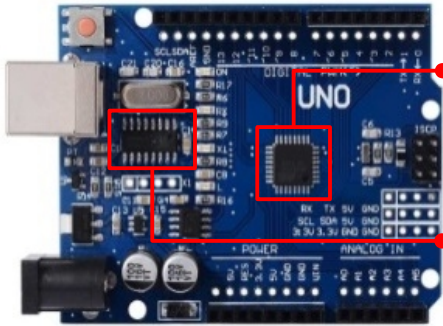


What is "Arduino" ?

"Arduino" is a collection of tools, to help us to operate and work on the **ATMEGA328P Micro-controller**

Arduino Uno Development Board



The **ATMEGA328 micro-controller chip** (QFP Package) on the **Arduino Uno Development Board**

USB-TTL Chip (CH340)

What is a Micro-Controller ?

A Micro-controller works similar to our regular Computer, it has Central Processing Unit (CPU), Working Memory and Permanent Storage Memory. In order for the micro-controller CPU to do task, we also need to give the micro-controller a series of instructions (**PROGRAM**) just like our Regular Computer

Unlike the Regular Computers, the micro-controllers have the Central Processing Unit (CPU), Working Memory, Permanent Storage Memory, all built into a single chip, that is why we not see things like RAM or Harddisk attached to our micro-controller, they are all inside the chip. Our PROGRAM is also stored directly into the chip

There are types many micro-controllers out there, with different size and specifications from different manufacturer. The **ATMEGA328P micro-controller** is just one of the many micro-controllers

We need to code our **PROGRAM** and put the **PROGRAM** into the **ATMEGA328P micro-controller chip**

What do we need to do ?



To Computer To Arduino Uno



Basic Hardware and Software Requirement

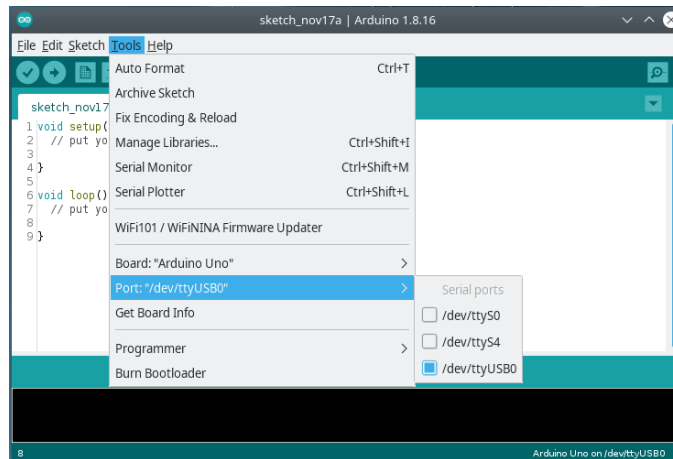
- 1x Computer with Arduino IDE Software Installed
- 1x Arduino Uno Board
- 1x USB 2.0 Type A/B Data Cable

Arduino Uno QuickStart

<https://github.com/teaksoon/lmaewapm>

Make a small PROGRAM to test our Basic Setup (Computer + Arduino IDE Software + USB Cable + Arduino Uno + ATMEGA328P micro-controller)

STEP 1/6: Test connection between the Arduino IDE Software and the USB/TTL chip on the Arduino Uno board



Test 1: Connect Arduino Uno the Desktop Computer with USB Cable, from Arduino IDE Software Menu, select **"Tools | Port >"** a new "Port" will appear.

Test 2: Disconnect Arduino Uno the Desktop Computer from USB Cable, from Arduino IDE Software Menu, select **"Tools | Port >"** a "Port" will disappear.

The **"Port"** that **"appear"** and **"disappear"** is the Connection name between our Desktop Computer and the Arduino Uno USB-TTL chip. **Make sure it is "Selected" and your Arduino IDE is ready to be used with that Arduino Uno Board.**

If you cannot see the "Port" "appear" and "disappear" in Test 1: and Test 2:



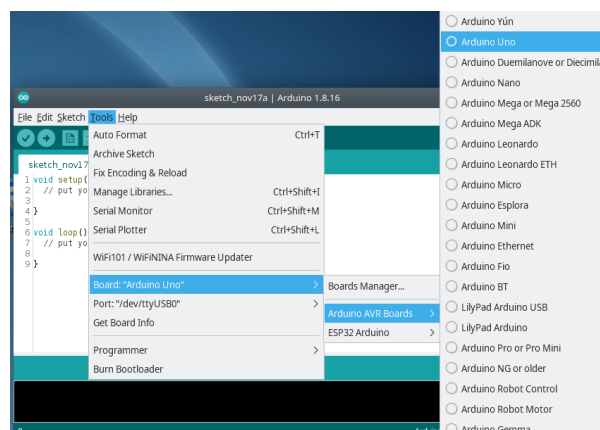
Most likely the driver(software) for the USB-TTL chip is not present on the Computer. We will need to install the USB-TTL driver. Most China made Arduino Uno clones uses the CH340 USB-TTL chip. We can download this USB-TTL driver from the following link

http://www.wch.cn/download/ch341ser_exe.html

If the Arduino Uno is using other USB-TTL chip, we have to find its driver and install it on the Computer.

STEP 2/6: Choose our Development Board. In our case, the Arduino Uno should be selected (normally Arduino Uno is already the default selection)

from Arduino IDE Software Menu, select **"Tools | Board >"**



Arduino Uno QuickStart

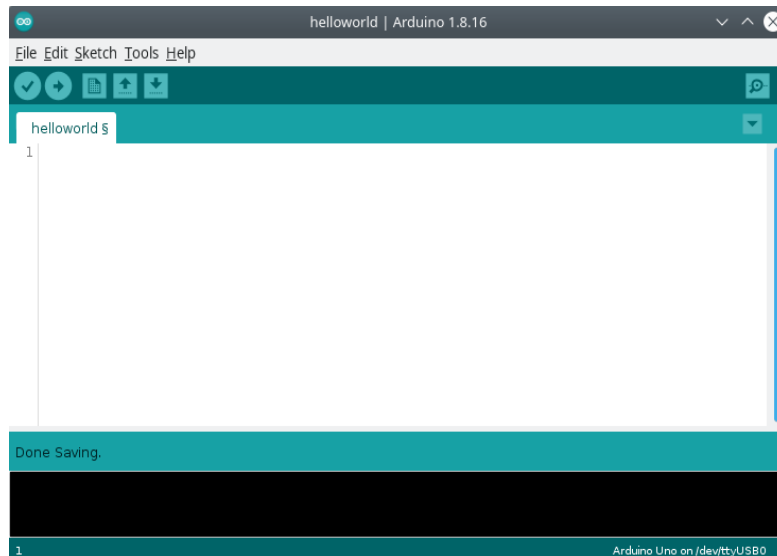
<https://github.com/teaksoon/lmaewapm>

STEP 3/6: Make a new PROGRAM using the Arduino IDE Software

The Arduino IDE Software starts with its **Code Editor** screen with a default PROGRAM loaded

To prevent potential mistakes of “over-writing” our existing PROGRAM, we do not immediately key-in our PROGRAM codes into the Arduino IDE Software Code Editor, we always start by making a new copy

1. From the Arduino IDE Software menu, select **“File | Save As...”**
2. Key in a new PROGRAM name = **“quickstart”** and then Click **“Save”** button
3. Remove all existing codes from the Arduino IDE Software Code Editor so that we can have a clean start

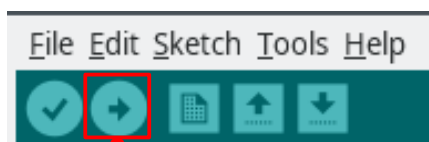


STEP 4/6: Key in the following PROGRAM codes into the Arduino IDE Software

```
void setup() {  
  pinMode(13, OUTPUT);  
}  
void loop() {  
  digitalWrite(13, HIGH);  
  delay(500);  
  digitalWrite(13, LOW);  
  delay(500);  
}
```

STEP 5/6: Upload the PROGRAM into the ATMEGA328P micro-controller from the Arduino IDE Software

Click the **“Upload Icon”** or From Arduino IDE Software menu, select **“Sketch | Upload”**



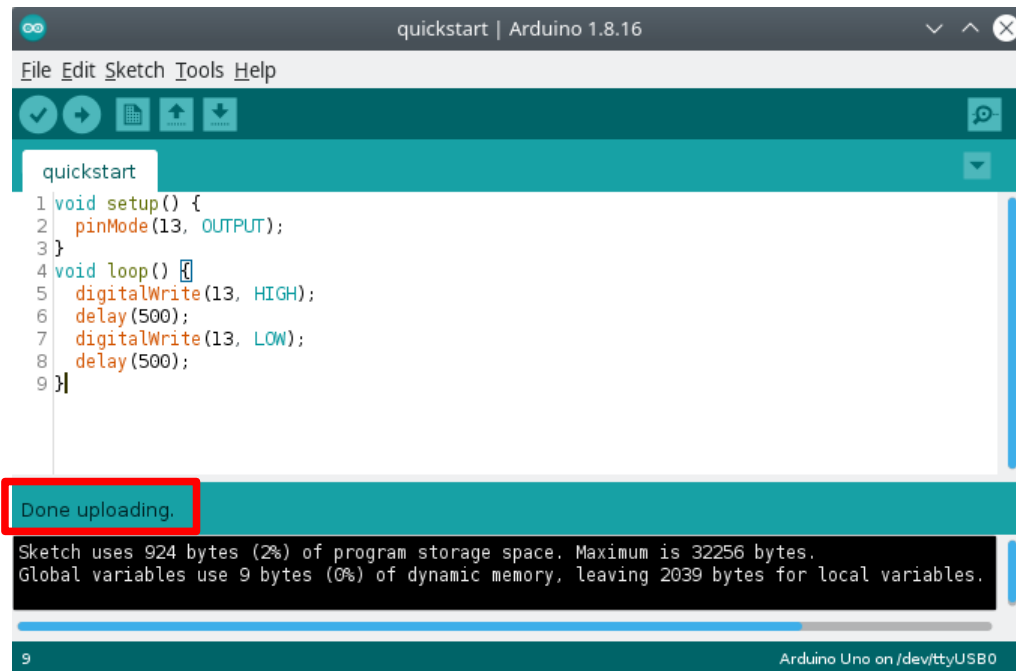
Upload Icon

Arduino Uno QuickStart

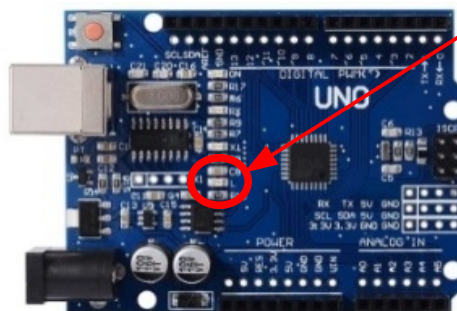
<https://github.com/teaksoon/lmaewapm>

STEP 6/6: Done Uploading

The Arduino IDE Software will show a status at the bottom of the screen "Done Uploading", means it has successfully sent our Program into the ATMEGA328 micro-controller. We are Done!!!



Watch the "test LED" on the Arduino Uno board, The "test LED" will be switched ON and OFF every 500 milliseconds



This PROGRAM will stay inside the micro-controller permanently (until we replaced it with a new upload). This PROGRAM will run automatically whenever we power-up or reset our Arduino Uno

The difference between "Compile" and "Upload" in the Arduino IDE Software

Sketch|Compile

File Edit Sketch Tools Help



"Compile" will check the PROGRAM codes for errors. If there are no errors, a "Machine Code" file will be generated. if there are errors, "Machine Code" file will not be generated and the errors will be shown at the bottom of the screen

Sketch|Upload

File Edit Sketch Tools Help



"Upload" will run the "Compile" process, if the "Machine Code" file is successfully generated by the "Compile" process, the "Upload" will continue to send the "Machine Code" file into the micro-controller. The micro-controller will then run the "Machine Code"

Arduino Uno QuickStart - SERIAL MONITOR -

<https://github.com/teaksoon/lmaewapm>

We are now able to send our PROGRAM into the ATMEGA328P micro-controller with our Arduino IDE Software "Upload" facility.

The Arduino IDE Software can also receive information from our PROGRAM running inside the Atmega328 micro-controller via the Arduino IDE Software, "Serial Monitor" facility

From the Arduino IDE Software

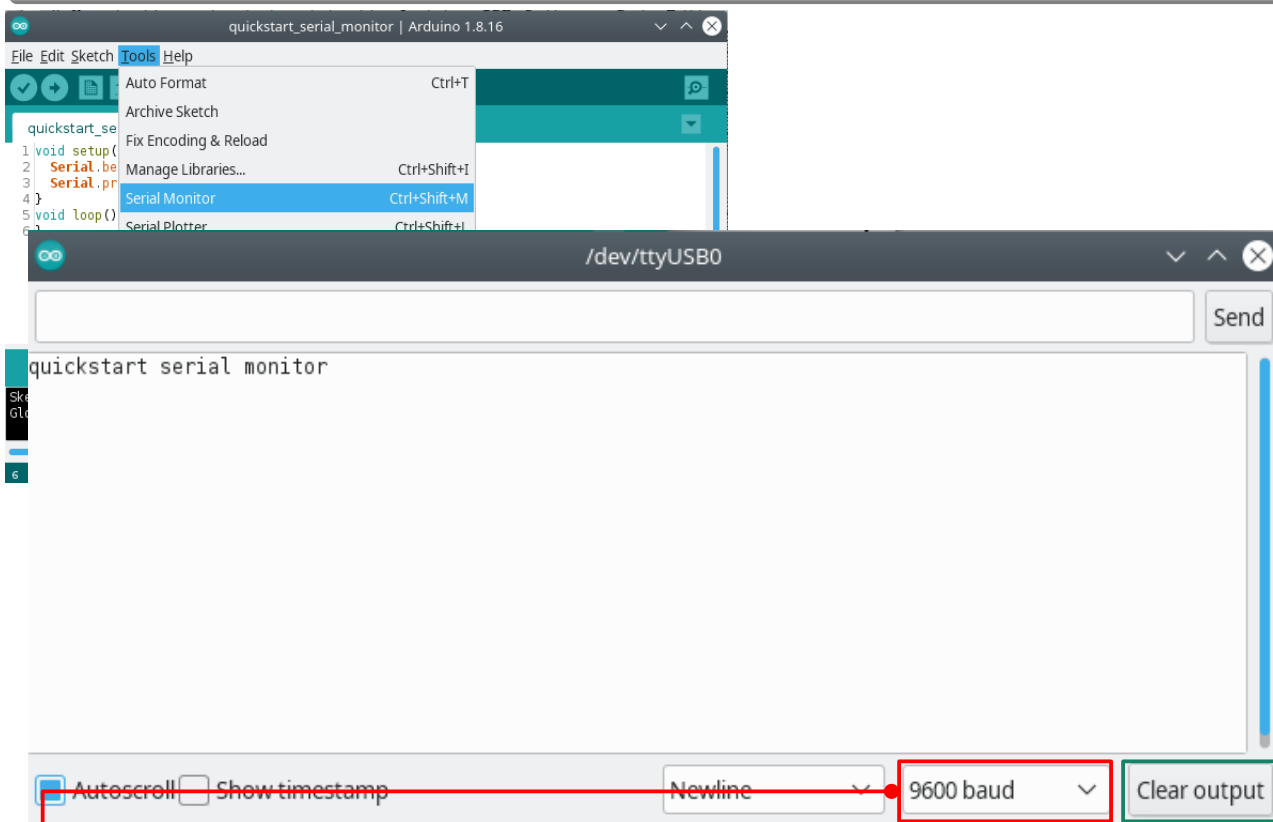
1. Save PROGRAM as "quickstart serial monitor"

2. Key in the codes below with Arduino IDE Software Code Editor and Upload

```
void setup() {  
  Serial.begin(9600);  
  Serial.println("quickstart serial monitor");  
}  
void loop() {  
}
```

This time we will not see blinking LED but we will see something on the "Serial Monitor" screen

From Arduino IDE Software menu, select **"Tools | Serial Monitor"**



Make sure "baud" is set to the same value as "Serial.begin(9600);" that is coded in our PROGRAM

Sometimes the Serial Monitor contains data from previous session, Click on "Clear Output" button and press "Reset" button on the Arduino Uno board to get a new clean session

When our ATMEGA328 micro-controller run the Serial.println("quickstart serial monitor") instruction code from our PROGRAM, the "quickstart serial monitor" text message will be sent from our ATMEGA328 micro-controller to our Computer via the USB interface, and is picked-up and displayed by our Arduino IDE Software's Serial Monitor facility