

# 六足机器人 Baize\_Hlmini 入门级制作教程

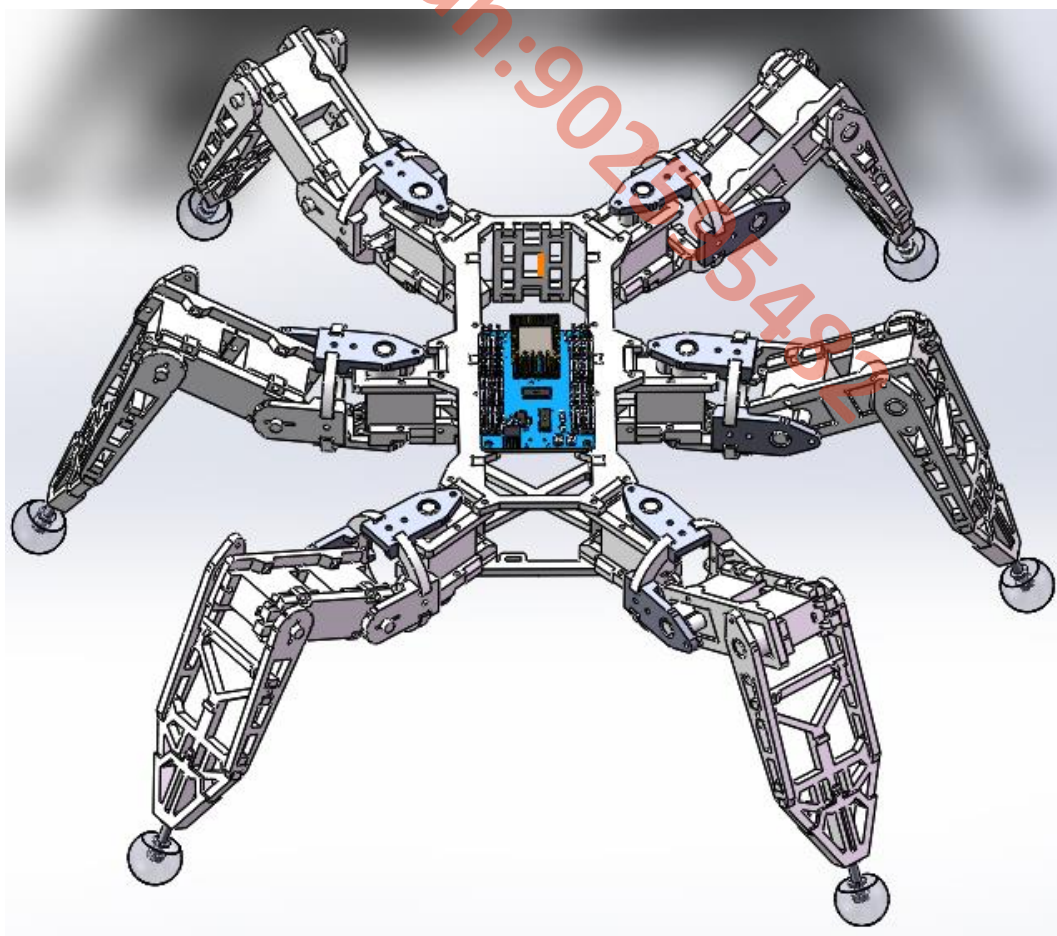
作者：人民群众

版本：20230723

Baize\_Hlmini 六足机器人是一个创客学习平台，通过该学习平台，可以学习 mixly 图形化编程，Arduino 平台的 C 语言编程，ROS 机器人编程，也可以学习一些传感器的控制。

在 Baize\_Hlmini 中，我们摒弃传统的开发板+舵机驱动板的方式。开发板+舵机驱动板的方式一方面集成度不高，杜邦线接线麻烦且可靠性差；另一方面，集成板降低了电器元件的成本，让机器人制作更加方便。

我们有一个 QQ 交流群：902595482，可以进来互相学习。



## 重要提示

1. 电池一定要小心接，最好反复确认自己理解了电源部分是怎么接线的，然后再通电；否则不能随便通电，以免短路起火或发生爆炸。
2. 对于新手来讲，一定要认真阅读理解本教程，不能一目十行不仔细阅读文字说明只看图片，这将会导致很多步骤理解不到。
3. 在碰到不懂且说明书上没有的内容时，一定要善于用百度进行搜索且多多尝试。（也可以在群里问，推荐在群里提问！）

## 一.项目介绍

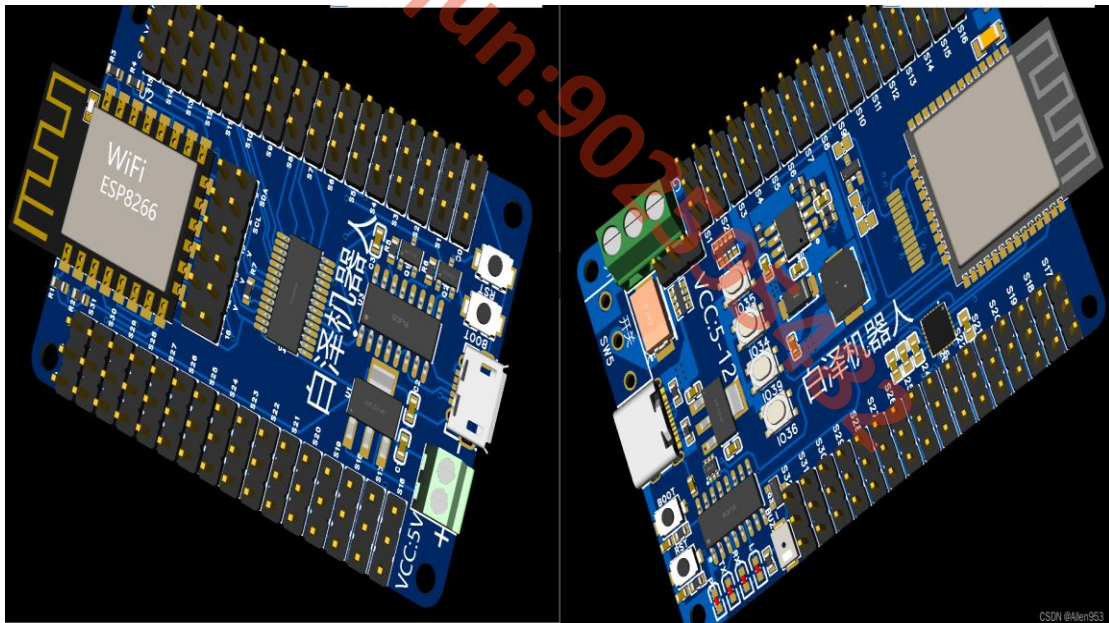
### 1.项目介绍

最新代码在这里：[https://github.com/Allen953/Baize\\_Hlmini](https://github.com/Allen953/Baize_Hlmini)

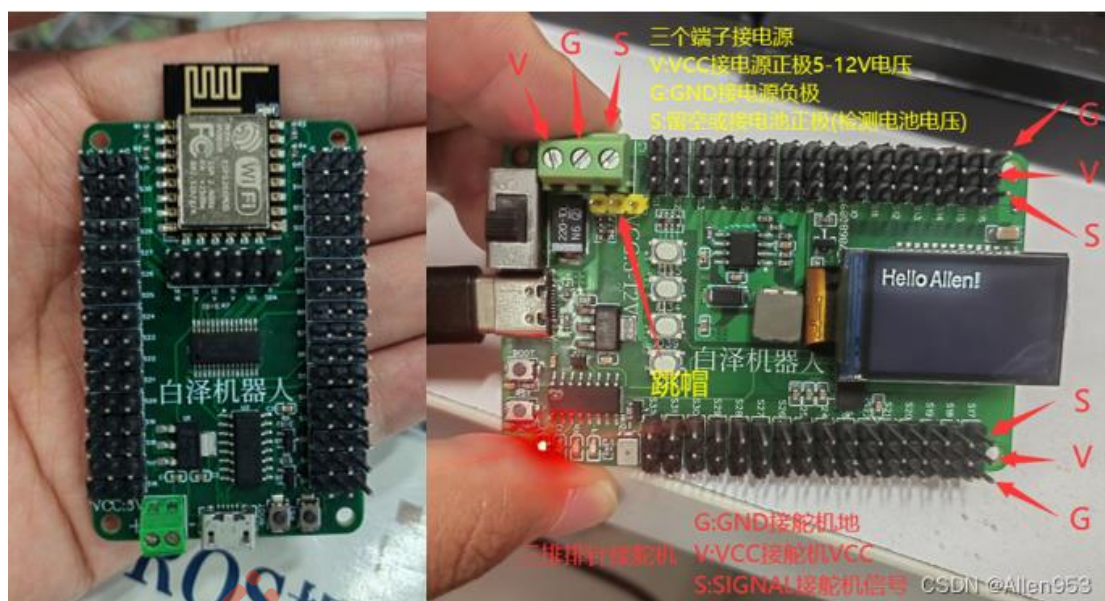
实际上这个机械结构是国外老哥做的一款 arduino 六足机器人。

这里我们把他改造成 arduino 版本的，提供更多丰富的功能。目前主要用 arduino 和 ros 两种软件平台来做，可以用于学习 arduino 或者 ros 编程，同时也可以将 ros 于 arduino 结合。

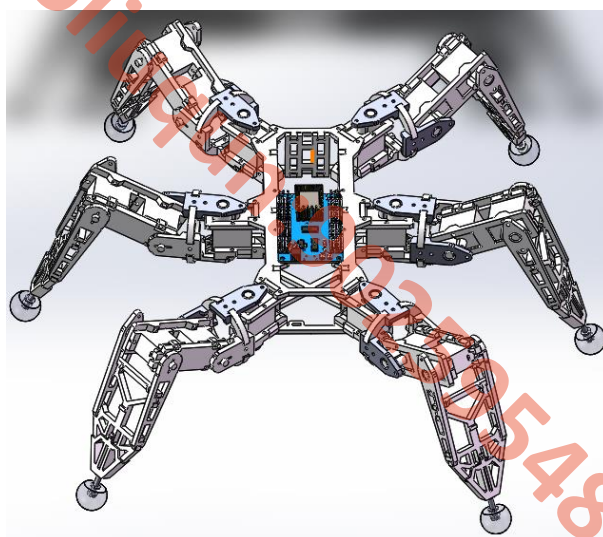
在这个六足机器人的制作中，我们摒弃传统的开发板+舵机驱动板的方式。开发板+舵机驱动板的方式一个方面是集成度不高，杜邦线接线麻烦且可靠性差。另一个方面，两块板子成本加起来也要将近 200 块，实在是浪费。所以我们自己做了两版舵机驱动板，用 esp8266 或者 esp32 做主控，更低的成本，更强大的性能，更丰富的功能且支持 WiFi 无线遥控，实在是香！他们分别是采用 esp8266 做主控的 Baize\_Servo8266 和采用 esp32 做主控的 Baize\_Servo32，如下图：



实物图



两个版本的机器人外观如下图所示：

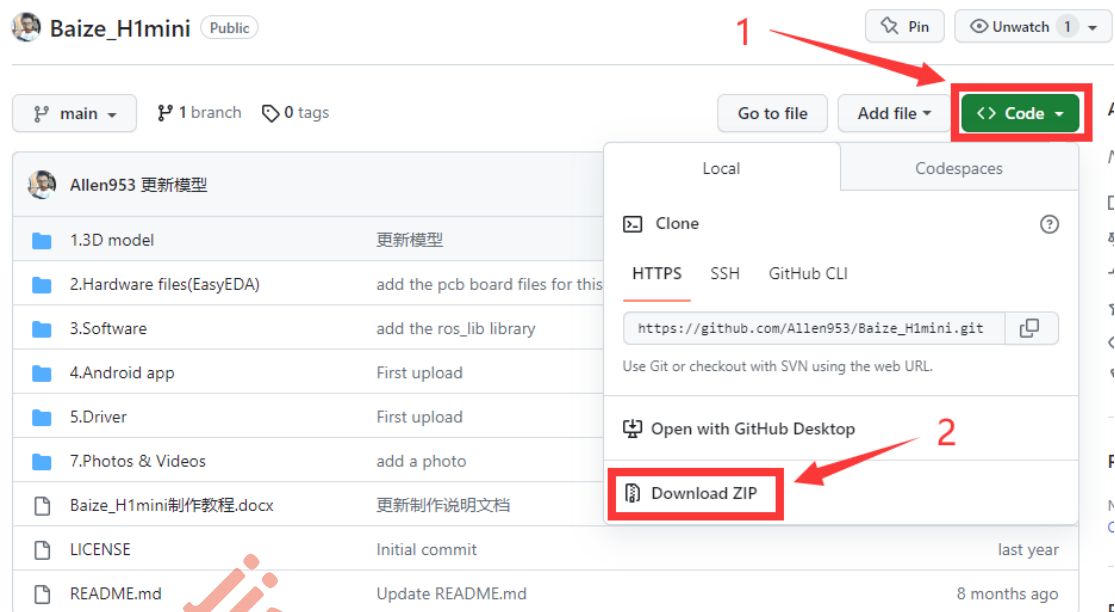


## 2.资料下载

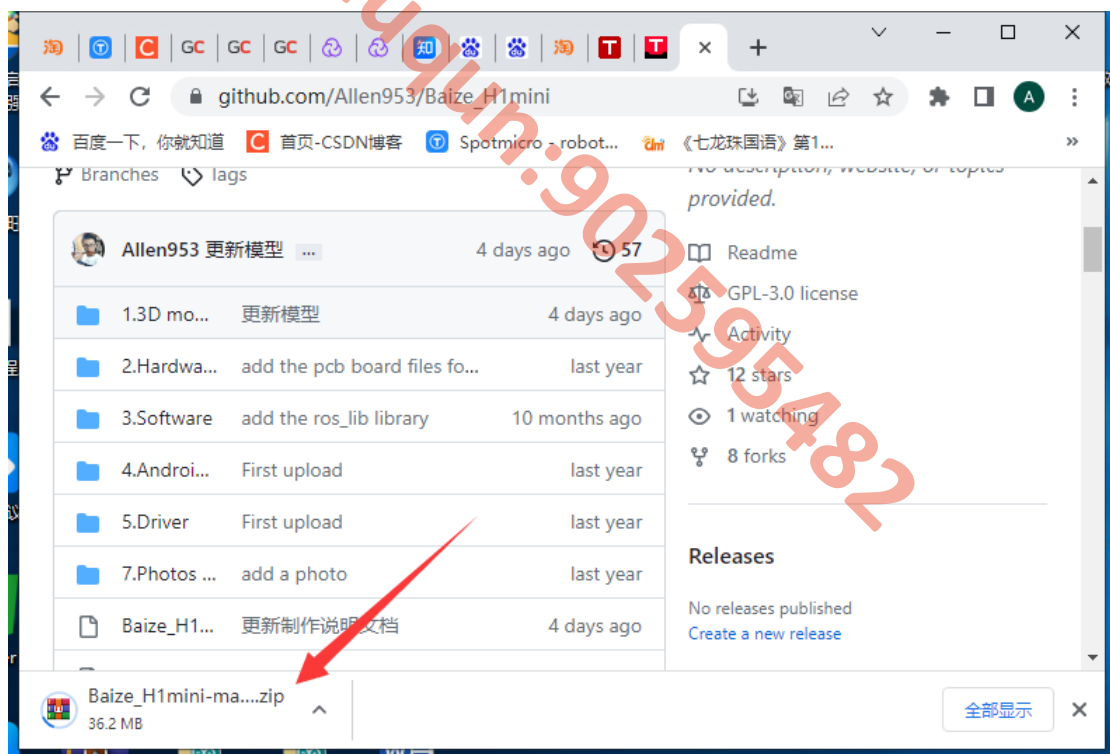
在开始制作该项目前，最好将所有的资料先下载到自己的电脑上，你需要按照以下步骤来下载该项目需要用到的全部资料：

首先访问项目网页：[https://github.com/Allen953/Baize\\_H1mini](https://github.com/Allen953/Baize_H1mini)





接着，你的浏览器会开始下载资料，左下角会出现正在下载的资料压缩包并显示已下载数据大小：



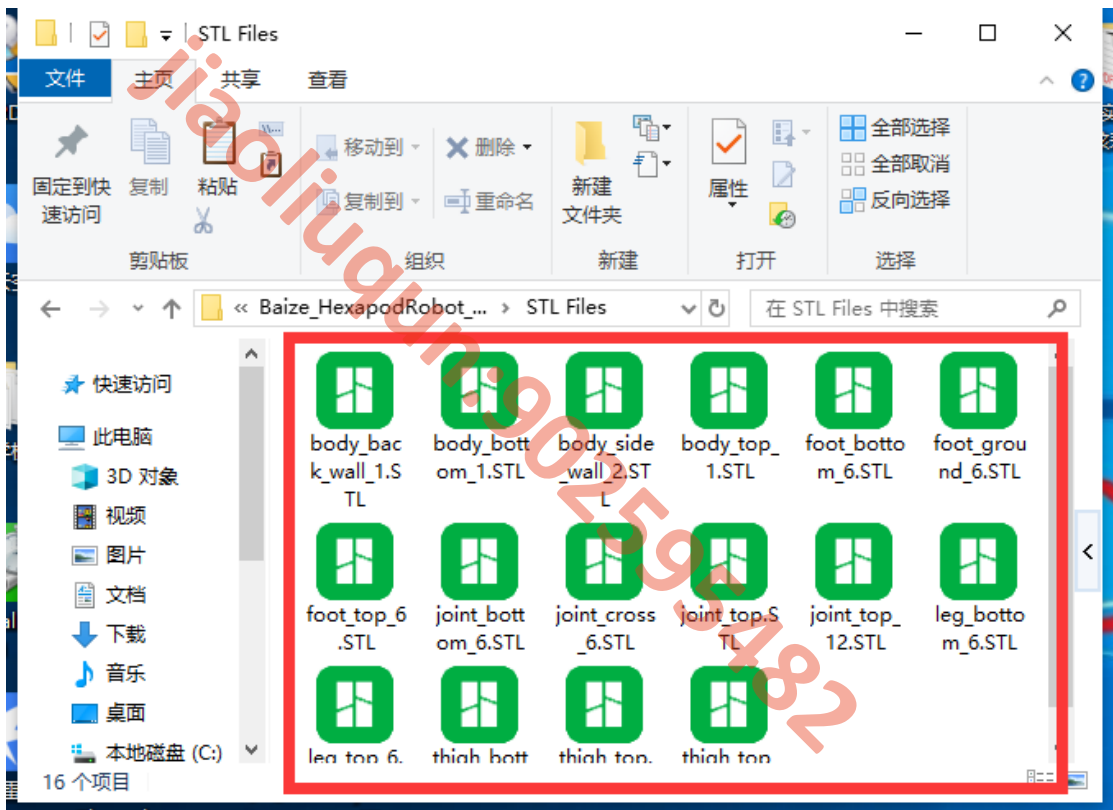
耐心等待下载完成并解压压缩包即可提取出完整的数据。

## 二.Baize\_H1mini 机器人零件制作准备

### 1.机械结构零件打印制作

**重要提示：**自己使用打印机打印零件时，对于新手来说肯定会由于参数设置问题将零件打废，因此要仔细配置参数确保参数正确。对于需要添加支撑的零件，一定要记得添加支撑。

首先要找到零件的图纸，打开我们刚刚下载的项目资料文件夹，找到如下路径 1. 3D model\Baize\_HexapodRobot\_Arduino\_3DPrinter\STL Files，可以看到该文件夹下有很多可以打印的 STL 格式图纸，如下图红框中所示：



接下来就可以开始打印了，我这里用的是拓竹 P1P 型号 3D 打印机，因此下面的打印教程也是用该款型号打印机制作的，如果你用的其他型号的 3D 打印机，则可以一次做参考并参照自己 3D 打印机的说明书进行打印。

下载 Bambu Studio: <http://bambulab.com/download>

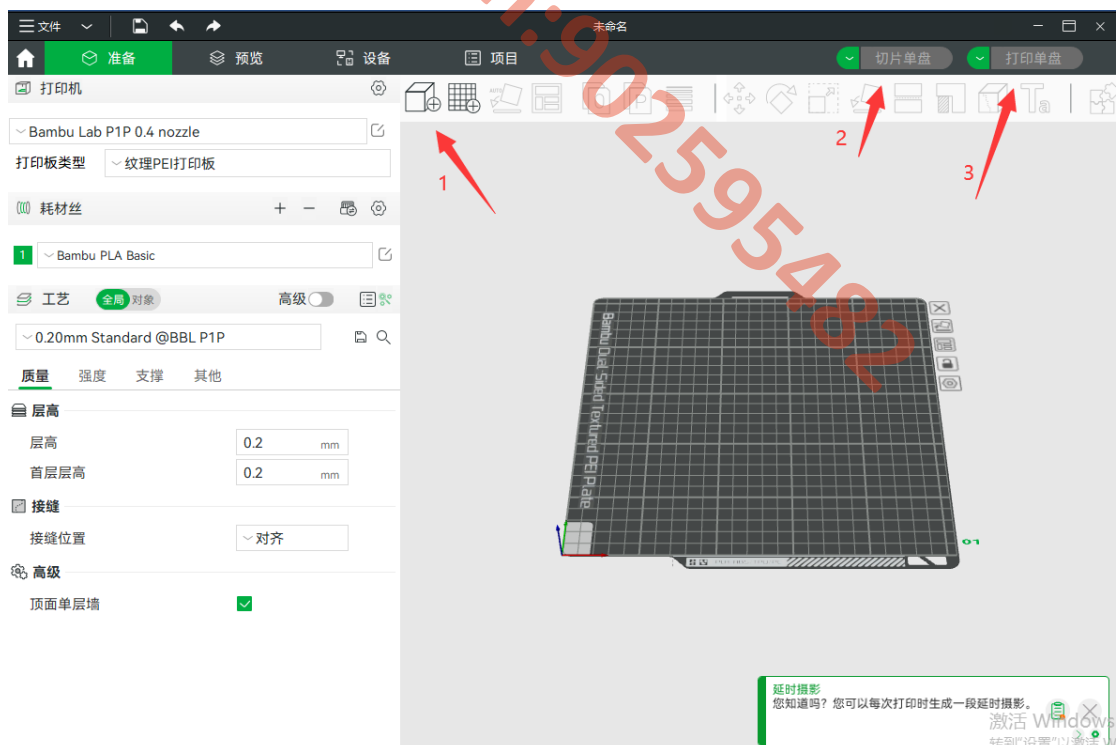
登录 Bambu Studio，新建或打开一个项目。



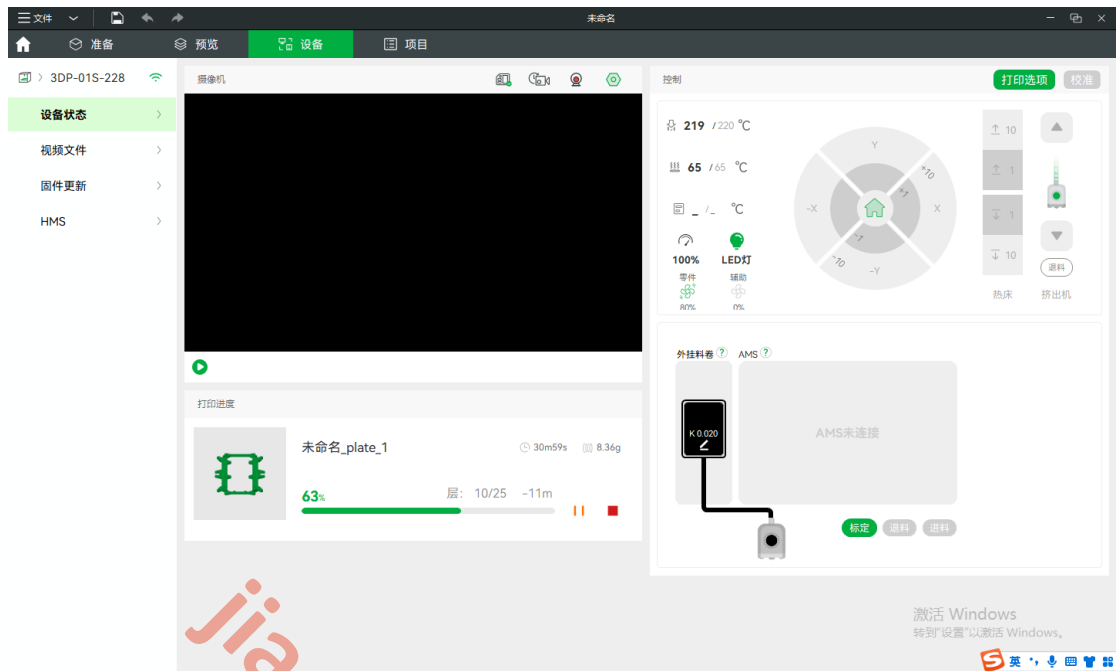
添加一个模型进行模型切片。

完成模型切片后，选择打印机并发送打印。

打印板类型选择纹理 PEI 打印板



打印材料是啥？ PLA 3D 打印材料一般有 PLA 和 ABS 两款材料  
这里需要选择打印材料。



## 2.主控板 PCB 制作及焊接

## 3.标准件准备（包括螺丝和电器原件等）

### 1.. 舵机 x18



### 2. 电池

电池要买 18650 大功率电池，单节电池是 3.7V 左右（之所以说是 3.7V 左右而不是 3.7V 是因为电池电压随着使用过程会发生改变，当单节电池充满电时大多数在 4.2V 左右，当电池放电到没电状态之后，电池电压会下降到 3.7V 甚至 3.2V，电池电压下降到 3.2V 的时候就要充电了，如果电池电压下降太多，可能



会发生永久性的损坏充不了电，因此在使用过成功要及时为电池充电以免电池损坏)，这里我们两节串联起来使用，加起来就是 7.2V 左右。



### 3. 螺丝等标准件

螺丝一共有以下几种：下面是螺丝型号及数量

M2\*30——36 个

M2\*8——12 个

M2\*15——6 个

M2\*5——42 个

圆柱销  $\phi 4 \times 12$ ——18 个

### 4. LM2596S 压降模块

这个压降模块可以将 2 节 18650 电池的 7.4~8.4V 电压降至 5V 电压，由于我们的板子供电电压是 5V，而且舵机供电电压也是 5V，因此需要用到这个模块。



需要注意：

1.长时间工作**建议在2.5A以内的电流使用，同时加上散热片(10W以上输出)**；由于是降压模块,为了保证输出稳定,请**保持最小1.5V压差**。

应用案例：

1.车载稳压电源，只需要将本模块输入端接上汽车点烟嘴供电，就可以调节电位器,输出电压就可以在1.25-30V任意调整，为您的手机、MP3、MP4、PSP充电等许多设备供电，非常简单方便。

2.给电子设备供电，当设备需要3-35V供电而手里对应电压电源时，用这个模块就可以方便的把电压调到所需电压，解决困扰。

3.系统工作电压测试，做项目时可以使用本模块调试出各种电压测试系统工作电压范围，非常轻松方便。

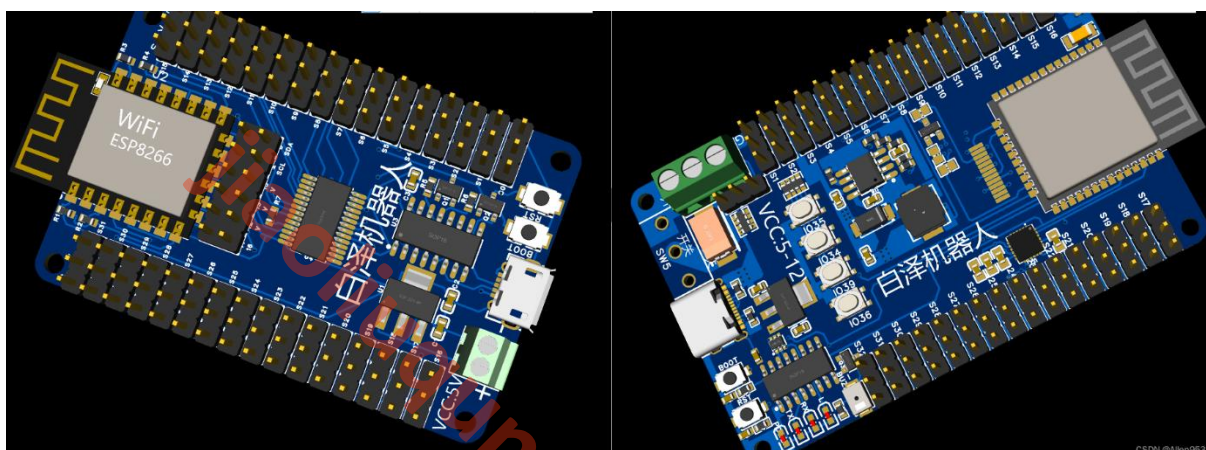
### 三.Baize\_H1mini 机器人组装

#### 1.零件准备

##### 1. 主控板 x1

Baize\_ServoDriver\_esp8266 或者 Baize\_ServoDriver\_esp32

如 下 图 左 边 为 Baize\_ServoDriver\_esp8266 ； 右 边 为 Baize\_ServoDriver\_esp32



建议 arduino 版六足用 Baize\_ServoDriver\_esp8266 而 ROS 版六足用 Baize\_ServoDriver\_esp32。

我们这里是做 arduino 版，所以就默认为 Baize\_ServoDriver\_esp8266。

##### 2.机械零件

链接：[https://github.com/Allen953/Baize\\_H1mini](https://github.com/Allen953/Baize_H1mini)

在如下目录下即可下载到

**Baize\_H1mini** Public

Pin Unwatch 1 Fork 8

main 1 branch 0 tags Go to file Add file <> Code About

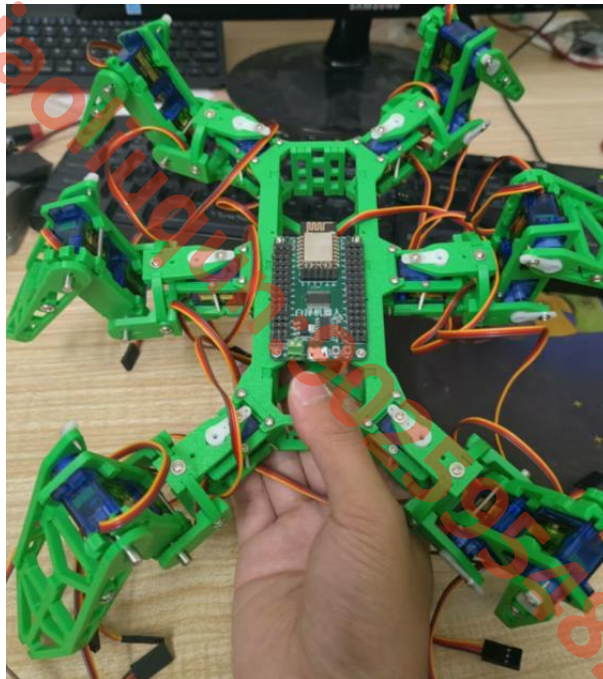
Allen953 添加制作教程文件 d24a9a8 on Apr 28 55 commits

1.3D model	add the original 3D model	last year
2.Hardware files(EasyEDA)	add the pcb board files for this robot	last year
3.Software	add the ros_lib library	10 months ago
4.Android app	First upload	last year
5.Driver	First upload	last year
7.Photos & Videos	add a photo	last year
Baize_H1mini制作教程.docx	添加制作教程文件	3 months ago
LICENSE	Initial commit	last year
README.md	Update README.md	8 months ago

**About**  
No description, website  
Readme  
GPL-3.0 license  
Activity  
12 stars  
1 watching  
8 forks

**Releases**  
No releases published  
[Create a new release](#)

**Packages**



## 2.组装教程

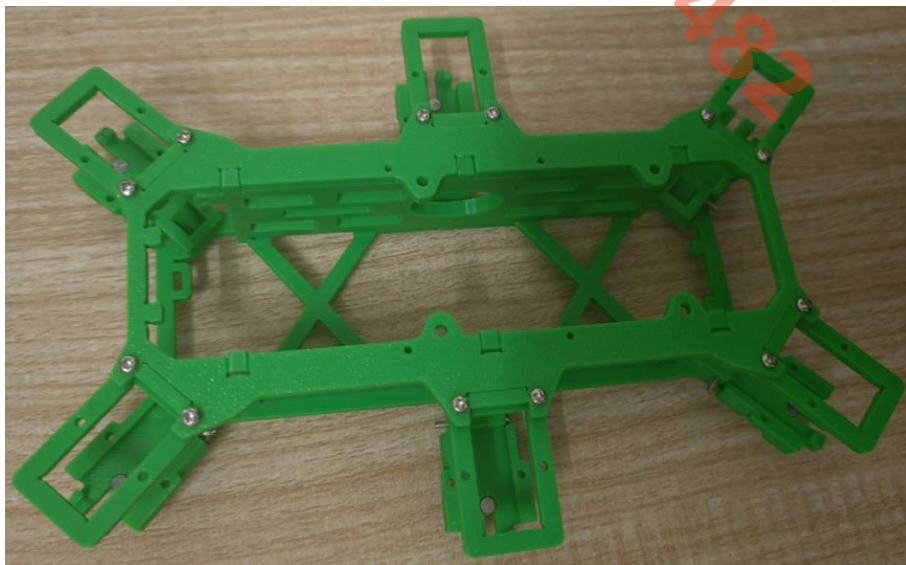
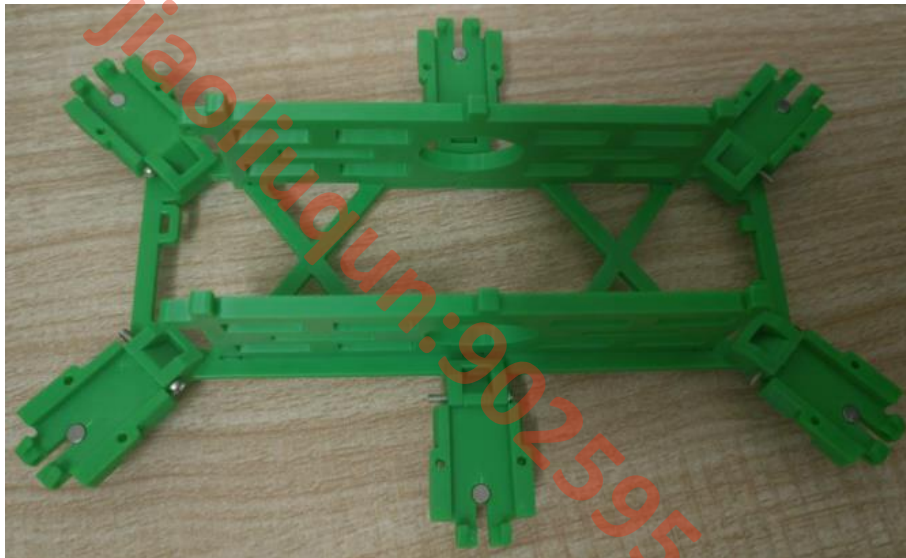
找到下面的  $\phi 4 \times 12$  的金属圆柱销，然后将圆柱销用力塞进或者敲进打印的零件里面



如下图所示，就证明装好了。







### 3.电路焊接

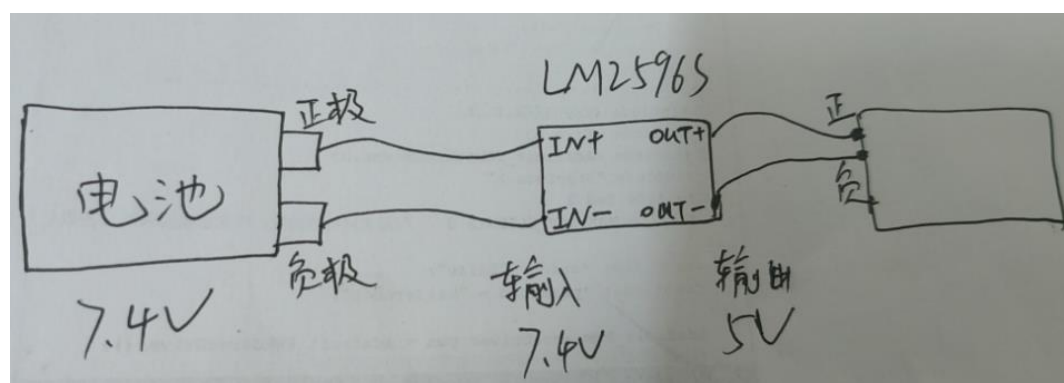
首先，需要找一个 18650 电池的电池盒，由于电池盒上面直接引出来了两条线，这两条电源线是先输入到 LM2596S 这个降压模块的，需要先降压到 5V。由于直接把电池盒上面的两条电源线焊接在 LM2596S 模块上那就进行了通电，而无法断电，因此需要一个开关。我们现在的做法是在电池的输出端和 LM2596S 的输入端各焊一个插接口的电源线，这样可以通过插拔的方式给整个电路通电和断电，起到开关的作用。

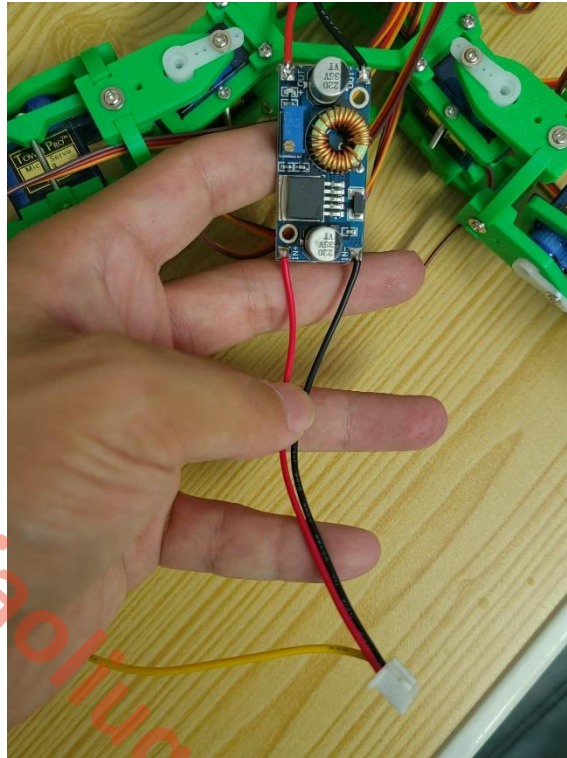


接着需要焊接 LM2596S 的输入和输出线

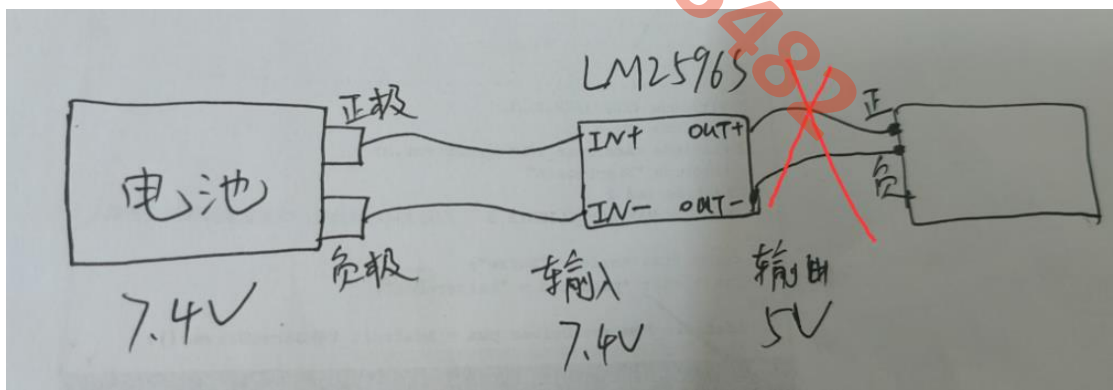
LM2596S 降压模块的作用就是把电压降至 5V，电流从电池里面出来的时候是 7.4V，进入该模块的时候也是 7.4V，从该模块出来的时候，电压就变成 5V 了，就可以输入到机器人板子上了。整个电源部分工作原理如下图所示，LM2596S 模块的 IN+ 表示输入正极，IN- 表示输入负极，也就是电池的电从这两个地方输入到模块；OUT+ 表示的是输出负极，OUT- 表示的是输出正极，输出的 5V 电压从这里流出去。

电源线红色是正极，黑色是负极。

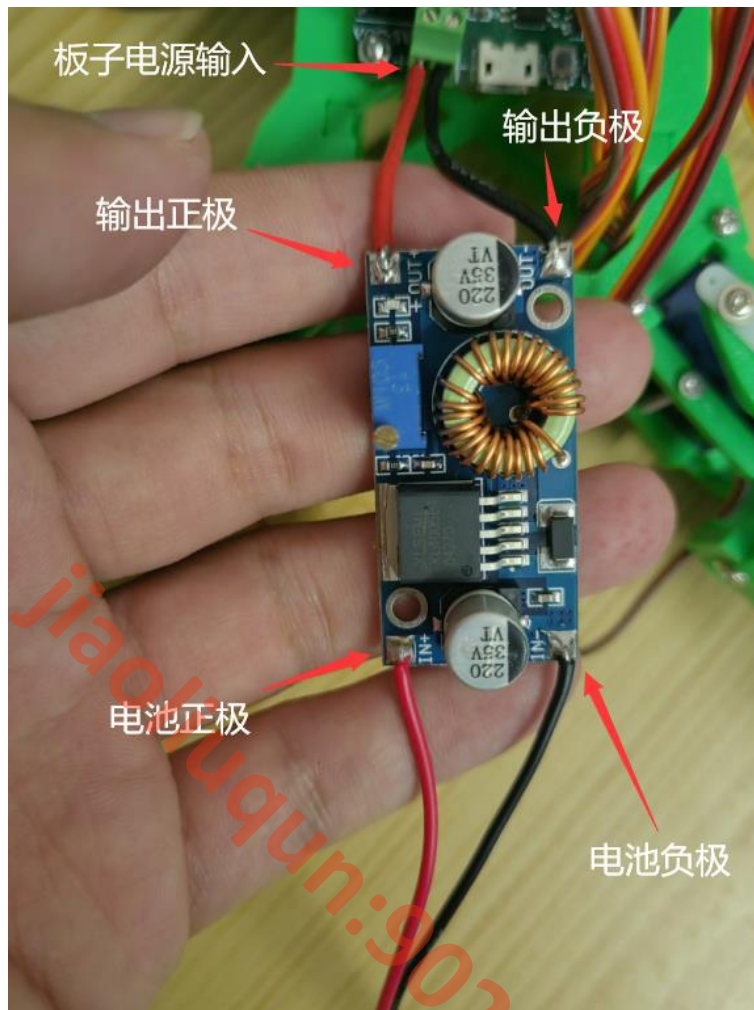




接下来，就可以直接把电池的输出接口插在 LM2596 的输入接口了，这个时候先别着急把 LM2596S 模块的输出接口接到舵机控制板上面，如下图所示，红色打叉号的两条线先断开别接上。因为 LM2596S 模块的输出电压我们还没调节，默认情况下 LM2596S 模块的输出电压太高，我们如果直接将 LM2596S 的输出线路接入舵机控制板，则会烧毁我们的舵机控制板，因此需要先行调节 LM2596S 模块的输出电压。

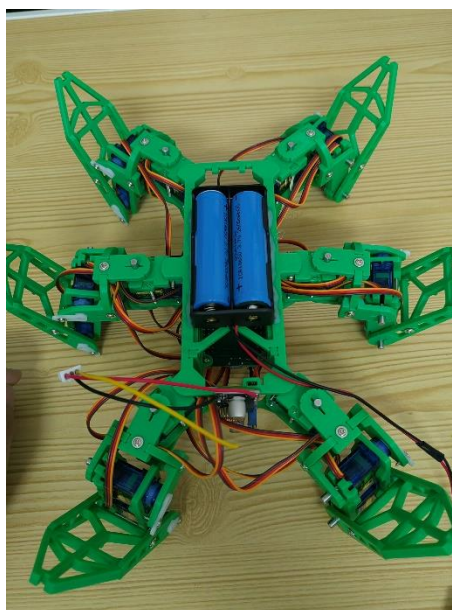






如下图所示，首先需要将电压调至 5V。接着再给机器人通电，如果通电测试正常，则可以将机器人电源板固定好。

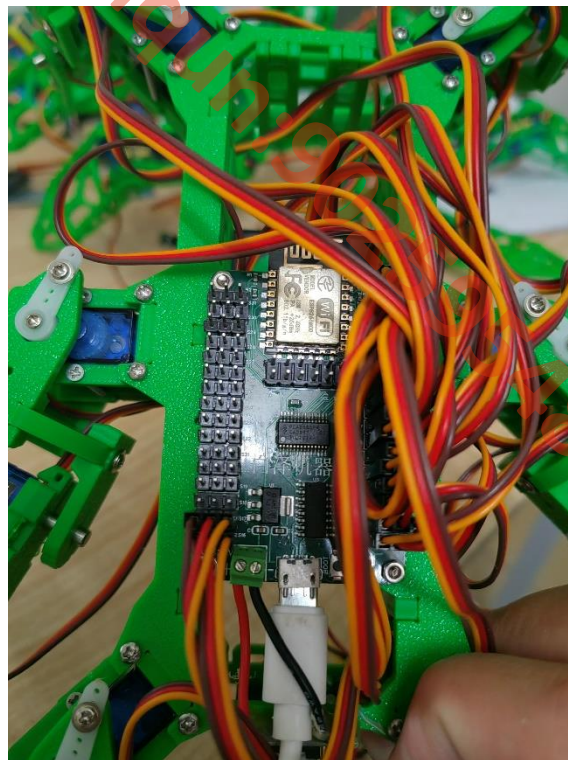
固定电池：



固定压降模块，注意别让该模块挡住主控板的 micro usb 口，我这里就挡住了口，不方便烧录代码了。



固定主板：

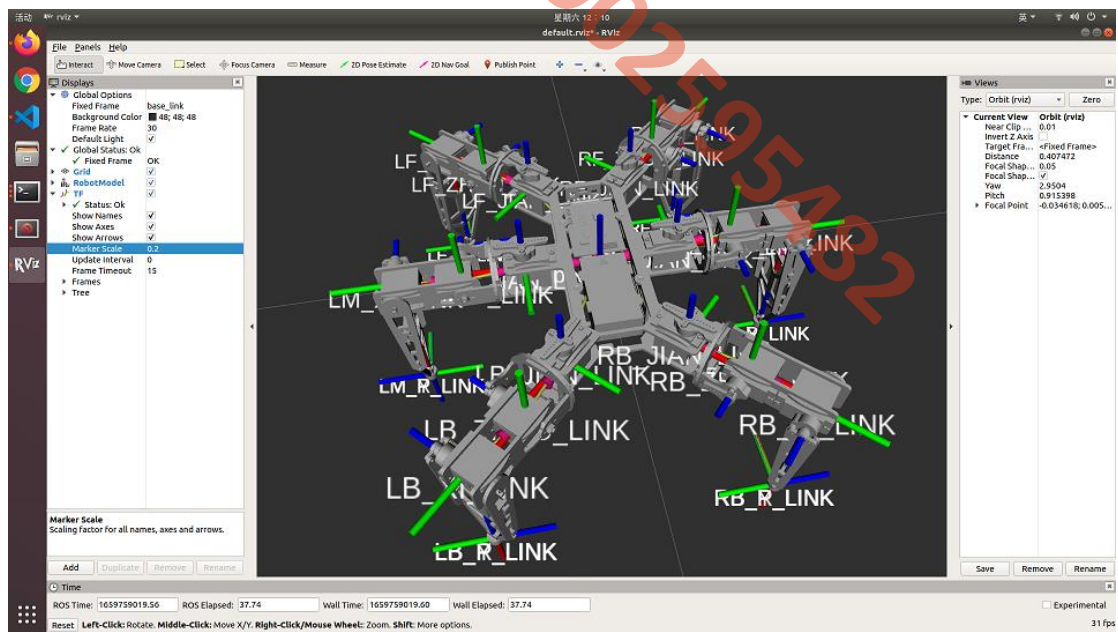
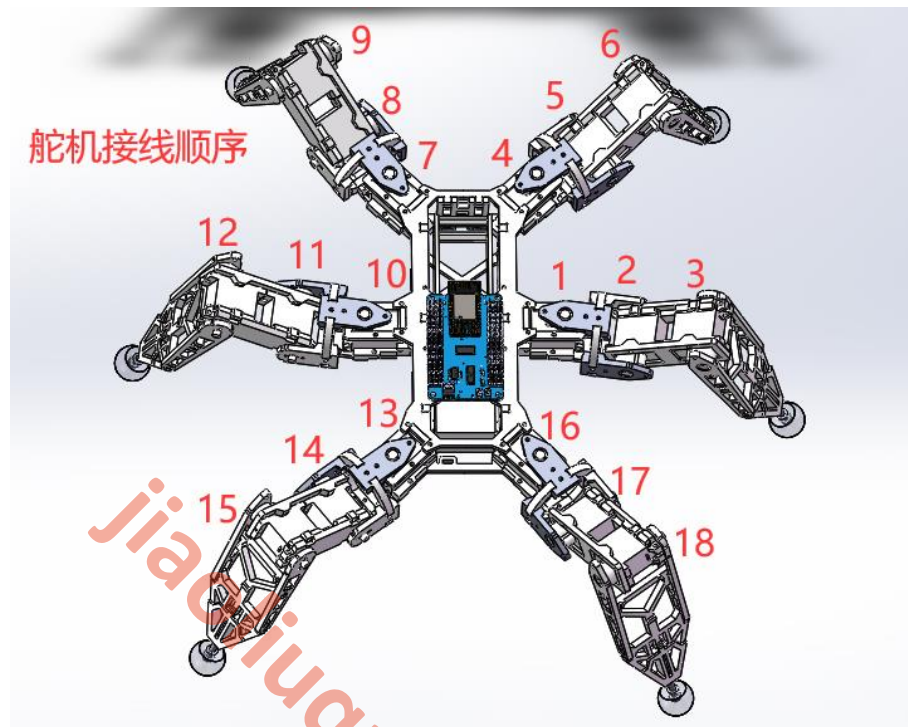


#### 4.舵机接线

接线的话，按照如下图所示进行接线即可。就是右腿第一个舵机接到板子的第一个口上面，注意有的板子是从 0 开始有的板子是从 1 号开始，如果板子上的舵机标号从 0 开始就从 0 开始接线，如果板子上的舵机标号从 1 开始就从 1



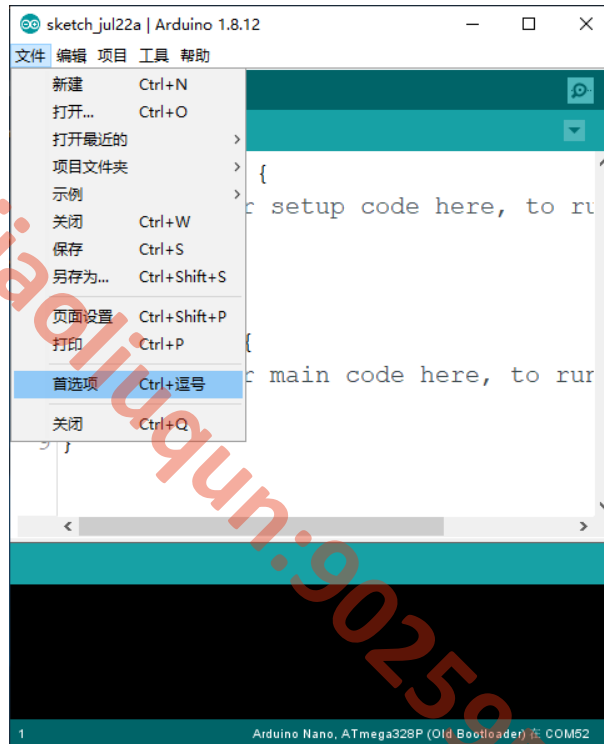
开始接线。



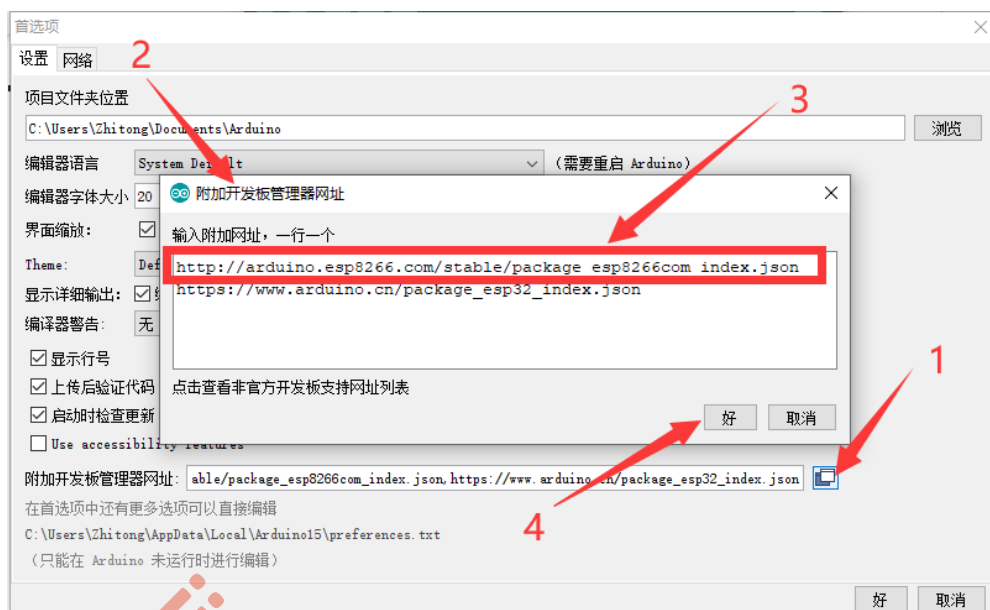
## 四.Baize\_H1mini 机器人代码烧录调试

### 1.Arduino IDE 配置 ESP8266 开发板

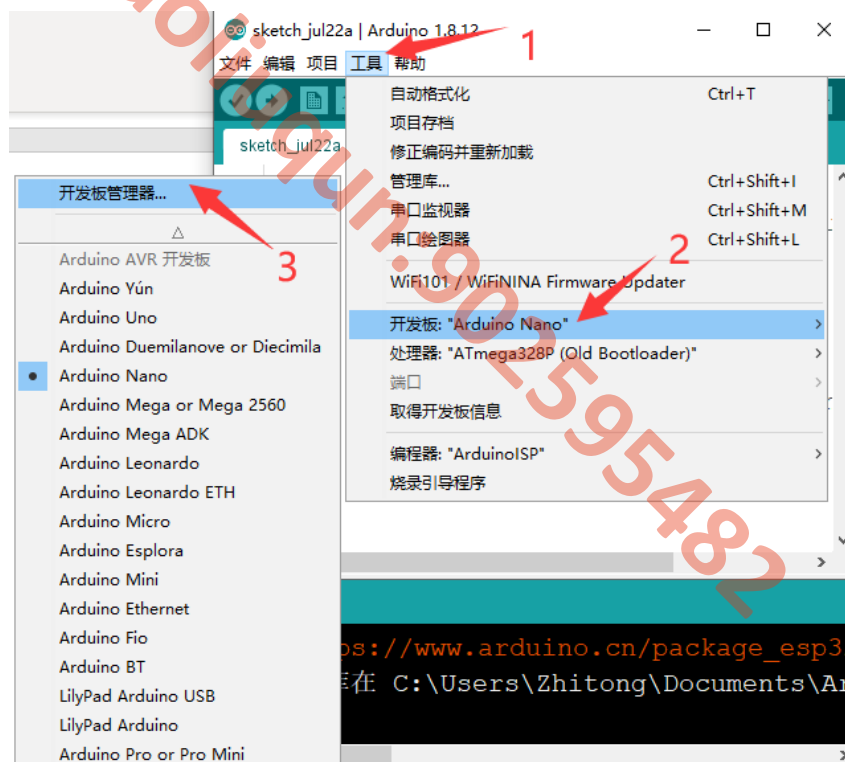
这一步主要是在 arduino ide 里面添加 esp8266 的硬件支持,我们默认安装完 arduino 之后,arduino 里的开发板选项是没有 esp8266 的,因此需要添加硬件支持,让 arduino ide 可以支持 esp8266 开发板。首先打开 Arduino IDE, 点击文件——首选项



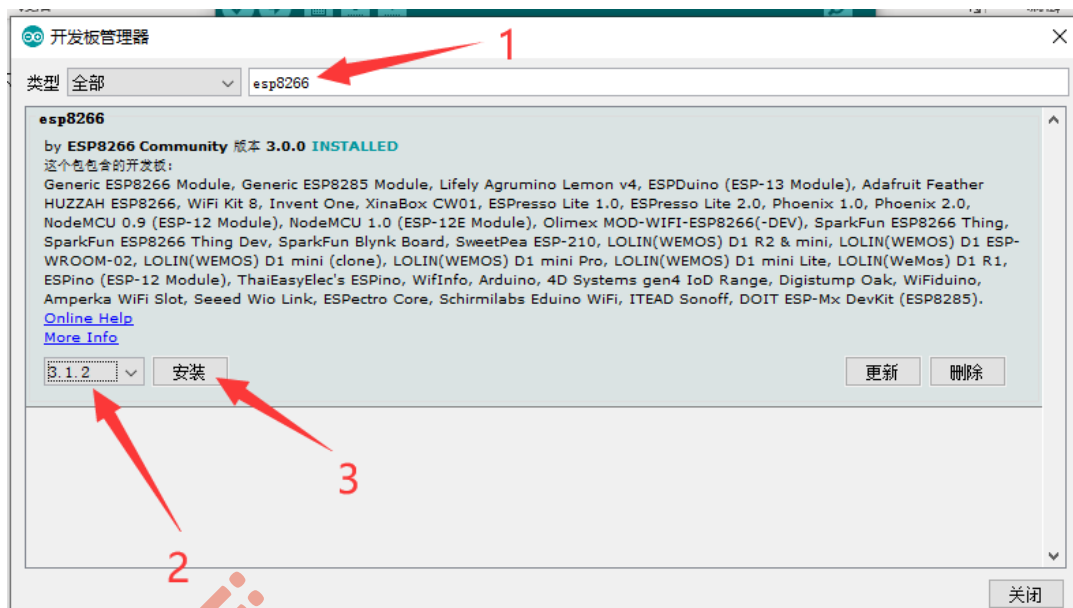
接下来会跳出下方界面,点击箭头 1 所示的那个白色小方框,接着会跳出箭头 2 所示的小窗口,将该网址——[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json), 复制粘贴到小窗口里面,如图 3 所示,保证这个网址在里面就可以了,然后点击箭头 4 所示的好即可。



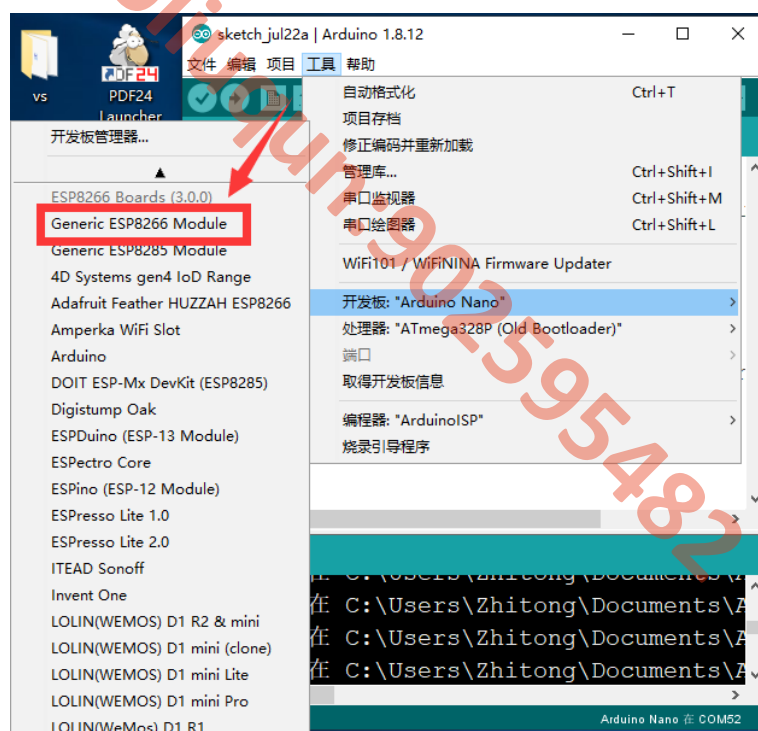
接下来，点击 Arduino IDE 上方的工具——开发板——开发板管理器



接着会跳出来如下窗口，在箭头 1 所示的窗口中输入 **esp8266** 并敲下回车按钮进行开发板搜索，接着将会出现图中的 **esp8266** 选项，点击箭头 2 所示的版本选择最新的版本，根据箭头 3 点击安装即可开始安装，等待进度条完成安装即可。



接着，我们在工具——开发板选项里面，就可以看到 esp8266 开发板了，我们选择如下图所示的开发板即可。



## 2.配置库文件

在配置了开发板之后，还需要配置我们项目需要用到的库文件，这样我们的程序代码才能正常编译烧录。

编译是程序编程机器代码的过程，计算机只能直接理解机器代码，我们编写的程序它是不能直接理解的，因此我们写的程序都需要进行编译计算机才能理解。

烧录是将编译后的程序从电脑上上传到机器人板子上的芯片里面的过程，只有将程序上传

到机器人的板子上面，程序才能正确执行并且控制机器人运行。  
找到如下文件夹：

此电脑 > 新加卷 (F:) > 我的机器人项目 > Baize\_H1mini > 3.Software > libraries >

名称	修改日期	类型	大小
Adafruit_PWM_Servo_Driver_Library	2023/7/19 20:32	文件夹	
ros_lib	2023/7/19 20:32	文件夹	

将其复制到自己电脑的如下文件夹里面：

> 此电脑 > 文档 > Arduino > libraries

名称
SoftwareSerial
SPI
StandardCplusplus-master
TFT_eSPI-master
TinyLiquidCrystal
U8g2
U8glib
units-main
usbdrv
VUsbDevice
VUsbKeyboard
...

### 3.烧录代码

烧录代码前，首先根据如下步骤，一步一步通过文件路径找到我们需要的代码文件。

首先打开 3. Software 这个文件夹

名称	修改日期	类型	大小
.git	2023/5/4 13:31	文件夹	
1.3D model	2023/5/4 13:31	文件夹	
2.Hardware files(EasyEDA)	2023/5/4 13:31	文件夹	
3.Software	2023/5/4 13:31	文件夹	
4.Android app	2023/7/8 22:35	文件夹	
5.Driver	2023/5/4 13:31	文件夹	
7.Photos & Videos	2023/5/4 13:31	文件夹	
Baize_H1mini制作教程.docx	2023/7/8 22:38	Microsoft Word ...	4,426 KB
LICENSE	2023/5/4 13:31	文件	35 KB
README.md	2023/5/4 13:31	稻壳阅读器 Mark...	2 KB

然后打开 Arduino 主程序这个文件夹



名称	修改日期	类型	大小
Arduino测试代码	2023/5/4 13:31	文件夹	
Arduino主程序	2023/5/4 13:31	文件夹	
libraries	2023/5/4 13:31	文件夹	
ROS_package	2023/5/4 13:31	文件夹	
固件烧录工具	2023/5/4 13:31	文件夹	

接着再打开第一个 Baize\_HexapodRobot\_Arduino 文件夹

名称	修改日期	类型	大小
Baize_HexapodRobot_Arduino	2023/5/4 13:31	文件夹	
SubscribeJointTopicViaWiFi_Baize_ServoDriver_esp32	2023/5/4 13:31	文件夹	
SubscribeJointTopicViaWiFi_BaizeServoDriverEsp8266	2023/5/4 13:31	文件夹	

然后找到 Baize\_HexapodRobot\_Arduino.ino 这个程序文件，找到之后可以双击直接打开。

名称	修改日期	类型	大小
Baize_HexapodRobot_Arduino.ino	2023/5/4 13:31	Arduino file	8 KB
StepData.h	2023/5/4 13:31	C Header File	34 KB

打开后的界面如下图所示：

```

1  /*****
2   主板: Baize_ServoDriver_esp8266
3   功能: UnderWaterHexapodRobot水下六足机器人Arduino程序
4   引脚: SDA:21  SCL:22
5   对于ARDUINO UNO, SCL:A5, SDA:A4
6   Designer: Allen
7   E-mail:953598974@qq.com
8   Date:2022-08-22
9  *****/
10 #include <ESP8266WiFi.h>
11 #include <Wire.h>
12 #include <Adafruit_PWMServoDriver.h>
13 #include "StepData.h"
14 #define led 2
15 #define MAX_SRV_CLIENTS 3 //最大同时联接数，即你想要接入的设备数量，
16
17 const char *ssid = "Baize";
18 const char *password = "baizerobot";
19
20 Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
  
```

上传成功。

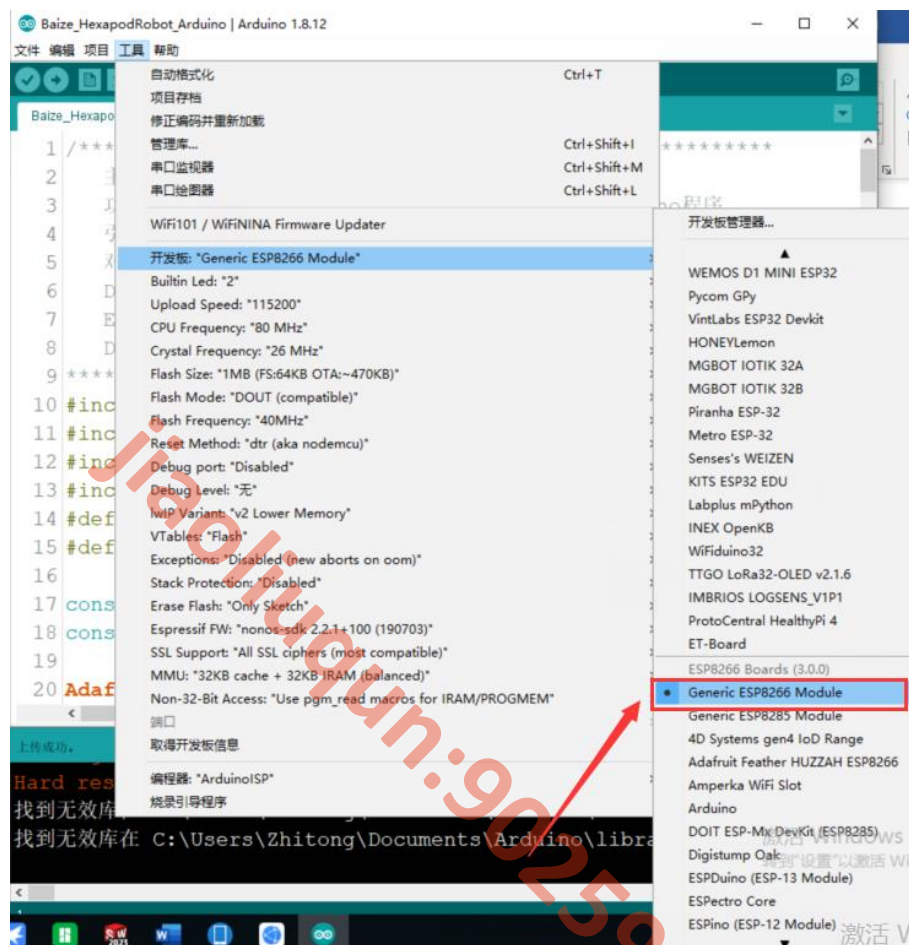
Hard resetting via RTS pin...

找到无效库在 C:\Users\Zhitong\Documents\Arduino\libraries\eigen-main: 激活 Win...

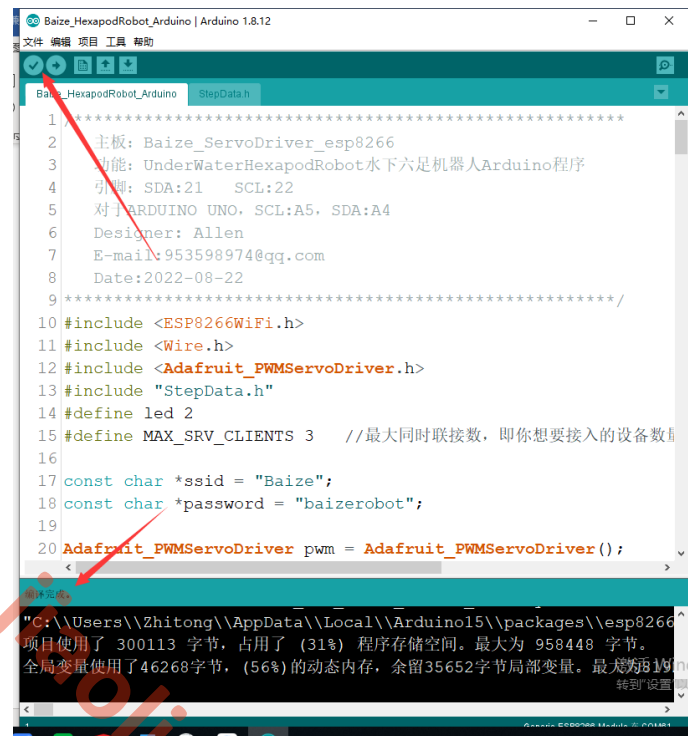
找到无效库在 C:\Users\Zhitong\Documents\Arduino\libraries\eigen-main: 转到“设置”以...

接下来，我们选择开发板类型，然后通过软件的验证功能看程序代码是否有格式错误。

选择开发板类型如下：



然后点击软件界面的验证按钮，可以看到如图，下面出现编译完成的提示，说明我们的代码没有问题，并且已经通过了编译。



接下来,我们将开发板通过 Micro USB 线连接到电脑上。

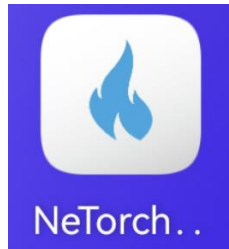
注意:当通过 Micro USB 线连接开发板和电脑时,开发板上不能连接任何舵机,否则可能烧坏电脑主板。

接下来,点击上传按钮,即可把程序上传到开发板上。



## 五.Baize\_H1mini 机器人手机 APP 遥控配置

在测试前，首先需要给你的手机安装一个遥控软件并且进行相应的设置，我们这里安装的遥控软件名字是 NeTorch，如果你的应用市场可以直接搜到的话，可以直接在应用市场进行安装。



如果应用市场搜索不到的话，可以将这个软件安装包发送到自己的安卓手机上，然后再进行安装。

软件安装包位置如下图所示，首先找到 4.Android app 这个文件夹。

.git	2023/5/4 13:31	文件夹	
1.3D model	2023/5/4 13:31	文件夹	
2.Hardware files(EasyEDA)	2023/5/4 13:31	文件夹	
3.Software	2023/5/4 13:31	文件夹	
4.Android app	2023/7/8 22:35	文件夹	
5.Driver	2023/5/4 13:31	文件夹	
7.Photos & Videos	2023/5/4 13:31	文件夹	
Baize_H1mini制作教程.docx	2023/7/8 21:29	Microsoft Word ...	4,407 KB
LICENSE	2023/5/4 13:31	文件	35 KB
README.md	2023/5/4 13:31	稿壳阅读器 Mark...	2 KB

如下图所示的 NeTorch.apk 就是我们需要找到的手机遥控软件安装包，你可以通过 QQ 或者微信将安装包发送到手机上进行安装。

名称	修改日期	类型	大小
NeTorch.apk	2023/7/8 22:32	Kuaizip.apk	2,514 KB
蓝牙调试器.apk	2023/5/4 13:31	Kuaizip.apk	2,464 KB

安装之后打开如下图所示：



接下来，需要你用自己的手机开个热点。

热点名字 (SSID) 是: Baize

热点密码 (PASS) 是: baizerobot

接下来需要给机器人通电，机器人通电之后，可看到主板上有个蓝灯亮起来了。

如果蓝灯过几秒之后变为闪烁状态，则代表着机器人主板已经连接到了手机热点，网络连接正常。

接下来可以设置手机遥控 APP 进行遥控了。

首先需要你查看一下机器人的 IP 地址，点击个人热点之后，可以看到有一个设备连接了你的手机，这里可以看到如下图所示界面，我们可以看到这个 IP 地址为: 192.168.43.227，这里需要你先记下这个 IP 地址。





然后打开 Netorch 软件界面，进行相应的设置：



然后我们设置按钮：

首先点击按钮模式，接下来长按按钮就可以进行按钮的设置了。我们点开按钮把相应的前后左右设置好，按照如下图完成按钮的设置。



控制指令表如下：

控制指令表

指令	动作	指令	动作
a	前进行走	f	向左横移
b	后退行走	g	向右横移
c	原地左转	h	步态切换
d	原地右转	i	身高切换
e	停止		

后续会更新更多动作指令！



接下来便可以进行遥控操作测试了，点击一下前进的按钮，是不是已经发现机器人开始前进了呢？

接下来可以愉快的玩耍了。

## 六.白泽舵机驱动板

这是一块 32 路舵机驱动板，做舵机机器人非常方便。主控用的 esp8266。

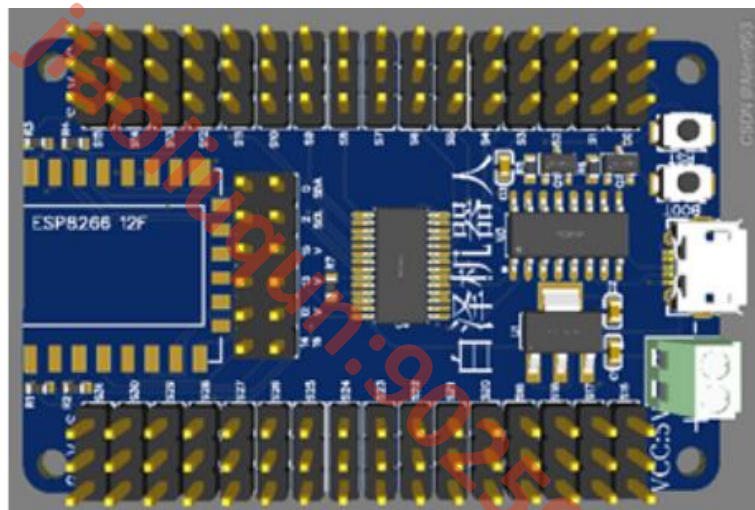
电源用的 5V 的电池，电池直接给舵机供电，经过压降之后产生 3.3V 电压给 esp8266 模组供电。

这块板儿主要是用于控制 sg90 舵机，就是那种蓝色舵机，或者 mg90s 舵机。

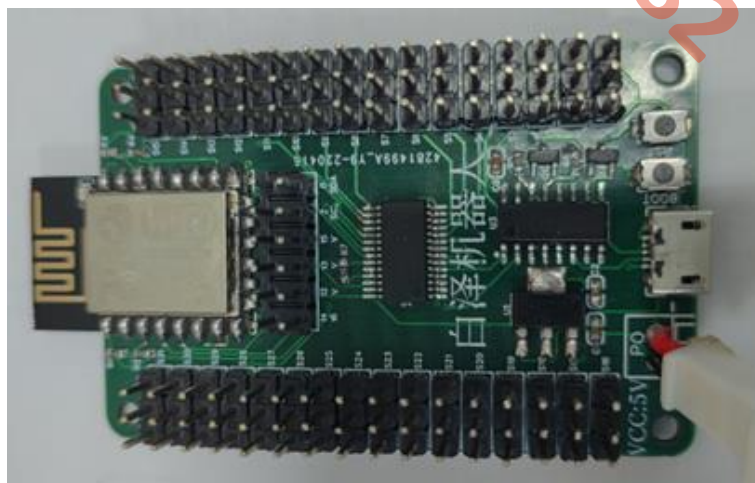
后续会升级，升级后可以控制 5-12V 的 pwm 舵机。

这样就可以控制高压的舵机，比如型号为 mg996r 或 mg995 的舵机。

3D 图片如下图所示：



实物图如下：



测试代码：

### 例程 1：控制 32 路舵机匀速旋转从 0-180 度

```
/******  
    主板：Baize_ServoDriver_esp8266          开源链接：  
https://github.com/Allen953/Baize_ServoDriver_esp8266  
    功能：控制 32 路舵机匀速旋转从 0-180 度  
    引脚：SDA:21  SCL:22  
    Designer: Allen  
    E-mail:953598974@qq.com  
    Date:2022-08-24  
*****/  
  
#include <Wire.h>  
#include <Adafruit_PWMServoDriver.h>  
  
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();  
Adafruit_PWMServoDriver pwml = Adafruit_PWMServoDriver(0x41);  
  
#define SERVOMIN  102           //0.5/20 * 4096 = 102  
#define SERVOMAX  512           //2.5/20 * 4096 = 512  
  
void setup() {  
    Serial.begin(115200);  
    Serial.println();  
    Serial.println("16 channel Servo test!");  
  
    pwm.begin();  
    pwml.begin();  
    pwm.setPWMFreq(50); // Analog servos run at ~50 Hz updates  
    pwml.setPWMFreq(50); // Analog servos run at ~50 Hz updates  
  
    delay(200);  
    for(int i=0;i<16;i++)  
    {  
        pwm.setPWM(i, 0, SERVOMIN); // 驱动 32 路舵机转到 0 度  
        pwml.setPWM(i, 0, SERVOMIN); // 驱动 32 路舵机转到 0 度
```

```

    }
    delay(2000);
    for (uint16_t pulselen = SERVOMIN; pulselen < SERVOMAX; pulselen++) {
        for(int i=0;i<16;i++)
        {
            pwm.setPWM(i, 0, pulselen);
            pwml.setPWM(i, 0, pulselen);
        }
        delayMicroseconds(200);
        ESP.wdtFeed();           //喂狗防止复位
    }
}

void loop() {
}

```

## 例程 2：控制 32 路舵机匀速旋转从 0-180 度，再从 180-0 度

```

/*****
    主板：Baize_ServoDriver_esp8266      开源链接：
https://github.com/Allen953/Baize\_ServoDriver\_esp8266
    功能：控制 32 路舵机匀速旋转从 0-180 度，再从 180-0 度
    引脚：SDA:21   SCL:22
    Designer: Allen
    E-mail:953598974@qq.com
    Date:2022-08-24
*****/

#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
Adafruit_PWMServoDriver pwml = Adafruit_PWMServoDriver(0x41);

#define SERVOMIN  102           //0.5/20 * 4096 = 102
#define SERVOMAX  512          //2.5/20 * 4096 = 512

void setup() {
    Serial.begin(115200);
    Serial.println("16 channel Servo test!");
}

```



```

    pwm.begin();
    pwml.begin();
    pwm.setPWMFreq(50); // Analog servos run at ~50 Hz updates
    pwml.setPWMFreq(50); // Analog servos run at ~50 Hz updates
}

void loop() {
    for (uint16_t pulselen = SERVOMIN; pulselen < SERVOMAX; pulselen++)
    {
        for(int i=0;i<16;i++)
        {
            pwm.setPWM(i, 0, pulselen);
            pwml.setPWM(i, 0, pulselen);
        }
        delayMicroseconds(200);
        ESP.wdtFeed(); //喂狗防止复位
    }
    for (uint16_t pulselen = SERVOMAX; pulselen > SERVOMIN; pulselen--)
    {
        for(int i=0;i<16;i++)
        {
            pwm.setPWM(i, 0, pulselen);
            pwml.setPWM(i, 0, pulselen);
        }
        delayMicroseconds(200);
        ESP.wdtFeed(); //喂狗防止复位
    }
}

```

### 例程 3：控制 32 路舵机从 0，90，180，90，0 度之间循环。

```

/*****
    主板：Baize_ServoDriver_esp8266          开源链接：
https://github.com/Allen953/Baize\_ServoDriver\_esp8266
    功能：控制 32 路舵机从 0，90，180，90，0 度之间循环
    引脚：SDA:21   SCL:22
    Designer: Allen
    E-mail:953598974@qq.com
    Date:2022-08-24
*****/

```

```

#include <Wire.h>

```

```

#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
Adafruit_PWMServoDriver pwml = Adafruit_PWMServoDriver(0x41);

#define SERVOMIN 102           //0.5/20 * 4096 = 102
#define SERVOMID 307          //(102+512)/2=307
#define SERVOMAX 512          //2.5/20 * 4096 = 512

//pwm.setPWM(i, 0, pulse);i=0~15 对应第 0-15 个舵机; pwml.setPWM(j, 0,
pulse);j=0~15 对应第 16-31 个舵机,

void setup() {
  Serial.begin(115200);
  Serial.println("16 channel Servo test!");

  pwm.begin();
  pwml.begin();
  pwm.setPWMFreq(50); // Analog servos run at ~50 Hz updates
  pwml.setPWMFreq(50); // Analog servos run at ~50 Hz updates
}

void loop() {
  //全部舵机转到 0 度
  for(int i=0;i<16;i++)
  {
    pwm.setPWM(i, 0, SERVOMIN);
    pwml.setPWM(i, 0, SERVOMIN);
  }
  delayMicroseconds(200);
  ESP.wdtFeed(); //喂狗防止复位
  delay(1000);
  //全部舵机转到 90 度
  for(int i=0;i<16;i++)
  {
    pwm.setPWM(i, 0, SERVOMID);
    pwml.setPWM(i, 0, SERVOMID);
  }
  delayMicroseconds(200);
  ESP.wdtFeed(); //喂狗防止复位
  delay(1000);
  //全部舵机转到 180 度
  for(int i=0;i<16;i++)
  {

```

```

        pwm.setPWM(i, 0, SERVOMAX);
        pwml.setPWM(i, 0, SERVOMAX);
    }
    delayMicroseconds(200);
    ESP.wdtFeed(); //喂狗防止复位
    delay(1000);
    //全部舵机转到 90 度
    for(int i=0;i<16;i++)
    {
        pwm.setPWM(i, 0, SERVOMID);
        pwml.setPWM(i, 0, SERVOMID);
    }
    delayMicroseconds(200);
    ESP.wdtFeed(); //喂狗防止复位
    delay(1000);
}

```

#### 例程 4：板子连到手机热点

手机需要开一个热点，名字：“Baize”，密码：“baizerobot”。

然后上传这个程序到板子上，在手机上就能看到板子连到手机热点了。

//修改自：

[https://blog.csdn.net/wangke0809/article/details/70146378?ops\\_request\\_misc=%257B%2522request%255Fid%2522%253A%2522163566195516780261935125%2522%252C%2522scm%2522%253A%25220140713.130102334..%2522%257D&request\\_id=163566195516780261935125&biz\\_id=0&utm\\_medium=distribute.pc\\_search\\_result.none-task-blog-2~all~sobaiduend~default-2-70146378.first\\_rank\\_v2\\_pc\\_rank\\_v29&utm\\_term=esp8266%E4%B8%B2%E5%8F%A3%E9%80%8F%E4%BC%A0%E7%A8%B8%E5%BA%8F&spm=1018.2226.3001.4187](https://blog.csdn.net/wangke0809/article/details/70146378?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522163566195516780261935125%2522%252C%2522scm%2522%253A%25220140713.130102334..%2522%257D&request_id=163566195516780261935125&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-2-70146378.first_rank_v2_pc_rank_v29&utm_term=esp8266%E4%B8%B2%E5%8F%A3%E9%80%8F%E4%BC%A0%E7%A8%B8%E5%BA%8F&spm=1018.2226.3001.4187)

//原作者：wangke0809

#include <ESP8266WiFi.h>

#define MAX\_SRV\_CLIENTS 3 //最大同时联接数，即你想要接入的设备数量，8266tcpserver 只能接入五个，哎

const char \*ssid = "Baize";

const char \*password = "baizerobot";

WiFiServer server(8266); //你要的端口号，随意修改，范围 0-65535

WiFiClient serverClients[MAX\_SRV\_CLIENTS];

void setup()

```

{
    Serial.begin(9600);
    delay(10);
    pinMode(16, OUTPUT);
    digitalWrite(16, 0);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
    }
    server.begin();
    server.setNoDelay(true); //加上后才正常些
}

void loop()
{
    blink();

    uint8_t i;
    if (server.hasClient())
    {
        for (i = 0; i < MAX_SRV_CLIENTS; i++)
        {
            if (!serverClients[i] || !serverClients[i].connected())
            {
                if (serverClients[i]) serverClients[i].stop(); //未连接,
就释放

                serverClients[i] = server.available(); //分配新的
                continue;
            }

            WiFiClient serverClient = server.available();
            serverClient.stop();
        }
        for (i = 0; i < MAX_SRV_CLIENTS; i++)
        {
            if (serverClients[i] && serverClients[i].connected())
            {
                digitalWrite(16, 0); //有链接存在,就一直长亮

                if (serverClients[i].available())
                {

```

```

        while (serverClients[i].available())
            Serial.write(serverClients[i].read());
    }
}
if (Serial.available())
{
    size_t len = Serial.available();
    uint8_t sbuf[len];
    Serial.readBytes(sbuf, len);
    //push UART data to all connected telnet clients
    for (i = 0; i < MAX_SRV_CLIENTS; i++)
    {
        if (serverClients[i] && serverClients[i].connected())
        {
            serverClients[i].write(sbuf, len); //向所有客户端发
送数据
            delay(1);
        }
    }
}

void blink()
{
    static long previousMillis = 0;
    static int currstate = 0;

    if (millis() - previousMillis > 200) //200ms
    {
        previousMillis = millis();
        currstate = 1 - currstate;
        digitalWrite(16, currstate);
    }
}

```

## 例程 5：发现（校验）板子上的 iic 设备

/\*\*\*\*\*  
 主板：Baize\_ServoDriver\_esp8266                      开源链接：  
[https://github.com/Allen953/Baize\\_ServoDriver\\_esp8266](https://github.com/Allen953/Baize_ServoDriver_esp8266)



功能：扫描 IIC 总线上挂载的设备

引脚：SDA:21 SCL:22

Designer: Allen

E-mail:953598974@qq.com

Date:2022-08-24

\*\*\*\*\*/

```
#include <Wire.h>
```

```
void setup()
```

```
{  
    Wire.begin();  
    Serial.begin(115200);  
}
```

```
void loop()
```

```
{  
    byte error, address;  
    int nDevices;  
    Serial.println();  
    Serial.println("Scanning...");  
  
    nDevices = 0;  
    for(address = 1; address < 127; address++ )  
    {  
        Wire.beginTransmission(address);  
        error = Wire.endTransmission();  
  
        if (error == 0)  
        {  
            Serial.print("I2C device found at address 0x");  
            if (address<16)  
                Serial.print("0");  
            Serial.print(address, HEX);  
            Serial.println(" !");  
  
            nDevices++;  
        }  
        else if (error==4)  
        {  
            Serial.print("Unknown error at address 0x");  
            if (address<16)  
                Serial.print("0");  
            Serial.println(address, HEX);  
        }  
    }  
}
```

```

    }
}
if (nDevices == 0)
    Serial.println("No I2C devices found\n");
else
    Serial.println("done\n");
delay(5000);
}

```

## 例程 6：带电压监测的板子测试程序

升级版，主控 esp32。供电范围 5-12V。当我用 7.4V 的 2S 锂电池给舵机供电时，用这个程序来进行板子测试，由于测试时间较长，担心电池过放，所以通过电压监测，当电压过低时，将不再驱动舵机工作。

```

/*****
对于 ARDUINO UNO，这时 SCL->A5，SDA ->A4
对于 ESP8266，这时 SCL->GPIO5，SDA ->GPIO4
*****/

#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
Adafruit_PWMServoDriver pwml = Adafruit_PWMServoDriver(0x41);

#define SERVOMIN 102           //0.5/20 * 4096 = 102
#define SERVOMAX 512          //2.5/20 * 4096 = 512
//pcb 板测试时，如果是 2S 的锂电池供电，则以此电压为阈值，低于此电压时，
不再驱动舵机运动，保护电池。
float voltage_threshold = 6.5;
int voltage_testpin = 26;
float voltage_b = 0.0;

bool servo_f = 1;

void setup() {
    Serial.begin(115200);
    Serial.println();
}

```

```

Serial.println("16 channel Servo test!");

pwm.begin();
pwml.begin();
pwm.setPWMFreq(50); // Analog servos run at ~50 Hz updates
pwml.setPWMFreq(50); // Analog servos run at ~50 Hz updates

delay(200);
for(int i=0;i<16;i++)
{
    pwm.setPWM(i, 0, SERVOMIN); // 驱动 32 路舵机转到 0 度
    pwml.setPWM(i, 0, SERVOMIN); // 驱动 32 路舵机转到 0 度
}
delay(2000);
for (uint16_t pulselen = SERVOMIN; pulselen < SERVOMAX; pulselen++) {
    for(int i=0;i<16;i++)
    {
        pwm.setPWM(i, 0, pulselen);
        pwml.setPWM(i, 0, pulselen);
    }
    delayMicroseconds(200);
}

void loop() {
    if(servo_f==1)
    {
        for (uint16_t pulselen = SERVOMIN; pulselen < SERVOMAX;
pulselen++) {
            for(int i=0;i<16;i++)
            {
                pwm.setPWM(i, 0, pulselen);
                pwml.setPWM(i, 0, pulselen);
            }
            delayMicroseconds(200);
        }
        //电压检测
        int voltage_sum = 0.0;
        for(int i=0;i<10;i++)
        {
            voltage_sum +=
float(analogRead(voltage_testpin))/4095.0*3.3*4.0;
            Serial.println(analogRead(voltage_testpin));
            delay(100);
        }
    }
}

```

```

    }
    voltage_b = voltage_sum / 10.0;
    Serial.println(voltage_b);
    if(voltage_b<voltage_threshold)
        servo_f = 0;

    for(int i=0;i<16;i++)
    {
        pwm.setPWM(i, 0, SERVOMIN); // 驱动 32 路舵机转到 0 度
        pwml.setPWM(i, 0, SERVOMIN); // 驱动 32 路舵机转到 0 度
    }
    delay(2000);
}
}

```

## 例程 7：串口控制舵机角度

```

/*****
    主板: Baize_ServoDriver_esp8266      开源链接:
    https://github.com/Allen953/Baize_ServoDriver_esp8266
    功能: 串口输入角度, 板子驱动舵机转到该角度
    引脚: SDA:21   SCL:22
    Designer: Allen
    E-mail:953598974@qq.com
    Date:2022-08-24
    *****/
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
//驱动 1~16 或 (0~15) 号舵机
Adafruit_PWMServoDriver pwml = Adafruit_PWMServoDriver(0x41);
//驱动 17~32 或 (16~31) 号舵机

//#define SERVOMIN  102           //0.5/20 * 4096 = 102
//#define SERVOMID  307           //1.5/20 * 4096 = 307
//#define SERVOMAX  512           //2.5/20 * 4096 = 512
//实际测试
#define SERVOMIN  102
#define SERVOMID  327

```

```

#define SERVOMAX 552

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);

  pwm.begin();
  pwml.begin();
  pwm.setPWMFreq(50); // Analog servos run at ~50 Hz updates
  pwml.setPWMFreq(50); // Analog servos run at ~50 Hz updates
}

void loop() {
  // put your main code here, to run repeatedly:
  while(!(Serial.available()>0))ESP.wdtFeed(); //喂狗防止复位
  int pwmsum=Serial.parseInt();

  for(int i=0;i<16;i++)
  {
    pwm.setPWM(i, 0, pwmsum);
  }
  pwml.setPWM(0, 0, pwmsum);
  pwml.setPWM(1, 0, pwmsum);
  delay(100);
}

```

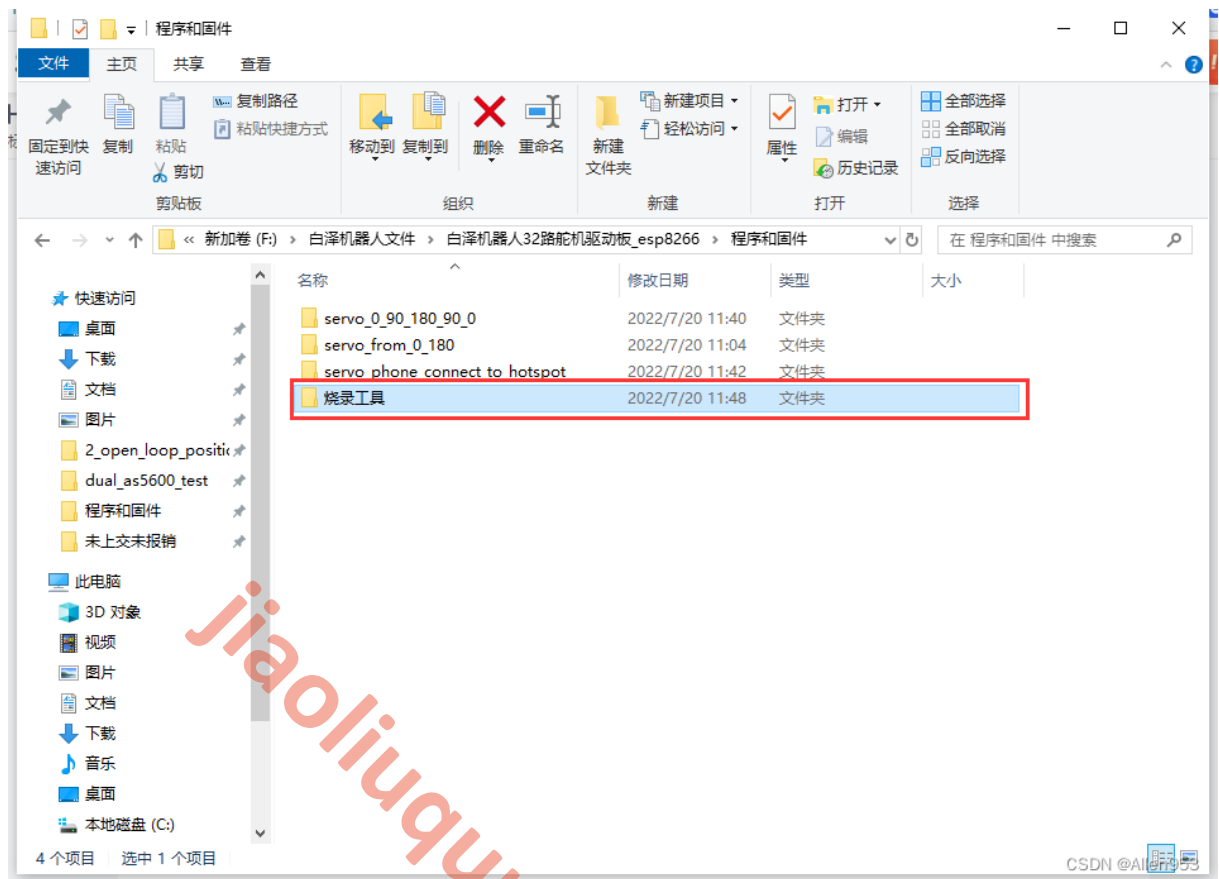
## 固件烧录

有些小伙伴没有安装好 arduino ide 或者没有安装好 esp8266 的硬件支持包，这个时候呢，就没法编译程序进行烧录了。

所以我在资料文件夹里面放上了编译好的固件，咱们就无需编译直接使用烧录工具就可以把固件给烧录到板子上了。

找到烧录工具并打开





编辑

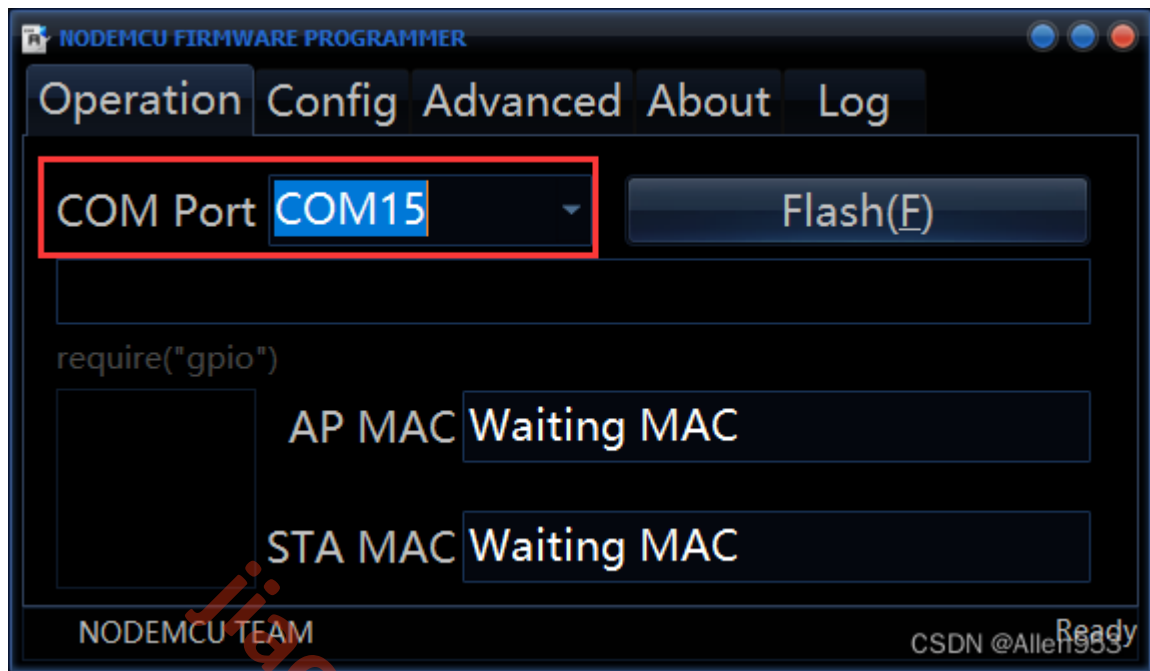
打开烧录程序

名称	修改日期	类型	大小
Config	2022/7/20 11:47	文件夹	
ESP8266Flasher.exe	2018/6/7 14:54	应用程序	6,934 KB

CSDN @Allen953

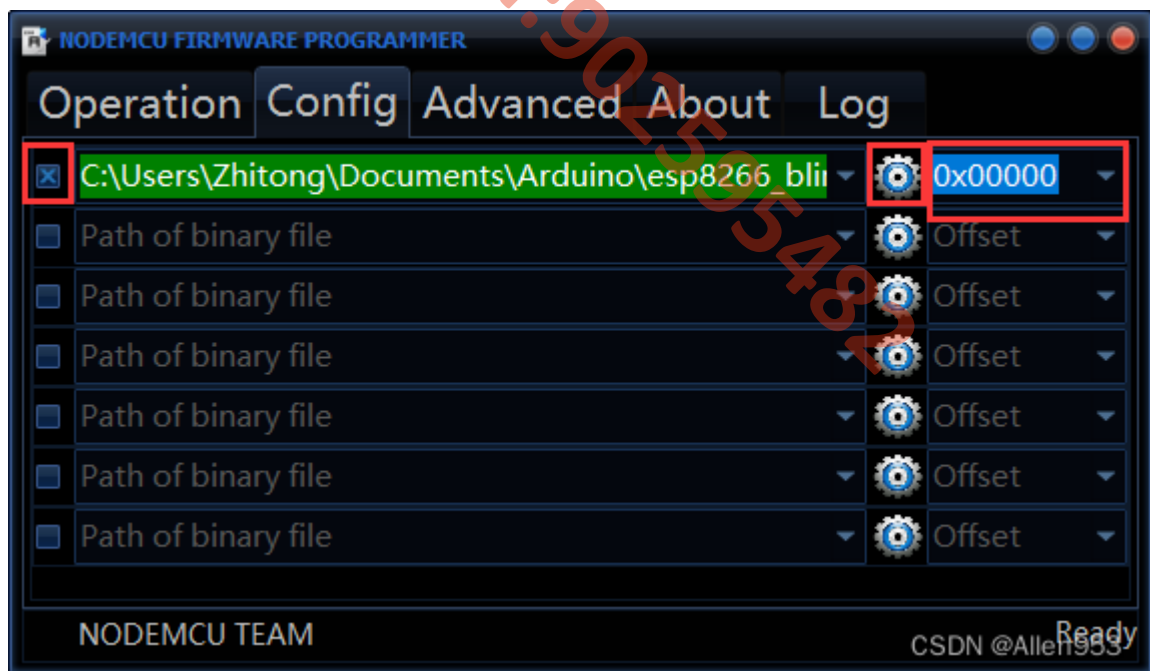
编辑

选中 com 口



编辑

点击齿轮选中要烧录的固件(.bin 格式)，填写烧录地址，并在要烧录的固件前面打勾。



编辑

选择上传的波特率，flash 大小，flash 速度等。

然后点击 flash 进行烧录，烧录完成后，进度条会跑满，左下角会出现绿色打勾符号。

按一下板子上的 rst 按键，重启板子，程序就开始执行了。

## 我遇到的问题

### 1.插电脑不出 com 口，怎么操作都没用

ch340 芯片焊错了，焊成了 ch340g 芯片，导致 usb-ttl 芯片无法正常工作。

### 2.插电脑不出 com 口，按住 rst 按键会出 com 口，松开又消失

原因是台式机电脑前面的 usb 口能供应的电流太小了，带不起来板子。

当按住 rst 的时候，esp8266 模组这个时候相当于是关机了，esp8266 模组不工作的时候，整个板子自然功耗下降下来，然后 ch340c 芯片工作正常，这样一来电脑就出 com 口了。

而一旦松开 rst，这个时候 esp8266 模组要开始工作，板子功耗又起来了，usb 口带不动过载了，所以 ch340c 芯片无法正常工作，com 口就消失了。

## 七 3D 打印开源六足机器人程序

### 3.编程例程

持续更新中!!!!!!

jiaoliuqun:902595482

## 附录

jiaoliuqun:902595482