



## HiSpark-智能小车套件 舵机编程指南

文档版本 00B01

发布日期 2020/8/3

**版权所有 © 上海海思技术有限公司。保留一切权利。**

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## **商标声明**



**HISILICON**、海思和其他海思商标均为上海海思技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## **注意**

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## **上海海思技术有限公司**

地址：上海市青浦区金泽镇（西岑）水秀路 318 号 101 室 邮编：201718

网址：<http://www.hisilicon.com>



# 前言

## 概述

本文档主要介绍基于海思 WiFi 芯片 Hi3861 开发的 HiSpark-WiFi-IoT 套件演示指导书。

## 产品版本

与本文档相对应的主芯片版本如下。

产品名称	产品版本
Hi3861	V100R001C00SPC021

## 读者对象

本文档（本指南）主要适用于以下工程师：

- 软件开发工程师
- 硬件开发工程师

## 修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2020-08-3	00B01	第一次临时版本发布。





# 目 录

前 言.....	i
1 智能小车套件舵机功能实现 .....	4
1.1 舵机硬件准备 .....	4
1.2 舵机硬件介绍 .....	5
1.3 舵机功能软件实现 .....	5



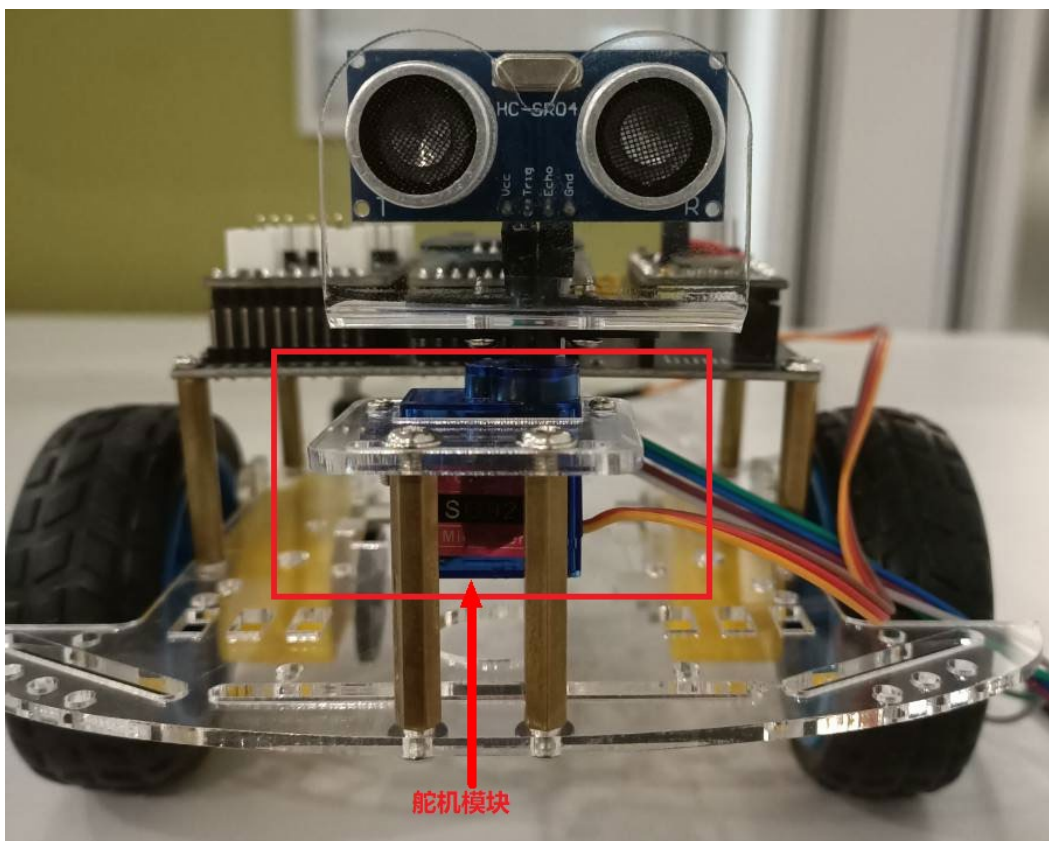
# 1 智能小车套件舵机功能实现

## 1.1 舵机硬件准备

图 1.1-1 舵机模块



图 1.1-1 舵机模块在智能小车上的安装





## 1.2 舵机硬件介绍

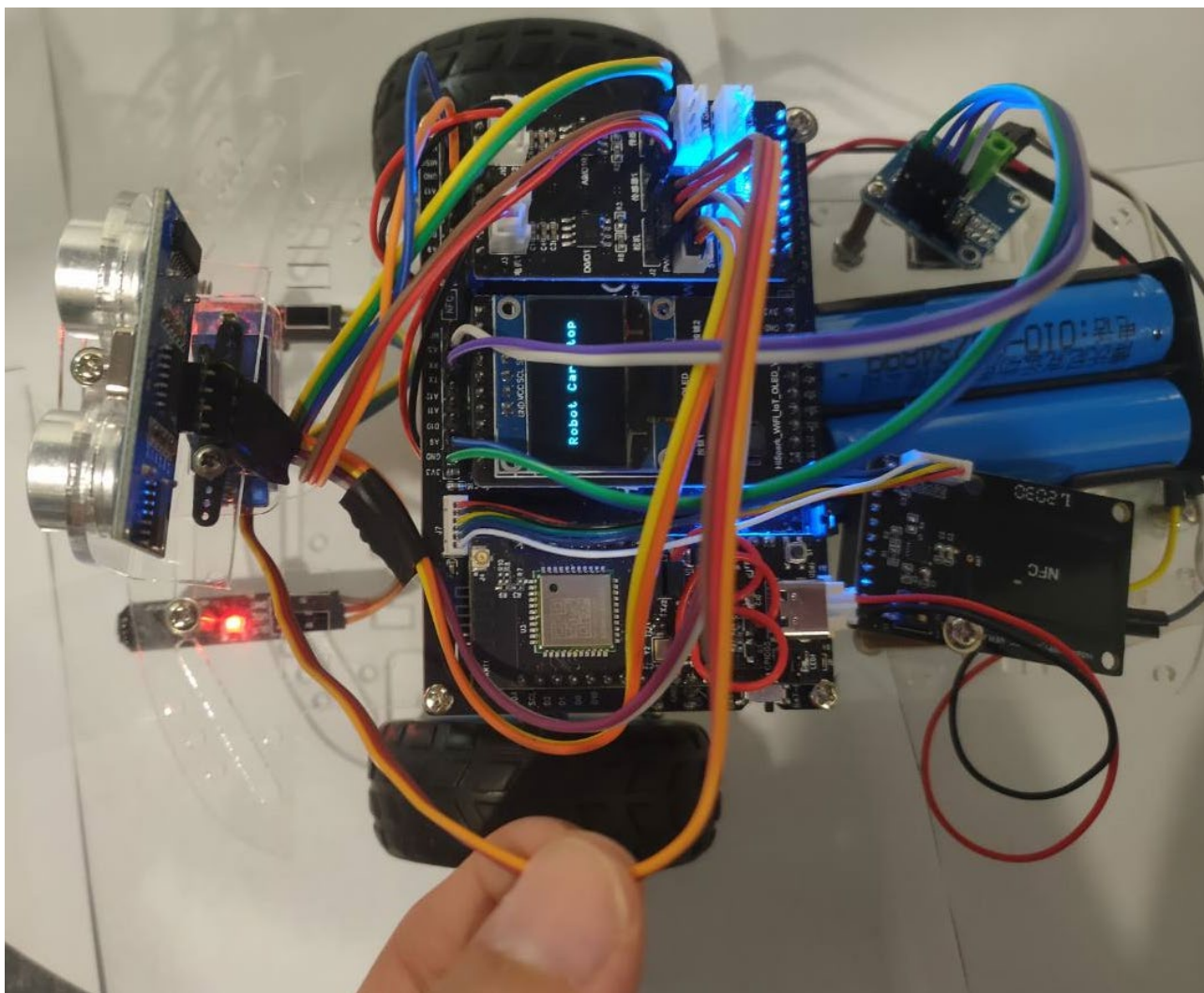
舵机的控制信号为周期是 20ms 的脉冲调制信号，其中脉冲宽度从 0.5ms-2.5ms,想对应舵盘的 90 ~ (-90) °，呈线性变化。

Demo 所用驱动舵机为 SR92R，IO 引脚的连接如下：

控制线 ----- GPIO2

舵机在小车上的连线如下所示：

图 1.2-1 舵机的接线图



## 1.3 舵机功能软件实现

(1) IO 口初始化

```
hi_u32 hi_io_set_func(hi_io_name id, hi_u8 val); //配置引脚复用
```



`hi_u32 hi_gpio_set_dir(hi_gpio_idx id, hi_gpio_dir dir);` //设置 IO 方向 (输入/输出)

**hi\_io\_name id:** gpio 硬件管脚编号, 对应的是硬件管脚上的 GPIO pin 脚, 枚举类型如下:

```
typedef enum {  
    HI_IO_NAME_GPIO_0,      /**< GPIO0 */  
    HI_IO_NAME_GPIO_1,      /**< GPIO1 */  
    HI_IO_NAME_GPIO_2,      /**< GPIO2 */  
    HI_IO_NAME_GPIO_3,      /**< GPIO3 */  
    HI_IO_NAME_GPIO_4,      /**< GPIO4 */  
    HI_IO_NAME_GPIO_5,      /**< GPIO5 */  
    HI_IO_NAME_GPIO_6,      /**< GPIO6 */  
    HI_IO_NAME_GPIO_7,      /**< GPIO7 */  
    HI_IO_NAME_GPIO_8,      /**< GPIO8 */  
    HI_IO_NAME_GPIO_9,      /**< GPIO9 */  
    HI_IO_NAME_GPIO_10,     /**< GPIO10 */  
    HI_IO_NAME_GPIO_11,     /**< GPIO11 */  
    HI_IO_NAME_GPIO_12,     /**< GPIO12 */  
    HI_IO_NAME_GPIO_13,     /**< GPIO13 */  
    HI_IO_NAME_GPIO_14,     /**< GPIO14 */  
    HI_IO_NAME_SFC_CSN,     /**< SFC_CSN */  
    HI_IO_NAME_SFC_IO1,     /**< SFC_IO1 */  
    HI_IO_NAME_SFC_IO2,     /**< SFC_IO2 */  
    HI_IO_NAME_SFC_IO0,     /**< SFC_IO0 */  
    HI_IO_NAME_SFC_CLK,     /**< SFC_CLK */  
    HI_IO_NAME_SFC_IO3,     /**< SFC_IO3 */  
    HI_IO_NAME_MAX,  
} hi_io_name;
```

**hi\_u8 val:** 对应 GPIO 引脚功能, 如果该引脚复用, 可以配置为其他的复用功能, 如: UART/PWM/I2C/SPI/SDIO 等功能;

**hi\_gpio\_idx id:** 硬件管脚编号, 对应的是硬件管脚上的 GPIO pin 脚, 枚举类型如下: ,

```
typedef enum {  
    HI_GPIO_IDX_0, /**< GPIO0*/  
    HI_GPIO_IDX_1, /**< GPIO1*/  
    HI_GPIO_IDX_2, /**< GPIO2*/  
}
```





```
HI_GPIO_IDX_3, /**< GPIO3*/  
HI_GPIO_IDX_4, /**< GPIO4*/  
HI_GPIO_IDX_5, /**< GPIO5*/  
HI_GPIO_IDX_6, /**< GPIO6*/  
HI_GPIO_IDX_7, /**< GPIO7*/  
HI_GPIO_IDX_8, /**< GPIO8*/  
HI_GPIO_IDX_9, /**< GPIO9*/  
HI_GPIO_IDX_10, /**< GPIO10*/  
HI_GPIO_IDX_11, /**< GPIO11*/  
HI_GPIO_IDX_12, /**< GPIO12*/  
HI_GPIO_IDX_13, /**< GPIO13*/  
HI_GPIO_IDX_14, /**< GPIO14*/
```

HI\_GPIO\_IDX\_MAX, /\*\*< Maximum value, which cannot be used.CNcomment:最大值，不可输入使用 CNend\*/

```
} hi_gpio_idx;
```

**hi\_gpio\_dir dir:** dir, GPIO 输出方向

```
typedef enum {
```

```
    HI_GPIO_DIR_IN = 0, /**< Input.CNcomment:输入方向 CNend*/
```

```
    HI_GPIO_DIR_OUT /**< Output.CNcomment:输出方向 CNend*/
```

```
} hi_gpio_dir;
```

在 demo 中示例：

(1) GPIO 初始化：

```
/*舵机gpio初始化*/  
hi_io_set_func(HI_IO_NAME_GPIO_2, HI_IO_FUNC_GPIO_2_GPIO);  
hi_gpio_set_dir(HI_GPIO_IDX_2, HI_GPIO_DIR_OUT);
```

(2) 向控制线引脚输出指定方波，方波输出函数：

```
hi_void set_angle(hi_s32 duty)  
{  
    hi_gpio_set_output_val(HI_GPIO_IDX_2, HI_GPIO_VALUE1);  
    hi_udelay(duty);  
    hi_gpio_set_output_val(HI_GPIO_IDX_2, HI_GPIO_VALUE0);  
    hi_udelay(20000-duty);  
}
```



周期为 20ms (20000us), duty 为高电平时间。

左转, 即舵机位于 -90°:

```
/*Steering gear turn left*/  
hi_void engine_turn_left(hi_void)  
{  
    for (int i = 0; i <10; i++) {  
        set_angle(2500);  
    }  
}
```

右转, 即舵机位于 90°:

```
/*Steering gear turn right*/  
hi_void engine_turn_right(hi_void)  
{  
    for (int i = 0; i <10; i++) {  
        set_angle(500);  
    }  
}
```

回归原始位置, 即舵机位于 0° :

```
/*Steering gear return to middle*/  
hi_void regress_middle(hi_void)  
{  
    for (int i = 0; i <10; i++) {  
        set_angle(1500);  
    }  
}
```