# Can舵机控制板使用说明 V0.8.5

## 前言

某宝无意淘到好货: 60K机器人数字舵机, 发现无法驱动。

基于网络整理了一些资料。

仅供学习交流使用,禁止商业用途!

串口预留: 支持WIFI, 蓝牙, 红外、433等外部控制。

支持树莓派, UNO, Stm32、51等任意单片机串口控制。

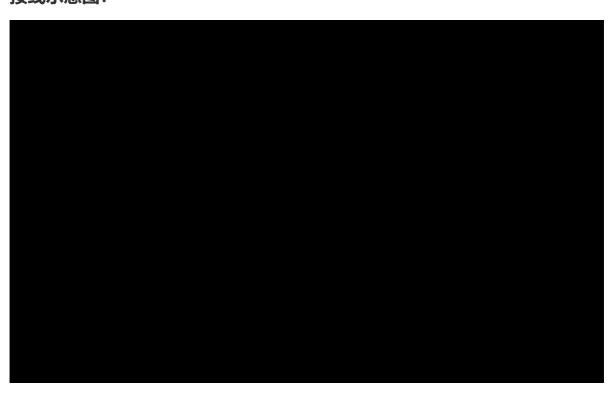
控制板就是管理多个舵机协调工作的系统,给个指令就可以控制,非常的友善。

## 驱动板说明

### 电源建议:

电源:使用24V3A动力电源、或者6S电池组(6节18650电池)。

### 接线示意图:



### 控制指令

```
为了简单明了,这里挑选几个常用命令:
//-----语法规则-----
// Command start with A9 9A
// \{len\} = cmd[2] = count(\{len\} xx .. xx)
// \{sum\} = cmd[\{len\} + 2] = sum(\{len\} xx .. xx)
// Size = A9 9A [{len} ...] {sum} ED = [{len} ...] + 4 = len + 4
// minimum : A9 9A 2 {cmd} {sum} ED = 6
// Command set:
//-----Mp3播放-----
// 32 - Stop MP3 (fix)
// 32 - Stop MP3 (fix) : A9 9A 02 32 34 ED 停止播放MP3
// 33 - Play File (fix) : A9 9A 04 33 01 03 3B ED 播放SD卡/01/003.mp3
                             : A9 9A 02 32 34 ED 停止播放MP3
01月录
//
                               A9 9A 04 33 01 04 3B ED 播放SD卡/01/004.mp3
01目录
// 34 - Play MP3 File (fix)
                             : A9 9A 03 34 01 38 ED
                                                       播放SD卡/MP3/001.mp3
mp3目录
                               A9 9A 03 34 FF 38 ED
                                                       播放SD卡/MP3/255.mp3
mp3目录
//-----动作控制-----
                              : A9 9A 03 41 01 45 ED
: A9 9A 03 41 03 47 ED
// 41 - Play Action
                                                           播放:动作组1
                                                           播放:动作组3
// 42 - Play PlayCombo
                              : A9 9A 04 42 01 02 49 ED
                                                           1号动作组,循环播放2次
[循环播放:ID,次数]
                                                            cmd[5]=0xFF 就是无限
循环

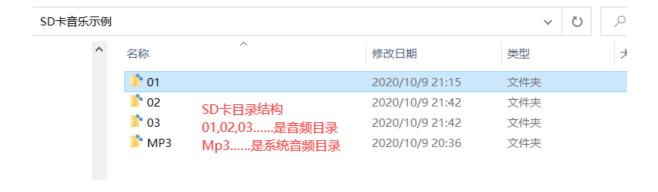
      4F - Stop playing
      : A9 9A 02 4F 51 ED

      75 - DeleteActionFile
      : A9 9A 03 75 06 7E ED

// 4F - Stop playing
                                                            停止播放: 动作组
                                                           删除6号动作组
// 84 - 播放指定动作组,特定的动作 : A9 9A 05 84 03 00 01 8D ED 播放3号动作组第1个pos
                                                            (适合做围棋机械臂、货
物多位置堆放等)
                               :cmd[4]=指定的动作组id
                               :cmd[6]=动作组里面指定的动作id
                               :cmd[9]=效验值
// 61 - Read Action Header (fix) : A9 9A 03 61 06 6A ED 读取6号动作组名称数
// 62 - Read Action Data (fix) : A9 9A 05 62 02 00 01 6A ED
                                                             读取2号动作组第01个
动作数据
//-----舵机控制-----
// 88 - 舵机查询
                             : A9 9A 05 88 03 00 02 92 ED 查询, (返回舵机ID,
角度)
// 86 - Can Servo move : A9 9A 09 88 06 02 01 5A 00 E8 03 DF ED 2号舵机移
动到90度
                               A9 9A 09 88 06 02 01 B4 00 E8 03 39 ED 2号舵机移
动到180度
// 89 - 修改ID
                             : A9 9A 05 89 03 01 02 94 ED
                                                            修改ID: 1号舵机修改
成2号 01-->02
                              : A9 9A 05 88 03 01 02 93 ED
// 88 - 查询
                                                            查询1
// 21 - 锁定
                              : A9 9A 03 21 02 26 ED
                                                             锁定: 2号舵机
// 22 - 解锁
                             : A9 9A 03 22 02 27 ED
                                                          解锁: 2号舵机
```

```
// 88 - 设置 0 度 : A9 9A 07 88 04 02 0A 00 00 9F ED 2号舵机设置0度
// 88 - 播放POS
                           : A9 9A 05 84 06 00 01 90 ED
                                                         播放:动作组6 第1个
pos [6,1]
// 11 - 查询所有舵机角度
                           : A9 9A 02 11 13 ED
                              A9 9A 08 11 FF 00 B6 01 FF 00 CE ED 返回数据包【3个
舵机的数据包,根据舵机数量动态计算出数据包】
                           :A9 9A 09 96 06 01 03 5A 00 E8 03 EE ED
// 96 - 多id舵机,多角度控制,
//1个舵机,3#移动90度
    (适合手臂等多舵机协同工作) A9 9A 0C 96 09 02 02 03 5A 00 5A 00 E8 03 51 ED
//2个舵机,2#3#移动90度
                              A9 9A 15 96 12 05 02 03 04 05 0E 5A 00 5A 00 5A 00
5A 00 5A 00 E8 03 8B ED //5个舵机,2#3#4#5#15#,移动90度
                              : cmd[2]=指令总长度
                                               [包含校验位]
                              :cmd[4]=指令长度
                                              [不包含校验位]
                              :cmd[5]=舵机数量
                              :cmd[6]=id1 cmd[7]=id2 cmd[8]=id3 cmd[9]=id4
cmd[10]=id5
                              :cmd[11]=ANGLE1_L 低字节
                              :cmd[12]=ANGLE1_H 高字节
                              :cmd[13]=ANGLE2_L 低字节
                              :cmd[14]=ANGLE2_H 高字节
                              :cmd[15]=ANGLE3_L 低字节
                              :cmd[16]=ANGLE3_H 高字节
                              :cmd[17]=ANGLE4_L 低字节
                              :cmd[18]=ANGLE4_H 高字节
                              :cmd[19]=ANGLE5_L 低字节
                              :cmd[20]=ANGLE5_H 高字节
                              :cmd[21]=moveTime_L 低字节
                              :cmd[22]=moveTime_H 高字节
// 97 - 手臂Led控制
                            :A9 9A 07 97 03 6F 07 A5 00 97 ED 6F左手 A5 白色闪
                            :A9 9A 07 97 03 6E 07 A2 00 B8 ED 6E右手 A2 红闪长
        (手臂专用)
亮
                               :cmd[5] id
                               :cmd[7] led数据 AO为关闭led,颜色很多A1-B2 ,喜欢的可
以自行测试
                               :cmd[9] 校验值
                               A0 关闭led A1 蓝色长亮 A2 红闪 A3黄闪
                                                                    Α4
绿闪 A5 白色长亮 A6 绿闪(慢)
                              A7 关闭led A8 白闪(慢) A9 黄闪(慢) AA 红闪(慢) AD
蓝色长亮 AE 绿色长亮
*/
```

			SU	M									
舵机控制指令	A9 9A 09	88 06	02 01	00 00	E8 (	3 85	ED	00 00 E8 03		0°	范围 0-3	59 对应舵	机的 0-359度
	A9 9A 09	88 06	02 01	5A 00	E8 (	03 DF	ED	5A 00 E8 03	3	90°	E8 03 对原	<u>v</u> 1000ms	
	A9 9A 09	88 06 0	02 01	B4 00	E8 (	39	ED	B4 00 E8 03		180°			
	A9 9A 09	88 06 0	02 01	0E 01	E8 (	3 94	ED	0E 01 E8 03		270°			
	A9 9A 09	88 06	02 01	67 01	E8 (	)3 ED	ED	67 01 E8 03		359°			
	約24月。	ID		角度									



## Arduino编程示列

编程非常方便。

通过编程来管理控制板是非常的方便的,发送一条指令就可以控制了。

### 列一: UNO编程

```
//使用软串口
//V-V(5V) G-G T-R R-T
#include <SoftwareSerial.h>
SoftwareSerial softser = SoftwareSerial(11, 12);
                                                        //软串口
byte Cmd_1[7] = {0XA9,0X9A,0X03,0X41,0X02,0X46,0XED}; //播放: 动作组2
byte Cmd_2[6] = \{0XA9, 0X9A, 0X02, 0X4F, 0X51, 0XED\};
                                                       //停止播放:动作组、动作
void setup() {
 softser.begin(115200);
 delay(10);
}
void loop() {
 softser.write(Cmd_1,7);
 delay(1000*13);
  softser.write(Cmd_2,6);
 delay(3);
}
```

## 列二: Mage2560

```
//使用串口3
//V-V(5V) G-G T-R R-T
#include <Arduino.h>

byte Cmd_1[7] = {0XA9, 0X9A, 0X03, 0X41, 0X02, 0X46, 0XED}; //播放: 动作组2
byte Cmd_2[6] = {0XA9, 0X9A, 0X02, 0X4F, 0X51, 0XED}; //停止播放: 动作组、动作
byte Cmd_3[13] = {0XA9, 0X9A, 0X09, 0X88, 0X06, 0X00, 0X01, 0X5A, 0X00, 0XF4, 0X01, 0XE7, 0XED}; //2号舵机90度
byte Cmd_4[13] = {0XA9, 0X9A, 0X09, 0X88, 0X06, 0X00, 0X01, 0XB4, 0X00, 0XF4, 0X01, 0X41, 0XED}; //2号舵机180度
```

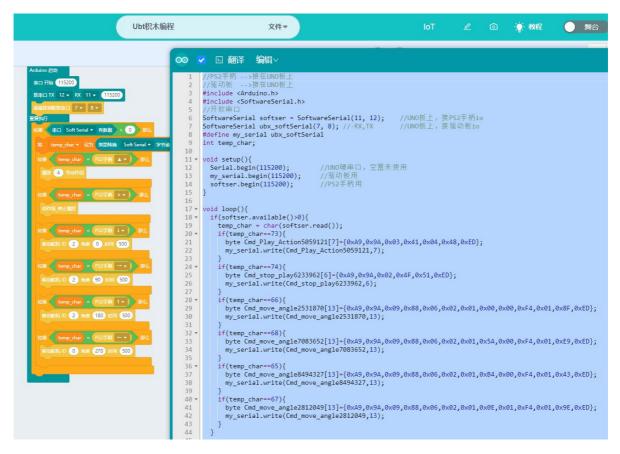
```
void setup() {
    Serial3.begin(115200);
    delay(6000);
}

void loop() {
    for (int i = 0; i < 8; i++)
    {
        delay(8000);
        Serial3.write(Cmd_3, 13);
        delay(8000);
        Serial3.write(Cmd_4, 13);
        delay(8000);
}</pre>
```

### 列三: PS2手柄

```
//PS2手柄 -->接在UNO板上
//驱动板 -->接在UNO板上
#include <Arduino.h>
#include <SoftwareSerial.h>
//开软串口
SoftwareSerial softser = SoftwareSerial(11, 12); //UNO板上,接PS2手柄io
SoftwareSerial ubx_softSerial(7, 8); //-RX,TX
                                                  //UNO板上,接驱动板io
#define my_serial ubx_softSerial
int temp_char;
void setup(){
 Serial.begin(115200); //UNO硬串口,空置未使用
 my_serial.begin(115200);
                            //驱动板用
 softser.begin(115200);
                             //PS2手柄用
}
void loop(){
  if(softser.available()>0){
   temp_char = char(softser.read());
   if(temp_char==73){
     byte Cmd_Play_Action5059121[7]={0xA9,0x9A,0x03,0x41,0x04,0x48,0xED};
     my_serial.write(Cmd_Play_Action5059121,7);
   }
   if(temp_char==74){
     byte Cmd_stop_play6233962[6]=\{0xA9,0x9A,0x02,0x4F,0x51,0xED\};
     my_serial.write(Cmd_stop_play6233962,6);
   }
   if(temp_char==66){
     byte Cmd_move_angle2531870[13]=
\{0xA9,0x9A,0x09,0x88,0x06,0x02,0x01,0x00,0x00,0xF4,0x01,0x8F,0xED\};
     my_serial.write(Cmd_move_angle2531870,13);
   }
   if(temp_char==68){
     byte Cmd_move_angle7083652[13]=
\{0xA9,0x9A,0x09,0x88,0x06,0x02,0x01,0x5A,0x00,0xF4,0x01,0xE9,0xED\};
     my_serial.write(Cmd_move_angle7083652,13);
   }
   if(temp_char==65){
      byte Cmd_move_angle8494327[13]=
\{0xA9,0x9A,0x09,0x88,0x06,0x02,0x01,0xB4,0x00,0xF4,0x01,0x43,0xED\};
```

```
my_serial.write(Cmd_move_angle8494327,13);
}
if(temp_char==67) {
    byte Cmd_move_angle2812049[13]=
{0xA9,0x9A,0x09,0x88,0x06,0x02,0x01,0x0E,0x01,0xF4,0x01,0x9E,0xED};
    my_serial.write(Cmd_move_angle2812049,13);
}
}
```

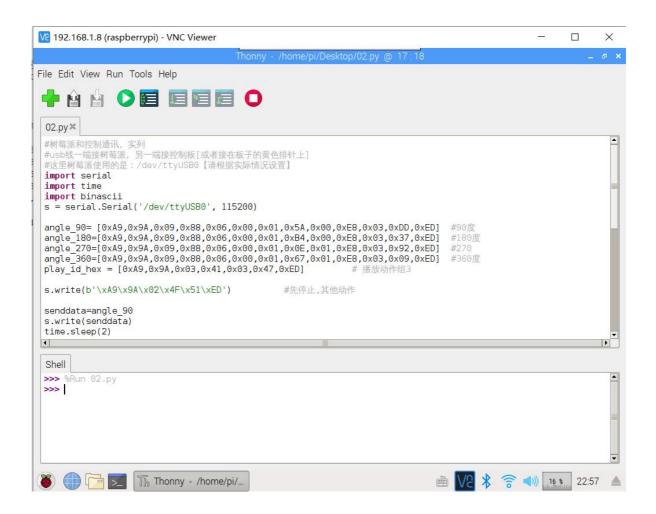


## 列四: 树莓派编程

```
#树莓派和控制通讯、实列
#usb线一端接树莓派,另一端接控制板[或者接在板子的黄色排针上]
#这里树莓派使用的是: /dev/ttyUSBO【请根据实际情况设置】
import serial
import time
import binascii
s = serial.Serial('/dev/ttyUSBO', 115200)

angle_90= [0xA9,0x9A,0x09,0x88,0x06,0x00,0x01,0x5A,0x00,0xE8,0x03,0xDD,0xED] #90度
angle_180=[0xA9,0x9A,0x09,0x88,0x06,0x00,0x01,0xB4,0x00,0xE8,0x03,0x37,0xED] #180度
angle_270=[0xA9,0x9A,0x09,0x88,0x06,0x00,0x01,0xB4,0x00,0xE8,0x03,0x92,0xED] #270度
angle_360=[0xA9,0x9A,0x09,0x88,0x06,0x00,0x01,0x0E,0x01,0xE8,0x03,0x09,0xED] #360度
play_id_hex = [0xA9,0x9A,0x09,0x88,0x06,0x00,0x01,0x67,0x01,0xE8,0x03,0x09,0xED] #360度
s.write(b'\xA9\x9A\x02\x4F\x51\xED') #先停止,其他动作
```

```
senddata=angle_90
s.write(senddata)
time.sleep(2)
senddata=angle_180
s.write(senddata)
time.sleep(2)
senddata=angle_270
s.write(senddata)
time.sleep(2)
senddata=angle_360
s.write(senddata)
time.sleep(2)
#senddata=play_id_hex
#s.write(senddata)
#time.sleep(2)
s.close()
```



## 示教动作组

#### 示教功能分2种情况:

第一种,在电脑上用上位机操作。配套资料包路面有操作视频资料里面有;

第二种, 离线示教。配合红外遥控器操作非常的方便。

这里我们来讨论第二种方式的实现。

#### 上位机软件右边依次点击: 主控板-->特殊动作-->事件处理系统:

默认情况下:示教动作组设置为6#【可以自己修改】,编辑动作的过程中OLED屏幕上会显示动作组Id号,第几条记录。

#### 红外遥控器设置如下:

\*: 所有舵机解锁。【舵机解锁后才能拖动关节】

OK: 示教动作添加1条Pos记录

4: 删除动作组6#, 【动作不满意,用次命令删除动作组重来】

6:播放动作组6#,【播放刚才编辑好的动作组】

如果打开"**事件处理系统**"里面没有加载积木设置,也可以在"**事件处理系统**"窗口的左下角点击"**载入档案**",加载文件选择:"**示教动作组\_调试**"来手动加载。

#### 示教流程 (默认绑定到6#动作组):

- 1.按"\*"键,解锁所有舵机。
- 2.用手搬动舵机关节到指定的位置。
- 3.按"OK"键,添加一帧动作数据。
- ......循环第2,3步。直到动作组编辑完成。
- 4.按"6"键,查看刚才拖动示教的6#动作组。

如果动作不满意,按"4"键,删除对应的6#动作组,重新拖动示教即可。

编辑好了, 如想更细微的调整可在上位机软件里面进行操作。



# 升级固件

如果正常使用过程中,又不想体验新功能的情况下,可以选择不升级固件。有时候新的固件程序还需要更新对应的上位机软件。(是否更新请谨慎考虑。)

#### 固件更新方式:

从上位机软件左上角菜单选择: 文件----->机器人固件烧录----->

烧录固件的窗口出来后:**窗口左边选择控制板对应的串口----->在线升级。** 

