Click to prove you're human

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.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.MicrocontrollerA microcontroller is a self-contained desktop that can be utilized in an embedded system. A few microcontrollers may run at clock rate rates and use four-bit expressions. Because many of the devices they control are battery-operated, microcontrollers must often be low-power. It is a small, low-cost computer built into a single chip. It is often used as part of an embedded system to control specific functions in electronic devices. A microcontroller includes:A processor that runs instructions.Memory to store programs and data.Input/Output ports to connect with buttons, sensors, screens, etc.Communication ports (to connect with other devices).Why Are Microcontrollers Useful?They are programmable, so you can make them do different tasks.Common programming languages used are C, C++, and assembly language.They are often low-power, which is great for devices that run on batteries.They are small, cheap, and perfect for controlling devices over a long time.Most microcontrollers are embedded, meaning they are built inside other devices.The microcontroller used in Embedded System. For example:• Security Systems• Laser Printers• Automation System• RoboticsMicrocontrollerWorking of MicrocontrollerThe 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 few 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 MicrocontrollerThe types of microcontroller are characterized by the bits, memory architecture, memory/devices and instruction set.Types Of MicrocontrollersClassification of Microcontrollers by Number of Bits8-bit MicrocontrollersThe internal bus is 8-bits wide.The ALU (Arithmetic Logic Unit) performs operations on 8 bits at a time.Example microcontrollers: Intel 8031/8051, PIC1x, Motorola MC68HC11.16-bit MicrocontrollersThe internal bus is 16-bits wide, providing better precision and performance than 8-bit.A 16-bit microcontroller can handle a wider range of numbers (0x0000 to 0xFFFF, or 0 to 65535) compared to the 8-bit range (0x00 to 0xFF, or 0 to 255).Example microcontrollers: Extended 8051XA, PIC2x, Intel 8096, Motorola MC68HC12.32-bit MicrocontrollersThese microcontrollers use 32-bit instructions for operations.Used in advanced applications like medical devices, engine control systems, office machines, and other embedded systems.Example microcontrollers: Intel/Atmel 251 family , PIC3x.Classification of Microcontrollers by Memory TypeEmbedded Memory MicrocontrollerAll functional blocks are built into the chip.Includes program memory, data memory, I/O ports, serial communication, counters, timers, and interrupts.Example: 8051 microcontroller (has everything on the chip).External Memory MicrocontrollerThe microcontroller does not have all functional blocks on the chip.Some components, like program memory, need to be connected externally.Example: 8031 microcontroller (does not have program memory on the chip).Classification of Microcontrollers by Instruction SetCISC (Complex Instruction Set Computer)Allows complex instructions that can do multiple tasks in one command.Reduces the need for many simpler instructions.Example: CISC systems shorten execution time by reducing the number of instructions in a program.RISC (Reduced Instruction Set Computer)Simplifies the instruction set, using fewer, simpler commands.Allows instructions to work on any register and access program and data simultaneously.Shortens execution time by reducing clock cycles per instruction.RISC systems generally offer better execution performance compared to CISC systems.Classification of Microcontrollers by Memory ArchitectureHarvard Memory Architecture MicrocontrollerHas separate memory for program and data.Allows simultaneous access to both program and data memory, improving efficiency.Common in microcontrollers that require faster data processing.Von Neumann Memory Architecture MicrocontrollerUses shared memory for both program and data.Simplifies design but can be slower since the program and data share the same bus.Microcontroller PropertiesMicrocontroller devices are capable of having words longer than 64 bits.Microcontroller consist of RAM , ROM , Timer , I/O Ports.Microcontroller ROM is used for program storage and RAM is used for data storage.It is designed by using CISC architecture.The power consumption of modern microcontrollers is significantly lower and have operating voltage range from 1.8V to 5.5VThe latest feature of microcontroller is flash memory like EPROM and EEPROM.The most recent feature of a microcontroller is flash memory, such as EPROM and EEPROM.For more about microcontrollers, please refer Advantages and Disadvantages of the microcontroller article.Uses of MicrocontrollerMicrocontrollers are used in a wide range of electronic devices and systems given below:Everyday Uses of Microcontrollers:Devices that detect and control light (like automatic street lights).Devices that monitor and adjust temperature (like smart thermostats).Fire detection and safety systems (like smoke alarms).Tools used in factories to measure and monitor things.Systems that control industrial processes automatically.Uses of Microcontrollers in Industrial Control:Devices that measure and track industrial data.Systems that manage and control production processes.Uses of Microcontrollers in Measuring Devices:Voltage meters.Devices that count rotating parts (like in motors).Current meters.Portable measuring tools (like digital multimeters).Issues in Microcontroller Some of the most common issues that can arise with microcontrollers:Timing IssuesMicrocontrollers need precise timing to run tasks correctly.Any timing errors can cause malfunctions that are often hard to detect and fix.Power IssuesThey need a stable power supply.Power fluctuations or drops can cause the microcontroller to stop working or behave unpredictably.Heat IssuesMicrocontrollers generate heat during use.Too much heat can damage the chip or cause it to fail.Caused by poor design, bad cooling, or high room temperatures.Noise IssuesCan be affected by electromagnetic interference (EMI) and radio frequency interference (RFI) from nearby devices.This can lead to errors or unexpected behavior.Code IssuesThe software or code running on the microcontroller may have bugs or errors.This can lead to malfunctions or failure of the system.Security IssuesMicrocontrollers can be hacked or infected with malware.Risks include unauthorized access, data theft, or system control by attackers.Compatibility IssuesMay not work well with other hardware or components.Can result in errors or system failures. What is a Microcontroller? A microcontroller is a semiconductor IC (integrated circuit) chip consists of multiple functional block such as CPU, ROM, RAM, Timer, Counter & interrupts etc. It is essentially a small computer with all peripheral inside a single package. Classification of Microcontrollers Different types of microcontrollers are being used for automatic control of industries, automobiles, electronic medical devices, appliance & many other embedded systems. They are mostly preferred due to their compact size, low power consumption & low cost. The microcontrollers are classified into different types on various basis: On the Basis of Bus-Width Bus Width: The Bus in a microcontroller refers to the parallel lines of connection between various components of the microcontroller. It transmits the instructions & data between the CPU, Memory & input/output ports. There are three types of buses inside a microcontroller; data bus, address bus, control bus. The bus width refers to the number of parallel lines or wires or connections present in the said bus. Increasing the bus-width of a microcontroller increases its precision & its overall performance of the microcontroller. On the basis of bus width, the microcontrollers are divided into 8-Bit, 16-bit & 32-bit microcontrollers. 8-bits Microcontroller: The bus width of such microcontrollers is 8 bit (1 byte) wide long. It means it can transfer & process the data of 8 bits in a single cycle. The main hindrance it poses is in the mathematical operations because its ALU (arithmetic logic unit) is also 8-bit. In order to process a large data of for example 16 bit, it uses multiple cycles to complete a simple mathematical function. It results in a poor performance of the overall logic circuit. Another key feature of 8bit microcontroller is its Timer. An 8-bit timer has a maximum range of 0x00 (0) to 0xFF (255). It can cause inaccuracy in generating time delay functions. Some of the common 8-bit microcontrollers are Intel 8031/8051, PIC1x and Motorola MC68HC11 families 16-bit Microcontroller: The bus width of such microcontroller is 16 bit (2 byte) wide. It can transfer & process a data of 16 bits in a single cycle. Its 16-bit ALU is more efficient in performance comparing to 8-bit microcontroller. And its 16-bit timer provides a wide range of 0x0000 (0) to 0xFFFF (65535) which provides the best accuracy per cost for any application or projects that require Timer functions. Common examples of a 16-bit microcontrollers are 8051XA, PIC2x, Intel 8096 and Motorola MC68HC12 families. Related Post: Difference Between Microprocessor and Microcontroller 32-bit Microcontroller A 32-bit microcontroller has a bus width of 32 bits or 4 bytes long. The performance & accuracy of such microcontroller is higher than 16-bit microcontroller but they are also expensive & consume a lot of power. Its higher processing speed makes it the best candidate for performing a complex tasks such as audio & video signal processing, Image processing, etc. It can also support multiple peripheral required in any embedded system projects or applications such as Ethernet, Universal Serial Bus (USB), Universal Asynchronous Receiver-Transmitter devices (UARTS), and a Controller Area Network (CAN) bus. The common 32-bit microcontrollers are Intel/Atmel 251 family, PIC3x. Related Post: Difference Between 8085 & 8086 Microprocessor – Comparison The microcontroller is also classified into two types on the basis of essential memory blocks. Embedded Memory Microcontroller Such type of microcontroller has all the essential memory blocks or modules embedded inside a single package. Some of these functional blocks are program & data memory, Timers & counter, interrupts etc. These memory blocks are fixed & not expendables but a microcontroller having the feature of supporting external ROMs can extend its storage memory. These are compact in design. External Memory Microcontroller Such type of microcontroller does not have one of the essential memory blocks inside its chip & it needs to be connected externally to function properly. The use of external modules increases the size of the overall device. On the Basis of Instruction Set Architecture Instruction set or Instruction set architecture (ISA) is a part of microcontroller that commands the microprocessor to perform a specific function. The instruction set includes addressing modes, instructions, data types, registers, interrupts, and external I/O. Here are some of the Examples of instruction in a microcontroller; read, write, copy/move, add, subtract, multiply, divide, increment, decrement, compare, call to address location etc. A complex instruction set is made from combination of multiple instructions i.e. to execute a complex instruction; the microcontroller needs to execute various simple instructions. These instructions set are used for instruction that contains several steps including the use of several functional blocks. Some of the complex function in a microcontroller is moving data to & from multiple registers, copying strings of data, square root, log, sine etc. Related Post: Difference Between CPU and GPU – Comparison According to the instruction set, the microcontrollers are classified into two types; CISC (Complex Instruction Set Computer) Such type of microcontroller's CPU is designed to execute a single complex command. It can execute multiple instructions or steps using a single instruction. The advantage of CISC microcontroller is its small-sized program. But due to the large size of its instruction set with many addressing modes, it takes a multiple machine cycle to execute & causes longer time to perform. Another problem is the parallel execution of an instruction which is not possible in CISC. RISC (Reduced Instruction Set Computers) This type of microcontroller's CPU is designed to execute smaller simple instructions. Since it takes one machine cycle to execute a single instruction, the number of instructions can be reduced to execute a complex task with higher speed than a CISC. Its instruction set size is small & the program code written for such MCU's are usually very lengthy & consists of many lines. On the Basis of Microcontroller Architecture: The microcontroller architecture is a set of rules that defines the functionality & implementation of a microcontroller. The program memory (ROM) is used for storing & accessing the code or the instruction. While the data memory (RAM) is used for storing & accessing the data used by these instructions. There are two types of architecture that differentiate from each other on the basis of using a single or separate memory for these data & instructions. Harvard Architecture Microcontroller: The Harvard architecture based microcontroller has physically separate memory storage for program code (instructions) & the data, known as the program memory & the data memory respectively. Thus, they have separate bus lines & can be accessed both at the same time. Therefore, the Harvard architecture based microcontroller can complete instruction in one machine cycle. Due to the two physically separate bus lines, the complexity of the microcontroller design & the development cost increases. They are mostly used in modern microcontrollers & digital signal processing units. Related Post: Difference Between RAM and ROM – Comparison Von Neumann (or Princeton) Architecture Microcontroller: Von Neumann or Princeton architecture suggests to use a single memory for both the program & data storage. This concept was proposed by a mathematician Von Neumann in 1945 & It is the most used architecture in all computers, desktop & laptops to date. There is only one bus required for accessing data & fetching instructions. Thus, both operations cannot be done at the same time & they must be scheduled. This is why von Neumann architecture based microcontroller takes two machine cycles to complete an instruction. Since it uses a single bus system, its design is simple & production is very low compared to Harvard architecture. Various Types of Generally used Microcontrollers PIC Microcontroller PIC is an acronym for the peripheral interface controller & it is based on Harvard architecture. It is mainly used for the development of embedded electronics systems in vehicles, robotics, medical devices, vending machines, office machines & various other devices. It is a family of microcontrollers made by Microchip technology. It can be easily programmed through software using Assembly, C or Basic C language to complete a specific task. It has a high-performance RISC (Reduced Instruction set computer) CPU with flash or Program memory & data memory. The program memory EEPROM (electrically erasable programmable read-only memory) is a reprogrammable memory. Since it is Harvard base architecture, it has separate program memory & data memory which increases its processing speed. Most PICs have their internal oscillator of usually 8MHz or 16MHz. They are available in 8-bit, 16-bit & their latest models in a 32-bit wide data bus. The instructions set of PIC microcontrollers vary from 35 instructions in low-end MCUs to 80 instructions in high-end MCU. The program instruction set width can be 12, 14, 16 & 24 bit long. It varies by family & model of PIC microcontroller. They are available in both SMD (surface mount device) & DIP (dual inline package) packaging having a minimum 6 to maximum 144 pins. Due to its low cost, low power consumption & high speed, they are used in almost every electronic device. The most common type of PIC microcontroller is PIC16F877A. Following are some of the most prominent features of PIC16F877A; It is a 40 pin IC in DIP packaging with 33 pins available for input/output. There is no internal oscillator, so an external oscillator is mandatory to generate a clock cycle. Since there is no internal clock, using an external clock up to 20 MHz results in faster operation. It has smaller instructions set of 35. It has 8 bit ADC with 8 channels for interfacing 8 analog inputs at the same time. It has 3 Timers including two 8-bit Timers & one 16-bit Timer. Its sleep mode feature provides ultra-low power consumption to extend battery life. The operating voltage ranges from 4.2v to 5.5v. 8051 Microcontroller The 8051 is the most common & cheap microcontroller made by Intel corporation in 1981. It is based on CISC Harvard architecture for having separate programs & data memory. The most prominent features of 8051 microcontrollers include; It is an 8bit microcontroller available in 40 pin DIP (Dual inline package). It has 4Kb of on-chip programmable ROM for storing program code. It has 128 bytes of on-chip RAM for temporary data storage. However, it can be extended by using external memory up to 64Kb. 32 out of its 40 pins are Input/output pins that are divided into four parallel 8-bit ports. & each of these bits is programmable & accessible. It has 2 16-bit timers/Counter & 2 external interrupts. It may also have various special features such as UARTs, ADC, Op-amp, etc The program for 8051 is written in C language which is converted into assembly language for the microcontroller to understand The 8051 Microcontroller is commonly used in electronics projects mainly for learning purposes. It is used for performing specific tasks such as fire alarms, temperature sensing, motor control, automobiles, light-sensing applications & RTC controlled application. AVR stands for Alf and Vegard's RISC Processor because it was first developed by Alf-Egil Bogen & Vegard Wollan, Norwegian students. It is a family of microcontroller developed by Atmel since 1996 & microchip since 2016. It is based on modified Harvard architecture that offers separate storage space for program & data. AVR has a very high speed when compared with PIC or 8051 microcontroller. The AVR microcontrollers are generally available in the following families; TinyAVR MegaAVR XMEGAAVR Application-specific AVR 32-bit AVR The most common AVR microcontrollers belong to the MegaAVR family known as ATmega16, ATmega32 & ATmega64. The difference between these microcontrollers is their flash storage space of 16KB, 32KB & 64KB identified by the last number in their name. The common feature of AVR ATmega32 microcontrollers are; It is an 8-bit microcontroller with 40 pins DIP (dual in-line package) The total number of input/output pins is 32 which are divided into four 8-bit ports. It has an on-chip reprogrammable flash ROM of 32 KB with RAM of 1KB for data. It contains 3 Timers/counters including two 8-bit timers & one 16-bit timer. A 10-bit ADC with 8 analog channels. It has 4 PWM channels for generating pulses. It has an internal oscillator of 8MHz which can be scaled down. Most popular AVR used due to its fast speed & low cost among the students & hobbyists. Related Post: Difference between Analog and Digital Circuit – Digital vs Analog ARM Microcontroller ARM is an acronym for Advanced RISC Machine & it is a family of microprocessor core available in both Harvard & von Neumann based architecture. The Harvard architecture has separate buses for accessing the program (ROM) & Data (RAM) Memory while von Neumann architecture uses a single memory for both thus compromising the speed. Basically the ARM cores are microprocessors designed to be used in chips such as microcontrollers. The ARM processor is based on RISC architecture which offers fast instruction execution. They are a popular choice for extensive use in consumer handheld electronic devices such as mobile phones, tablets, multimedia players, and other wearable devices. There are different types of ARM processors specifically designed for various purposes; Cortex-A: these processors are built for advanced operating systems and they offer the best performance; Cortex-R: these processors have fast response time thus they are used for real-time applications. Cortex-M: Cortex-M processor are specifically built for microcontrollers, The most common type of ARM microcontroller is ARM Cortex-M. It is a family of ARM processors built for microcontrollers available at different processing speeds. Cortex M0 processor (based on von Neumann architecture) offers low speed at a very low cost mostly used in development boards. While it's improved models such as Cortex-M3 (based on Harvard architecture) is used in the renowned Adruino boards. It is a 32-bit RISC processor It is energy efficient having higher performance. RENESAS Microcontroller RENESAS stands for Renaissance Semiconductor for Advanced Solutions & they offer microprocessor & microcontrollers having best performance features in addition to its ultra-low power consumption & compact packaging. Due to the availability of the largest range of memory capacity & pin count, they are used in various advance automotive control embedded applications. The RL78 & RX family microcontrollers are most popular where the former is used for its low power consumption feature while the latter is famous for its high performance. Some of the most prominent feature of RENESAS RL78 & RX family microcontrollers; They are based on CISC Harvard architecture to offer high performance. The RL78 family is available in an 8-bit & 16bit microcontroller while RX is a 32-bit microcontroller. The RL78 family is a low power microcontroller while RX offers high performance & efficiency. The pin count for RL78 Family ranges from 20 pins to 128 pins while RX microcontrollers are available in 48 pins to 176 pin package. The flash memory is also scaled from 16KB to a maximum of 512KB for RL78 & 32KB to 2MB for RX family microcontrollers. The RAM in the RX family ranges from 2KB to 128KB. Related Post: Difference between Inverter & UPS – Uninterruptible Power Supply Applications of Microcontrollers Microcontrollers can be used to complete any task by interfacing sensors, actuators, motors & appliances, etc. they are used in embedded systems based applications to control a specific task automatically. Following are some of the applications of microcontrollers; Smart Phones & handheld mobile devices Automobiles Cameras Appliances Fire detection, Security Alarms & safety devices (including temperature & smoke sensing) Electronic Measurements Instruments Domestic Appliances such as Microwave oven, Alarm clock, Washing machines, air conditioners. Industrial automation (Conveyor belts, sorting, pick & place bots, etc) Industrial Instrumentation & measurement devices such as volt & current meter, object detection, inspection & sorting devices, Communication devices Motor control applications Test and measurement Medical Devices such as glucose meter, Blood pressure machine, MRI, CT & X-ray machines etc. Related Posts: A microcontroller unit (MCU) is essentially a small computer on a single chip. It is designed to manage specific tasks within an embedded system without requiring a complex operating system. These compact integrated circuits (ICs) contain a processor core (or cores), random-access memory (RAM) and electrically erasable programmable read-only memory (EEPROM) for storing the custom programs that run on the microcontroller, even when the unit is disconnected from a power supply. Unlike general-purpose microprocessors, microcontrollers integrate processing, memory and input/output (I/O) peripherals—including timers, counters and analog-to-digital converters (ADCs)—into one efficient and cost-effective standalone unit. By combining multiple components into a single system, microcontrollers are well-suited for applications requiring real-time signal processing, such as controlling motors and servos and interfacing with various types of sensors and communications. Discover expertly curated insights and news on AI, cloud and more in the weekly Think Newsletter. The following are key components of a microcontroller: Central processing unit (CPU): Colloquially referred to as the "brain" of the computer, the CPU serves as the core component responsible for executing instructions and controlling operations. Memory: Microcontrollers contain both volatile memory (RAM)—which, unlike program memory, stores temporary data that may be lost if the system loses power—and non-volatile flash memory for storing the microcontroller's programming instruction set (firmware). Peripherals: Depending on the intended application, a microcontroller might contain various auxiliary components, such as input/output (I/O) interfaces—including timers, counters, analog-to-digital (ADC) and digital-to-analog (DAC) signal converters (ADCs)—and communication protocols (UART, SPI, I2C). Auxiliaries might also include components like LCD screens, ethernet connectivity ports or interfaces for such types of modules. Microcontrollers are ideal for battery-operated consumer electronics, such as smartphones, smartwatches and other wearables, because they are lightweight, small and require comparatively low power. A favorite among open-source hobbyists, low-cost microcontrollers and microcontroller development boards, such as those made by Arduino and Adafruit, can be easily configured within an integrated development environment (IDE) using common programming languages like C, C++ and Python. Although readily accessible to even beginner developers, microcontrollers are also frequently used to control systems in a wide range of professional applications, including prototyping, robotics, automotive systems, industrial automation and Internet of Things (IoT) applications. Microcontrollers and microprocessors share many similarities. Both can be described as single-chip processors capable of executing computing logic, and both are highly valuable in the development and proliferation of general computing technology. However, the two components differ in both hardware architecture and application. The defining characteristic of a microcontroller is the combination of all necessary computing elements into a single chip—microcontrollers do not require any additional external circuits to operate. Conversely, microprocessors consist of a CPU and several supporting chips providing memory, serial interface, I/O and other necessary features. While the terms microprocessor and CPU are sometimes used interchangeably, it's more accurate to describe microprocessor semiconductors as single integrated circuits that contain a CPU and can be connected to other external auxiliaries, such as input/output devices. The main difference between these two types of microchips is that microcontrollers are self-contained, whereas microprocessors are designed to interface with external auxiliaries. As a result, generalized and demanding tasks that might require specialized hardware with increased processing power are better suited for microprocessors. Specific tasks within embedded systems, such as sensor or motor control, are good examples of suitable microcontroller applications. When comparing microcontrollers and microprocessors, it's helpful to consider four key features: Integrations:Microcontrollers integrate CPU, memory and I/O auxiliaries onto a single chip.Microprocessors require external memory and additional auxiliaries. Applications:Microcontrollers are better suited for specific, low-power or ultra-low power applications within embedded systems, such as home appliances or IoT devices.Microprocessors are more suitable for general-purpose and high-performance applications requiring more processing power, such as personal computers or the data centers' servers. Performance:Microcontrollers are optimized for efficiency and real-time processing and operate at lower clock speeds up to 200MHz.Microprocessors are built for more demanding and complicated computations and can operate at clock speeds above 1 GHz. Operational costs:Microcontroller hardware is inexpensive, and because they do not require specialized programming knowledge, they add little to overall project budgets.Microprocessors are more expensive and more complex. Configuring an advanced microprocessor-based system might require specialized skills. The first types of microcontrollers grew out of advancements made in microprocessor manufacturing as researchers developed techniques to integrate CPU, memory and peripheral components into single chips. Texas Instruments engineers Gary Boone and Michael Cochran are credited with creating the first microcontroller in 1971. Manufacturers like Intel and various Japanese electronics vendors followed quickly. Today, dozens of different microcontroller manufacturers—such as Intel, NXP and Arm—offer hundreds of varieties, ranging from general-purpose options for hobbyists and amateurs to highly specialized solutions for professional technologists and all manner of industries. Here are some of the more common types of microcontrollers: 8-bit microcontrollers The most basic type of microcontroller, featuring limited processing and memory and typically used in small appliances like toys and remote controls. 16-bit microcontrollers Twice as capable as 8-bit models, 16-bit microcontrollers are used for more complex applications, including medical devices, automotive systems and industrial control systems. 32-bit microcontrollers The most powerful and feature-rich type of microcontrollers, these are used for demanding applications, such as gaming consoles, entertainment devices and high-end industrial automation. Reduced instruction set computer (RISC) microcontrollers RISC microcontrollers incorporate a design architecture that simplifies and improves operations by executing fewer compute instructions faster than other methodologies, such as the complex instruction set computer (CISC) architecture. ARM microcontrollers Formerly an acronym for Advanced RISC Machines, these types of microcontrollers incorporate the ARM architecture, including the modern ARM Cortex subset, that bolsters performance and reliability. ARM microcontrollers are widely used in mobile devices, automotive systems and industrial control systems. PIC microcontrollers Developed by Microchip Technology, the PIC microcontroller is the world's smallest microcontroller, found frequently in robotics, home and industrial automation and renewable energy systems. FPGA-based microcontrollers Commonly used in applications necessitating digital signal processing, video processing and high-speed networking, these microcontrollers use field-programmable gate arrays (FPGAs), which can be configured and reconfigured on the hardware level, to achieve highly versatile and customizable processing results. In the early 1970s, two American companies, Intel and Texas Instruments, introduced microprocessors and microcontrollers to the world. These companies envisioned a future dominated by single-chip integrated computers. Today, such devices play critical roles in almost all consumer electronic devices. You can find microcontrollers in nearly every home on the planet. We have become dependent on microcontrollers, yet surprisingly, only a few people really know what a microcontroller is. In this tutorial, we will answer that question. We will also compare features of the most popular microcontrollers on the market today. The diagram below illustrates the microcontroller system as a collection of parts or devices with three features: input, process, and output. A system accepts at least one input, performs some action on that input, and prodces one or more outputs. The inputs and outputs of a microcontroller system are voltages that we can use to determine the state of external devices. The microcontroller reads the voltages from an input device and uses this information to decide on the correct voltage to output. A microcontroller system is embedded in an integrated circuit (IC). A typical microcontroller includes a processor, program memory, RAM, input/output pins, and more on a single chip. Microcontrollers can be used to control a wide variety of electronic devices. They are perfect for applications requiring multiple repetitive operations or high-speed computations. Here's a list of some devices that a microcontroller could be used to control: ComputersComputer peripheralsTelephone systemsHome appliancesIndustrial equipmentSecurity systemsSensors and sensor arrays Experienced DIY electronics builders find it easy to build circuits on breadboards. However, as the complexity of a circuit increases, it becomes difficult too to use breadboards. This is where development boards come in handy. Development boards are PCBs that contain a particular microcontroller IC and all of the supporting external circuitry to make the microcontroller easier to use for quick development and prototyping. The most popular development boards on the market are without doubt the Raspberry Pi and the Arduino Uno. Their introduction to the DIY electronics hobby has revolutionized the way people learn about electronics and build electronic projects. Arduino Uno Development Board Development boards always have input/output pins to make adding sensors, displays, motors, and servos easy. Some microcontrollers are easier to program than others, but every common microcontroller should have lots of resources online that can help you learn how to program it. Some microcontrollers like the Arduino have their own integrated development environment (IDE). In the Arduino IDE, you can write Arduino code and upload it to the microcontroller via a USB cable. Lots of other microcontrollers can be programmed using the Arduino IDE, for example the ESP8266 WiFi module. There are so many microcontrollers on the market today, and selecting the best microcontroller for your project may be a bit challenging for new designers. It's true, there are a lot of factors to consider when selecting a microcontroller. But the most important are functionality, ease of programming, cost, and of course, the availability of support. Popular microcontrollers will have the best support for troubleshooting any issues you might have. Using a more widely accepted microcontroller means that it will be easier to get help quickly when you get stuck. To help you decide which microcontroller is best for your project, the table below compares the features of the most popular microcontrollers available today: MicrocontrollerFeaturesPerformance specificationsProgramming languageCompiler/IDECommon applicationsAtmel ATtiny85Pins: 8GPIO pins: 6512B RAM8KB SRAM512 Byte EEPROMClock speed: 20 MHz internal clockTwo timersAnalog to digital converterOperating voltage: 2.7V to 5.5VOperating temperature: -40°C to +85°CArduino (with limitations),C/C++, or assembly codeAVR GCC,MPLAB XC8, MPLAB X IDE,Microchip Studio,IAR Embedded WorkbenchSafety-critical circuits, industrial control systems, SMPS and power regulation systems, and analog signal analysis.Espressif ESP8266Pins: 30GPIO pins: 16Flash Memory: 4 MBSRAM: 64 KBClock Speed: 80 MHzUART, I2C, and SPI communication Operating voltage: 3.3VSupply voltage: 7V-12VArduinoArduino IDENetworked and internet connected devices, prototype IoT devices, low power circuits, and projects requiring interfaces with WiFi and Bluetooth.Intel 8051Pins: 404KB ROM128 bytes RAMFull-duplex UART communicationOperating voltage: 5VSupply voltage: 2.5V to 5.5VOperating temperature: -0°C to +70°CAssembly,BASIC, C\C++CX51 C Compiler,IAR Embedded Workbench,BASCOM-AVRAutomobiles, medical devices, consumer appliances, communication systems, robotics, industrial control systems, radio and networking equipment, andremote sensing.Texas Instruments MSP430GPIO pins: 51Non-volatile memory: 64 kBRAM: 2 kBAnalog to digital converterReal-time clockUART, I2C, and SPI communicationOperating voltage: 2.5V – 5.5VActive mode: 330 µA at 1 MHz, 3 V.Assembly,C/C++Texas Instruments C/C++compiler,MSP430 – GCC,BASCOM-AVRIndustrial automation, home automation, infrastructure metering networks, portable test equipment, health and medical electronics, and consumer electronicsPJRC Teensy 3.2Pins: 34 GPIO pins: 3432 bit ARM Cortex-M4 72 MHz CPUFlash memory: 256 kBRAM: 64 kB EEPROM: 2 kBPulse width modulationSeven timers USB, UART, SPI, I2C, and I2S communicationReal time clockOperating voltage: 3.3VSupply voltage: 3.6V to 6.0VAdapted Arduino,C/C++,PythonAVR-GCC,BASCOM-AVR,MicroPythonMotor control, small electronics, drones, remote controlled vehicles, robots, and sensor networks.STMicroelectronics STM32GPIO pins: 51-140USB, UART, SPI, I2C, and I2S communication Twelve 16-bit timersTwo 32-bit timersTemperature sensor,Analog to digital converterDigital to analog converterReal time clockOperating voltage: 3.3VSupply voltage: 1.7V to 3.6VC languageEmbedded ARM GCC compiler, Arm Keil MDK,PlatformIO IDE,STM32CubeIDE,Segger Embedded, Studio,SW4STM32Industrial PLC controllers, printers and scanners, security systems, video surveillance systems, HVAC systems, power meters, motor drivers, and PC peripherals.Silicon Labs EFM32GPIO pins: 18-26ARM Cortex-M3 CPU Flash memory: 512 kBRAM: 32 kBUSART, I2C, and SPI communicationAnalog to digital converterOperating voltage: 1.98V to 3.8VOperating temperature: -40°C to +85°CC languageGNU ARM C CompilerLow power devices, IoT projects, and security systems Microcontrollers can handle a wide range of complex circuit designs, but they still may not be sufficient for some complex projects. In this case, you might need to use a single-board computer. Single-board computers pack all of the functionality of a computer in a device the size of a credit card. Here is a comparison of the most popular Raspberry Pi single-board computers: DeviceRAM ProcessorUSBEthernetWiFiBluetoothHDMIOther VideoMicroSDRaspberry Pi A+512MB700 MHz ARM111 Port–––YesDSI, CompositeYesRaspberry Pi B512MB700 MHz ARM114 Ports10/100Mbps––YesDSI, CompositeYesRaspberry Pi 2B1GB900 MHz Quad-Core ARM Cortex-A74 Ports10/100Mbps––YesDSI, CompositeYesRaspberry Pi 3B1GBQuad-Core 64-bit ARM Cortex A534 Ports10/100Mbps802.11n4.1YesDSI, CompositeYesRaspberry Pi 3B+1GB64-bit ARM Cortex A534 Ports300/Mbps/PoE802.11ac4.2YesDSI, CompositeYesRaspberry Pi Zero512MB1 GHz single-core ARM111 Micro USB–––Mini-HDMI–YesRaspberry Pi Zero wireless512MB1 GHz single-core ARM111 Micro USB–802.11n4.1Mini- HDMI–Yes Here's a comparison of the most popular BeagleBone single-board computers: DeviceMemoryProcessorUSBNetworkVideoSupported expansion interfacesMicroSDPocketBeagle512MB DDR3 (800MHz x 16)AM3358, 1GHz ARM Cortex-A8USB 2.0 480Mbps host/client port, USB 2.0 on expansion headeradd-onsSPI displays3x UART, 4x PWM, 2x SPI, 2x I2C, 8x A/D converter, 2x CAN bus (w/o PHY), 2x quadrature encoder, USBYesBeagleBone Black512MB DDR3 (800MHz x 16), 4GB on-board storage using eMMCAM3358, 1GHz ARM Cortex-A8USB 2.0 480Mbps host/client port, USB 2.0 host port10/100 EthernetmicroHDMI, Cape add-ons4x UART, 12x PWM/Timers, 2x SPI, 2x I2C, 7x A/D converter, 2x CAN bus (w/o PHY), LCD, 3x quadrature encoder, SD/MMC, GPMCYesBeagleBone Blue512MB DDR3 (800MHz x 16), 4GB on-board storage using eMMCAM3358, 1GHz ARM Cortex-A8USB 2.0 480Mbps host/client port, USB 2.0 host port2.4GHz WiFi, Bluetooth, BLESPI displays4x UART, 2-cell LiPo, 2x SPI, I2C, 4x A/D converter, CAN bus (w/ PHY), 8x 6V servo motor, 4x DC motor, 4x quadrature encoderYesBeagleBone AI1GB DDR3 (2x 512Mx16, dual-channel), 16GB on-board storage using eMMCAM5729, 2x ARM Cortex-A15USB 3.0 5Gbps host/client port, USB 2.0 host portGigabit Ethernet, 2.4/5GHz WiFi, Bluetooth, BLEmicroHDMI, Cape add-ons4x UART, 12x PWM/Timers, 2x SPI, 2x I2C, 7x A/D converter, CAN bus (w/o PHY), LCD, 3x quadrature encoder, SD/MMCYesBeagleBone Green512MB DDR3 RAM.4GB 8-bit eMMC on-board flash storageAM3358 1GHz ARM Cortex-A8USB 2.0 480Mbps host/client port, USB 2.0 host port Ethernet UART, PWM/Timers, SPI, I2C, A/D converter, CAN bus (w/o PHY), LCD, SD/MMC YesBeagleBone Green Wireless512MB DDR3 RAM.4GB 8-bit eMMC onboard flash storageAM3358 1GHz ARM Cortex-A84x USB 2.0 480Mbps host/client port, USB 2.0 host port Wi-Fi 802.11 b/g/n 2.4 GHz and Bluetooth 4.1 LE 4x UART, 12x PWM/Timers, 2x SPI, 2x I2C, 7x A/D converter, CAN bus (w/o PHY), LCD, 3x quadrature encoder, SD/MMC YesBeagleBoard XM128MB-512MB DDR3 RAM. 8-bit eMMC onboard flash storageAM37x 1GHz ARM Cortex-A8USB 2.0 480Mbps host/client port, USB 2.0 host port EthernetS-Video portUART, PWM/Timers, SPI, I2C, A/D converter, CAN bus (w/o PHY), LCD, SD/MMC Yes So that's a brief overview of the most popular platforms on the market today! Be sure to leave a comment below if you have questions about anything! Microcontrollers are the backbone of countless embedded systems, driving innovations in various industries, from consumer electronics to automotive and industrial applications. If you're new to the world of microcontrollers, this article serves as an introduction to these versatile devices, their key features, and examples of their practical applications.What is a Microcontroller?A microcontroller is a compact, integrated circuit that combines a processor, memory, and peripheral devices on a single chip. They are designed for embedded applications where low power consumption, cost-effectiveness, and space efficiency are crucial. Microcontrollers are used to perform specific tasks in a wide range of systems, such as controlling appliances, monitoring sensors, and communicating with other devices.Anatomy of MicrocontrollersMicrocontrollers typically consist of the following components:1) Central Processing Unit (CPU):The CPU is the heart of the microcontroller, responsible for executing instructions stored in memory. Depending on the microcontroller, the CPU may be an 8-bit, 16-bit, or 32-bit processor, with each type offering different levels of performance and complexity.2) Memory:Microcontrollers include two types of memory: program memory and data memory. Program memory stores the code to be executed, while data memory is used for temporary storage of variables and intermediate results. Microcontrollers often utilize flash memory for program storage and RAM (Random Access Memory) for data storage.3) Input/Output (I/O) Ports:I/O ports enable microcontrollers to interface with external components, such as sensors, switches, and LEDs. They are typically organized into groups, called ports, with each port consisting of multiple pins that can be configured as inputs or outputs.4) Peripherals:Microcontrollers come equipped with a variety of built-in peripheral devices to handle specific tasks, such as analog-to-digital converters (ADCs), timers, and communication interfaces (e.g., UART, SPI, I2C).Understanding Microcontroller ArchitecturesMicrocontroller architectures are categorized by their data bus width, determining the volume of data processed in a single instruction cycle. The most prevalent architectures include:8-bit: These microcontrollers feature an 8-bit data bus, making them ideal for simple applications where low power consumption and cost outweigh processing speed. Examples include remote controls and basic motor controllers.16-bit: With a 16-bit data bus, these microcontrollers deliver higher processing capabilities than 8-bit counterparts while maintaining relatively low power consumption and cost. They suit more complex applications like automotive systems and industrial controllers.32-bit: High-performance 32-bit microcontrollers offer faster processing and cater to more complex applications such as advanced robotics and IoT systems. However, they are generally more expensive and consume more power than 8-bit and 16-bit alternatives.Popular Microcontroller FamiliesThere are numerous microcontroller families available, each with its own unique features and capabilities. Some popular microcontroller families include:1) Atmel AVR:Known for their ease of use and low power consumption, AVR microcontrollers are widely used in hobbyist and commercial applications. The popular Arduino development platform is based on AVR microcontrollers, such as the ATmega328P.AVR® MCUs | Microchip Technology2) Microchip PIC:The PIC family of microcontrollers offers a wide range of devices with various performance levels, peripheral options, and package sizes. PIC microcontrollers are commonly used in industrial and automotive applications.Get Started Now With PIC® MCUs | Microchip Technology3) ARM Cortex:ARM Cortex microcontrollers are based on a standardized processor architecture and are known for their high performance and energy efficiency. They are used in a variety of applications, from simple control systems to complex IoT devices.Essential Components of a MicrocontrollerA microcontroller comprises several essential components that collaborate to execute tasks and process data:CPU (Central Processing Unit): The CPU, the heart of the microcontroller, is responsible for executing instructions and performing arithmetic and logical operations. It fetches, decodes, and executes program instructions stored in memory.Memory: Memory stores program instructions and data. Microcontrollers typically feature two types of memory: non-volatile memory (e.g., flash memory) for firmware storage and volatile memory (e.g., RAM) for temporary data storage during program execution.I/O Ports: Input/Output (I/O) ports enable microcontrollers to interact with external devices such as sensors, actuators, and displays. They can be configured as digital or analog inputs or outputs, depending on application requirements.Timers and Counters: Timers generate precise time delays and measure time intervals, while counters count external events or track the number of clock cycles. Timers and counters are vital for tasks like generating PWM signals, measuring frequencies, and implementing real-time scheduling.Communication Interfaces: Communication interfaces facilitate microcontroller communication with other devices or microcontrollers. Common interfaces include UART (Universal Asynchronous Receiver/Transmitter), SPI (Serial Peripheral Interface), and I2C (Inter-Integrated Circuit).Analog-to-Digital Converters (ADCs) and Digital-to-Analog Converters (DACs): ADCs convert analog signals from sensors into digital values for microcontroller processing. Conversely, DACs convert digital values into analog signals to drive actuators or control external devices.Interrupt Controller: The interrupt controller manages external and internal interrupt sources, allowing the microcontroller to efficiently respond to events without constantly polling for system changes.Getting Started with MicrocontrollersTo begin working with microcontrollers, follow these steps:1) Choose a Microcontroller:There isn't a microcontroller that can be considered to be the best. The best microcontroller is the one that fits your application's needs. Select a microcontroller that suits your application's requirements in terms of processing power, memory, I/O, and peripherals. This is one of the most important steps in designing an embedded application and also difficult. Choosing a microcontroller requires good knowledge and experience. If you are a beginner, choose from one of the most popular ones like AVRs or PICs. The reason is that you can find many tutorials and support communities online.2) Set Up a Development Environment:Acquire the necessary hardware and software tools, such as a programmer, debugger, and integrated development environment (IDE). Many microcontroller manufacturers provide free or low-cost development tools.3) Learn the Programming Language:Microcontrollers are typically programmed in C or C++, although other languages, such as Python and Rust, are gaining popularity. Familiarize yourself with the chosen programming language and the microcontroller's peripheral libraries. It's best to start with C. The simplest and even the most powerful one's all support C. A good place to start learning Embedded C is the following tutorial series on YouTube:Or, if you prefer reading, Embedded C Tutorials – NerdyElectronics4) Build a Prototype:Design a basic circuit, connecting the microcontroller to relevant components, such as sensors, actuators, and power supplies. Write, compile, and upload code to the microcontroller to test the functionality of your prototype.Examples of Microcontroller ApplicationsMicrocontrollers are used in a wide variety of applications, demonstrating their versatility and adaptability. Here are a few examples:1) Home Automation: Microcontrollers can be used to create smart home systems, controlling lighting, temperature, and security features. For instance, a microcontroller could be programmed to automatically adjust the thermostat based on ambient temperature or turn on lights when motion is detected.2) Robotics: Microcontrollers are at the heart of many robotic systems, controlling motors, sensors, and communication interfaces. They can be used to build simple line-following robots or more complex robotic arms capable of performing precise movements.3) Wearables: Wearable devices, such as fitness trackers and smartwatches, often incorporate microcontrollers to monitor sensor data, manage user interfaces, and communicate with smartphones.4) Automotive: Microcontrollers play a crucial role in modern vehicles, managing various systems such as engine control, transmission, and safety features like airbags and anti-lock braking systems.5) Environmental Monitoring: Microcontrollers can be utilized in environmental monitoring applications, such as weather stations, air quality sensors, and soil moisture probes. These systems typically involve collecting data from various sensors and transmitting it to a central server for analysis.ConclusionMicrocontrollers are powerful, versatile devices that form the foundation of countless embedded systems. By understanding their key components, exploring popular microcontroller families, and familiarizing yourself with practical applications, you can embark on a rewarding journey into the world of microcontroller-based projects. As you gain experience, you'll discover the incredible potential these devices offer for creating innovative, efficient, and cost-effective solutions across a wide range of industries.Hi, I'm Vivek, a Senior Embedded Innovation Specialist. I have been working on Embedded Systems and IoT for the past 11 years. I love to share my knowledge and train those who are interested. Nerdyelectronics.com was started out of this interest. You can read my full profile in this link.nerdyelectronics.com/vivek-bhageria/ - Advertisement -A microcontroller is a small, low-cost computer-on-a-chip that is designed to perform a specific set of tasks. It is used in a variety of applications, including controlling machines, sensing and monitoring devices, and automated systems. A microcontroller typically consists of a central processing unit (CPU), memory, input/output (I/O) ports, and support for various peripherals such as timers, counters, and analog-to-digital converters. It is programmed using a specialized computer language and is capable of executing instructions stored in its memory. Some common examples of microcontrollers include the Arduino, Raspberry Pi, and the PIC microcontroller. There are many different types of microcontrollers available in the market, with varying architectures, performance levels, and feature sets. Some common microcontroller architectures include: 8-bit microcontrollers: These microcontrollers have 8-bit CPUs and are typically used in low-end applications that do not require a lot of processing power. They are often less expensive than other types of microcontrollers and are commonly used in consumer products, such as appliances, toys, and other devices. 16-bit microcontrollers: These microcontrollers have 16-bit CPUs and are more powerful than 8-bit microcontrollers. They are commonly used in industrial and automotive applications, as well as in consumer products that require more processing power. 32-bit microcontrollers: These microcontrollers have 32-bit CPUs and are among the most powerful types of microcontrollers. They are often used in high-end applications that require a lot of processing power, such as in medical equipment, military systems, and other demanding applications. Other types of microcontrollers include microprocessors, which are similar to microcontrollers but are typically used in larger, more complex systems; digital signal processors, which are specialized microcontrollers designed for processing digital signals; and field-programmable gate arrays, which are microcontrollers that can be programmed by the user. How Microcontrollers Work? Microcontroller Working Principle source The CPU is the brain of the microcontroller, responsible for executing instructions and controlling the other components of the microcontroller. It retrieves instructions from memory, decodes them, and then performs the operations specified by the instructions. The memory stores the instructions and data that the CPU uses. Microcontrollers typically have two types of memory: read-only memory (ROM) and random access memory (RAM). ROM is used to store the program instructions, while RAM is used to store temporary data that the microcontroller needs to access quickly. I/O ports allow the microcontroller to communicate with the outside world. These can include digital inputs and outputs, as well as analog inputs and outputs, depending on the specific capabilities of the microcontroller. Peripherals are additional components that are built into the microcontroller to perform specialized tasks. These can include timers, counters, and analog-to-digital converters, among others. To use a microcontroller, it must first be programmed using a specialized computer language. Once programmed, the microcontroller can execute the instructions stored in its memory and perform the tasks it was designed to do. Microcontroller Applications Microcontrollers are used in a wide variety of applications, including controlling machines, sensing and monitoring devices, and automated systems. Some common examples of microcontroller applications include: Industrial automation: Microcontrollers are commonly used to control industrial machines, such as robotic arms, conveyor belts, and other manufacturing equipment. Automotive: Microcontrollers are used in automobiles to control a variety of systems, including the engine, transmission, and brakes. Home appliances: Many household appliances, such as washing machines, refrigerators, and microwave ovens, contain microcontrollers to control their various functions. Consumer electronics: Microcontrollers are used in a wide range of consumer electronics, including smartphones, tablets, and other portable devices. Medical devices: Microcontrollers are used in a variety of medical devices, such as heart rate monitors, blood pressure monitors, and insulin pumps. Military systems: Microcontrollers are used in military systems, such as missiles, aircraft, and other defense systems. Environmental monitoring: Microcontrollers are used in sensors and other devices that monitor environmental conditions, such as temperature, humidity, and air quality. Robotics: Microcontrollers are used in robots to control their movements and perform various tasks. Internet of Things (IoT): Microcontrollers are used in IoT devices to collect and transmit data, and to control various functions. Recommended: Microcontroller based Projects Ideas Are Microcontrollers and Microprocessors the Same? Microprocessors and microcontrollers are similar in that they are both integrated circuits that can be used to control electronic devices. However, they have some key differences. In summary, a microprocessor is a general-purpose processor that can be used to perform a wide range of tasks, while a microcontroller is a specialized processor that is designed to control a specific device or perform a specific task. We have a detailed comparison article: Difference between Microcontroller and Microprocessor Microcontroller FAQs Q1. What is the difference between a microcontroller and a microcomputer? A microcomputer is a computer that is built around a microprocessor. It typically includes a processor, memory, and I/O peripherals, and can be programmed to perform a wide range of

tasks. A microcontroller is similar to a microcomputer, but it is typically smaller and more specialized. It is designed to control a specific device or perform a specific task. Q2. How do I get started with microcontroller programming? To get started with microcontroller programming, you will need a microcontroller development board, a computer, and a programming toolchain. You can then write and upload programs to the microcontroller using a programming language of your choice. There are many resources available online that can help you learn more about microcontroller programming, including tutorials, documentation, and forums. Q3: What programming languages can be used to program microcontrollers? Microcontrollers can be programmed in a variety of languages, including C, C++, and assembly language. Q4: How do microcontrollers store and execute programs? A: Microcontrollers typically have onboard memory where programs can be stored and executed from. This can be a read-only memory (ROM), random access memory (RAM), or a combination of both. The type and amount of memory can vary depending on the specific microcontroller. Q5: What are some common input/output (I/O) peripherals found on microcontrollers? Some common I/O peripherals found on microcontrollers include: Digital input/output (I/O) pins: These can be used to read the state of buttons or switches or to control the state of LEDs or other devices. Analog-to-digital converters (ADCs): These can be used to convert analog signals (such as voltage levels) into digital values that can be read by the microcontroller. Timers: These can be used to generate timing signals or measure the duration of events. Serial communication interfaces (SCIs): These can be used to communicate with other devices over a serial link. Examples include UART, I2C, and SPI. Q6: How do I choose the right microcontroller for my project? When selecting a microcontroller for a project, you should consider the following factors: Processor type and speed: Consider the amount of processing power you will need and choose a microcontroller with a processor that meets your needs. Memory: Consider the amount of program and data storage you will need and choose a microcontroller with sufficient memory. I/O peripherals: Consider the types of I/O peripherals you will need and choose a microcontroller with the necessary peripherals. Package type: Consider the physical size and pin count of the microcontroller and choose a package that fits your needs. Cost: Consider your budget and choose a microcontroller that fits within your price range. A microcontroller (µC or uC) is a solitary chip microcomputer fabricated from VLSI fabrication. A micro controller is also known as embedded controller. Today various types of microcontrollers are available in market with different word lengths such as 4bit, 8bit, 64bit and 128bit microcontrollers. Microcontroller is a compressed micro computer manufactured to control the functions of embedded systems in office machines, robots, home appliances, motor vehicles, and a number of other gadgets. A microcontroller is comprises components like – memory, peripherals and most importantly a processor. Microcontrollers are basically employed in devices that need a degree of control to be applied by the user of the device. Microcontroller Basics: Any electric appliance that stores, measures, displays information or calculates comprise of a microcontroller chip inside it. The basic structure of a microcontroller comprise of:- CPU – Microcontrollers brain is named as CPU. CPU is the device which is employed to fetch data, decode it and at the end complete the assigned task successfully. With the help of CPU all the components of microcontroller is connected into a single system. Instruction fetched by the programmable memory is decoded by the CPU. Memory – In a microcontroller memory chip works same as microprocessor. Memory chip stores all programs & data. Microcontrollers are built with certain amount of ROM or RAM (EPROM, EEPROM, etc) or flash memory for the storage of program source codes. Input/output ports – I/O ports are basically employed to interface or drive different appliances such as- printers, LCD's, LED's, etc. Serial Ports – These ports give serial interfaces amid microcontroller & various other peripherals such as parallel port. Timers – A microcontroller may be in-built with one or more timer or counters. The timers & counters control all counting & timing operations within a microcontroller. Timers are employed to count external pulses. The main operations performed by timers' are- pulse generations, clock functions, frequency measuring, modulations, making oscillations, etc. ADC (Analog to digital converter) – ADC is employed to convert analog signals to digital ones. The input signals need to be analog for ADC. The digital signal production can be employed for different digital applications (such as- measurement gadgets). DAC (digital to analog converter) – this converter executes opposite functions that ADC perform. This device is generally employed to supervise analog appliances like- DC motors, etc. Interpret Control- This controller is employed for giving delayed control for a working program. The interpret can be internal or external. Special Functioning Block – Some special microcontrollers manufactured for special appliances like- space systems, robots, etc, comprise of this special function block. This special block has additional ports so as to carry out some special operations. Types of Microcontroller: Microcontrollers are divided into categories according to their memory, architecture, bits and instruction sets. So let's discuss types of microcontrollers:- Bits: 8 bits microcontroller executes logic & arithmetic operations. Examples of 8 bits micro controller is Intel 8031/8051. 16 bits microcontroller executes with greater accuracy and performance in contrast to 8-bit. Example of 16 bit microcontroller is Intel 8096. 32 bits microcontroller is employed mainly in automatically controlled appliances such as office machines, implantable medical appliances, etc. It requires 32-bit instructions to carry out any logical or arithmetic function. Memory: External Memory Microcontroller – When an embedded structure is built with a microcontroller which does not comprise of all the functioning blocks existing on a chip it is named as external memory microcontroller. For illustration- 8031 microcontroller does not have program memory on the chip. Embedded Memory Microcontroller – When an embedded structure is built with a microcontroller which comprise of all the functioning blocks existing on a chip it is named as embedded memory microcontroller. For illustration- 8051 microcontroller has all program & data memory, counters & timers, interrupts, I/O ports and therefore its embedded memory microcontroller. Instruction Set: CISC- CISC means complex instruction set computer, it allows the user to apply 1 instruction as an alternative to many simple instructions. RISC- RISC means Reduced Instruction Set Computers. RISC reduces the operation time by shortening the clock cycle per instruction. Memory Architecture: Harvard Memory Architecture Microcontroller Princeton Memory Architecture Microcontroller 8051 Microcontroller: The most universally employed set of microcontrollers come from the 8051 family. 8051 Microcontrollers persist to be an ideal choice for a huge group of hobbyists and experts. In the course of 8051, the humankind became eyewitness to the most ground-breaking set of microcontrollers. The original 8051 microcontroller was initially invented by Intel. The two other members of this 8051 family are- 8052 – This microcontroller has 3 timers & 256 bytes of RAM. Additionally it has all the features of the traditional 8051 microcontroller. 8051 microcontroller is a subset of 8052 microcontroller. 8031 – This microcontroller is ROM less, other than that it has all the features of a traditional 8051 microcontroller. For execution an external ROM of size 64K bytes can be added to its chip. 8051 microcontroller brings into play 2 different sorts of memory such as- NV-RAM, UV-EPROM and Flash. 8051 Microcontroller Architecture: 8051 microcontroller is an eight bit microcontroller launched in the year 1981 by Intel Corporation. It is available in 40 pin DIP (dual inline package). It has 4kb of ROM (on-chip programmable space) and 128 bytes of RAM space which is inbuilt, if desired 64KB of external memory can be interfaced with the microcontroller. There are four parallel 8 bits ports which are easily programmable as well as addressable. An on-chip crystal oscillator is integrated in the microcontroller which has crystal frequency of 12MHz. In the microcontroller there is a serial input/output port which has 2 pins. Two timers of 16 bits are also incorporated in it; these timers can be employed as timer for internal functioning as well as counter for external functioning. The microcontroller comprise of 5 interrupt sources namely- Serial Port Interrupt, Timer Interrupt 1, External Interrupt 0, Timer Interrupt 0, External Interrupt 1. The programming mode of this micro-controller includes GPRs (general purpose registers), SFRs (special function registers) and SPRs (special purpose registers). PIC Microcontroller: Peripheral Interface Controller (PIC) provided by Micro-chip Technology to categorize its solitary chip microcontrollers. These appliances have been extremely successful in 8 bit micro-controllers. The foremost cause behind it is that Micro-chip Technology has been constantly upgrading the appliance architecture and included much required peripherals to the micro-controller to go well with clientele necessities. PIC microcontrollers are very popular amid hobbyists and industrialists; this is only cause of wide availability, low cost, large user base & serial programming capability. PIC Microcontroller Architecture: The architecture of the 8 bit PIC microcontrollers can be categorized as below – Base Line Architecture – In the base-line architecture PIC microcontrollers of PIC10F family is included, other than that a fraction of PIC12 & PIC16 families are also included. These gadgets make use of 12 bit program word architecture with six to twenty-eight pin package alternatives. Briefly defined attribute set of baseline architecture allows the most lucrative product solutions. This architecture is perfect for battery enabled gadgets. The PIC10F200 series is another reasonably priced 8 bit flash micro-controller with a 6 pin package. Mid Range Architecture – In this midline member of PIC12 & PIC16 families are added that attribute 14 bit program word architecture. The midrange PIC16 gadgets proffer a broad variety of package alternatives (from 8 to 64 package), with low to high levels of peripheral incorporation. This PIC16 appliance attributes a variety of analog, digital & serial peripherals, like- SPI, USART, I2C, USB, LCD & A/D converters. The mid-range PIC16 micro-controllers have suspended controlling ability with an eight level hardware load. High Performance Architecture – The high performance architecture included the PIC18 family of appliances. These micro-controllers make use of 16 bit program word architecture along with 18 to 100 pin package alternatives. The PIC18 appliances are high performance micro-controllers with incorporated Analog to Digital converters. All PIC18 micro-controllers integrate a highly developed RISC architecture that supports flash appliances. The PIC18 has improved foundation attributes, 32 level deep load and several inner and exterior interrupts. Note: List of Top PIC Microcontroller Projects for Engineering Students AVR Microcontroller: AVR also known as Advanced Virtual RISC, is a customized Harvard architecture 8 bit RISC solitary chip micro-controller. It was invented in the year 1966 by Atmel. Harvard architecture signifies that program & data are amassed in different spaces and are used simultaneously. It was one of the foremost micro-controller families to employ on-chip flash memory basically for storing program, as contrasting to one time programmable EPROM, EEPROM or ROM, utilized by other micro-controllers at the same time. Flash memory is a non-volatile (constant on power down) programmable memory. AVR Microcontroller Architecture: AVR microcontrollers' architecture was developed by Alf-Egil Bogen and Vegard Wollan. The name AVR is derived from the names of the architecture developers of the microcontroller. The AT90S8515 was the foremost micro-controller which was AVR architecture based; on the other hand the foremost micro-controller to strike the commercial marketplace was AT90S1200 which was launched in the year 1997. The SRAM, Flash and EEPROM all are incorporated on a single chip, thereby eliminating the requirement of any other external memory in maximum devices. Several appliances comprise of parallel external bus alternative, so as to add extra data memory gadgets. Approximately all appliances, except TinyAVR chips comprise serial interface, which is used to link large serial Flash & EEPROMs chips. ARM Microcontroller: ARM is the name of a company that designs micro-processors architecture. It is also engaged in licensing them to the producers who fabricate genuine chips. In actuality ARM is a 32 bit genuine RISC architecture. It was initially developed in the year 1980 by Acorn Computers Ltd. This ARM base microprocessor does not have on-board flash memory. ARM is particularly designed for micro-controller devices, it is simple to be trained and make use of, however powerful enough for the most challenging embedded devices. ARM Microcontroller Architecture: The ARM architecture is a 32 bit RISC processor developed by ARM Ltd. Owing to its power-saving attributes, ARM central processing units are prevailing in the mobile electronics marketplace, where less power expenditure is a vital design aim. ARM architecture comprise of the underneath RISC elements:- Maximum single cycle functioning Constant 16×32 bit register file. Load or store architecture. Preset instruction width of 32 bits so as to simplify pipe-lining and decoding, at minimized code density. For misaligned memory access there is no support. Microcontroller Applications: Microcontrollers are intended for embedded devices, in comparison to the micro-processors which are used in PCs or other all-purpose devices. Microcontrollers are employed in automatically managed inventions and appliances like- power tools, implantable medical devices, automobile engine control systems, , office machines, remote controls appliances, toys and many more embedded systems. By dipping the size and expenditure in comparison to a design that make use of a different micro-processor, I/O devices and memory, micro-controllers formulate it inexpensive to digitally control more & more appliances and operations. Mixed signal micro-controllers are general; putting together analog constituents required controlling non-digital electronic structures. Application of Microcontroller in Day to Day Life Devices: Light sensing & controlling devices Temperature sensing and controlling devices Fire detection & safety devices Industrial instrumentation devices Process control devices Application of Microcontroller in Industrial Control Devices: Industrial instrumentation devices Process control devices Application of Microcontroller in Metering & Measurement Devices: Volt Meter Measuring revolving objects Current meter Hand-held metering systems Related Link: Difference Between Microcontroller and Microprocessor A microcontroller is a compact integrated circuit designed to govern a specific operation in an embedded system. A typical microcontroller includes a processor, memory and input/output (I/O) peripherals on a single chip. Sometimes referred to as an embedded controller or microcontroller unit (MCU), microcontrollers are found in automobile engine control systems, robots, office machines, medical devices, mobile radio transceivers, vending machines and home appliances, among other devices. They're simple miniature PCs designed to control small features of a larger component without a complex front-end operating system. A microcontroller is embedded inside of a system to control a single function in a device. It uses its central processor to interpret data it receives from its I/O peripherals. The information that the microcontroller receives is temporarily stored in its data memory, where the processor accesses it and uses instructions stored in its program memory to decipher and apply the incoming data. It then uses its I/O peripherals to communicate and take the appropriate action. Microcontrollers are used in an array of systems and devices. Devices often use multiple microcontrollers that work together in the device to handle their respective tasks. For example, a car has many microcontrollers that control various individual systems, such as the antilock braking system, traction control, fuel injection and suspension control. Each microcontroller communicates with the others to inform them of the correct actions. Some might communicate with a more complex central computer within the car, and others might only communicate with other microcontrollers. They send and receive data using their I/O peripherals and process that data to perform their designated tasks. The core elements that make up a microcontroller are the central processing unit (CPU), memory and I/O peripherals. CPU Also known as a processor, a CPU is the brain of the device. It processes and responds to various instructions that direct the microcontroller's function. This involves performing basic arithmetic, logic and I/O operations. It also performs data transfer operations, which communicate commands to other components in the larger embedded system. Memory A microcontroller's memory stores the data that the processor receives and uses to respond to instructions it's programmed to carry out. A microcontroller has two main memory types: Program memory. This stores long-term information about the instructions that the CPU carries out. Program memory is non-volatile memory, meaning it stores information over time without needing a power supply. Data memory. This temporary data storage is used while the instructions are being executed. Data memory is volatile, meaning the data it holds is temporary and is only maintained if the device is connected to a power source. I/O peripherals The I/O devices are the interface for the processor to the outside world. The input ports receive information and send it to the processor in the form of binary data. The processor receives that data and sends the necessary instructions to output devices, which execute tasks external to the microcontroller. Other elements While the processor, memory and I/O peripherals are the defining elements of the microprocessor, there are other elements that are frequently included. The term I/O peripheral refers to a supporting component that interfaces with the memory and processor. There are many supporting components that can be classified as peripherals. Having some manifestation of an I/O peripheral is elemental to a microprocessor because it is the mechanism through which the processor functions. Other supporting elements of a microcontroller include the following: Analog-to-digital converter. An ADC is a circuit that converts analog signals to digital signals. It lets the processor at the center of the microcontroller interface with external analog devices, such as sensors. Digital-to-analog converter. A DAC performs the inverse function of an ADC, letting the microcontroller's processor communicate its outgoing signals to external analog components. System bus. The system bus is the connective wire that links together all components of the microcontroller. Serial port. The serial port is one example of an I/O port that enables the microcontroller to connect to external components. It has a similar function to a USB or a parallel port but differs in the way it exchanges bits. Microcontrollers include various components. Microcontroller processors vary based on the application. Options range from the simple 4-bit, 8-bit or 16-bit processors to more complex 32-bit or 64-bit processors. Microcontrollers can use volatile memory, such as RAM, and non-volatile memory types, including flash memory, erasable programmable read-only memory and electrically erasable programmable ROM. Generally, microcontrollers are usable without additional computing components. They're designed with sufficient onboard memory, as well as offering pins for general I/O operations, so they can directly interface with sensors and other components. Microcontroller architecture is based on the Harvard architecture or Von Neumann architecture. They offer different methods of exchanging data between the processor and memory. With Harvard architecture, the data bus and instruction are separate, enabling simultaneous transfers. With a Von Neumann architecture, one bus is used for both data and instructions. Microcontroller processors are based on complex instruction set computer (CISC) or reduced instruction set computer (RISC). CISC generally has around 80 instructions, while RISC has about 30. CISC also has more addressing modes, 12 to 24 compared to RISC's three to five. CISC is easier to implement and uses memory more efficiently, but it can have performance degradation because of the higher number of clock cycles needed to execute instructions. RISC places more emphasis on software and provides better performance than CISC processors, which emphasize hardware. CISC has a simplified instruction set and, therefore, increased design simplicity. However, because of the emphasis RISC places on software, the software can be more complex. Which one is used depends on the application. When they first became available, microcontrollers only used assembly language. Today, the C programming language is a popular option. Python and JavaScript are also common microprocessor languages. MCUs feature I/O pins to implement peripheral functions, such as ADCs, liquid-crystal display controllers, real-time clocks, universal synchronous/asynchronous receiver-transmitters, timers, universal asynchronous receiver-transmitters and USB connectivity. Internet of things (IoT) sensors that gather data such as humidity and temperature are also often attached to microcontrollers. Microcontrollers can be classified according to data size and architecture. Common types include the following: 8-bit microcontroller. These MCUs can only transmit 8 bits of data at a given time. However, they consume less power compared to larger data sizes. 16-bit microcontroller. These microcontrollers have higher clock speeds and more memory than 8-bit microcontrollers. They are two times faster than 8-bit microcontrollers. 32-bit microcontroller. These high-speed microcontrollers are faster and have more processing capacity than 16-bit ones. However, their power consumption is significantly higher. Microcontrollers use one of two architectures: Von Neumann architecture microcontrollers perform one operation at a time because there is only one internal bus to handle both memory and data. Harvard architecture microcontrollers provide high performance compared with Von Neumann ones. This is because they have separate buses for processing instructions and moving data. Examples of microcontroller models include the following: MCS-51. Intel developed this single-chip microcontroller type in 1980. It is also referred to as an 8051 microcontroller. It used CISC and the Harvard architecture and came in 8-, 16- and 32-bit data sizes. Intel stopped making MCS-51 in the early 2000s, though other chipmakers offer enhanced versions of it. AVR. Atmel developed this 8-bit single-chip RISC microcontroller in 1996, using a modified Harvard architecture. It became a family of microcontrollers that was one of the first to use on-chip flash computer memory to provide program storage. Microchip Technology acquired Atmel in 2016 and continues to make AVR microcontrollers. Programmable Intelligent Computer. General Instrument developed the PIC microcontroller in 1976 under the name Programmable Interface Controller. This family of microcontrollers can be programmed to carry out different tasks, such as controlling electrical processes in homes, vehicles and medical facilities. Advanced RISC Machines. Arm microcontrollers are also known as Arm Cortex-M microcontrollers. These lightweight microcontrollers are used in mobile electronic devices, as well as in manufacturing settings. They are designed to be energy-efficient and suitable for a range of embedded systems. These microcontrollers are part of the Arm family of processors that Acorn Computers developed in the early 1980s. Microcontrollers are used in multiple industries and applications, including in home and enterprise, building automation, manufacturing, robotics, automotive, lighting, smart energy, industrial automation, communications and IoT applications in business settings. Basic areas where microcontrollers are used include the following: Digital signal processors (DSPs). One application of a microcontroller is its use as a DSP. Frequently, incoming analog signals come with a certain level of noise. Noise in this context means ambiguous values that can't be readily translated into standard digital values. A microcontroller can use its ADC and DAC to convert the incoming noisy analog signal into an even outgoing digital signal. Home appliances. The simplest microcontrollers facilitate the operation of electromechanical systems found in everyday convenience items, such as ovens, refrigerators, toasters, mobile devices, key fobs, video game systems, televisions and lawn-watering systems. Office machines. Microcontrollers are also common in office machines, such as photocopiers, scanners, fax machines and printers, as well as smart meters, ATMs and security systems. More sophisticated applications. Microcontrollers perform critical functions in aircraft, spacecraft, oceangoing vessels and robots. Medical applications. In medical scenarios, microcontrollers can regulate the operations of an artificial heart, kidney or other organs. They can also be instrumental in the functioning of prosthetic devices. The main difference between microcontrollers and microprocessors is in the level of functionality. Microcontrollers function on their own with a direct connection to sensors and actuators. Microprocessors are designed to maximize compute power on the chip with internal bus connections rather than direct I/O to supporting hardware, such as RAM and serial ports. Simply put, coffee makers use microcontrollers; desktop computers use microprocessors. The distinction between microcontrollers and microprocessors has become less clear as denser and more complex chips have become relatively cheap to manufacture. This trend has let microcontrollers take on more general-purpose computer functionalities. Microcontrollers are less expensive and use less power than microprocessors. Microprocessors don't have built-in RAM, ROM or other peripherals on the chip, but rather attach to these with their pins. A microprocessor is considered the heart of a computer system, whereas a microcontroller is the heart of an embedded system. Microcontrollers and microprocessors are two forms of chip technology that have key differences. There are a few technology and business considerations to keep in mind when choosing a microcontroller for a project. Many businesses focus on using low-cost microcontrollers. However, it's also important to consider speed, the amount of RAM or ROM provided, the number and types of I/O pins on an MCU, power consumption and development support. Be sure to ask questions such as the following: What hardware peripherals are required? Are external communications needed? Is the device in question battery-operated, requiring a low-power MCU? Are additional storage devices needed, such as a flash memory card? What architecture should be used? What sort of community and resources are available for the microcontroller? What is the market availability of the microcontroller, and what are potential alternatives? Embedded system technology is crucial to the success of IoT. Learn more about how IoT and embedded systemst work together.