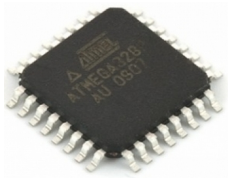
ATMEGA328/Arduino Uno – C-Language_00
https://github.com/teaksoon/lmaewapm



First, we need the ATmega328P Datasheet from the Supplier of the ATMEGA328P Micro-controller (since most of our Arduino Uno has this chip)

I have a copy in my repository,

https://github.com/teaksoon/lmaewapm/blob/main/ATmega328P_Datasheet.pdf

You should download and keep one copy on your Computer all the time if you are working with Arduino Uno or the ATMEGA328P micro-controller directly.

The Datasheet file is very big, there is no need to print it or read everything in it. We may only need to use some of it. For now, take a look at these two topic first (dont worry if they dont make sense to you at all now)

1. from ATMEGA328 Datasheet: Topic 31. from Page 281 “31. Instruction Set Summary” – for **“AVR INSTRUCTION SET”**

2. from ATMEGA328 Datasheet: Topic 30. from Page 271 “30. Register Summary” – for **ATMEGA328 CPU Memory(Register)**

Those two Topic are the two main things that we need, to get the micro-controller CPU do work for us.

There are many Register in the CPU, some **values in the Registers will cause a task being performed.** Sometimes CPU also updates the Registers with certain values for our Program to use (The Datasheet will tell us every single one of them). In order to **update or retrieve the Registers**, we need to use the instructions from the **INSTRUCTION SET**

For example:

```
SBI  DDRB, 5
```

The code above has an instruction **“SBI”** from **AVR INSTRUCTION SET**, will set the **CPU Memory(Register) “DDRB”**, 5th bit, to have a value of ‘1’

Once the CPU see this instruction “SBI DDRB 5”, will set our Arduino Uno I/O Pin 13 to become an OUTPUT Pin.

If we program in **Assembly Language**, we will have to code exclusively with the **INSTRUCTION SET and** the **CPU Memory(Register)**

However, when we Program using **C-Language with the Arduino Libraries**, we **seldom see the INSTRUCTION SET or the Registers.** This is because they are all hidden from us by our C-Language and the Arduino Libraries (which makes things alot easier). The **C-Language and Arduino Libraries has it own set of instructions** which will be **converted into AVR INSTRUCTION SET/Registers** internally **by C-Language Compiler.**

Although most of the time we might not need to use the INSTRUCTION SET and Registers in our C-Language when using the Arduino Libaries, we need to know they exist. As we progress, sooner or later we will be digging into them. **Good thing about C-Language is that, we can access into the INSTRUCTION SET and CPU Memory(Register) if we wish to do so.**