Microcontrollers AVR Versus ARM versus PIC versus 8051

A micro-controller can be comparable to a little stand alone computer; it is an extremely powerful device, which is able of executing a series of pre-programmed tasks and interacting with extra hardware devices. Being packed in a tiny integrated circuit (IC) whose size and weight is regularly negligible, it is becoming the perfect controller for as robots or any machines required some type of intelligent automation. A single microcontroller can be enough to manage a small mobile robot, an automatic washer machine or a security system. Several microcontrollers contains a memory to store the program to be executed, and a lot of input/output lines that can be a used to act jointly with other devices, like reading the state of a sensor or controlling a motor.

Microcontrollers are small, low-cost, and programmable devices that can control other hardware components, such as sensors, motors, LEDs, and displays. They are often used for embedded systems, which are specialized computers that perform specific tasks within a larger system. Microcontrollers have a CPU, memory, input/output ports, and sometimes other peripherals, such as timers, ADCs, and communication modules.

The main parts that a microcontroller has are: central processing unit (CPU), memory unit, and I/O interfaces. All these components of the microcontroller are formed on a **single chip**. A microcontroller can be programmed using various programming languages like C language, assembly language, etc. to perform a specific function. Microcontrollers are widely used in a variety of electronic devices, such as medical instruments, manufacturing machinery, robotics, automobiles, home appliances, toys, etc. to automate their operation.

**What is an 8051 Microcontroller?**

The microcontroller like 8051 was designed in the year 1981 by Intel. The microcontroller is one kind of integrated circuit that includes 40-pins with dual inline package or DIP, RAM-128 bytes, ROM-4kb & 16-bit timers–2. Based on the requirement, it includes addressable & programmable 4 – parallel 8-bit ports. In the 8051 microcontroller architecture, the system bus plays a key role to connect all the devices to the central processing unit. This bus includes a data bus- an 8-bit, an address bus-16-bit & bus control signals. Other devices can also be interfaced throughout the system bus like ports, memory, interrupt control, serial interface, the CPU, timers.

There are two buses in 8051 Microcontroller one for the program and another for data. As a result, it has two storage rooms for both programs and data of 64K by 8 sizes. The microcontroller comprises of 8-bit accumulator & an 8-bit processing unit. It also consists of 8 bit B register as majorly functioning blocks and 8051 microcontroller programming is done with [embedded C language](https://www.elprocus.com/basics-and-structure-of-embedded-c-program-with-examples-for-beginners/) using Keil software. It also has several other 8 bit and 16-bit registers.

The 8051 has been in utilized in a wide number of devices, mostly because it is easy to integrate into a project or make a device approximately. The following are the major areas of focus:

**Energy Management:** Efficient metering systems facilitate in controlling energy usage in homes and manufacturing applications. These metering systems are prepared capable by incorporating microcontrollers.

**Touch screens:** A high number of microcontroller providers incorporate touch-sensing capabilities in their designs. Portable electronics such as cell phones, media players and gaming devices are examples of microcontroller-based touch screens.

**Automobiles:** The 8051 finds wide taking in providing automobile solutions. They are broadly used in hybrid vehicles to handle engine variants. Furthermore, functions such as cruise control and anti-brake system have been prepared more capable with the use of microcontrollers.

**Medical Devices:** Moveable medical devices such as blood pressure and glucose monitors use microcontrollers will to show data, thus provided that higher reliability in providing medical results.

## PIC microcontrollers

Peripheral Interface Controller (PIC) is microcontroller developed by a Microchip, [PIC microcontroller](https://www.elprocus.com/introduction-to-pic-microcontrollers-and-its-architecture/)is fast and simple to implement program when we contrast other microcontrollers. The ease of programming and simple to interfacing with other peripherals PIC become successful microcontroller.

We know that microcontroller is an integrated chip which is consists of RAM, ROM, CPU, [TIMER and COUNTERS](https://www.elprocus.com/8051-microcontroller-8-16-bit-timers-and-counters/). The PIC is a microcontroller which as well consists of RAM, ROM, CPU, timer, counter, ADC ([analog to digital converters](https://www.elprocus.com/analog-to-digital-converter/)), DAC (digital to analog converter). PIC Microcontroller also support the protocols like CAN, SPI, UART for an interfacing with additional peripherals. PIC mostly used to modify Harvard architecture and also supports [RISC (Reduced Instruction Set Computer)](https://www.elprocus.com/what-is-risc-and-cisc-architecture-and-their-workings/) by the requirement RISC and Harvard we can simply that PIC is faster than the 8051 based controllers which is prepared up of Von-Newman architecture.

PIC is a family of 8-bit and 32-bit microcontrollers developed by Microchip, based on the Harvard architecture, which provides separate memory spaces for data and instructions. These microcontrollers are often used for industrial and automotive applications, as well as for hobby projects. PIC microcontrollers have a high reliability and robustness with a long lifespan, as well as a rich set of peripherals and modules such as PWM, UART, SPI, I2C, and USB. Additionally, they have a low cost per unit and a high availability in the market. However, they can be difficult to program and debug due to their steep learning curve and complex instruction set. Furthermore, they have limited code compatibility and portability, leading to less flexibility and adaptability to different projects. Finally, they have a higher power consumption and lower performance than some other microcontrollers.

## What is an AVR Microcontroller?

**AVR** is an abbreviation for **Alf and Vegard’s RISC processor, also Advanced Virtual RISC.** It is named in the honor of its developers, Alf-Egil Bogen and Vegard Wollan. AVR is a RISC (Reduced Instruction Set Computer) based microcontroller architecture. It was first produced by Atmel Corporation in the year of 1997.

The AT90S8515 was the first microcontroller developed based on the AVR microcontroller architecture. AVR microcontrollers have simple instruction sets, making them fast and efficient. The major advantages of AVR microcontrollers include low power consumption, low cost, and high performance. We can use assemble language as well as high-level languages like C, C++, etc. to program these microcontrollers for a specific function.

AVR microcontrollers are widely used in several different applications like robotics, home and office appliances, industrial automation systems, automobiles, etc.

## What is an ARM Microcontroller?

**ARM** is the abbreviation for **Advanced RISC Machine. ARM microcontroller** is a 32-bit architecture microcontroller that was developed by Acorn Computers in 1983. ARM is basically a family of Reduced Instruction Set Computing (RISC) architecture-based microprocessors. ARM microcontrollers consist of ARM processors, RAM, ROM, and I/O peripherals. ARM microcontrollers are used in a wide range of applications due to their low power consumption, low cost, and high performance.

One of the important features of ARM microcontrollers is that they are highly customizable depending on requirements of the applications. Therefore, it is highly versatile microcontroller architecture.

We can use assembly language as well as high level programming languages such as C, C++ to program the ARM microcontrollers. ARM microcontrollers are highly scalable; hence they can be used in several applications, from simple embedded systems to high-end computing systems.

ARM is a family of 32-bit and 64-bit microcontrollers developed by ARM Holdings, which licenses its technology to various manufacturers such as STMicroelectronics, NXP, and Texas Instruments. The ARM architecture is a family of RISC architectures that can vary in features and performance, making **ARM microcontrollers suitable for high-end applications such as smartphones, tablets, and wearables. They boast a high speed and resolution, scalability and compatibility, as well as low power consumption and high efficiency**. However, they can also be **complex and difficult to program and debug due to their sophisticated instruction set and multi-layered architecture. Additionally, ARM microcontrollers are costly and have limited availability in the market**, making them incompatible with other microcontrollers which can make them less suitable for mixed projects.

After getting an overview of AVR and ARM individually, let us now discuss their important differences.

Difference between AVR and ARM

|  |  |  |
| --- | --- | --- |
| **Parameter** | **AVR Microcontroller** | **ARM Microcontroller** |
| **Basic** | AVR stands for "Alf and Vegard’s RISC processor" or "Advanced Virtual RISC". | ARM stands for Advanced RISC Machine. |
| **Bus width** | AVR microcontrollers have a bus width of 8 bits. It also available in 32 bits bus width. | ARM microcontrollers have a bus width of 32 bits. It also available in 64 bits. |
| **Developer** | AVR microcontroller was developed by Atmel Corporation. | ARM microcontroller was developed by Acorn Computers. |
| **Release date** | AVR microcontroller was released in 1997. | ARM microcontroller was released in 1983. |
| **Communication protocols** | AVR microcontrollers use UART, USART, SPI, and I2C communication protocols. | ARM Microcontrollers use UART, USART, I2C, I2S, LIN, CAN, USB, Ethernet, SAI, and DSP communication protocols. |
| **Memory** | AVR microcontrollers use SRAM, Flash Memory, and EEPROM. | ARM Microcontrollers use SDRAM, Flash Memory, and EEPROM. |
| **Unique features** | AVR microcontrollers are known for their low cost and high performance. | ARM microcontrollers are known for their high-speed operation. |
| **Peripherals** | AVR microcontrollers have a smaller number of built-in peripherals. | ARM microcontrollers have a greater number of built-in peripherals. |
| **Real time processing** | AVR microcontrollers are not much effective in real time processing applications. | ARM microcontrollers are suitable for real time processing applications. |
| **Community** | AVR microcontrollers have a very good community to provide support to developers. | ARM microcontrollers have a vast community that is more focused and specialized. |
| **Power consumption** | AVR microcontrollers consume less power. | ARM microcontrollers consume slightly more power than AVR. |
| **Cost** | AVR microcontrollers are relatively less expensive. | ARM microcontrollers are more expensive than AVR. |
| **Popular microcontrollers** | Some popular microcontrollers of AVR family are ATmega 8/16/32, Arduino community, etc. | Some popular microcontrollers of ARM family are ARM Cortex-M0 to ARM Cortex M7, LPC2148, etc. |
| **Applications** | AVR microcontrollers are used in applications like robotics, home and office appliances, industrial automation systems, automobiles, etc. | ARM microcontrollers are used in a wide range of applications, from simple embedded systems to high-end computing systems. |

**Main Difference between AVR, ARM, 8051 and PIC Microcontrollers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **8051** | **PIC** | **AVR** | **ARM** |
| **Bus width** | 8-bit for standard core | 8/16/32-bit | 8/32-bit | 32-bit mostly also available in 64-bit |
| **Communication Protocols** | UART, USART,SPI,I2C | PIC, UART, USART, LIN, CAN, Ethernet, SPI, I2S | UART, USART, SPI, I2C, (special purpose AVR support CAN, USB, Ethernet) | UART, USART, LIN, I2C, SPI, CAN, USB, Ethernet, I2S, DSP, SAI (serial audio interface), IrDA |
| **Speed** | 12 Clock/instruction cycle | 4 Clock/instruction cycle | 1 clock/  instruction cycle | 1 clock/ instruction cycle |
| **Memory** | ROM, SRAM, FLASH | SRAM, FLASH | Flash, SRAM, EEPROM | Flash, SDRAM, EEPROM |
| **ISA** | CLSC | Some feature of RISC | RISC | RISC |
| **Memory Architecture** | Harvard architecture | Von Neumann architecture | Modified | Modified Harvard architecture |
| **Power Consumption** | Average | Low | Low | Low |
| **Families** | 8051 variants | PIC16,PIC17, PIC18, PIC24, PIC32 | Tiny, Atmega, Xmega, special purpose AVR | ARMv4,5,6,7 and series |
| **Manufacturer** | NXP, Atmel, Silicon Labs, Dallas, Cyprus, Infineon, etc. | Microchip Average | Atmel | Apple, Nvidia, Qualcomm, Samsung Electronics, and TI etc. |
| **Cost (as compared to features provide)** | Very Low | Average | Average | Average to high |
| **Other Feature** | Known for its Standard | Cheap | Cheap, effective | High speed operation  Vast |
| **Popular Microcontrollers** | AT89C51, P89v51, etc. | PIC18fXX8, PIC16f88X, PIC32MXX | Atmega8, 16, 32, Arduino Community | LPC2148, ARM Cortex-M0 to ARM Cortex-M7, etc. |