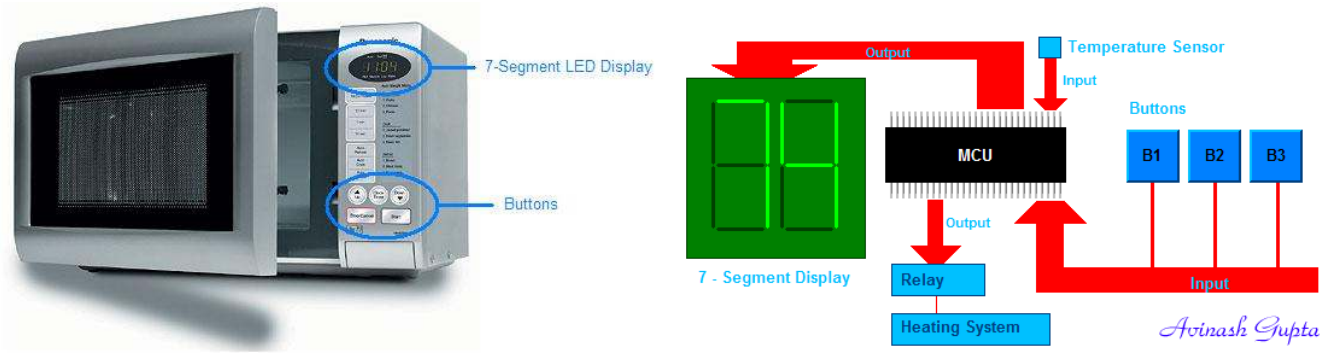
# Introduction

AVR Tutorial Series

## Just what is a microcontroller?

To get you understand quickly I define a microcontroller as a single chip computer.Yes it is a full blown computer in its own. It has a C.P.U., RAM,some amount of EEPROM(for secondary storage i.e. permanent storage without power), many on-chip peripherals(Timer,Serial communication, Analogue to Digital converters etc.).If you don't understand, no problem I will be dealing them in detail in next tutorials. But compared to a P.C. their resources(RAM,speed etc)are less.But that is what is required !!! Because P.C. is a general purpose computer, which means it can run thousands of different softwares that are available for specific needs.Like it can run a game.The same P.C. can run this browser in which you are reading this!It can run a custom solution for banks,railways and airways.It can run a 3D modeling,video editing & image editing software for a production company. Many of these are huge software,requiring lots of memory and CPU power.And a P.C. can run simultaneously many of this !!! So to run them the host computer should have enough RAM and CPU power so that it can run heaviest of them.

But in case of a microcontroller(aka MCU) which is used for a specific purpose like switching a Microwave oven heating off after a preset time,and also when tempereture exceed preset value.




**A Typical MCU based solution.**

This design also involve controlling few indicator L.E.D. , 2 or 3 seven segment display, few switches and a relay controlling the oven.This doesn't need a monster with 3.2 GHz Intel Core 2 Quad extreme processor with 2 GB Dual channed DDR3 RAM, 320 GB HDD and Dedicated nVIDIA GEFORCE graphics !!!. What is of main concern is cost and size.What it will be running through out its life is a simple program that is hardly more than 4KB in size, requiring less than 128 Bytes of RAM to store few varriables and optionaly a few bytes to store permantly the temperature and time set on last use so that it can read those on statup.

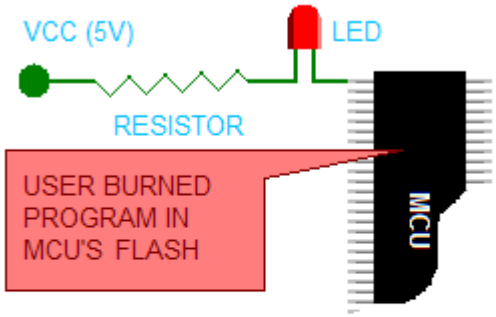
Other Example where MCUs are used are

- Pendrives(for controlling the communication between P.C. and Flash Chip and also the small LED!)
- Hardisks( again for the same purpose)
- Mouse(Reads and Interprets the Sensors and send final result to P.C.),Keyboards
- Printers : Ever opened a printer for installing ink cartridge ? Then you must have seen the printed head. There are motors to control the print head and the paper movement. Your P.C. is not directly connected to them but the in-built MCU of printer control all these. Your P.C. just sends the data(pixels) through the communication line(USB or parallel).But the MCU used here is fairly fast and has lots of RAM.
- Automobiles
- Calculators,Electronic wending machines, Electronic weighing scales, Phones(digital with LCD and phonebook)
- Cell phones
- Any thing that is small, but has great functions and is cheap !!!

They are every where !!!

## A Simple MCU based system.

A simplest MCU system may look like below




A Simplest MCU based system.

The program it is executing is like this( In C language).The MCU contains a flash memory where it stores its program(its like a hard disk to it). The flash memory can be easily erased and a new program can burned. This makes them very flexible. MCUs can be programmed few thousand times before they die.

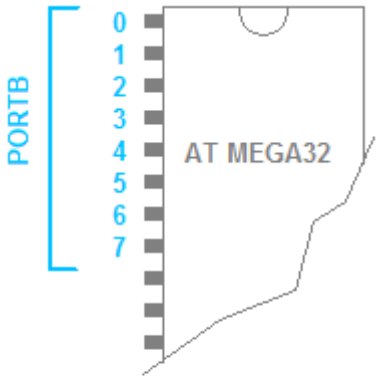


## A Simple MCU Based System.

**Sample Program**

```
void main()
{
    SetPortDirection();
    while(1)
    {
     PORTA=0b00000001;
     Delay(0.5);
     PORTA=0b00000000;
     Delay(0.5);
    }
}
```

A MCU has some ports.Ports are PINs on the MCU that can be turned on and off by program.By on I mean 5V and off means 0V or GND.This behaviour is for OUTPUT mode. They can also be put in INPUT mode. In INPUT mode they can read what is the signal level on them (only on and off).If voltage is more than a threshold voltage(usually half the supply) it is reported as ON(1) otherwise OFF(0).This is how MCU control everything.Majority of the PINs of a MCU are PORT so you can hookup lots of gizmos to it !!!They are named PORTA,PORTB,PORTC,PORTD etc.They are of one byte which means 8 Bits.Generaly all bits of them are connected to external pins and are available outside the chip.In smaller chips only some of the eigth bits are available.




Physical Location of PORTB bits



For example the image shows the PORTB and it bit numbering.Setting PORTB=0b00000001 will set PORTB's zeroth bit high that is 5V while remaining PINs will be low(GND).

[NOTE:To write a binary number in c prefix it with 0b ex 0b00001000. It is decima 8 not 1000 !!! ]

The program above is not complete program it is just a pseudo-code.So it won't compile.Also you may be wondering where it will be entered, compiled and how the heck it will pushed into that chip?They are the subject of next tutorails and I will be covering them in detail.

**MCU Ports**

What the program does
STEP 1 SetPortDirection();

This Function Makes the PORTB as OUTPUT.Its implementation detail is not shown.
STEP 2 PORTB=0b00000001;

makes the 0th bit high, switching off the L.E.D. because other end of LED is connected to VCC(i.e. Supply voltage 5V)
STEP 3 Delay(0.5);

Halts the MCU for 0.5 Sec
STEP 4 PORTB=0b00000000;

Switches on the LED

STEP 5 Delay(0.5);

STEP 2 to 5 are within an infinite while loop so it runs till MCU is powered.

The program is compiled, and burned into the chip and power is turned on and woillaaaa that LED is blinking once every second.You have just understood the "HELLO WORLD" program for MCU empire.Although the given program doesn't do something very important and can be done without a MCU and in a cheap manner, but it introduce you to microcontroller programming.Doing this with MCU has some advantages also.You can change the way LED blinks by simply writting a new program without touching the harware part.You can also connect 8 LEDs to all 8 PINs of the PORTB, write a nice program them to light them in varrious pattern and you have a delux decorating lights !!! Which you can't make easily withou MCUs.So you have seen that major functioning of a MCU project are made in software and hardware part is simple and small.

So you have learned the basics of MCUs and their use.Hope you have nice time reading.So, good bye for now, we will meet in next tutorial.

***Whats next***

Next we will be selecting a microcontroller and getting familier with the over all development process.