Baize\_H1mini制作教程

作者：Allen

## 一.项目介绍

代码：<https://gitee.com/li9535/baize_hexapod-robot>

最新代码在这里：

Arduino版：[https://github.com/Allen953/Baize\_HexapodRobot\_Arduino](https://github.com/Allen953/Baize_HexapodRobot_Arduino" \o "https://github.com/Allen953/Baize_HexapodRobot_Arduino)

ROS版：[https://github.com/Allen953/Baize\_HexapodRobot\_ROS](https://github.com/Allen953/Baize_HexapodRobot_ROS" \o "https://github.com/Allen953/Baize_HexapodRobot_ROS)

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ros和arduino两个版本的，提供更多丰富的功能。

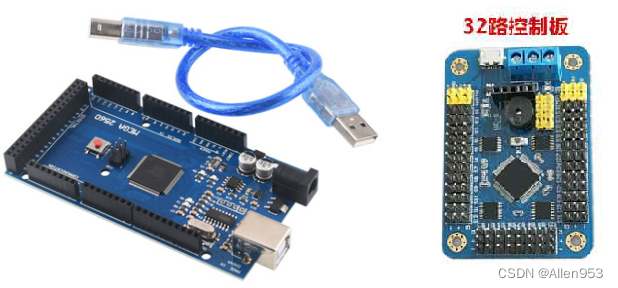
下面我们在这个项目后面加上“\_arduino”和"\_ros"的后缀来区分arduino和ros两个版本。

再在后面加上"\_v1"来表示版本号。

例如:“白泽六足机器人\_arduino\_v1”就是表示这个是arduino版本的第一版。

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。

最初这个机器人用的也是arduino mega2560当主控，外加一块舵机驱动板，如下图

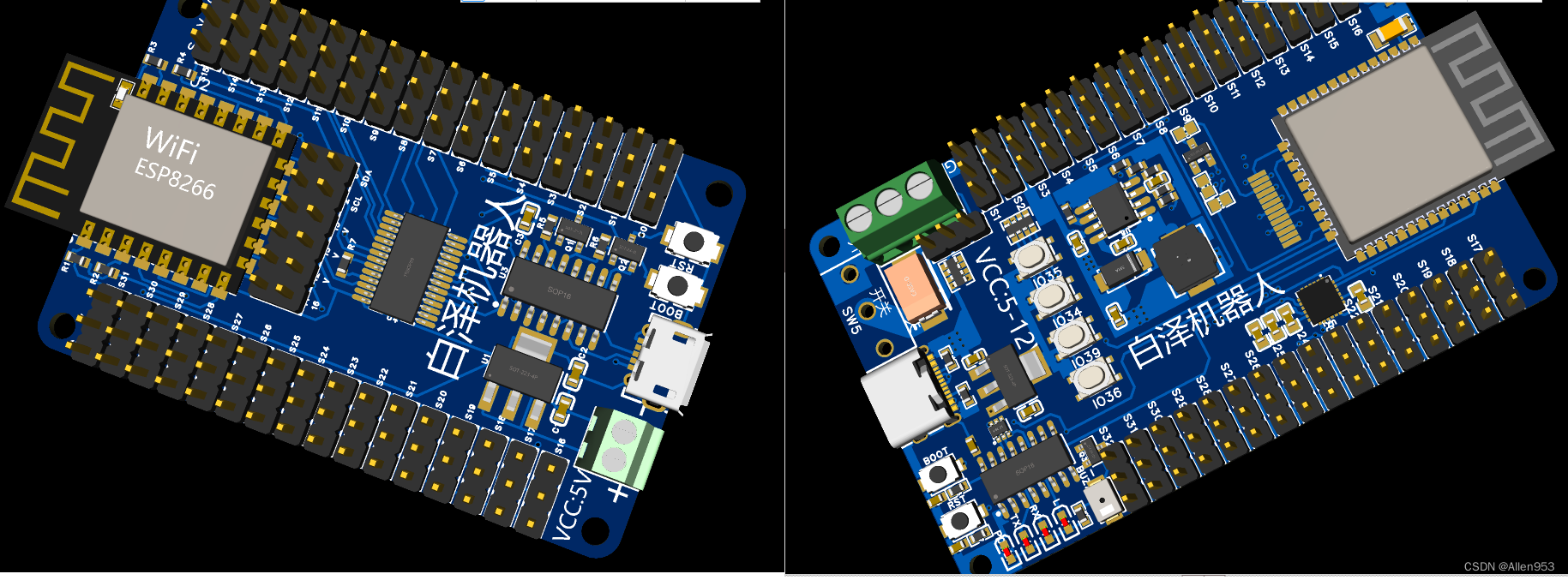
​编辑

在新版本中，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。

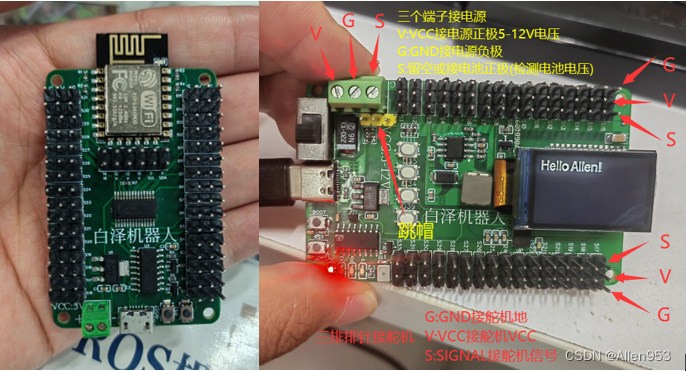
所以我们自己做了两版舵机驱动板，用esp8266或者esp32做主控，更低的成本，更强大的性能，更丰富的功能且支持WiFi无线遥控，实在是香！

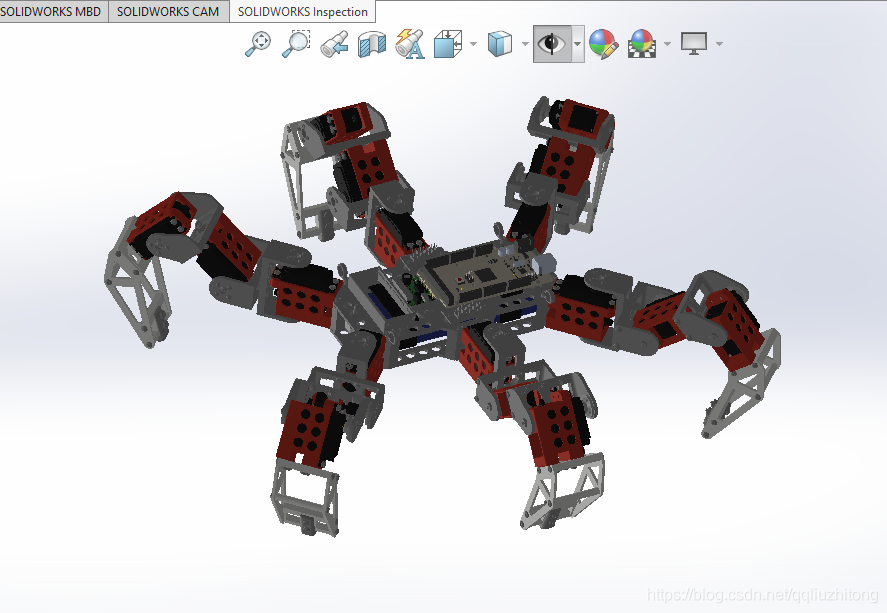
他们分别是采用esp8266做主控的[Baize\_ServoDriver\_esp8266](https://github.com/Allen953/Baize_ServoDriver_esp8266" \o "Baize_ServoDriver_esp8266)和采用esp32做主控的[Baize\_ServoDriver\_esp32](https://github.com/Allen953/Baize_ServoDriver_esp32" \o "Baize_ServoDriver_esp32)。

如下图，都是开源的，点击项目名称即可跳转到开源链接。

​编辑

实物图

​编辑

两个版本的机器人外观大致如下图所示：

## 二.Baize\_H1mini制作组装机器人机械结构

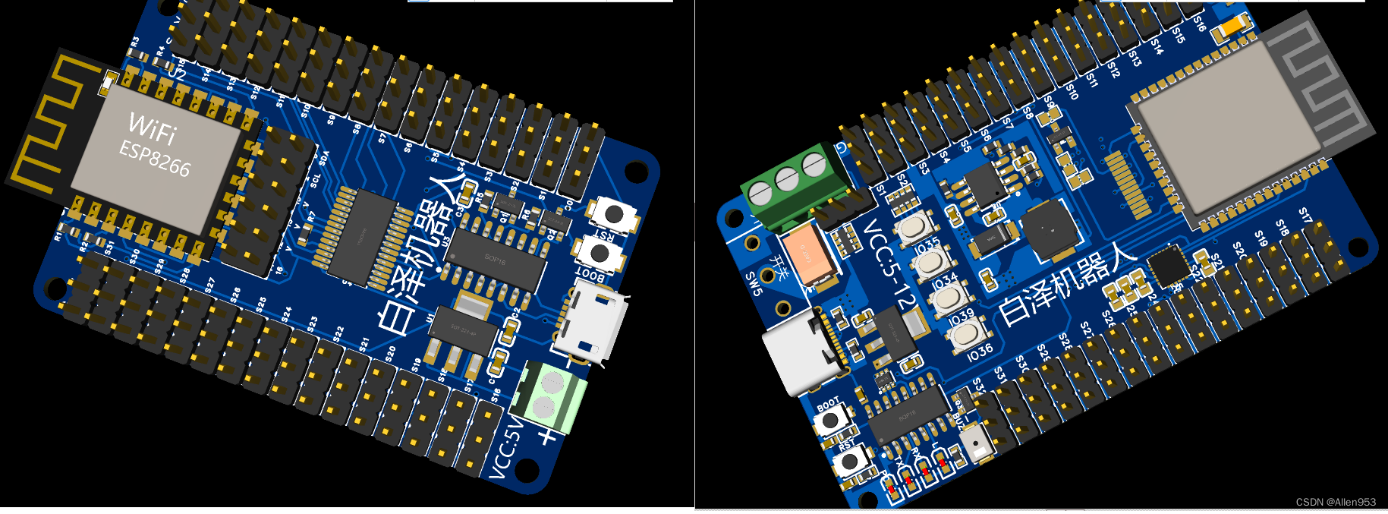
### 1.组装调试教程

**（1）**[**零件准备**](https://blog.csdn.net/qqliuzhitong/article/details/118224735)

1.主控板x1

[Baize\_ServoDriver\_esp8266](https://github.com/Allen953/Baize_ServoDriver_esp8266)或者[Baize\_ServoDriver\_esp32](https://github.com/Allen953/Baize_ServoDriver_esp32)

如下图左边为 Baize\_ServoDriver\_esp8266；右边为Baize\_ServoDriver\_esp32

​编辑

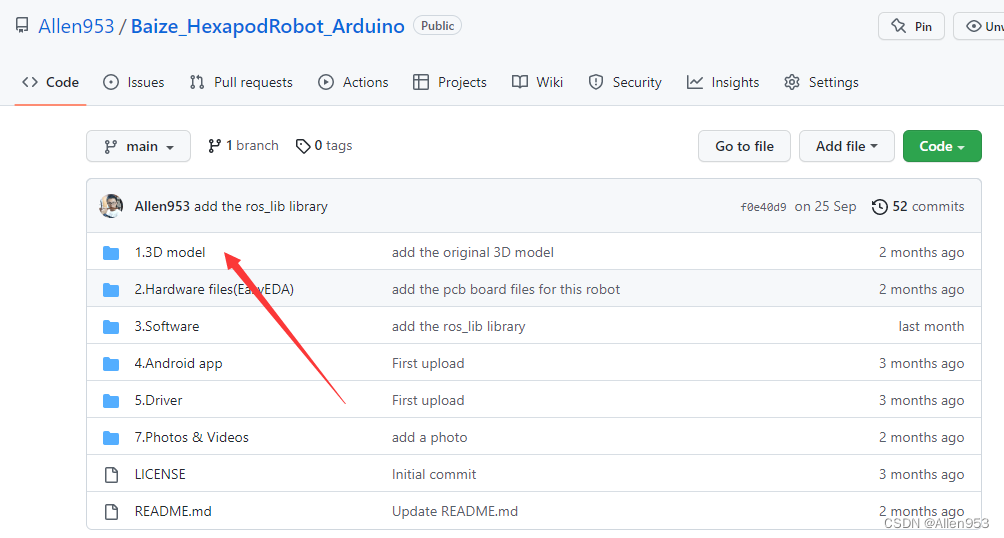
 建议arduino版六足用Baize\_ServoDriver\_esp8266而ROS版六足用Baize\_ServoDriver\_esp32。

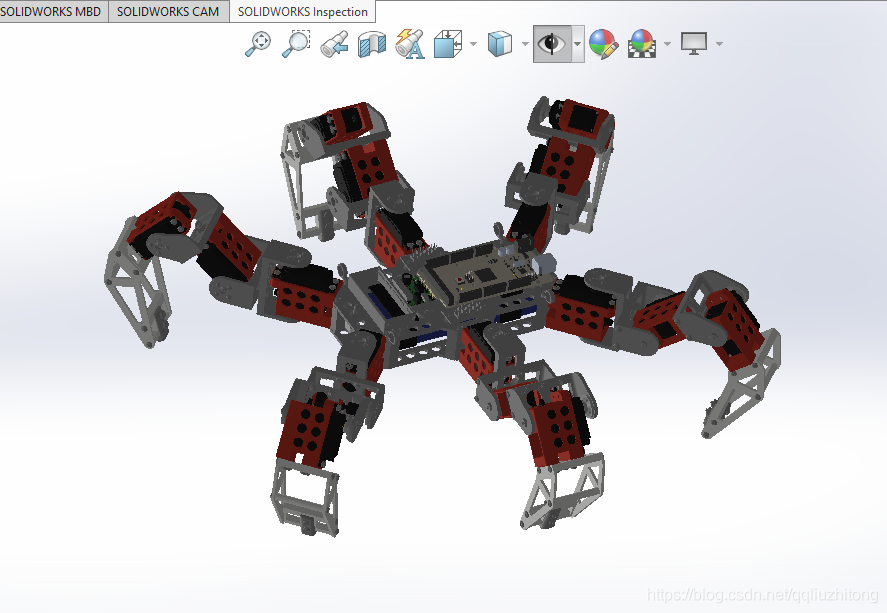
我们这里是做arudino版，所以就默认为Baize\_ServoDriver\_esp8266。

2.机械零件

 链接：[GitHub - Allen953/Baize\_HexapodRobot\_Arduino](https://github.com/Allen953/Baize_HexapodRobot_Arduino)

 在如下目录下即可下载到

​编辑

​编辑

4.舵机x18

​编辑

5.轴承

**624ZZ**

​编辑

6.电池

电池要买大功率电池，7.4V的大电流电池。

​编辑

7.螺丝

螺丝主要有两种，一种是M3\*10，一种是M3\*12。下面是螺丝型号及数量。

M3\*10——200个

M3\*12——200个

**（2）**[**白泽舵机驱动板**](https://blog.csdn.net/qqliuzhitong/article/details/124497060)

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.

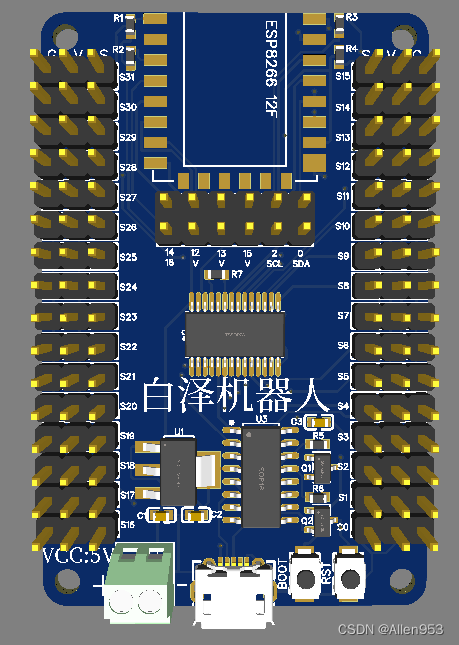
电源用的5V的电池，电池直接给舵机供电，经过压降之后产生3.3V电压给esp8266模组供电。

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。

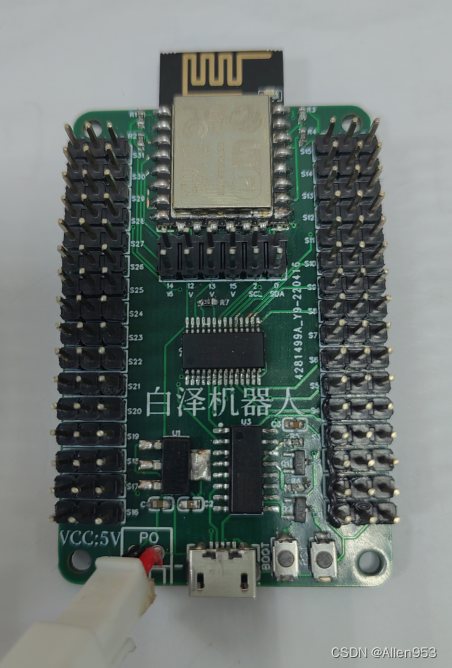
这样就可以控制高压的舵机，比如型号为mg996r或mg995的舵机。

3D图片如下图所示：



编辑

实物图如下：



编辑

#### 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 pwm1 = 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();

pwm1.begin();

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

pwm1.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

pwm1.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);

pwm1.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 pwm1 = 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();

pwm1.begin();

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

pwm1.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);

pwm1.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);

pwm1.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 pwm1 = 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；pwm1.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();

pwm1.begin();

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

pwm1.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);

pwm1.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);

pwm1.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);

pwm1.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);

pwm1.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%252220140713.130102334..%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%8B%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();//未联接,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

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 pwm1 = 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();

pwm1.begin();

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

pwm1.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

pwm1.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);

pwm1.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);

pwm1.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

pwm1.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 pwm1 = 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();

pwm1.begin();

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

pwm1.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);

}

pwm1.setPWM(0, 0, pwmsum);

pwm1.setPWM(1, 0, pwmsum);

delay(100);

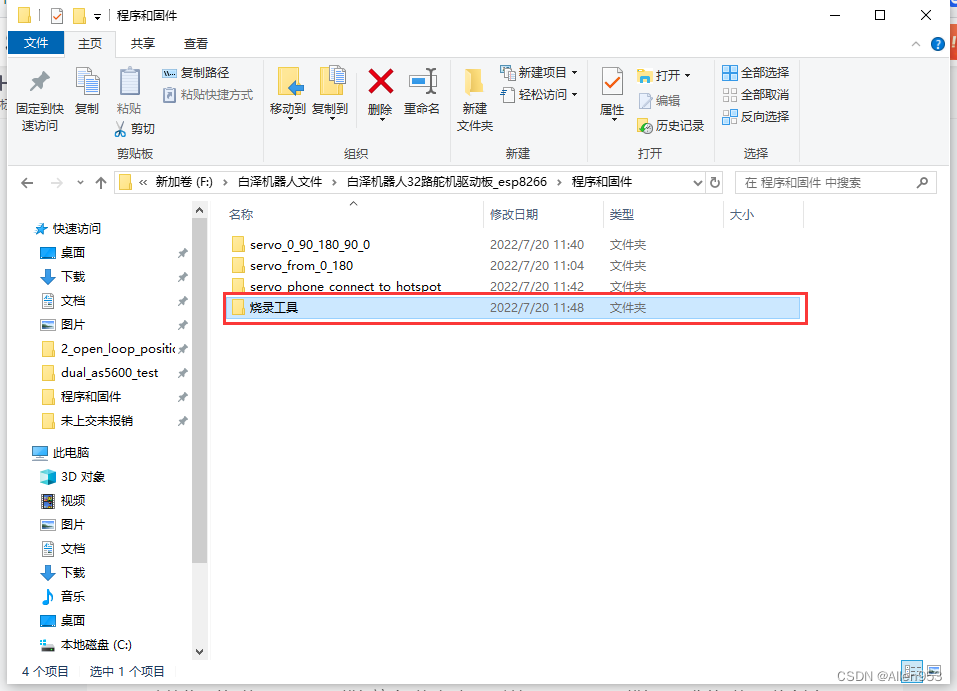
}

#### 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

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。

**（3）**[**3D打印开源六足机器人程序**](https://blog.csdn.net/qqliuzhitong/article/details/116492164)

### 2.编程例程

持续更新中！！！！！！！