

# HTS-35H 总线串行舵机



## 1. 产品说明

### 1.1 产品简介

这是一款常见的串行总线舵机，它是由串口指令进行控制的，串口波特率为 115200。用户可根据我司提供的通信协议，发送对应的指令给舵机，即可控制舵机转动，或者读取舵机信息。在控制舵机之前，我们需要设置好舵机的各项参数及 ID。

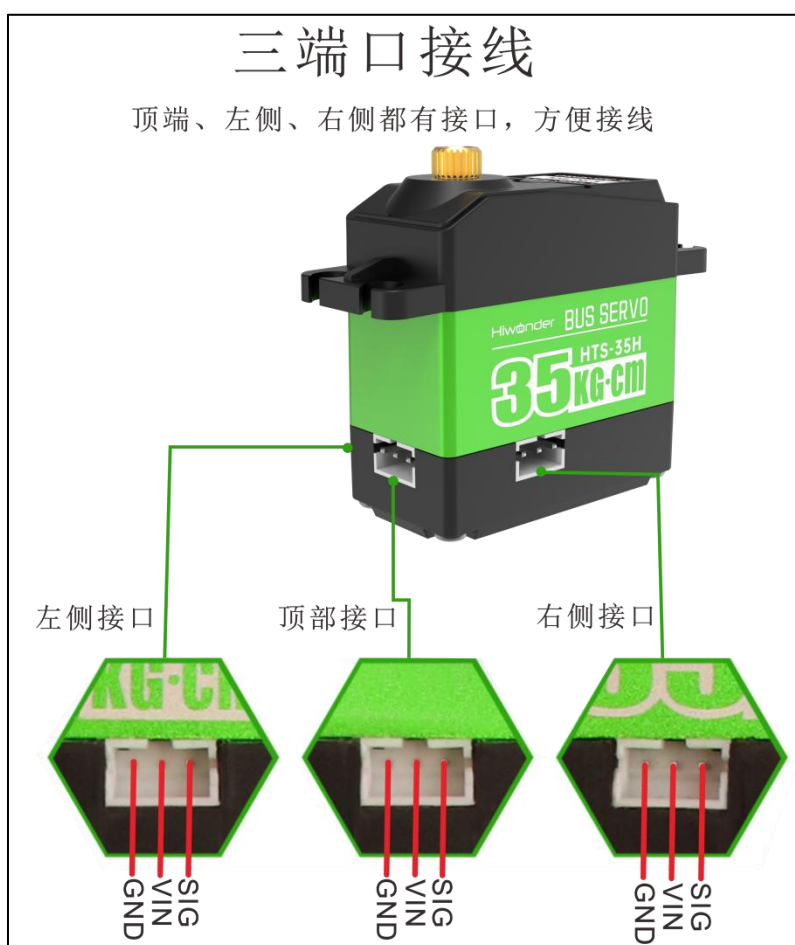
这个舵机的接口为半双工 UART 异步串行接口，信号端既可以发送也可以接收信号。使用时，我们可以根据不同的 ID 通过串口发送不同的指令，从而达到单独控制舵机的效果。这款舵机应用广泛，主要用于各类仿生机器人的关节设计。

### 1.2 舵盘安装和接口说明

舵盘安装参考下面图片，对准红十字架安装即可。



接口分布和说明，可参考下图和下表：



引脚	引脚说明
GND	电源地
VIN	电源输入
SIG	信号端(半双工 UART 异步串行接口)

### 1.3 舵机特点

#### 1. 高压舵机，更省电：

相对于传统 7.4V 的舵机，11.1V 的高压舵机可以减少 60%以上的电流，使得机器人的续航时间大大提升。

#### 2. 串行总线接口：

控制板引出一个 I/O 连接串行舵机，舵机之间通过三接口进行上下串联。这样可以减少串口的占用。简单的布线，让产品更加简单、精致、美观。

#### 3. ID 识别，总线通讯：

每个舵机都可以设置 ID 号，用于舵机的识别。舵机 ID 默认为 1，可以自行修改。控制器与舵机之间采用单总线方式通讯，通讯波特率为 115200。用户可以对每个舵机分配一个 ID 号，控制器发出的指令中包含 ID 信息。只有匹配上 ID 号的舵机才能完整接受这个指令，并按照指令执行动作。

#### 4. 高精度电位器：

舵机内部采用进口高精度电位器，来作为角度反馈。舵机的精度和线性度都很好，使得机器人运行更加稳定。舵机的使用寿命也大幅加长。

## 5. 超大扭力

35KG 大扭力，让你的机器人动力十足。

## 6. 位置、温度、电压反馈：

带有位置、温度反馈和电压反馈，可以实时了解舵机内部数据，来对舵机进行保护。

## 7. 两种工作模式：

支持舵机模式和减速电机模式。

1) 在舵机模式下可以在 240 度范围内控制转动定位；

2) 在减速电机模式下可以 360 度连续转动，可以控制转动方向和速度。

## 8. 金属齿轮：

高精度的齿轮镶嵌减少了齿轮摩擦带来的噪声。

## 9. 金属外壳：

绿色氧化金属外壳，散热能力强，外观炫酷。

# 1.4 通讯协议

这款舵机兼容幻尔科技总线舵机通信协议，大家可以在本目录下查看具体协议。

# 2.技术参数

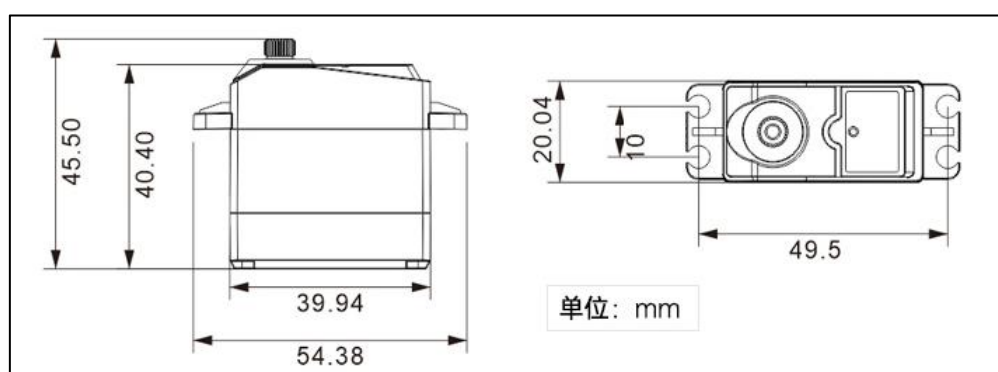
## 2.1 规格参数

工作电压	DC 9-12.6V
转动速度	0.18sec/60° （DC 11.1V）

转动扭矩	35kg.cm (DC 11.1V)
静态最大扭力	35kg.cm (DC 11.1V)
转动范围	0~ 240°
空载电流	100mA
堵转电流	3A
舵机精度	0.2°
控制角度范围	0-1000, 对应 0~ 240°
控制方式	UART 串口指令
通信波特率	115200
存储	掉电保存用户设置
舵机 ID	0-253 用户可设置, 默认为 1
回读功能	支持角度回读
保护	堵转保护/过温保护
参数反馈	温度、电压、位置
工作模式	舵机模式和减速电机模式
齿轮类型	金属齿

配套线长	20cm，可选其他线长
插件型号	PH2.0-3P
产品重量	64g
产品尺寸	54.38mm*20.14mm*45.5mm
适用于	各类仿生机器人关节

## 2.2 舵机尺寸图



## 3.案例使用

为了帮助大家快速上手，我们提供 2 个和 UNO 主板搭配使用的案例。案例 1 是通过 UNO 主板设置舵机 ID，案例 2 是通过 UNO 主板控制 2 个不同 ID 舵机进行转动。

### 3.1 准备工作

#### 3.1.1 硬件准备

使用前需要准备以下配件：

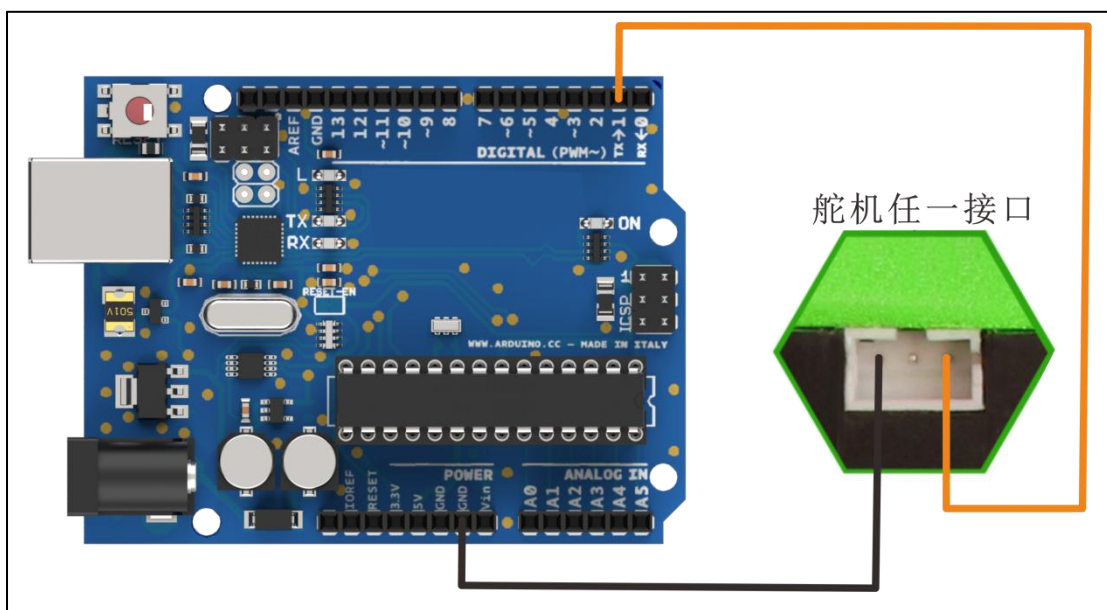
- ① UNO 主板 1 块

- ② HTS-35H 总线串行舵机 2 个
- ③ USB 数据线 1 根
- ④ 公对公杜邦线 3 根
- ⑤ 配套接线 2 根

### 3.1.2 接线准备

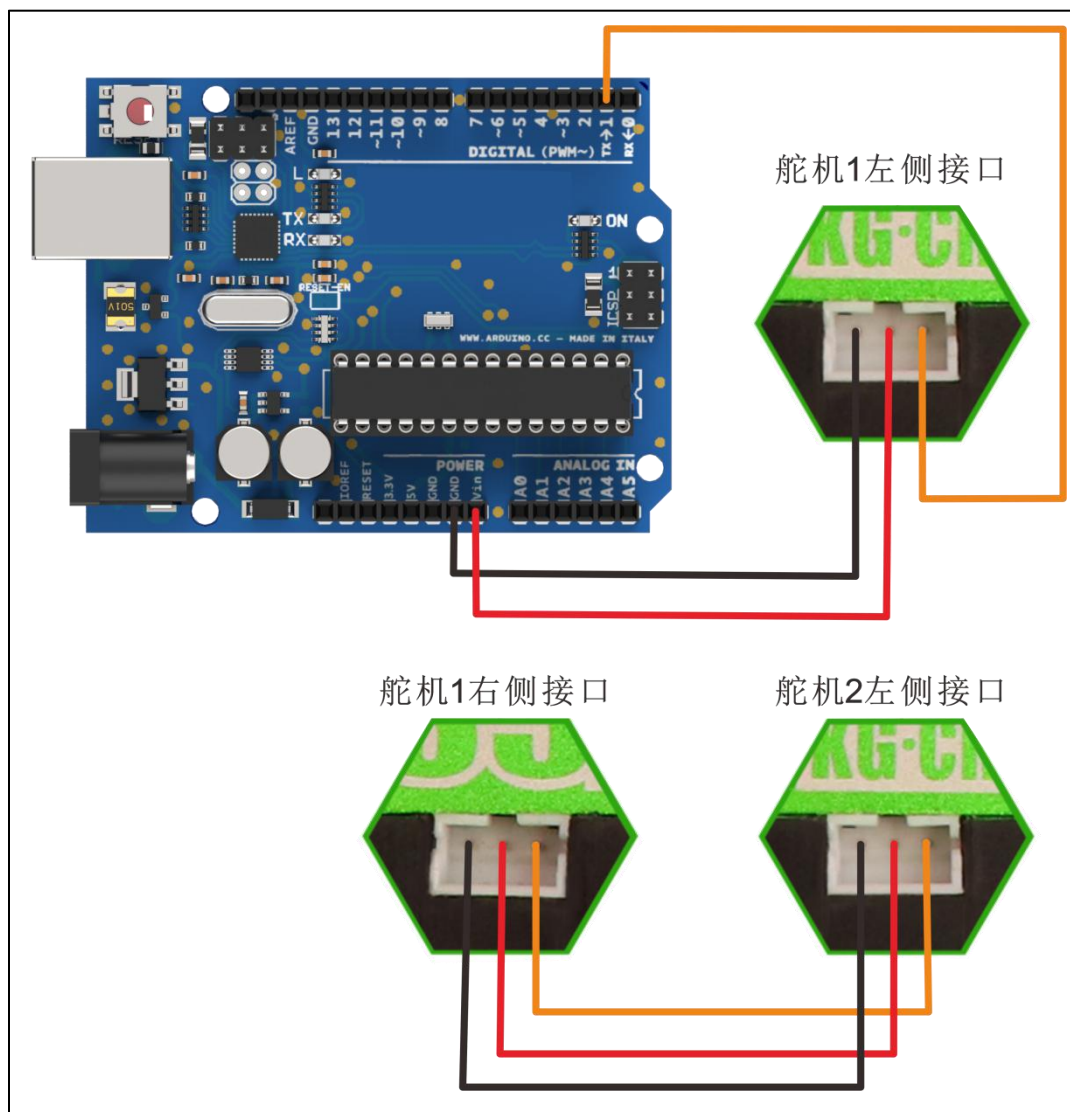
案例 1 接线图（ID 设置）

通过公对公杜邦线将 1 个舵机连接至 UNO 主板上，连接示意图如下所示：



案例 2 接线图（舵机转动）

通过公对公杜邦线将 2 个舵机连接 UNO 主板上，连接示意图如下所示：



## 3.2 案例 1-ID 设置

本款舵机出厂默认 ID 为 1，这里我们以修改为 ID2 为例进行示范。

### 3.2.1 实验流程

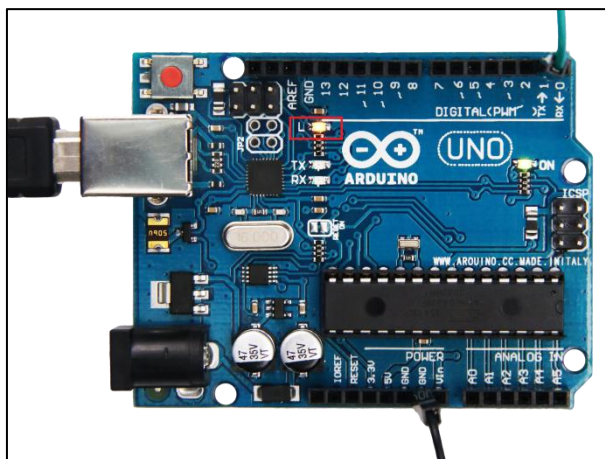
- 1) 在电脑上下载并安装 Arduino IDE，具体的操作方法，可以本目录中“xxx”文档。
- 2) 参照接线图 1 将 1 个 HTS-35H 总线串行舵机连接在 UNO 主板上。
- 3) 将 UNO 主板通过 USB 线连入电脑。打开 Arduino IDE，依次点击“File->New”，将“3.2.3 示例代码”复制进去。
- 4) 选择正确的开发板及端口，将程序编译并上传。



5) 代码上传成功后，观察现象。

### 3.2.2 示例结果

上传成功后，舵机的 ID 由默认的 1 改为 2，UNO 主板上的 D13 指示灯开始闪烁。



### 3.2.3 示例代码

```
/******HTS-35H 串口舵机测试程序 1*****  
  
* Arduino 型号: Arduino UNO  
  
*****/  
  
#define GET_LOW_BYTE(A) (uint8_t)((A))  
  
//宏函数 获得 A 的低八位  
  
#define GET_HIGH_BYTE(A) (uint8_t)((A) >> 8)  
  
//宏函数 获得 A 的高八位  
  
#define BYTE_TO_HW(A, B) (((uint16_t)(A)) << 8) | (uint8_t)(B))  
  
//宏函数 以 A 为高八位 B 为低八位 合并为 16 位整形  
  
#define LOBOT_SERVO_FRAME_HEADER          0x55  
  
#define LOBOT_SERVO_ID_WRITE              13
```

```
byte LobotChecksum(byte buf[])

{

    byte i;

    uint16_t temp = 0;

    for (i = 2; i < buf[3] + 2; i++) {

        temp += buf[i];

    }

    temp = ~temp;

    i = (byte)temp;

    return i;

}

void LobotSerialServoSetID(HardwareSerial &SerialX, uint8_t oldID, uint8_t newID)

{

    byte buf[7];

    buf[0] = buf[1] = LOBOT_SERVO_FRAME_HEADER;

    buf[2] = oldID;

    buf[3] = 4;

    buf[4] = LOBOT_SERVO_ID_WRITE;

    buf[5] = newID;

    buf[6] = LobotChecksum(buf);

    SerialX.write(buf, 7);

}
```

```
}

void setup() {

    // put your setup code here, to run once:

    Serial.begin(115200); //波特率 115200

    pinMode(13, OUTPUT);

    delay(1000);

}

void loop() {

    // put your main code here, to run repeatedly:

    delay(500);

    digitalWrite(13,HIGH); //指示灯，运行指示

    LobotSerialServoSetID(Serial, 1, 2); // 第一个参数为通信所用串口，第二个参数为旧
ID（旧 ID 为 1 就是向所有在线的舵机广播此命令，对所有在线舵机有效）

    // 第三个参数为新的 ID

    delay(500);

    digitalWrite(13,LOW);

}
```

### 3.3 案例 2-舵机转动

本实验，我们控制 2 个不同 ID 的舵机进行转动。

#### 3.3.1 实验流程

- 1) 参照接线图 2 将 2 个 HTS-35H 总线串行舵机连接在 UNO 主板上。

- 2) 参照前面方法将“3.3.3 示例代码”上传到 UNO 主板上。
- 3) 代码上传成功后，观察现象。

### 3.3.2 示例结果

上传成功后，ID1 的舵机和 ID2 的舵机的舵机角度不断变换，每隔 1 秒变换 1 次。ID1 的舵机的角度变换幅度明显大于 ID2 的舵机。

### 3.3.3 示例代码

```
/******HTS-35H 串口舵机测试程序 2*****  
  
* Arduino 型号: Arduino UNO  
  
*****/  
  
#define GET_LOW_BYTE(A) (uint8_t)((A))  
  
//宏函数 获得 A 的低八位  
  
#define GET_HIGH_BYTE(A) (uint8_t)((A) >> 8)  
  
//宏函数 获得 A 的高八位  
  
#define BYTE_TO_HW(A, B) (((uint16_t)(A)) << 8) | (uint8_t)(B))  
  
//宏函数 以 A 为高八位 B 为低八位 合并为 16 位整形  
  
#define LOBOT_SERVO_FRAME_HEADER          0x55  
  
#define LOBOT_SERVO_MOVE_TIME_WRITE      1  
  
byte LobotChecksum(byte buf[])  
  
{  
  
    byte i;  
  
    uint16_t temp = 0;  
  
    for (i = 2; i < buf[3] + 2; i++) {
```

```
        temp += buf[i];

    }

    temp = ~temp;

    i = (byte)temp;

    return i;

}

void LobotSerialServoMove(HardwareSerial &SerialX, uint8_t id, int16_t position, uint16_t
time)
{

    byte buf[10];

    if(position < 0)

        position = 0;

    if(position > 1000)

        position = 1000;

    buf[0] = buf[1] = LOBOT_SERVO_FRAME_HEADER;

    buf[2] = id;

    buf[3] = 7;

    buf[4] = LOBOT_SERVO_MOVE_TIME_WRITE;

    buf[5] = GET_LOW_BYTE(position);

    buf[6] = GET_HIGH_BYTE(position);

    buf[7] = GET_LOW_BYTE(time);

    buf[8] = GET_HIGH_BYTE(time);
```

```
    buf[9] = LobotChecksum(buf);

    SerialX.write(buf, 10);

}

void setup() {

    // put your setup code here, to run once:

    Serial.begin(115200);

    delay(1000);

}

#define ID1    1

#define ID2    2

void loop() {

    // put your main code here, to run repeatedly:

    LobotSerialServoMove(Serial, ID1, 100, 500);

    LobotSerialServoMove(Serial, ID2, 500, 500);

    delay(1000);

    LobotSerialServoMove(Serial, ID1, 500, 500);

    LobotSerialServoMove(Serial, ID2, 600, 500);

    delay(1000);

    LobotSerialServoMove(Serial, ID1, 900, 500);

    LobotSerialServoMove(Serial, ID2, 700, 500);

    delay(1000);
```

```
LobotSerialServoMove(Serial, ID1, 500, 500);  
  
LobotSerialServoMove(Serial, ID2, 600, 500);  
  
delay(1000);  
  
}
```

## 4.常见问题

Q1: 测试时，代码上传成功后，测试无现象？

A: 检查接线，舵机信号端需接控制板的 D1 接口。

Q2: 测试时，我们可以串联几个舵机？

A: 理论来说，最多可以同时串联 253 个舵机。

Q3: 测试案例 2 时，我们应该如何给 UNO 主板供电？

A: 测试案例 2 时，我们给黑色 DC 头外接电源，同时给 UNO 主板和舵机供电。

UNO 主板的供电电压范围为 DC 7-12V,舵机的工作电压范围为 DC 9-12.6V。我们提供的电源电源，需要满足两个要求，因此我们需要提供的外接电源电压范围为 DC 9-12V。

Q4: 测试案例 2 时，两个舵机现象为什么一样？

A: 测试前，两个舵机默认 ID 是 1，我们需要按照案例 1 的方法，把其中一个舵机 ID 改为 2。测试案例 1 时，特别注意以下几点：

(1) 使用的杜邦线不能太长。

(2) 接好并上传好代码后，UNO 主板 D13 开始闪烁后，不要立刻拔掉接线，等待一会，确保 ID 更改完成后，再拔线。