

Chapter 1:

Foundations of Smart Systems

Or

Driving Intelligence

Introduction:

Microcontrollers and microprocessors are vital components of modern electronics, playing a significant role in various applications. Microprocessors are single-chip CPUs designed for high-performance computing, commonly used in personal computers and servers. They require additional components to function as complete systems. On the other hand, microcontrollers integrate the CPU, memory, I/O peripherals, and other essential features on a single chip, making them suitable for specific tasks within embedded systems found in automotive systems, home appliances, and medical devices. Microcontrollers are optimized for low power consumption and real-time control, making them ideal for applications where cost and energy efficiency are crucial. With evolving technology, the distinctions between microcontrollers and microprocessors are becoming less clear as both types incorporate features from one another. Therefore, understanding their differences and applications is vital for designing efficient embedded systems, particularly in the context of the Internet of Things (IoT) and artificial intelligence (AI).

Microcontroller

A microcontroller (MCU) is a small computer on a single integrated circuit that is designed to control specific tasks within electronic systems. It combines the functions of a central processing unit (CPU), memory, and input/output interfaces, all on a single chip.

Microcontrollers are widely used in embedded systems.

A typical microcontroller consists of a processor core, volatile and non-volatile memory, input/output peripherals, and various communication interfaces. The processor core is responsible for executing instructions and controlling the other components of the microcontroller. The memory is used to store data and program code, while the input/output peripherals are used to interact with the external environment.

Microcontrollers are programmable, which means that they can be customized to perform specific tasks. The programming languages used to write code for microcontrollers vary depending on the manufacturer and the type of microcontroller. Some of the commonly used programming languages include C, C++, and assembly language.

Working of Microcontroller

The microcontroller chip is a high-speed device, yet it is slow when compared to a computer. As a result, each command will be executed quickly within the microcontroller. The quartz oscillator is enabled and through control logic register once the supply is powered on. Parasite capacitors will be recharged for a few seconds while the early preparation is taking place. Once the voltage level reaches its maximum value and the oscillator's frequency stabilizes, the operation of writing bits through special function registers becomes stable. Everything is controlled by the oscillator's CLK, and the whole electronics will begin to function. All of this happens in a matter of nanoseconds.

A microcontroller's major role is that it can be thought of as a self-contained system with a processor memory. Its peripherals can be used in the same way that an 8051 microcontroller can. The bulk of microcontrollers in use today are embedded in other types of machinery such as telephones, appliances, vehicles, and computer system peripherals.

Types of Microcontrollers

1. Arduino Uno

Arduino UNO is a microcontroller board based on the ATmega328P, as shown in figure 1. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button.

➤ Features of Arduino Uno

1. Microcontroller:

ATmega328P: This 8-bit microcontroller operates at a clock speed of 16 MHz and is known for its reliability and efficiency.

2. I/O Pins:

- **Digital Pins:** The Arduino UNO has 14 digital input/output pins that can be configured as either inputs or outputs. Out of these, 6 pins support PWM (Pulse Width Modulation) to simulate analog outputs.
- **Analog Pins:** There are 6 analog input pins that can read varying voltages, allowing for the connection of sensors that output analog signals.

3. Memory:

- **Flash Memory:** 32 KB, of which 0.5 KB is used for the bootloader. This memory is used to store the program code.
- **SRAM:** 2 KB, used for runtime data storage.
- **EEPROM:** 1 KB, which allows for storing data that persists even when the board is powered off.

4. Power Supply Options:

The board can be powered via USB or an external power supply. The recommended voltage for external power is between 7-12V, while USB supplies 5V.

5. USB Interface:

The Arduino UNO features a USB-B connector for programming and serial communication with a computer. It allows easy upload of code from the Arduino IDE.

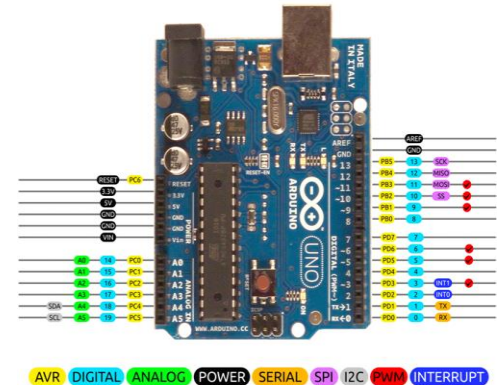


Figure 1. Arduino UNO.

6. Development Environment:

The board is programmed using the Arduino Integrated Development Environment (IDE), which simplifies the coding process and includes a variety of libraries for different functions and sensors.

7. Built-in LED:

The board includes a built-in LED connected to digital pin 13, which is useful for testing and debugging.

8. Shield Compatibility:

The UNO supports various shields, which are stackable circuit boards that extend its functionality (e.g., Ethernet, Wi-Fi, GPS, LCD displays).

9. Reset Button:

A reset button allows users to restart the program without disconnecting the power supply.

10. Size and Form Factor:

The compact form factor (about 68.6 mm x 53.4 mm) makes it easy to integrate into various projects and prototypes.

➤ Advantages of Arduino Uno

- **User-Friendly:** The Arduino IDE is accessible for beginners, allowing them to quickly learn programming and circuit design.
- **Extensive Community Support:** A large and active community provides a wealth of resources, including forums, tutorials, and project examples.
- **Cost-Effective:** The Arduino UNO is affordable, making it a popular choice among hobbyists, educators, and professionals.
- **Versatile:** It can interface with a wide range of sensors, actuators, and other devices, making it suitable for diverse applications.
- **Open-Source:** The Arduino platform is open-source, encouraging collaboration and innovation in the maker community.

2. Arduino Mega 2560

Arduino Mega 2560 is a 32-bit microcontroller board based on the Atmega2560 chip, as shown in figure 2. This board is a good match for projects that require more GPIO pins and memory space because it carries 16 analog pins and 54 digital I/O pins out of which 15 pins

are used for PWM output. The board comes with a DC power jack to power up this unit and you can also turn on the board using VIN pin on the board. The unit also supports a USB interface where a USB cable is used to connect the board with the computer.

The Arduino Mega 2560 is similar to Arduino UNO but comes with more GPIO pins, more memory space, and is bigger in size.

The unit also supports the ICSP header which is used to program the board without disconnecting it from the main circuitry.

Two voltage regulators are included on the board through which you can regulate the voltage as you like better.

Arduino Mega 2560 is programmed using Arduino IDE (Integrated Development Environment) software that is the official software introduced by Arduino.cc

The ATmega2560 controller on the board comes with 256 KB of flash memory used for storing code (out of which 8 KB is used for the Bootloader), while the SRAM is 8 KB of SRAM and EEPROM is 4 KB of EEPROM.

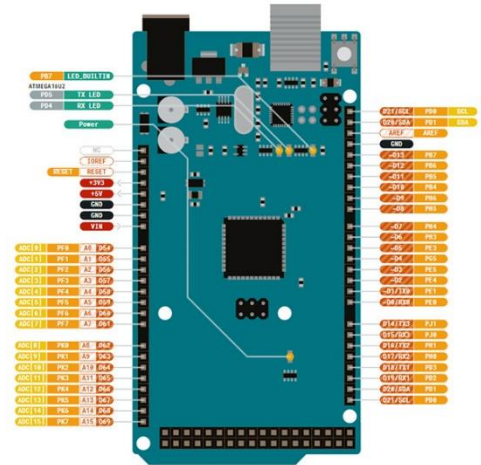


Figure 2. Arduino Mega2560.

➤ Arduino Mega 2560 Pin Description

In this section, we'll cover the pin description of each pin incorporated on the board.

- **Digital I/O Pins:** There are total of 54 digital I/O pins available on the board which can be used to connect the board with external components.
- **PWM:** 15 pins are used for PWM which is a process used to control the speed of the motor or brightness of the LED.
- **LED:** This is the built-in LED connected to pin 13. When 5V is provided to this pin, it will turn ON the LED while ground or zero V will turn it OFF.
- **Analogue Pins:** There are 16 analogue pins incorporated on the board marked as A0 to A15. These pins can measure voltage from ground to 5V and each pin is a 10-bit resolution pin.
- **GND:** This board carries 5 ground pins which are used for projects where more than one ground is required.
- **External Interrupts:** Six pins are reserved for generating external interrupts. Those are pin number 0, 3, 18, 19, 20 & 21.
- **Reset:** This is the reset pin of the board. This pin is useful when your code gets stuck in the middle of the running program, pressing this pin will reset the code compiled into the board.
- **Vin:** This is the input voltage of the board which ranges from 6V to 12V, however, recommended input voltage ranges from 7V to 12V.
- **AREF:** This is the analogue reference voltage that is a reference voltage for the analogue inputs.
- **USART Communication:** The board comes with USART serial communication where two pins TX and RX are used for the transmission and receiving of serial data.

- **SPI Communication:** The device supports SPI (serial peripheral interface) communication which allows the transmission of data between the controller and other peripheral devices.
- **I2C Communication:** The unit supports the I2C serial communication protocol where two pins 20 & 21 are reserved for this communication. The 20 is an SDA pin which is a serial data line used for holding the data and the 21 is an SCL pin which is a serial clock line used employed for offering data synchronization between the devices.

3. STM32F103C8T6

The STM32F103C8T6 is a 32-bit microcontroller based on the ARM Cortex-M3 core, running at 72 MHz with up to 128 Kbytes of Flash memory and 20 Kbytes of SRAM, as shown in figure 3. It offers two 12-bit ADCs, three 16-bit timers, a PWM timer, and multiple communication interfaces, including I2C, SPI, USART, USB, and CAN. Operating between 2.0 to 3.6 V, the devices are designed for low-power applications and are available in temperature ranges of -40 to +105°C. With packages ranging from 36 to 100 pins, they are ideal for applications like motor drives, medical devices, industrial control, and GPS platforms.

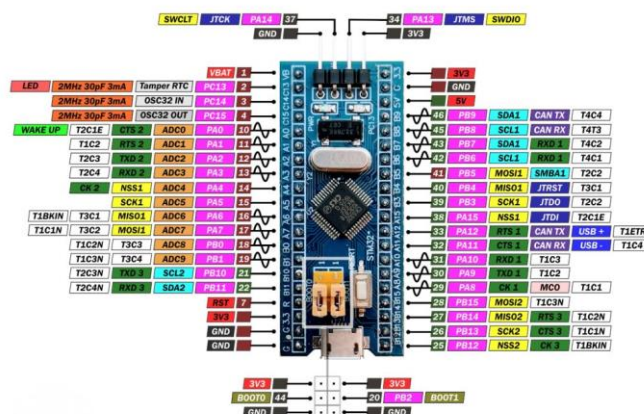


Figure 3. STM32F103C8T6.

Key Features and Specifications:

- **CPU:** ARM Cortex-M3, running at up to 72 MHz
- **Memory:** 64 KB Flash memory and 20 KB SRAM
- **I/O:** 48 pins, including various types like GPIO, analog inputs, and communication interfaces
- **Communication:** I2C, SPI, UART, CAN, USB 2.0 full-speed
- **Timers:** 3x 16-bit general-purpose timers, 1x 16-bit motor control PWM timer
- **ADC:** 2x 12-bit, 1 μ s analog-to-digital converters
- **Other features:** RTC, DMA controller, watchdog timers, and more

Advantages:

- **Cost-effective:** Offers a good balance of performance and price
- **Low power consumption:** Ideal for battery-powered applications
- **Rich feature set:** Provides a wide range of peripherals and functionalities
- **Extensive development ecosystem:** Supported by various tools, libraries, and communities
- **Flexibility:** Can be used in a variety of applications

Disadvantages:

- **Limited performance compared to newer Cortex-M architectures:** May not be suitable for extremely demanding tasks
- **Smaller memory size:** Might restrict the complexity of applications

Additional Information:

- **Development boards:** Many popular development boards like the Blue Pill and STM32duino are based on the STM32F103C8T6, making it easy to start experimenting.
- **Programming languages:** Can be programmed in C, C++, and other languages using tools like Keil vision, STM32CubeIDE, and Arduino IDE (for compatible boards).
- **Resources:** STMicroelectronics provides comprehensive documentation, tutorials, and software libraries to support development.

4. Raspberry Pi Pico

The Raspberry Pi Pico is a compact, low-cost microcontroller board, designed for embedded applications, physical computing, and IoT projects. Unlike other Raspberry Pi boards, which function as full computers, the Pico is meant for controlling hardware and handling low-level operations. It is based on the RP2040 microcontroller chip, as shown in figure 4.

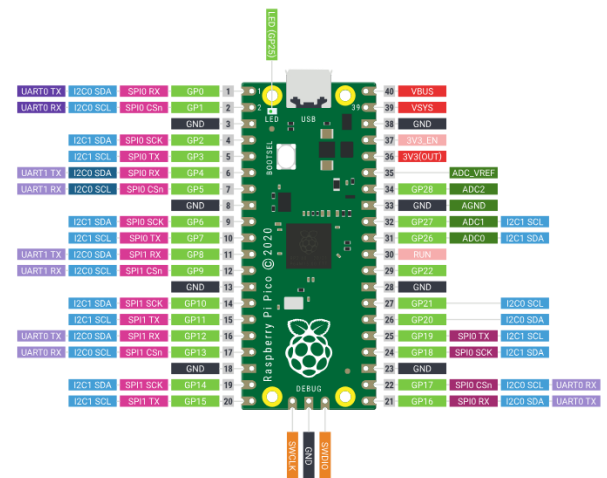


Figure 4. Raspberry Pi Pico.

Key Specifications of Raspberry Pi Pico**1. Microcontroller Chip:**

- **RP2040:** A custom-made, dual-core ARM Cortex-M0+ microcontroller running at 133 MHz, making it suitable for simple control tasks and low-power applications.
- **Architecture:** ARM Cortex-M0+ is a 32-bit RISC processor optimized for low power, ideal for microcontroller applications where efficiency is key.

2. Memory:

- **RAM:** 264 KB of SRAM, enough to handle most microcontroller tasks and data buffering.
- **Flash Storage:** 2 MB of onboard QSPI flash memory for storing programs and data.

3. Connectivity:

- The Pico lacks built-in Wi-Fi or Bluetooth, focusing instead on low-power, direct control applications. However, it can connect to external devices via various communication interfaces:
 - **UART, SPI, and I2C:** Common protocols for communicating with other microcontrollers, sensors, and peripherals.

4. GPIO (General Purpose Input/Output):

- 26 multi-function GPIO pins, including three analog inputs (ADC). These pins can be used to control LEDs, motors, sensors, and other electronic components.
- PIO (Programmable I/O): The RP2040 includes two programmable I/O blocks, allowing for custom protocols and timing-critical functions, which enhances its versatility in handling custom input/output tasks.

5. Power Supply:

- The Pico can be powered via micro-USB (up to 5V), which also allows for programming and serial communication with a computer. It also has a 3.3V output for powering external components and sensors.

6. Programming:

- The Raspberry Pi Pico supports programming in MicroPython and C/C++, making it accessible to both beginners and advanced users.
- **MicroPython:** A simplified version of Python designed for microcontrollers, which allows for easy, high-level programming.
- **C/C++:** For more performance-intensive applications, programming in C or C++ provides fine-grained control over the hardware.

Advantages of Raspberry Pi Pico

- **Low Cost:** At around \$4 USD, the Pico is one of the most affordable microcontrollers available, making it accessible for beginners and hobbyists.
- **Energy Efficient:** ARM Cortex-M0+ is optimized for low power, ideal for battery-operated projects or devices needing long standby times.
- **Flexible GPIO:** With programmable I/O and versatile GPIO pins, it offers great flexibility for controlling various external components.
- **Open Source Support:** Extensive documentation and community support, including tutorials and libraries for common tasks.

Microprocessors

Microprocessor is a controlling unit of a micro-computer, fabricated on a small chip capable of performing ALU (Arithmetic Logical Unit) operations and communicating with the other devices connected to it.

Microprocessor consists of an ALU, register array, and a control unit. ALU performs arithmetical and logical operations on the data received from the memory or an input device. Register array consists of registers identified by letters like B, C, D, E, H, L and accumulator. The control unit controls the flow of data and instructions within the computer.

Working of Microprocessors

Modern microprocessors are essentially tiny chips made up of millions of transistors, resistors, and diodes. These components are arranged in specific configurations, known as computer architecture, to perform calculations and execute instructions.

Microprocessors work by fetching instructions from memory, decoding them, and executing them. This process is repeated continuously, allowing the microprocessor to perform various tasks like calculations, data movement, and decision-making. The working process can be divided into the following stages:

1. **Fetch:** The microprocessor retrieves (fetches) the next instruction from the program memory (RAM or ROM).
2. **Decode:** The fetched instruction is decoded by the microprocessor's control unit. The microprocessor interprets the instruction to understand what operation needs to be performed.
3. **Execute:** The microprocessor performs the action specified by the instruction. This could be an arithmetic operation (like addition or subtraction), logical operations (AND, OR, etc.), or controlling peripheral devices.
4. **Store:** If the instruction requires storing data (such as the result of a calculation), the microprocessor writes the data back to memory or registers.

This **fetch-decode-execute** cycle is repeated continuously as the microprocessor processes instructions from the running program.

Components of a Microprocessor

1. **Arithmetic Logic Unit (ALU):** Performs arithmetic and logical operations, such as addition, subtraction, and comparisons.
2. **Control Unit (CU):** Decodes instructions and directs the operation of the other components of the microprocessor.
3. **Registers:** Small, fast storage areas inside the microprocessor that temporarily hold data and instructions.
4. **Clock:** Synchronizes the operation of the microprocessor by providing a timing signal, determining how fast instructions are executed.
5. **Bus:** A system of communication pathways that allows data to move between the microprocessor, memory, and I/O devices.

Figure 5. Raspberry Pi 4 model B

- **USB Boot Support:** Supports booting from an external USB storage device, including SSDs, for faster read/write speeds and extended storage.

6. GPIO (General Purpose Input/Output) Pins:

- 40-pin GPIO header, enabling direct connection to various sensors, displays, and other hardware components, making it ideal for electronics projects and prototyping.

7. Operating System:

- **Raspberry Pi OS (formerly Raspbian):** A Debian-based Linux distribution optimized for the Raspberry Pi, though it also supports Ubuntu, Windows IoT Core, and many specialized OSes for specific applications.

Advantages of the Raspberry Pi 4 Model B

- **Affordable:** Low-cost entry to computing and prototyping.
- **High Performance:** Quad-core processor and large RAM options enable multitasking and demanding applications.
- **Connectivity:** Gigabit Ethernet, USB 3.0, and wireless connectivity provide a robust platform for IoT and networking projects.
- **Versatile OS Support:** Compatible with many operating systems, giving users flexibility for specific applications.
- **Community Support:** A large community offers extensive documentation, tutorials, and resources, especially valuable for beginners.