

Building the Arduino Development Environment

Arduino development language

Arduino uses C/C++ to write programs, so before learning Arduino, you need to master the C/C++ language. Although C++ is compatible with the C language, these are two different languages. C is a process-oriented programming language, and C++ is an object-oriented programming language. The early Arduino core library was written in C language. Later, object-oriented ideas were introduced. At present, the latest Arduino core library is written in C and C++.

Generally speaking, the Arduino language refers to a collection of various Application Programming Interfaces (APIs) provided by the Arduino core library files. These APIs are formed by secondary packaging of the lower-level microcontroller support library. For example, the core library of Arduino using AVR microcontroller is the secondary packaging of AVR-Libc (GCC-based AVR support library).

In the traditional development method, multiple registers need to be configured to achieve the corresponding functions. In Arduino, the complicated registers are encapsulated into simple APIs, which can be intuitively controlled, enhancing the readability of the program and improving the development efficiency.

Arduino program structure

The Arduino program structure is different from the traditional C/C++ program structure-there is no `main()` function in the Arduino program. In fact, it is not that there is no

main() function in the Arduino program, but that the definition of the main() function is hidden in the core library file of the Arduino. In the development of Arduino, the main function is not directly operated, but the two functions of setup() and loop() are used instead.

The construction of the Arduino development environment

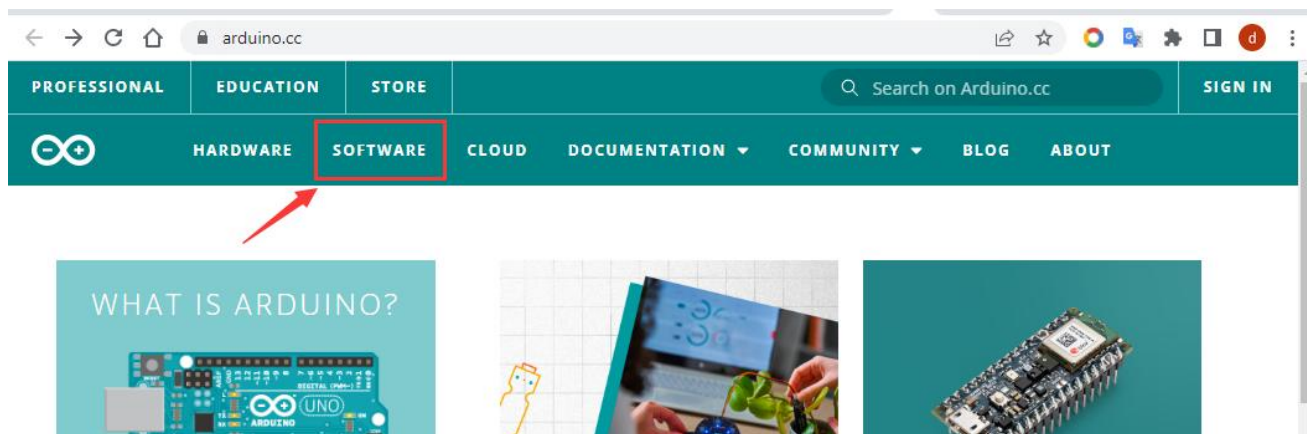
The IDE of the Arduino development environment can be downloaded from the official website. The download address of the Arduino IDE is: <https://www.arduino.cc/>

(1) Install Arduino IDE under Windows

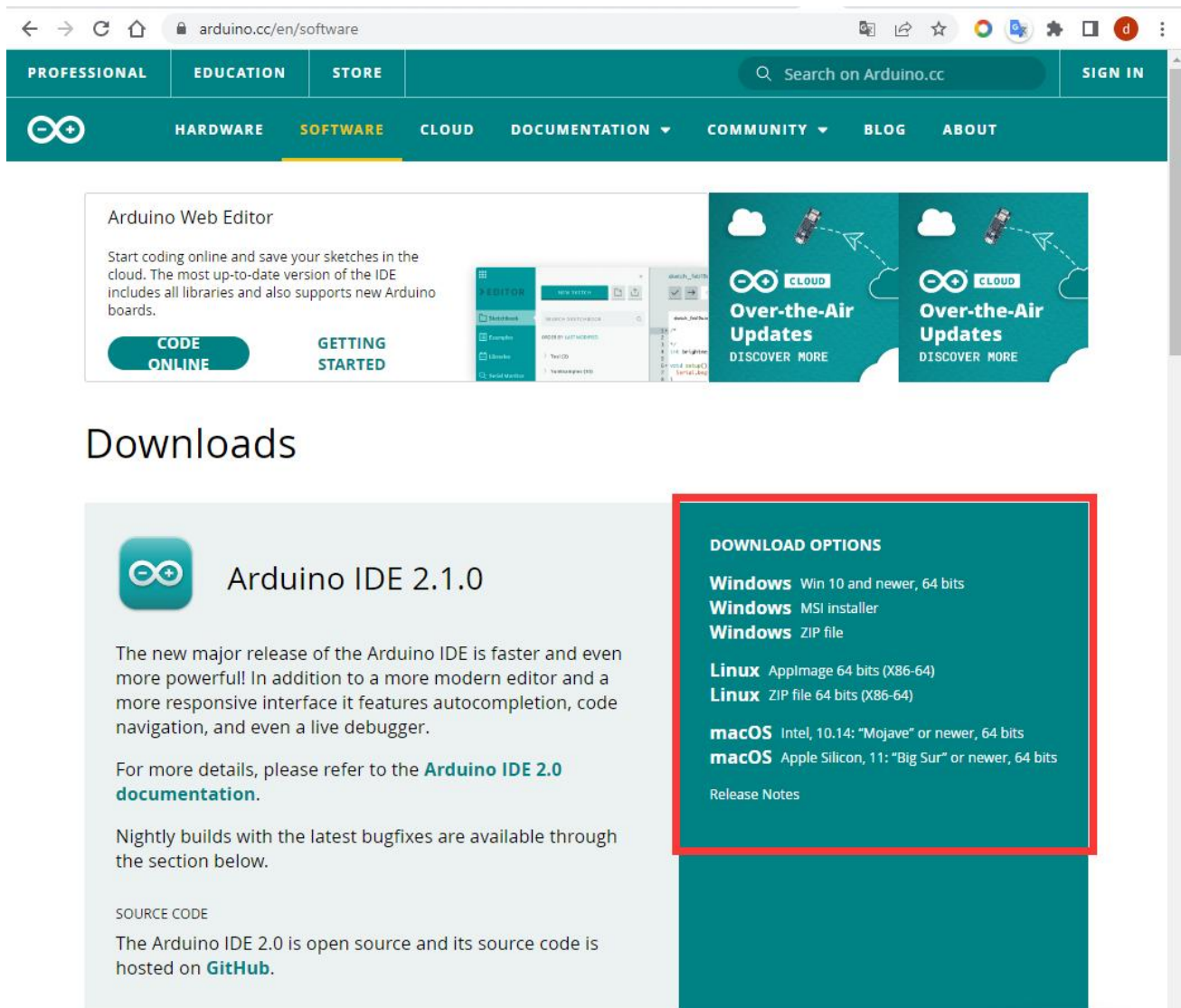
We will teach you how to download and install:

1. Open Google Chrome and enter the URL in the address bar: <https://www.arduino.cc/>

After successfully opening the interface as shown below, we click SOFTWARE.



2. After jumping to the following interface, slide the mouse to the middle to find the part marked in the red circle. You can find that the official website provides us with installation files for Windows, Mac OS X, and Linux systems.



Arduino Web Editor

Start coding online and save your sketches in the cloud. The most up-to-date version of the IDE includes all libraries and also supports new Arduino boards.

[CODE ONLINE](#) [GETTING STARTED](#)

Downloads

Arduino IDE 2.1.0

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file


Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.14: "Mojave" or newer, 64 bits
macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

You can also download older 1.8.x versions of Arduino.

Legacy IDE (1.8.X)



Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.


Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file

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

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

Mac OS X 10.10 or newer

[Release Notes](#)

[Checksums \(sha512\)](#)

3. The tutorial uses Windows as an example. The usage steps of MAC and Linux are similar to Windows. Select the appropriate file for your computer to download. After the interface jumps, we select **JUST DOWNLOAD**. And then start the download. The download status will be displayed in the lower left of Google Chrome. Then we wait for the download to complete.



Arduino IDE 2.1.1

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

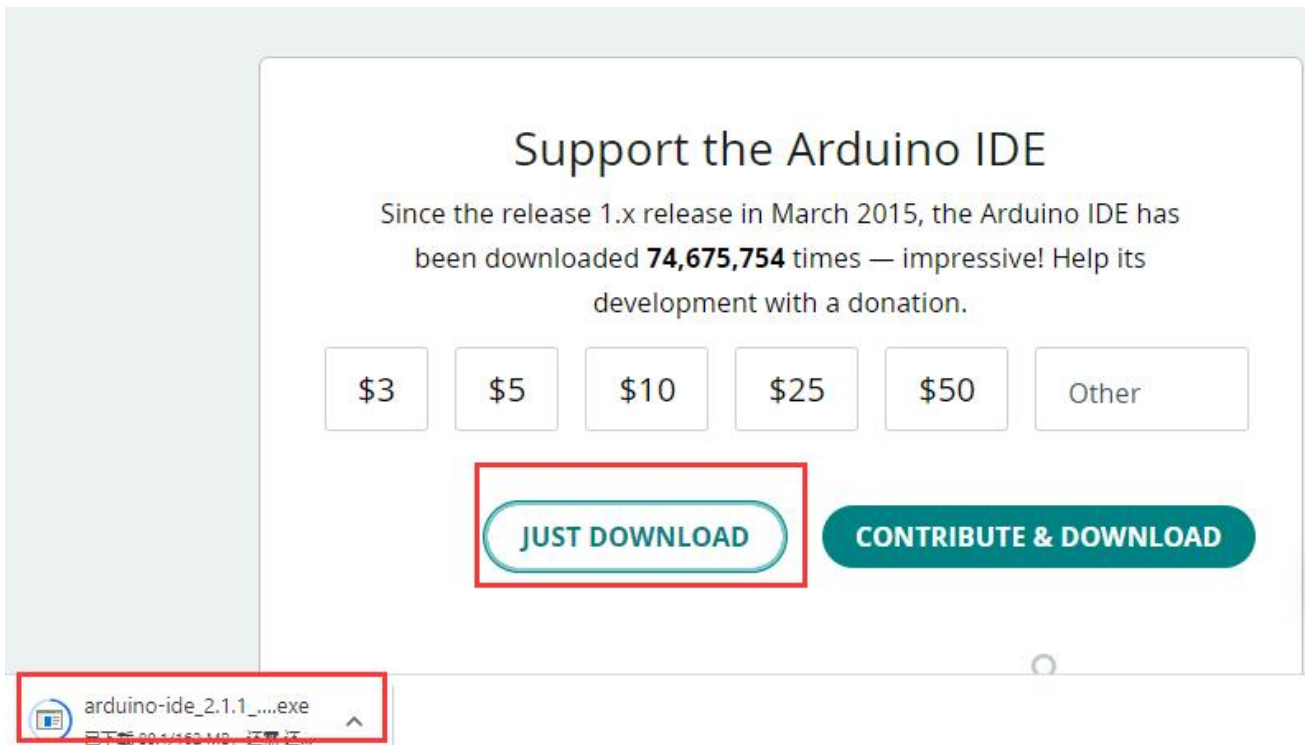
DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file

Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)

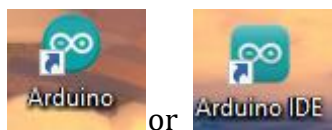
macOS Intel, 10.14: "Mojave" or newer, 64 bits
macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

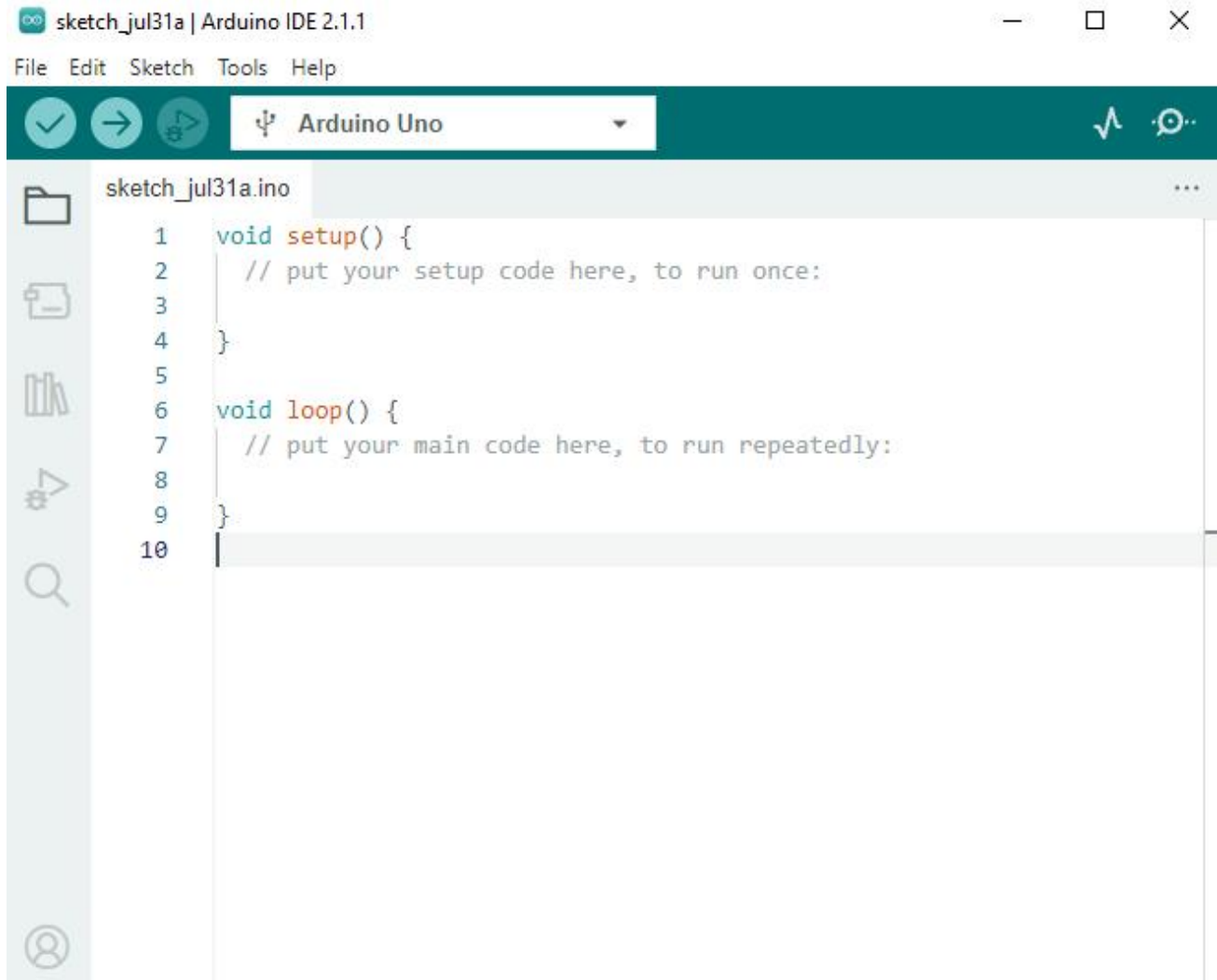


4. After the download completes, run the installer. For Windows users, there may pop up an installation dialog during the installation. When it pops up, please allow the installation.

5. After installation is completed, an Arduino Software shortcut will be generated in the desktop. Run the Arduino software.

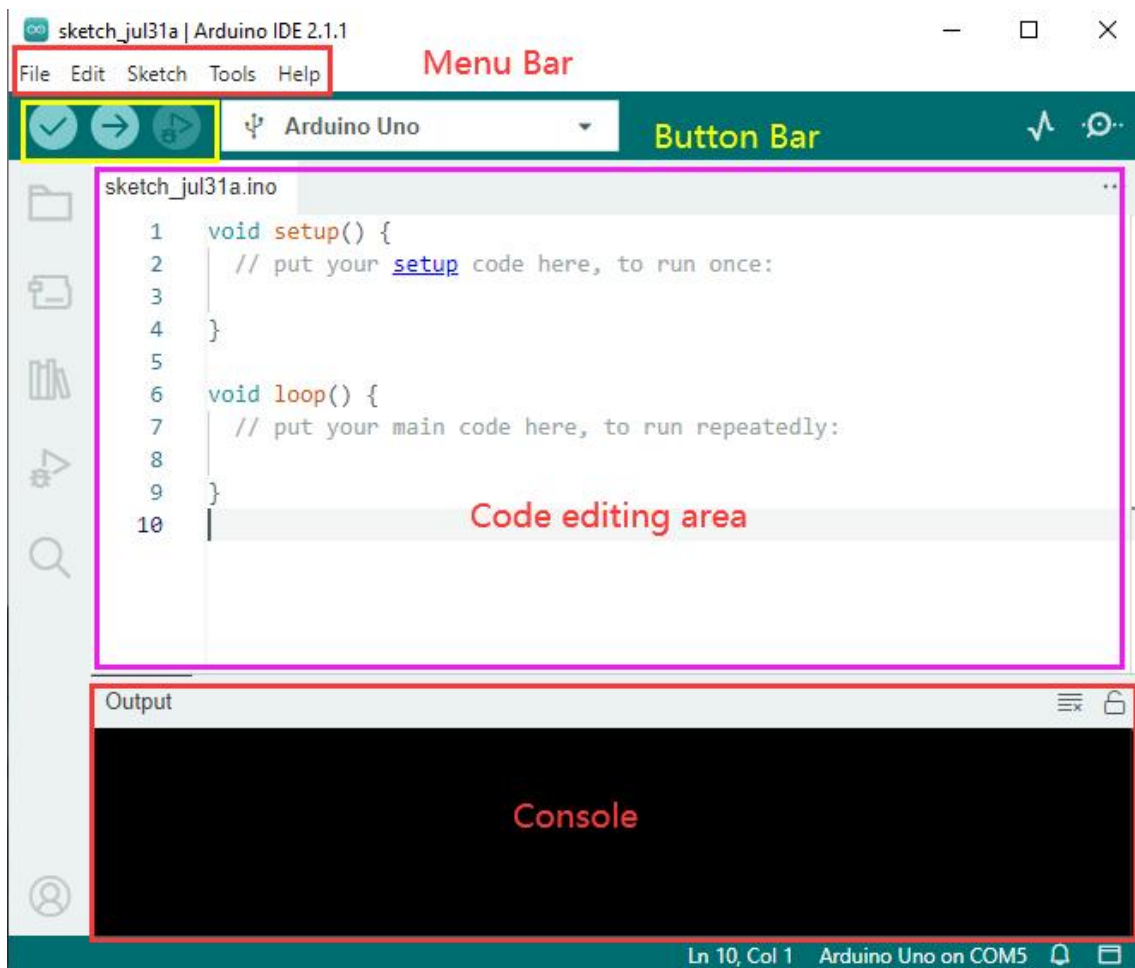


6. The interface will show as follows after the Arduino software is opened, indicating that our software has been downloaded and installed successfully.



Introduction of Arduino software interface

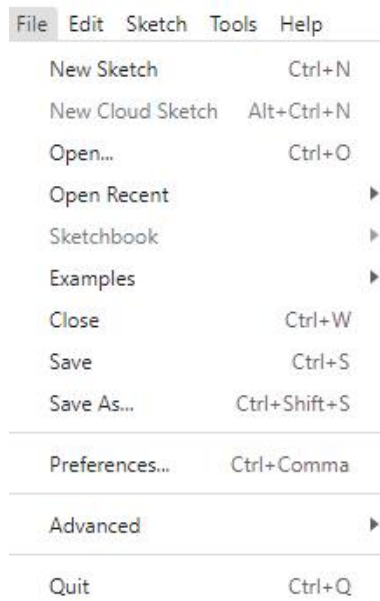
The following figure is the interface introduction of Arduino software



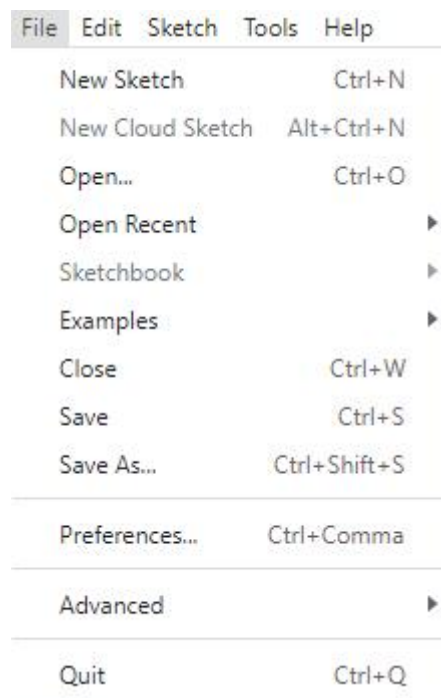
(1)Menu bar

Menu bar contains File, Edit, Sketch, Tools and Help.

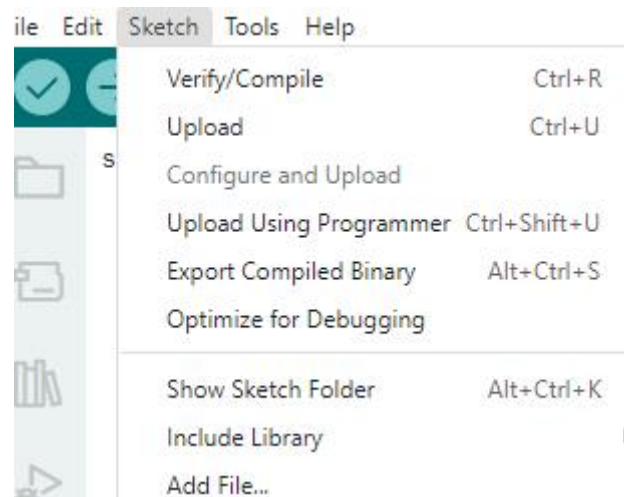
(1) "File" can operate new file, open file, save file, close file, save, etc. For the Examples, you can check the official sample program.



(2) "Edit" has the functions for the program code of editing, copying and pasting, commenting, indenting, searching, etc.

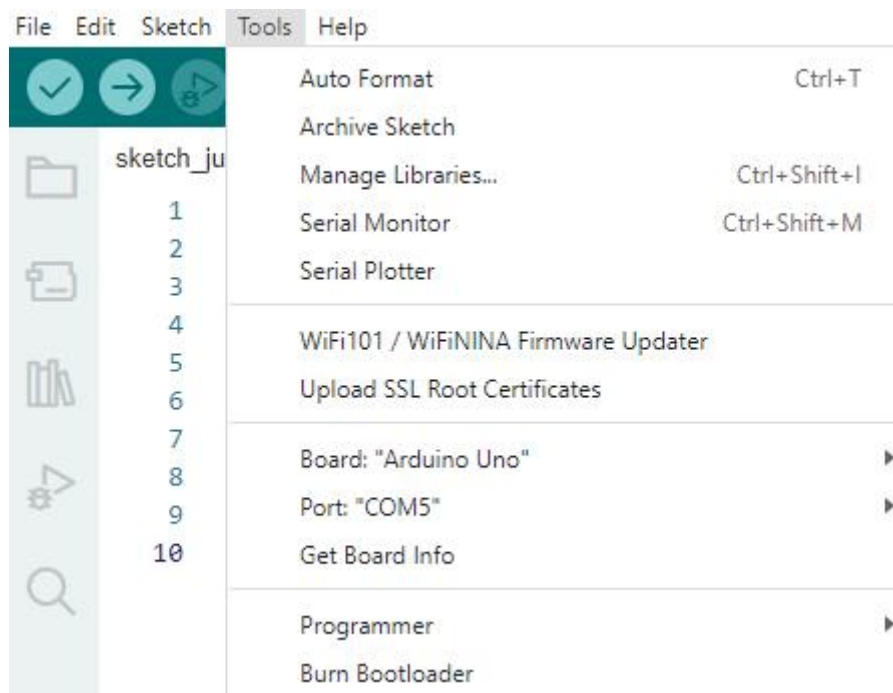


(3) Sketch can perform Verify/Compile, Upload and other operations on the written project.



The Include Library can load the library. After selecting the library file in the list, the relevant header files are automatically added in the code editing area.

(4) Board and Port are often used in "Tools".



Board can choose different development boards.

(2) Button bar

Button bar includes functions of Verifying, Uploading, Building New, Opening and Saving.

(1) Verify .

Checking and compilation. This button is used to check the correctness of your "syntax" or code. If your code has any syntax errors or undefined variables, an error message will appear at the bottom of the IDE screen. At the same time, the line of error code will be marked with a red background color for easy modification. But if it is correct, you will see the message that the compilation is complete.

(2) Upload .

Download the program code to the Arduino development board. It is better to click Verify first, and then click Upload.

(3) Code editing area

The code editing area is where to write program code and code comments.

(4) Console

The debug window will output information showing various compilation and debugging results. For example, if your code is written incorrectly, you will be prompted about what went wrong.

Install CH340 driver

Connecting the Arduino and the computer

You need to use USB Cable to connect the Arduino to the computer.

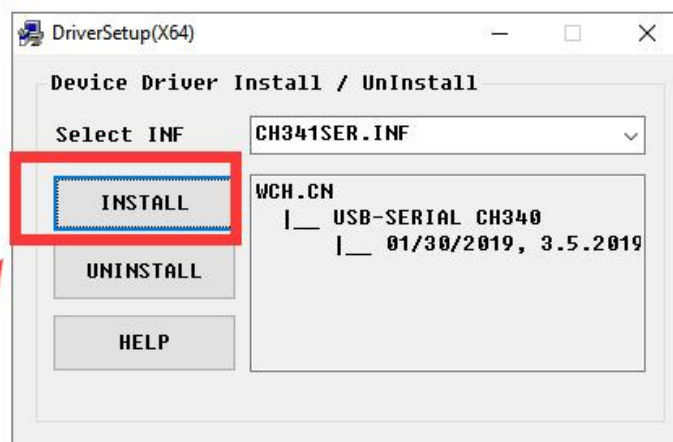
Install CH340 driver

1. You need to find the user folder provided by Adeept: UnoCar-B, find the 01 Software Package folder, and open the Adeept driver folder. If you are using a Windows system, you can directly double-click to open CH341SER_Windows.EXE, install corresponding driver according to the computer operating system. For MAC and Linux systems, you need to decompress the file corresponding to the system name, and then install the CH340 driver.

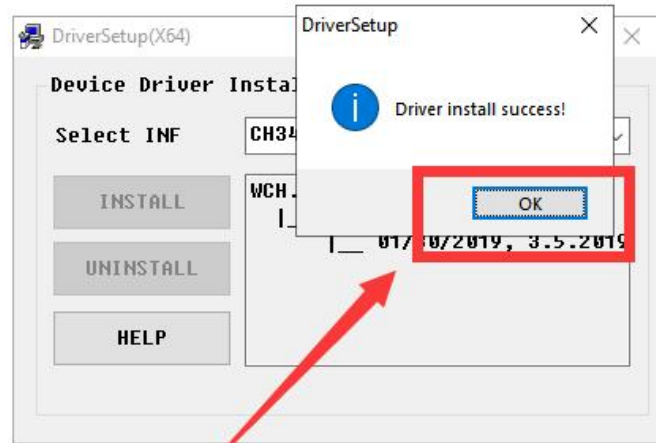
Name	Date modified	Type	Size
Adeept driver introduction.txt	6/17/2020 2:18 PM	Text Document	1 KB
CH341SER_ANDROID.ZIP	6/17/2020 2:18 PM	WinRAR ZIP 压缩...	2,360 KB
CH341SER_LINUX.ZIP	6/17/2020 2:18 PM	WinRAR ZIP 压缩...	9 KB
CH341SER_MAC.ZIP	6/17/2020 2:18 PM	WinRAR ZIP 压缩...	149 KB
CH341SER_Windows.EXE	6/17/2020 2:18 PM	Application	277 KB

3. Click INSTALL.Wait for the installation to succeed.And click OK.

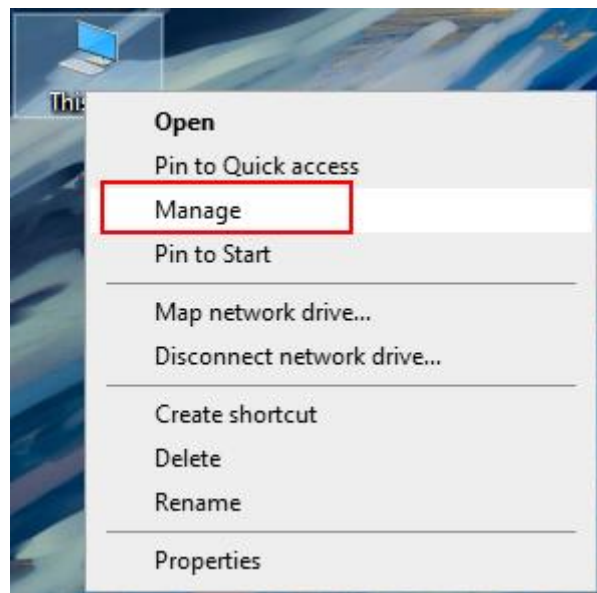
Name	Date modified	Type	Size
Adeept driver introduction.txt	6/17/2020 2:18 PM	Text Document	1 KB
CH341SER_ANDROID.ZIP	6/17/2020 2:18 PM	WinRAR ZIP 压缩...	2,360 KB
CH341SER_LINUX.ZIP	6/17/2020 2:18 PM	WinRAR ZIP 压缩...	9 KB
CH341SER_MAC.ZIP	6/17/2020 2:18 PM	WinRAR ZIP 压缩...	149 KB
CH341SER_Windows.EXE	6/17/2020 2:18 PM	Application	277 KB

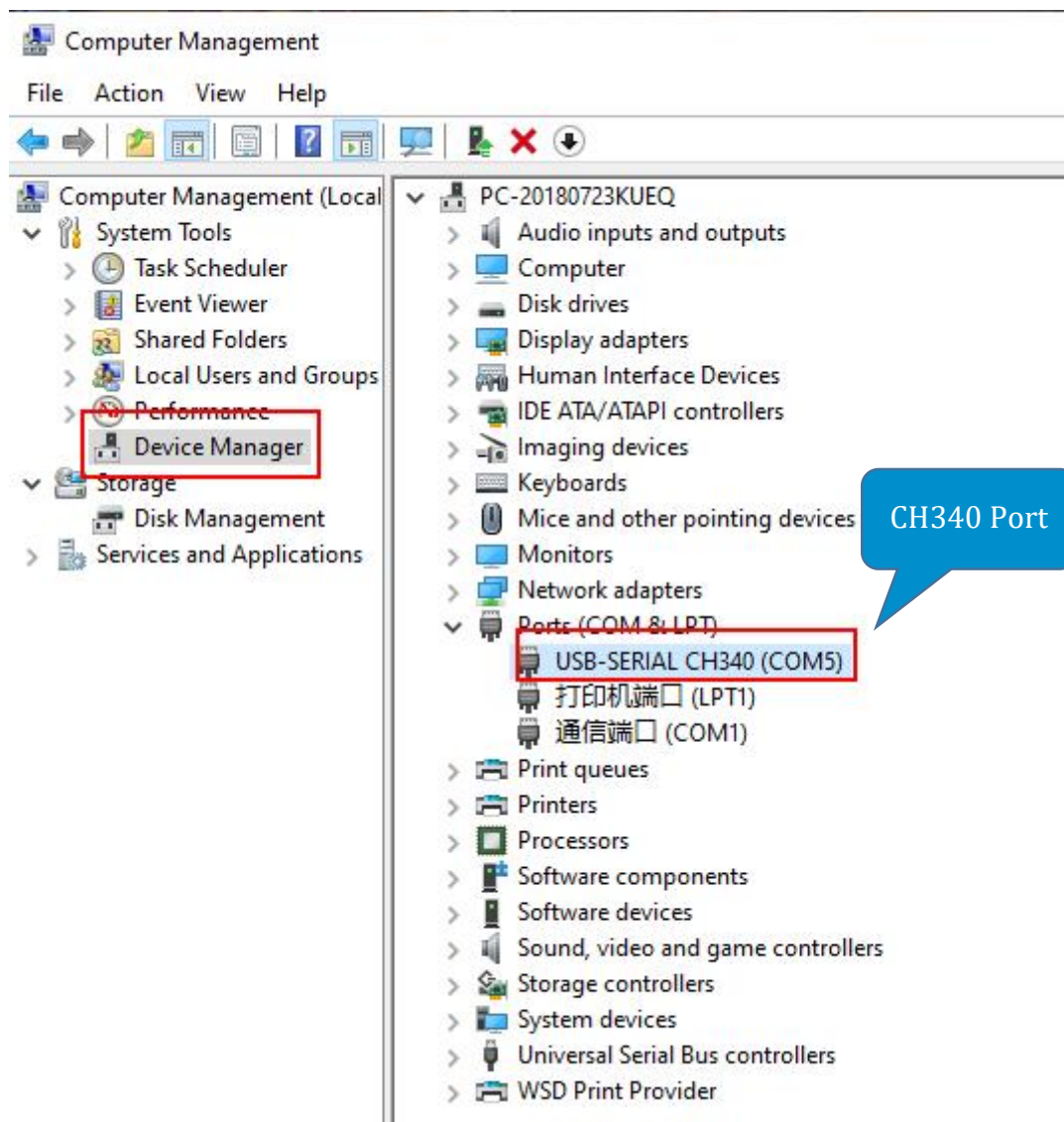


Adeept driver introduction.txt	6/17/2020 2:18 PM	Text Document	1 KB
CH341SER_ANDROID.ZIP	6/17/2020 2:18 PM	WinRAR ZIP 压缩...	2,360 KB
CH341SER_LINUX.ZIP	6/17/2020 2:18 PM	WinRAR ZIP 压缩...	9 KB
CH341SER_MAC.ZIP	6/17/2020 2:18 PM	WinRAR ZIP 压缩...	149 KB
CH341SER_Windows.EXE	6/17/2020 2:18 PM	Application	277 KB



4. Turn to the main interface of your computer, select “This PC” and right-click to select “Manage”.





The solution for situation that Arduino IDE cannot be opened

When opening the Arduino IDE, you will suddenly encounter a situation that it cannot be opened.

**【Solution】**

You need to find the Arduino15 folder in the \Users\ASUS\AppData\Local\Arduino15 directory of the C drive. As shown below:

Name	Date modified	Type	Size
cache	5/21/2020 6:35 PM	File folder	
logs	5/21/2020 6:34 PM	File folder	
library_index.json	6/10/2020 10:45 AM	JSON File	12
library_index.json.sig	6/10/2020 10:45 AM	SIG File	
package_index.json	6/10/2020 10:45 AM	JSON File	
package_index.json.sig	6/10/2020 10:45 AM	SIG File	
preferences.txt	6/10/2020 10:43 AM	Text Document	

You need to delete the package_index.json file, and then reopen the Arduino IDE.

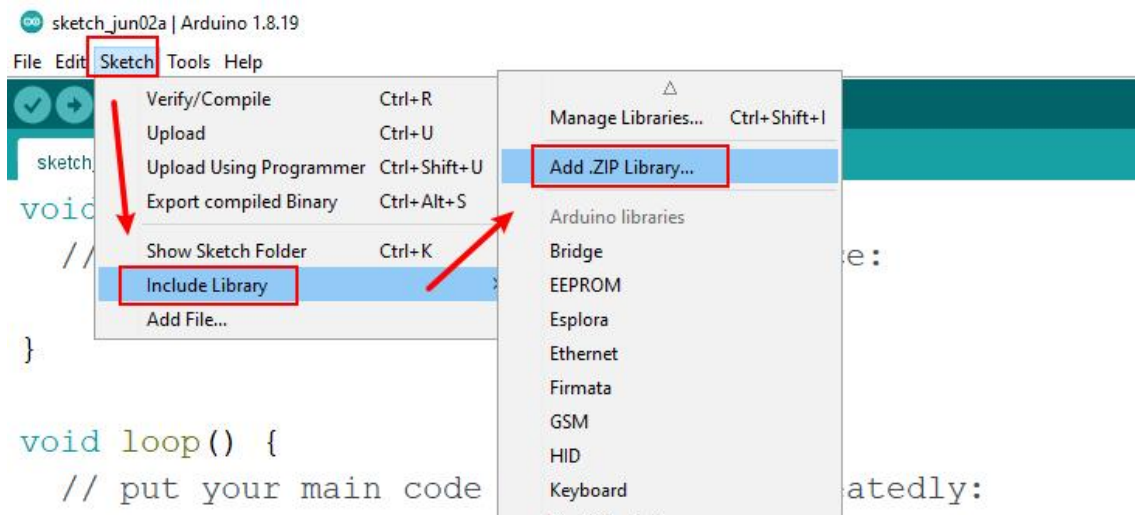
cache	5/21/2020 6:35 PM	File folder
logs	5/21/2020 6:34 PM	File folder
library_index.json	6/10/2020 10:58 AM	JSON File
library_index.json.sig	6/10/2020 10:58 AM	SIG File
package_index.json	6/10/2020 10:58 AM	JSON File
package_index.json.sig	6/10/2020 10:58 AM	SIG File
package_index.txt	6/10/2020 10:45 AM	Text Document
preferences.txt	6/10/2020 10:58 AM	Text Document

Uploading the First Code

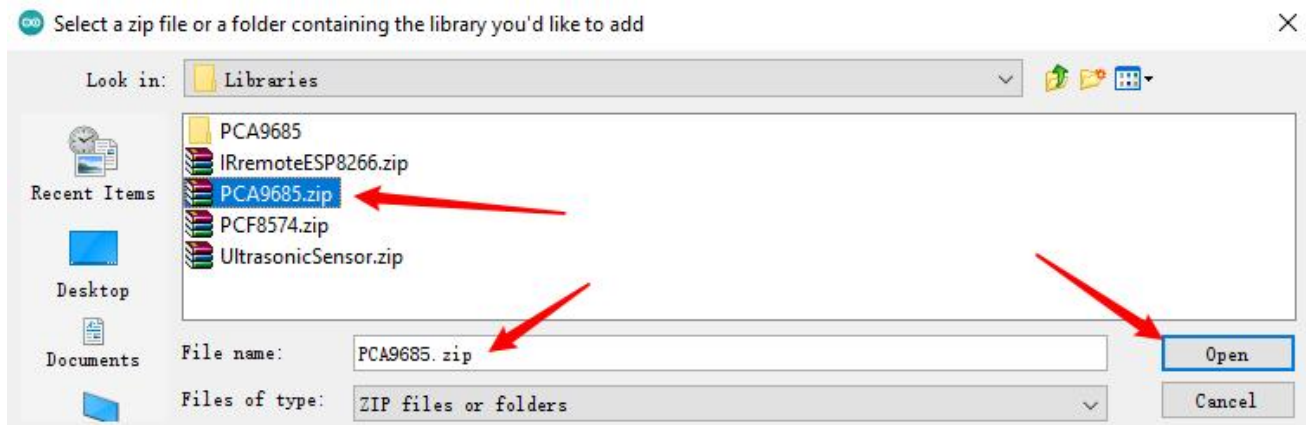
Here we use “00_Servo_90” in “**Adeept_UnoCar-B\Code**” as an example. The servo on the car is controlled by PCA9685. Therefore, it is necessary to add the related libraries to Arduino IDE.

How to Add libraries

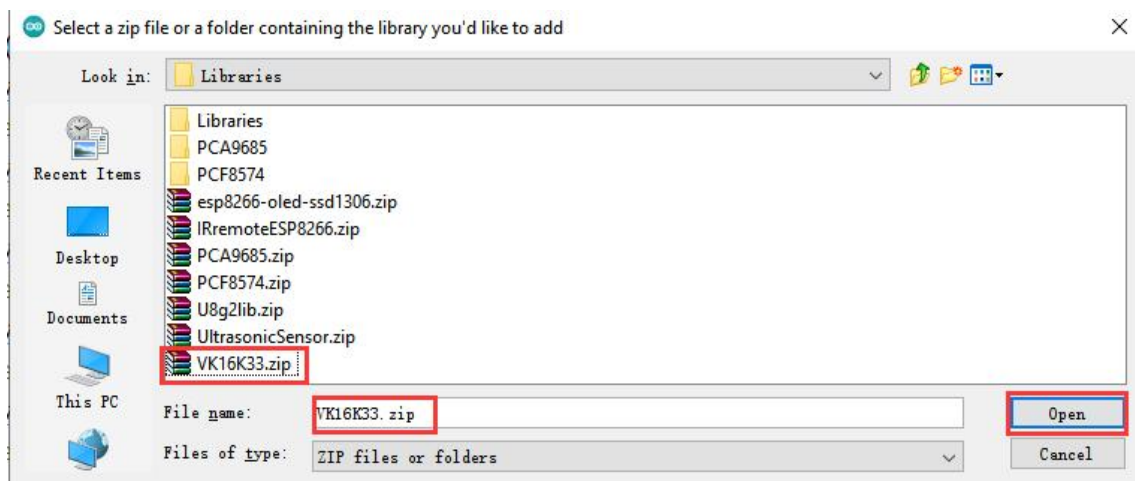
Open Arduino IDE, click **Sketch** on Menu bar, move your mouse to **Include Library** and then click **Add .ZIP library**.



Enter **Adeept_UnoCar-B\Libraries** folder, Add PCA9685.



Add VK16K33.



You need to add **all the ZIP files** in the folder to the Arduino IDE one by one.

Name

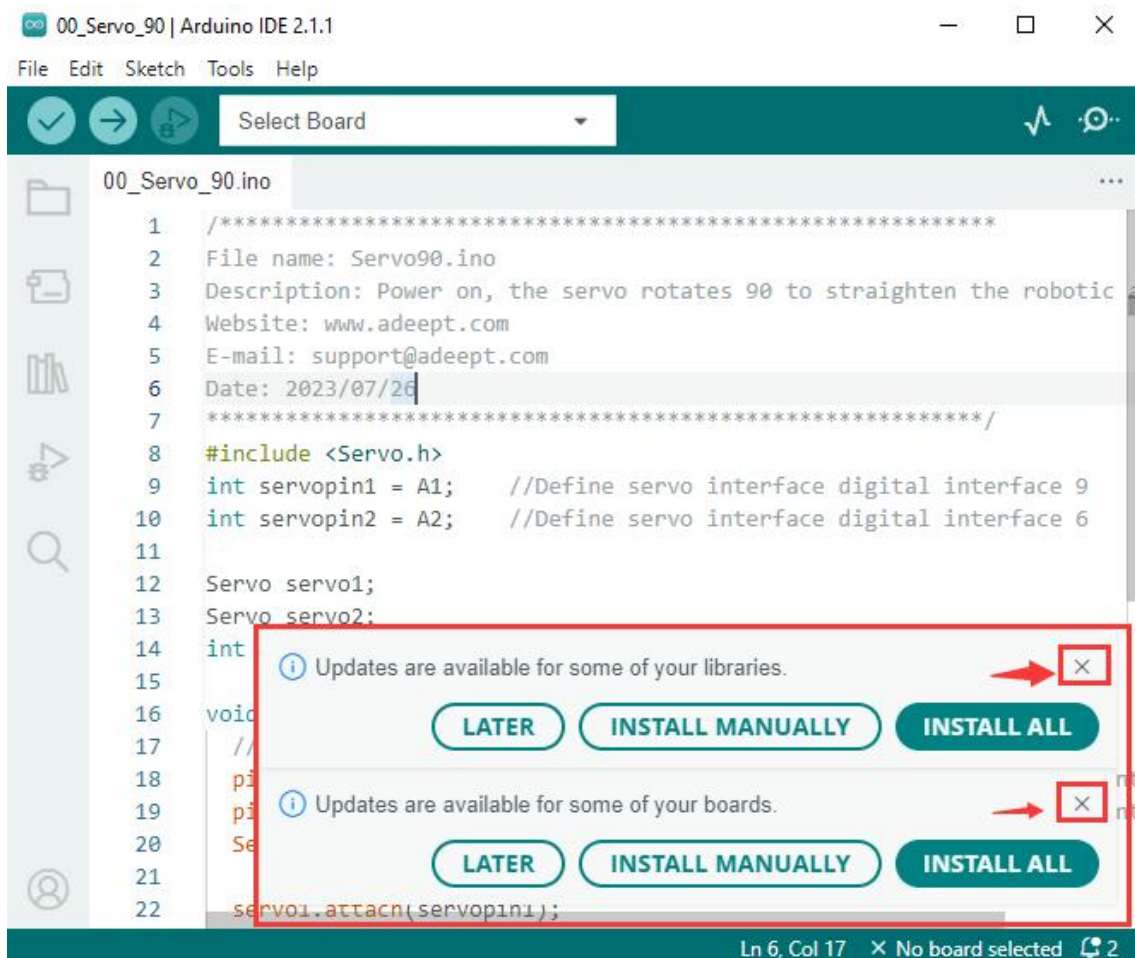
- Adafruit_BusIO.zip
- Adafruit_GFX_Library.zip
- Adafruit_NeoPixel.zip
- Adafruit_PWM_Servo_Driver_Library.zip
- Adafruit_SSD1306.zip
- IRremote.zip
- RF24.zip
- Servo.zip
- VK16K33.zip
- wimleers-flexitimer2-7338cfb.zip

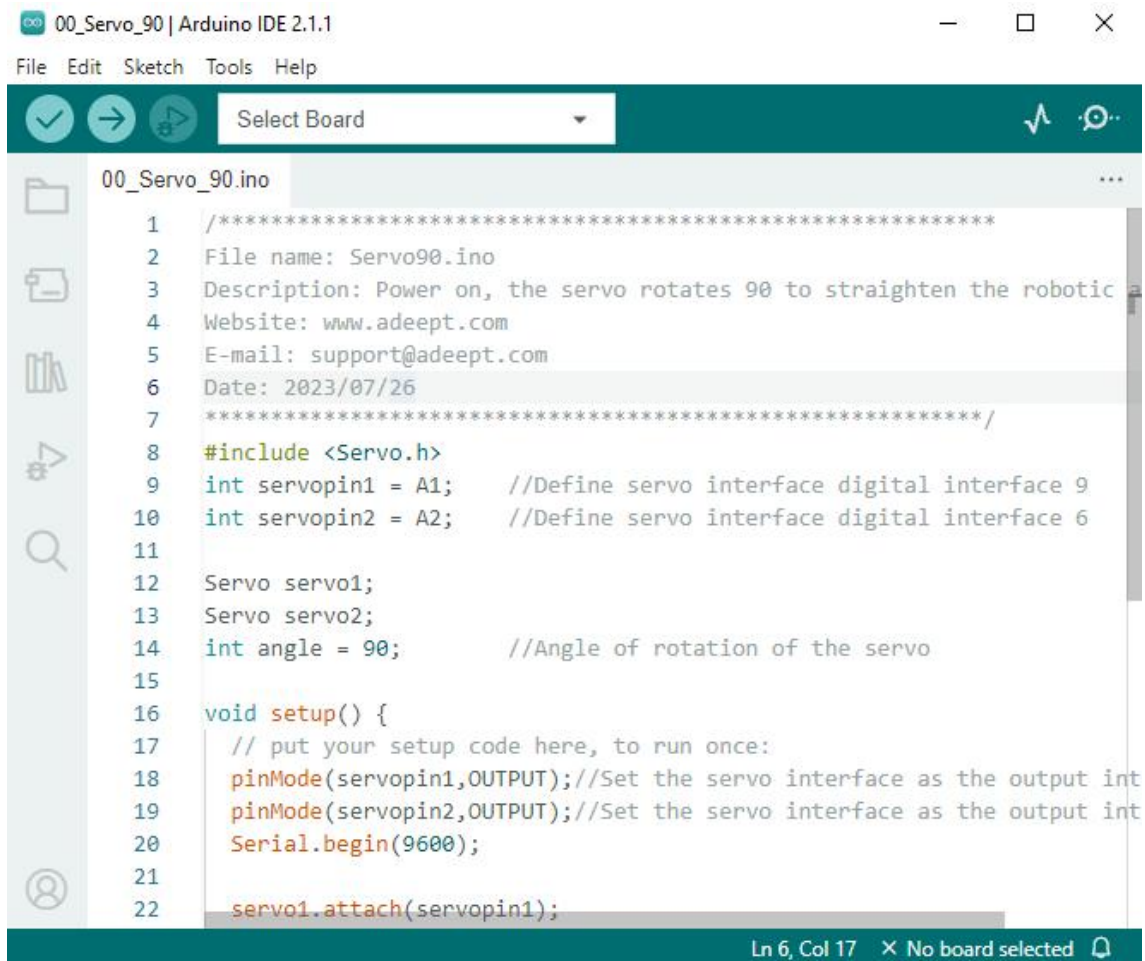
Upload code

1. Connect your computer and Arduino Board with a USB cable.
2. Open “00.0_Servo_90” folder in “Adeept_UnoCar-B/Code”, double-click “00_Servo_90.ino”.

The code is to rotate the servo motors to 90°

Please ignore the update prompt.



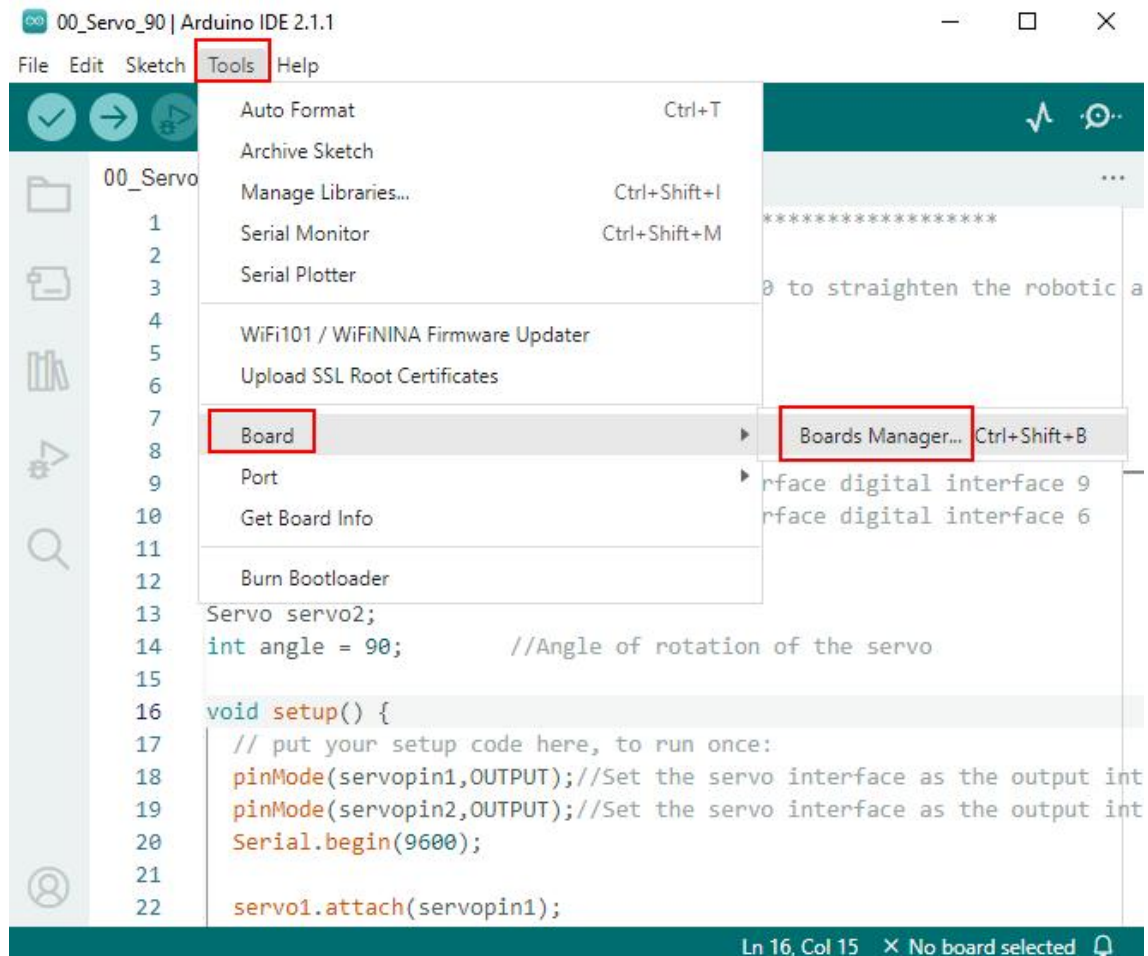


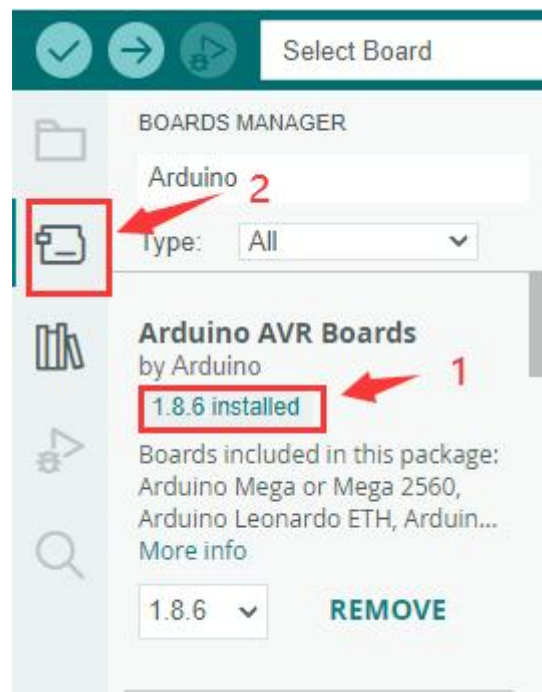
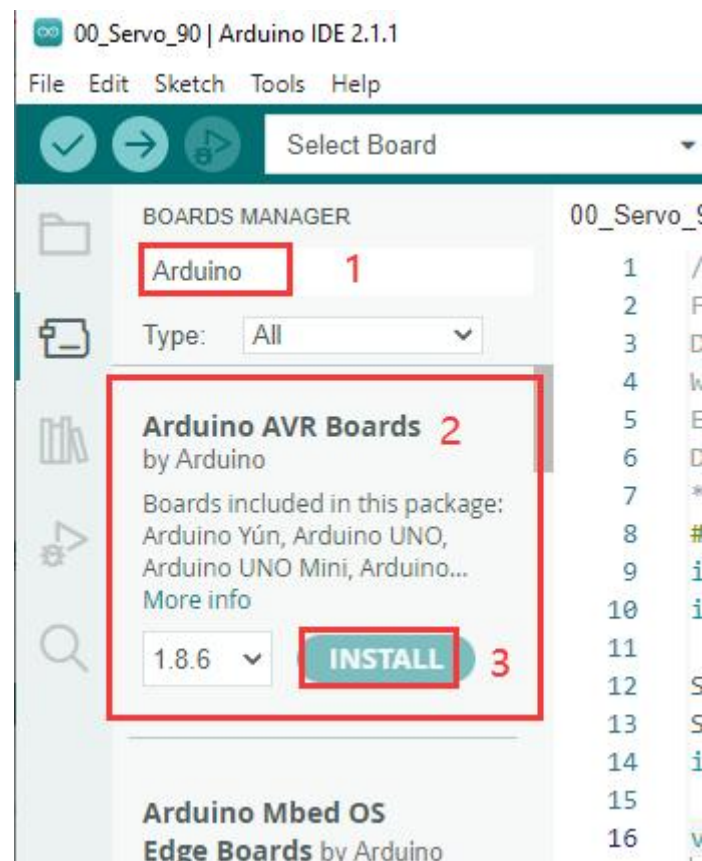
```
00_Servo_90 | Arduino IDE 2.1.1
File Edit Sketch Tools Help

00_Servo_90.ino
1  /*****
2  File name: Servo90.ino
3  Description: Power on, the servo rotates 90 to straighten the robotic
4  Website: www.adeept.com
5  E-mail: support@adeept.com
6  Date: 2023/07/26
7  *****/
8  #include <Servo.h>
9  int servopin1 = A1;    //Define servo interface digital interface 9
10 int servopin2 = A2;    //Define servo interface digital interface 6
11
12 Servo servo1;
13 Servo servo2;
14 int angle = 90;        //Angle of rotation of the servo
15
16 void setup() {
17   // put your setup code here, to run once:
18   pinMode(servopin1,OUTPUT); //Set the servo interface as the output int
19   pinMode(servopin2,OUTPUT); //Set the servo interface as the output int
20   Serial.begin(9600);
21
22   servo1.attach(servopin1);
```

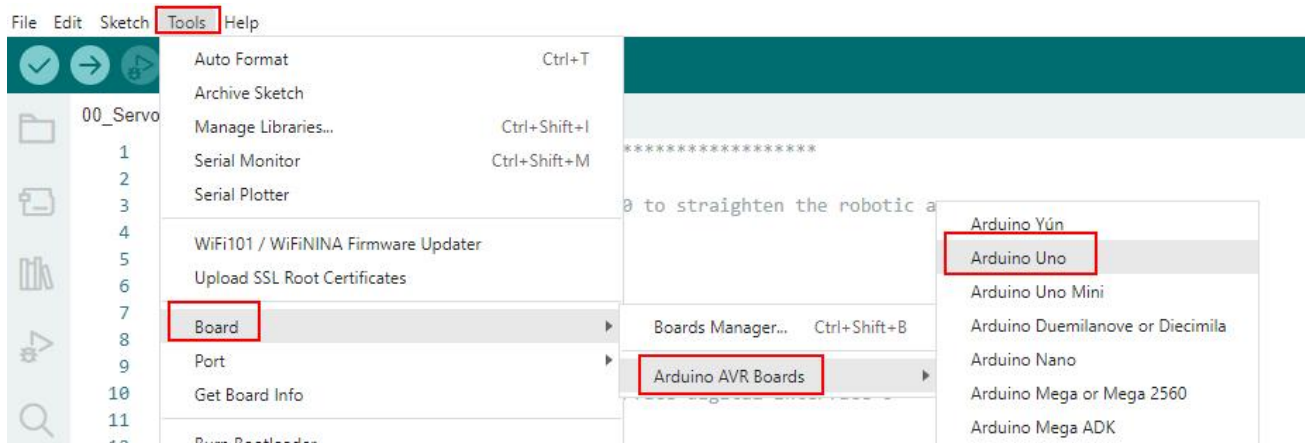
3. Install development board.

Click **Tools** --> **Board** --> **Board Manager..** , Enter "**Arduino**" in the left column, select "**Arduino AVR Boards**", click "**INSTALL**". Wait for the installation to complete. Once the installation is complete, click the second icon from the left to close the left sidebar.



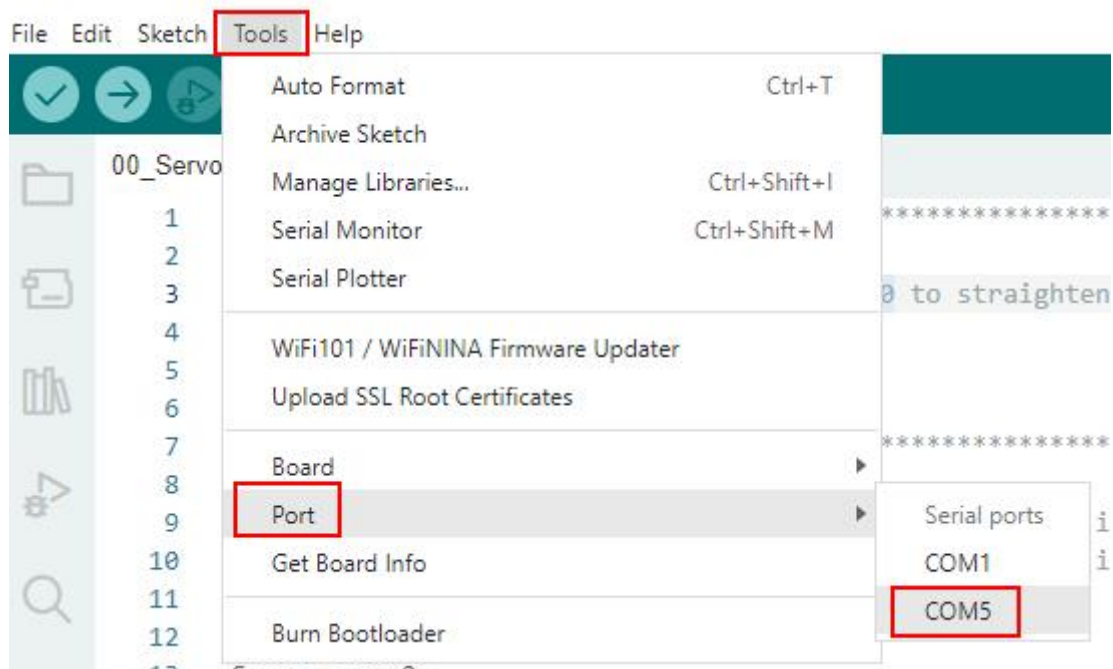


4. After the installation is complete, Select development board. Click [Tools](#) --> [Board](#) --> [Arduino AVR Boards](#) --> [Arduino UNO](#).



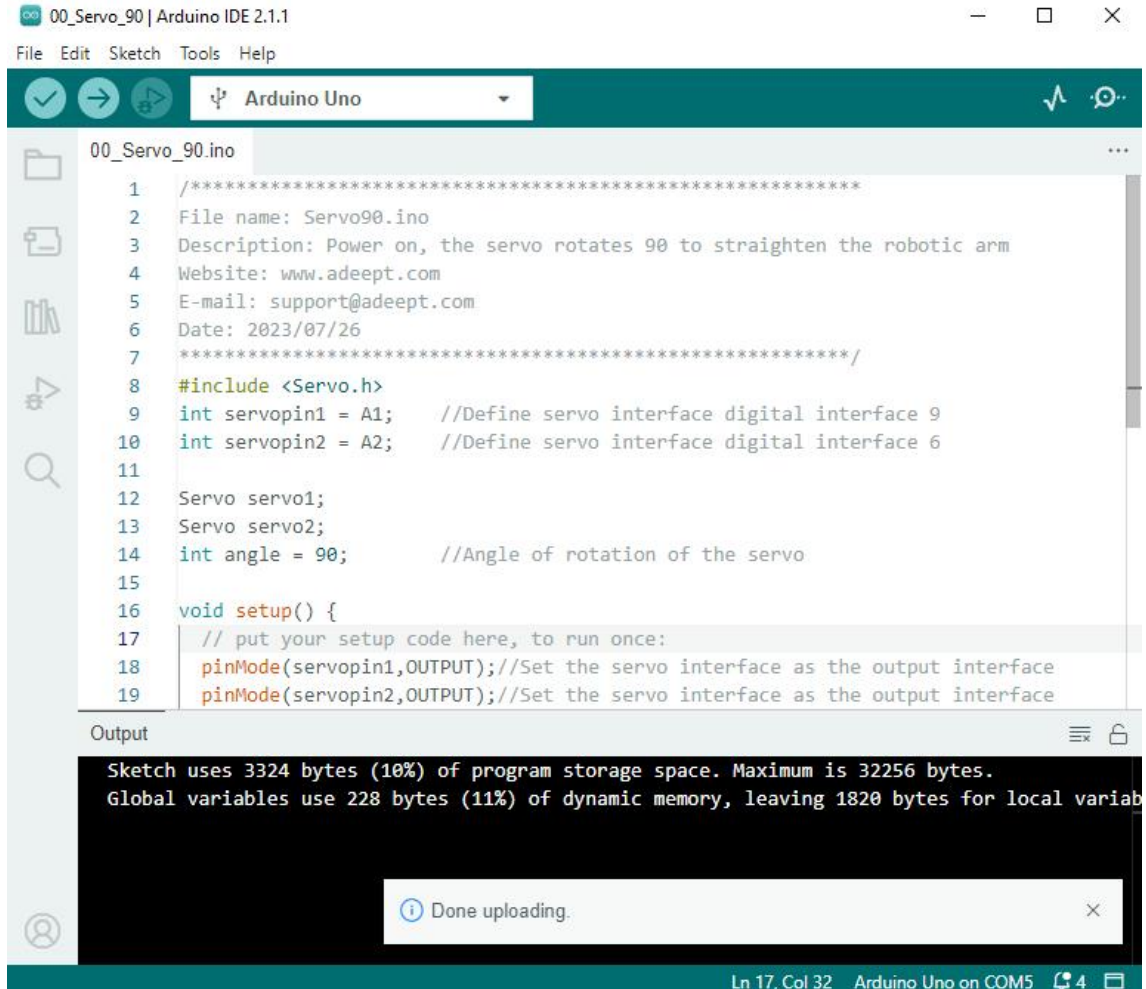
5. Select serial port.

Click Tools on Menu bar, move your mouse to Port and select COMx on your computer. The value of COMx varies in different computers, but it won't affect the download function of Arduino, as long as you select the correct one.



6. Click "Upload" and the program will be downloaded to Arduino board.

The red error message is not displayed in the Output. "Done uploading." appear to indicate a successful upload.



The screenshot shows the Arduino IDE 2.1.1 interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar shows icons for checking, running, and uploading. The target board is set to "Arduino Uno". The sketch file is "00_Servo_90.ino". The code in the editor includes a header comment, pin definitions, servo object declarations, and a setup function that configures two servos as outputs. The Output window at the bottom shows the memory usage: "Sketch uses 3324 bytes (10%) of program storage space. Maximum is 32256 bytes. Global variables use 228 bytes (11%) of dynamic memory, leaving 1820 bytes for local variables." A status message "Done uploading." is displayed in a white box at the bottom of the Output window. The status bar at the bottom indicates "Ln 17, Col 32" and "Arduino Uno on COM5".

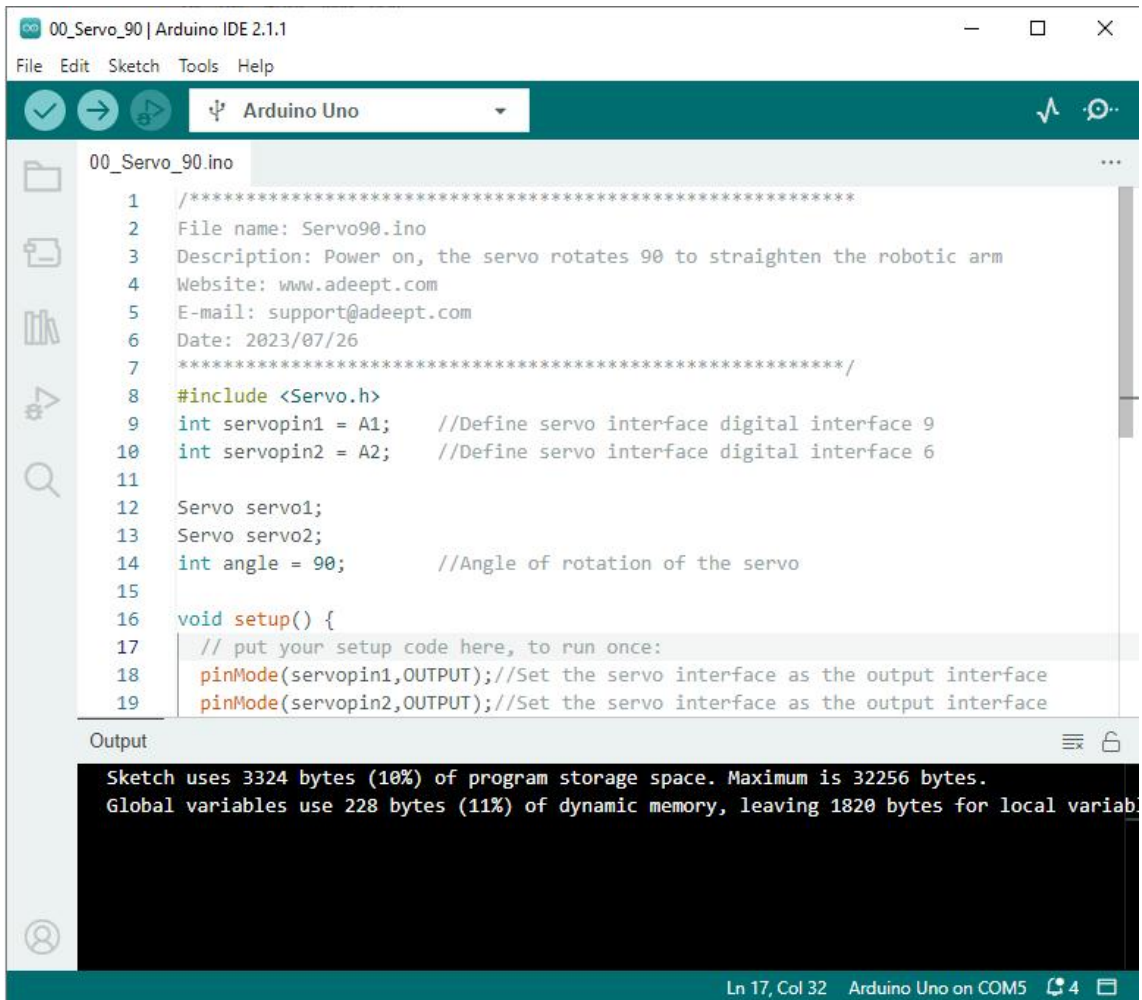
```
1  /*****
2  File name: Servo90.ino
3  Description: Power on, the servo rotates 90 to straighten the robotic arm
4  Website: www.adeept.com
5  E-mail: support@adeept.com
6  Date: 2023/07/26
7  *****/
8  #include <Servo.h>
9  int servopin1 = A1;    //Define servo interface digital interface 9
10 int servopin2 = A2;    //Define servo interface digital interface 6
11
12 Servo servo1;
13 Servo servo2;
14 int angle = 90;        //Angle of rotation of the servo
15
16 void setup() {
17   // put your setup code here, to run once:
18   pinMode(servopin1,OUTPUT);//Set the servo interface as the output interface
19   pinMode(servopin2,OUTPUT);//Set the servo interface as the output interface
```

Output

Sketch uses 3324 bytes (10%) of program storage space. Maximum is 32256 bytes.
Global variables use 228 bytes (11%) of dynamic memory, leaving 1820 bytes for local variables

Done uploading.

Ln 17, Col 32 Arduino Uno on COM5



```
00_Servo_90 | Arduino IDE 2.1.1
File Edit Sketch Tools Help
Arduino Uno
00_Servo_90.ino
1  /*****
2  File name: Servo90.ino
3  Description: Power on, the servo rotates 90 to straighten the robotic arm
4  Website: www.adeept.com
5  E-mail: support@adeept.com
6  Date: 2023/07/26
7  *****/
8  #include <Servo.h>
9  int servopin1 = A1;    //Define servo interface digital interface 9
10 int servopin2 = A2;    //Define servo interface digital interface 6
11
12 Servo servo1;
13 Servo servo2;
14 int angle = 90;        //Angle of rotation of the servo
15
16 void setup() {
17   // put your setup code here, to run once:
18   pinMode(servopin1,OUTPUT);//Set the servo interface as the output interface
19   pinMode(servopin2,OUTPUT);//Set the servo interface as the output interface

```

Output

```
Sketch uses 3324 bytes (10%) of program storage space. Maximum is 32256 bytes.
Global variables use 228 bytes (11%) of dynamic memory, leaving 1820 bytes for local variab

```

Ln 17, Col 32 Arduino Uno on COM5 4

Each file upload will overwrite the existing program in Arduino Board.