Chapter 1

Background

Background

- Microcontroller (MCU) In Depth
- Rust vs C

Microcontroller In Depth

Microcontroller In Depth

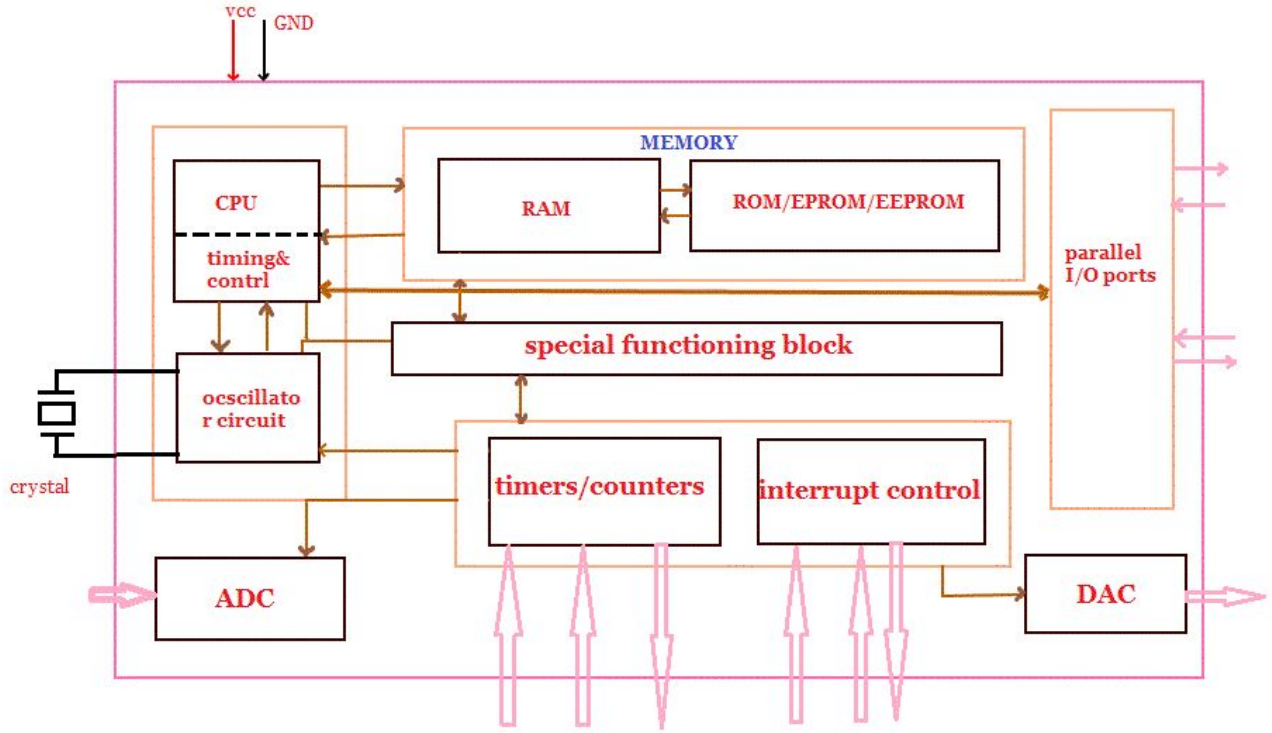
- What is Microcontroller?
- Components of MCU
- Applications
- Why Microcontrollers?
- What are the pros and cons?

What Is Microcontroller?

- **MCU** in simplest means are Integrated Circuit (**IC**)
- **MCU** is a system on chip (**SOC**); a chip consist of several components. We'll discuss these components
- **MCUs** are designed for **embedded** applications

Microcontroller Components[1]

- **Central processing unit(CPU)**
- **Random Access Memory)(RAM) - Volatile Memory**
- **Read Only Memory(ROM)/ Flash Memory - Non-Volatile Memory**
- **Input/output ports**
- Timers and Counters
- Interrupt Controls
- Analog to digital converters
- Digital analog converters
- Serial interfacing ports
- Oscillatory circuits



Microcontroller - Structure Block Diagram

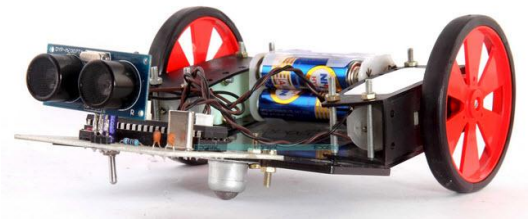
Applications

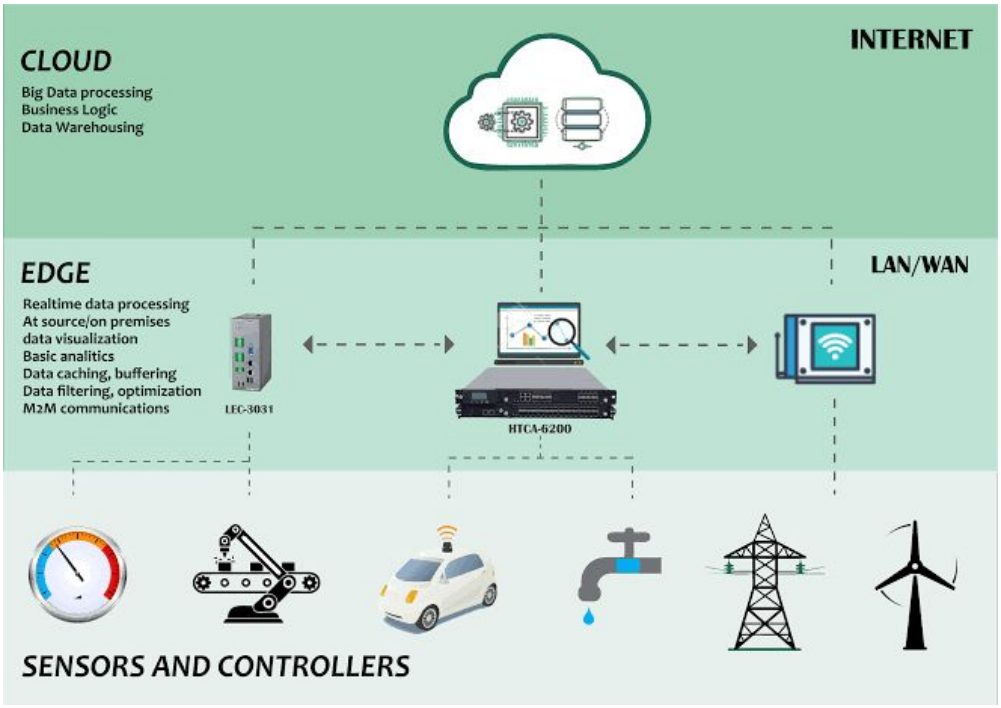
You can find microcontrollers all around, every device that calculates, stores, controls or displays information must have microcontrollers. From cell phones to smart watches to ACs to washing machines to vehicle all are the applications of microcontrollers.

Applications (cont.)

Further the following are areas where MCUs are extensively being used:

- Used in biomedical instruments
- Communication Systems
- Robotics
- Automobile Industry
- IoT Application





IoT Architecture - Edge Computing

Why Microcontroller?

All the applications mentioned in previous slides, can be implemented using just Raspberry Pi. Then why we should use microcontroller that operates een without OS?

There are 3 reasons that pushes us to use MCUs:

- Low cost
- Low power consumption
- Real time constraints

Pros

- Act as a microcomputer without any digital part.
- Usage is simple, easy to troubleshoot and system maintaining.
- Immediately respond to any event occur.
- Most of the pins are programmable by user.

Cons

- Got complex architecture than microprocessors.
- Perform limited number of executions simultaneously.
- Cannot perform heavy computations due to limited resources.


PRESIDENTIAL
INITIATIVE
ARTIFICIAL INTELLIGENCE AND COMPUTING

Rust vs C

C

- Difficult to learn for beginners at least.
- C lacks official package manager.
- C ecosystem is mature. Solution to many problems already exist.

Rust

- Easy to learn because of easy syntax (compare to C).
- Rust has Cargo.
- Less support and existing solutions available for Rust.