1. Microcontrollers are designed to perform a task within a specific time: This feature is enhanced by a timing mechanism.

2. Easy to design and relatively low cost: The cost of buying ICs, resistors, transistors and other electronic components that make up the entire functionality of a Microcontroller is expensive when procured separately. The assembling of these components isn't easy aswell. When compared to a microcontroller.

3. Performs a specific task: A specific program is written for the use of a Microcontroller. The actuator is designed to respect the instructions and intelligence of the processor, hence can't perform other function other than the specific task. The interrupt can only "distract" the system temporarily from this specific task, but cant change it.

Microcontrollers are dependable:

The entire circuitry of a Microcontroller is compared to a dedicated computer which serves a particular purpose. It has a dedicated processor, memory, power etc that serves the said purpose, with a written instructions (program) on the chip for implementation.

A Microcontroller is highly dependable if it is suitable for a particular task.

Due to wider data bus, a 32 bit mc takes fewer instruction cycles to handle a task. But when used for wireless IoT sensors or other battery-operated circuits for instance, the dependability could be questionable. This is because it would drain the battery so fast.

Generally, we can describe the dependability by:

**Security:** Its ability to prevent another user from accessing the file without a prior access.

**Reliability:** For a system/microcontroller to be dependable, it must perform its function precisely with minimal downtime. If its operation is interrupted while in use, the dependability is affected.

**Safety:** A damage or injury to the people, environment and processes is highly prevented.

**Microcontroller Processors.**

There are different processors used in a microcontroller; the most popular are summerised below

1. PIC (Programmable Interface Controller): Is a great processor with amazing features that is capable of handling wonderful projects. It contains processor, memory, input/output. The memory uses a flash ROM to store program, with 368 bytes of RAM and 33 Input/Output pins. It has many families with other amazing features. For instance; 12F675 (8pin) has:

• 64 bytes RAM

• 128 bytes EEPROM memory

• External oscillations of upto 20MHz

• Two timers

• ADC

Generally, PIC can be re-programmed using Assembler. This can be done with the chip inserted on a compiler. You can equally program while it's in the circuit, using the ICSP interface.

It also has a Synchronous Serial Port(SSP) that allows you to communicate with other devices that uses 12C and SPI protocols.

ATmel Microcontrollers.

It's a single chip developed by ATmel. It is based on advanced RISC architecture. Atmel is amazingly fast with lots of features;

32K bytes flash Memory

1024 bytes EEPROM

32 programmable I/O lines

An Internal SRAM

Programmable serial USART

10 bit ADC

At 1MHz it can execute one million instructions per second.

ATmega 8 and ATmega 32 are ATmel/AVR Microcontrollers.

Arduino

C. Overview of a Microcontroller

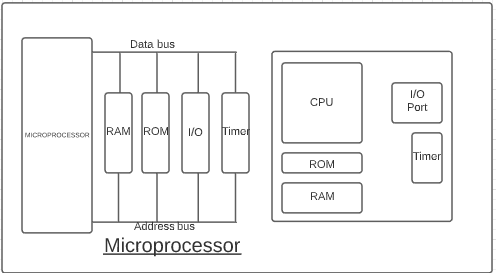
Microcontroller can be compared to a computer. The elements of a Microcontroller is summarized below;

CPU

This is the general controller of a Microcontroller. It could be likened to the brain because it controls the operation of other elements of the component. It primarily fetch and decode piece of instructions needed to execute an operation.

Memory

It stores data in a Microcontroller. The information could be program code and/or data used by chip during normal operation.

MICROCONTROLLER