4DOF Smart Robot Mechanical Arm Kit

Content

Company Profile

Established in 2011, lafvin is a manufacturer and trader specialized in research,development and production of 2560 uno,nano boards,and all kinds of accessories or sensors use for arduino,rasperrry.We also complete starter kits designed for interested lovers of any levels to learn Arduino or Raspberry.We are located in Shenzhen,China.All of our products comply with international quality standards and are greatly appreciated in a variety of different markets throughout the world.

Customer Service

We are cooperating with a lot of companies from diffirent countries.Also help them to purchase electronic component products in china,and became the biggest supplier of them. We look forward to build cooperate with more companies in future.

By the way,We also look forward to hearing from you and any of your critical comment or suggestions.Pls email us by lafvin_service@163.com if you have any questions or suggestions.

As a continuous and fast growing company. We keep striving our best to offer you excellent products and quality service.

Our Store

Aliexpress store: https://www.aliexpress.com/store/1942043Brand in Amazon:LAFVIN

Product Catalog

https://drive.google.com/drive/folders/0BwvEeRN9dKllblZING00TkhYbGs?usp=sharing

Tutorial

This tutorial include codes,labraries and lessons. It is designed for beginners. It will teach every users how to assembly the smart robotic arm kit and use Arduino controller board, sensors and modules.Our robotic arm kit has the following characteristics:
Visualized Installation Steps.
Rich control can meet the needs of DIY players,includes Wired JoyStick Control; Phone Bluetooth Control; Wireless PS2 JoyStick Control.

Packing list

Packing list:

1pcs Uno r3 board

4pcs MG90S Servo

1pcs V5.0 extension board

1pcs USB Cable

2pcs Joystick Module

1pcs 20PIN F-F Dupont Wire

1pcs Bluetooth Module

1pcs Screwdriver

1pcs Wrench

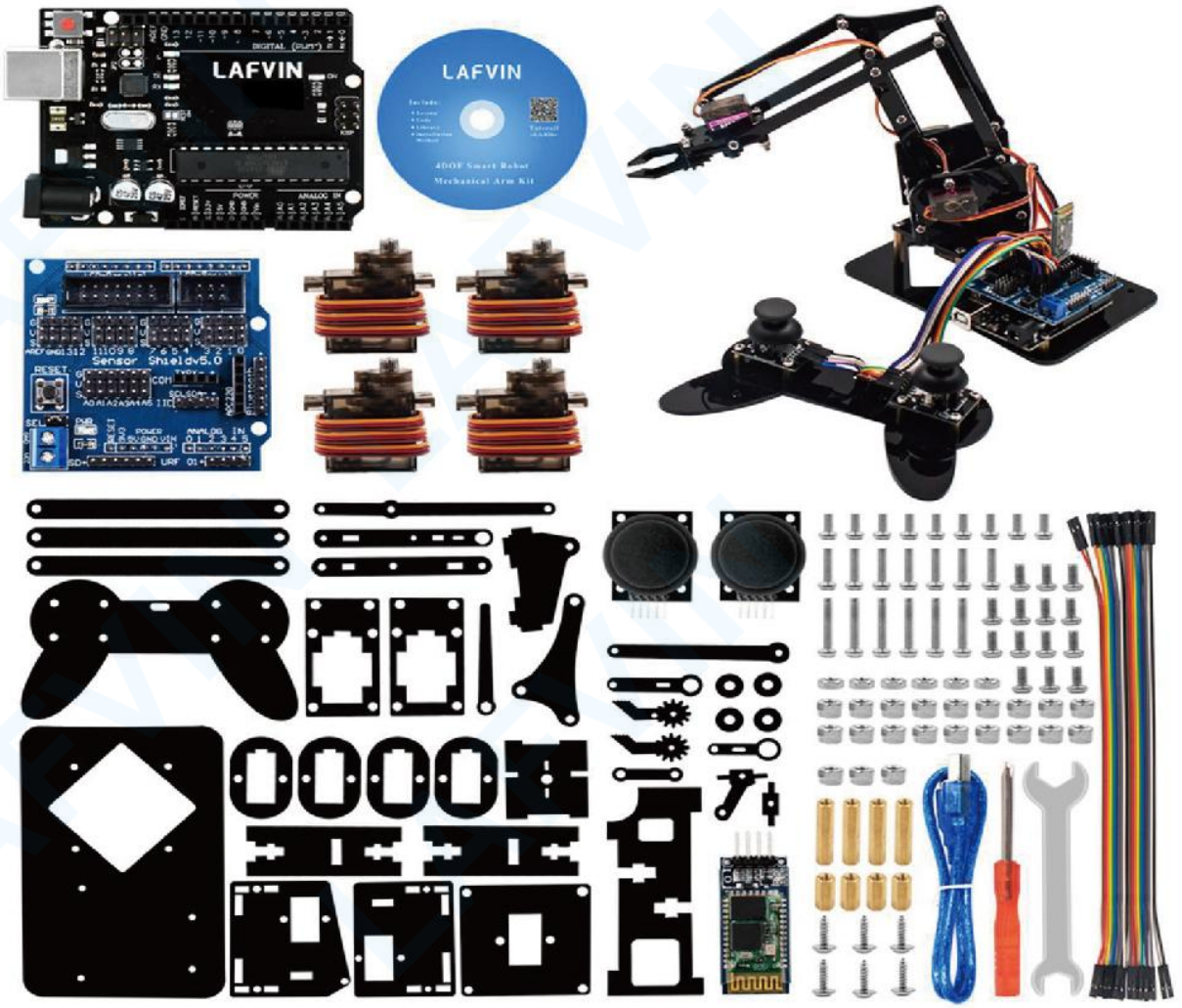
1set Acrylic Board

1set Screw Kit

1pcs CD Tutorial

2pcs Bunding belt

4pcs Rubber Mat



Lesson 1 Installing IDE

Introduction

The Arduino Integrated Development Environment (IDE) is the software side of the Arduino platform.

In this lesson, you will learn how to setup your computer to use Arduino and how to set about the lessons that follow.

The Arduino software that you will use to program your Arduino is available for Windows, Mac and Linux. The installation process is different for all three platforms and unfortunately there is a certain amount of manual work to install the software.

STEP 1: Go to [https://www.arduino.cc/en/Main/Software and find below page.](https://www.arduino.cc/en/Main/Software)

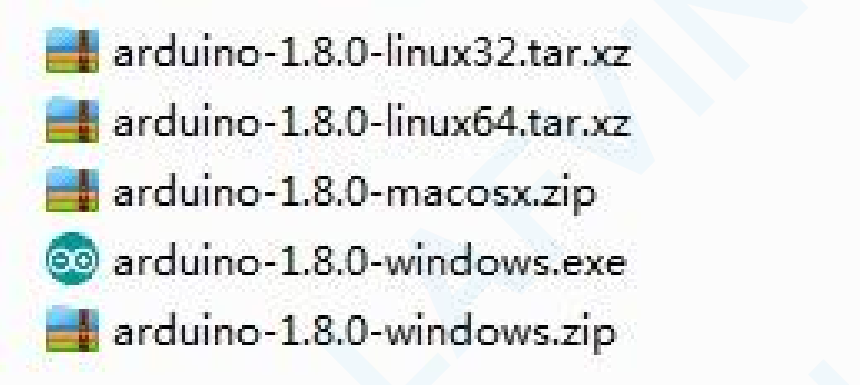
The version available at this website is usually the latest version, and the actual version may be newer than the version in the picture.

STEP2：Download the development software that is compatible with the operating system of your computer. Take Windows as an example here.

Click Windows Installer.

Click JUST DOWNLOAD.

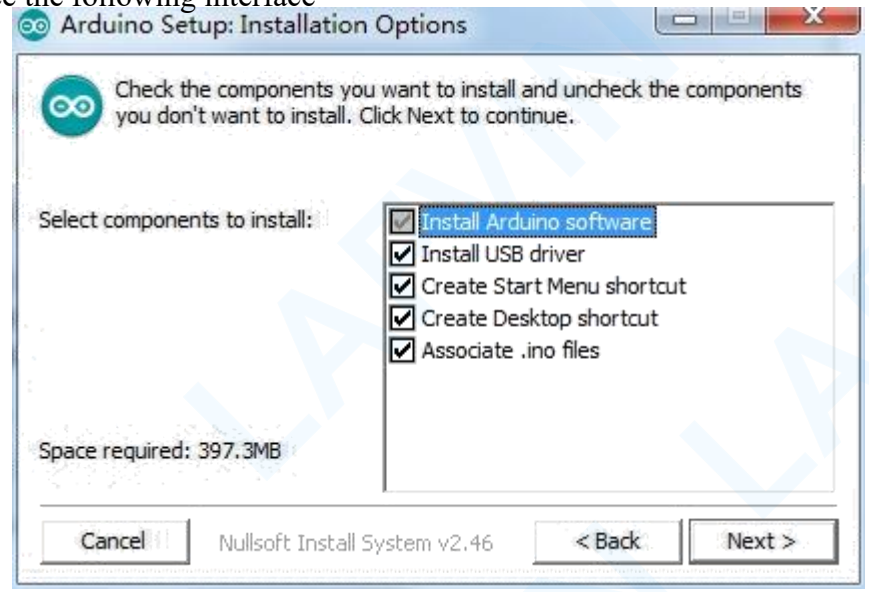
Also version 1.8.0 is available in the material we provided, and the versions of our materials are the latest versions when this course was made.



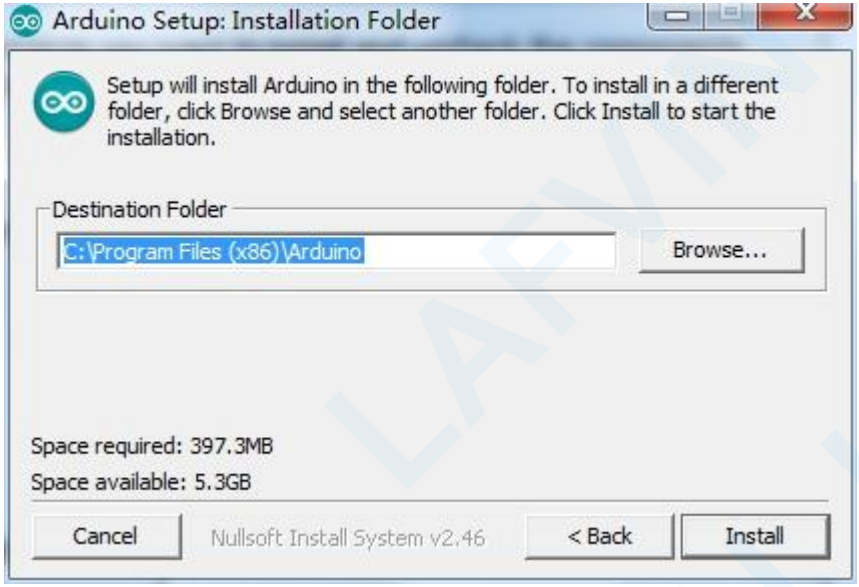
Installing Arduino (Windows)

Install Arduino with the exe. Installation package.

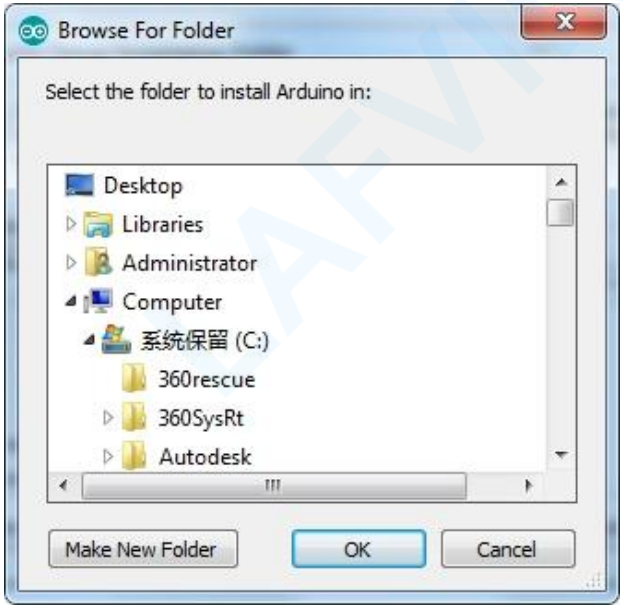
Click I Agree to see the following interface



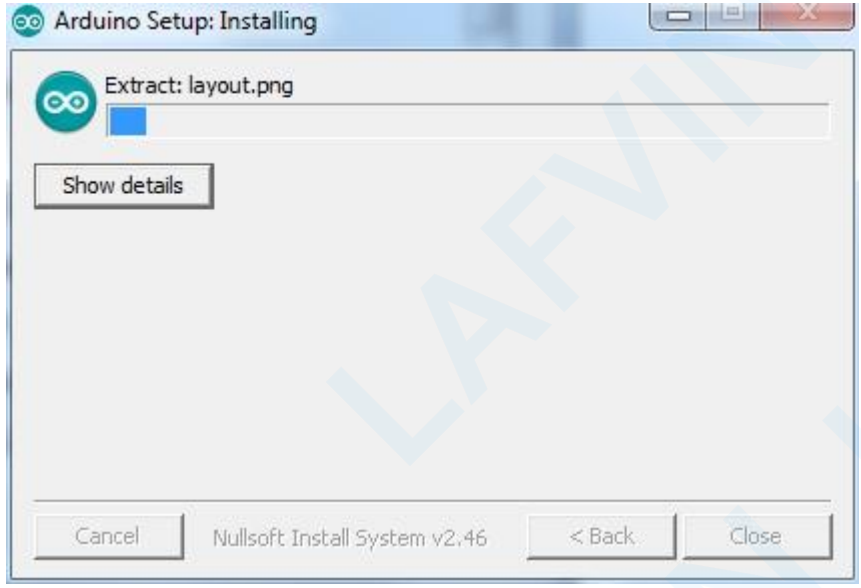
Click Next



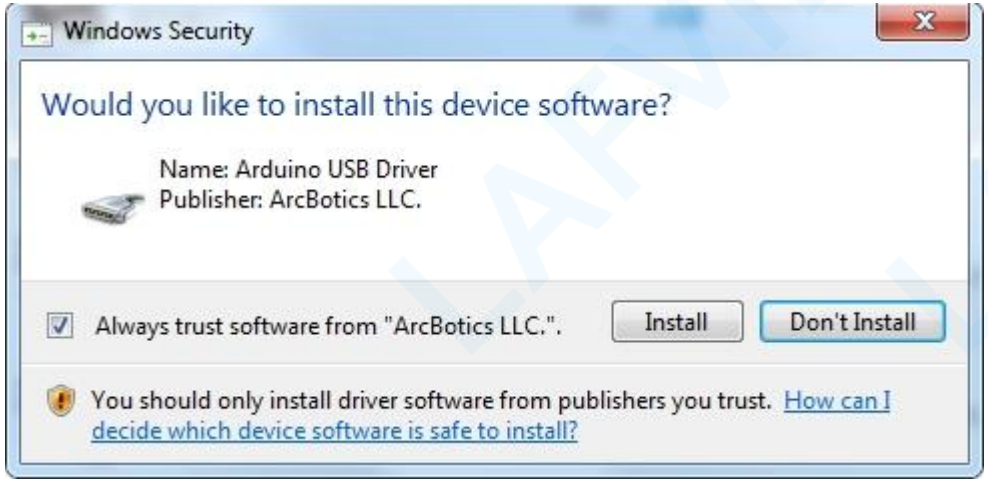
You can press Browse… to choose an installation path or directly type in the directory you want.



Click Install to initiate installation

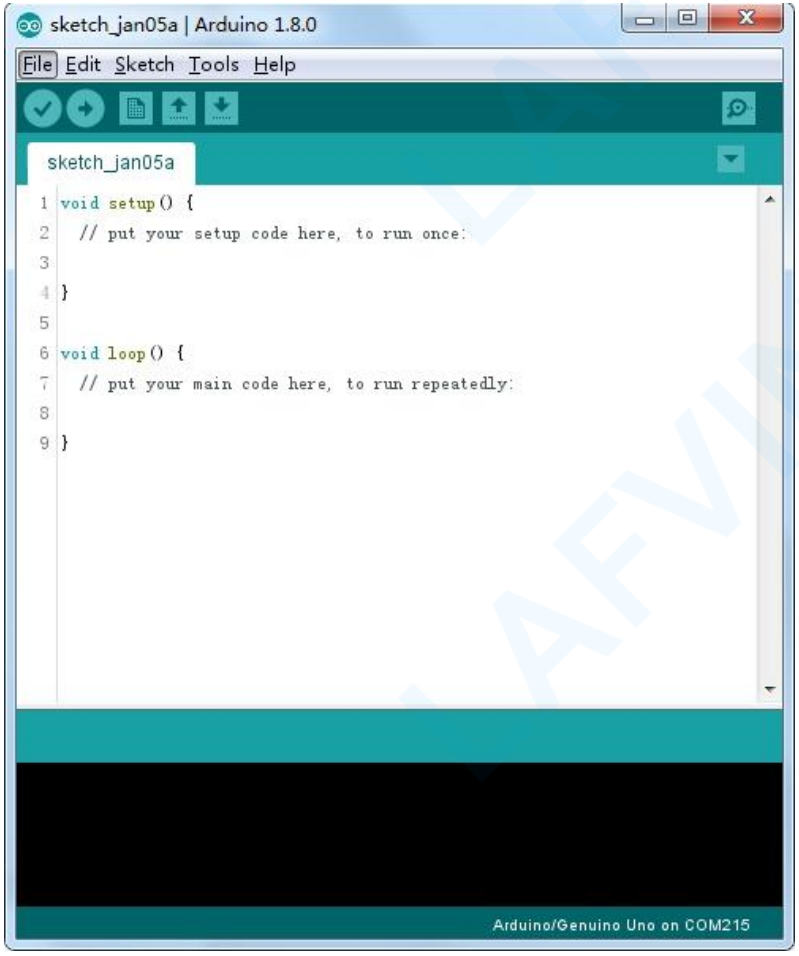


Finally, the following interface appears, click Install to finish the installation.



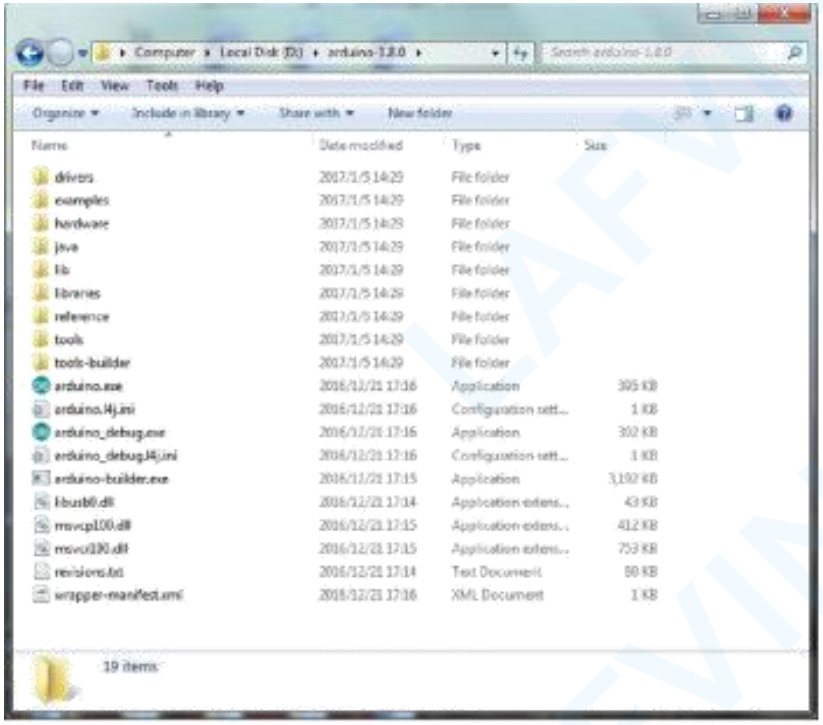
Next, the following icon appears on the desktop

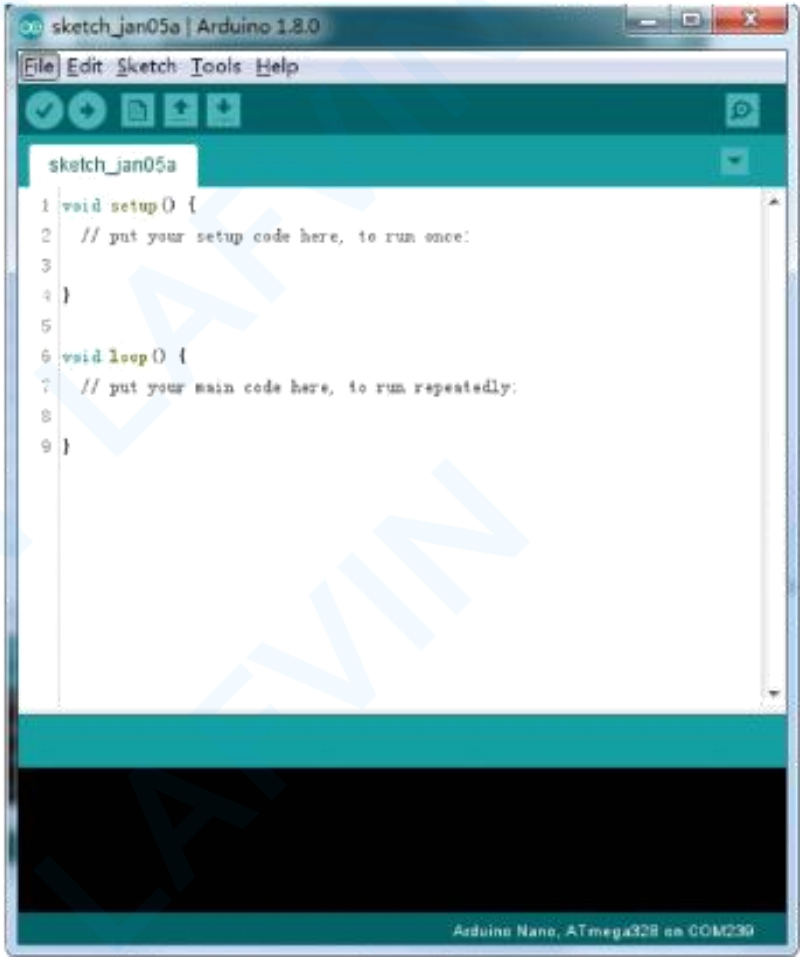
Double-click to enter the desired development environment



You may directly choose the installation package for installation and skip the contents below and jump to the next section. But if you want to learn some methods other than the installation package, please continue to read the section.

Unzip the zip file downloaded, Double-click to open the program and enter the desired development environment


Computer ▸ Local Disk (D:) ▸ arduino-1.8.0 ▸
File Edit View Tools Help
Organize ▾ Include in library ▾ Share with ▾ New folder
Name Date modified Type Size
drivers 2017/1/5 14:29 File folder
examples 2017/1/5 14:29 File folder
hardware 2017/1/5 14:29 File folder
java 2017/1/5 14:29 File folder
lib 2017/1/5 14:29 File folder
libraries 2017/1/5 14:29 File folder
reference 2017/1/5 14:29 File folder
tools 2017/1/5 14:29 File folder
tools-builder 2017/1/5 14:29 File folder
arduino.exe 2016/12/21 17:16 Application 305 KB
arduino.l4j.ini 2016/12/21 17:16 Configuration sett... 1 KB
arduino_debug.exe 2016/12/21 17:16 Application 302 KB
arduino_debug.l4j.ini 2016/12/21 17:16 Configuration sett... 1 KB
arduino-builder.exe 2016/12/21 17:15 Application 3,192 KB
libusb0.dll 2016/12/21 17:14 Application extens... 43 KB
msvcp100.dll 2016/12/21 17:15 Application extens... 412 KB
msvcr100.dll 2016/12/21 17:15 Application extens... 753 KB
revisions.txt 2016/12/21 17:14 Text Document 80 KB
wrapper-manifest.xml 2016/12/21 17:16 XML Document 1 KB
19 items


sketch_jan05a | Arduino 1.8.0
File Edit Sketch Tools Help
sketch_jan05a
1 void setup() {
2 // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7 // put your main code here, to run repeatedly:
8
9 }
Arduino Nano, ATmega328 on COM239

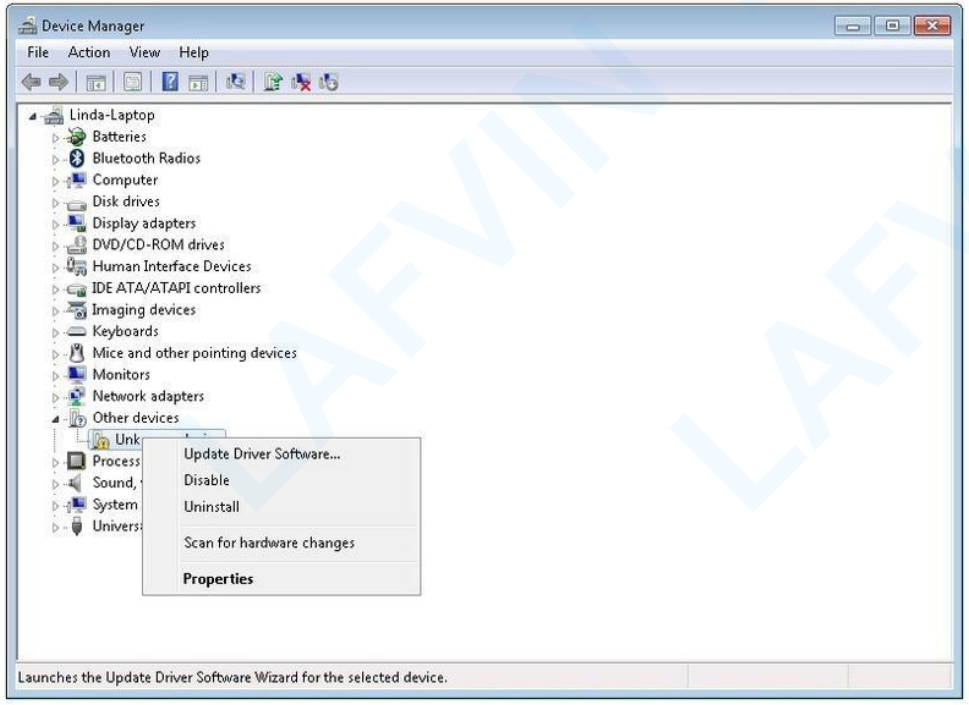
However, this installation method needs separate installation of driver.

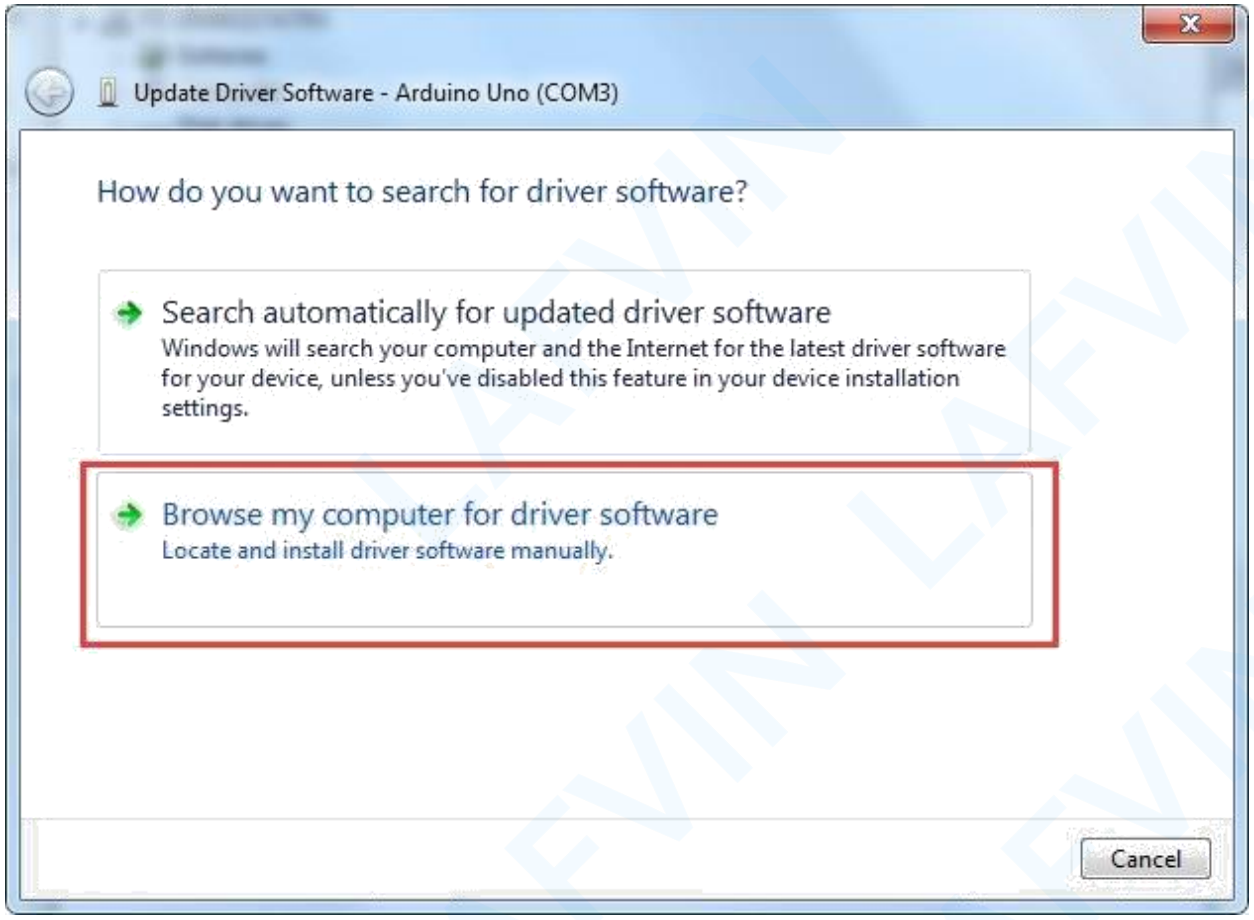
The Arduino folder contains both the Arduino program itself and the drivers that allow the Arduino to be connected to your computer by a USB cable. Before we launch the Arduino software, you are going to install the USB drivers.

Plug one end of your USB cable into the Arduino and the other into a USB socket on your computer. The power light on the LED will light up and you may get a 'Found New Hardware' message from Windows. Ignore this message and cancel any attempts that Windows makes to try and install drivers automatically for you.

The most reliable method of installing the USB drivers is to use the Device Manager. This is accessed in different ways depending on your version of Windows. In Windows 7, you first have to open the Control Panel, then select the option to view Icons, and you should find the Device Manager in the list.

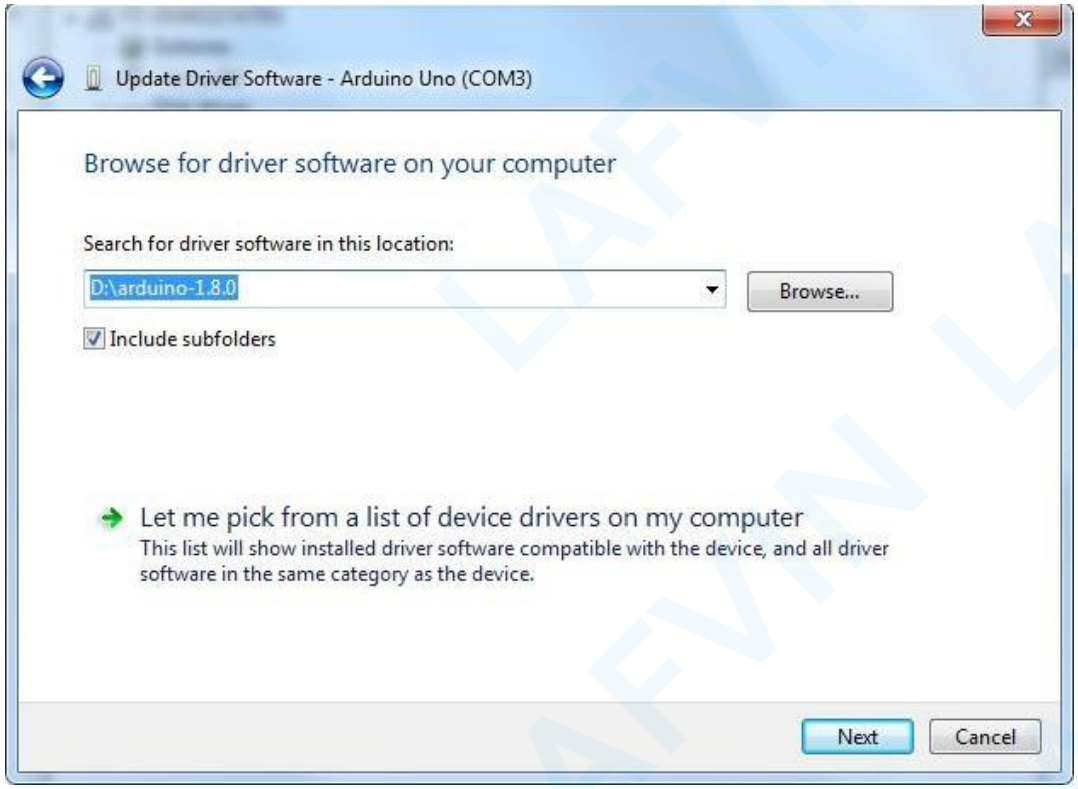
Under ‘Other Devices’, you should see an icon for ‘unknown device’ with a little yellow warning triangle next to it. This is your Arduino.



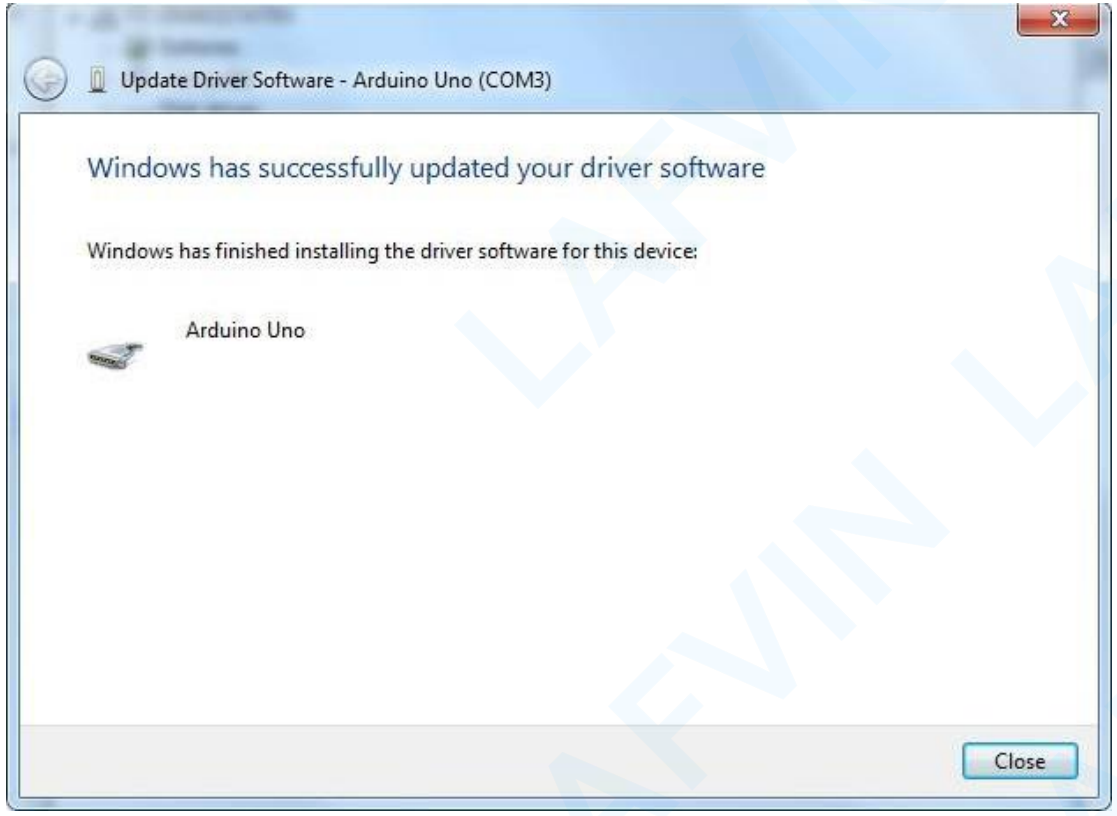

Update Driver Software - Arduino Uno (COM3)
How do you want to search for driver software?
Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.
Browse my computer for driver software
Locate and install driver software manually.
Cancel

Right-click on the device and select the top menu option (Update Driver Software...).

You will then be prompted to either ‘Search Automatically for updated driver software’ or ‘Browse my computer for driver software’. Select the option to browse and navigate to the X\arduino1.8.0\drivers.



Click 'Next' and you may get a security warning, if so, allow the software to be installed. Once the software has been installed, you will get a confirmation message.



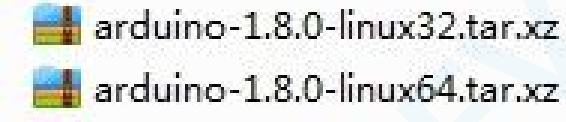
Windows users may skip the installation directions for Mac and Linux systems and jump to Lesson 1. Mac and Linux users may continue to read this section.

Installing Arduino (Mac OS X)

Download and Unzip the zip file, double click the Arduino.app to enter Arduino IDE; the system will ask you to install Java runtime library if you don't have it in your computer. Once the installation is complete you can run the Arduino IDE.

Installing Arduino (Linux)

You will have to use the make install command. If you are using the Ubuntu system, it is recommended to install Arduino IDE from the software center of Ubuntu.



Lesson 2 Add Libraries and Open Serial Monitor

Installing Additional Arduino Libraries

Once you are comfortable with the Arduino software and using the built-in functions, you may want to extend the ability of your Arduino with additional libraries.

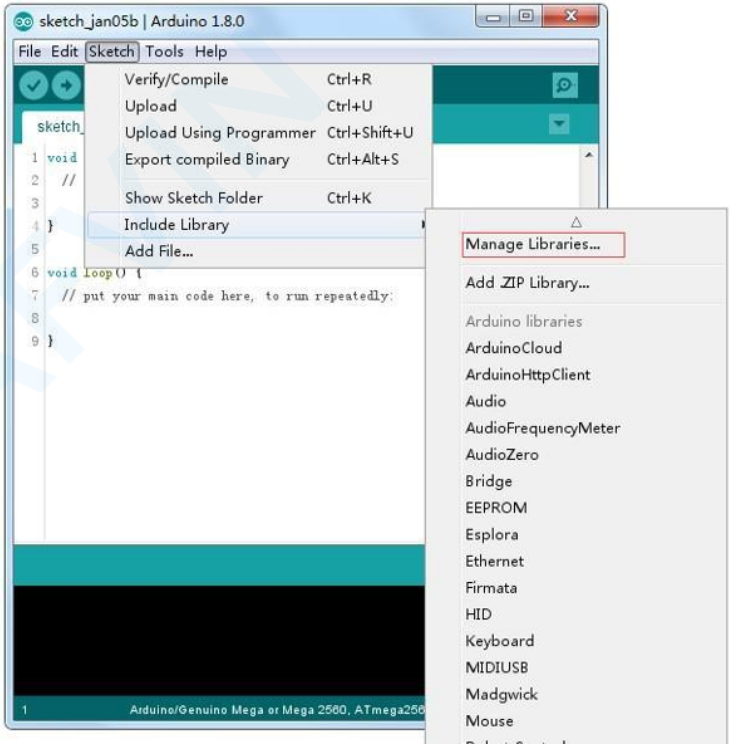
What are Libraries?

Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. For example, the built-in LiquidCrystal library makes it easy to talk to character LCD displays. There are hundreds of additional libraries available on the Internet for download. The built-in libraries and some of these additional libraries are listed in the reference. To use the additional libraries, you will need to install them.

How to Install a Library

Using the Library Manager

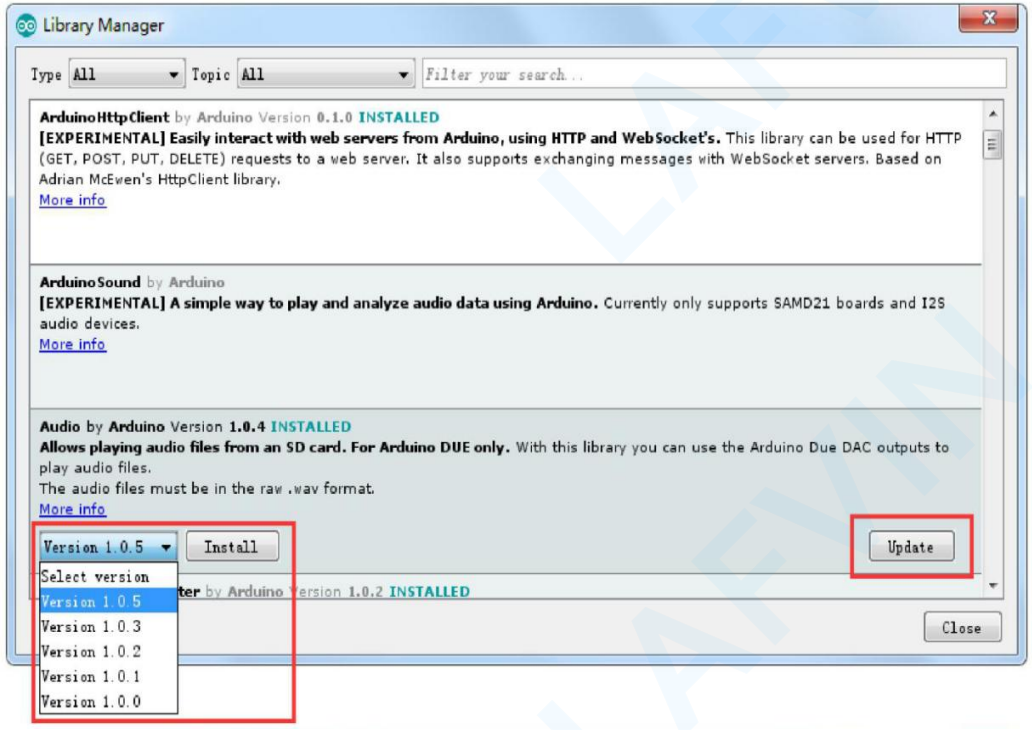
To install a new library into your Arduino IDE you can use the Library Manager (available from IDE version 1.8.0). Open the IDE and click to the "Sketch" menu and then Include Library > Manage Libraries.

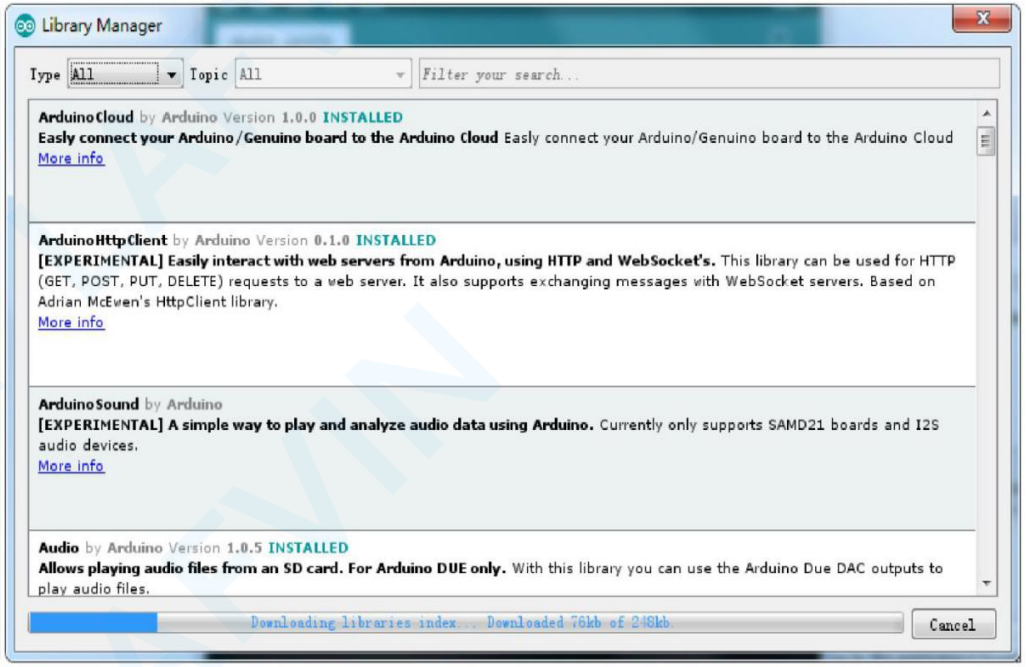


Then the library manager will open and you will find a list of libraries that are already installed or ready for installation. In this example we will install the Bridge library. Scroll the list to find it, then select the version of the library you want to install. Sometimes only one version of the library is available. If the version selection menu does not appear, don't worry: it is normal.

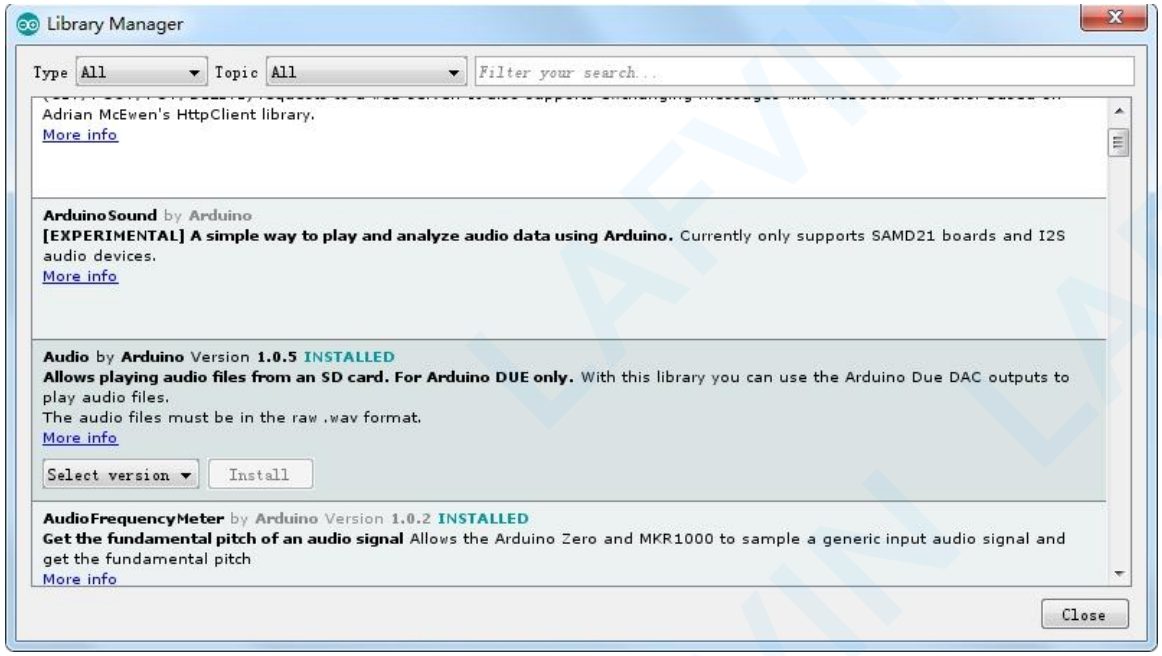
There are times you have to be patient with it, just as shown in the figure. Please refresh it

and wait.





Finally click on install and wait for the IDE to install the new library. Downloading may take time depending on your connection speed. Once it has finished, an Installed tag should appear next to the Bridge library. You can close the library manager.



You can now find the new library available in the Include Library menu. If you want to add your own library open a new issue on Github.

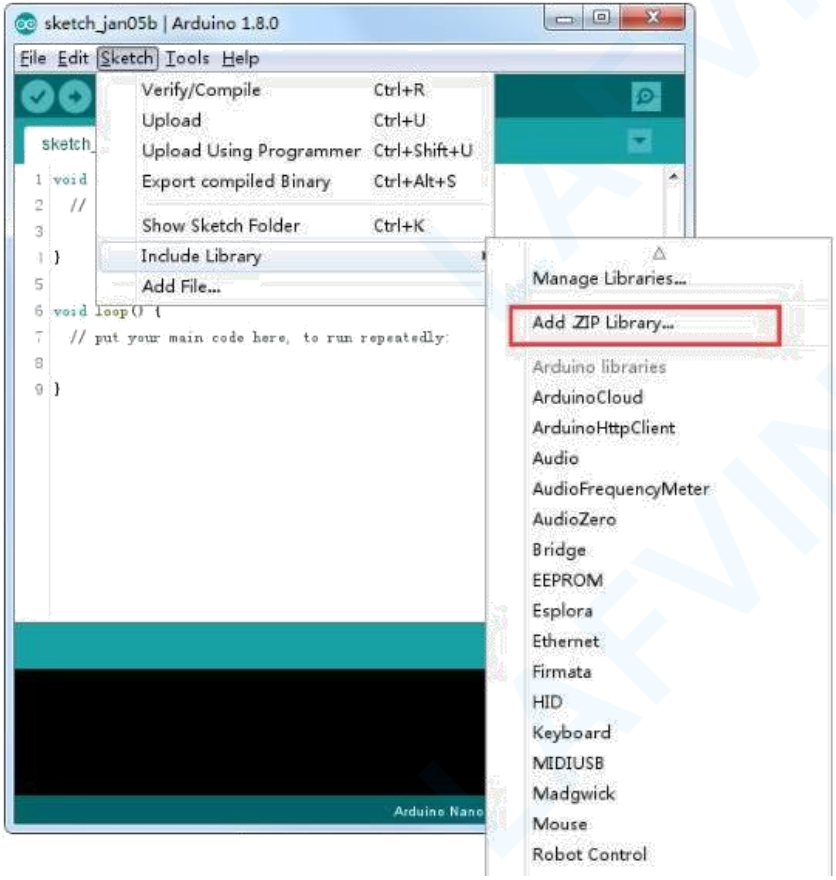
Importing a .zip Library

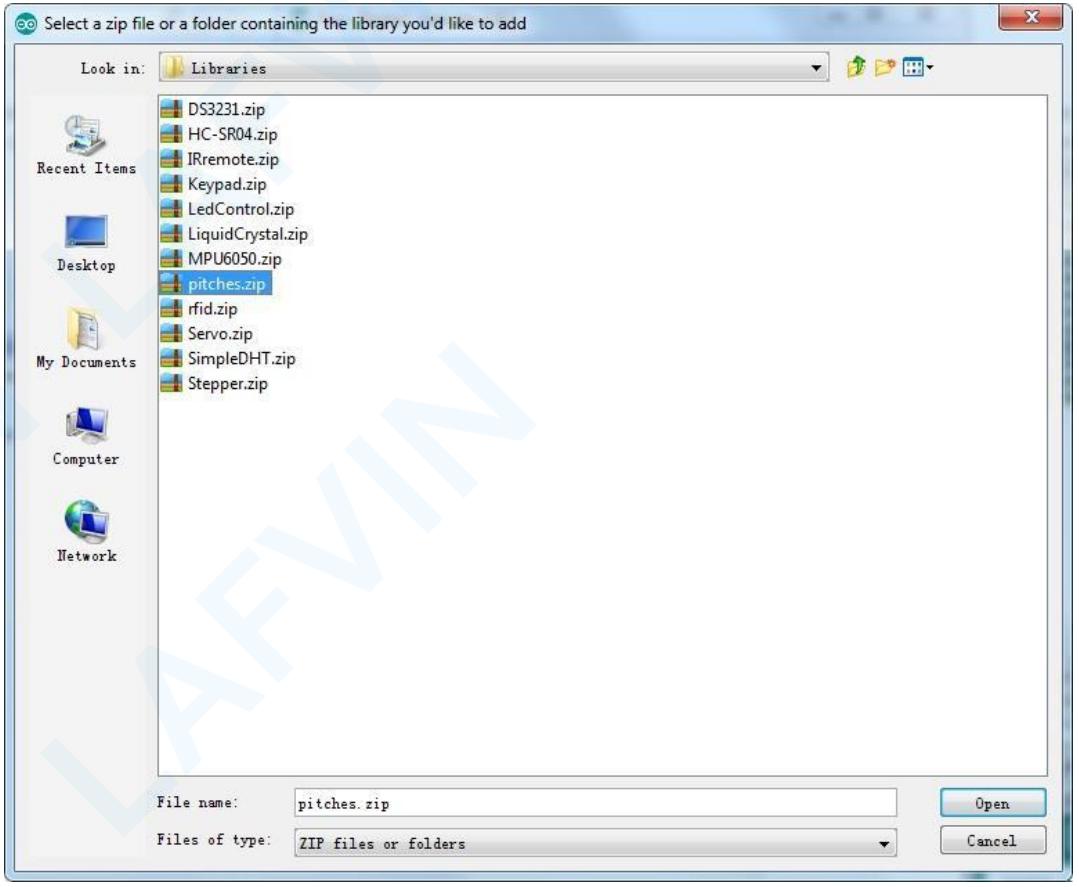
Libraries are often distributed as a ZIP file or folder. The name of the folder is the name of the library. Inside the folder will be a .cpp file, a .h file and often a keywords.txt file, examples folder, and other files required by the library. Starting with version 1.0.5, you can install 3rd party

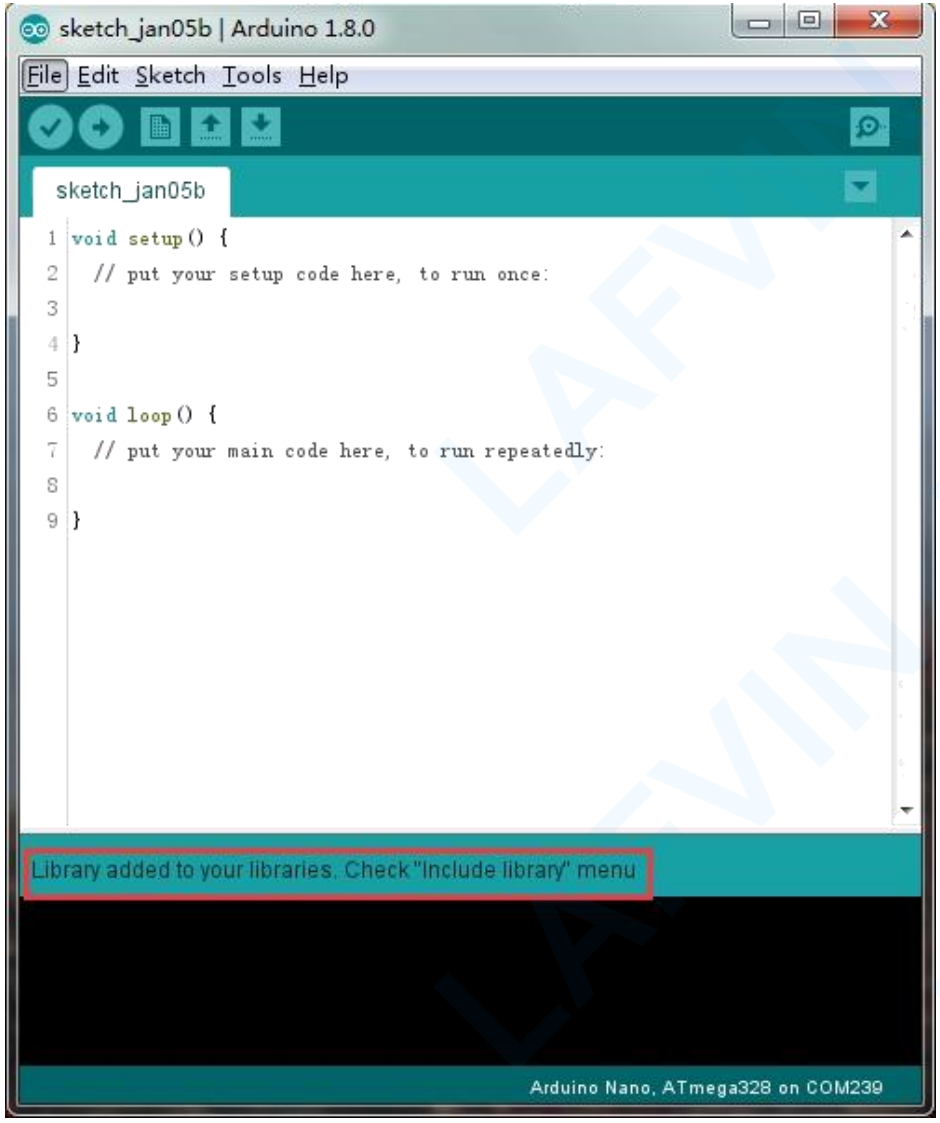
libraries in the IDE. Do not unzip the downloaded library, leave it as is.

In the Arduino IDE, navigate to Sketch > Include Library. At the top of the drop down list, select the option to "Add .ZIP Library"
You will be prompted to select the library you would like to add. Navigate to the .zip

file's location and open it.






sketch_jan05b | Arduino 1.8.0
File Edit Sketch Tools Help
sketch_jan05b
1 void setup() {
2 // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7 // put your main code here, to run repeatedly:
8
9 }
Library added to your libraries. Check "Include library" menu
Arduino Nano, ATmega328 on COM239

Return to the Sketch > Import Library menu. You should now see the library at the bottom of the drop-down menu. It is ready to be used in your sketch. The zip file will have been expanded in the libraries folder in your Arduino sketches directory. NB: the Library will be available to use in sketches, but examples for the library will not be exposed in the File > Examples until after the IDE has restarted.

Those two are the most common approaches. MAC and Linux systems can be handled likewise. The manual installation to be introduced below as an alternative may be seldom used and users with no needs may skip it.

Manual installation

To install the library, first quit the Arduino application. Then uncompress the ZIP file containing the library. For example, if you're installing a library called

"ArduinoParty", uncompress ArduinoParty.zip. It should contain a folder calledArduinoParty, with files like ArduinoParty.cpp and ArduinoParty.h inside. (If the .cpp and .h files aren't in a folder, you'll need to create one. In this case, you'd make a folder called "ArduinoParty" and move into it all the files that were in the ZIP file, like ArduinoParty.cpp and ArduinoParty.h.)

Drag the ArduinoParty folder into this folder (your libraries folder). Under Windows, it will likely be called "My Documents\Arduino\libraries". For Mac users, it will likely be called "Documents/Arduino/libraries". On Linux, it will be the "libraries" folder in your sketchbook.

Your Arduino library folder should now look like this (on Windows): My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.cpp My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.h My Documents\Arduino\libraries\ArduinoParty\examples

or like this (on Mac and Linux): Documents/Arduino/libraries/ArduinoParty/ArduinoParty.cpp Documents/Arduino/libraries/ArduinoParty/ArduinoParty.h Documents/Arduino/libraries/ArduinoParty/examples

....

There may be more files than just the .cpp and .h files, just make sure they're all there. (The library won't work if you put the .cpp and .h files directly into the libraries folder or if they're nested in an extra folder. For example: Documents\Arduino\libraries\ArduinoParty.cpp and Documents\Arduino\libraries\ArduinoParty\ArduinoParty\ArduinoParty.cpp won't work.)

Restart the Arduino application. Make sure the new library appears in the Sketch->Import Library menu item of the software. That's it! You've installed a library!

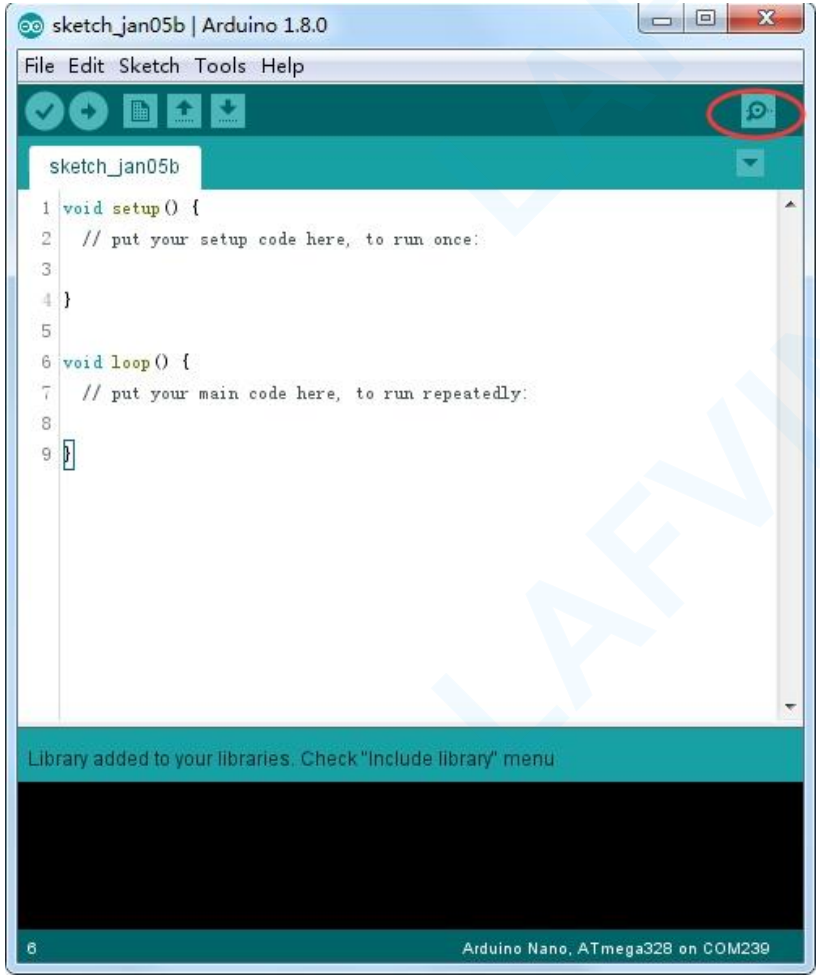
Arduino Serial Monitor (Windows, Mac, Linux)

The Arduino Integrated Development Environment (IDE) is the software side of the Arduino platform. And, because using a terminal is such a big part of working with

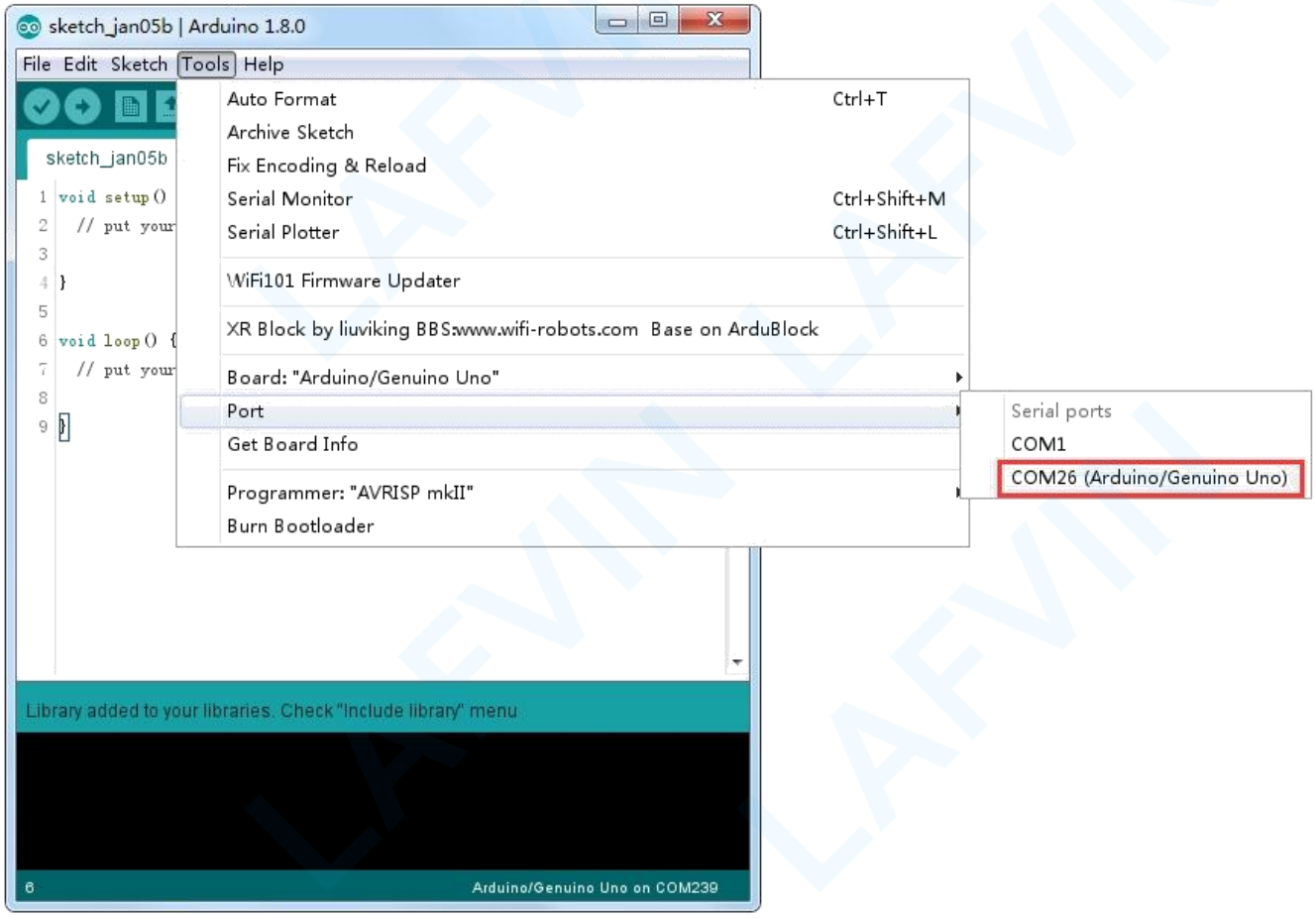
Arduinos and other microcontrollers, they decided to include a serial terminal with the software. Within the Arduino environment, this is called the Serial Monitor.

Making a Connection

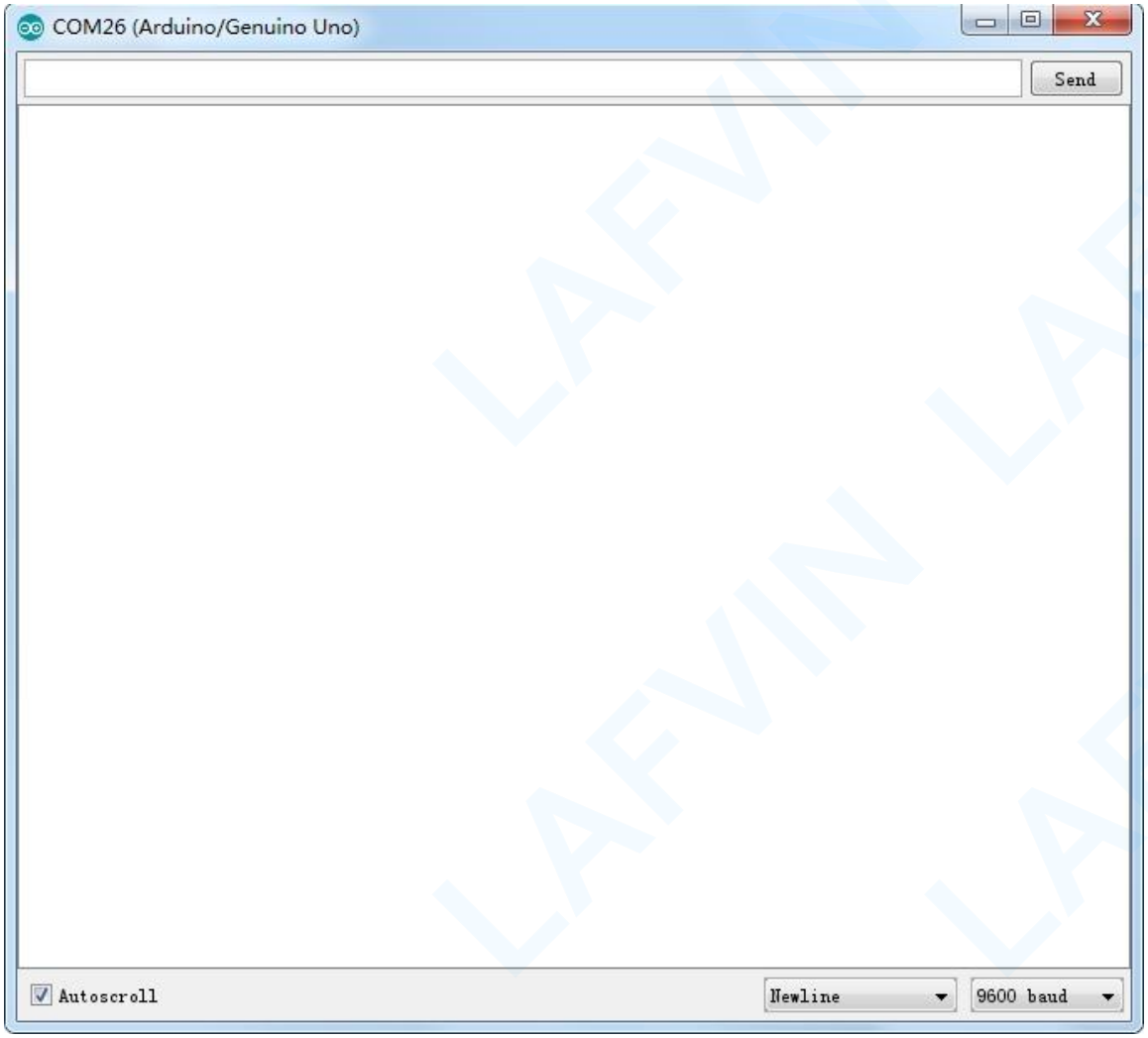
Serial monitor comes with any and all version of the Arduino IDE. To open it, simply click the Serial Monitor icon.



Selecting which port to open in the Serial Monitor is the same as selecting a port for uploading Arduino code. Go to Tools -> Serial Port, and select the correct port. Tips: Choose the same COM port that you have in Device Manager.

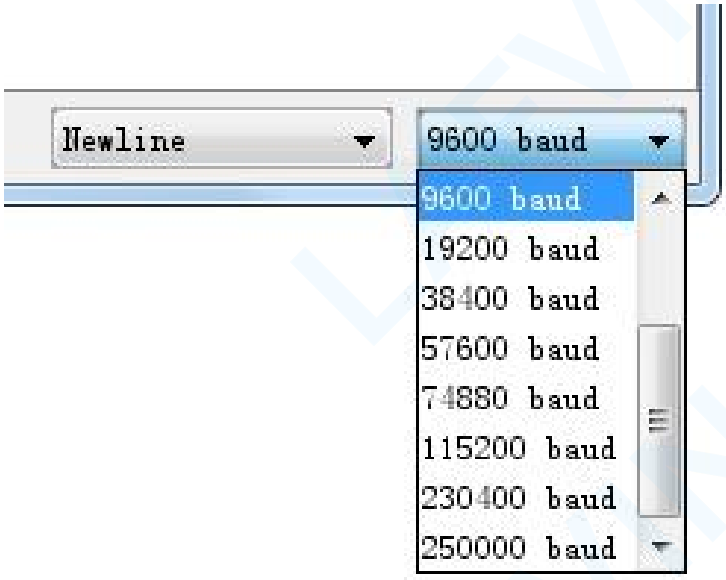


Once open, you should see something like this:



Settings

The Serial Monitor has limited settings, but enough to handle most of your serial communication needs. The first setting you can alter is the baud rate. Click on the baud rate drop-down menu to select the correct baud rate. (9600 baud)



Last, you can set the terminal to Autoscroll or not by checking the box in the bottom left corner.

Pros

The Serial Monitor is a great quick and easy way to establish a serial connection with your Arduino. If you’re already working in the Arduino IDE, there’s really no need to open up a separate terminal to display data.

Cons

The lack of settings leaves much to be desired in the Serial Monitor, and, for advanced serial communications, it may not do the trick.

Lesson 3 Blink

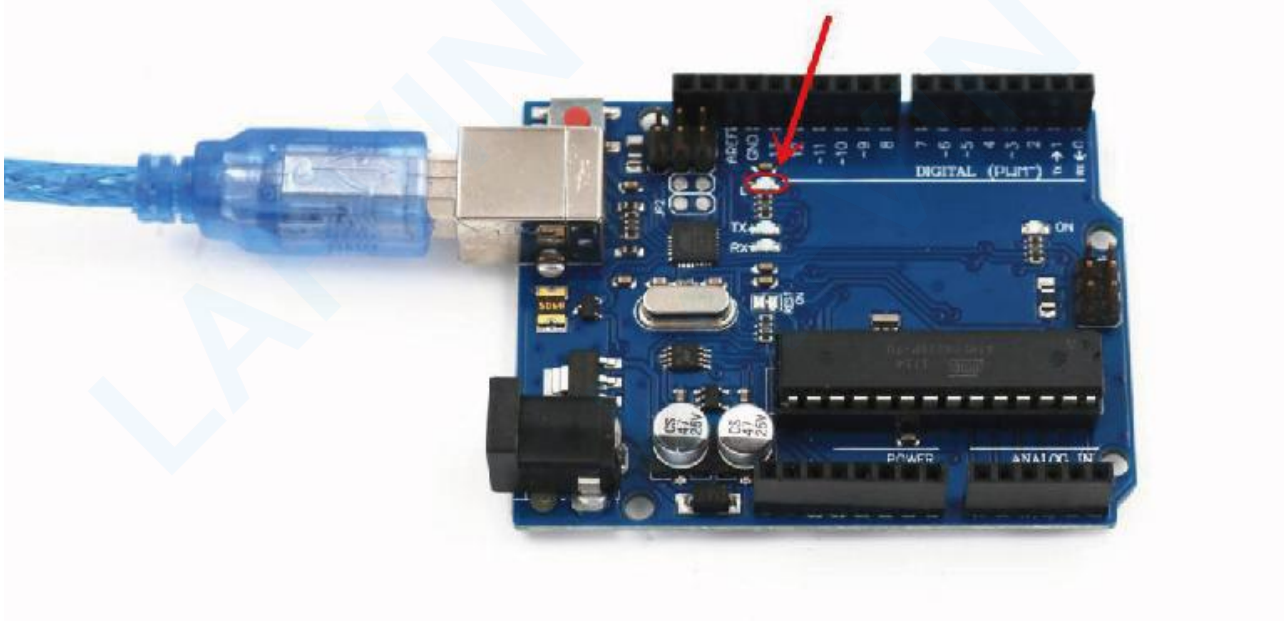
About this lesson:

In this lesson, you will learn how to program your UNO R3 controller board to blink the Arduino's built-in LED, and how to download programs by basic steps.

Principle

The UNO R3 board has rows of connectors along both sides that are used to connect to several electronic devices and plug-in 'shields' that extends its capability.

It also has a single LED that you can control from your sketches. This LED is built onto the UNO R3 board and is often referred to as the 'L' LED as this is how it is labeled on the board.



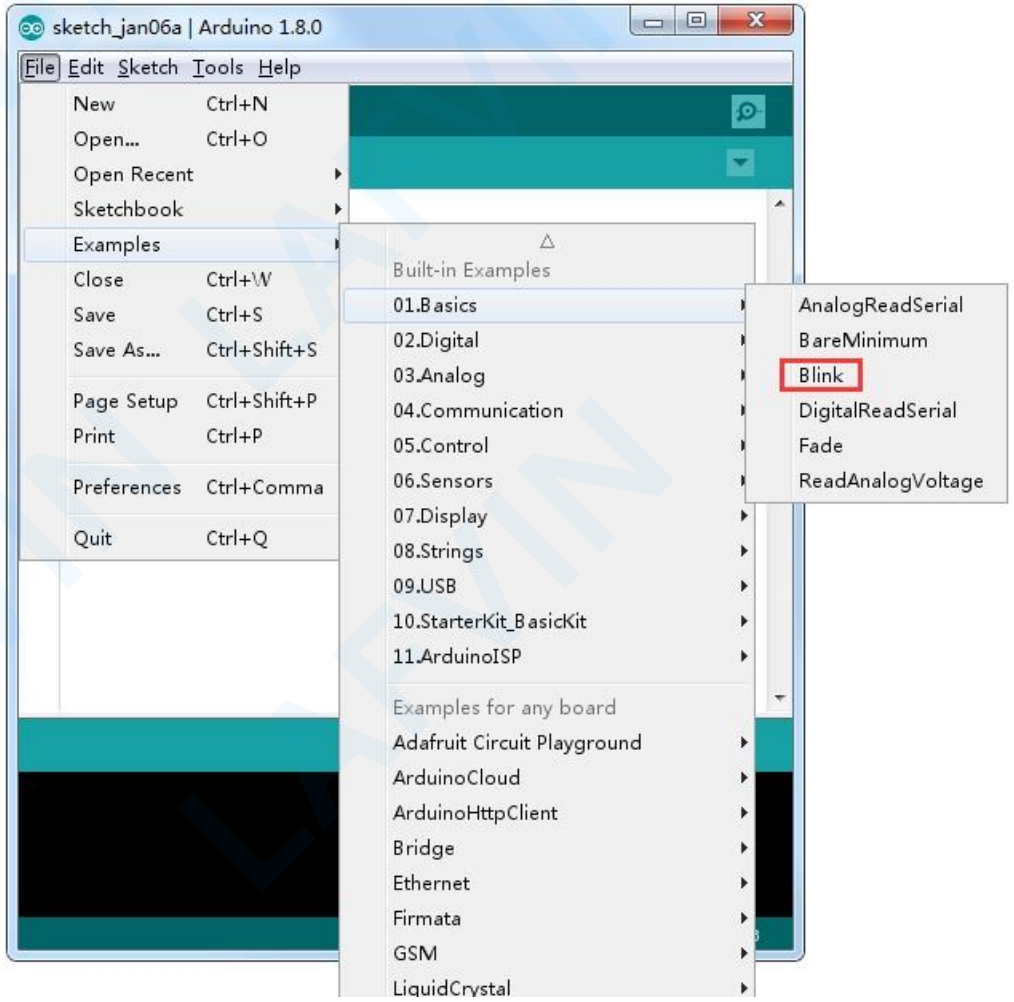
You may find that your UNO R3 board's 'L' LED already blinks when you connect it to a USB plug. This is because the boards are generally shipped with the 'Blink' sketch pre-installed.

In this lesson, we will reprogram the UNO R3 board with our own Blink sketch and then change the rate at which it blinks.

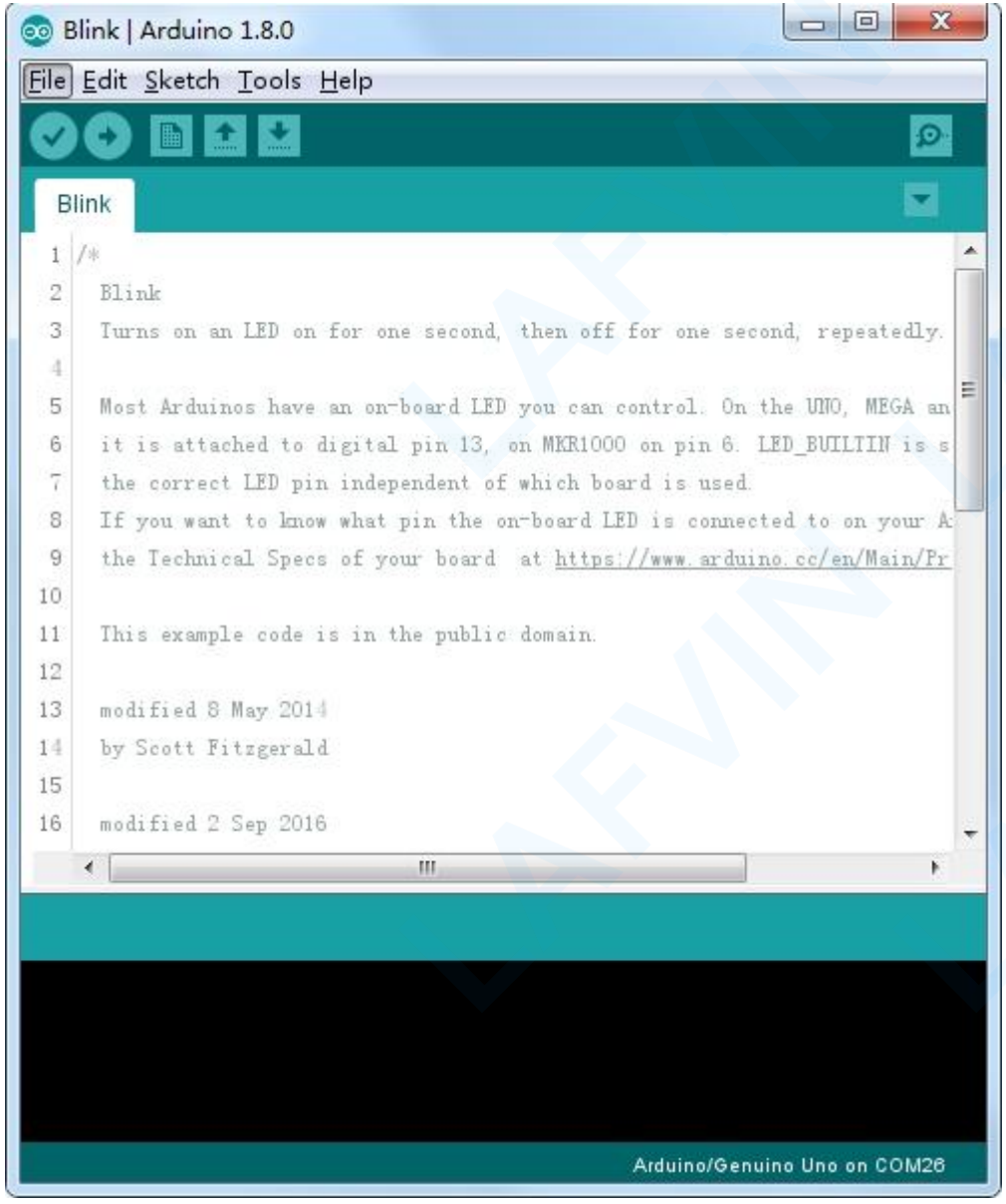
In Lesson 0, you set up your Arduino IDE and made sure that you could find the right serial port for it to connect to your UNO R3 board. The time has now come to put that connection to the test and program your UNO R3 board.

The Arduino IDE includes a large collection of example sketches that you can load up and use. This includes an example sketch for making the 'L' LED blink.

Load the 'Blink' sketch that you will find in the IDE's menu system under File > Examples > 01.Basics



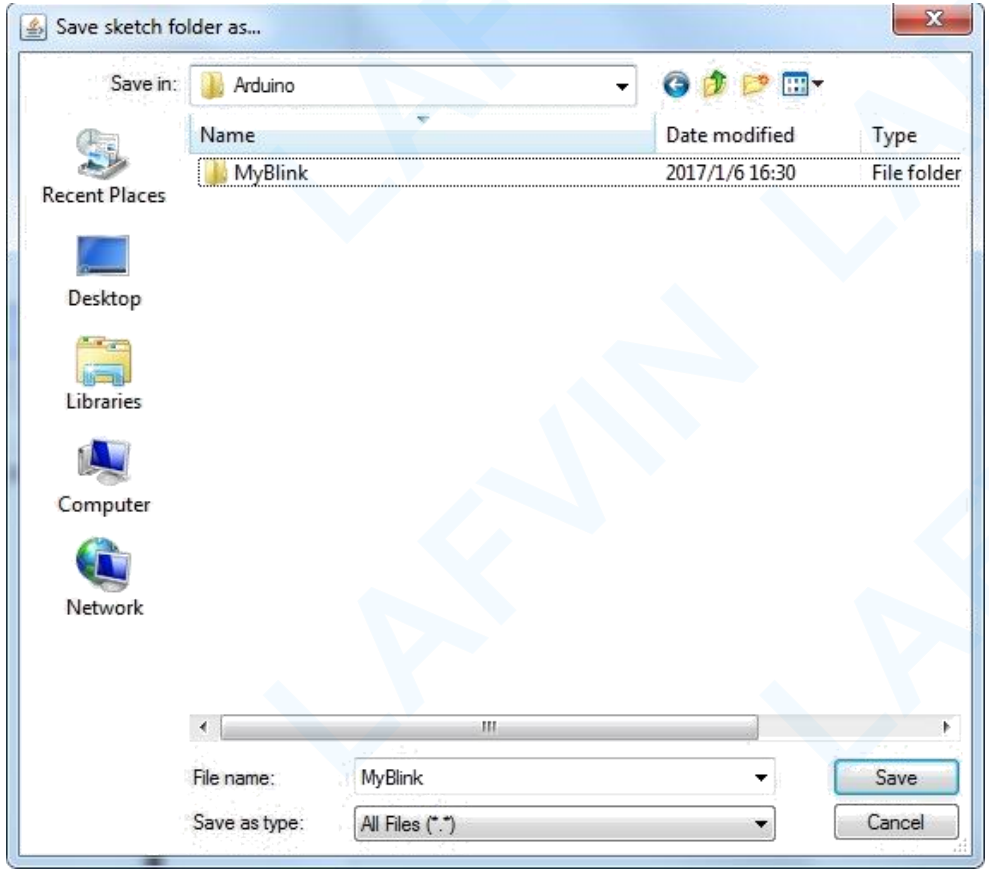
When the sketch window opens, enlarge it so that you can see the entire sketch in the window.



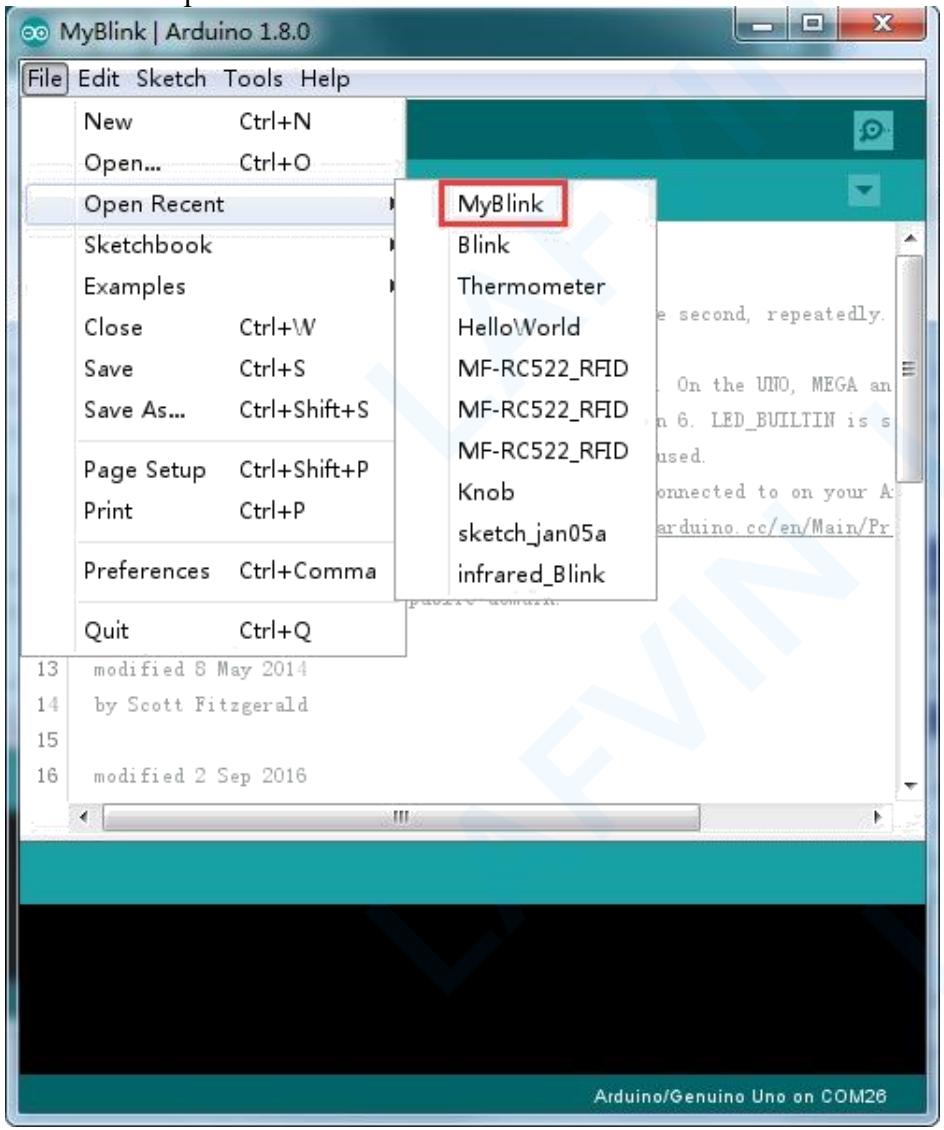
The example sketches included with the Arduino IDE are 'read-only'. That is, you can upload them to an UNO R3 board, but if you change them, you cannot save them as the same file.

Since we are going to change this sketch, the first thing you need to do is save your own copy.

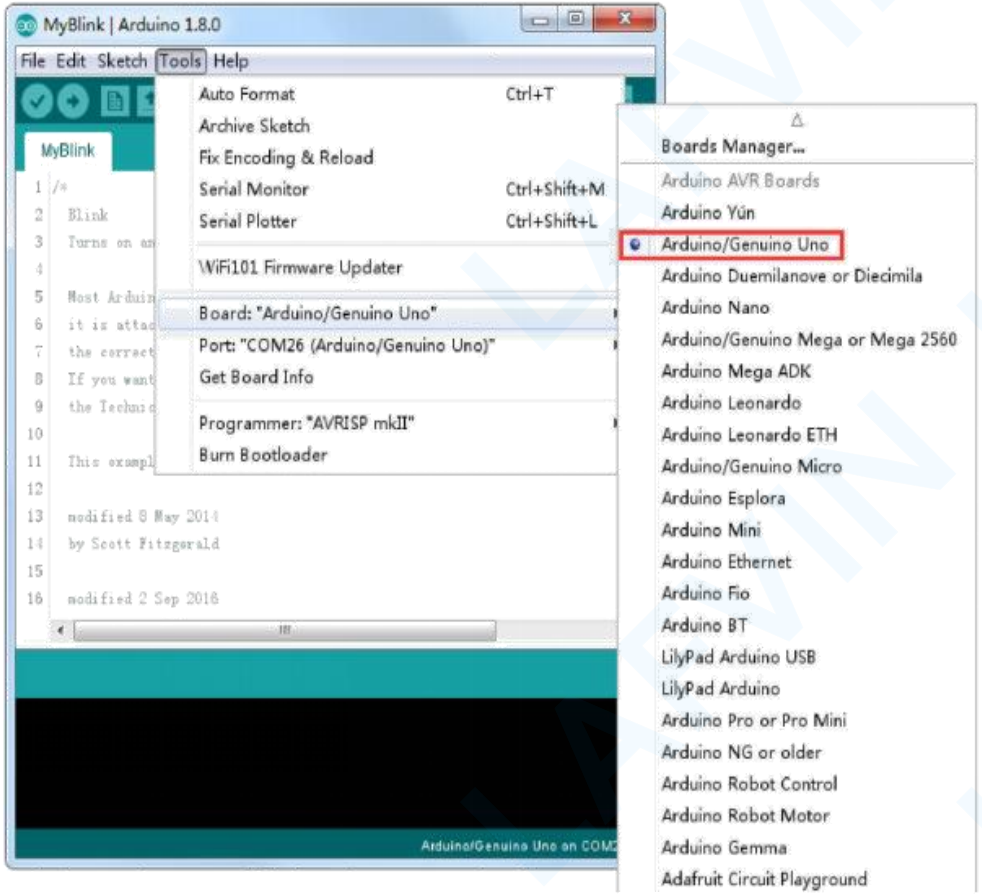
From the File menu on the Arduino IDE, select 'Save As..' and then save the sketch with the name 'MyBlink'.

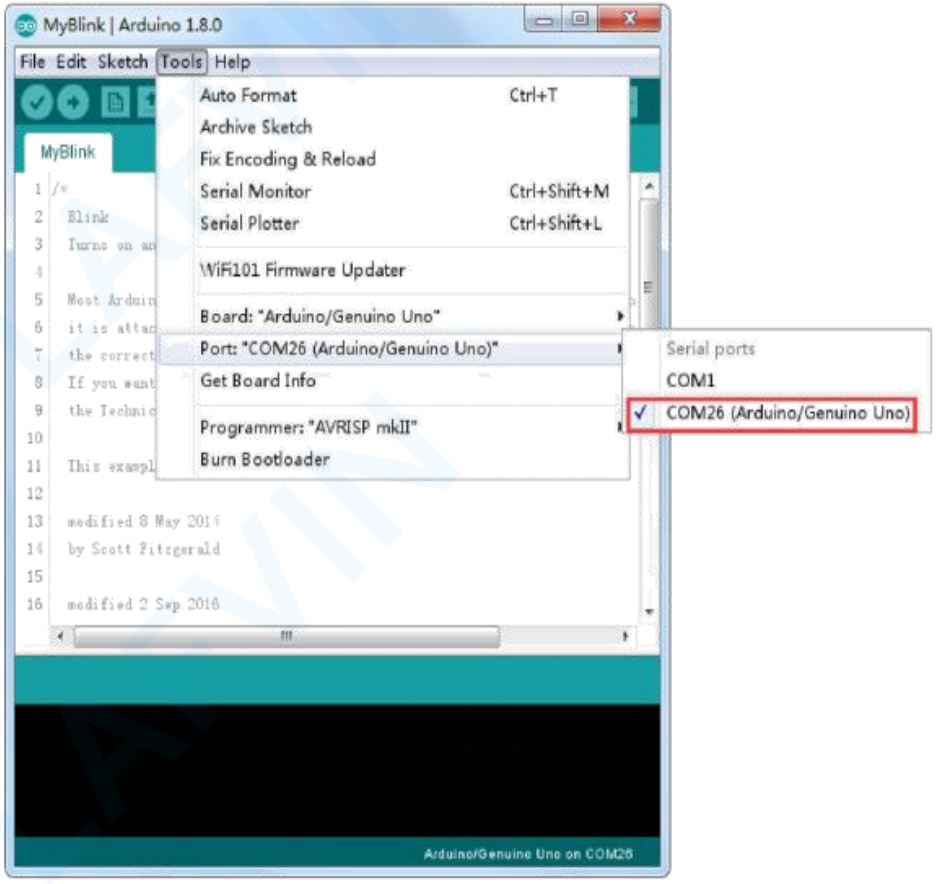


You have saved your copy of 'Blink' in your sketchbook. This means that if you ever want to find it again, you can just open it using the File > Sketchbook menu option.



Attach your Arduino board to your computer with the USB cable and check that the 'Board Type' and 'Serial Port' are set correctly.





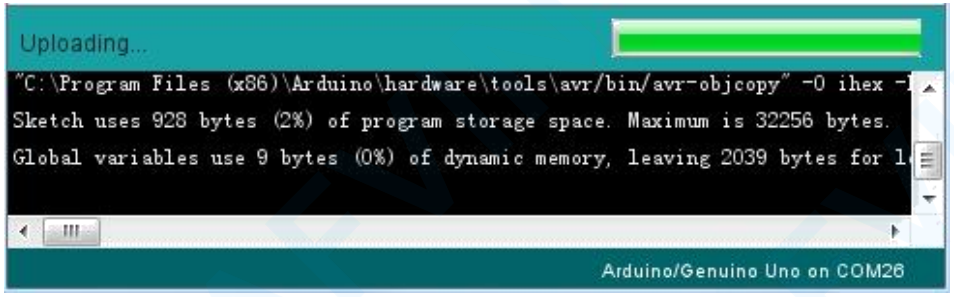
Note: The Board Type and Serial Port here are not necessarily the same as shown in picture. If you are using 2560, then you will have to choose Mega 2560 as the Board Type, other choices can be made in the same manner. And the Serial Port displayed for everyone is different, despite COM 26 chosen here, it could be COM3 or COM4 on your computer. A right COM port is supposed to be COMX (arduino XXX), which is by the certification criteria.

The Arduino IDE will show you the current settings for board at the bottom of the window.

Click on the 'Upload' button. The second button from the left on the toolbar.

If you watch the status area of the IDE, you will see a progress bar and a series of messages. At first, it will say 'Compiling Sketch...'. This converts the sketch into a format suitable for uploading to the board.

Next, the status will change to 'Uploading'. At this point, the LEDs on the Arduino should start to flicker as the sketch is transferred.



Finally, the staus will change to 'Done'.

The other message tells us that the sketch is using 928 bytes of the 32,256 bytes available.After the 'Compiling Sketch..' stage you could get the following error message:

It can mean that your board is not connected at all, or the drivers have not been installed (if necessary) or that the wrong serial port is selected.

If you encounter this, go back to Lesson 0 and check your installation.

Once the upload has completed, the board should restart and start blinking. Open the code

Note that a huge part of this sketch is composed of comments. These are not actual program instructions; rather, they just explain how the program works. They are there for your benefit.

Everything between /* and */ at the top of the sketch is a block comment; it explains what the sketch is for.

gle line comments start with // and everything up until the end of that line is considered a comment.

The first line of code is: int led = 13;

As the comment above it explains, this is giving a name to the pin that the LED is attached to. This is 13 on most Arduinos, including the UNO and Leonardo.

Next, we have the 'setup' function. Again, as the comment says, this is executed when the reset button is pressed. It is also executed whenever the board resets for any reason, such as power first being applied to it, or after a sketch has been uploaded.

```
void setup() {
// initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}
```

Every Arduino sketch must have a 'setup' function, and the place where you might want to add instructions of your own is between the { and the }.

In this case, there is just one command there, which, as the comment states tells the Arduino board that we are going to use the LED pin as an output.

It is also mandatory for a sketch to have a 'loop' function. Unlike the 'setup' function that only runs once, after a reset, the 'loop' function will, after it has finished running its commands, immediately start again.

```
void loop() { digitalWrite(led,// turn the LED on (HIGH is the voltage
   HIGH);              delay(1000);level) // wait for a second
   digitalWrite(led, LOW); delay(1000);
}
```

Inside the loop function, the commands first of all turn the LED pin on (HIGH), then 'delay' for 1000 milliseconds (1 second), then turn the LED pin off and pause for another second.

You are now going to make your LED blink faster. As you might have guessed, the key to this lies in changing the parameter in () for the 'delay' command.

```
30 // the loop function runs over and over again forever
31 void loop() {
32   digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the volt
33   delay(500);                        // wait for a second
34   digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the vo
35   delay(500);                        // wait for a second
36 }
```

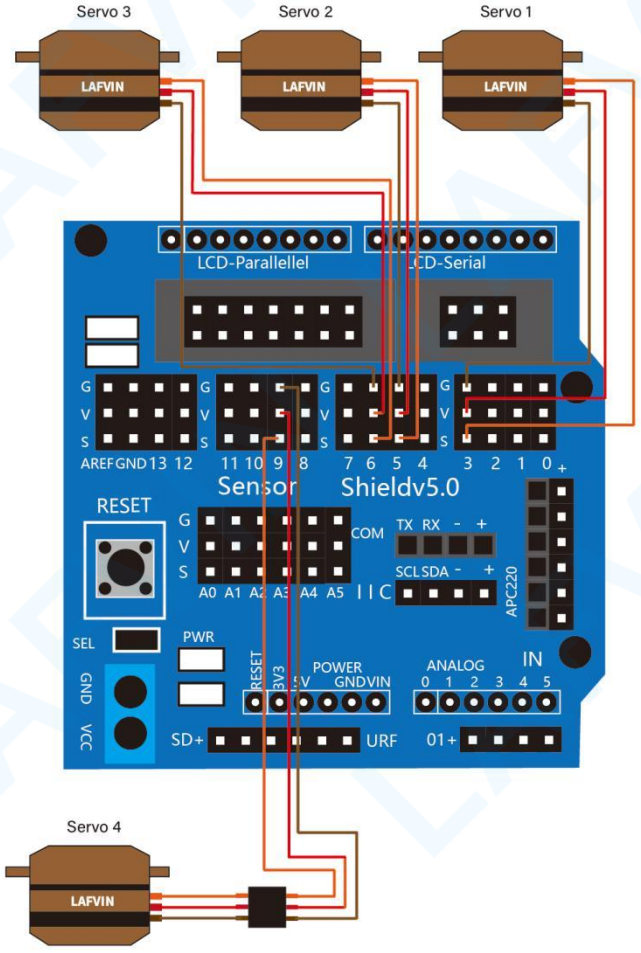
This delay period is in milliseconds, so if you want the LED to blink twice as fast, change the value from 1000 to 500. This would then pause for half a second each delay rather than a whole second.

Upload the sketch again and you should see the LED start to blink more quickly

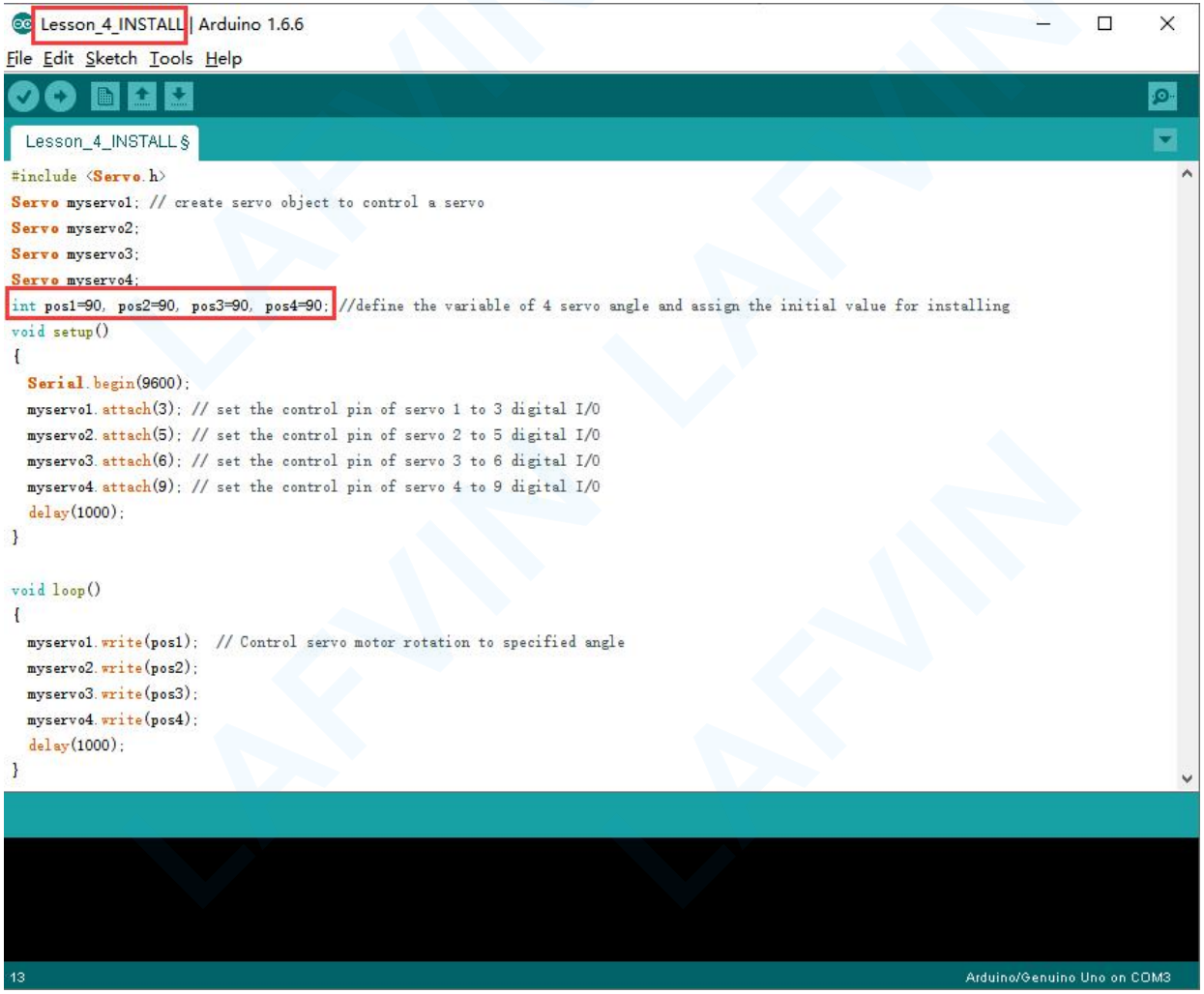
Lesson 4 Installation Method

Before installing the arm, we need to set the initial position of the four servo motors. After the angle is set, you can install it according to the vertical or parallel direction indicated by the picture.The angle calibration method of the servo motor is as follows:

First,First, install the Arduino sensor shield v5.0 board on the LAFVIN Arduino board and then connect servo1 /servo2 / servo3 / servo4 to 3/5/6/9pin.



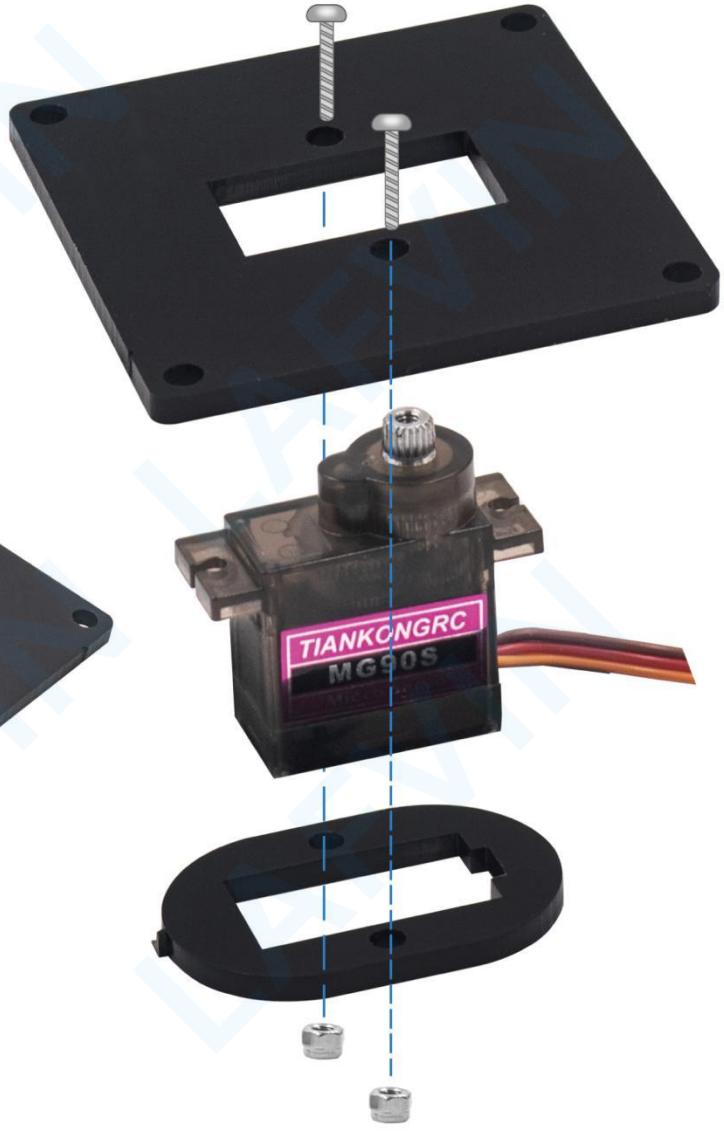
Connect the UNO R3 board and the computer with a USB cable. Open the program Lesson_4_INSTALL and upload the program to the board. Press the reset button, The servo motor will be transferred to the 90 degree position required for installation.Keep 90 degrees before installation, do not rotate servo motor gear at will.Now let's start installing the servo section.



Now the four servo motors are turned to the 90 degree position. Next, install the servo motor's swing arm according to the direction indicated in the picture. It may be vertical or parallel. Next we started to install the robot arm.

Step 2

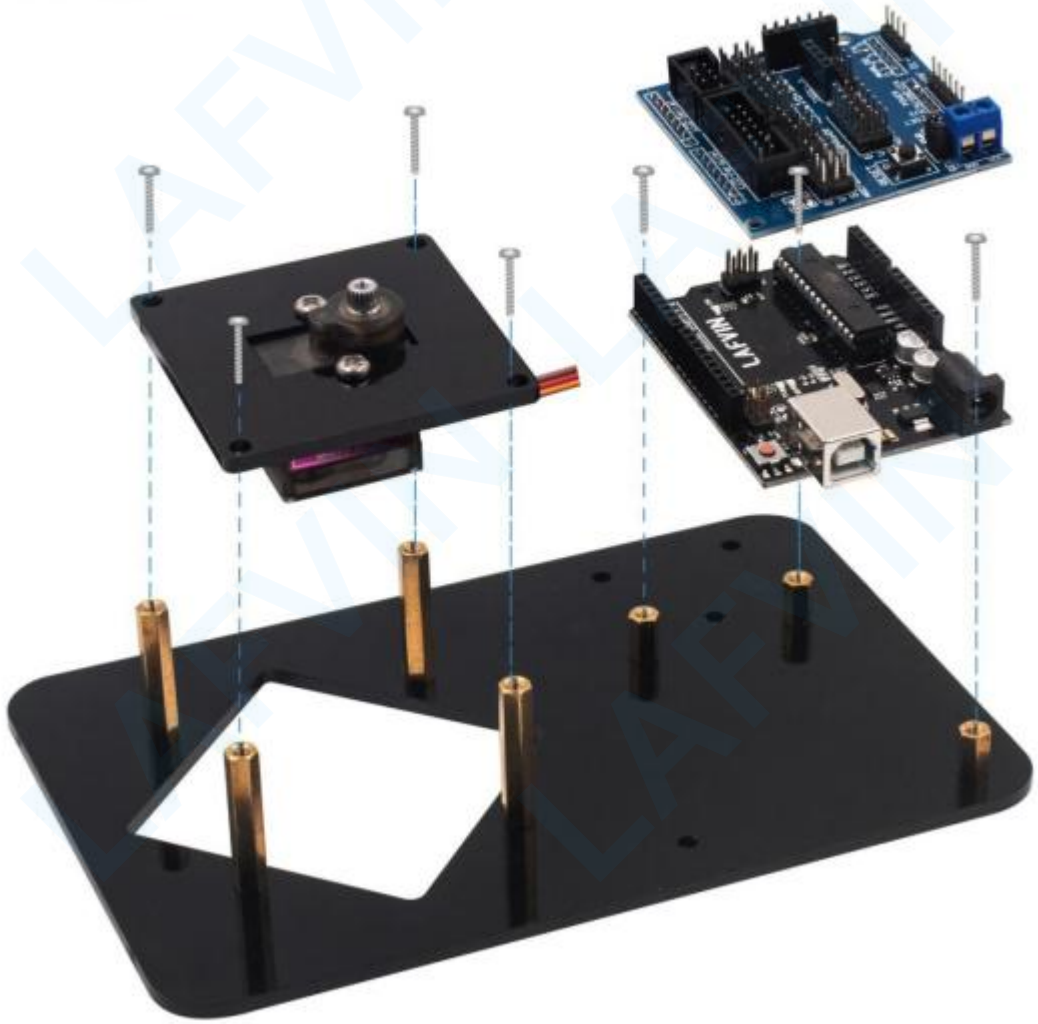
1. M3*12mm
2. M3 Lock Nut

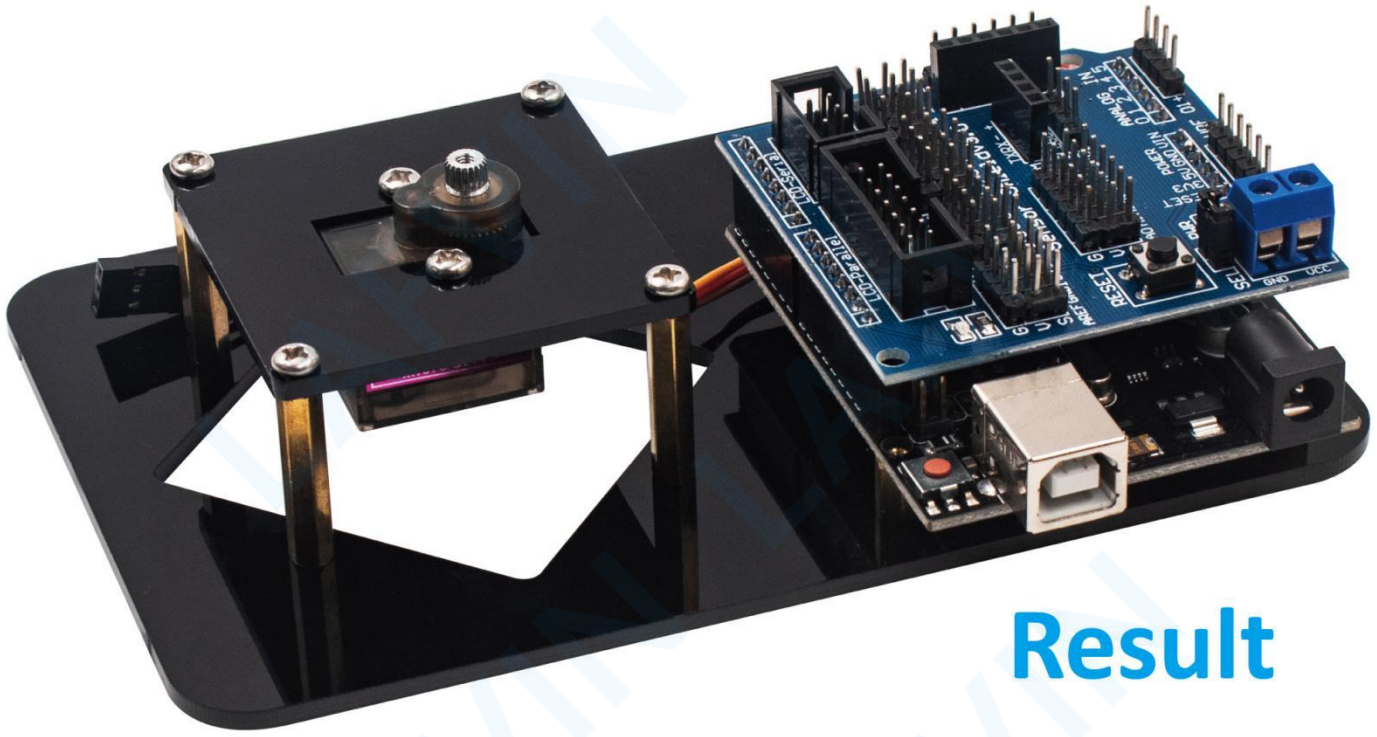


Result

Step 3

1. M3*6mm

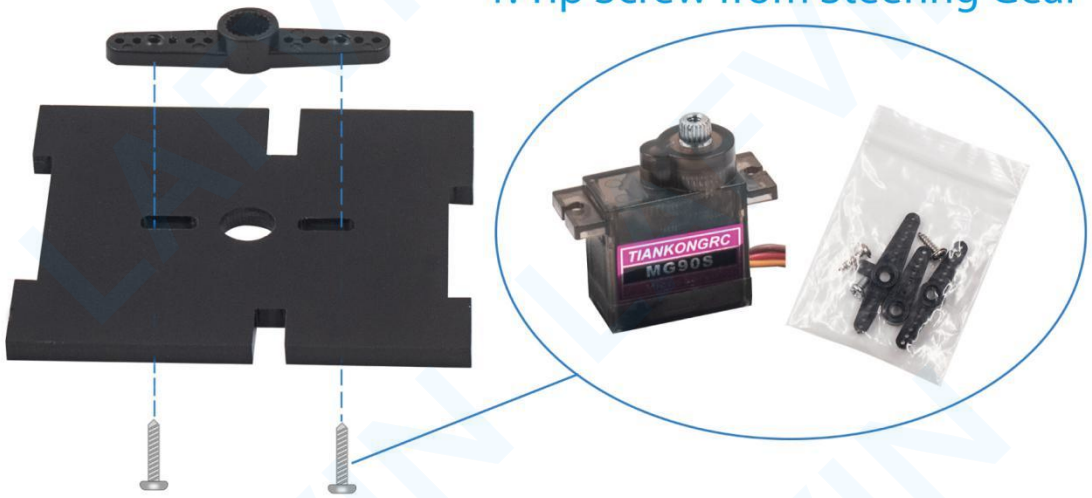




Result

Step 4

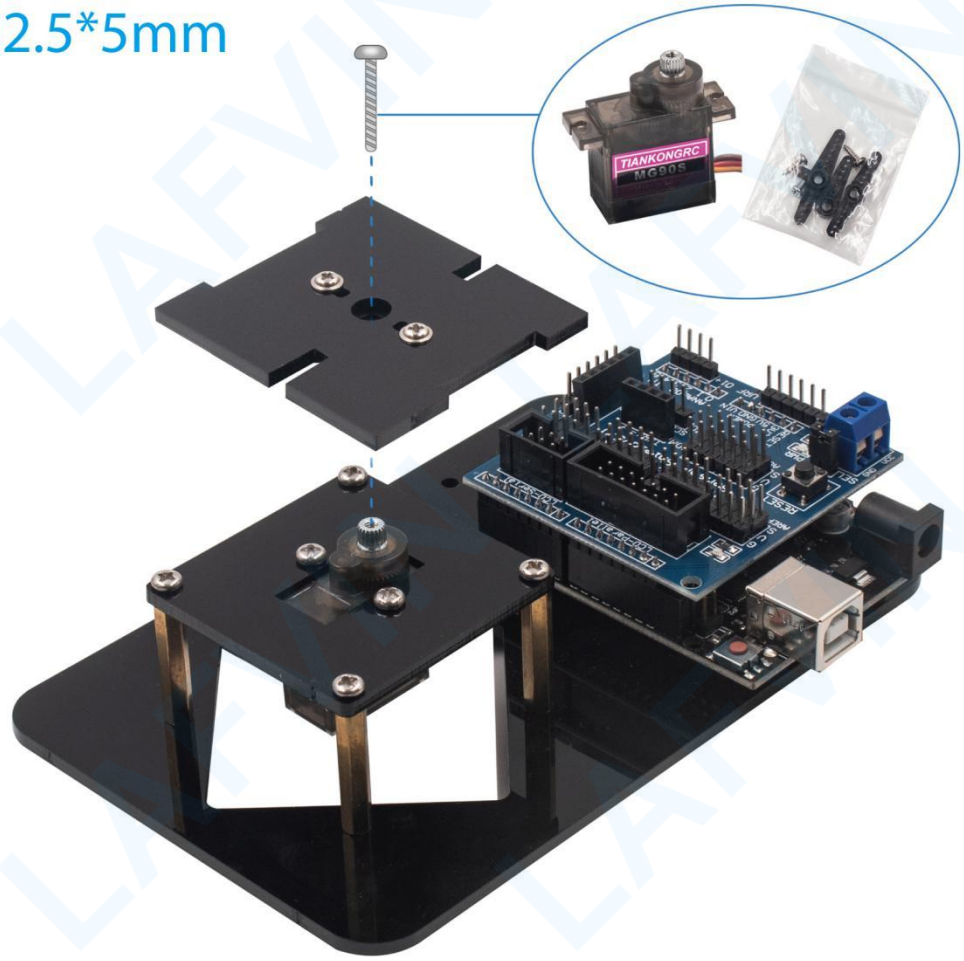
1. Tip Screw from Steering Gear

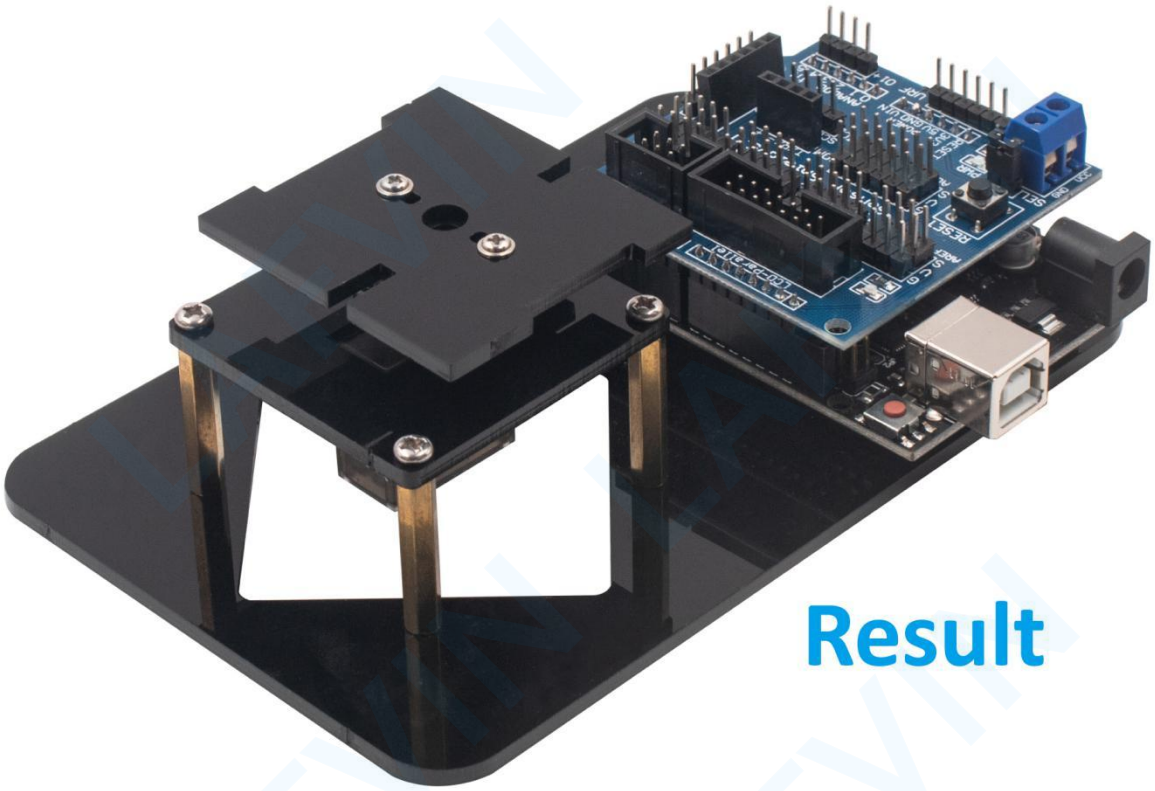


Result

Step 5

1. M2.5*5mm

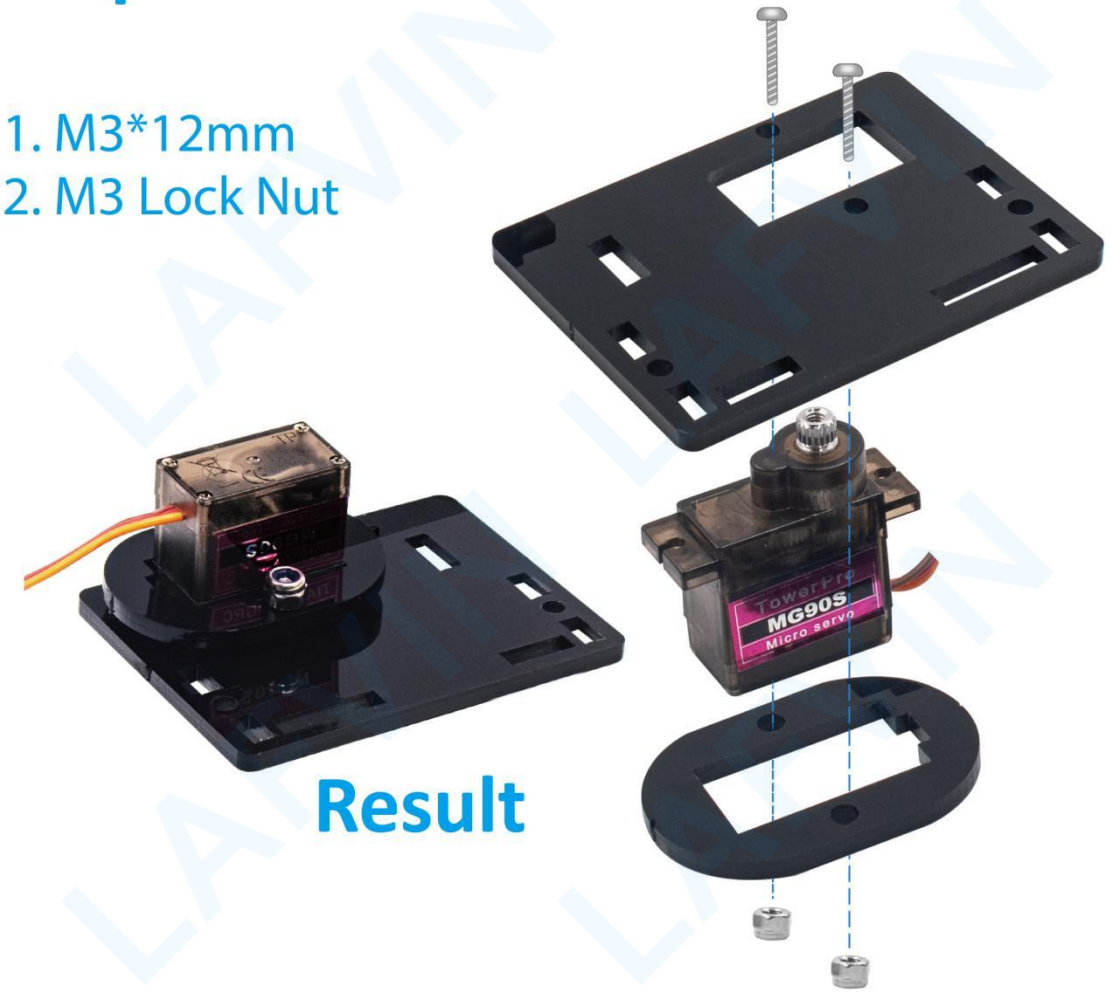




Result

Step 6

1. M3*12mm
2. M3 Lock Nut



Result

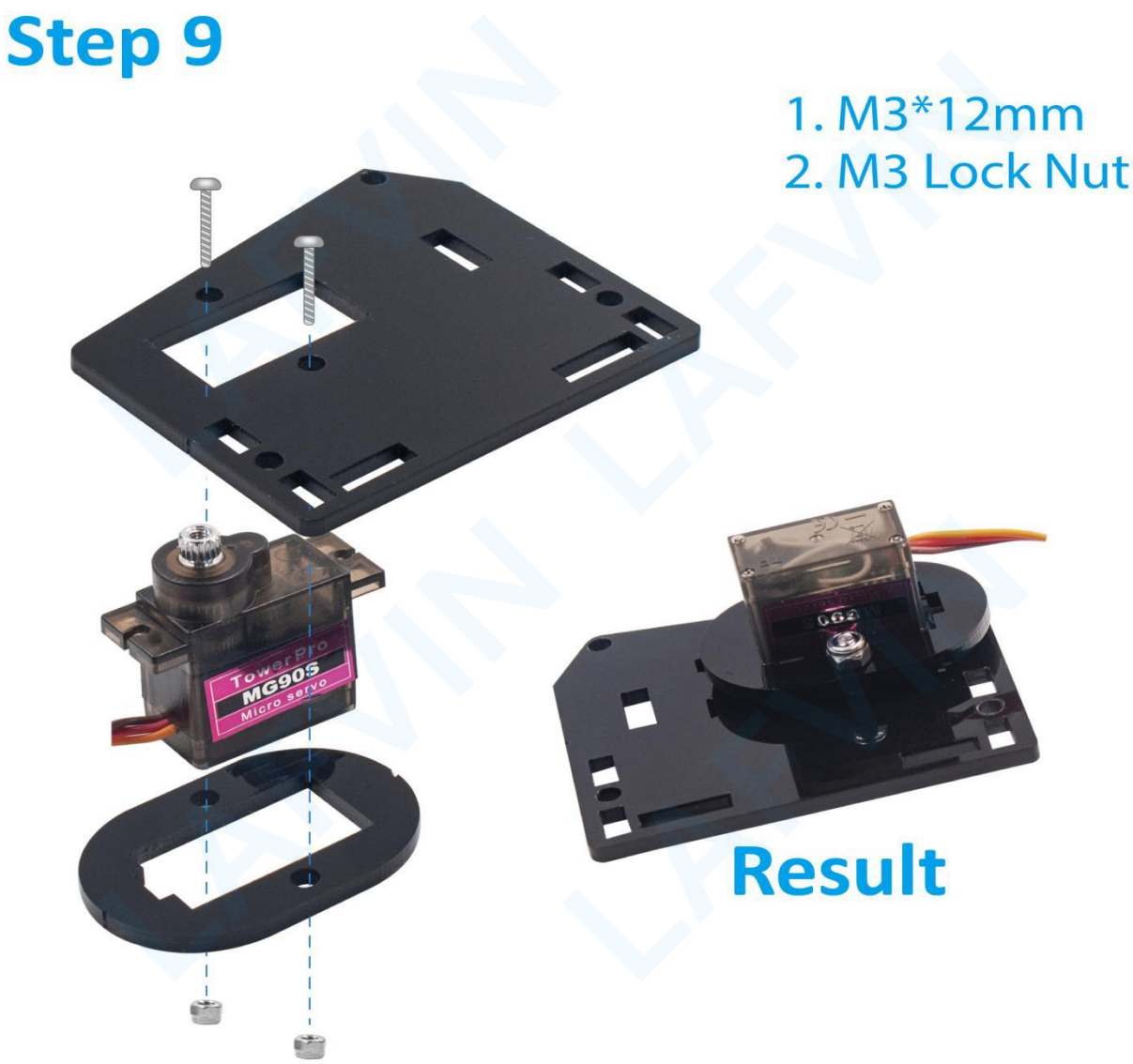
Step 7

Result

1. Tip Screw from Steering Gear

Step 8

1. M2.5*5mm


Step 9
1. M3*12mm
2. M3 Lock Nut
TowerPro
MG90S
Micro servo
Result

Step 10

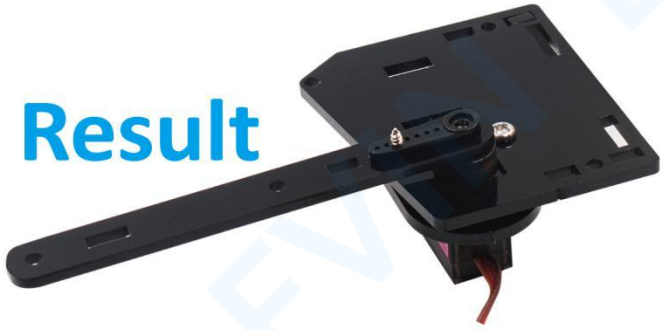
1. Tip Screw from Steering Gear

Result

Step 11

1. M2.5*5mm

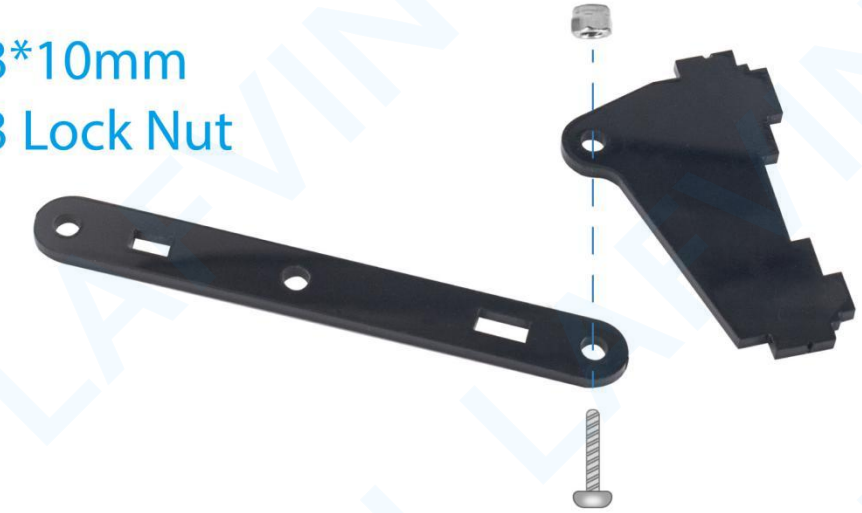
Result



Result-Top View

Step 12

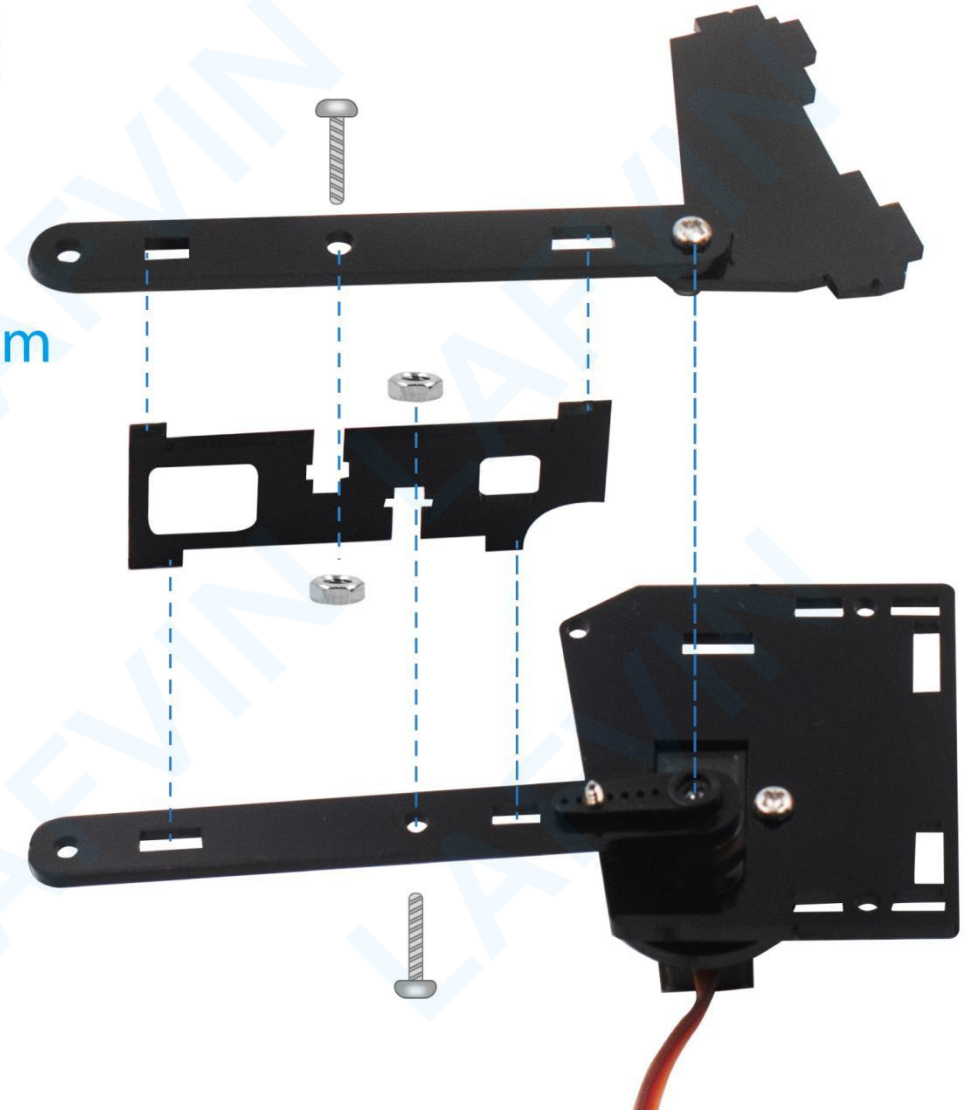
1. M3*10mm
2. M3 Lock Nut

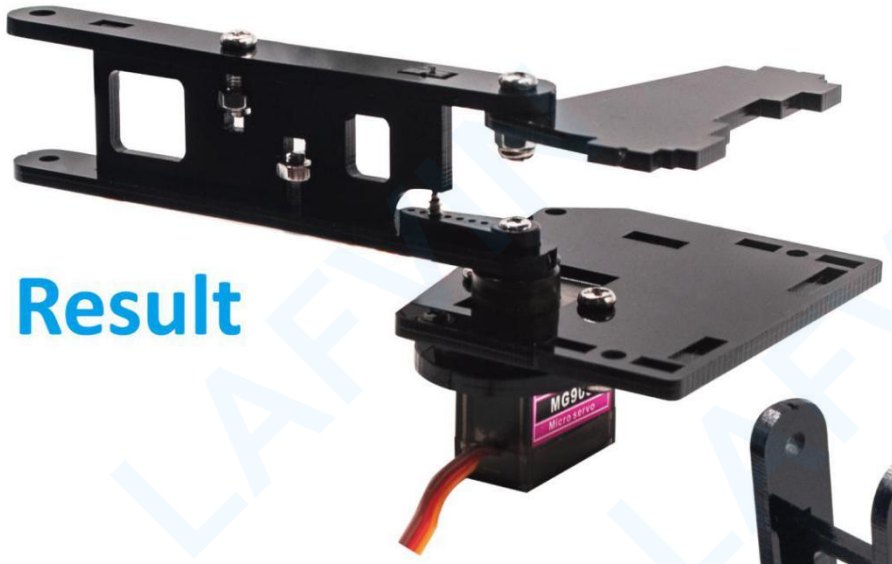


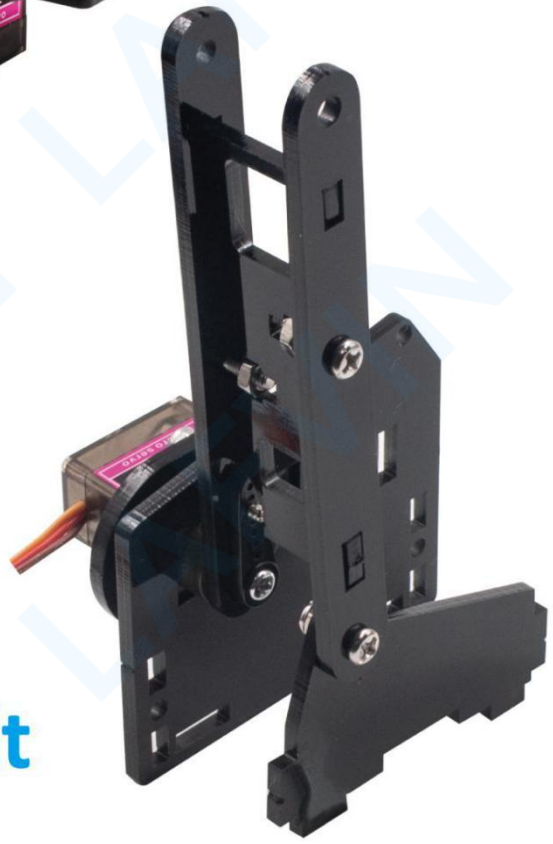
Result

Step 13

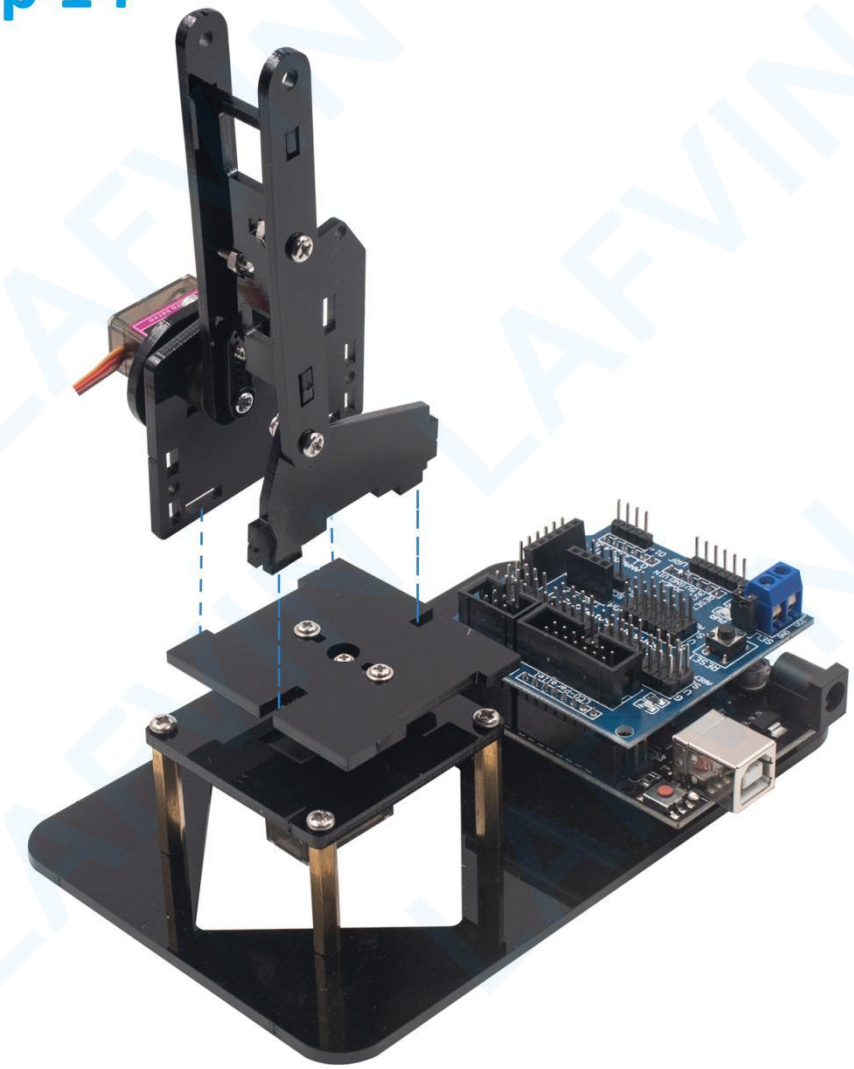
1. M3*10mm
2. M3 Nut

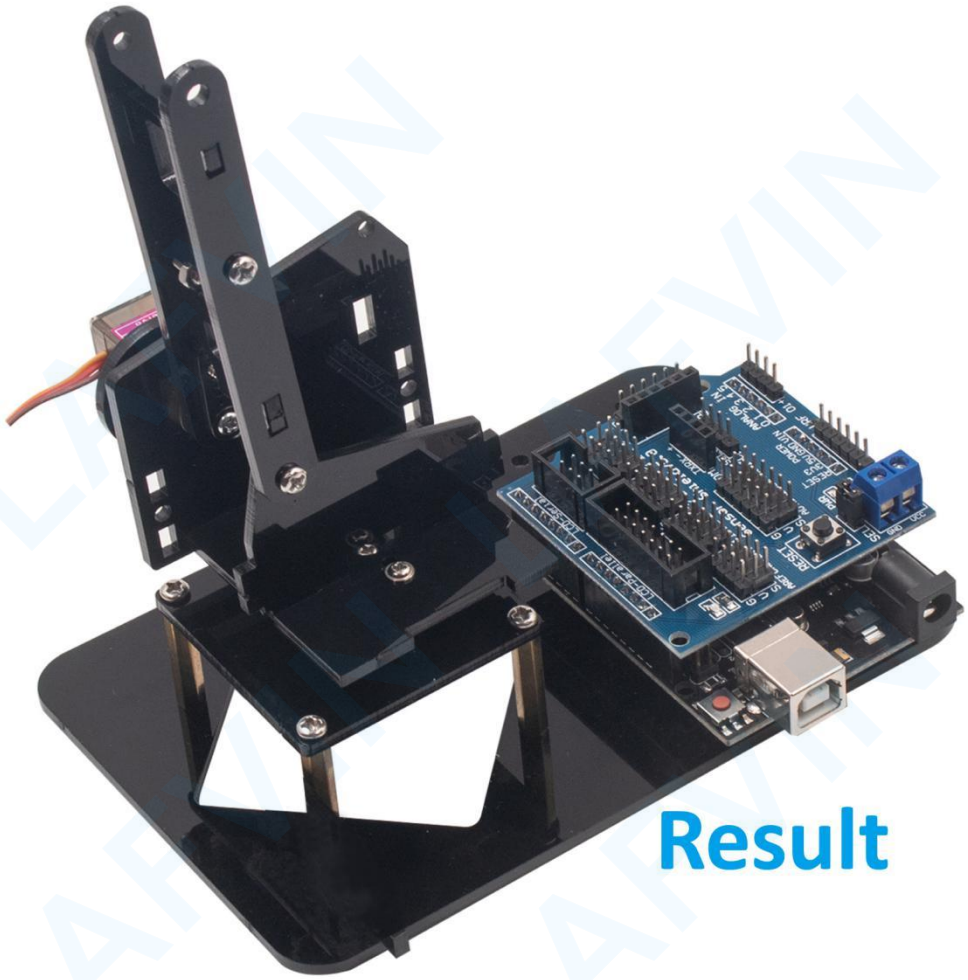



Result

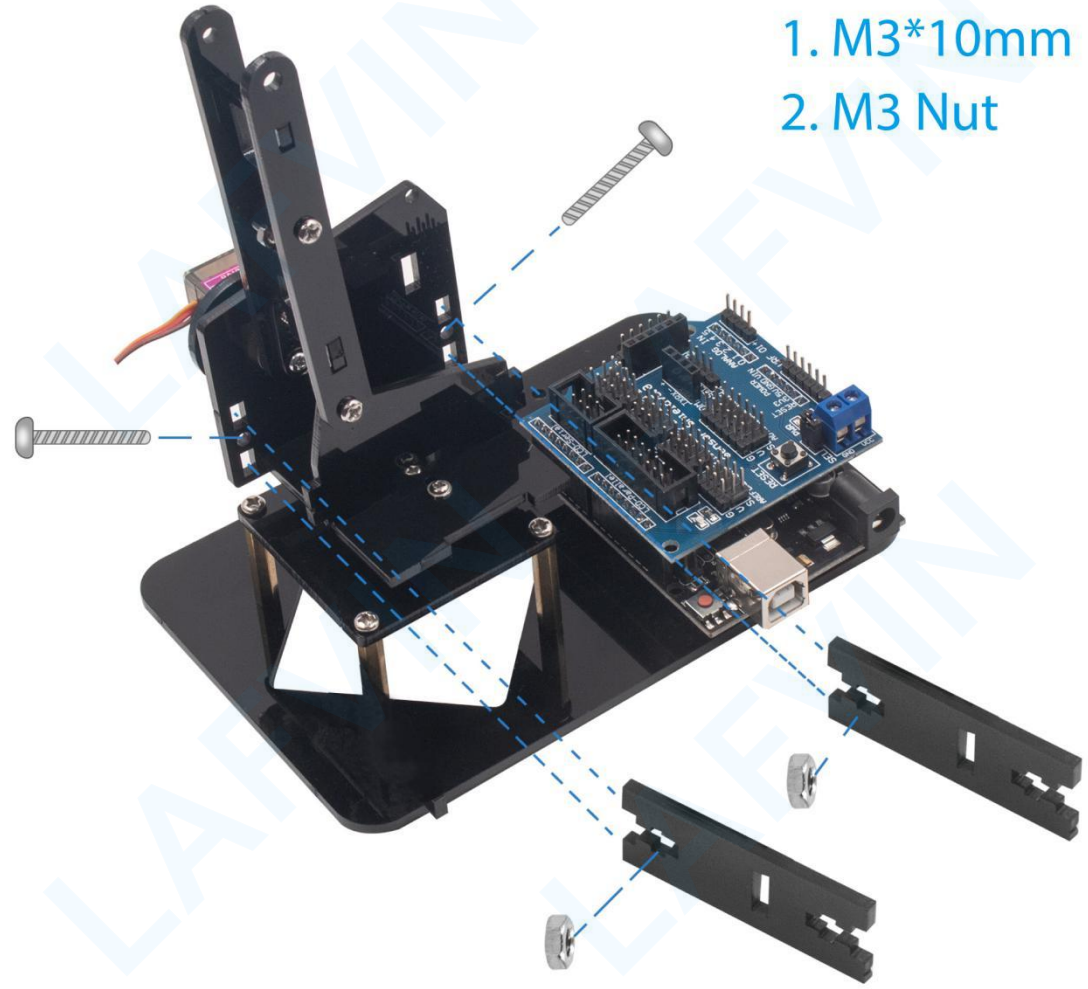

Result

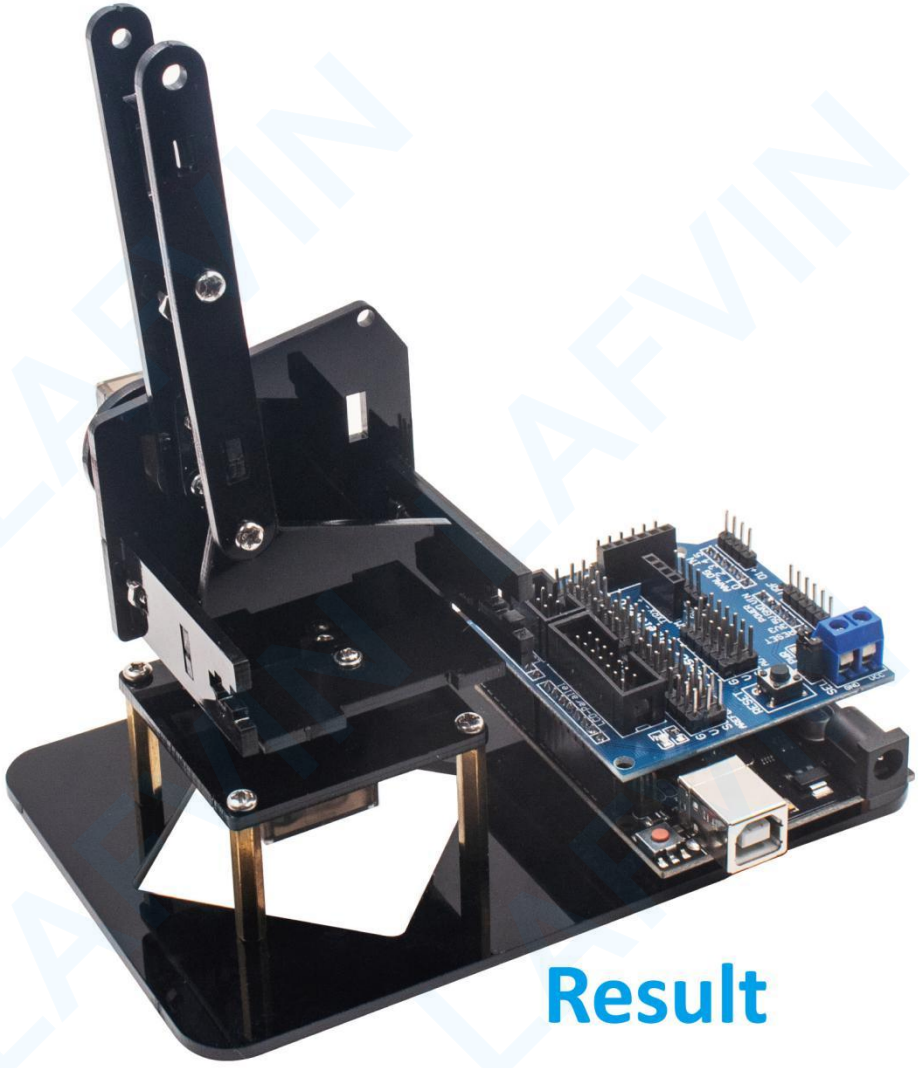
Step 14

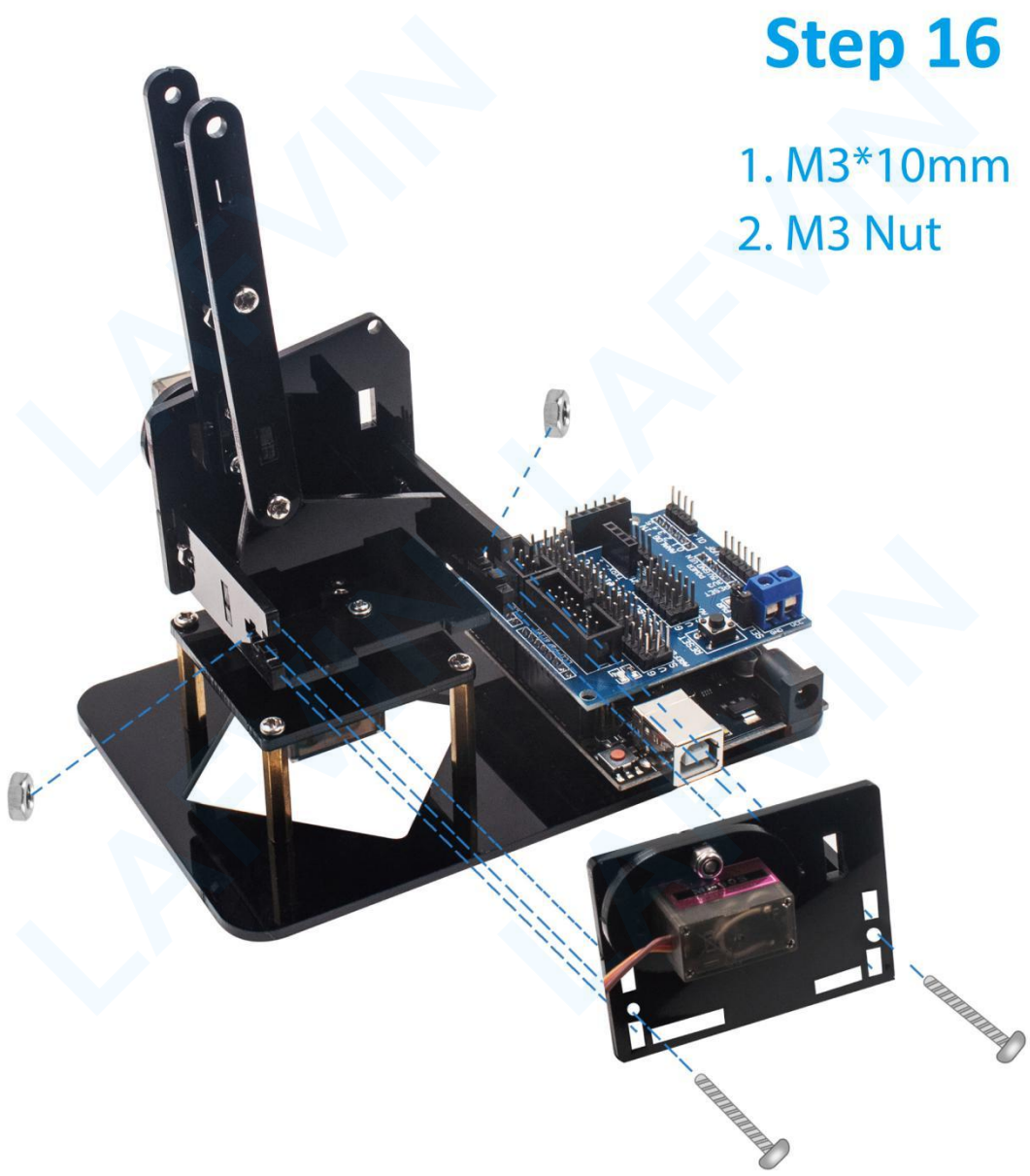


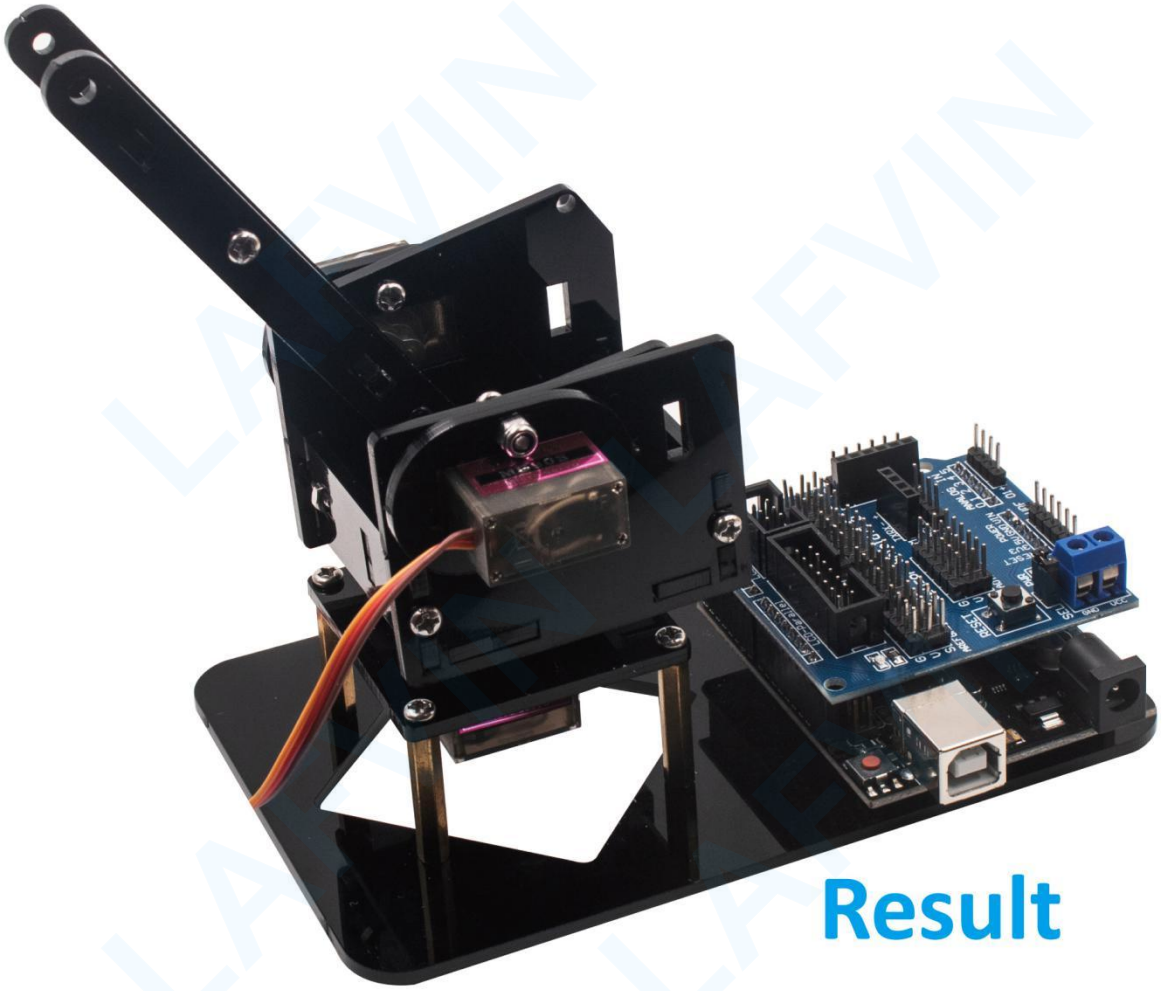

Result

Step 15




Result

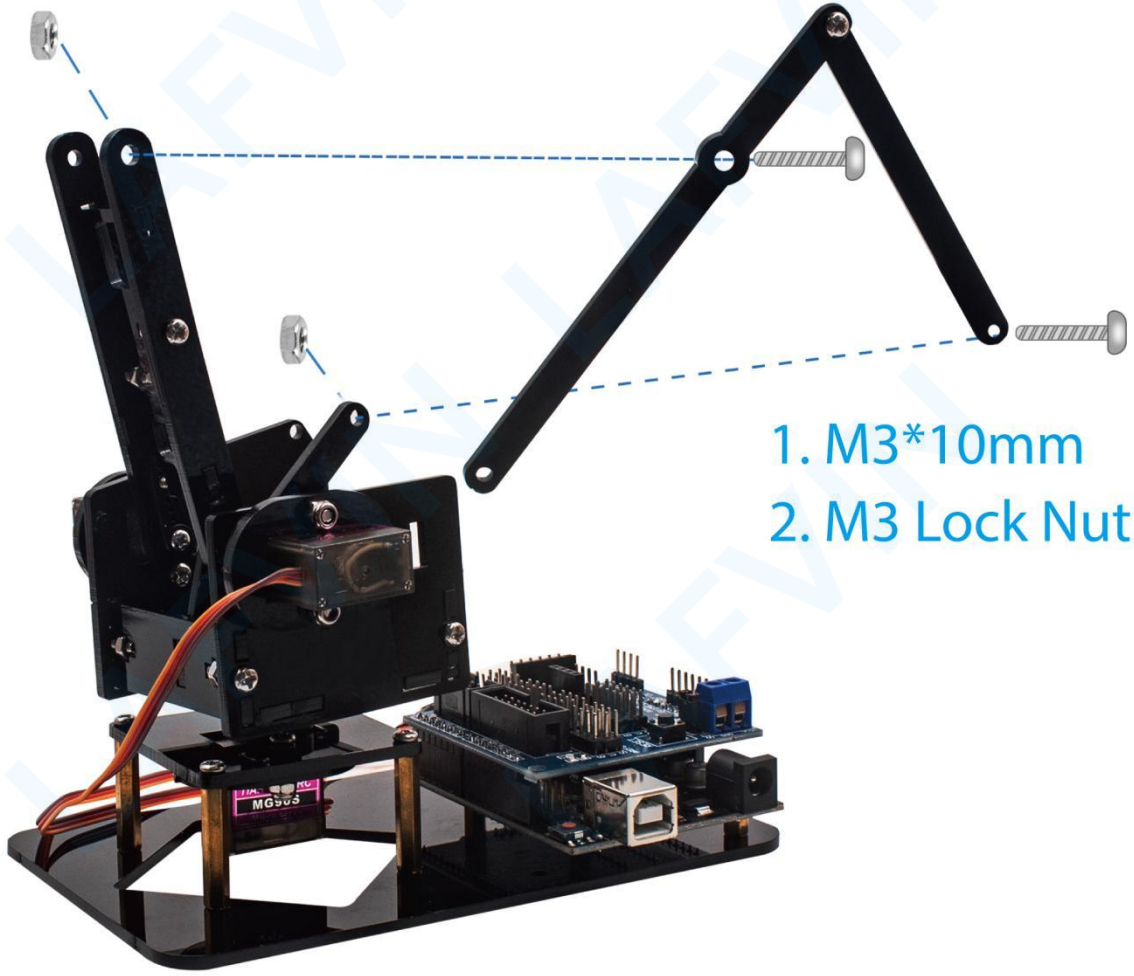

Step 16
1. M3*10mm
2. M3 Nut

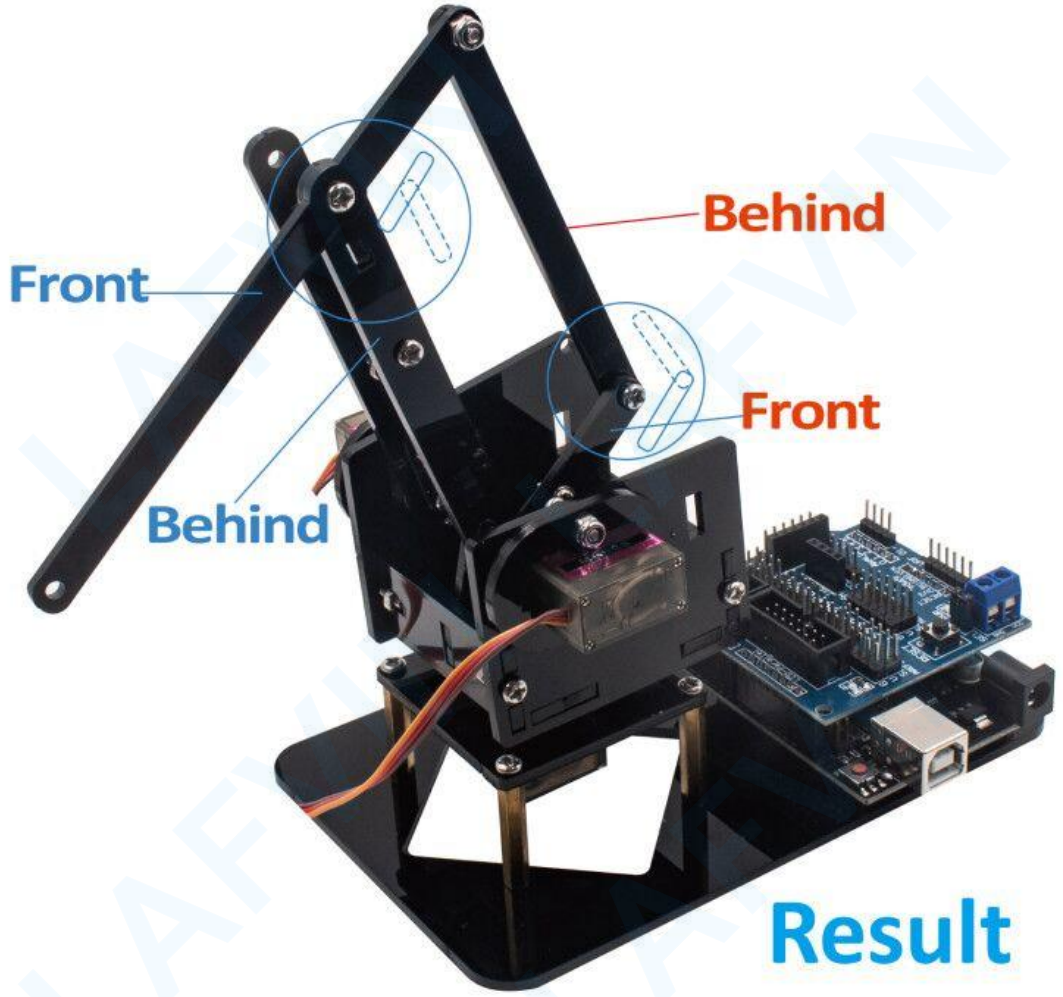

Result

Steps 17


Steps 18
1. M3*12mm
2. M3 Lock Nut
Result

Step 19

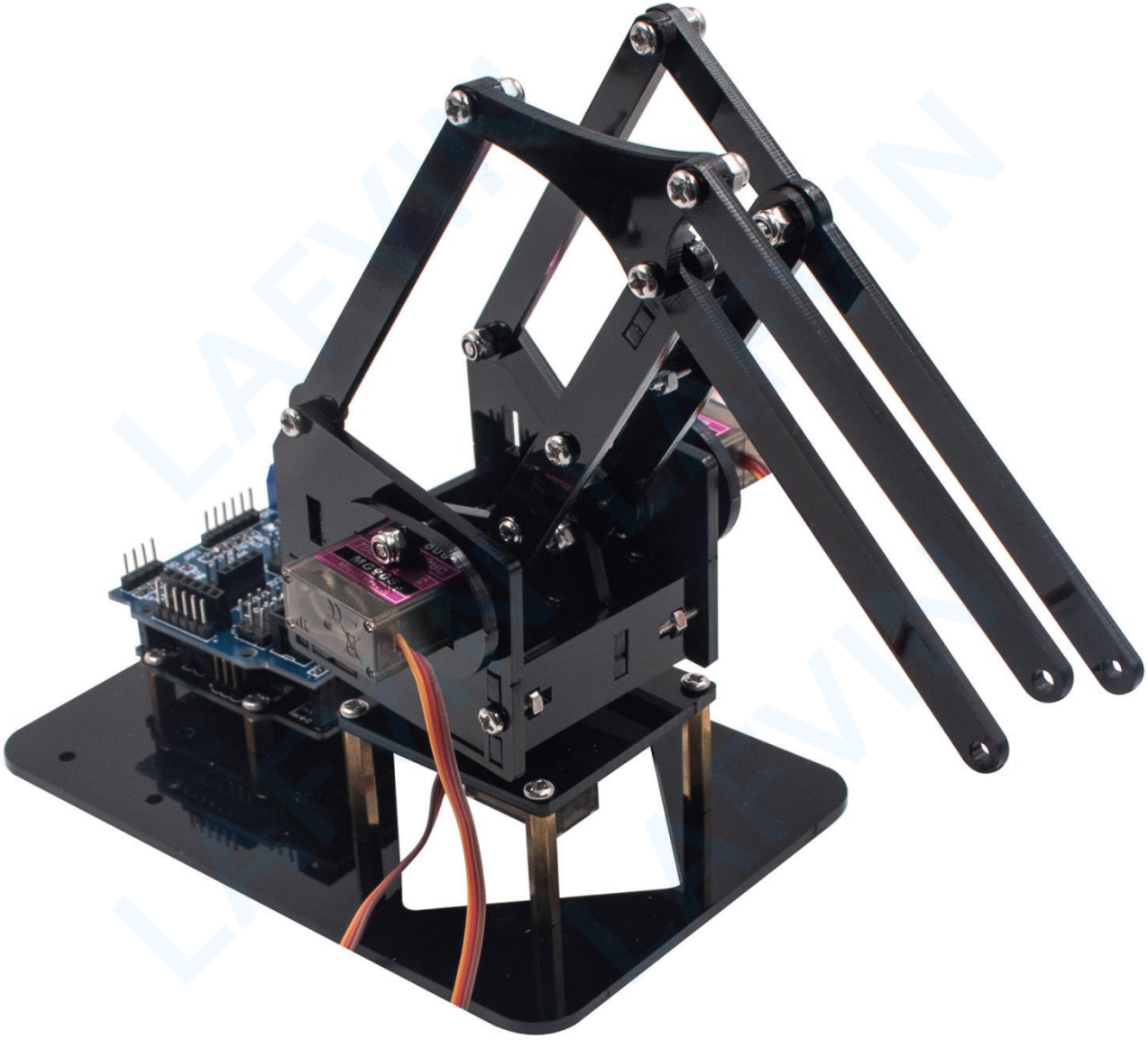



Behind
Front
Front
Behind
Result

Step 20

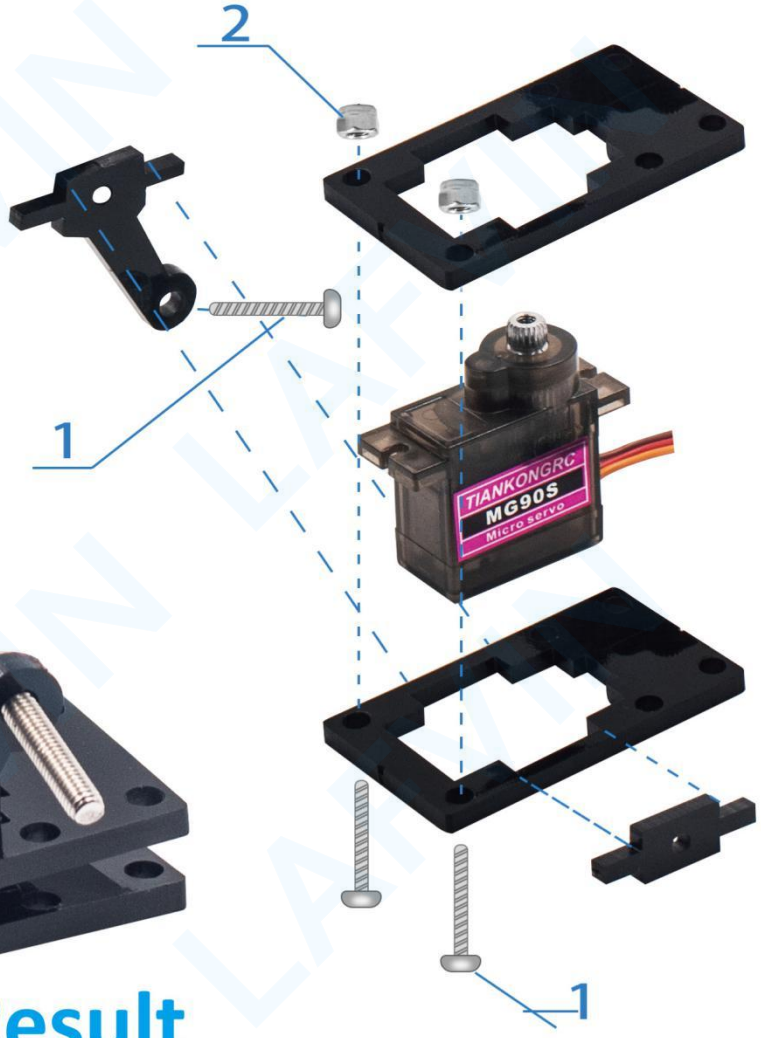
1. M3*10mm
2. M3*12mm
3. M3 Nut


Result

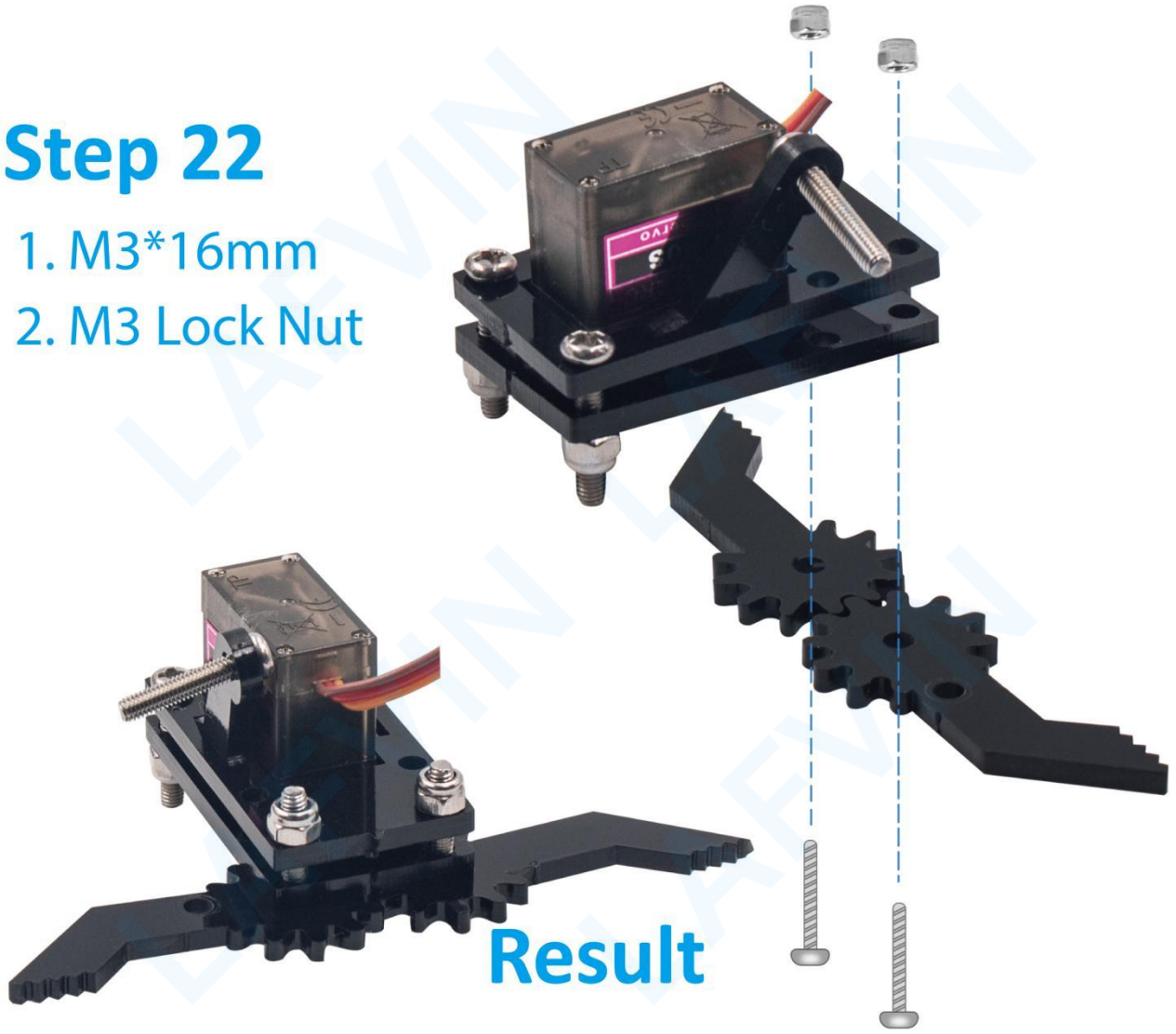


Step 21

1. M3*16mm
2. M3 Lock Nut



Result

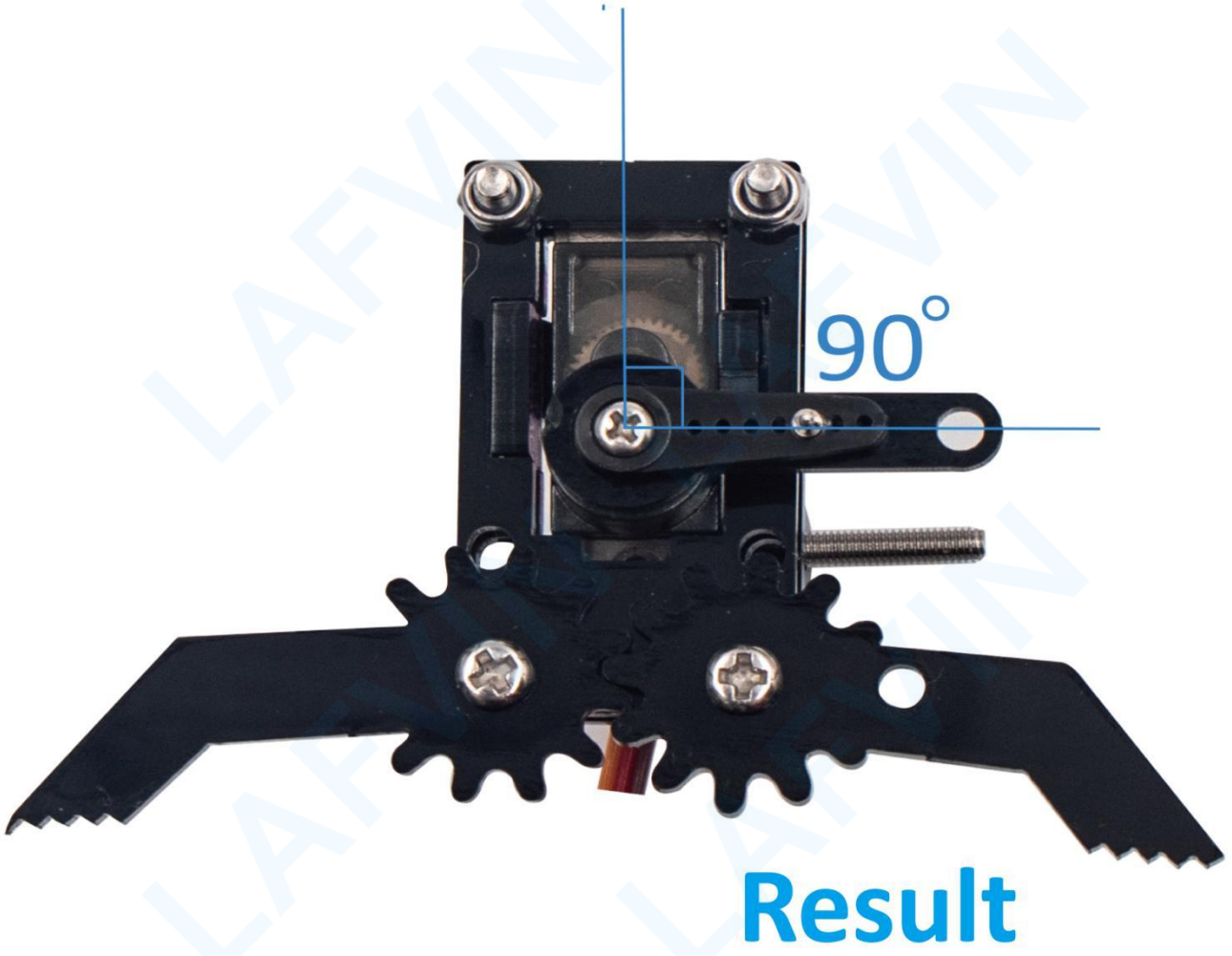

Step 22
1. M3*16mm
2. M3 Lock Nut
Result

Step 23

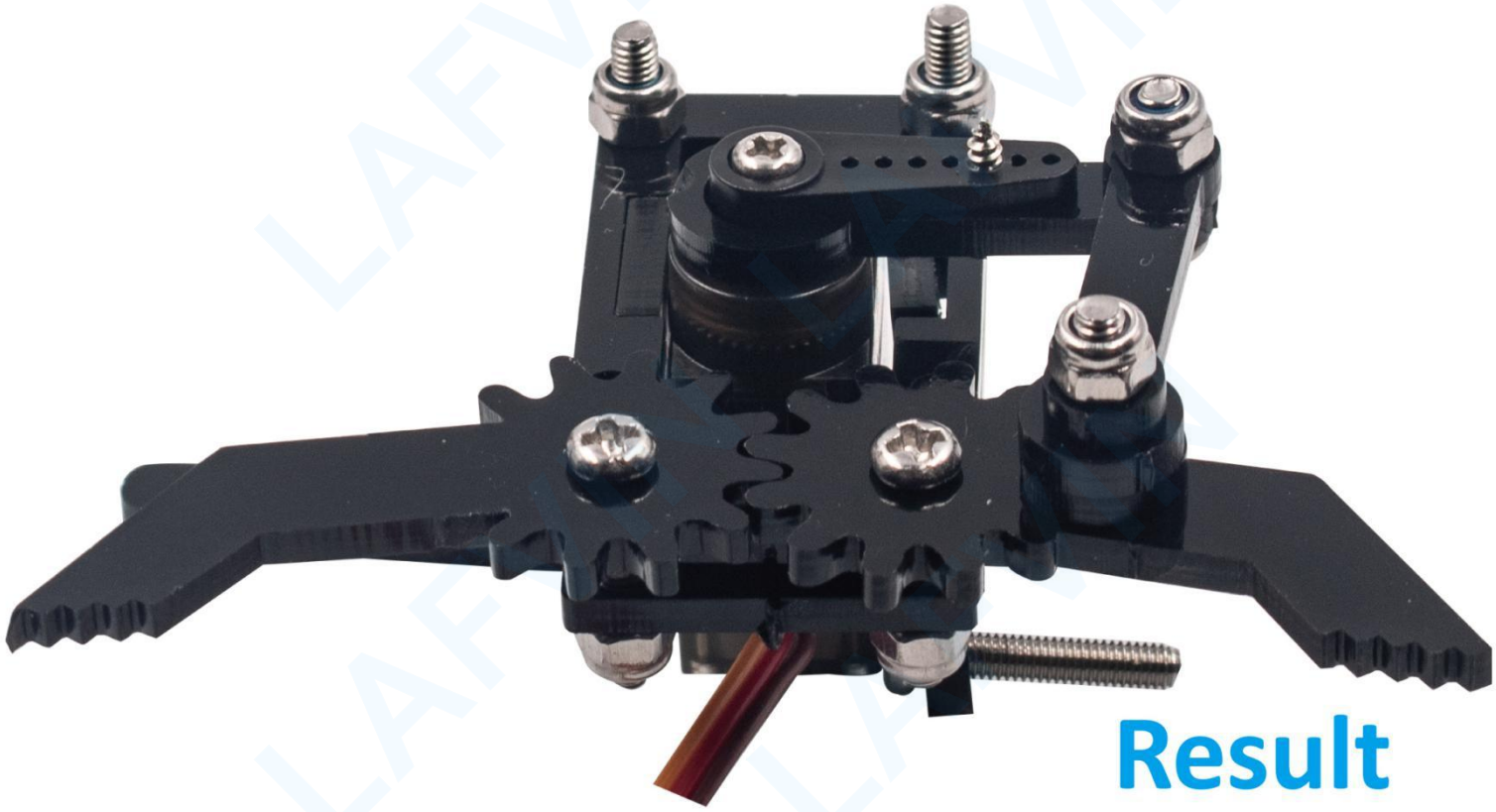
1. Tip Screw from Steering Gear

Result

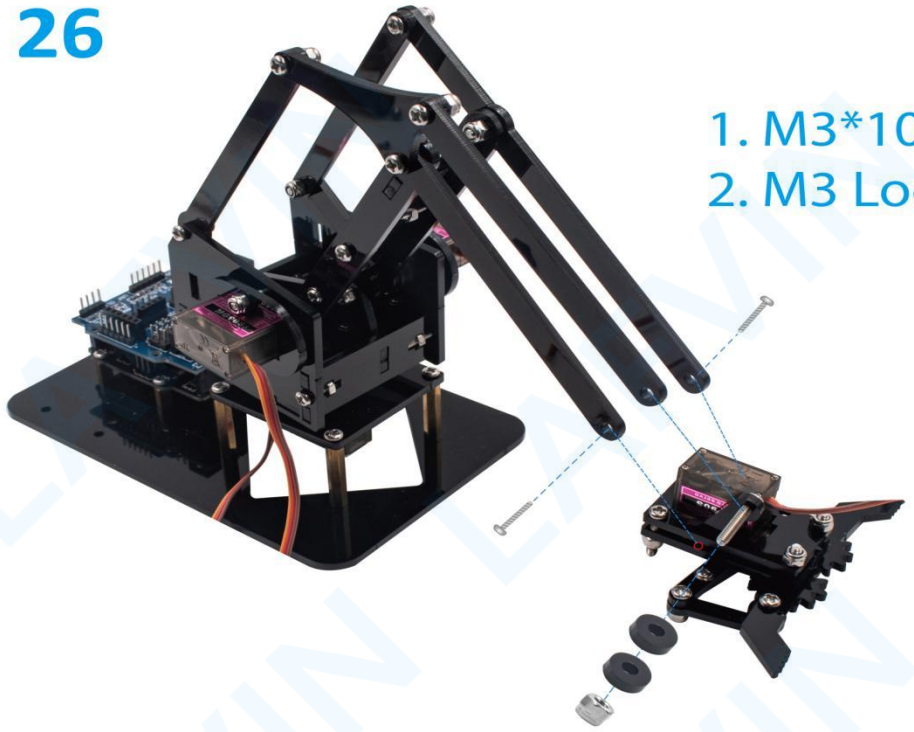
Step 24


90°
Result

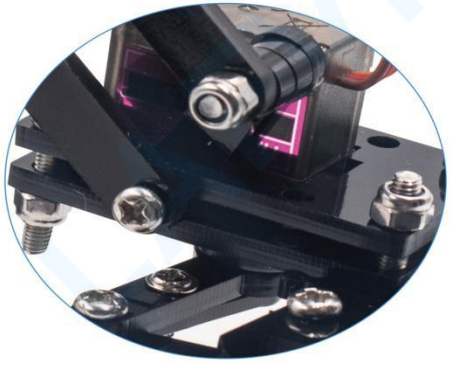

Steps 25
1. M3*10mm
2. M3*16mm
3. M3 Lock Nut
4. Gasket
3
4
1
2


Result

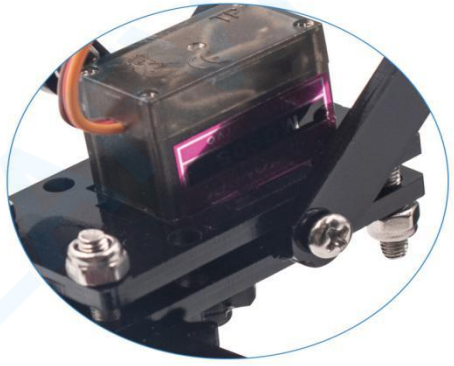
Step 26



1. M3*10mm
2. M3 Lock Nut



Result



Result


Result


Result

Step 27

1. M3*6mm


Step 28
1. M3*6mm


Result

After the installation is complete, note : the screws on the joints that the robot arm can rotate are slightly looser. If the screws are too tight, they will not rotate, so you should check that the screws on the joints that can be rotated are not too tight. If you find that your robot arm is not free to rotate or some parts are stuck after the installation, you should carefully check the above installation steps. In particular, pay attention to the points in the installation step diagram that are “parallel”, “90 degrees”, or marked to other texts. There are a few places where buyers often make mistakes.

Lesson 5 Control of servo

About this lesson:

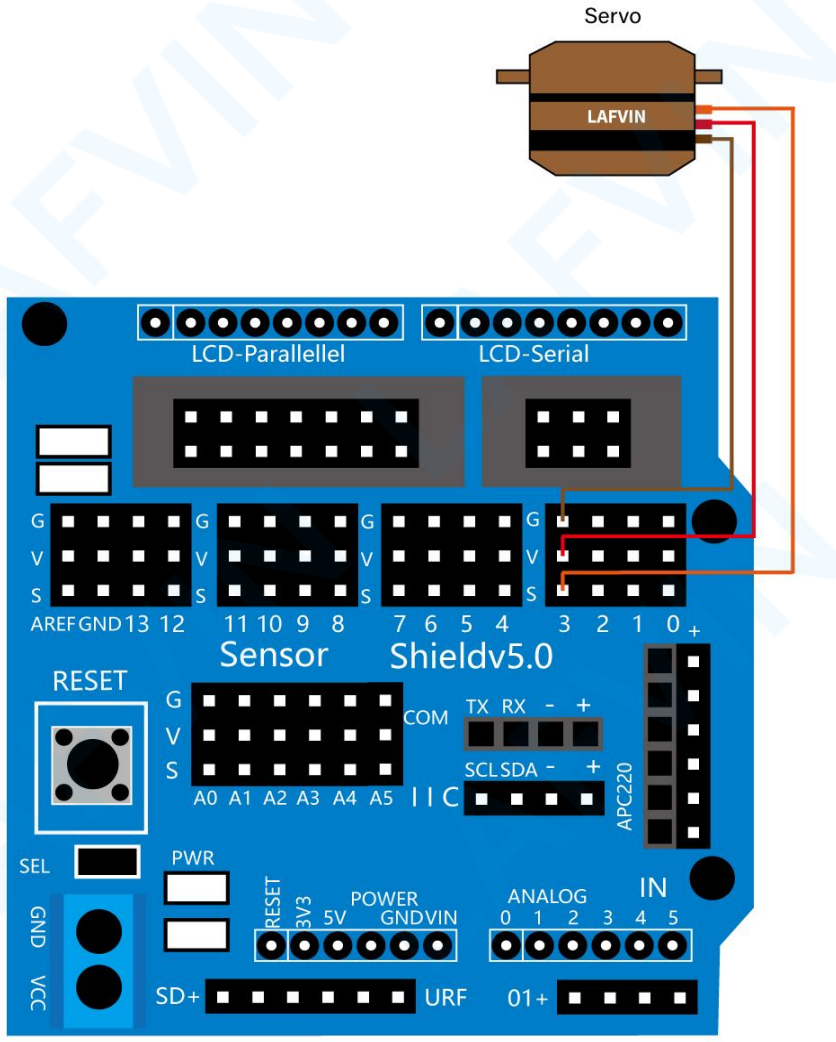
This lesson is divided into two parts. First, learn how to control a single servo motor, and then learn to control four servo motors at the same time.

Introduction

Servo motors are great devices that can turn to a specified position.Usually, they have a servo arm that can turn 180 degrees. Using the Arduino, we can tell a servo to go to a specified position and it will go there. As simple as that!Servo motors were first used in the Remote Control (RC) world, usually to control the steering of RC cars or the flaps on a RC plane. With time, they found their uses in robotics, automation, and of course, the Arduino world.

There are two ways to control a servomotor with Arduino. One is to use a common digital sensor port of Arduino to produce square wave with different duty cycle to simulate PWM signal and use that signal to control the positioning of the motor. Another way is to directly use the Servo function of the Arduino to control the motor. In this way, the program will be easier.Next, we learn how to control the servo.The servo motor has three leads. The color of the leads varies between servo motors, but the red lead is always 5V and GND will either be brown.The red one is the power wire and should be connected to the 5v port and signal control line is usually orange

Wiring diagram

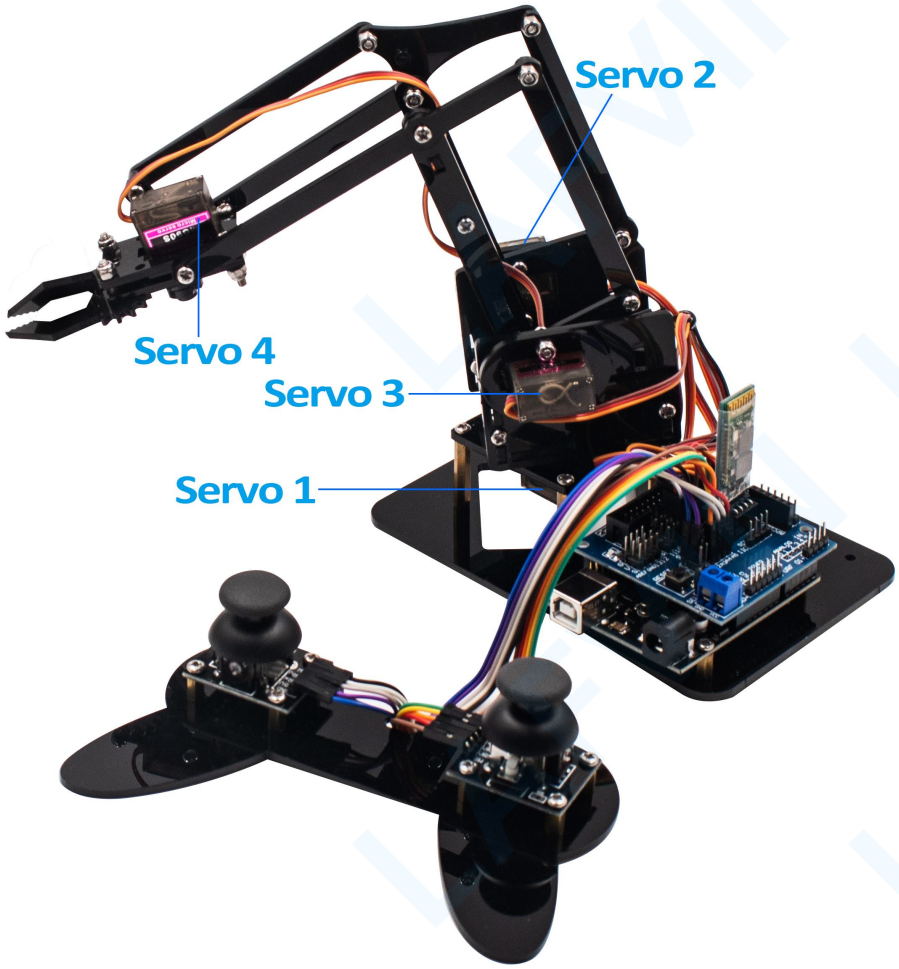


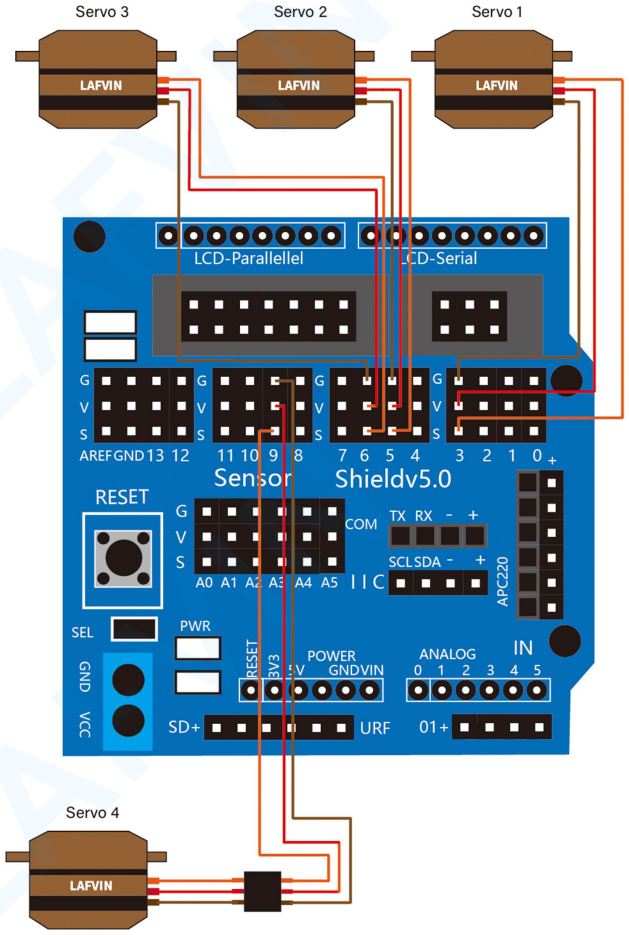
Test instructions

After connecting,please open the the program and load up the code - **Lesson_5_Single_Servo** onto your Arduino board. See Lesson 3 for details about program uploading if there are any errors.Before you can run this, make sure that you have installed the < Servo> library or re-install it, if necessary. Otherwise, your code won't work.For details about loading the library file, see Lesson 2.

On the basis of single servo motor control, we learn to control four servo motors at the same time.At this point, we need to be aware that the four servo motors are connected to different digital interfaces on the control board.

Wiring diagram





Test instructions

After connecting,please open the the program and load up the code - **Lesson_5_Multiple_servo** onto your Arduino board.upload well the code. Powered on, press the reset button, the robot arm will realize a cyclic motion. The claw is clamped,after one second the claw is opened.Turn to the right,after one second turned to the left.stretch out the arm,after one second the arm is retracted.the claw is raised after one second lowered.

Lesson 6 Analog Joystick Module

About this lesson:

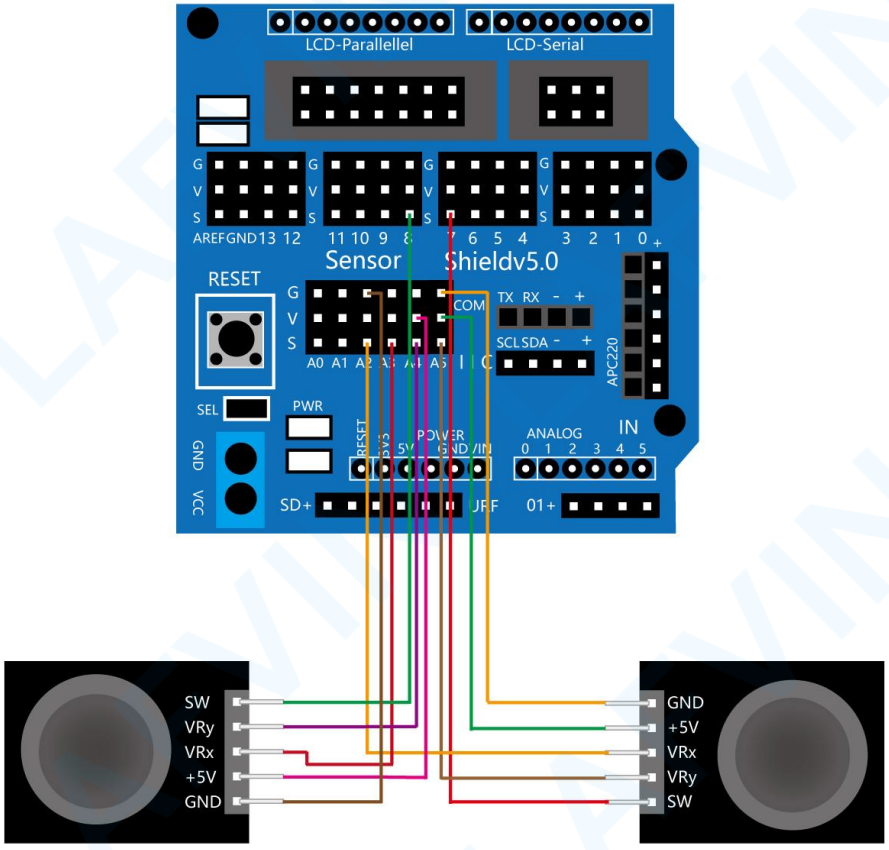
Analog joysticks are a great way to add some control in your projects. In this tutorial we will learn how to use the analog joystick module.

Introduction

Joystick

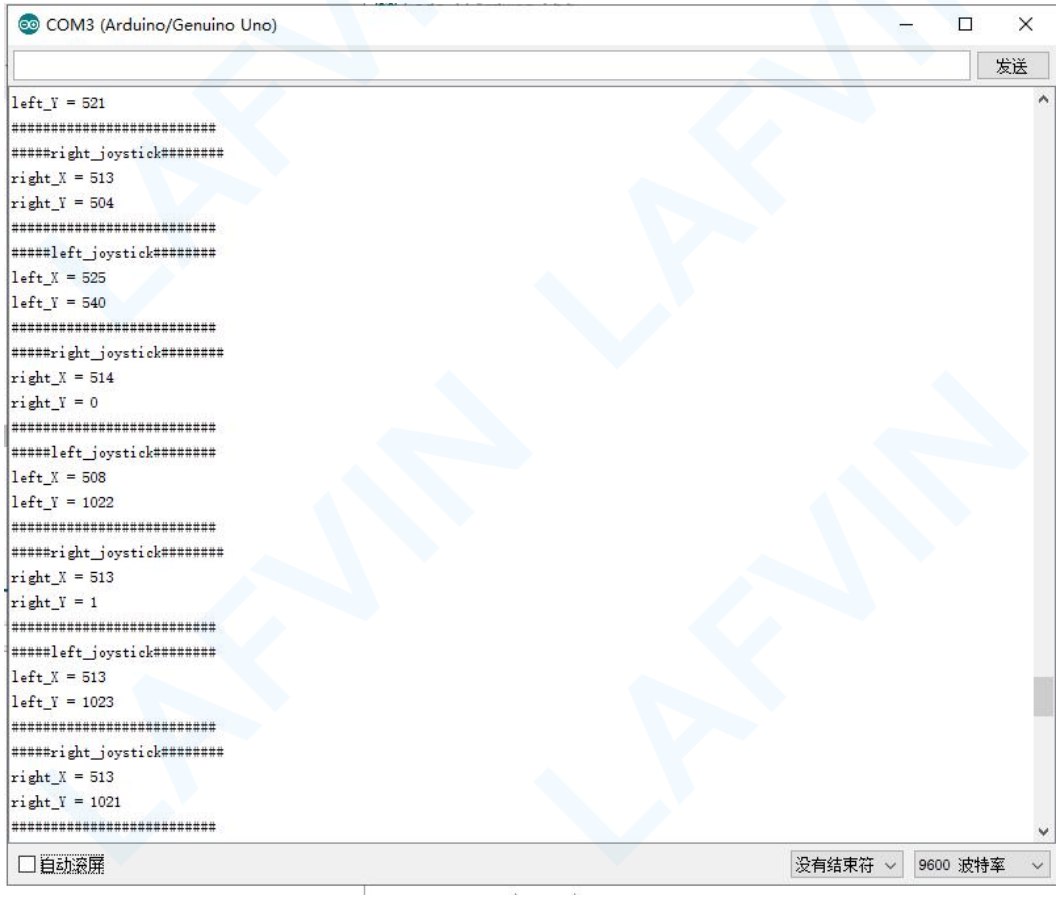
The module has 5 pins: VCC, Ground, X, Y, Key. Note that the labels on yours may be slightly different, depending on where you got the module from. The thumb stick is analog and should provide more accurate readings than simple ‘directional’ joysticks tact use some forms of buttons, or mechanical switches. Additionally, you can press the joystick down (rather hard on mine) to activate a ‘press to select’ push-button.
We have to use analog Arduino pins to read the data from the X/Y pins, and a digital pin to read the button. The Key pin is connected to ground, when the joystick is pressed down, and is floating otherwise. To get stable readings from the Key /Select pin, it needs to be connected to VCC via a pull-up resistor. The built in resistors on the Arduino digital pins can be used. For a tutorial on how to activate the pull-up resistors for Arduino pins, configured as inputs

Wiring diagram



Test instructions

After wiring, please open the program in the code folder- Lesson_6_ Analog_Joystick_Module and click UPLOAD to upload the program.Connect the UNO R3 to computer using a USB cable, then open the serial monitor and set the baud rate to 9600, you should see the analog value of the right Joystick pin X,Y.

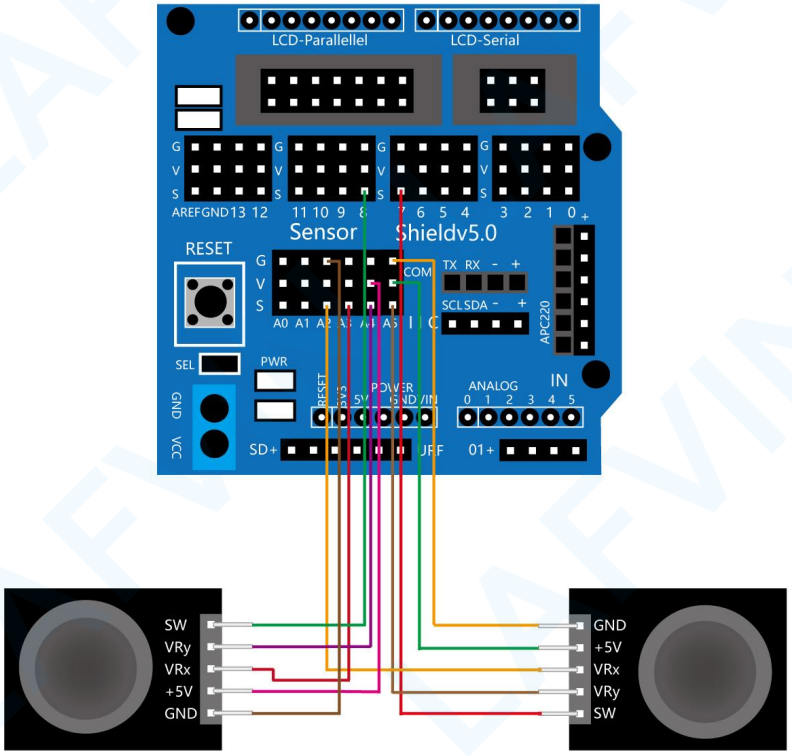


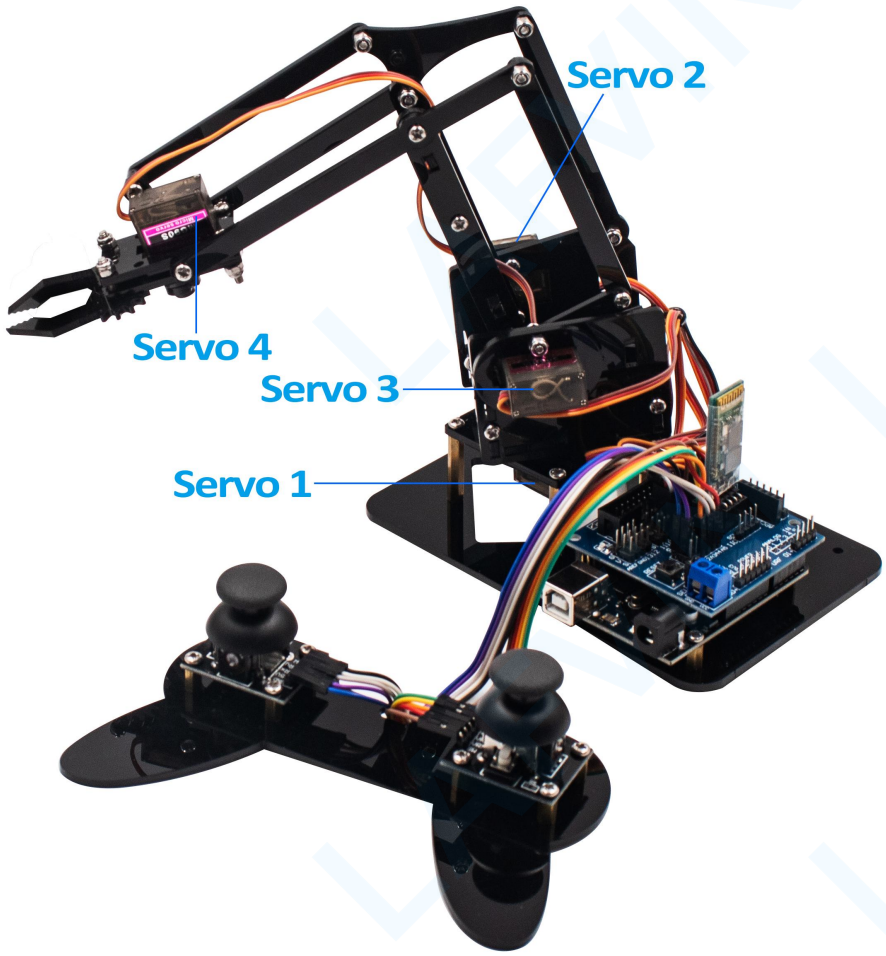
Lesson 7 Dual-JoyStick Controlled Robot Arm

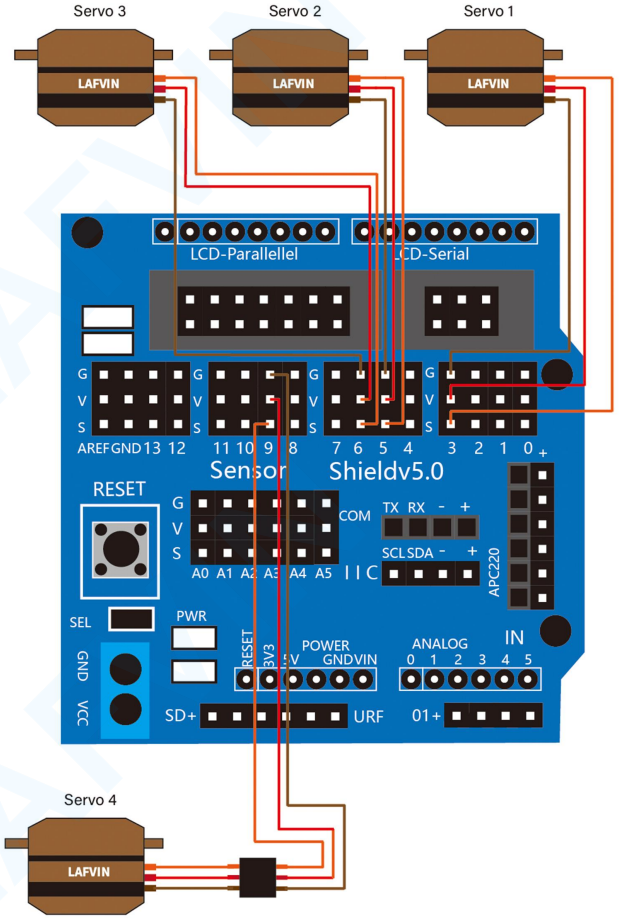
About this lesson:

In this lesson, we have introduced how to use 4 Servo to control the robot arm. Next, combine those two experiments. Use two Joystick modules to control 4DOF robot arm realize different motions.

Wiring diagram




Servo 2
Servo 4
Servo 3
Servo 1


Servo 3
Servo 2
Servo 1
LAFVIN
LAFVIN
LAFVIN
LCD-Parallellel
LCD-Serial
Sensor Shieldv5.0
RESET
COM
TX RX
SCL SDA
APC220
SEL
PWR
POWER
ANALOG
IN
GND
VCC
SD+
URF
Servo 4
LAFVIN

Test instructions

After wiring,please open the program in the code folder- **Lesson_7_Dual_JoyStick_Controlled_Robot_Arm**. connect the UNO R3 to computer using a USB cable,click UPLOAD to upload the program.now you can use two Joystick modules to control 4DOF robot arm realize different motions.The relationship between joystick and manipulator joint control is as follows.

Left Joystick	arm	Right Joystick	arm
X1<50	push the left joystick to the right, the claw opens	X2<50	push the right joystick to the left, the servo that controls the arm rotation turns left
X1>1000	push the left joystick to the left, the claw closed	X2>1000	push the right joystick to the right, the servo that controls the arm rotation turns right
Y1<50	push the left joystick to the down , lower the robot upper arm	Y2>1000	push the right joystick to the down,draw back the robot lower arm
Y1>1000	push the left joystick to the up,lift up the robot upper arm	Y2<50	push the right joystick to the up,stretch out the robot lower arm

Lesson 8 Bluetooth Controlled Robot Arm

About this lesson:

This lesson is divided into two parts, first learning the Bluetooth module communication test, and then learning the app to control the robotic arm

1) Bluetooth test

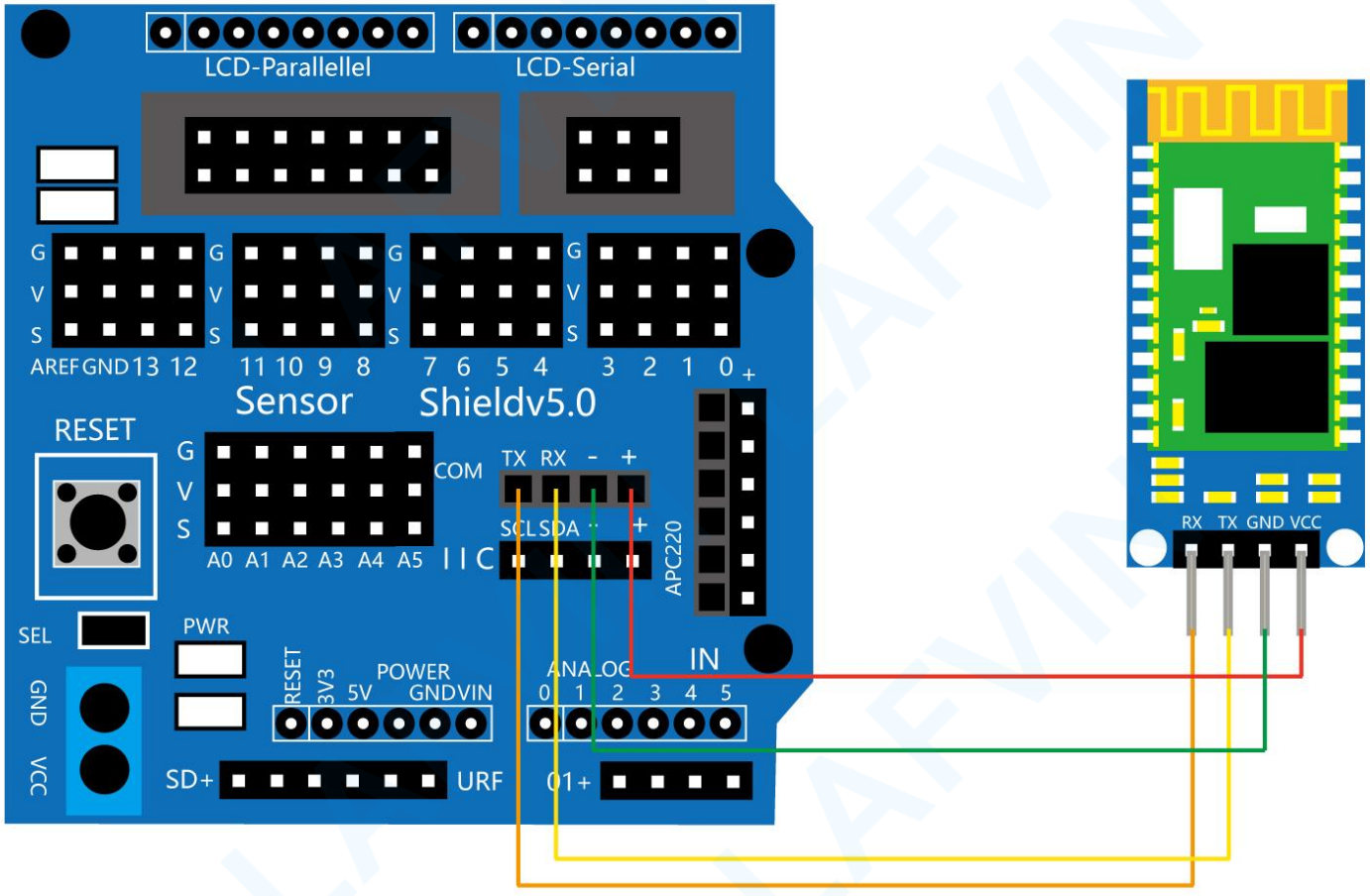
Introduction:

The HC06 is a Serial port Bluetooth module which having fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Blue core 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature).

The HC-06 Bluetooth module to LAFVIN UNO R3:

VCC>>>>5v
GND>>>>GND
TXD>>>>RX
RXD>>>>TX

Wiring diagram

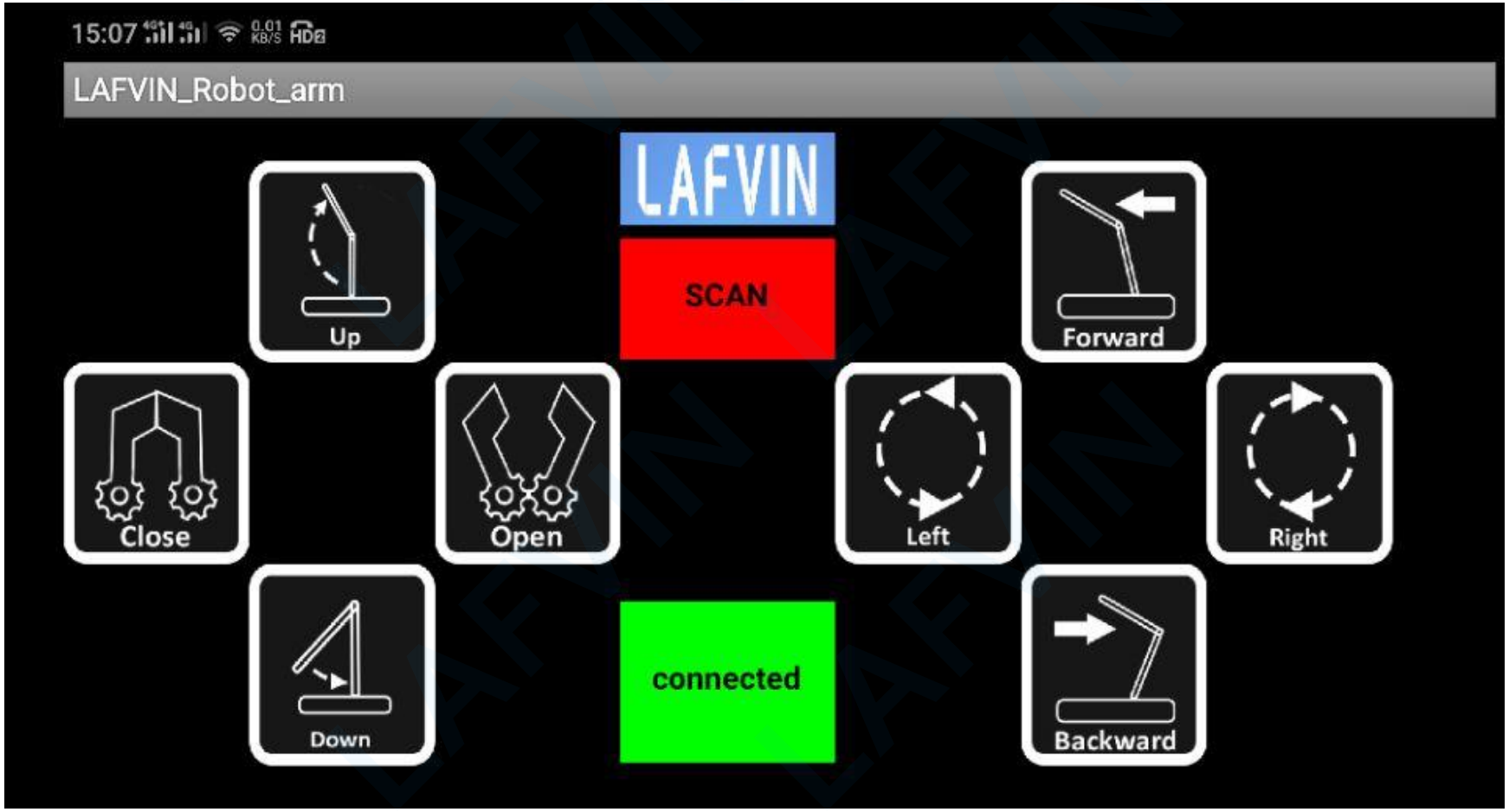


Upload code

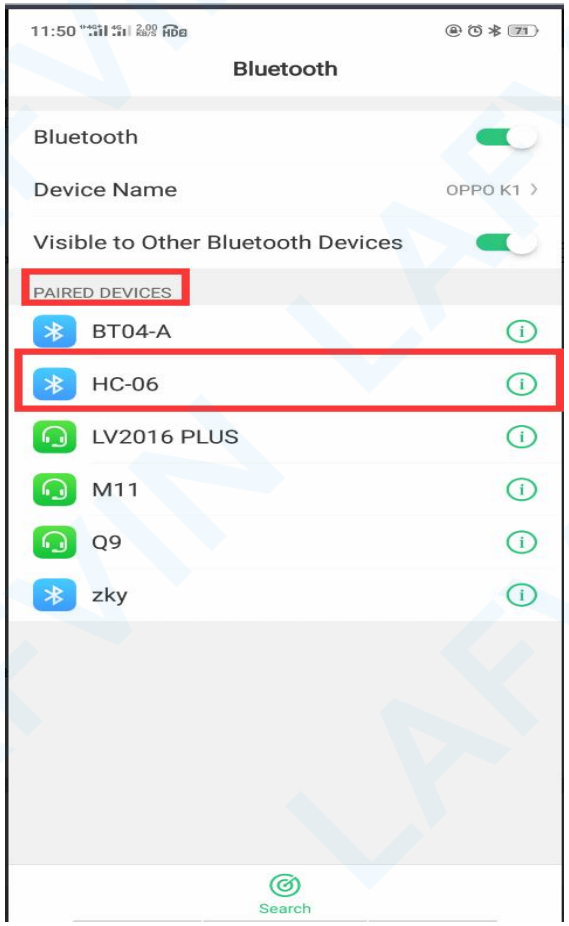
After wiring, please open the program in the code folder--**Lesson_8_Bluetooth_Test_** and click UPLOAD to upload the program. See Lesson 3 for details about program uploading if there are any errors.Attention:The bluetooth module should be pulled out before you upload the program every time,or it will be failed to upload the program.After the code has been successfully uploaded to the UNO board, reconnect the Bluetooth.

Instructions for the use of app

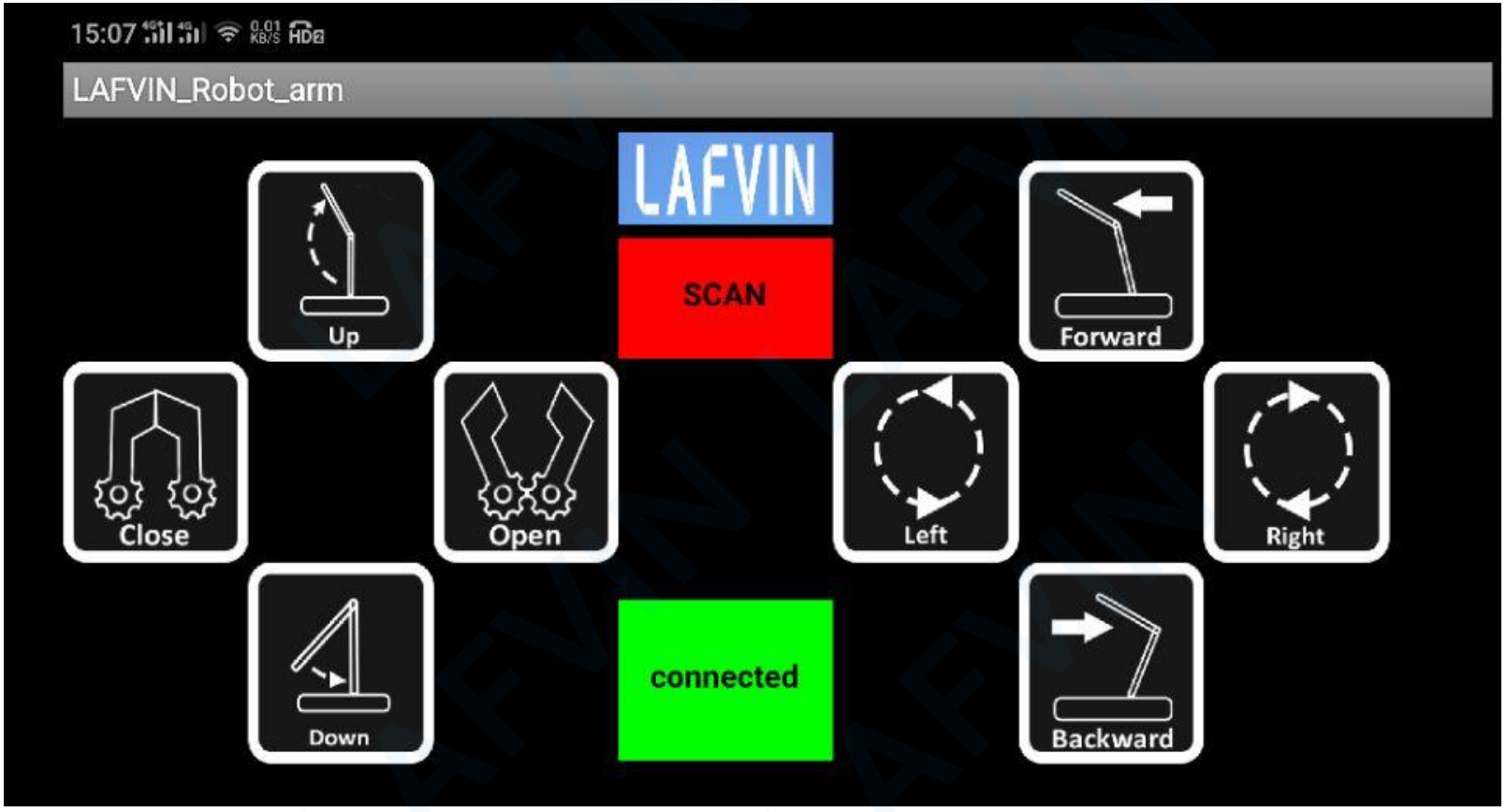
Firstly, download the LAFVIN_Ropot_arm.APK file from the folder in Lesson 8 to your mobile phone and install it into an application software.



Then make sure the Bluetooth module is connected. Pair your phone with HC-06. for doing this go to Settings->Bluetooth->Scan device->select HC-06 and pair it. Pass code to pair is '1234'.Open Bluetooth Terminal software, go to options and select 'connect a device - secure' option. It it ask for pass code enter 1234.If your phone is connected to the Bluetooth module, you will see a usable device called HC-06 on the PAIRED DEVICES(As shown below).If the HC-06 does not appear on the PAIRED DEVICES, reoperate the above steps.



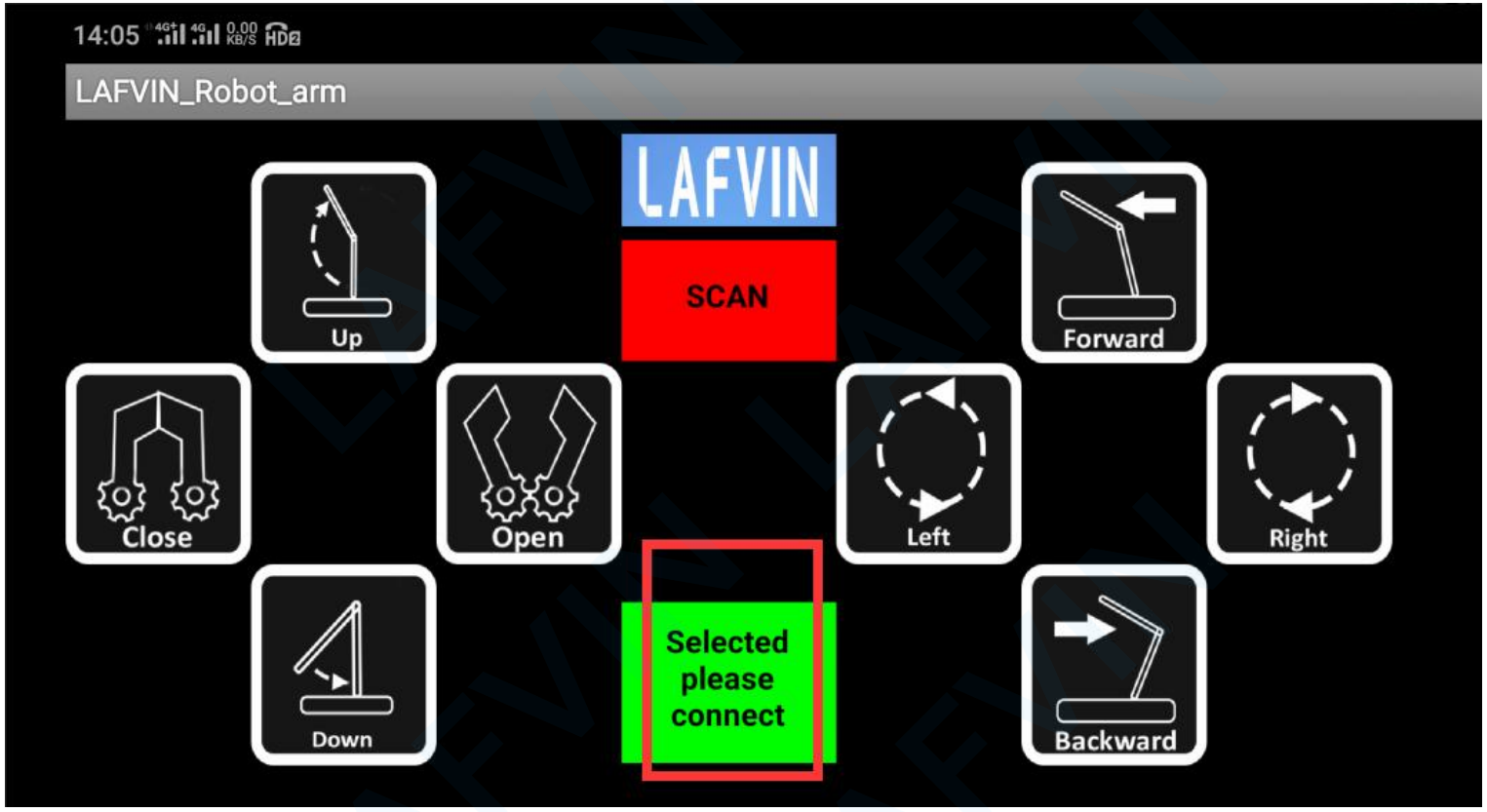
After the above steps are complete, we open the **LAFVIN_Robot_arm app**.



Then click on the SCAN, and the HC-06 will appear in our scan results.Select HC-06.

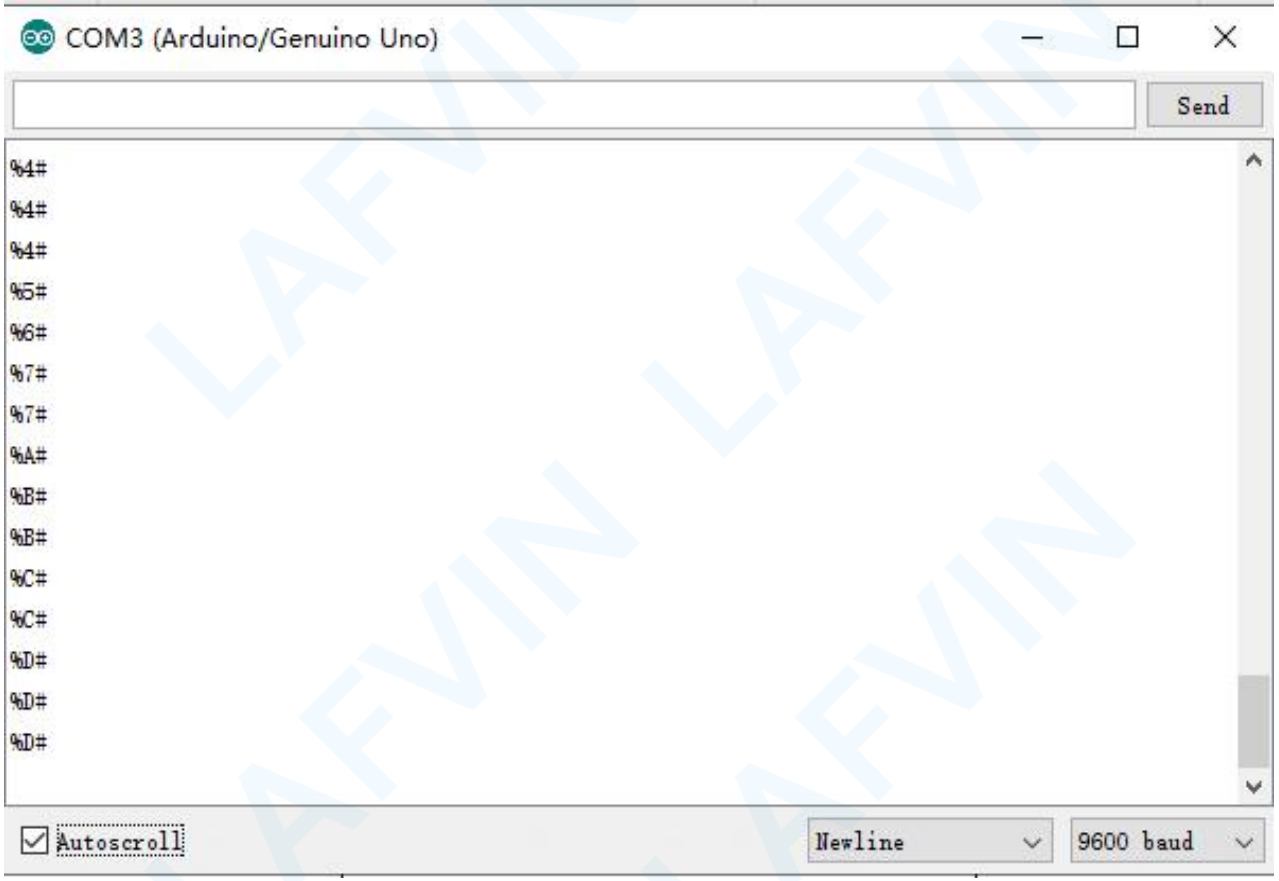
After selecting the hc-06 device, click on the green button and wait 5 seconds. The green button shows the connected , which means that the app is connected to the Bluetooth module successfully.(Note: When the Bluetooth module is not successfully connected, the red LED light will

continue to flash. When the connection is successful, the red LED light will be remain on.)



Test instructions

Upload the code, hookup well and power up. After connecting the Bluetooth APP, open the serial monitor and set the baud rate to 9600, press down the control key on the APP, you should see the corresponding value. Shown below.

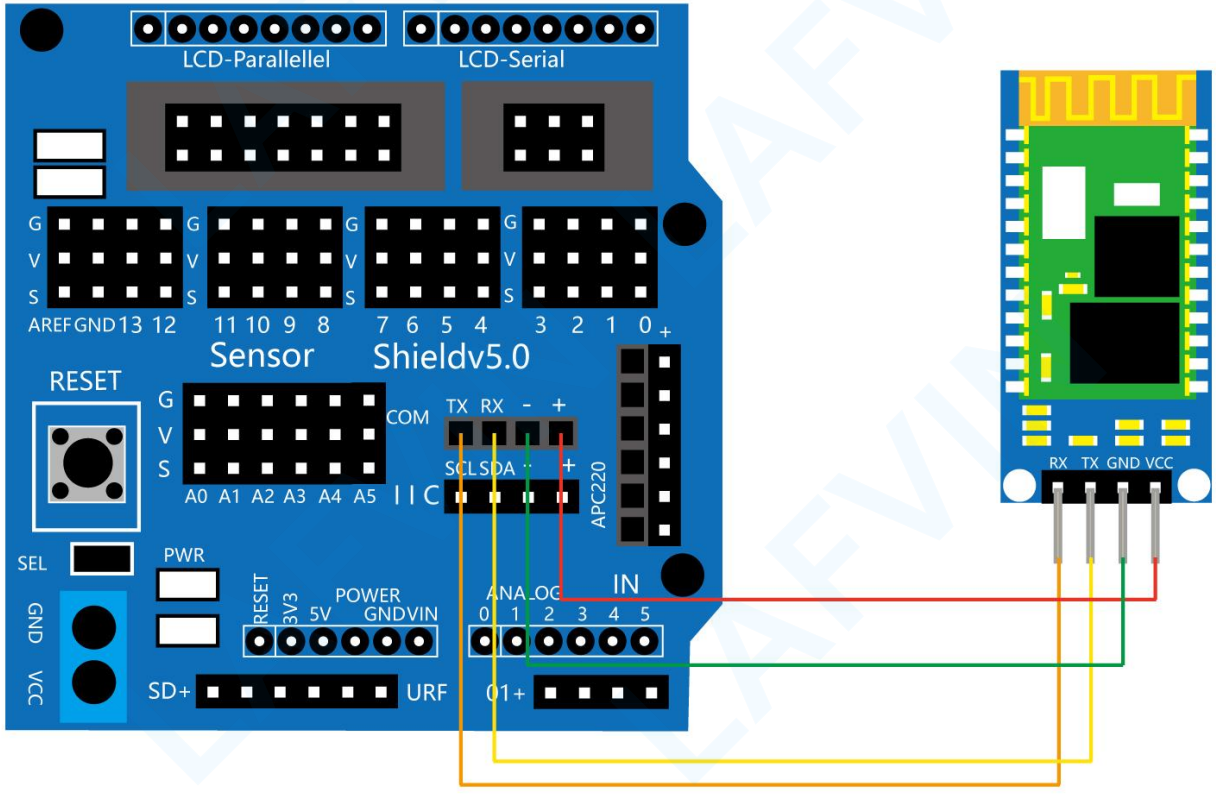


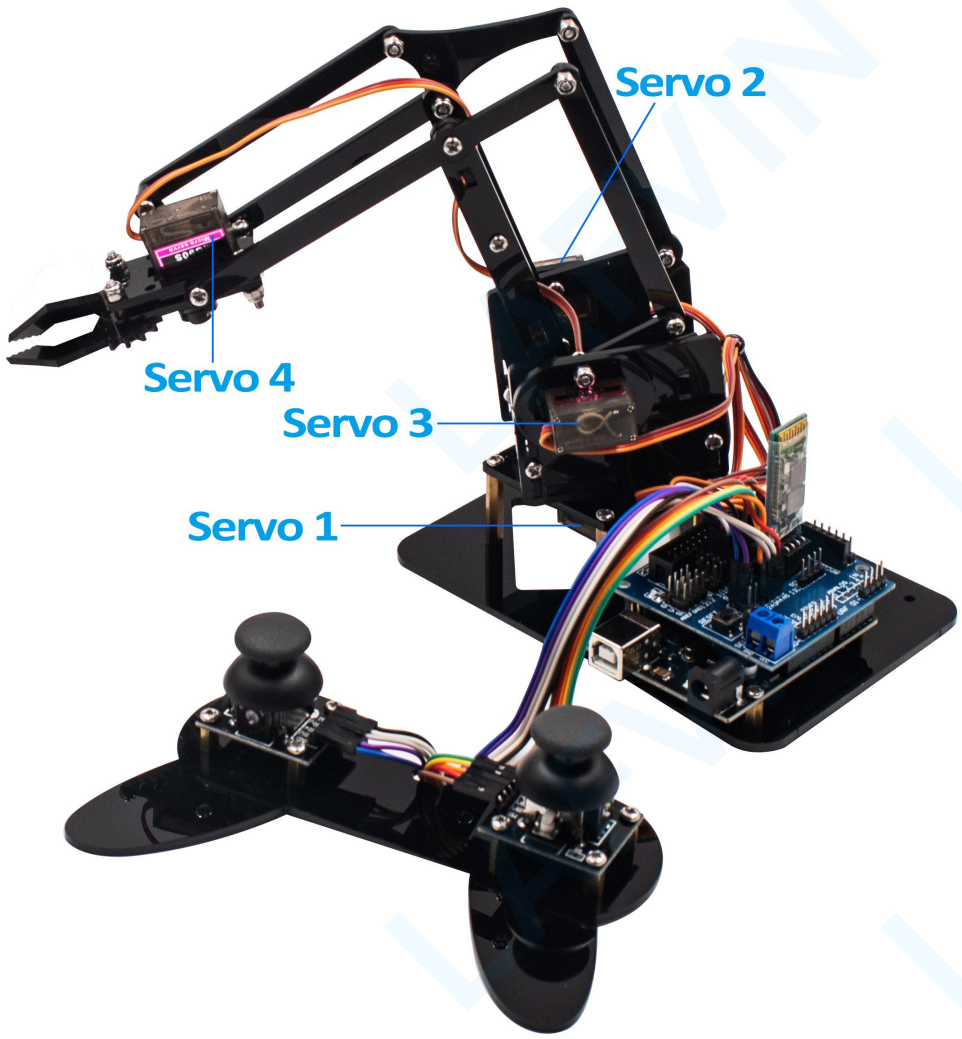
APP KEY	value
	%4#
	%5#
	%6#
	%7#
	%A#
	%B#
	%C#
	%D#

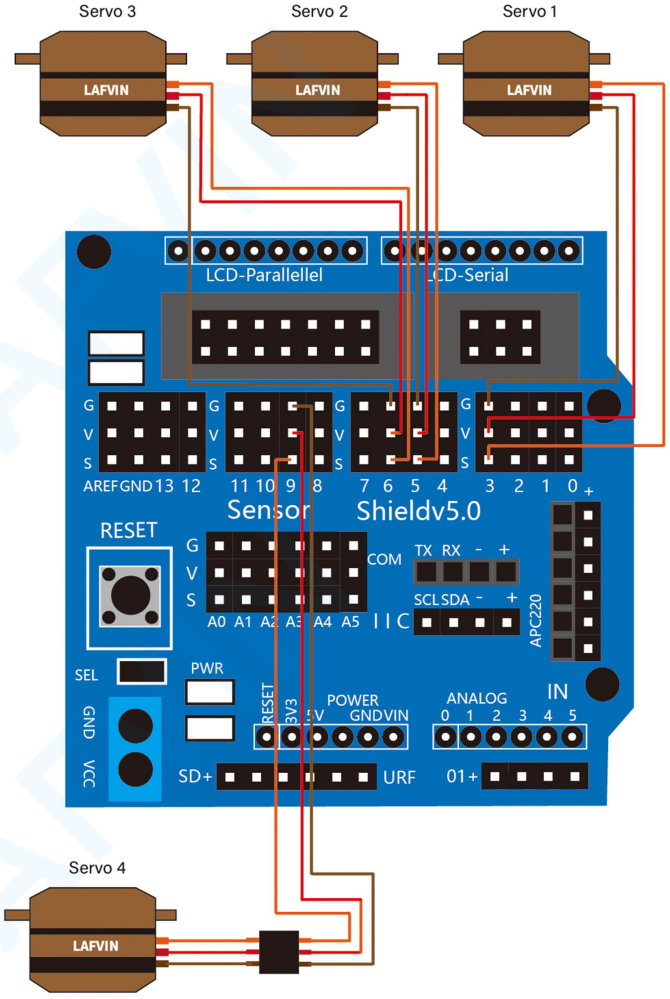
2)Bluetooth Controlling Arm

After learning how to send instructions to the Bluetooth module, learn how to control the four servo motors of the robot arm through the app.

Wiring diagram




Servo 2
Servo 4
Servo 3
Servo 1


Servo 3
Servo 2
Servo 1
LAFVIN
LAFVIN
LAFVIN
LCD-Parallellel
LCD-Serial
Sensor Shieldv5.0
RESET
COM
TX RX
SCL SDA
APC220
SEL
PWR
POWER
ANALOG
IN
SD+
URF
Servo 4
LAFVIN

Test instructions

After wiring, please open the program in the code folder--**Lesson_8_Bluetooth_Controlling_Arm** and click UPLOAD to upload the program. Attention:The bluetooth module should be pulled out before you upload the program every time,or it will be failed to upload the program.After the code has been successfully uploaded to the UNO board, reconnect the Bluetooth.Repeat the steps above regarding the app connection to the Bluetooth module.Now that the app has successfully connected to Bluetooth, we can begin to control the robotic arm through the app.

APP KEY	value
	lower the robot upper arm
	the claw closed
	the claw opens
	lower the robot upper arm
	stretch out the robot lower arm
	turns left
	turns right
	draw back the robot lower arm

Lesson 9 PS2 Controlled Robot Arm (Extension)

About this lesson:

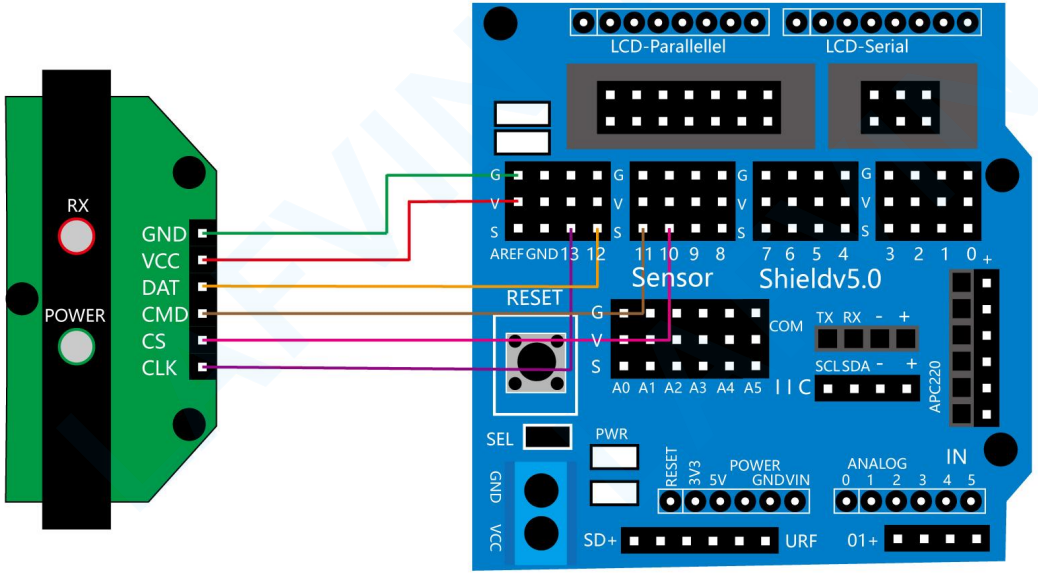
This lesson is divided into two parts. First, learn the PS2 Joypad to control the movement of the robot arm, and then learn to remember the action posture through the PS2 Joypad.But you need to purchase it by yourself because the PS2 Joypad is not included in the kit.

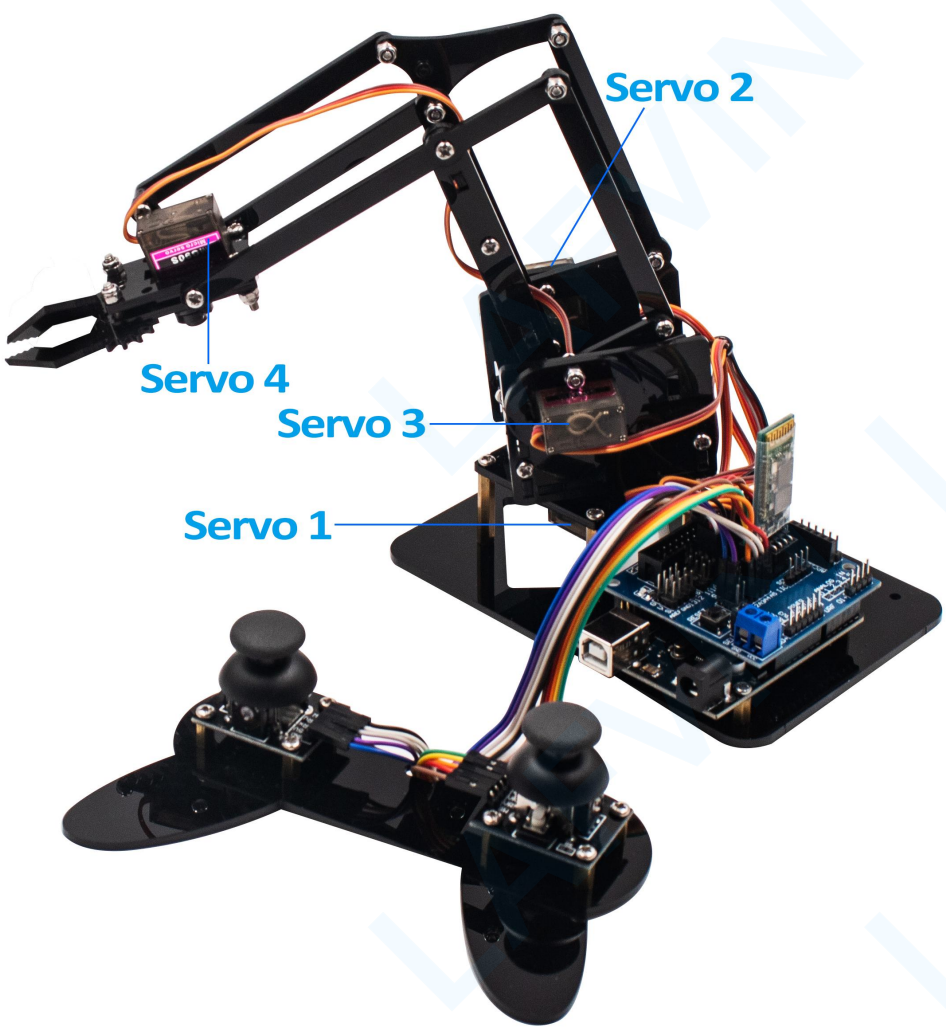
1）PS2 control arm

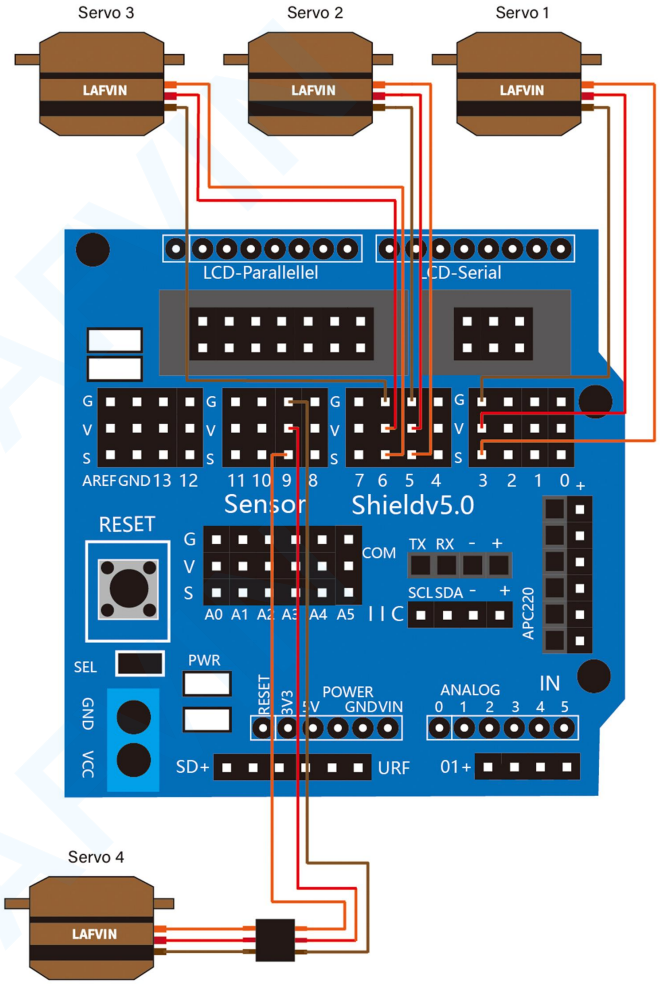
Introduction:

The PS2 handle consists of a handle and a receiver. The handle is mainly responsible for sending key information. Receiver and MCU(also known as the host, which can be used directly on the PS2 game console) is connected to receive information from the handle and transmitted to Arduino.

Wiring diagram




Servo 2
Servo 4
Servo 3
Servo 1


Servo 3
Servo 2
Servo 1
LAFVIN
LAFVIN
LAFVIN
LCD-Parallellel
LCD-Serial
Sensor Shieldv5.0
RESET
COM
TX RX
SCL SDA
APC220
SEL
PWR
POWER
ANALOG IN
GND
VCC
SD+
URF
Servo 4
LAFVIN

Test instructions

After wiring, please open the program in the code folder--**Lesson_9_PS2_Control_Arm** and click UPLOAD to upload the program. But before upload the program,should place the PS2X_lib folder inside the libraries folder of Arduino IDE directory. Method for Adding Library Files to DIE Reference **Lesson 2-Add Libraries and Open Serial Monitor**.We provide **PS2X_lib** library files in the library on the CD.

After successfully uploading the program, turn on the PS2 handle power switch. The signal light will flash. When the signal stop flashing, the controller and the receiver will be connected successfully.The control relationship between joystick and manipulator on handle is as follows

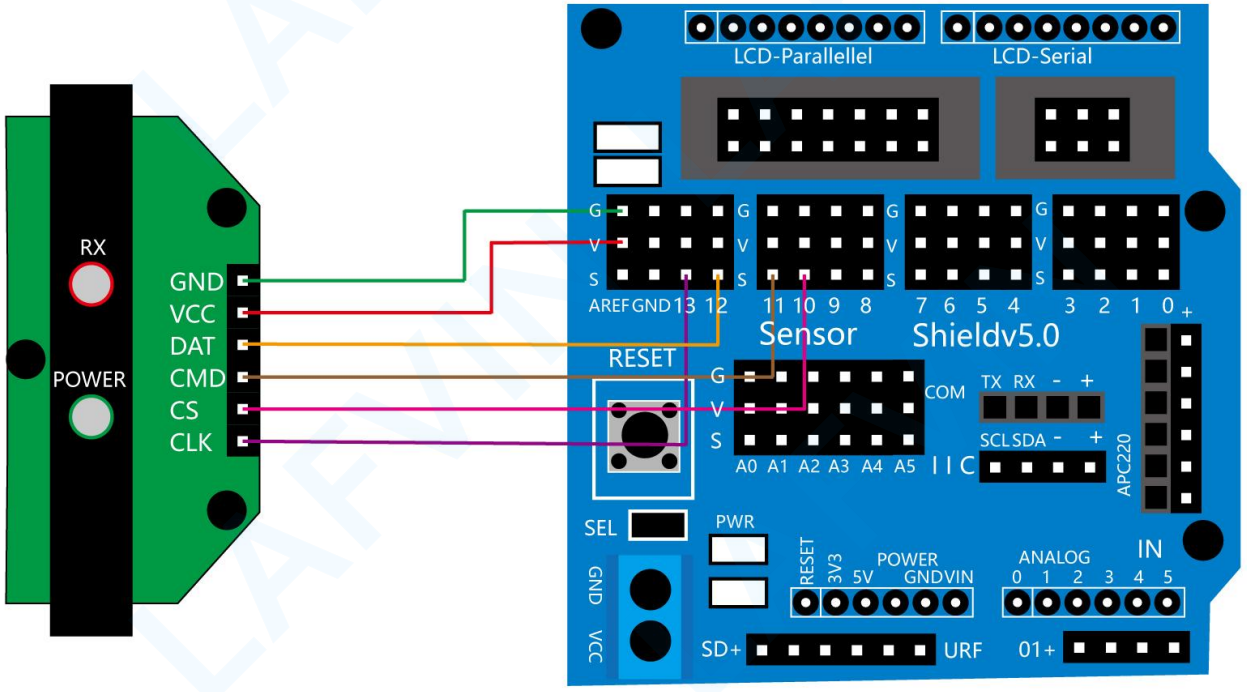
Right Joystick	arm	Left Joystick	arm
X1<50	push the left joystick to the right, the claw opens	X2<50	push the right joystick to the left, the servo that controls the arm rotation turns left
X1>1000	push the left joystick to the left, the claw closed	X2>1000	push the right joystick to the right, the servo that controls the arm rotation turns right
Y1>1000	push the left joystick to the down , lower the robot upper arm	Y2>1000	push the right joystick to the down,draw back the robot lower arm
Y1<50	push the left joystick to the up,lift up the robot upper arm	Y2<50	push the right joystick to the up,stretch out the robot lower arm

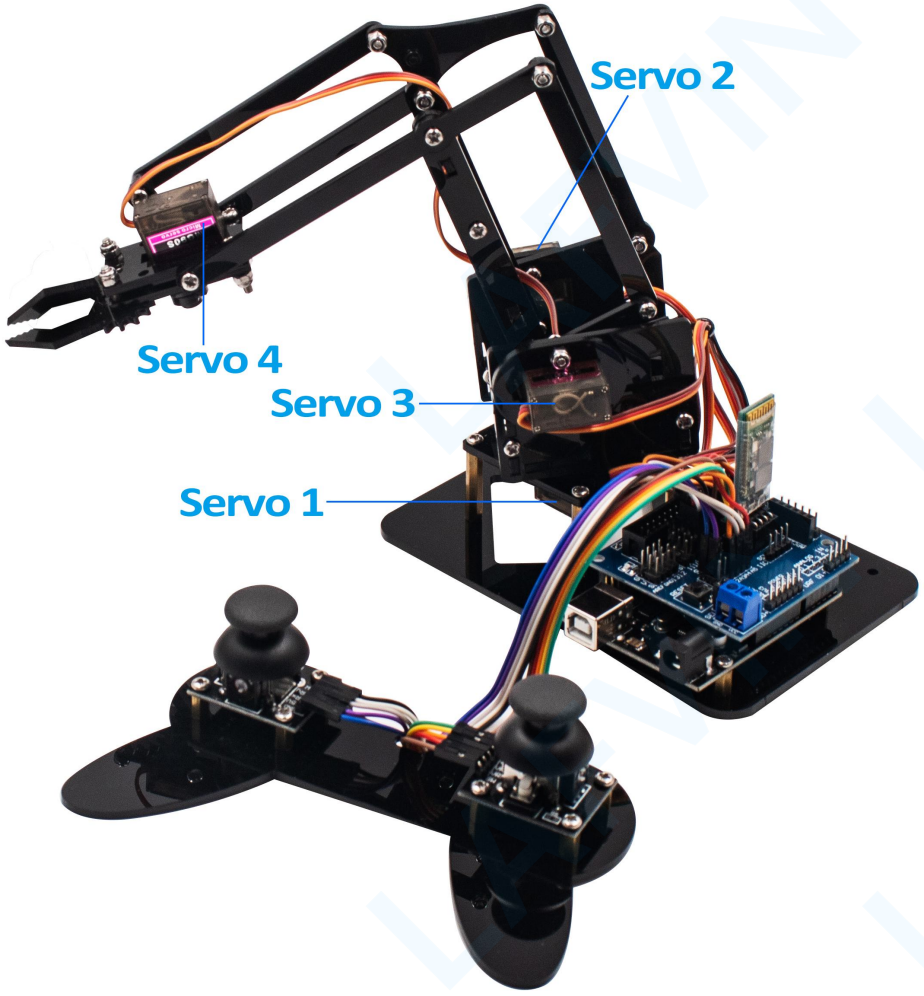
2）PS2 Controlling Posture Memory

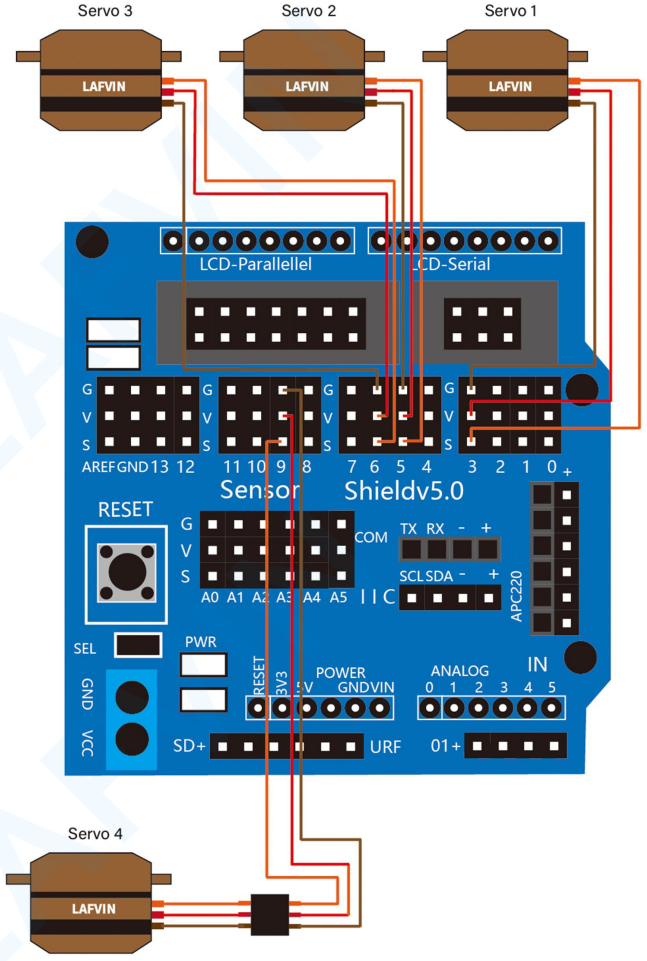
Introduction:

On the basis of the joystick control robot arm, increase the memory function of the action. Press the key Z1 of right Joystick to save the angle value of 4 servos control. Press the key Z2 of left Joystick to operate a servo posture saved in the variable.

Wiring diagram




Servo 2
Servo 4
Servo 3
Servo 1


Servo 3
Servo 2
Servo 1
LAFVIN
LAFVIN
LAFVIN
LCD-Parallellel
LCD-Serial
AREFGND13 12 11 10 9 8 7 6 5 4 3 2 1 0
Sensor Shieldv5.0
RESET
COM
TX RX
SCL SDA
A0 A1 A2 A3 A4 A5
APC220
SEL
PWR
GND
VCC
RESET
3V3
POWER
GNDVIN
ANALOG
0 1 2 3 4 5
IN
SD+
URF
01+
Servo 4
LAFVIN

Test instructions

After wiring, please open the program in the code folder--**Lesson_9_PS2_Controlling_Posture_Memory** and click UPLOAD to upload the program.You can control Arduino and remember no more than 20 actions.Press the key Z1 of right Joystick to save the angle value of 4 servos control.Then change the movement of the arm through the joystick and press the key Z1 of right Joystick to remember this action as the second action. Press the key Z2 of left Joystick to operate all posture saved in the variable.