

Lesson 1 - Getting Started with Arduino and UNO

Points of this section

You will learn:

- ◆ How to install the Arduino Software (IDE) on Windows PCs
- ◆ Use your uno on the arduino desktop ide and upload your first sketch

Need to prepare:

- ◆ A computer with a reliable Internet connection
- ◆ A UNO R3 board
- ◆ A USB cable

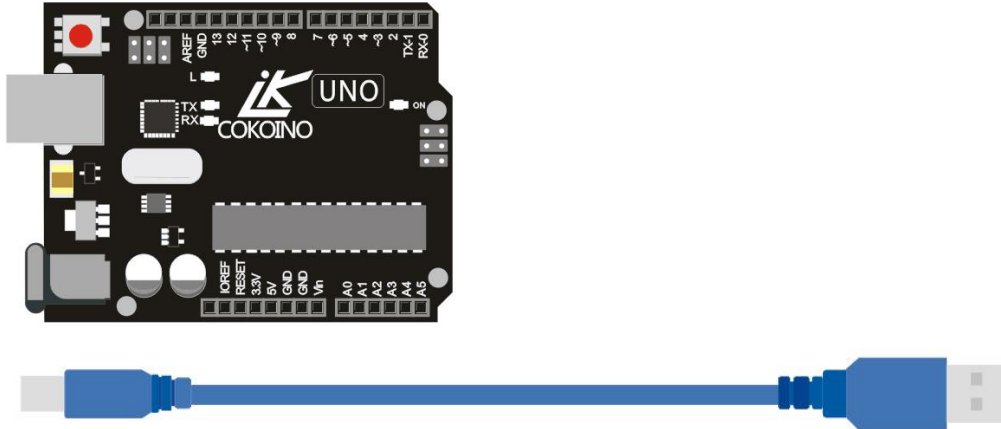


Table of Contents

The Arduino IDE.....	3
Install the Arduino Software (IDE) on Windows PCs.....	3
Download the Installer (.exe).....	4
Download the Zip package.....	5
Install The Board Drivers.....	6
The functions of each button on the Toolbar are listed below:	9
Introduction to the arduino IDE.....	10
Open your first sketch.....	14
arduino library file.....	17
What is the arduino library file.....	17
Where is the library file placed?.....	18

The Arduino IDE

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.

Install the Arduino Software (IDE) on Windows PCs

Get the latest version from this link (<https://www.arduino.cc/en/Main/Software>). You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. The Zip file is also useful if you want to create a portable installation.



ARDUINO 1.8.9

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, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
[Get](#)

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

HOME STORE SOFTWARE EDUCATION RESOURCES COMMUNITY HELP

Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **33,452,969** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$3 **\$5** **\$10** **\$25** **\$50** **OTHER**

[JUST DOWNLOAD](#) [CONTRIBUTE & DOWNLOAD](#)

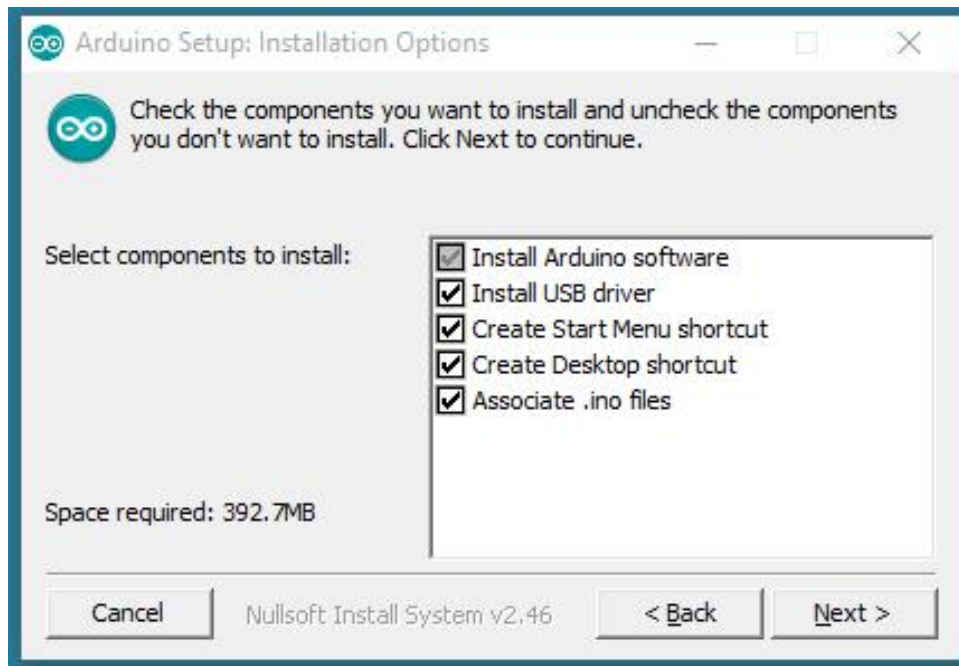
Previous Releases

Download the [previous version](#) of the current release the classic Arduino 1.0.x, or the Arduino 1.5.x Beta version.

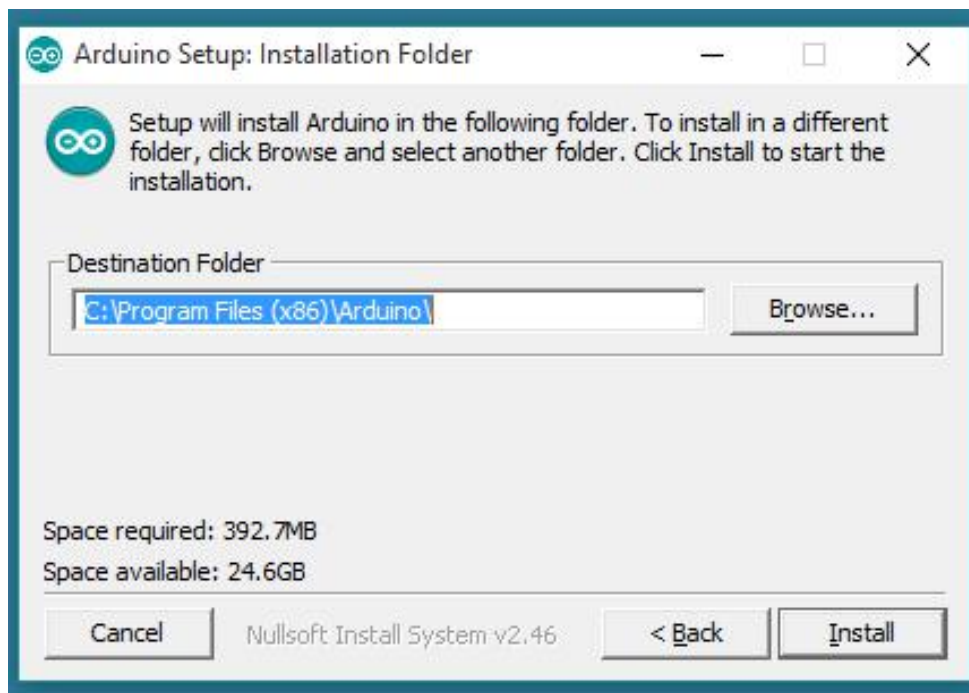
<https://www.arduino.cc/en/Main/OldSoftwareReleases#previous>

Download the Installer (.exe)

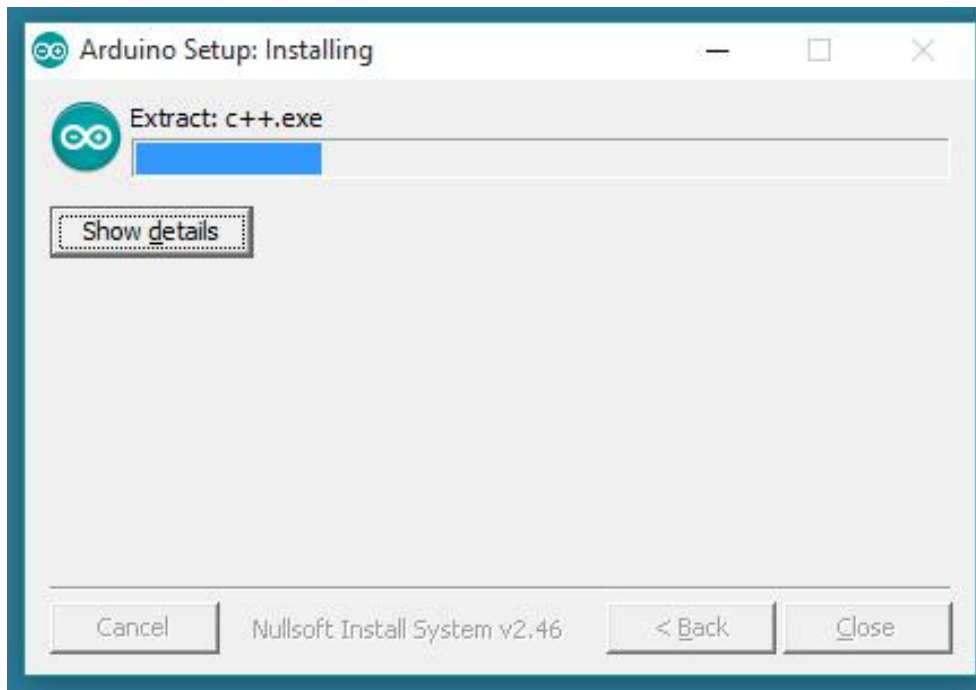
When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.



Choose the components to install



Choose the installation directory (we suggest to keep the default one)

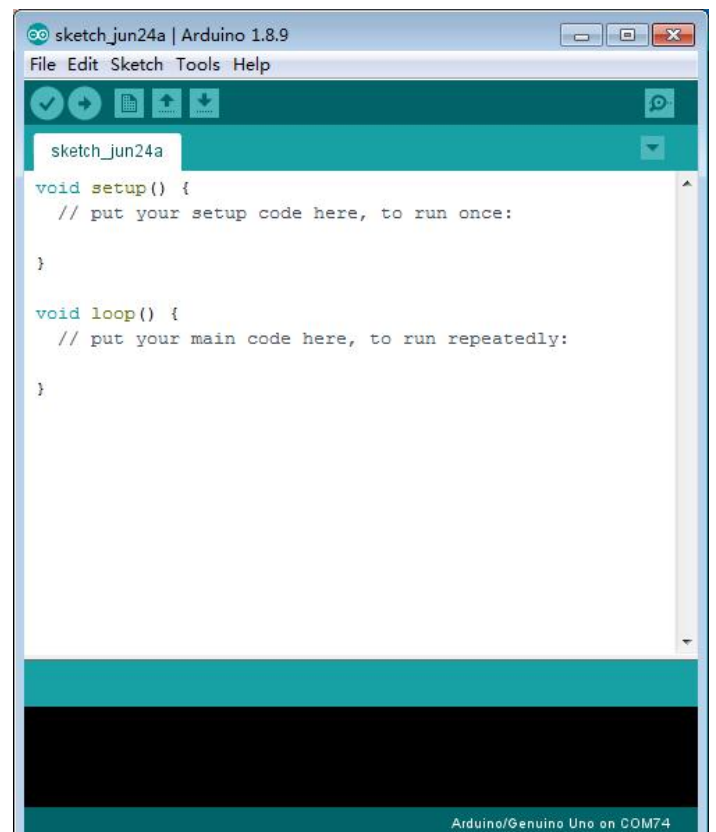
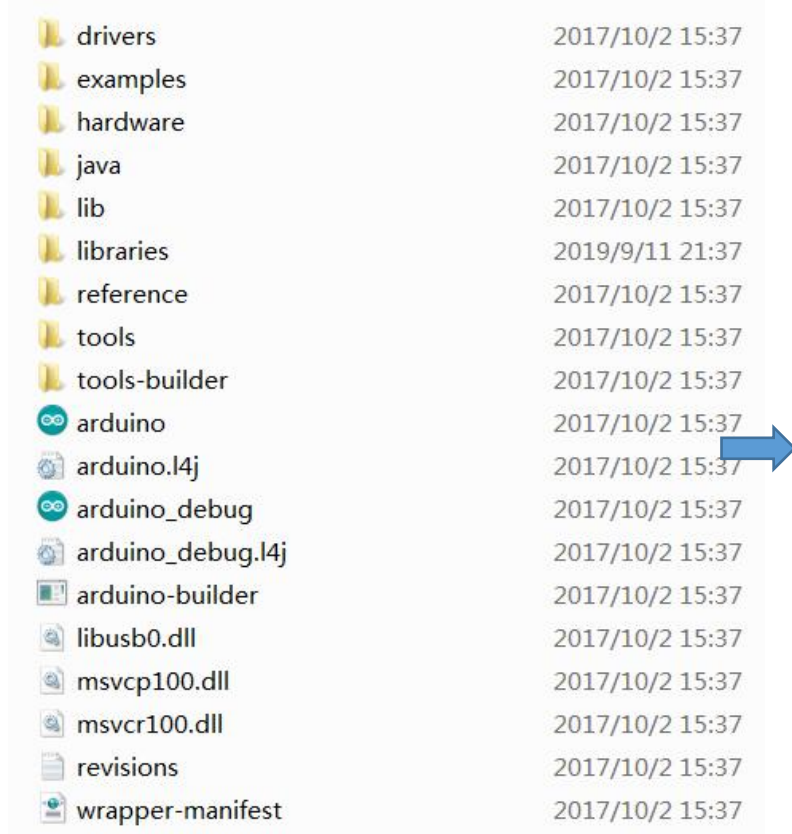


The process will extract and install all the required files to execute properly the Arduino Software (IDE)

Download the Zip package



Unzip the downloaded zip file, put it in a system disk, double-click to open the unzipped folder, and then click the arduino icon to open the IDE, as shown below:



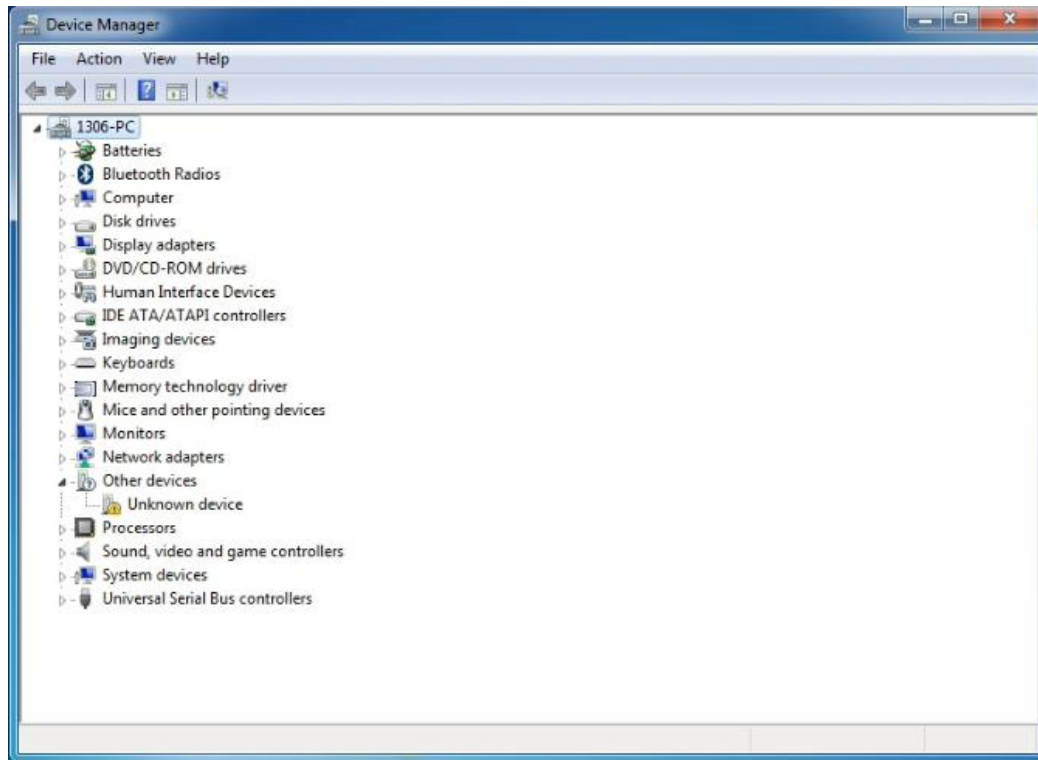
Install The Board Drivers

If you downloaded and expanded the Zip package or, for some reason, the board wasn't properly recognized, please follow the procedure below.

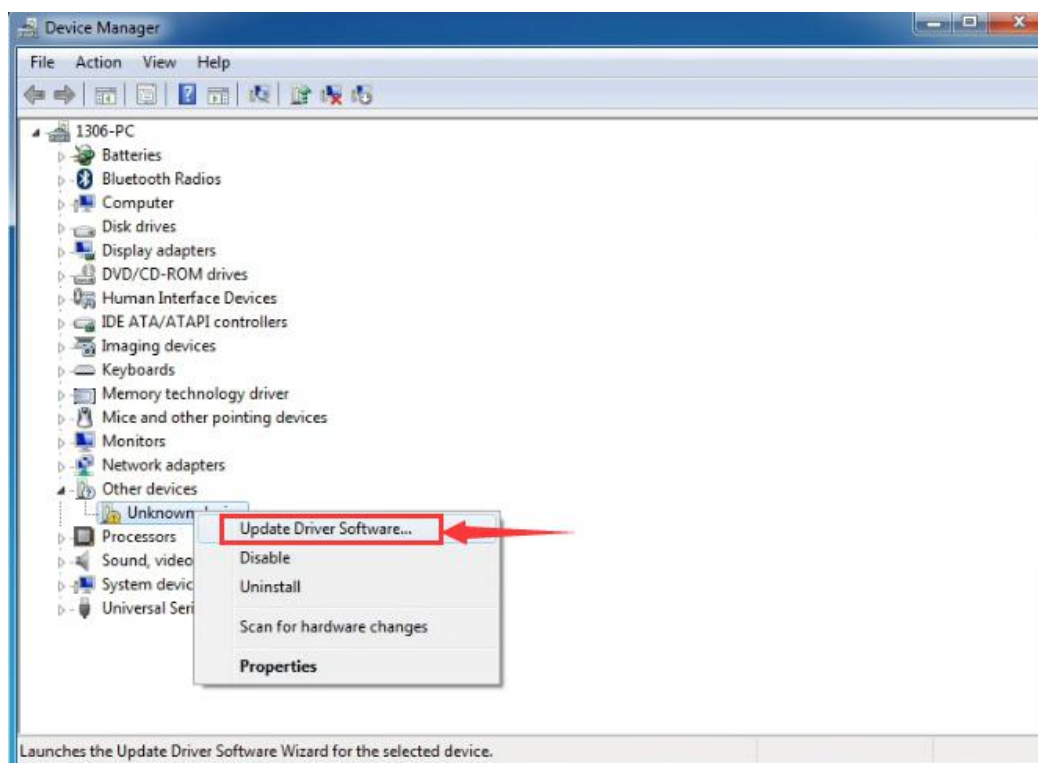
Click on the Start Menu, and open up the Control Panel.

While in the Control Panel, navigate to System and Security. Next, click on System. Once the System window is up, open the Device Manager.

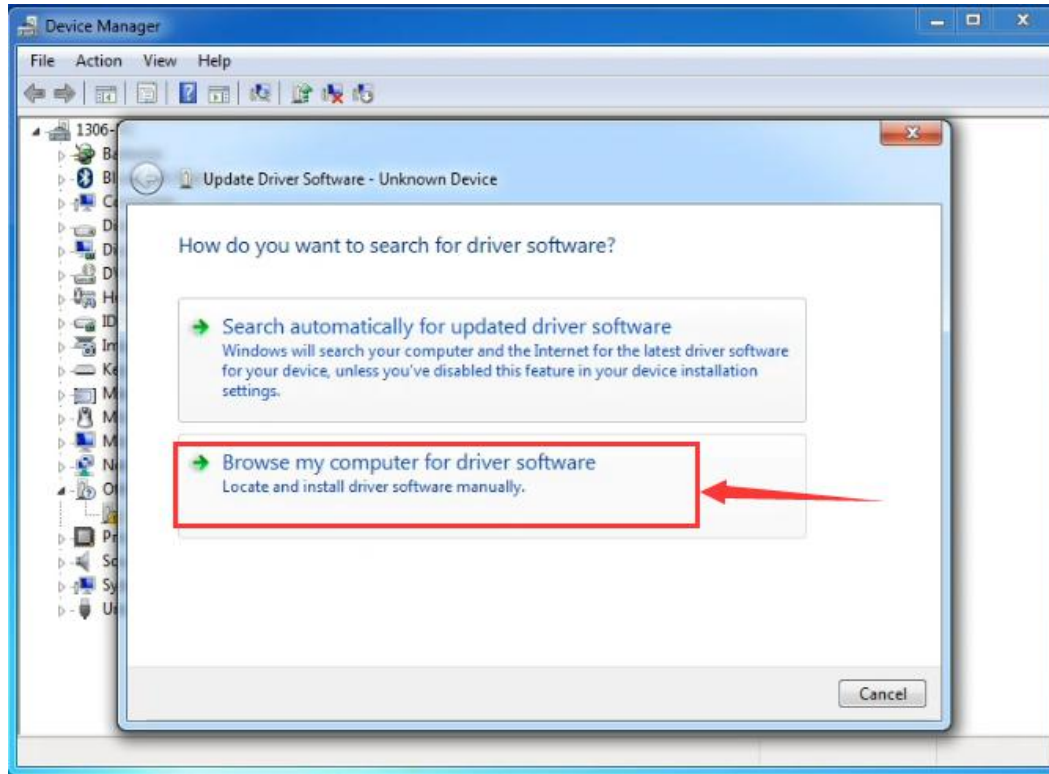
Look under Ports (COM & LPT). You should see an open port named "Arduino UNO (COMxx)". If there is no COM & LPT section, look under "Other Devices" for "Unknown Device".



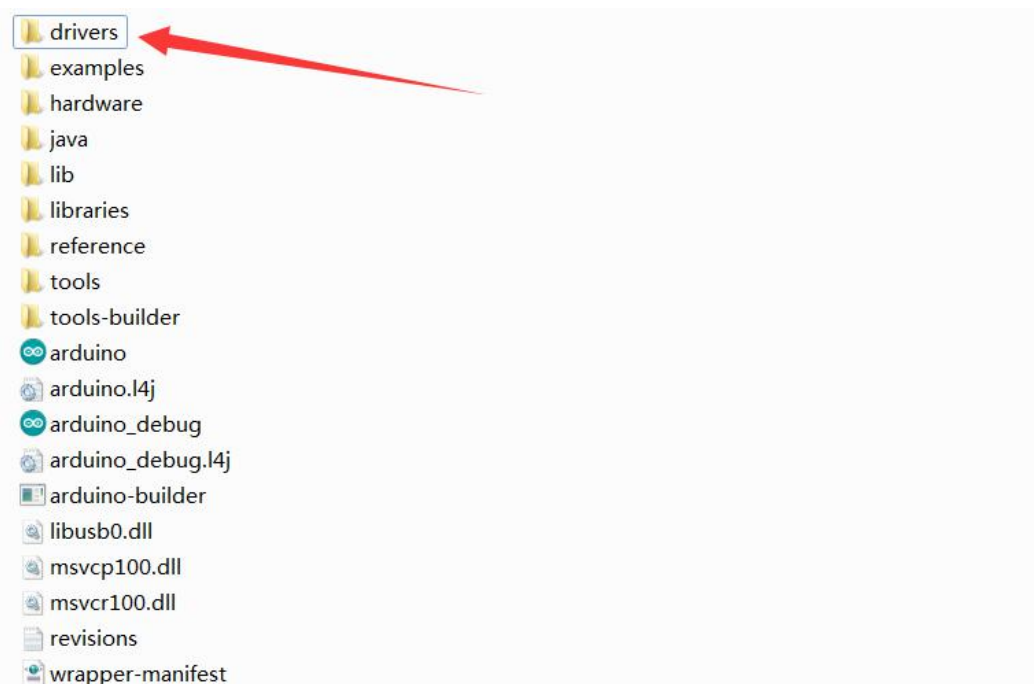
Right click on the "Arduino UNO (COMxx)" port and choose the "Update Driver Software" option.



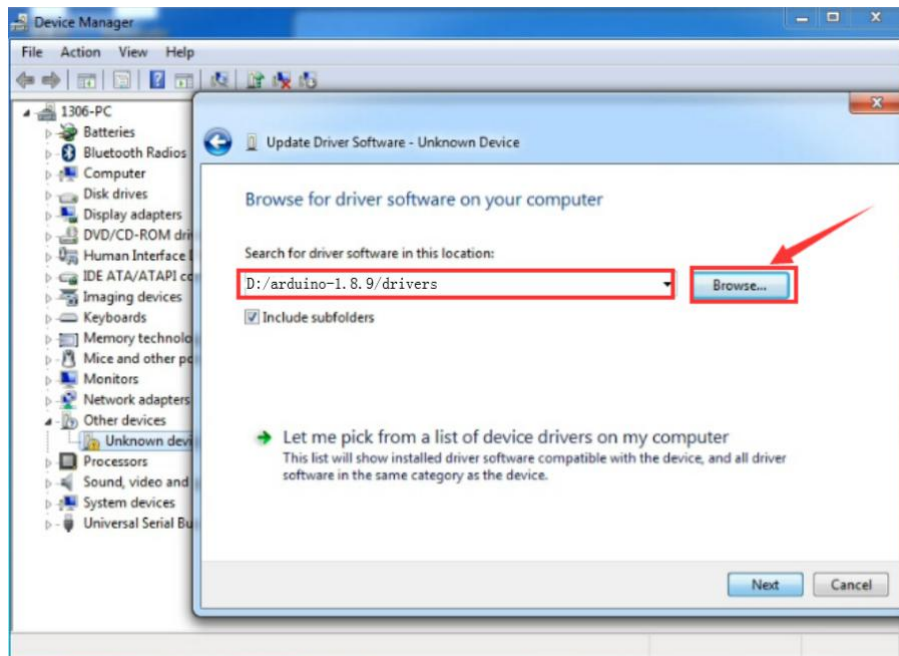
Next, choose the "Browse my computer for Driver software" option.



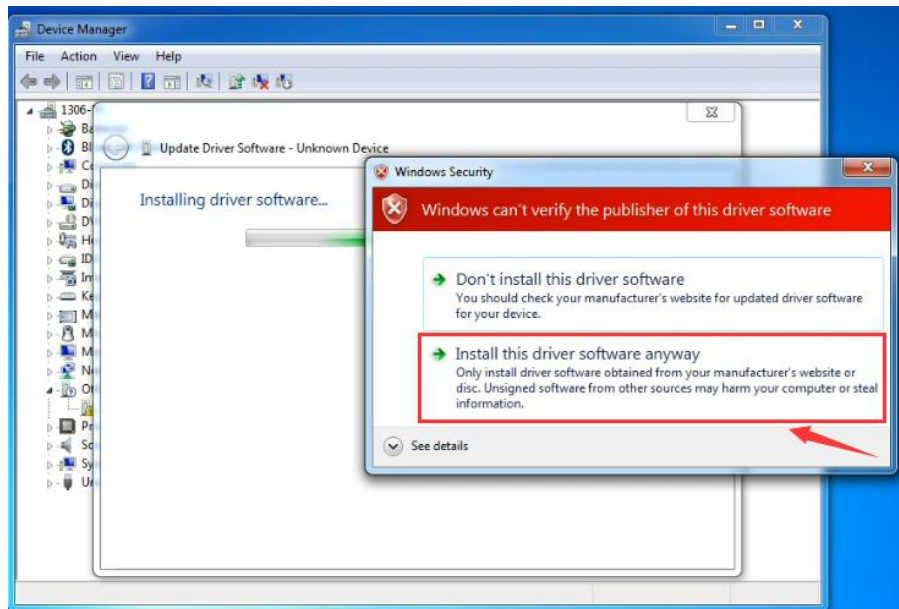
Finally, navigate to and select the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub-directory).



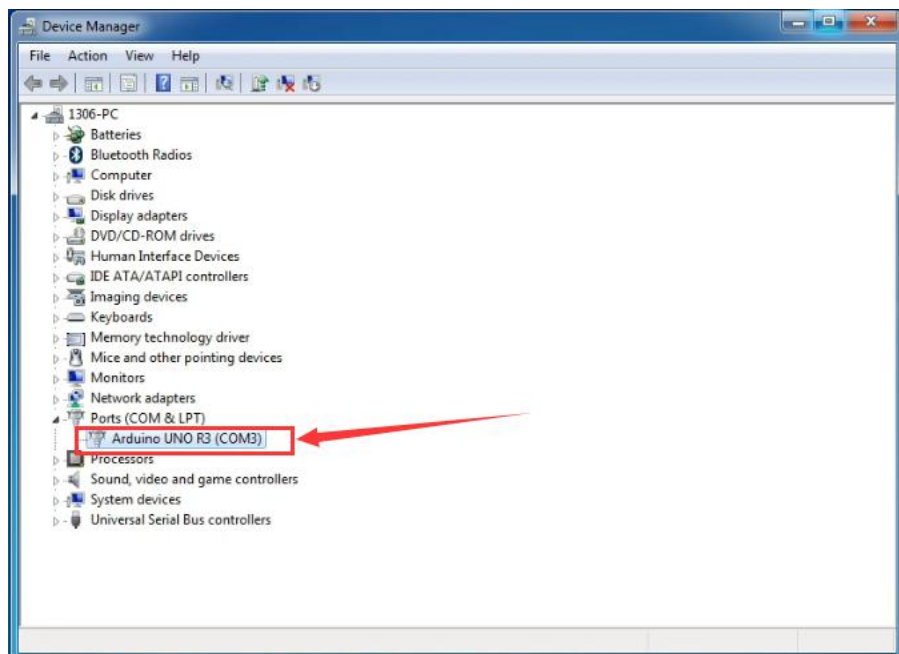
Click next



Install this driver software anyway









After the installation driver is complete, click Close.



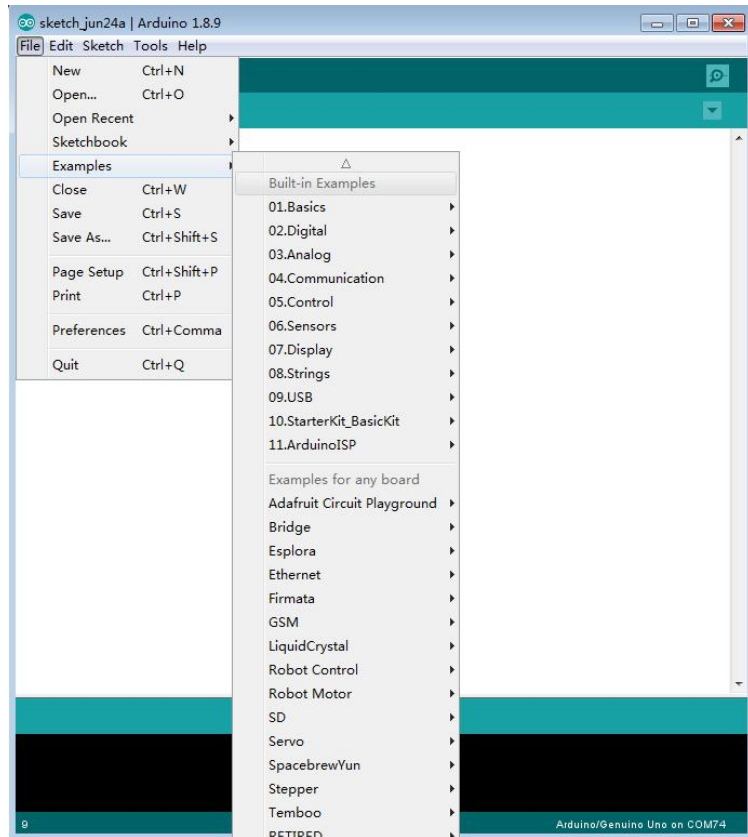
The functions of each button on the Toolbar are listed below:



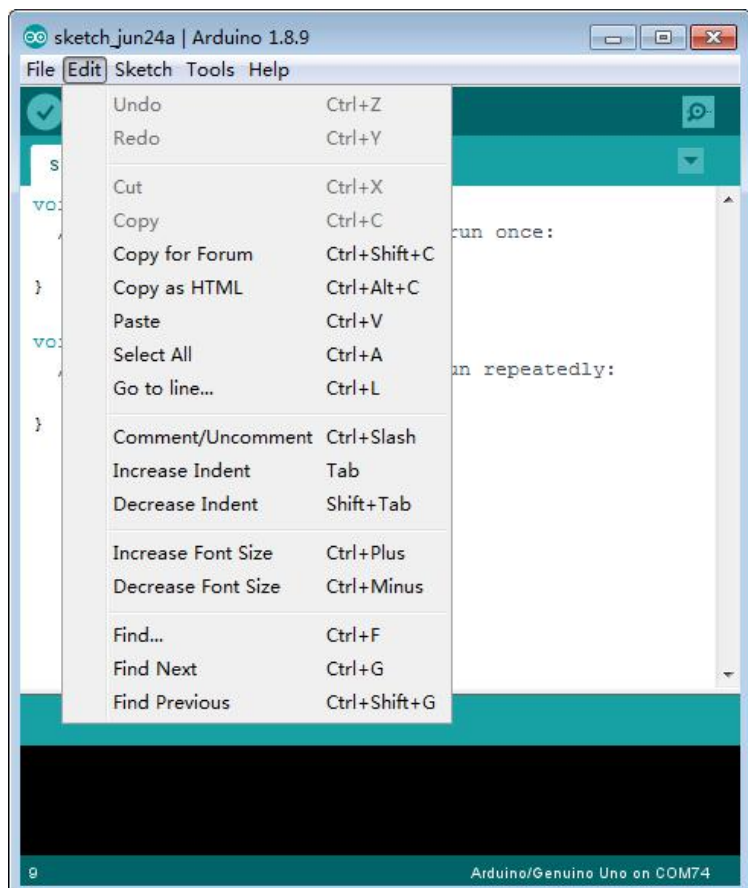
 Verify/Compile	Check the code for errors
 Upload	Upload the current Sketch to the Arduino
 New	Create a new blank Sketch
 Open	Show a list of Sketches
 Save	Save the current Sketch
 Serial Monitor	Display the serial data being sent from the Arduino

Introduction to the arduino IDE

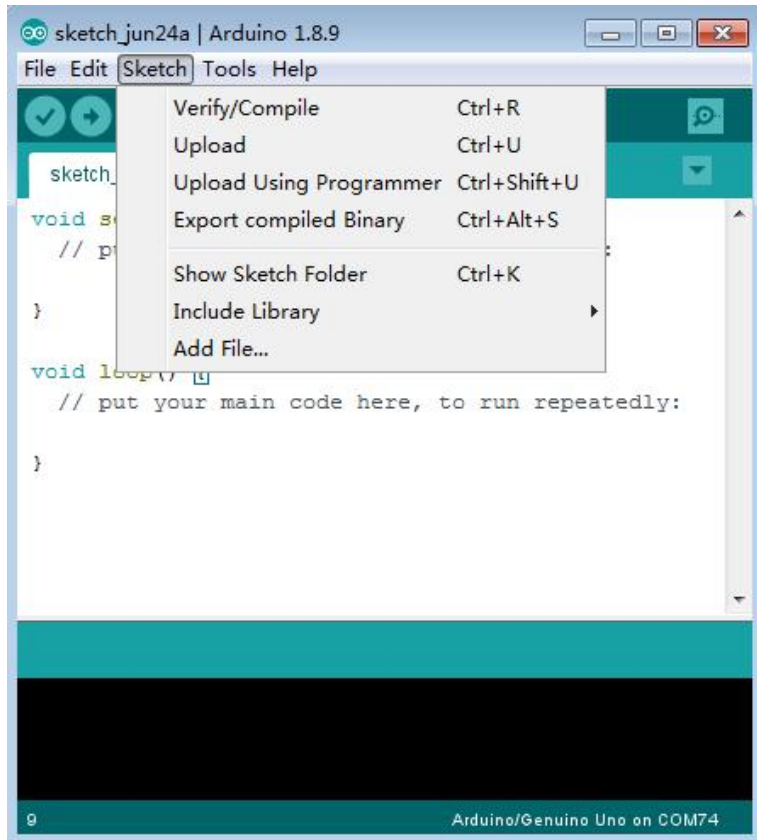
File: New sketch, open existing sketch, etc. There are many ready-made sample sketches under Examples, as shown below:



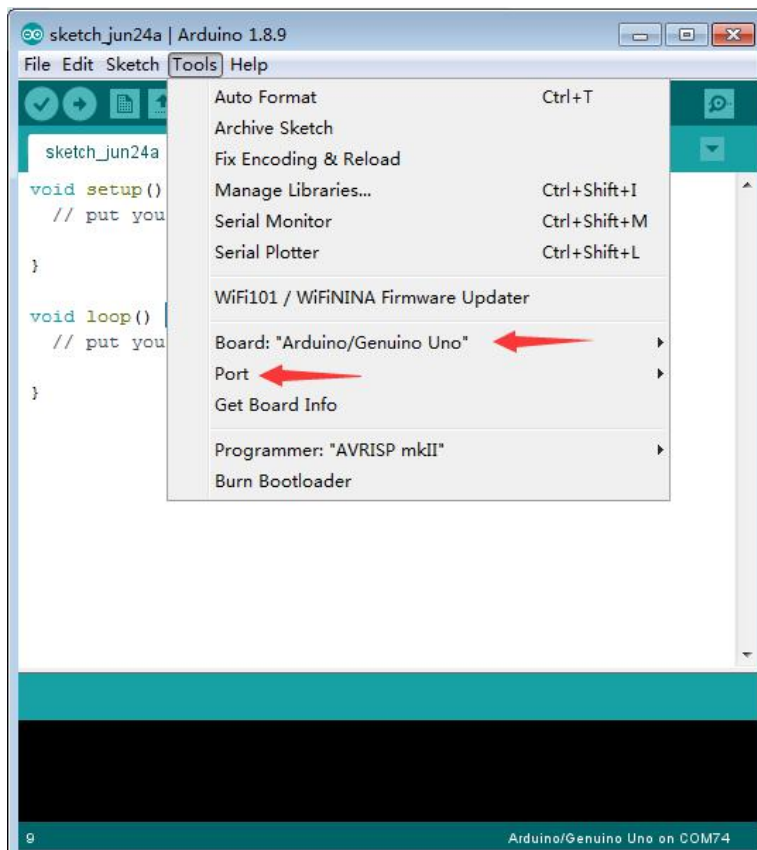
Edit: Undo and restore the sketch. When you write the wrong sketch, you want to restore the original sketch. This function is often used.



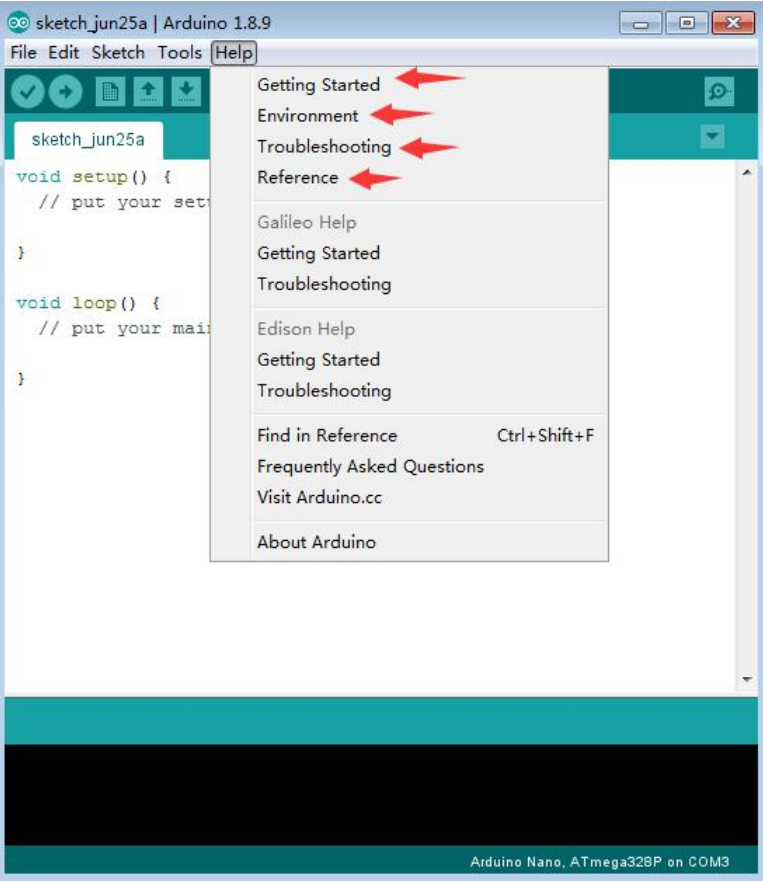
Sketch : compile and download the sketch



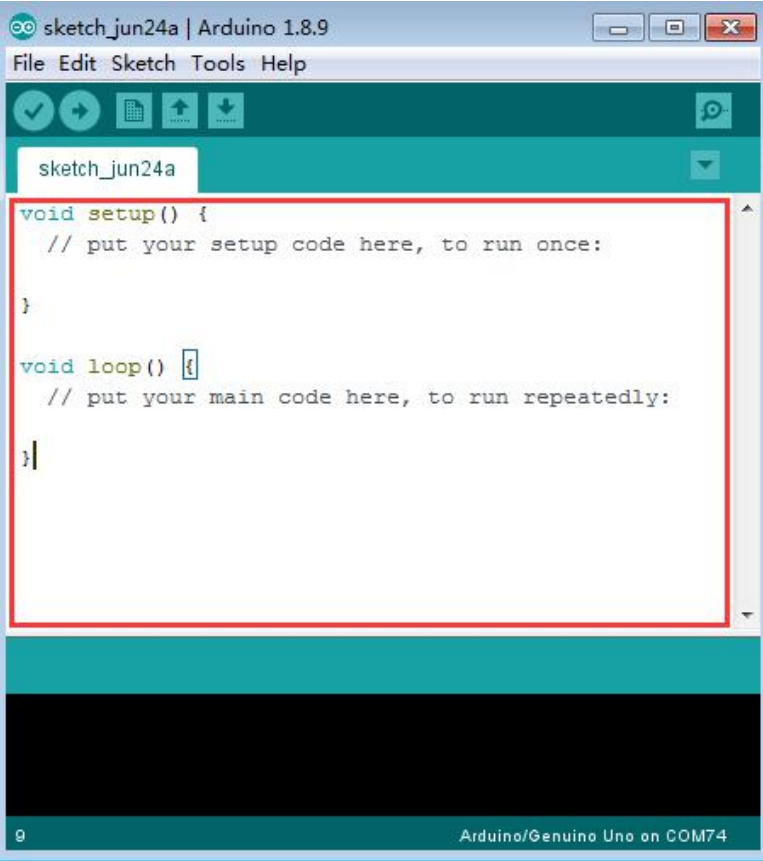
Tools : When uploading the program, you need to select the corresponding board type and port number, otherwise the program will not be burned into the board, as shown below:



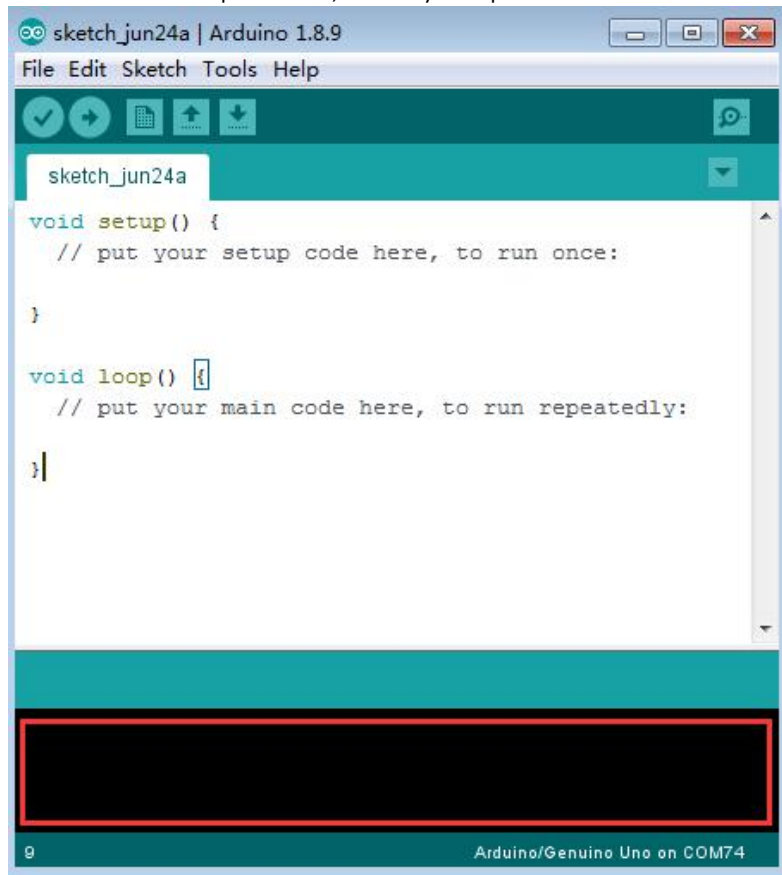
Help :About the official tutorial, IDE detailed introduction, common error resolution, programming language learning, etc., it is recommended that beginners must see, you will benefit a lot.



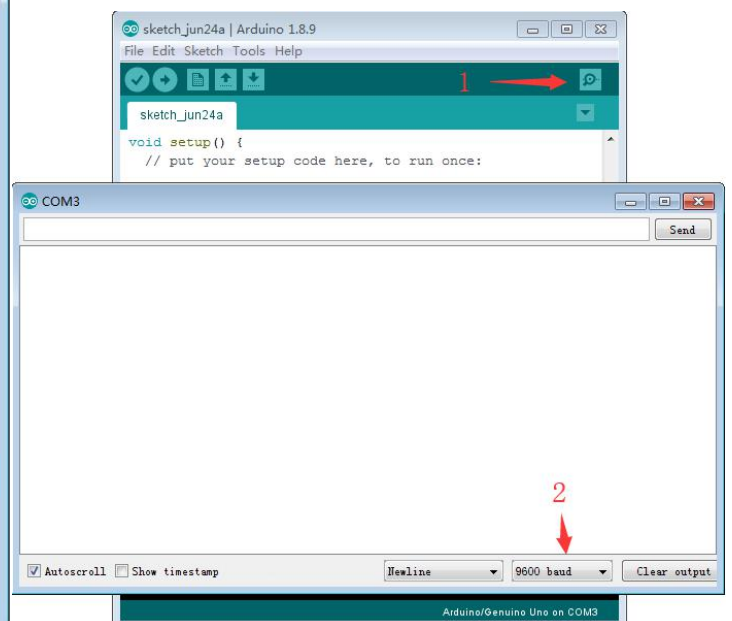
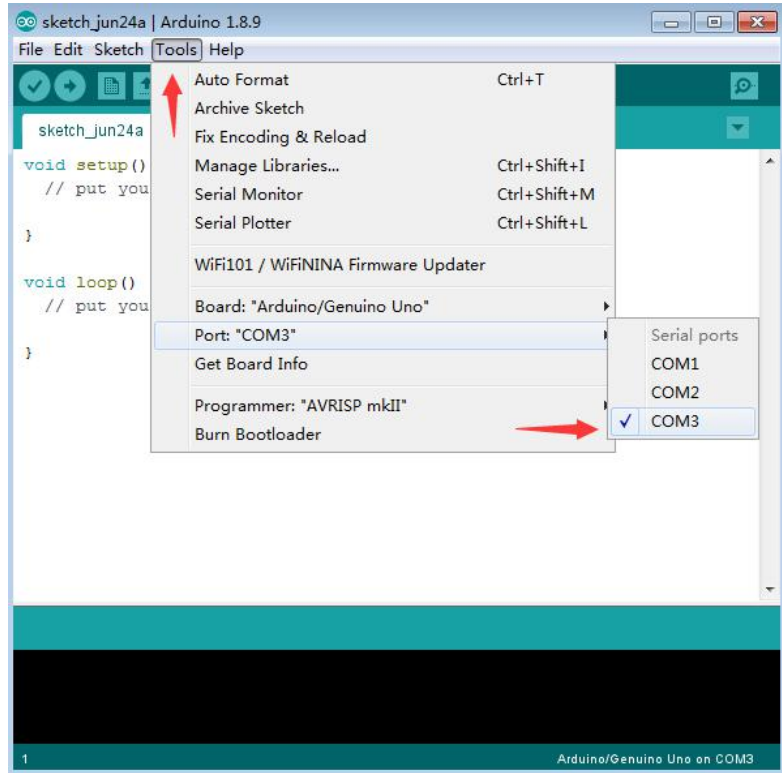
The sketch writing area, as shown below:
(Programming Learning Website: <https://www.arduino.cc/reference/en/>)



Information output area, mainly output information of compiling and uploading sketch.

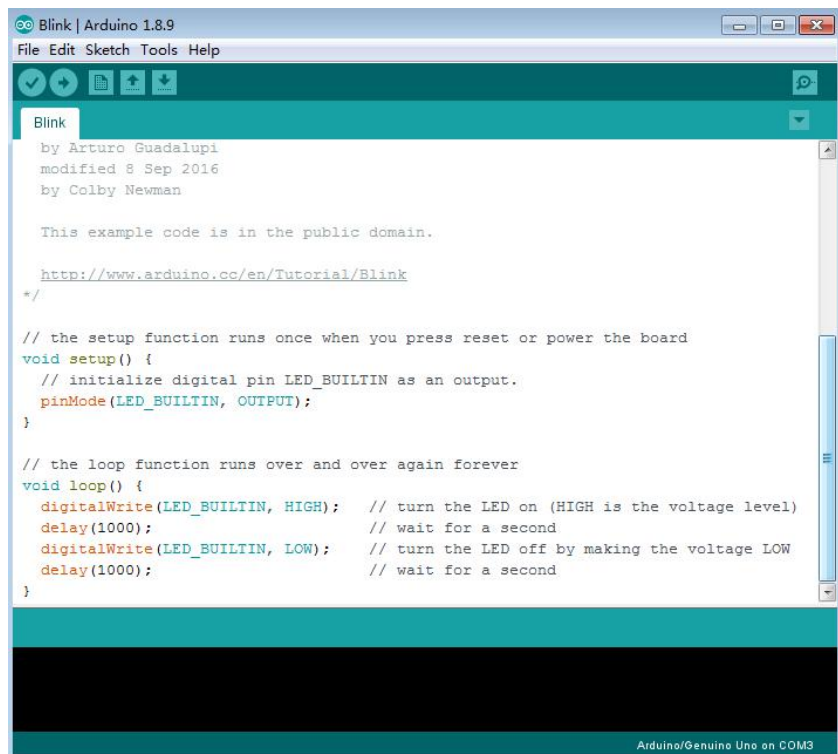
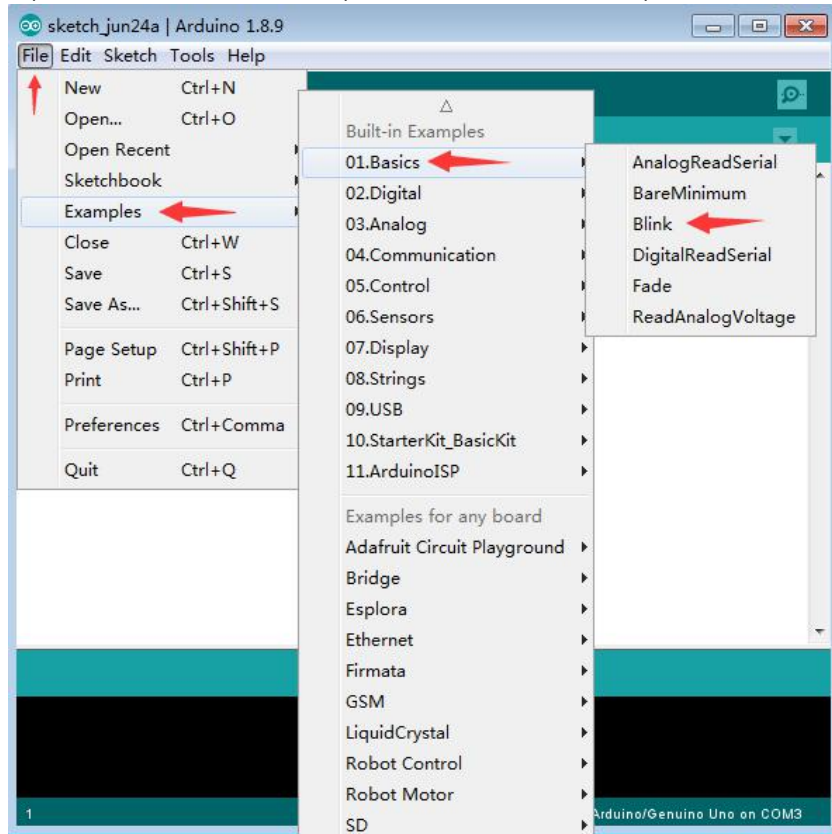


Serial monitor, you can open the serial monitor after selecting the port corresponding to the board in "tools"--->"Port". The UNO board selects the COM3 port and sets the baud rate to 9600.



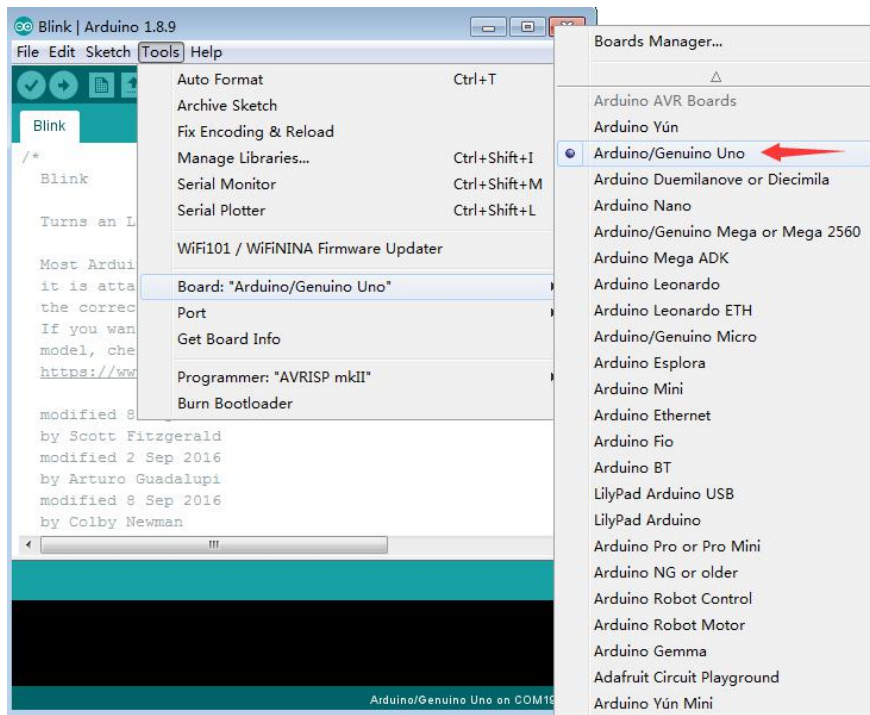
Open your first sketch

Open the LED blink example sketch: File > Examples > 01.Basics > Blink.

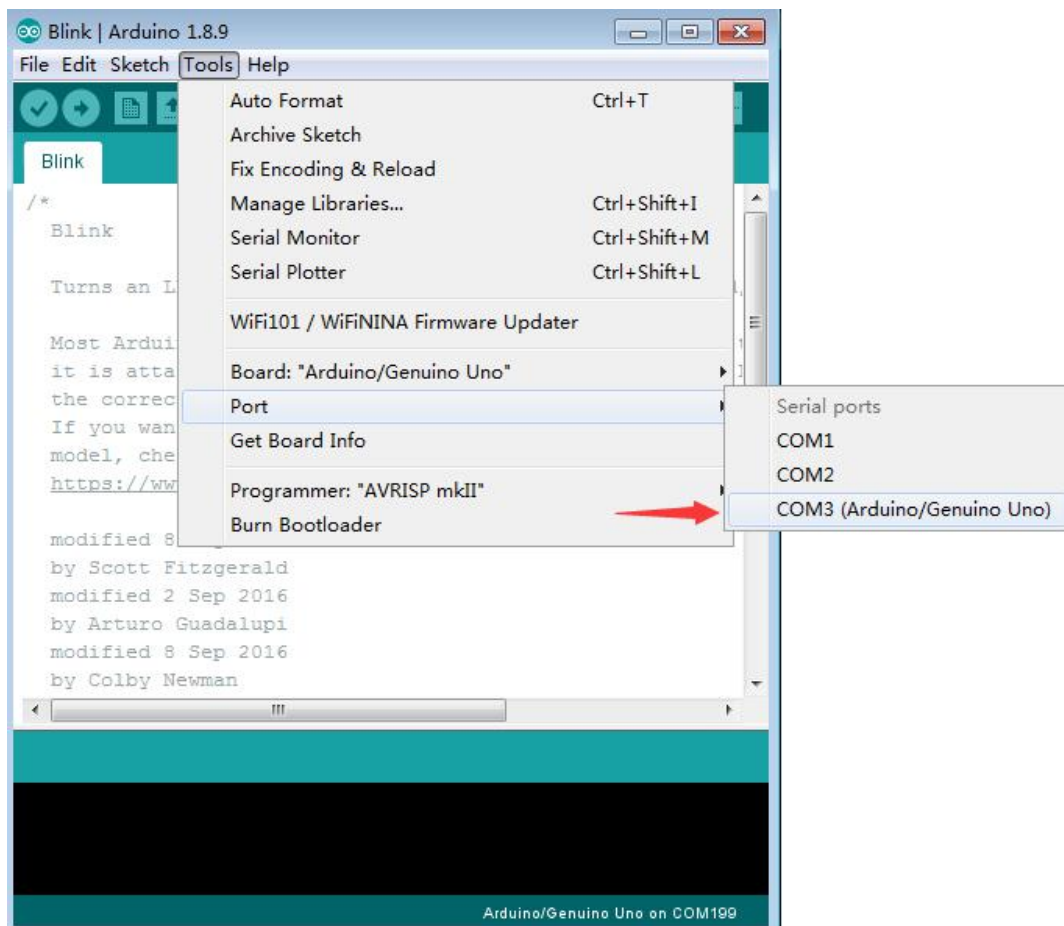


Select your board type and port

You'll need to select the entry in the Tools > Board menu that corresponds to your Arduino or Genuino board.

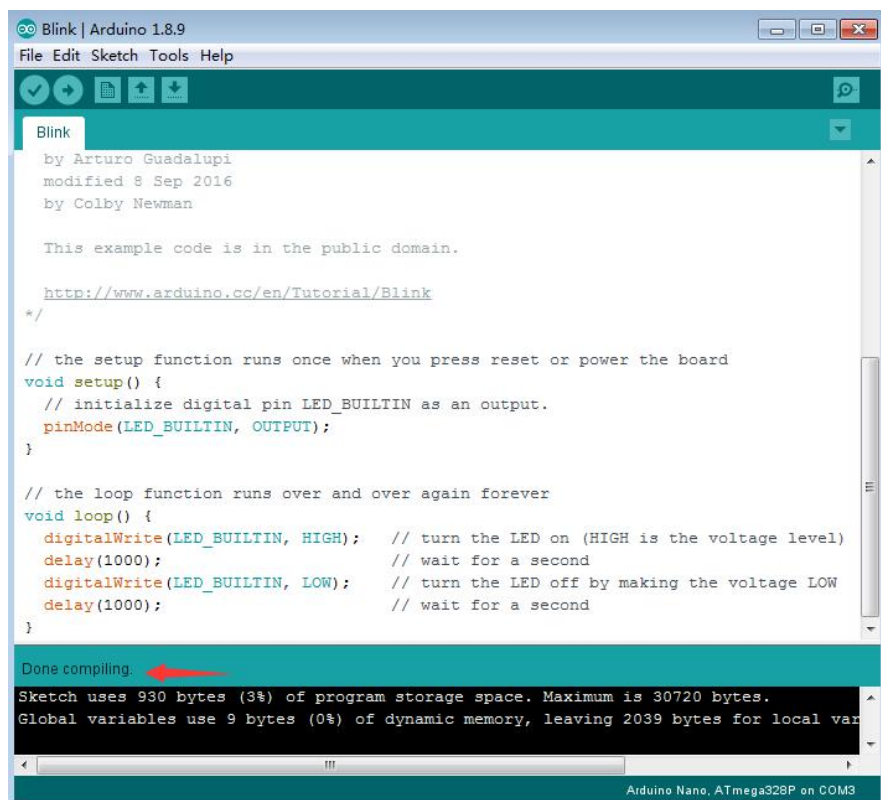


Select the serial device of the board from the Tools | Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your board and re-open the menu; the entry that disappears should be the Arduino or Genuino board. Reconnect the board and select that serial port.



Upload the sketch

Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.



```
Blink | Arduino 1.8.9
File Edit Sketch Tools Help

Blink
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/

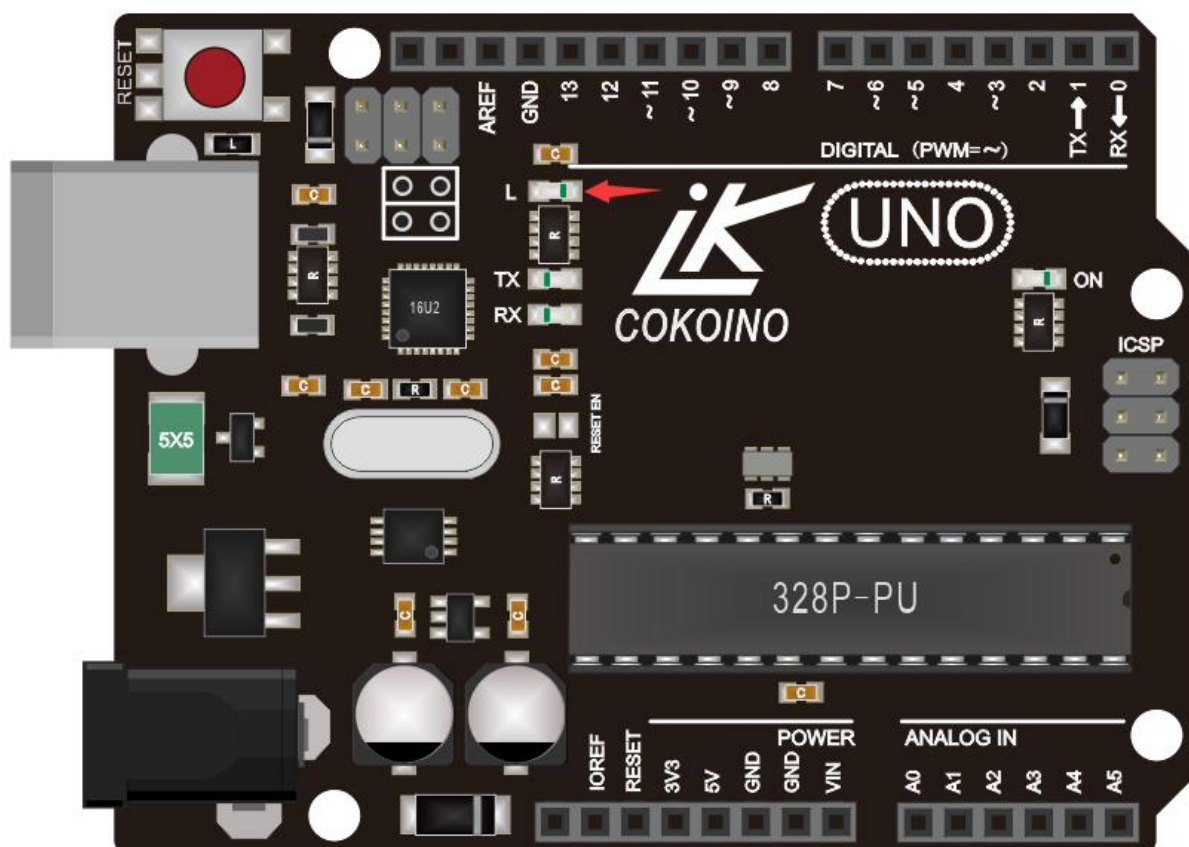
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

Done compiling.
Sketch uses 930 bytes (3%) of program storage space. Maximum is 30720 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables.

Arduino Nano, ATmega328P on COM3
```

A few seconds after the upload finishes, you should see the pin 13 (L) LED on the board start to blink (in orange). If it does, congratulations! You've gotten your uno up-and-running.



arduino library file

What is the arduino library file

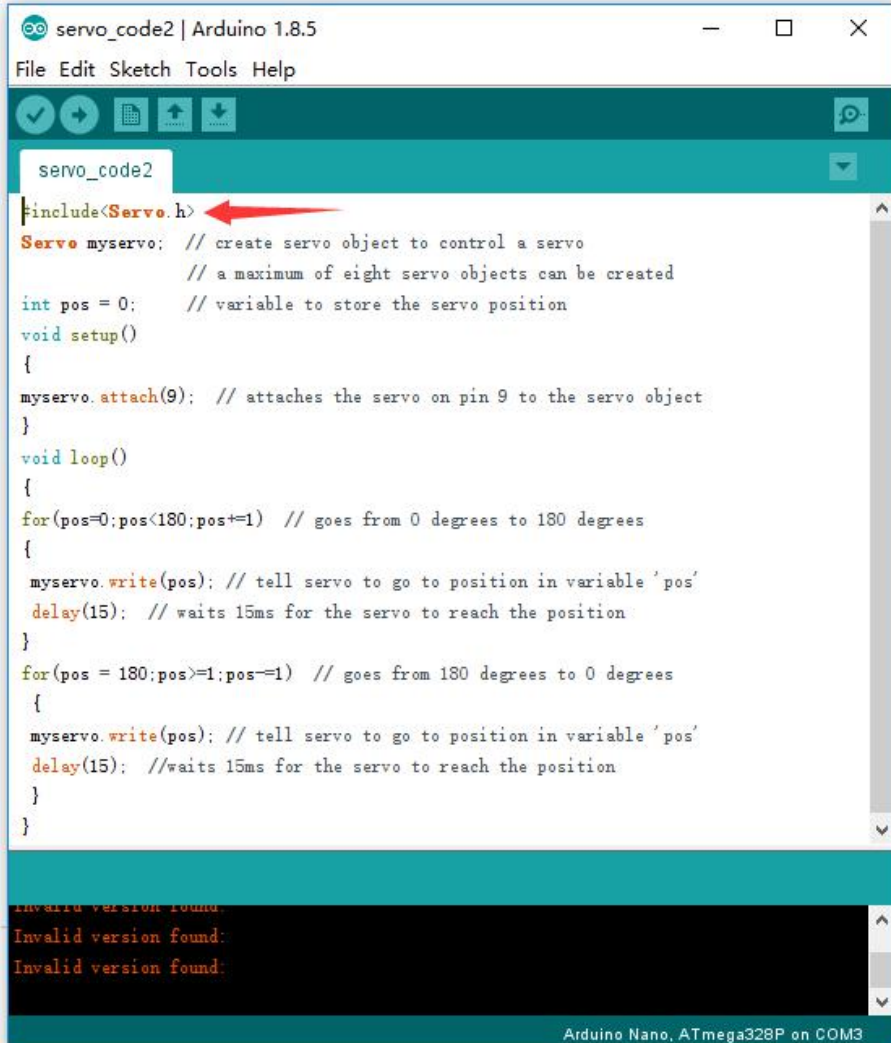
Libraries are files written in C or C++ (.c, .cpp) which provide your sketches with extra functionality (e.g. the ability to control an LED matrix, or read an encoder, etc.).

To use an existing library in a sketch simply go to the Sketch menu, choose "Import Library", and pick from the libraries available. This will insert an `#include` statement at the top of the sketch for each header (.h) file in the library's folder. These statements make the public functions and constants defined by the library available to your sketch. They also signal the Arduino environment to link that library's code with your sketch when it is compiled or uploaded.

To install your own library, create a folder inside `ARDUINO/hardware/libraries` with the name of your library. The folder should contain a C or C++ file with your code and a header file with your function and variable declarations. It will then appear in the Sketch | Import Library menu in the Arduino IDE.

Because libraries are uploaded to the board with your sketch, they increase the amount of space used by the ATmega8 on the board. See the FAQ for an explanation of various memory limitations and tips on reducing program size. If a sketch no longer needs a library, simply delete its `#include` statements from the top of your code. This will stop the Arduino IDE from linking the library with your sketch and decrease the amount of space used on the Arduino board.

The following figure shows the library function of the servo motor:



```
servo_code2 | Arduino 1.8.5
File Edit Sketch Tools Help

servo_code2
#include<Servo.h>
Servo myservo; // create servo object to control a servo
               // a maximum of eight servo objects can be created
int pos = 0;   // variable to store the servo position
void setup()
{
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}
void loop()
{
  for(pos=0;pos<180;pos+=1) // goes from 0 degrees to 180 degrees
  {
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(15); // waits 15ms for the servo to reach the position
  }
  for(pos = 180;pos>=1;pos-=1) // goes from 180 degrees to 0 degrees
  {
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(15); //waits 15ms for the servo to reach the position
  }
}

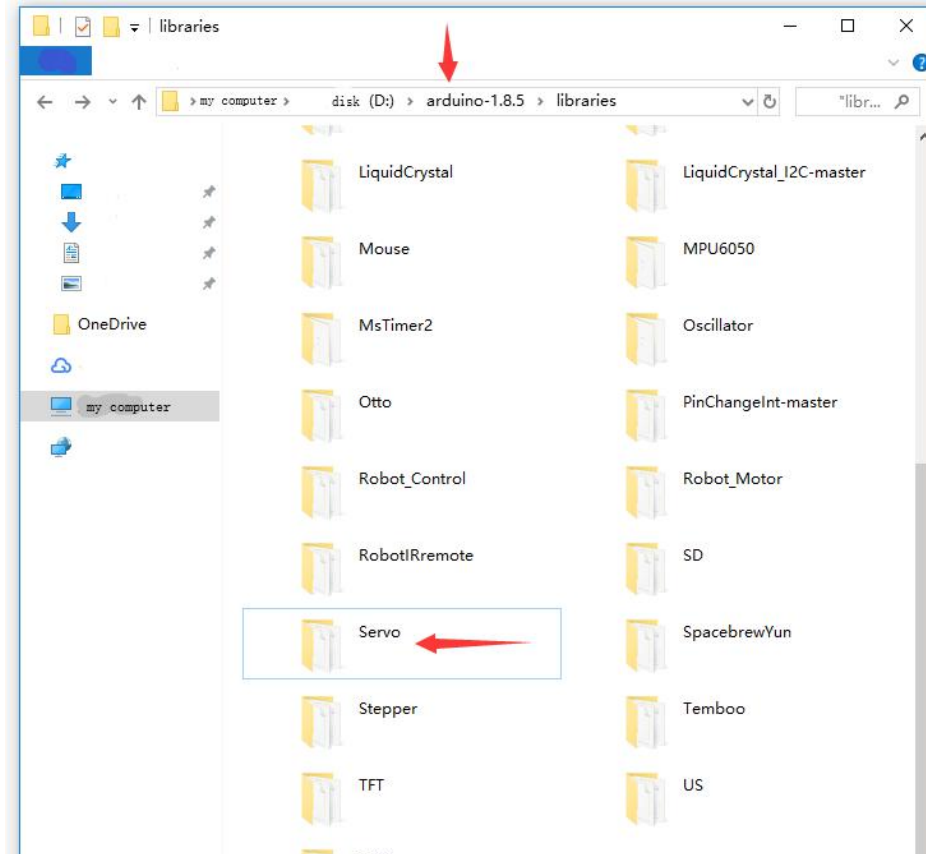
Invalid version found:
Invalid version found:
Invalid version found:

Arduino Nano, ATmega328P on COM3
```

Where is the library file placed?

The library file is a combination of some header files ending in ".h" and source files ending in ".c" in a folder.

For information on how to write arduino library files, please learn about it in the Google or arduino community. We only need to put the existing or self-written library files in the libraries folder of the Arduino IDE installation package, then the arduino ide will run it, otherwise arduino ide will report an error when compiling the program. The following figure shows the placement path of my computer arduino library files:



Remarks: For more information on arduino, please refer to the official website below.

<https://www.arduino.cc/en/Reference/Libraries>