



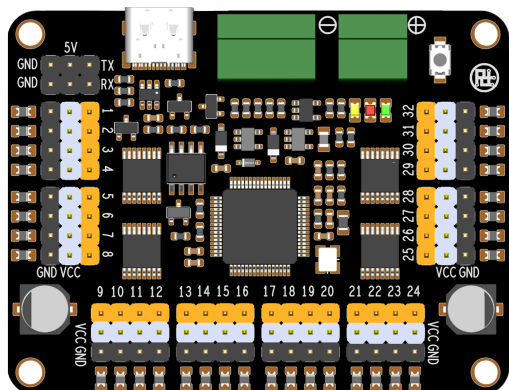
Async Servo Motor Controller
Instructions for use
舵机异步控制器使用说明

Ver 1.0

参数对比:

硬件	异步 32 通道	同步 32 通道
工作电压	5V	
舵机输入电压	根据舵机为准	
CPU	32 位	
波特率 (USB)	115200	
波特率 (蓝牙、UART)	4800、9600、19200、38400、57600、115200	
可同时操控舵机数	32 通道	
储存 Flash 容量	无	16M
最大动作组数	无	255 组
控制精度	1us	
频率	50Hz	
舵机信号隔离	有	有
限流保护	有	无
并联支持	支持 (UART/RS485)	不支持
通道独立速度	支持	不支持
3D 展示效果界面	全部	全部
指示灯	1.CPU 电源指示灯 (红) 2.舵机指示灯 (绿) 3.低压警报 (黄)	1.CPU 电源指示灯 (红) 2.舵机指示灯 (绿) 3.无线手柄模式 (黄)
板子尺寸	66mm X 50.5mm	64mm X 45mm
通讯协议	RTrobot、Modbus RTU(UART/RS485)	RTrobot
硬件接口	UART、USB、RS485	UART、USB
支持系统	Windows 10 以上, Mac OS 10.8 以上, Linux (glibc-2.35 以上)	
低压警报	支持	
舵机初始值	默认 1500	
支持舵机类型	9G~55G	
脱机独立运行	不支持	支持
联机操作支持	C51、Arduino、ARM、MSP、DSP、蓝牙、Wi-Fi、电脑	
外设支持	无	PS2 手柄、电位器

32 通道:



指令:

通讯协议:

串口通讯	波特率	校验位	数据位	停止位
TTL	115200 (默认)	无	8	1

指令格式:

名称	指令	说明
控制单个舵机	#1P1500T1000\r\n	数据 1 为舵机编号 数据 1500 为舵机角度值, 范围为 500-2500 数据 1000 为舵机执行速度, 范围是 1-9999
控制多个舵机	#1P1500T789#2P1234T1000\r\n	数据 1、2 为舵机编号 数据 1500 为舵机 1 角度值, 范围为 500-2500 数据 789 为舵机 1 执行速度, 范围是 1-9999 数据 1234 为舵机 2 角度值, 范围为 500-2500 数据 1000 为舵机 2 执行速度, 范围是 1-9999

注意: “\r\n”转化为 16 进制为“0X0D 0X0A”; 所有指令都为 ASCII 码。

“0x0D” == “\r” == “CR”

“0x0A” == “\n” == “LF”

提示: 如果编程中使用的函数或者软件有“\r\n”功能则末尾无需再加。当指令执行完成后控制器不会反馈。

系统指令：

指令	说明
~RESTART	重新启动控制器
~STOP	停止通道，（注意：停止后不可恢复，只能发送新的指令重新执行） 参数： =ALL（停止所有通道） =15（停止 15 号通道） 例子： ~STOP=15\r\n
~GET+POS	获取通道当前值 参数： =15（获取 15 号通道当前值） 例子： ~GET+POS=15\r\n 反馈： ~POS15:1500\r\n

Modbus RTU 协议:

寄存器表:

寄存器地址	通道号	功能
0x00	Null	停止通道运行 (写入对应序号则停止对应通道, 写入 0x00FF 停止所有通道)
0x01	1	舵机目标位置值(uint16)
0x02	1	舵机运行速度(uint16)
0x03	2	舵机目标位置值(uint16)
0x04	2	舵机运行速度(uint16)
0x05	3	舵机目标位置值(uint16)
0x06	3	舵机运行速度(uint16)
0x07	4	舵机目标位置值(uint16)
0x08	4	舵机运行速度(uint16)
0x09	5	舵机目标位置值(uint16)
0x0A	5	舵机运行速度(uint16)
0x0B	6	舵机目标位置值(uint16)
0x0C	6	舵机运行速度(uint16)
0x0D	7	舵机目标位置值(uint16)
0x0E	7	舵机运行速度(uint16)
0x0F	8	舵机目标位置值(uint16)
0x10	8	舵机运行速度(uint16)
0x11	9	舵机目标位置值(uint16)
0x12	9	舵机运行速度(uint16)
0x13	10	舵机目标位置值(uint16)
0x14	10	舵机运行速度(uint16)
0x15	11	舵机目标位置值(uint16)
0x16	11	舵机运行速度(uint16)
0x17	12	舵机目标位置值(uint16)
0x18	12	舵机运行速度(uint16)
0x19	13	舵机目标位置值(uint16)
0x1A	13	舵机运行速度(uint16)
0x1B	14	舵机目标位置值(uint16)
0x1C	14	舵机运行速度(uint16)
0x1D	15	舵机目标位置值(uint16)
0x1E	15	舵机运行速度(uint16)

0x1F	16	舵机目标位置值(uint16)
0x20	16	舵机运行速度(uint16)
0x21	17	舵机目标位置值(uint16)
0x22	17	舵机运行速度(uint16)
0x23	18	舵机目标位置值(uint16)
0x24	18	舵机运行速度(uint16)
0x25	19	舵机目标位置值(uint16)
0x26	19	舵机运行速度(uint16)
0x27	20	舵机目标位置值(uint16)
0x28	20	舵机运行速度(uint16)
0x29	21	舵机目标位置值(uint16)
0x2A	21	舵机运行速度(uint16)
0x2B	22	舵机目标位置值(uint16)
0x2C	22	舵机运行速度(uint16)
0x2D	23	舵机目标位置值(uint16)
0x2E	23	舵机运行速度(uint16)
0x2F	24	舵机目标位置值(uint16)
0x30	24	舵机运行速度(uint16)
0x31	25	舵机目标位置值(uint16)
0x32	25	舵机运行速度(uint16)
0x33	26	舵机目标位置值(uint16)
0x34	26	舵机运行速度(uint16)
0x35	27	舵机目标位置值(uint16)
0x36	27	舵机运行速度(uint16)
0x37	28	舵机目标位置值(uint16)
0x38	28	舵机运行速度(uint16)
0x39	29	舵机目标位置值(uint16)
0x3A	29	舵机运行速度(uint16)
0x3B	30	舵机目标位置值(uint16)
0x3C	30	舵机运行速度(uint16)
0x3D	31	舵机目标位置值(uint16)
0x3E	31	舵机运行速度(uint16)
0x3F	32	舵机目标位置值(uint16)
0x40	32	舵机运行速度(uint16)

停止示例：

设备号	功能位	起始位高	起始位低	寄存器数量高	寄存器数量低	数据长度	序号高	序号低	CRC 高	CRC 低
-----	-----	------	------	--------	--------	------	-----	-----	-------	-------

停止所有通道运行：

0x01,0x10,0x00,0x00,0x00,0x01,0x02,0x00,0xFF,0xE6,0x10

停止通道 7 运行：

0x01,0x10,0x00,0x00,0x00,0x01,0x02,0x00,0x07,0xE7,0x92

停止通道 16 运行：

0x01,0x10,0x00,0x00,0x00,0x01,0x02,0x00,0x10,0xA7,0x9C

读取通道值示例：

设备号	功能位	起始位高	起始位低	寄存器数量高	寄存器数量低	CRC 高	CRC 低
-----	-----	------	------	--------	--------	-------	-------

读取通道 5 的当前位置值(寄存器地址:0x09):

0x01,0x04,0x00,0x09,0x00,0x01,0xE1,0xC8

读取通道 32 的当前位置值(寄存器地址:0x3F):

0x01,0x04,0x00,0x3F,0x00,0x01,0x01,0xC6

写入通道值示例：

设备号	功能位	起始位高	起始位低	寄存器数量高	寄存器数量低	数据长度	通道位置高	通道位置低	速度高	速度低	CRC 高	CRC 低
-----	-----	------	------	--------	--------	------	-------	-------	-----	-----	-------	-------

写入通道 1 的目标值为 1734(0x06C6)与速度值 5478(0x1566) (寄存器地址 0x01, 0x02):

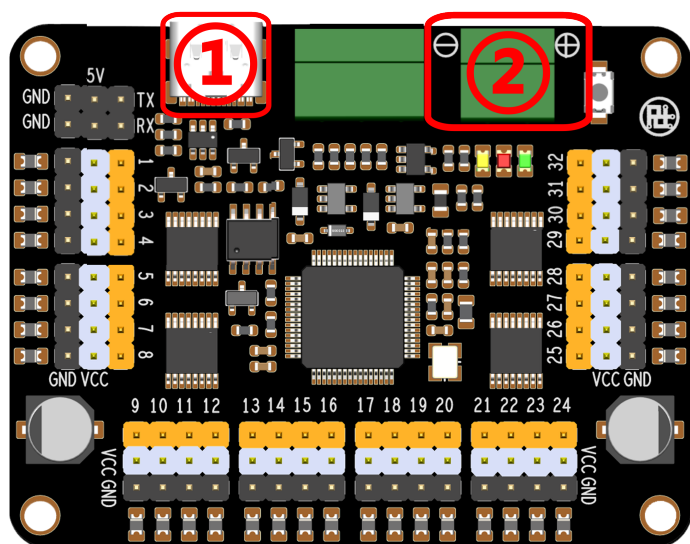
0x01,0x10,0x00,0x01,0x00,0x02,0x04,0x06,0xC6,0x15,0x66,0x5C,0x6C

写入通道 31 的目标值为 534(0x0216)与速度值 478(0x01DE):

0x01,0x10,0x00,0x3D,0x00,0x02,0x04,0x02,0x16,0x01,0xDE,0x50,0x96

接线方法:

一、电源接入方法，如图一位置：



图一

VCC：为舵机电源 VCC 输入口，可接入 4.2V~12V 电源，请接入电源正极。

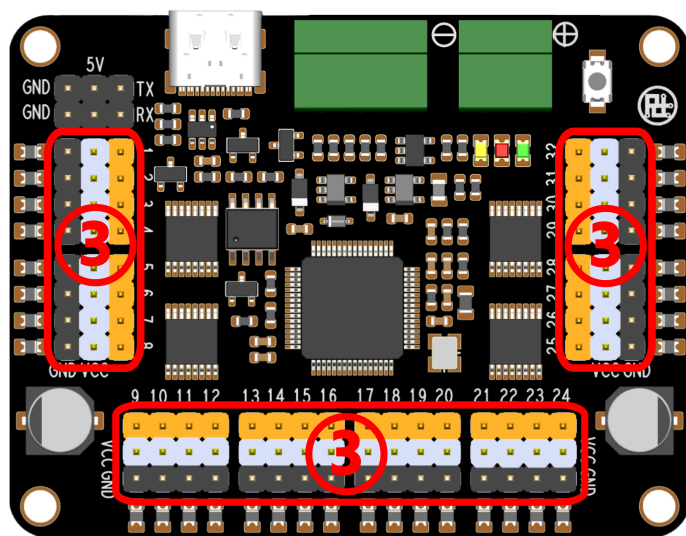
注意：控制器的 VCC 接口是舵机的电源输入口。VCC 接口应该根据舵机的要求选择，例如 1 个舵机需要 6V 电压峰值 2A 电流，10 个舵机电源则应该选择为 6V 电压，10A 电流的电源。

GND：为舵机控制器的整体 GND，可接入舵机电源 GND 或者是 CPU 电源 GND，请接入电源负极。

USB (1)：同为舵机控制器的 CPU 电源输入口和数据通信口。

USB 接口只支持 RTrobot 协议，不支持 Modbus 协议。

二、舵机接线方法，如图二位置：



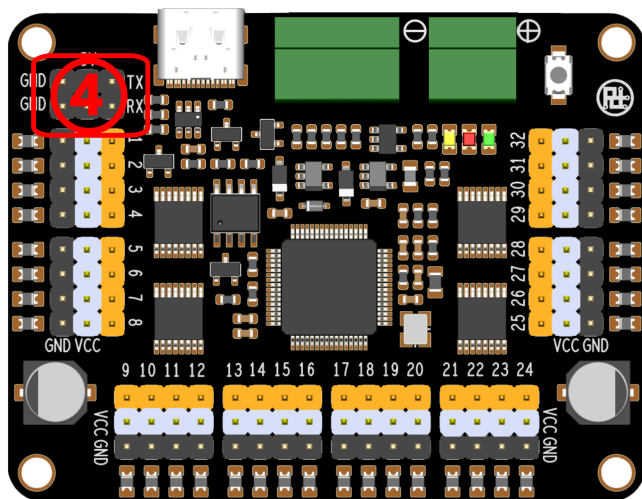
图二

黄色排针：为舵机 I/O 接入口，一般是舵机的黄色或者是土黄色。

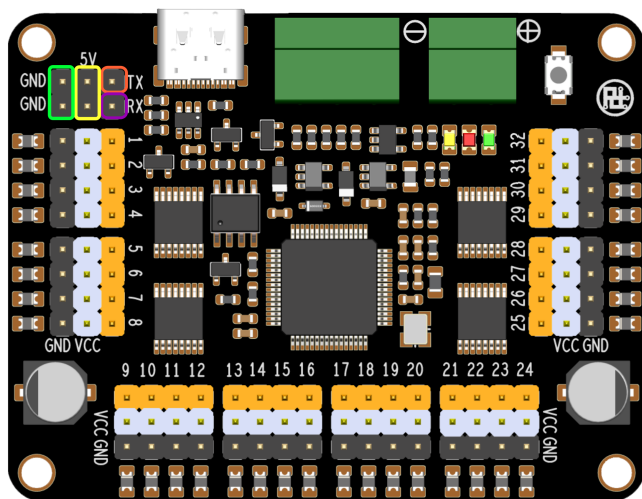
白色排针：为舵机 VCC 接入口，一般是舵机的白色、红色或者是暗红色。

黑色排针：为舵机 GND 接入口，一般是舵机的褐色或者是黑色。

三、UART 接线方法，如图三位置，配合图四阅读：



图三



图四

绿色圈位置：为舵机控制器 CPU 电源入口 GND。

黄色圈位置：为舵机控制器 CPU 电源入口 VCC，仅可接 5V。

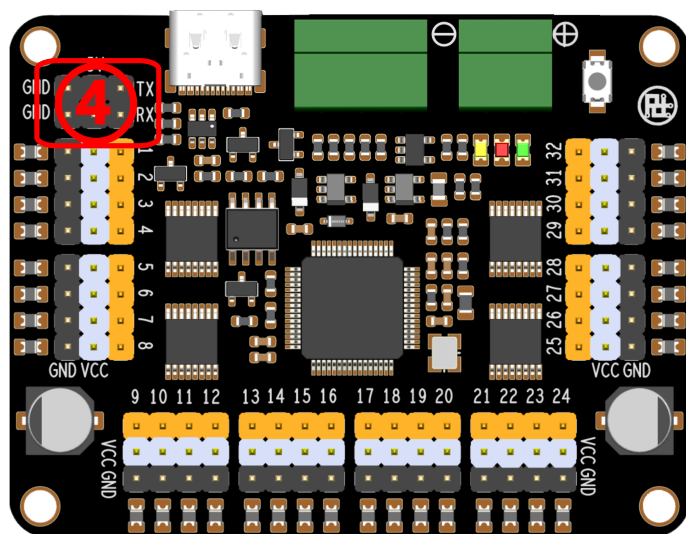
紫色圈位置：为舵机控制器 UART 的 RX 口，一般接入其他 UART 设备的 TX 口。

橙色圈位置：为舵机控制器 UART 的 TX 口，一般接入其他 UART 设备的 RX 口。

UART 可以支持两种协议，通过上位机可修改为 Modbus 协议。

UART 接口不可以并联。

四、蓝牙、Wi-Fi 传感器接线方法，如**图五**位置：



图五

图五位置使用 4 根杜邦线与蓝牙、Wi-Fi 模块链接，5V-VCC，GND-GND，RX-TX，TX-RX。

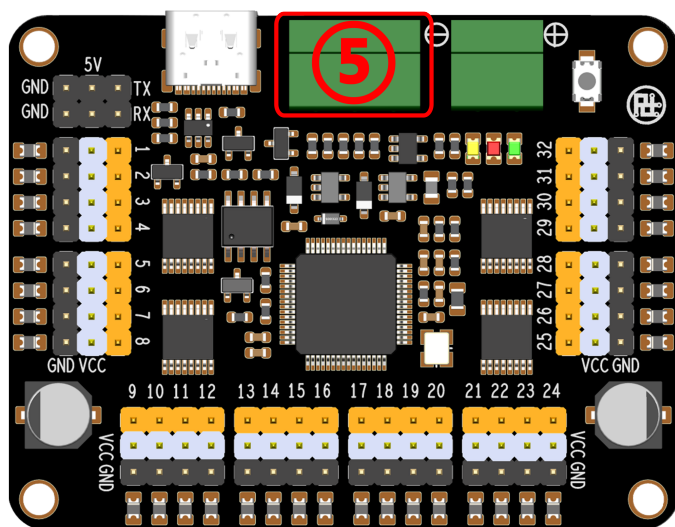
使用手机与蓝牙模块配对，安装打开手机控制软机即可进行控制。

安装打开手机控制软机，填写 Wi-Fi 模块设置的 TCP 地址即可进行控制。

温馨提示：在实际使用手机遥控前，应把蓝牙、Wi-Fi 模块接入电脑，使用串口调试软件查看是否能接收到相应指令。

首次使用手机软件的时候需要输入验证码，验证码为：RTrobot（注意大小写区分）

五、RS485 接口如图六:

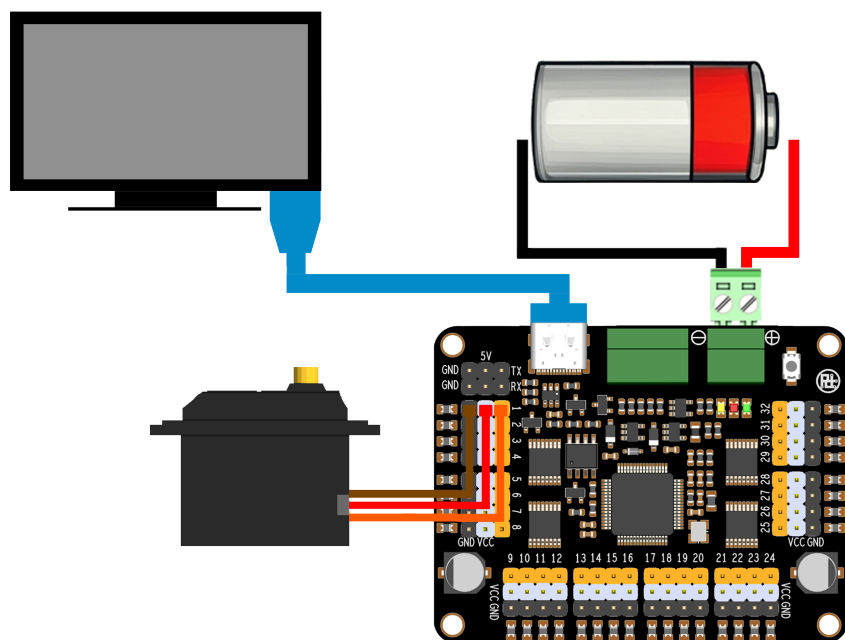


图六

图六位置为 RS485 2-Wire 接口，可以支持两种协议，通过上位机可修改为 Modbus 协议。RS485 接口可以并联多个控制器，但需要先修改为 Modbus 协议，如果设置为 RTrobot 协议无法并联。

整体接线例示:

一、使用电脑控制:



图七

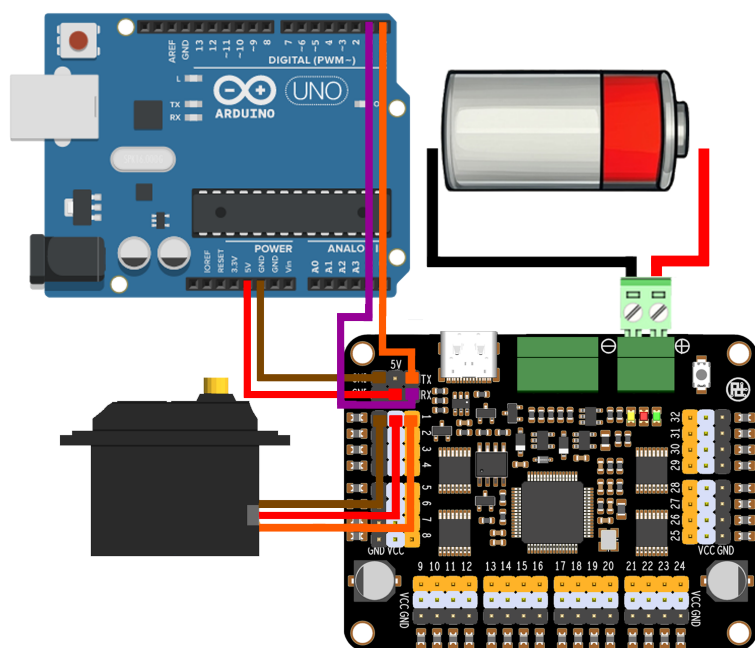
先把舵机与舵机的电源接入，然后使用 USB 线把电脑与舵机控制器链接在一起。

舵机的电源接入请参考**接线方法一**

温馨提示：舵机电源应根据舵机要求选择。

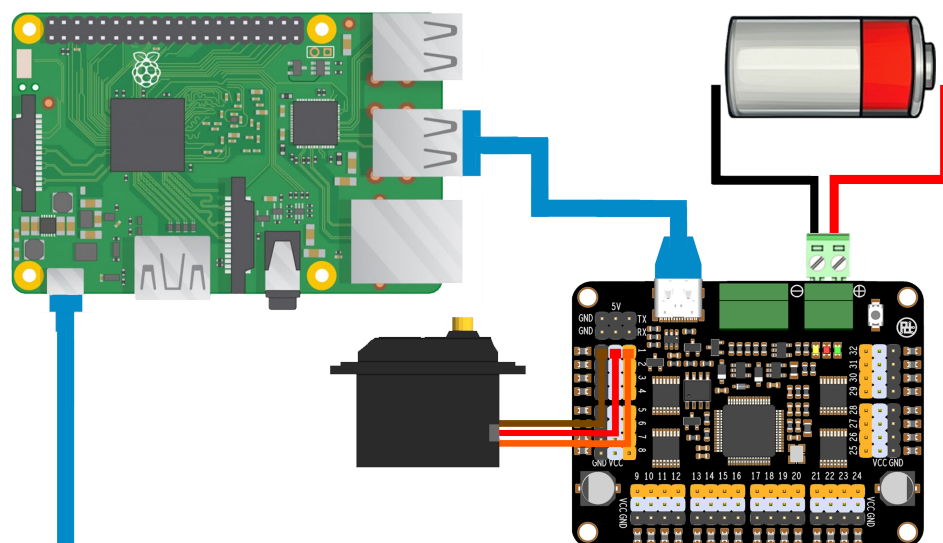
二、使用 MCU 控制：

这里以 Arduino uno 为例子，其他 MCU 参考此处。控制器 5V 接 Arduino uno 的 5V，控制器 GND 接 Arduino uno 的 GND，控制器 TX 接 Arduino uno 的 RX，控制器 RX 接 Arduino uno 的 TX。如图八。



图八

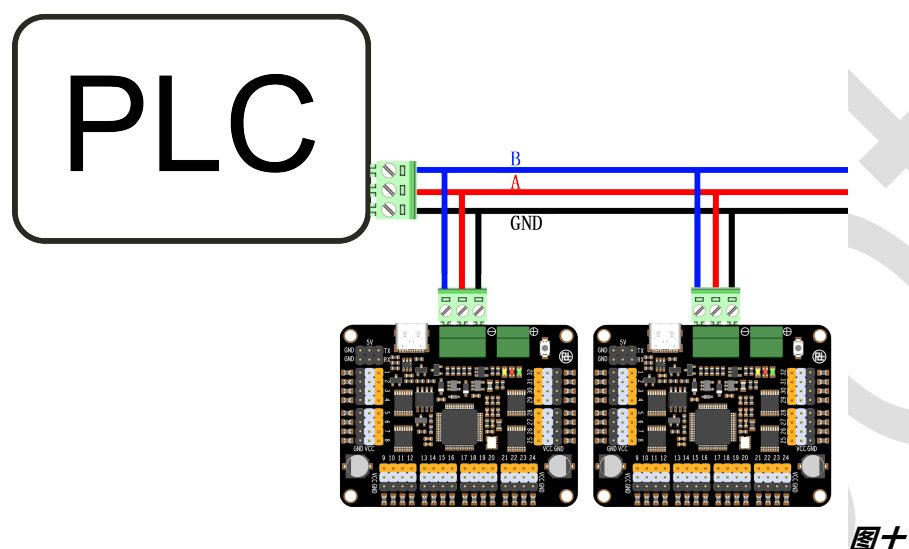
Raspberry Pi 用户可以使用 USB 线将控制器连接到 Raspberry Pi 上，把它当作一台电脑。



图九

三、使用 RS485 控制：

RS485 2-Wire 接口可以设置为 Modbus 协议，使用 Modbus 协议可以并联多个控制器以达到更多舵机的控制。在使用前，需要在上位机中设置 Modbus 的从地址。



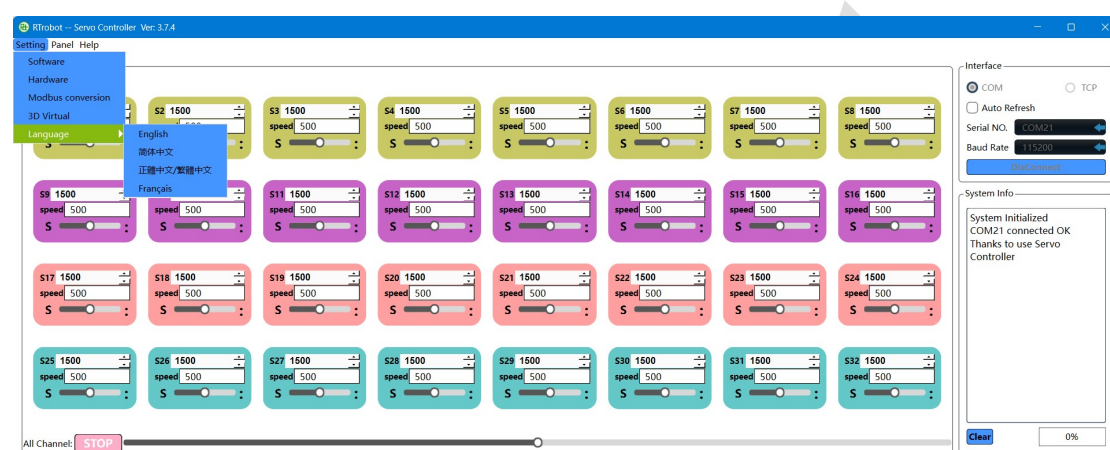
图十

软件操作方法:

注意：需打开串口才能使用软件的全部功能

一、语言设置:

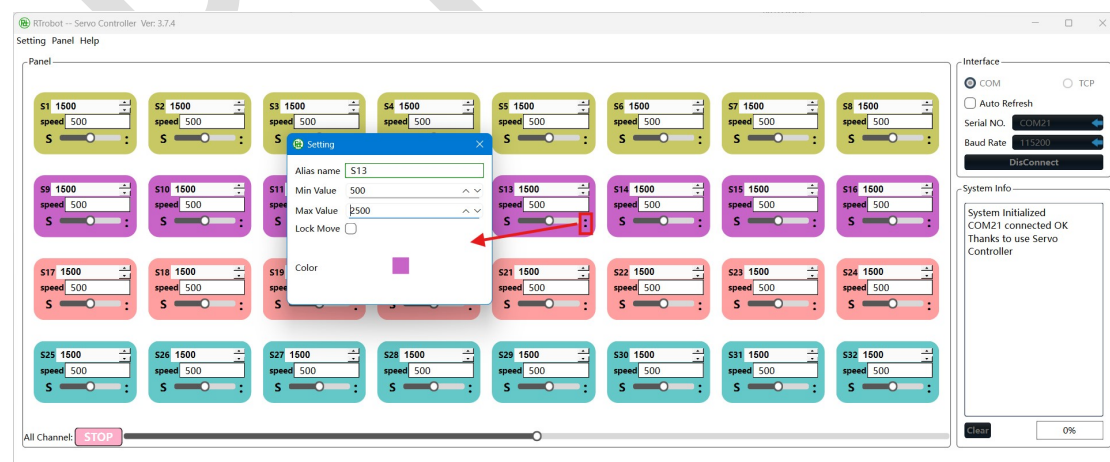
点击“Setting”->“Language”选择语言（简体/繁体/English）。



图十一

二、软件设置:

点击舵机窗口的“⚙️”，可以单独对每个舵机命名与设置最大最小值、颜色和锁定位置。



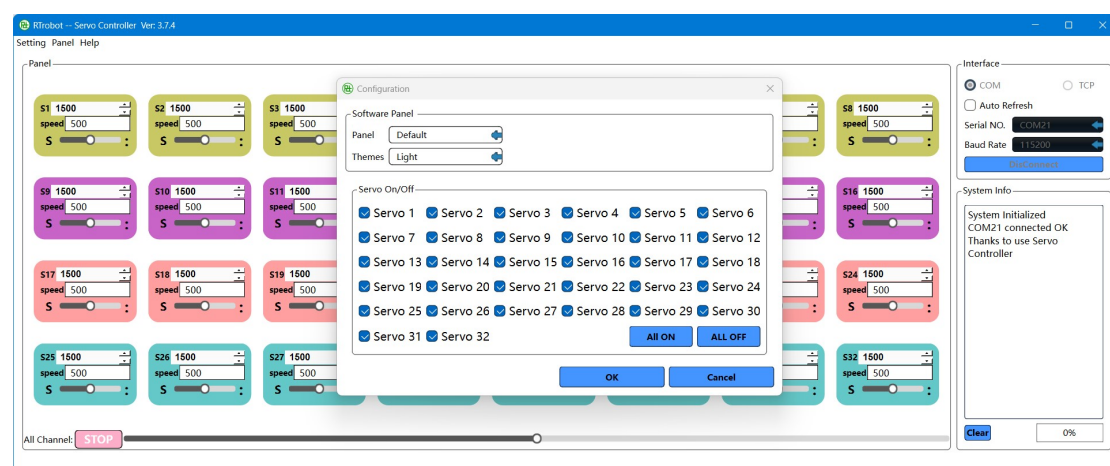
图十二

点击“Setting”->“Software”可对软件进行设置，如图十三。

Software Panel：设置软件控制面板。

Servo On/Off：显示需要控制的舵机序号。

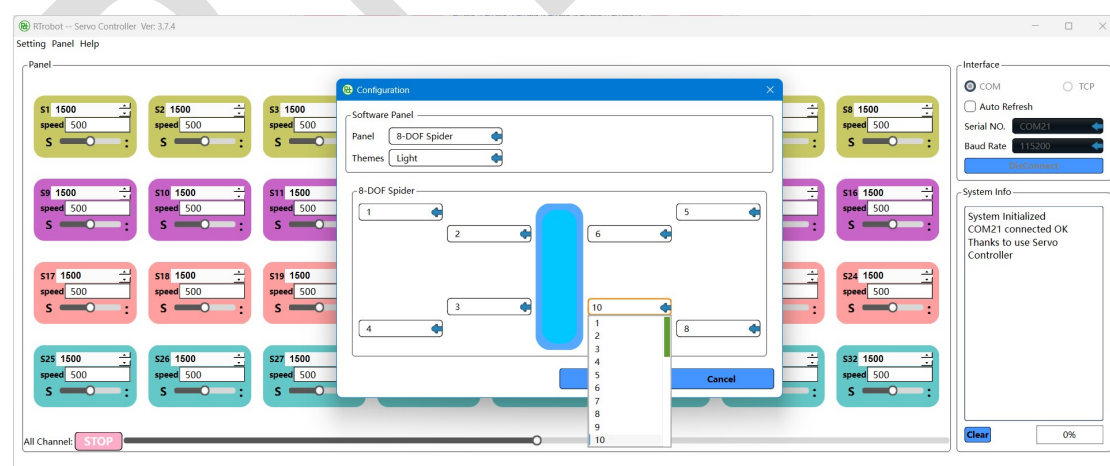
软件设置完成后点击“OK”会自动重新启动软件。



图十三

Software Panel 选择界面后可指定每个位置的舵机序号，如图十四。

注意：如果存在重复序号的舵机，则无法保存。



图十四

三、控制器设置：

点击“Setting”->“Hardware”可对控制器进行设置，如图十五。

Servo initial value：设置每个舵机上电启动的初始值。

Servo Deviation Value：设置每个舵机的偏差值（有效值：**-99~99**），如图十六。

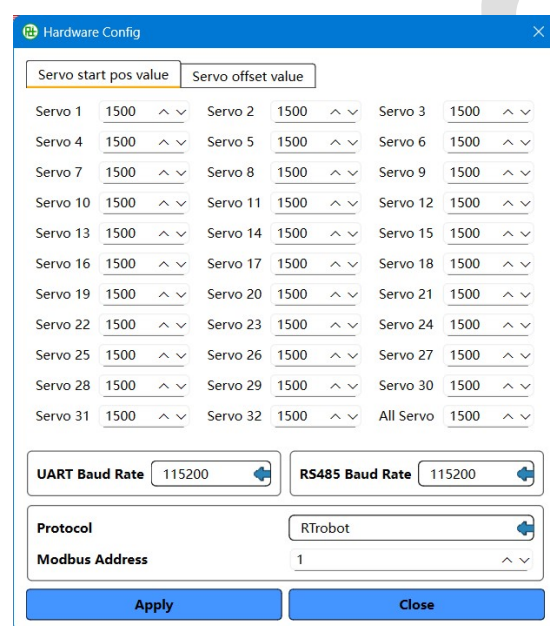
UART Baud Rate：设置图五中④位置的 UART 波特率。

RS485 Baud Rate：设置图六中⑤位置的 RS485 波特率。

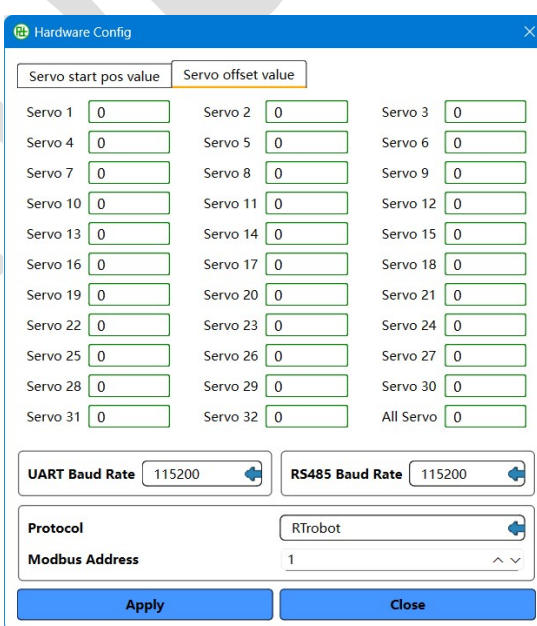
Protocol：设置 UART/RS485 的协议。

Modbus：Modbus 的从地址。

注意：设置完毕后记住点击 Apply，等待设置完毕。每次设置后控制器需要重新启动才会生效。



图十五

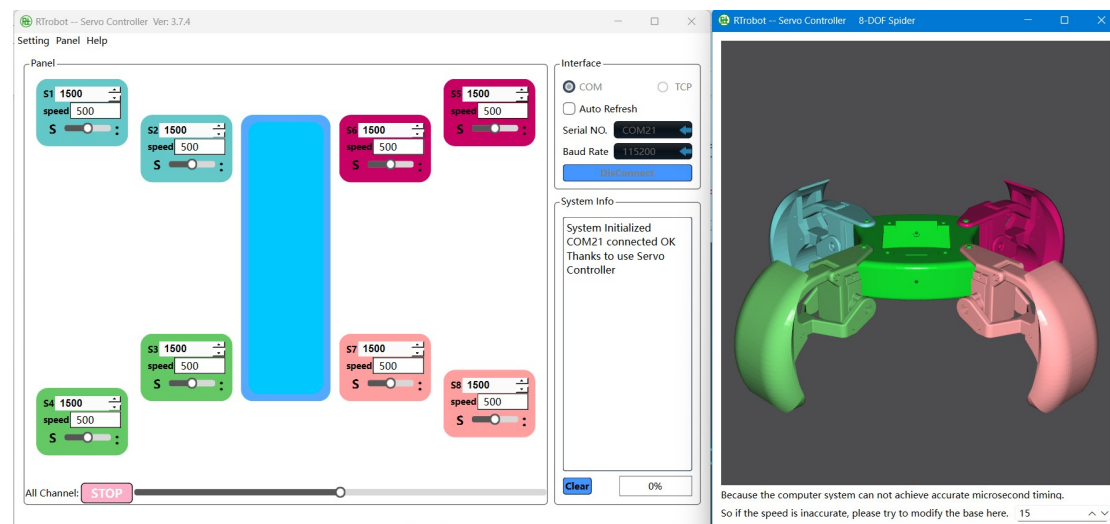


图十六

四、3D 展示效果：

点击“Setting”->“3D Virtual”可展示 3D 效果界面

注意：需要先在“Software”设置页选择好控制界面后打开串口才能展示 3D 效果界面



图十七

五、软件控制：

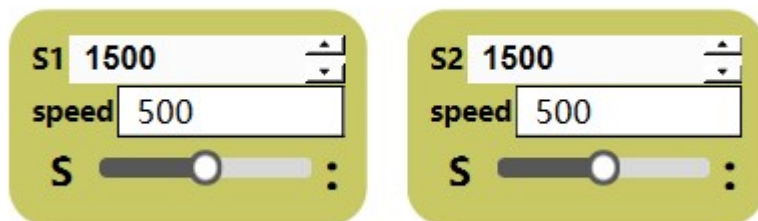
1. 选择一种合适自己的接线方式，并使用 USB 线与电脑连接好。
2. 安装控制器的驱动（Servos Controller Drive.exe）（Windows10 以上无需安装驱动）。

注意：如提示无数字签名无法安装，则把系统禁用驱动数字签名方式启动电脑系统再次运行安装。
3. 打开软件“ServoController.exe”。
4. 选择好串口，并且打开串口。如果使用 Wi-Fi 模式，选择 TCP，填写好 Wi-Fi 模块 TCP 服务器设定的 TCP 地址与端口。

注意：只有使用 USB 线连接电脑时才能使用全部功能。

单个舵机操作:

如图十八所示，拖动或者是填写都可以改变舵机角度值，每个通道都可以设置不同的速度，范围是：1~9999。"S"为停止按钮，可以停止当前通道。



图十八

多个舵机操作:



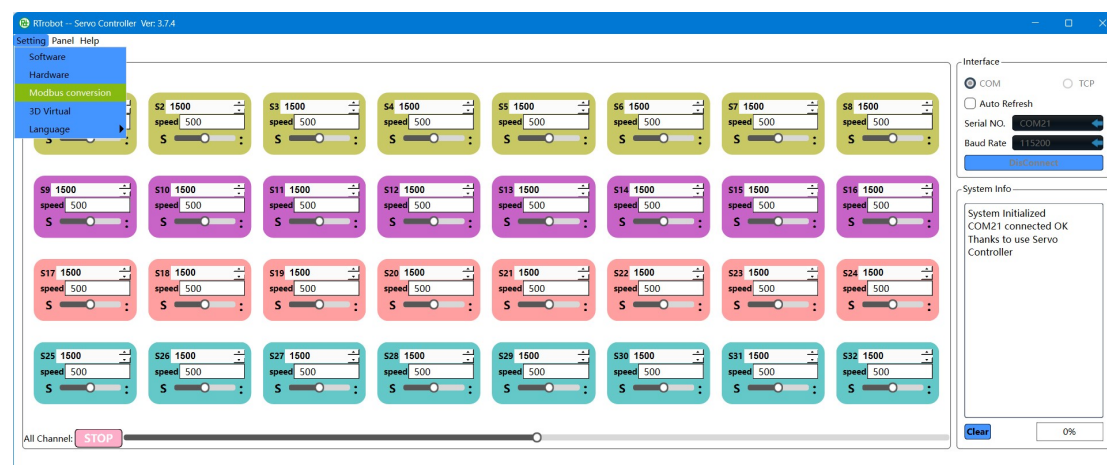
图十九

"STOP"为全部通道的停止按键，对应指令为"**~STOP=ALL\r\n**"。拖动条可以全通道控制，速度为每个通道独立的"speed"值。

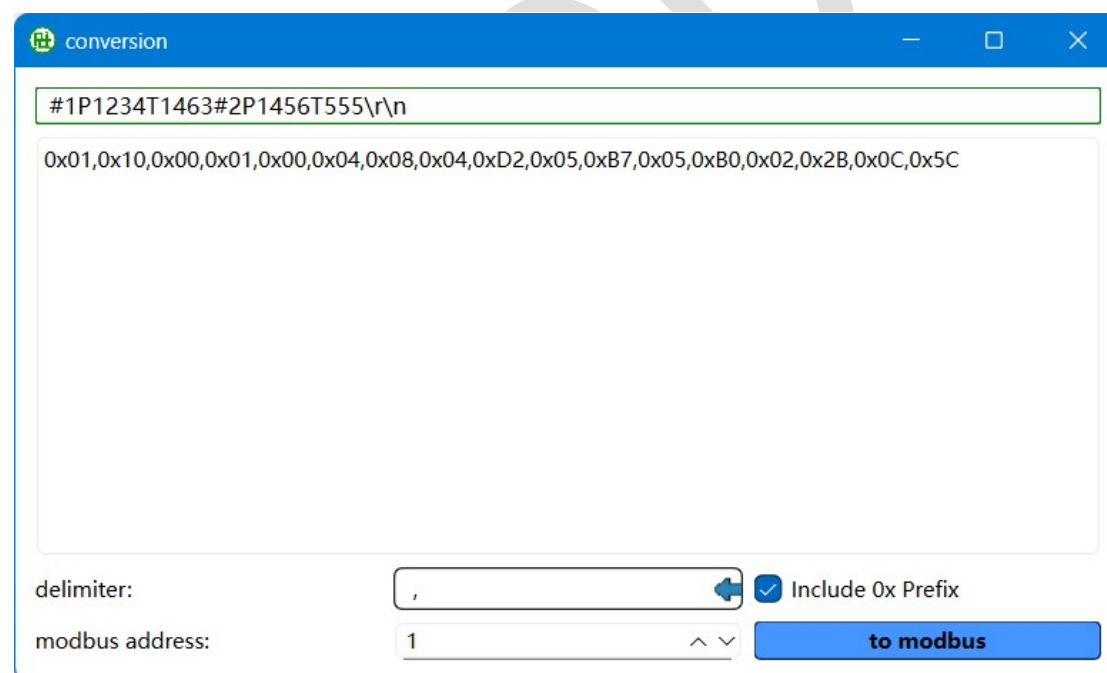
RTrobot 转 Modbus 协议:

为了方便使用 Modbus 协议，免于查表快速输入 Modbus 指令，可以使用软件自带的功能

"Modbus conversion"进行快速转换。



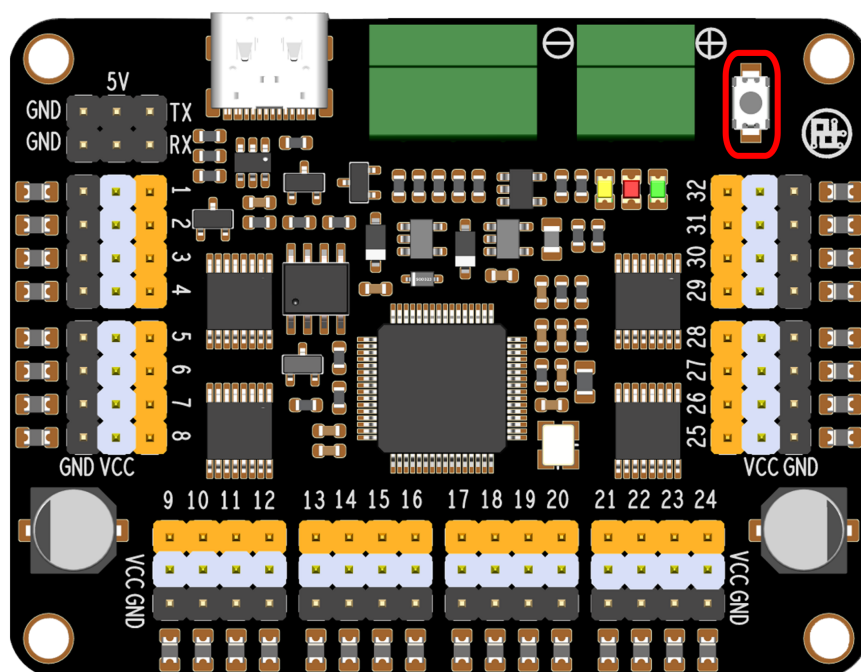
图二十



图二十一

六、升级固件方法

1. 在 “<http://www.rtrobot.org/software/>” 网站下载最新的上位机软件
2. 打开最新上位机软件
3. 按住舵机控制器上的按键不放的状态下链接 USB 数据线后松开按键

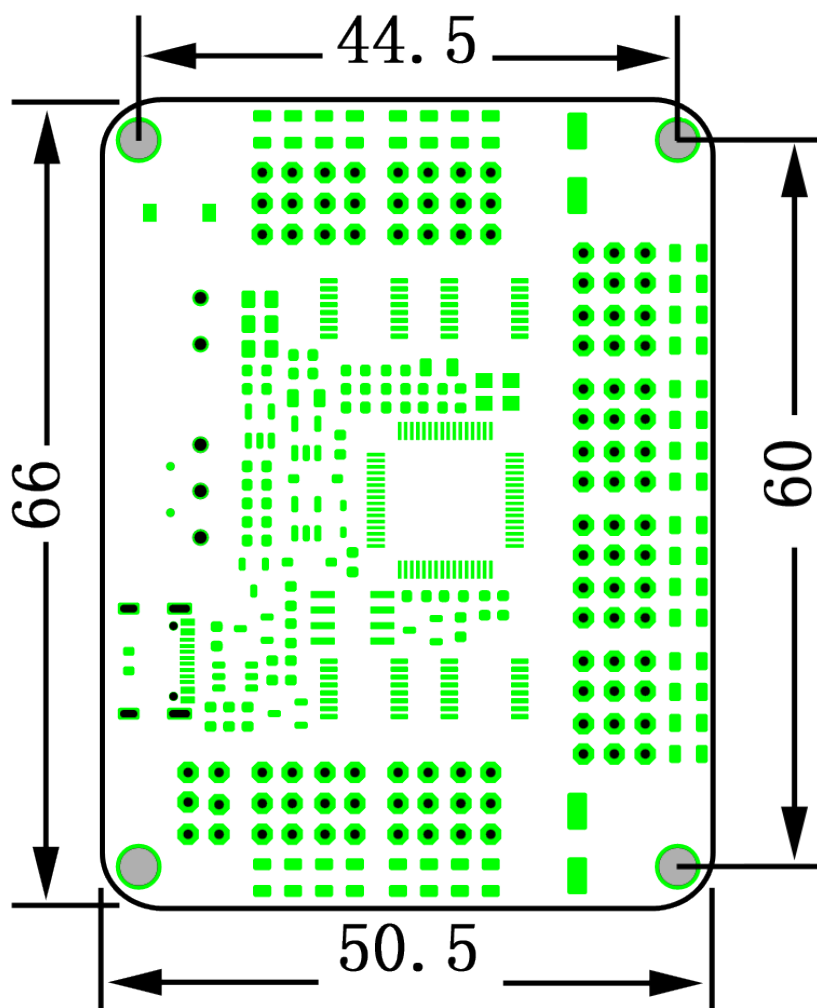


图二十二

4. 打开最新上位机软件的串口，这时固件开始升级，升级完后会提示 “Update Success, Restart the controller, Please. ”
5. 重启舵机控制器。
6. 如已是最新固件的话会提示 “ERROR Don't need Update!”

尺寸图:

32:



十、关于:

感谢使用 RTrobot 出品的舵机控制器，在使用中对控制器有任何疑问需要咨询可

EMAIL: admin@rtrobot.org。

RTROBOT 用心为您服务

® robot
Welcome