

Installing Arduino Software (IDE) & FAQ

Preparations




Prepare a USB cable (type A to B), a SunFounder MARS board and a computer to install the Arduino IDE. The computer may be running on an operating system such as Windows, Linux or Mac OS X. You may install the corresponding IDE based on the system type.

Download and Install the Arduino Software (IDE)

Before plugging the USB cable to your computer, you need to first install the Arduino IDE. Go to the Download page on the arduino.cc: <https://www.arduino.cc/en/Main/Software>



Download the Arduino Software



ARDUINO 1.6.8

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer
Windows ZIP file for non admin install

Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits

[Release Notes](#)
[Source Code](#)
[Checksums](#)

Now find the steps suitable for your operating system to install the IDE:

Windows

Linux

Mac OS X

For Windows users

There are two choices for Windows version: Installer or ZIP file. You're recommended to use the Installer since it will automatically install the driver for Arduino IDE installation. So you can just download it and run the executable file to begin installation. If you download the zip file, you need to unzip the file and install the driver by yourself.

For Installer

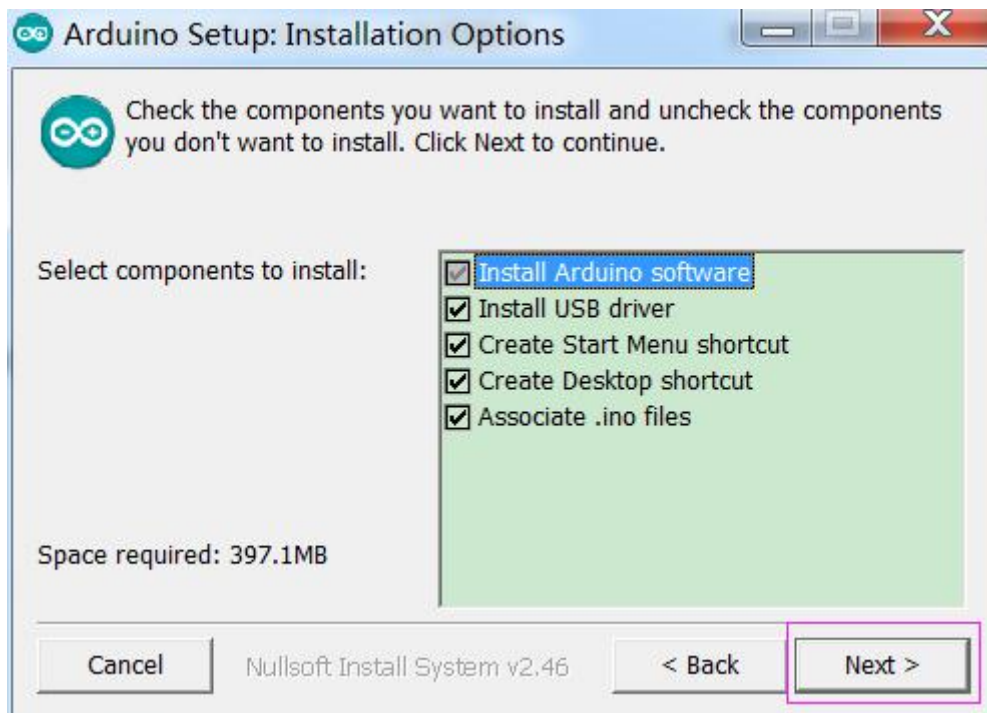
Step 1: Find the .exe file just downloaded

 `arduino-1.6.8-windows.exe`

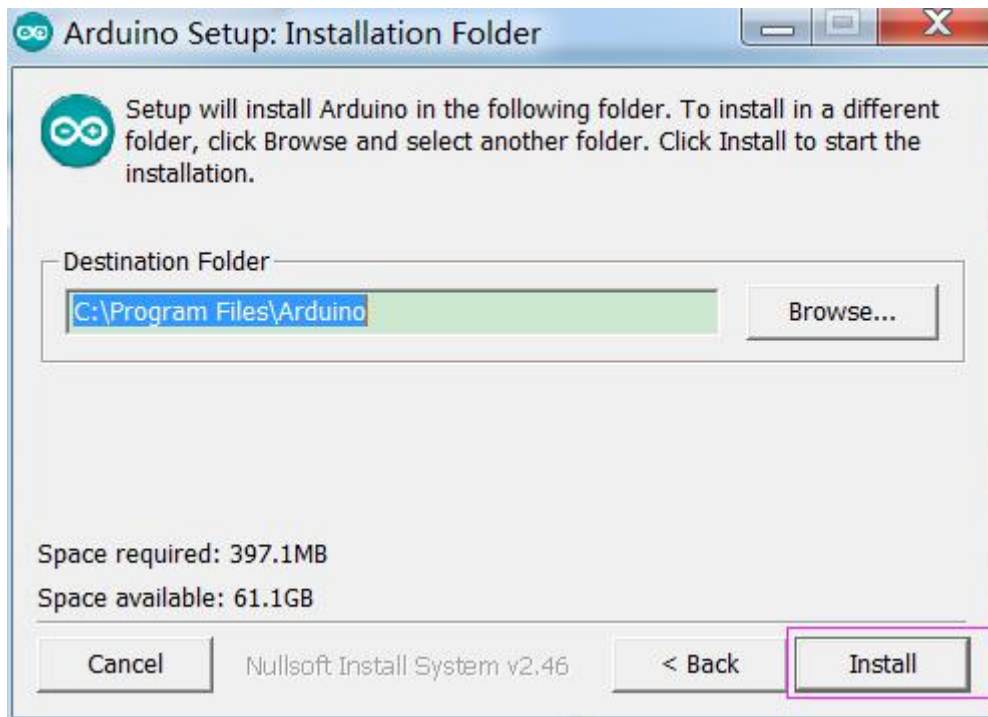
Step 2: Double click the file and a window will pop up as below. Click **I Agree**.



Step 3: Click **Next**.



Step 4: Select the path to install. By default, it's set in the C disk: `C:\Program Files\Arduino`. You can click Browse and choose another path. Click **OK**. Then click Install.



Step 5: After the installation is done, click **Close**.

Please note that the new IDE may prompt errors when you're compiling code under Windows XP. So if your computer is running on XP, you're suggested to install Arduino 1.0.5 or 1.0.6. Also you can upgrade your computer.

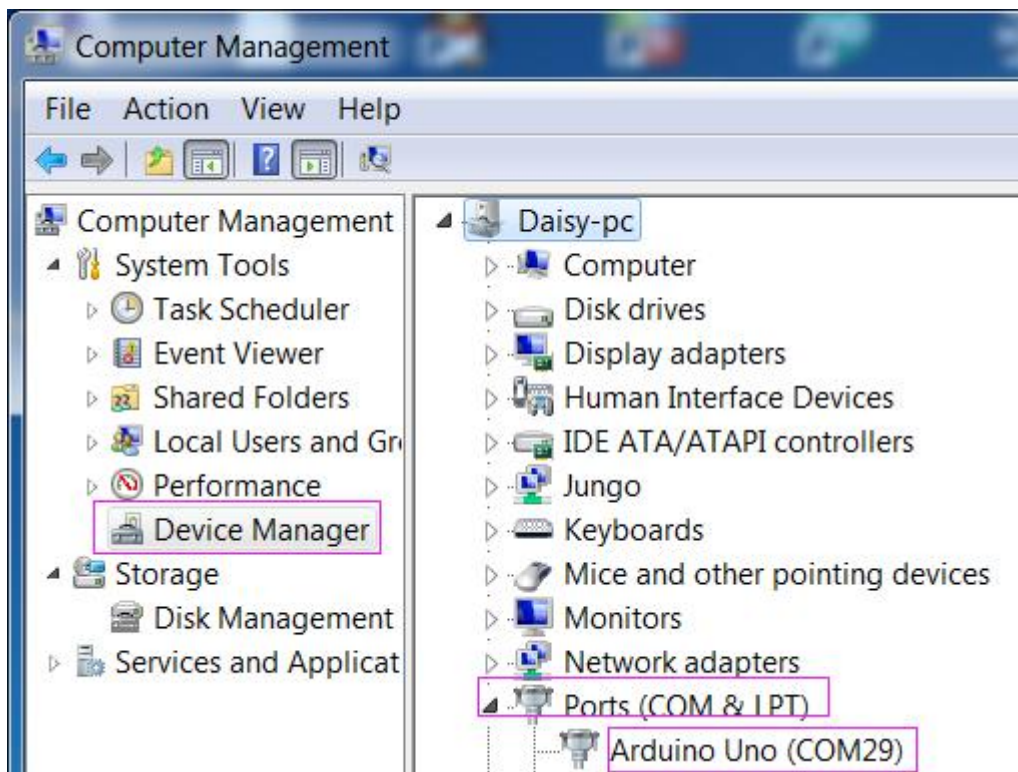
Plug in the board

Connect the MCU to your computer via a USB cable (type-A to B). The cable can be used not only to upload sketches to the board, but also supply power. After plugging in, the green LED on the board labeled with ON, will brighten. The yellow LED, labeled with L, will flicker first and then keep steady on.

Install the driver

If you downloaded the installer, upon your connecting the board to the computer, the system will help you install the driver automatically. After a while, a prompt may appear in the taskbar indicating the driver is installed successfully. It depends on computers. So if you don't see it in yours, it's still OK.

You can also check the port on **Device Manager**. Right click on your computer and click **Manage**. Find Device Manager under **Computer Management (local)**. Then you can see the Mega 2560 at COM54. The other boards are similar. It's normal if yours is different from mine. So now the board is recognized by the computer.



If you download the zip file before, when you connect the MCU to the computer, it may not be recognized (it can happen). Then you need to install the driver manually. Take the following steps.

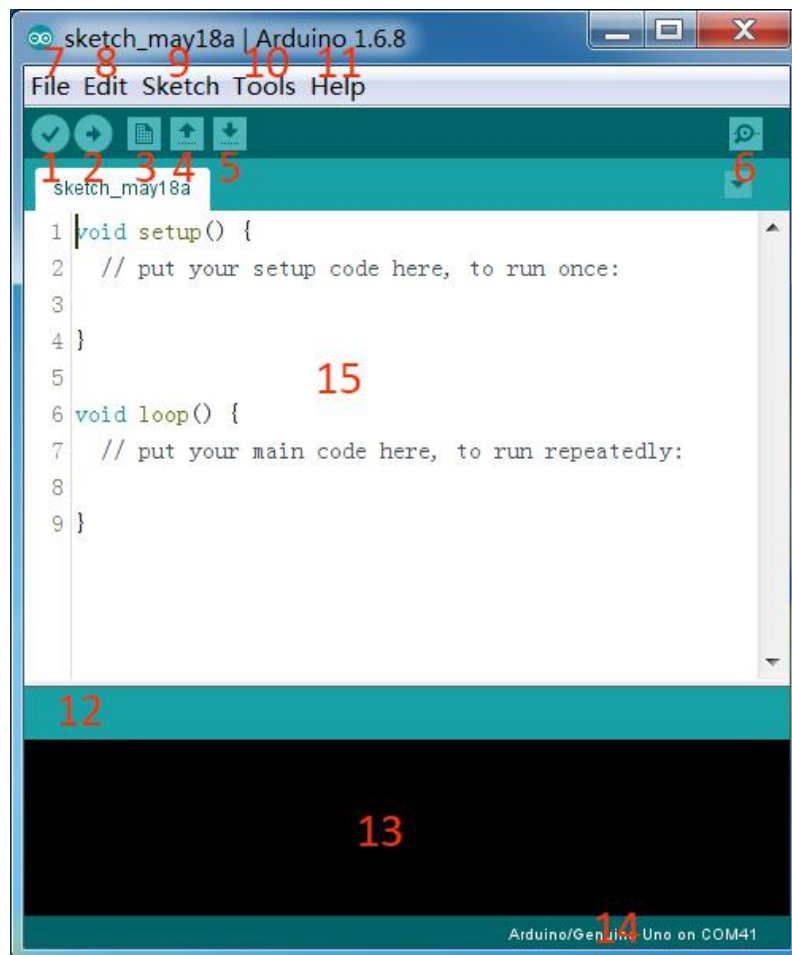
- A: Extract the zip file you downloaded. Then right click on **Computer** and select **Management** or **Device Manager**. Find **Other Devices**.
- B: Right click on **Unknown Devices** or **Arduino board (COMxx)** and select **Update Driver Software**.
- C: Choose the second option, **Browse my computer for Driver software**.
- D: A window pops up then. Click **Browse**. Then go to the folder where you just extracted the file. Go to the *drivers* folder and click **OK** -> **Next**. It may need a sec. Then the system prompts you the driver has been installed successfully. And the driver is for Uno. So the computer can recognize the board now. Click **Close**.

Open the Arduino Software (IDE)

Double-click the Arduino icon (arduino.exe) created by the installation process



Then the Arduino IDE will appear. Let's check details of the software.



1. **Verify:** Compile your code. Any syntax problem will be prompted with errors.
2. **Upload:** Upload the code to your board. When you click the button, the RX and TX LEDs on the board will flicker fast and won't stop until the upload is done.
3. **New:** Create a new code editing window.
4. **Open:** Open an .ino sketch.
5. **Save:** Save the sketch.
6. **Serial Monitor:** Click the button and a window will appear. It receives the data sent from your control board. It is very useful for debugging.
7. **File:** Click the menu and a drop-down list will appear, including file creating, opening, saving, closing, some parameter configuring, etc.
8. **Edit:** Click the menu. On the drop-down list, there are some editing operations like Cut, Copy, Paste, Find, and so on, with their corresponding shortcuts.
9. **Sketch:** Includes operations like Verify, Upload, Add files, etc. More important function is Include Library – where you can add libraries.
10. **Tool:** Includes some tools – the most frequently used Board (the board you use) and Port (the port your board is at). Every time you want to upload the code, you need to select or check them.
11. **Help:** If you're a beginner, you may check the options under the menu and get the help you need,

including operations in IDE, introduction information, troubleshooting, code explanation, etc.

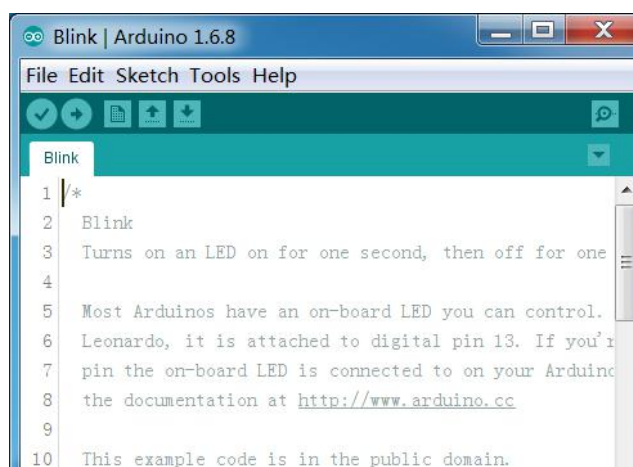
12. In this message area, no matter when you compile or upload, the summary message will always appear.

13. Detailed messages during compile and upload. For example, the file used lies in which path, the details of error prompts.

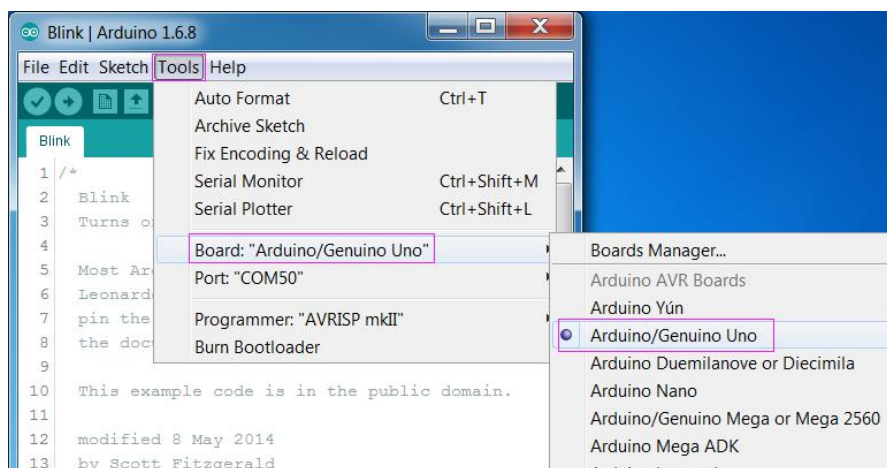
14. **Board and Port:** Here you can preview the board and port selected for code upload. You can select them again by **Tools -> Board / Port** if any is incorrect.

15. The editing area of the IDE. You can write code here.

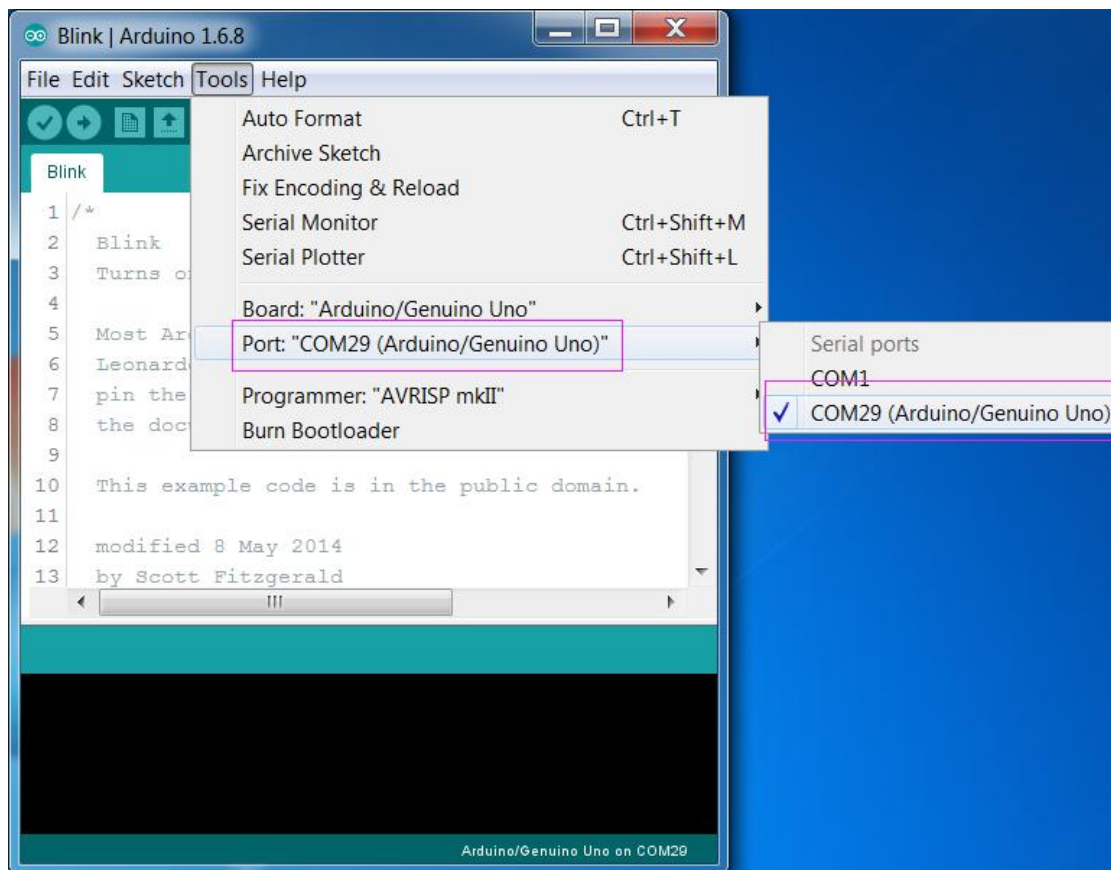
Now let's some basic operations in IDE with an example code. Click **File > Examples > 01.Basics > Blink** and a new window will show up.



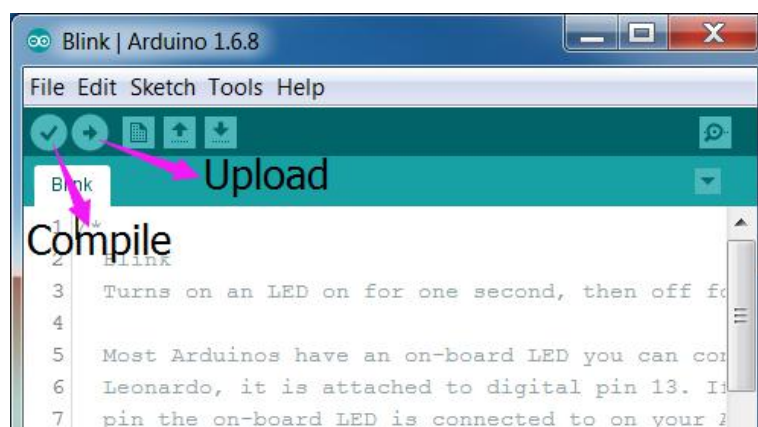
Click **Tools -> Board** and select **Arduino/Genuino Uno**.



Then **Tools -> Port** and select the COM port you just checked on **Device Manager** or the system prompted you in the taskbar when you plug in the board.



Click the **Upload** icon to upload the code to the board. And the icon **Compile** to compile sketches (usually used to refer to a code file), which always checks the code. Also when you click **Upload**, the code will be compiled. The sketches can be uploaded to the board when there is nothing wrong with them. Therefore, generally you just need to click **Upload**. During the upload, the TX LED and the RX LED will be alternately flickering. It means the board is sending signal to the computer and then receives the signal from the latter. After upload is completed, the two LEDs will go out.

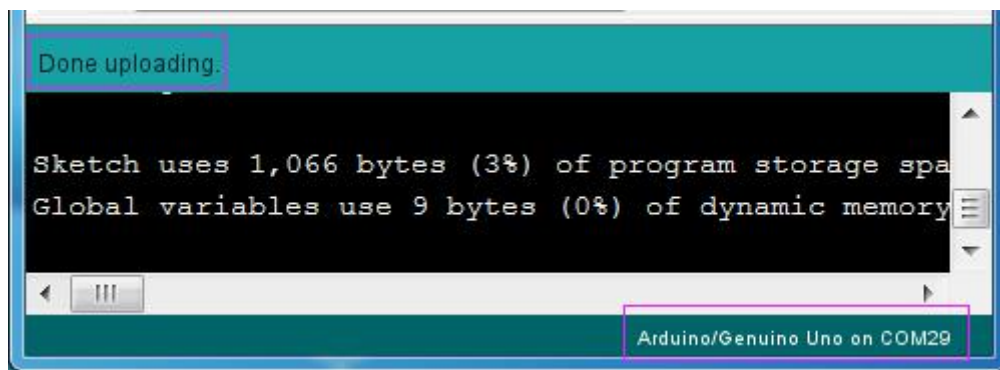


If "Done uploading" appears at the bottom of the window, it means the sketch has been successfully uploaded. And if you see the pin 13(L) LED starts to flicker, it means the code has been successfully run.

For more details about Arduino IDE, go to **Learning->Getting started->Foundation** on the [arduino.cc](http://www.arduino.cc) and click **Arduino Software (IDE)** on the page <http://www.arduino.cc/en/Guide/Environment>

If your sketch fails upload, on the same page click **Troubleshooting**

<http://www.arduino.cc/en/Guide/Troubleshooting>.

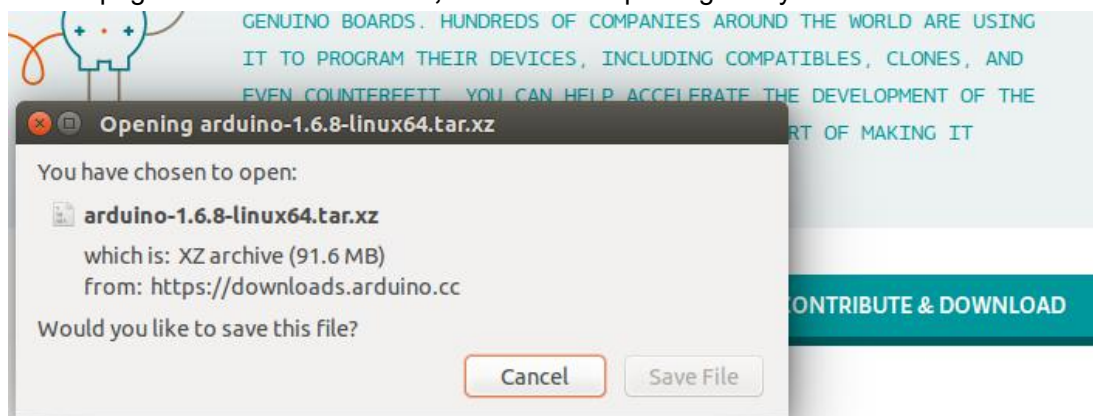


Notes:

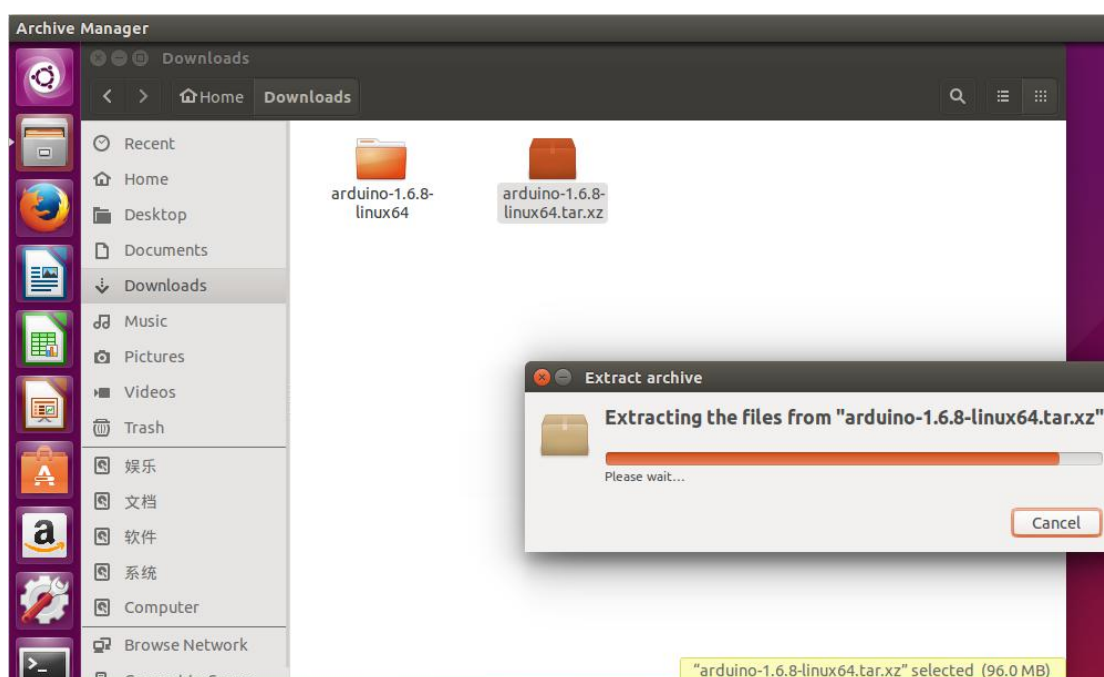
- If your computer is running on the Windows XP system, the new version IDE will prompt errors when running the code. You are recommended to download the Arduino 1.0.5 or Arduino 1.0.6. Or you can also upgrade your Window system.

For Linux Users

On the page of **Arduino Software**, download the package for your 32-bit or 64-bit Linux.



Extract the zip file after download. Usually the downloaded file is saved under the **Download** folder.

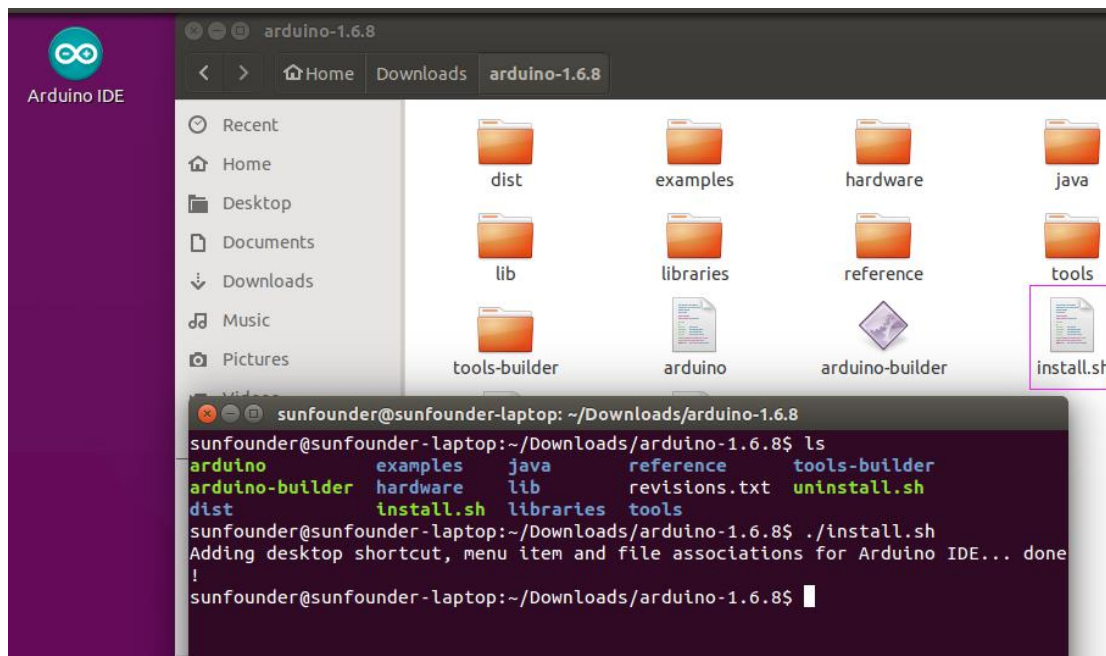


Install Arduino Software (IDE)

1. Open the extracted folder and find the installation file *install.sh*. Right click on it and selected **Run in Terminal** in the drop-down list. Then the system will install the Arduino software. After installation, you'll see the IDE icon on your desktop.



If you don't see any **Run in Terminal** when you right click on the icon, don't worry. You still have another method to accomplish it. In the folder, right click on a blank area and select **Open in Terminal** in the drop-down list. Now a window pops up and you've at the folder where the file *install.sh* lies. Type in `./install.sh` in the terminal to run the file. After a while, the software is installed successfully. You can see the IDE icon on the desktop. Now you can close the terminal and folder.

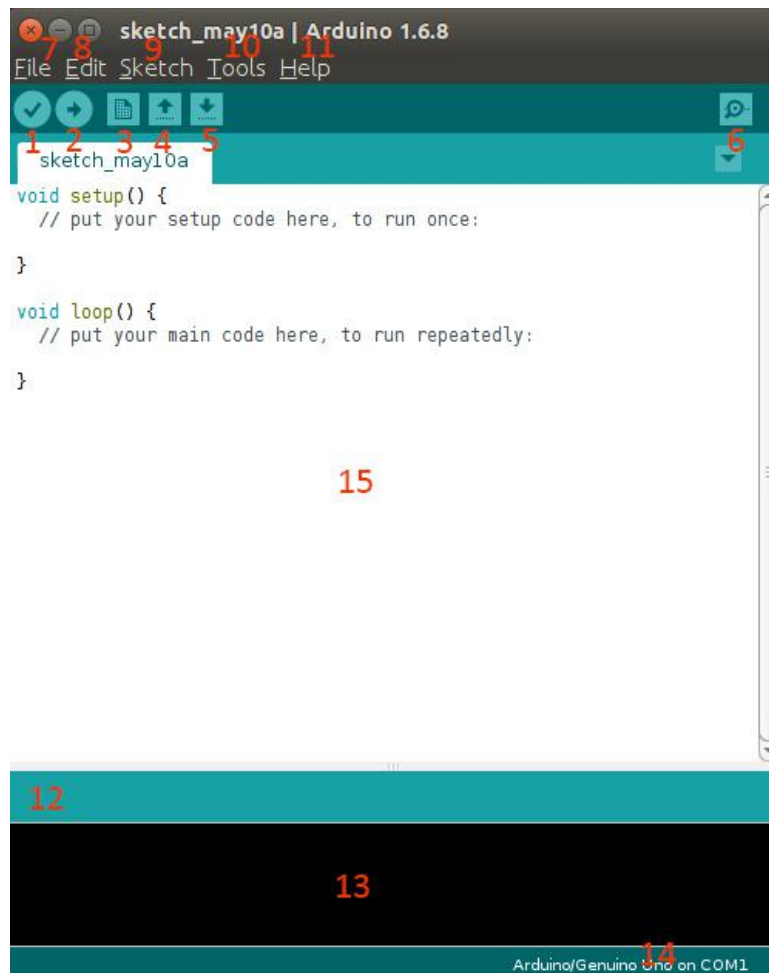


Plug in the Board

Connect the board with the computer via a USB cable (A to B). The USB cable can be used not only to upload sketches to the board, but also supply power. After plugging in, the green LED on the board labeled with ON will brighten. The yellow LED labeled with L, will flicker first and then keep steady on. Generally you do not need to install driver since when you plug the board in, the system will do that automatically.

Open the IDE

Double click the Arduino IDE icon and the Arduino IDE window will appear. Now check the operations in the window. They are the same with those for Windows.



1. **Verify:** Compile your code. Any syntax problem will be prompted with errors.
2. **Upload:** Upload the code to your board. When you click the button, the RX and TX LEDs on the board will flicker fast and won't stop until the upload is done.
3. **New:** Create a new code editing window.
4. **Open:** Open an .ino sketch.
5. **Save:** Save the sketch.
6. **Serial Monitor:** Click the button and a window will appear. It receives the data sent from your control board. It is very useful for debugging.
7. **File:** Click the menu and a drop-down list will appear, including file creating, opening, saving, closing, some parameter configuring, etc.
8. **Edit:** Click the menu. On the drop-down list, there are some editing operations like Cut, Copy, Paste, Find, and so on, with their corresponding shortcuts.
9. **Sketch:** Includes operations like Verify, Upload, Add files, etc. More important function is Include Library – where you can add libraries.
10. **Tool:** Includes some tools – the most frequently used Board (the board you use) and Port (the port your board is at). Every time you want to upload the code, you need to select or check them.
11. **Help:** If you're a beginner, you may check the options under the menu and get the help you need,

including operations in IDE, introduction information, troubleshooting, code explanation, etc.

12. In this message area, no matter when you compile or upload, the summary message will always appear.

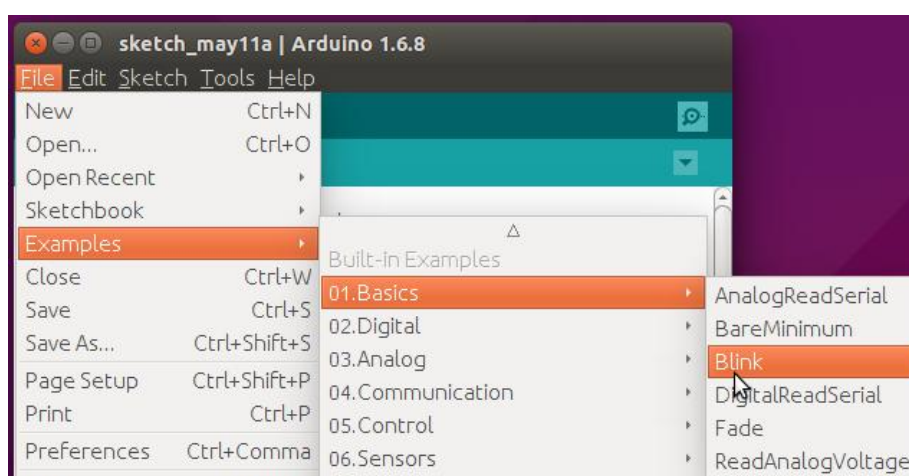
13. Detailed messages during compile and upload. For example, the file used lies in which path, the details of error prompts.

14. **Board and Port:** Here you can preview the board and port selected for code upload. You can select them again by **Tools -> Board / Port** if any is incorrect.

15. The editing area of the IDE. You can write code here.

Now let's some basic operations in IDE with an example code.

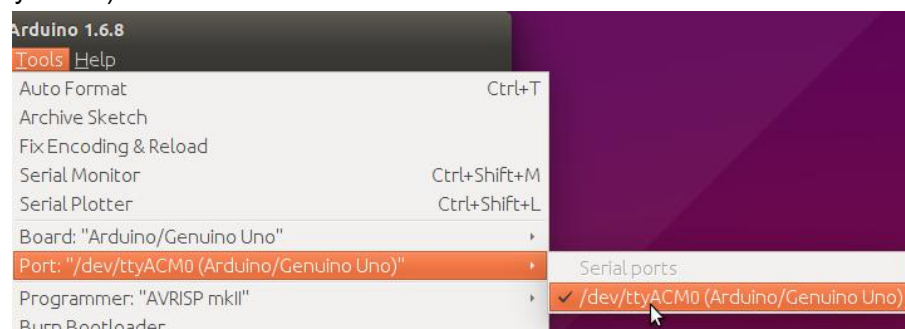
Open a file in the Examples and check the details. There are many example sketches under the Examples which are built-in code files in the Arduino software. You can start from these files if you're just starting to learn Arduino.



Select the correct board before uploading.

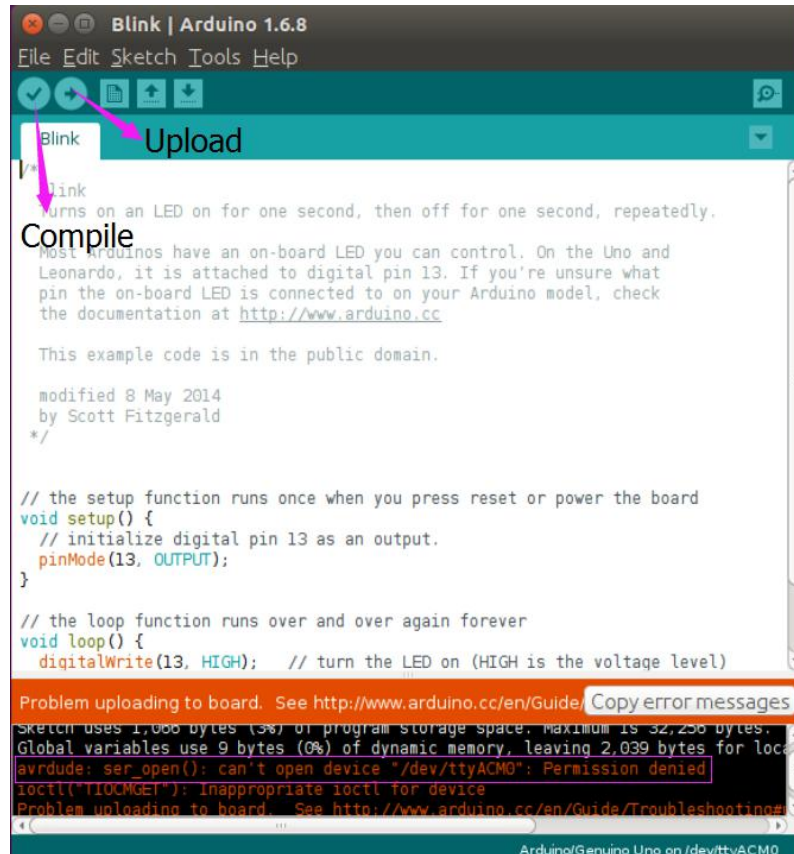


and Port (/dev/ttyACM0)



Click the **Upload** icon to upload the code to the board. And the icon **Compile** to compile sketches (usually used to refer to a code file), which always checks the code. Also when you click **Upload**, the code will be compiled. The sketches can be uploaded to the board when there is nothing wrong with them. Therefore, generally you just need to click **Upload**.

During the upload, the TX LED and the RX LED will be alternately flickering. It means the board is sending signal to the computer and then receives the signal from the latter. After upload is completed, the two LEDs will go out.



If you get an error message: `avrdude:ser_open():can't open device "/dev/ttyACM0":Permission denied`, it means you need to enable the permission for port. Take the following steps:

Open a terminal and type in the command:

```
ls -l /dev/ttyACM*
```

Then you will see:

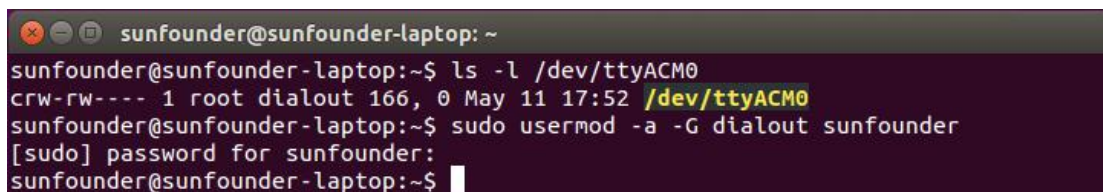
```
crw-rw---- 1 root dialout 166,0 May 11 17:52 /dev/ttyACM0
```

The data we need is "dialout" (the group owner of the file).

Now we just need to add our user to the group:

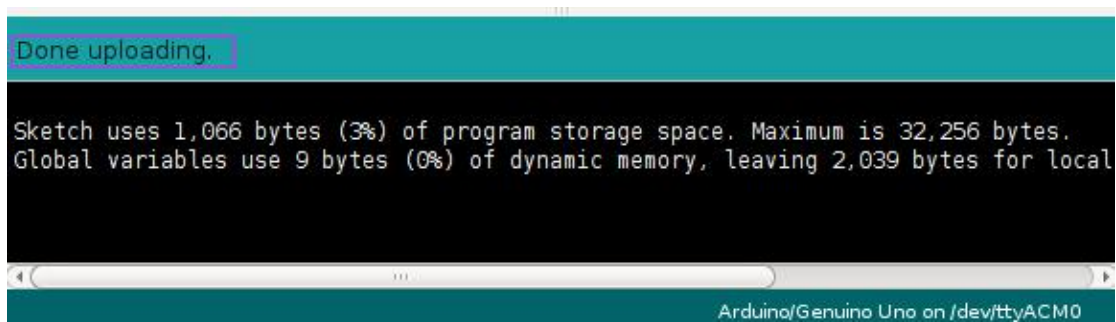
```
sudo usermod -a -G dialout <username>
```

where <username> is your linux user name. Then type in the password.



Note: After the modification, you need to log out and log in to take it into effect.

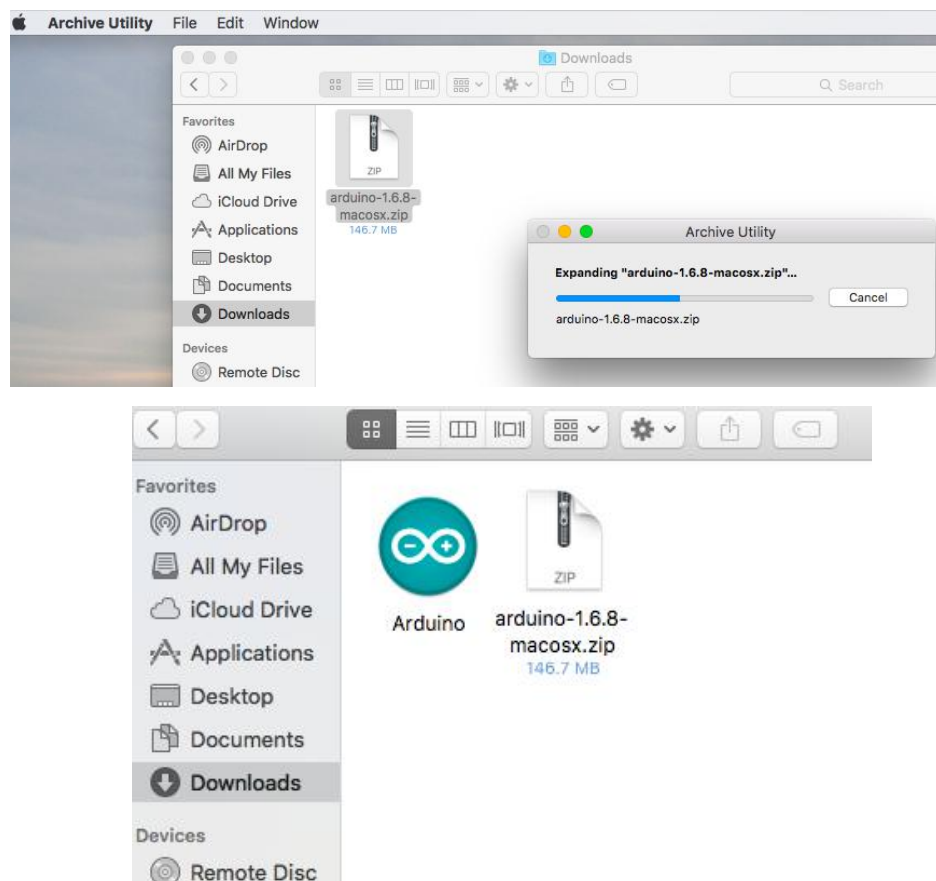
Now open the Arduino IDE again. Upload the code. If "Done uploading" appears at the bottom of the window, it means the sketch has been successfully uploaded. And if you see the pin 13(L) LED starts to flicker, it means the code has been successfully run.



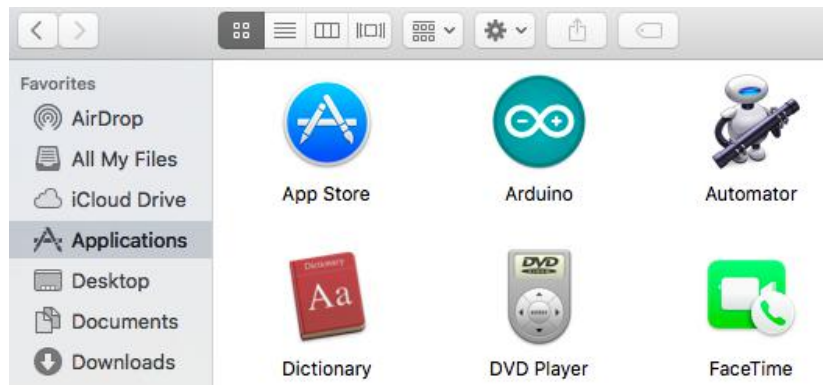
Mac OS X

Download the IDE

- 1) On the page of **Arduino Software**, download the package for **Mac OS X**. After download is done, double click the zip file to unzip.



- 2) Following that, you'll need to copy the **Arduino** into **your applications folder** to complete installation.



Plug in the Board

Connect the board with the computer via a USB cable (A to B). The USB cable can be used not only to upload sketches to the board, but also supply power. After plugging in, the green LED on the board labeled with ON will brighten. The yellow LED labeled with L, will flicker first and then keep steady on. Generally you do not need to install driver since when you plug the board in, the system will do that automatically.

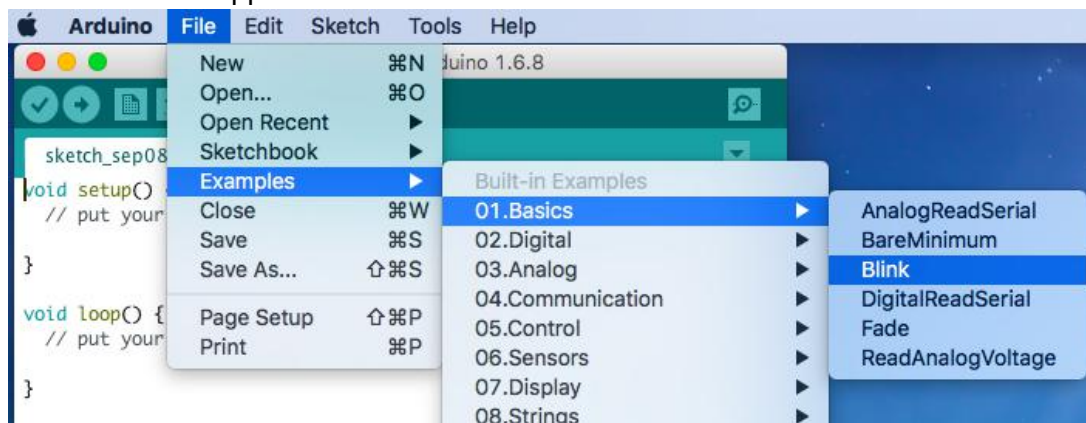
The serial port for the controller will be assigned by the computer.

Using the IDE

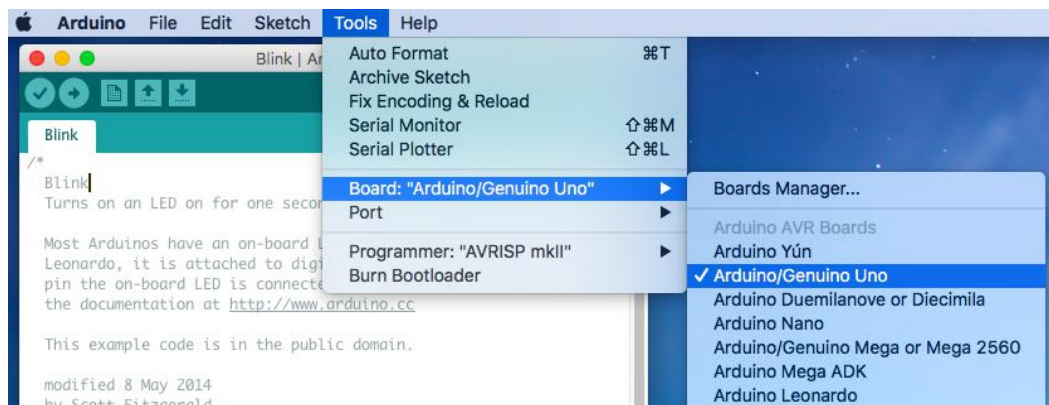
Double-click the Arduino icon (arduino.exe) created by the installation process



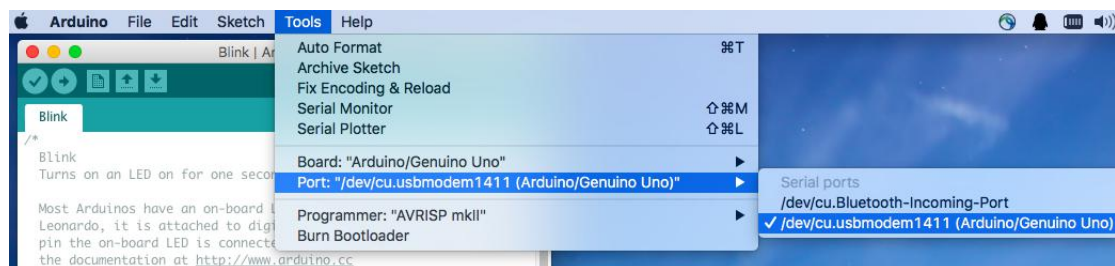
Then the Arduino IDE will appear. Let's check details of the software.



- 1) Before uploading the code to your board, first you need to select the correct board and the correct port. (Uno, etc.)

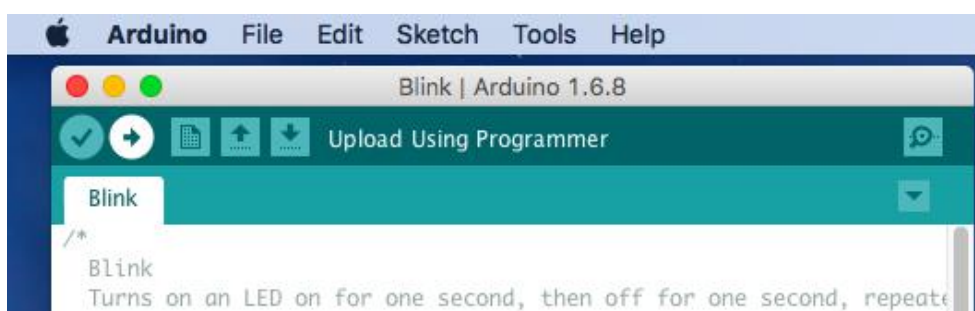


then port, The port here (`/dev/cu.usbmodem1411`) may be different from the Arduino board name.

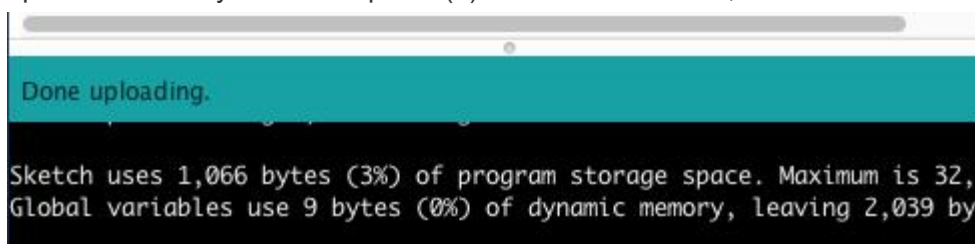


4) Click the **Upload** icon to upload the code to the board. And the icon **Compile** to compile sketches (usually used to refer to a code file), which always checks the code. Also when you click **Upload**, the code will be compiled. The sketches can be uploaded to the board when there is nothing wrong with them. Therefore, generally you just need to click **Upload**.

During the upload, the TX LED and the RX LED will be alternately flickering. It means the board is sending signal to the computer and then receives the signal from the latter. After upload is completed, the two LEDs will go out.



5) If "Done uploading" appears at the bottom of the window, it means the sketch has been successfully uploaded. And if you see the pin 13(L) LED starts to flicker, it means the code has been successfully run.



FAQs

External Power Source for Mars and Mercury

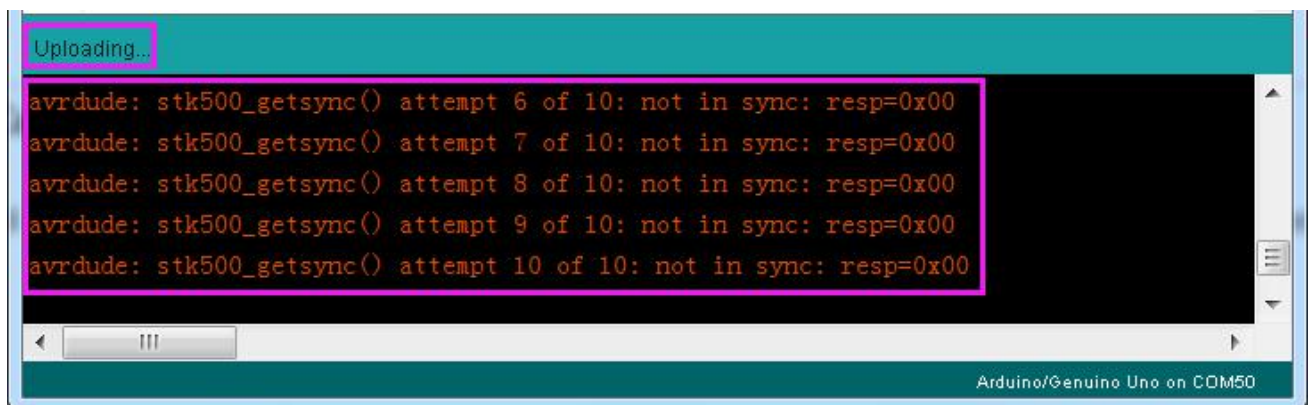
If you want to take the board out for use, you can power the circuit with an external supply. It should be within **7V-12V** (it'd better not be lower than 6.5V, or higher than or equal to 12V). Also it should has a **5.5x2.1mm DC Power Plug Male**.

You're recommended to use a 9V-DC-Jack battery holder, 9V Battery Buckle Connector and a **9V Power Plug Adapter**.



Error: avrdude: stk500_getsync(): not in sync: resp=0x00

If an error "**avrdude: stk500_getsync() attempt 10 of 10: not in sync: resp=0x00**" prompts, it means you haven't selected the right board. In Arduino IDE, select **Tools-> Board** and select the correct board.



Error: avrdude: ser_open(): can't open device "\\.\COM50": The system cannot find the file

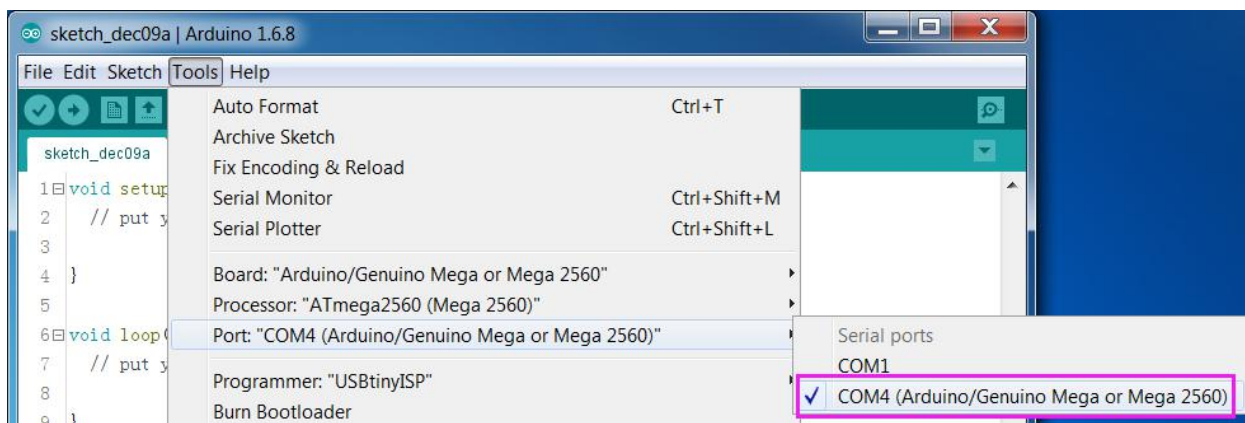
```
Uploading...

Using Port          : COM50
Using Programmer    : arduino
Overriding Baud Rate : 115200
avrdude: ser_open(): can't open device "\\.\COM50": The system cannot find the file specified.
```

If this error occurs, it may be a wrong port, nonexistent COM50 or an unrecognized Arduino board on your computer.

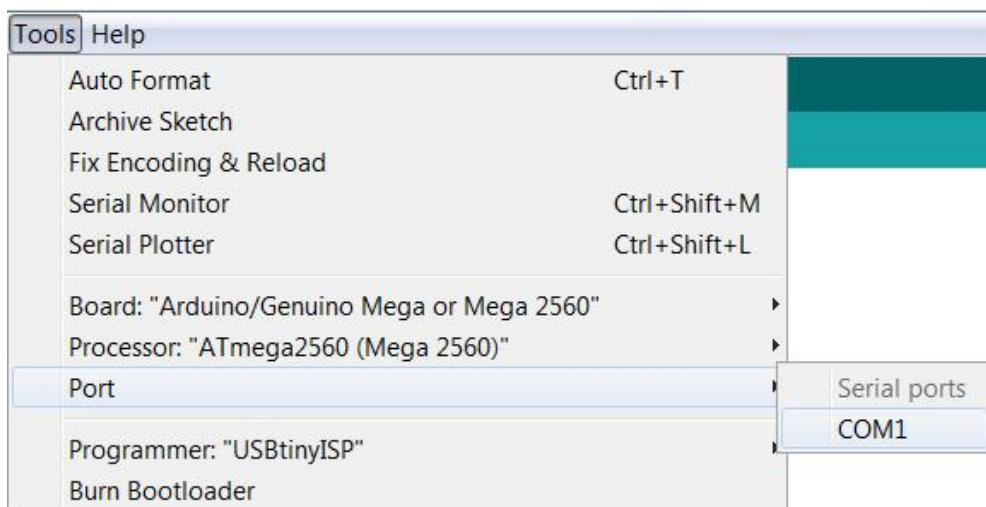
1) Port is wrong; COM50 does not exist

Select **Tools** -> **Port**, click the right port and upload the sketch.



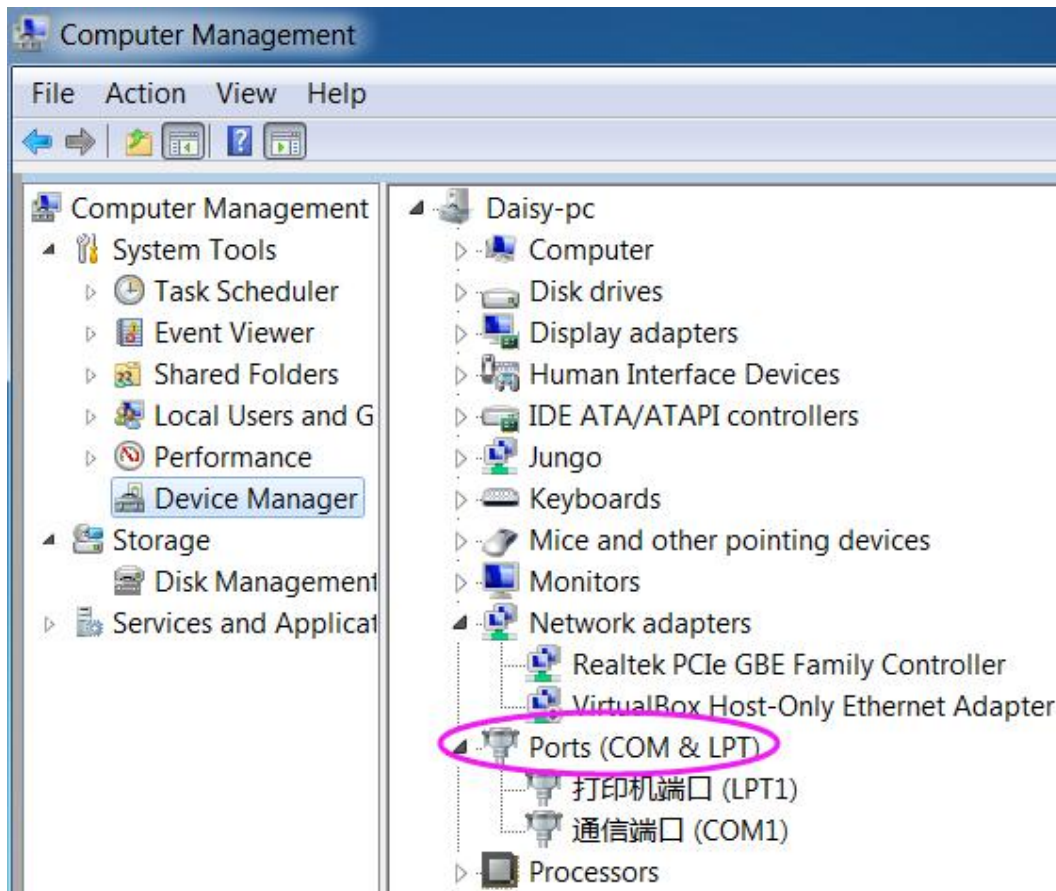
2) Arduino board is not recognized

Select **Tools** -> **Port**. If there's just COM1 or Port is grey and unselectable, it means that the Arduino board is not recognized.



You need to check whether the board is still being installed. If not, go to **Device Manager**. If

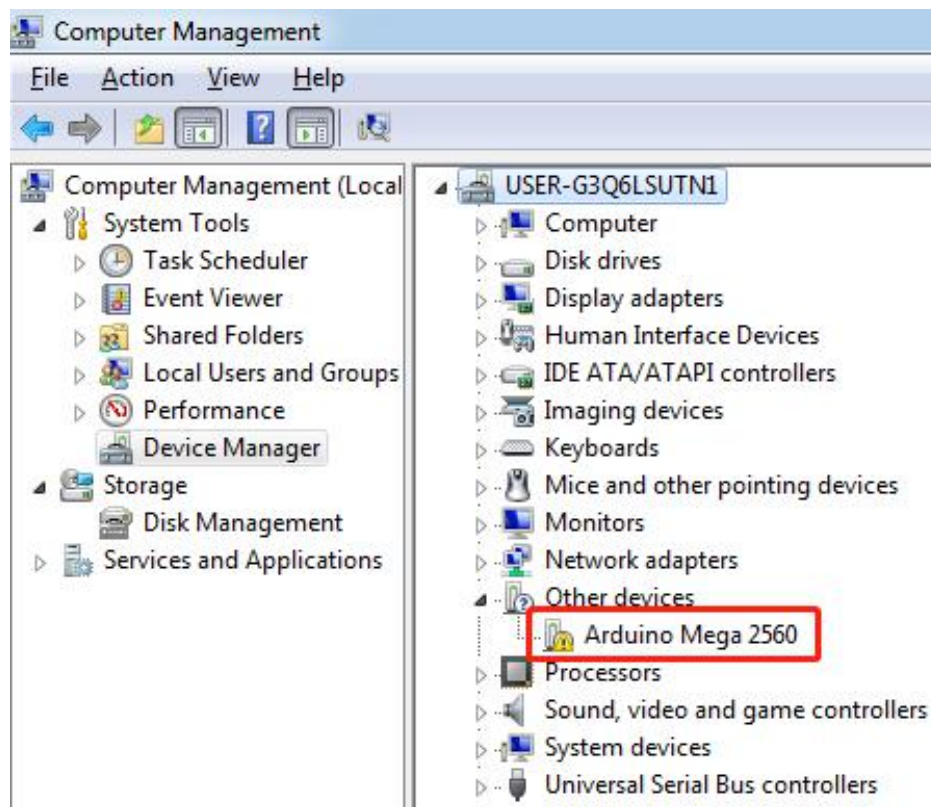
there's no port or any unrecognizable device under **Port(COM&lpt)**, you need to check whether the Arduino board has been plugged into the computer successfully, and whether the power LED on the board brightens or not.



If all work well, it can be that the Bootloader or firmware of the board is missing. You can find the instruction articles under WIKI on our website.

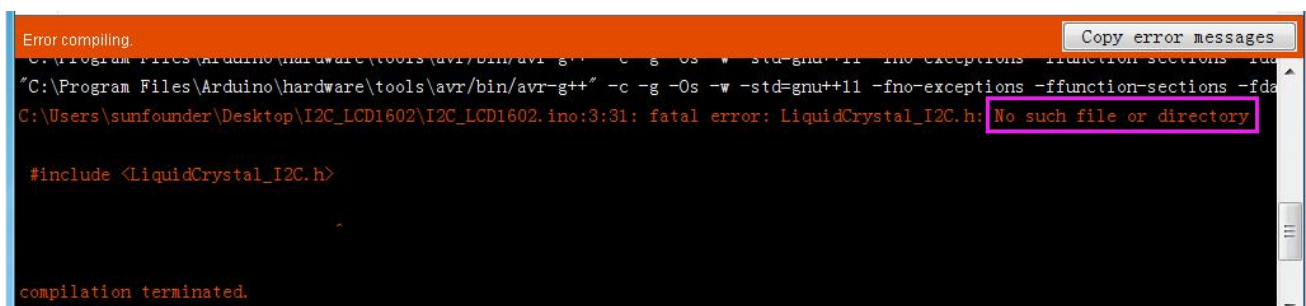
[http://wiki.sunfounder.cc/index.php?title=How to Reburn the firmware onto Atmega16u2](http://wiki.sunfounder.cc/index.php?title=How_to_Reburn_the_firmware_onto_Atmega16u2)
http://wiki.sunfounder.cc/index.php?title=Arduino_Bootloader_Burned_Method

But if **Arduino Mega2560** with an exclamation mark icon appears under **Other devices**, you'll need to install the driver manually, please refer to the Lesson 1 Install and introduce Arduino IDE.



This may be a problem encountered for old versions of IDE. Now it should be fixed in new releases.

No such file or directory



If "xxxxx.h : No such file or directory" appears, it means the library is not added. E.g. "LiquidCrystal_I2C.h: No such file or directory." is displayed, so you need to find this library LiquidCrystal_I2C.h and add it to Arduino/libraries.

For more FAQs, please check the related topic under FORUM on our website:

Forum Name

[FORUM RULES & FAQS – Guidelines for Forum behaviors, posting, etc.](#)

Created: Feb 22, 2016 09:48 am

Created By: admin

We'll keep updating the questions and if you have any problems, feel free to ask our engineers! We'll reply as quickly as possible.

Post

FAQs

These frequently asked questions are mainly from our customers. After reorganization and selection, they come to some basic areas.

Before asking on our website, you can first check here and see if your question can be answered yet. If the answer here cannot unlock the mystery or you get ot
PS: Updates will be made irregularly.

1. [How to connect components on a "breadboard"](#)
2. [How to select the right Board in Arduino IDE](#)
3. [How to set Arduino IDE for my Uno/Mega/Nano...](#)
4. [How to identify the resistance of our resistors](#)
5. [How to power Arduino boards with extra sources](#)
6. [How to open Serial Monitor window](#)
7. [What's the difference between UNO and MEGA boards? How to apply other board for the kit user manual?](#)
8. [How to set up Raspberry Pi at first?](#)
9. [What can I, a complete noob, do with Sensor Kit v2.0 for Raspberry Pi](#)
10. [I don't quite understand the pins on the Raspberry Pi GPOI Extension Board. They seem to be different from those on a Pi. How to figure out?](#)
11. [Some basic functions in IDE](#)
12. [I want to power the Raspberry Pi with a 5V extra power. What do I need to pay attention?](#)
13. [How to power Arduino Uno with extra power sources?](#)
14. [When I plug the USB cable to the PC, it cannot identify the board. What should I do?](#)