcontrollers?
    a) Clock
    b) USB port
    c) Timer
    d) ADC
6. Which component provides timing signals for the microcontroller?
    a) CPU
    b) Timer
    c) Clock Generator
    d) EEPROM
7. Input/Output (I/O) ports are used to:
    a) Store programs
    b) Communicate with sensors and actuators
    c) Run an operating system
    d) Convert analog to digital
8. What is a key difference between a microcontroller and SBC?
    a) SBCs are smaller
    b) SBCs don't have memory
    c) SBCs run full operating systems
    d) Microcontrollers run Linux
9. Which of the following is **not** a communication interface?
    a) I2C
    b) UART
    c) Flash
    d) SPI
10. What does a block diagram help with?
    a) Shows screen output
    b) Shows how code runs
    c) Visually represents system components and connections
    d) Displays memory contents

True or False

1. Microcontrollers are larger than SBCs.
2. Timers help perform actions at specific intervals.
3. ADC converts digital data to analog.
4. SBCs can run full operating systems like Linux.
5. Block diagrams are used to write programs.

Fill in the Blanks

1. The _______ is known as the brain of a microcontroller.
2. _______ memory stores the running program in a microcontroller.

3. SBC stands for ______.
4. ______ ports help the microcontroller connect with the outside world.
5. ______ diagrams show the system's parts and their connections.

Assertion and Reason

1. **Assertion (A):** The CPU is called the "brain" of a microcontroller.
   Reason (R): The CPU controls instruction execution and data flow inside the system.
   A. Both A and R are true, and R is the correct explanation of A.
   B. Both A and R are true, but R is not the correct explanation of A.
   C. A is true, but R is false.
   D. A is false, but R is true.
2. **Assertion (A):** Flash memory is used for temporary data storage during program execution.
   Reason (R): RAM stores the program code in a microcontroller.
   A. Both A and R are true, and R is the correct explanation of A.
   B. Both A and R are true, but R is not the correct explanation of A.
   C. A is true, but R is false.
   D. A is false, but R is true.
3. **Assertion (A):** ADC is used to convert digital signals to analog in a microcontroller.
   Reason (R): Microcontrollers work with digital data, not analog.
   A. Both A and R are true, and R is the correct explanation of A.
   B. Both A and R are true, but R is not the correct explanation of A.
   C. A is true, but R is false.
   D. A is false, but R is true.
4. **Assertion (A):** SBCs are capable of running full operating systems like Linux.
   Reason (R): SBCs have more memory and processing power compared to microcontrollers.
   A. Both A and R are true, and R is the correct explanation of A.
   B. Both A and R are true, but R is not the correct explanation of A.
   C. A is true, but R is false.
   D. A is false, but R is true.
5. **Assertion (A):** Block diagrams are helpful in understanding the internal wiring of microcontrollers.
   Reason (R): Block diagrams show the key components and their connections at a conceptual level.
   A. Both A and R are true, and R is the correct explanation of A.
   B. Both A and R are true, but R is not the correct explanation of A.
   C. A is true, but R is false.
   D. A is false, but R is true.

Short Type Questions

1. What is the main difference between a microcontroller and a Single Board Computer?
2. Why is a clock generator important in a microcontroller?
3. List three key components of microcontroller architecture.
4. What is the role of the ADC in a robot’s control system?

5. How do input and output ports help in robotic tasks?

Long Type Questions

1. Explain how the components of a microcontroller work together to control a robot.
2. Describe the differences between microcontrollers and SBCs with examples.
3. What are block diagrams, and how do they help understand system architecture in robotics?
4. Write a detailed note on the communication interfaces used in microcontrollers.
5. Create and explain a block diagram of a microcontroller-based robot that senses and moves.