

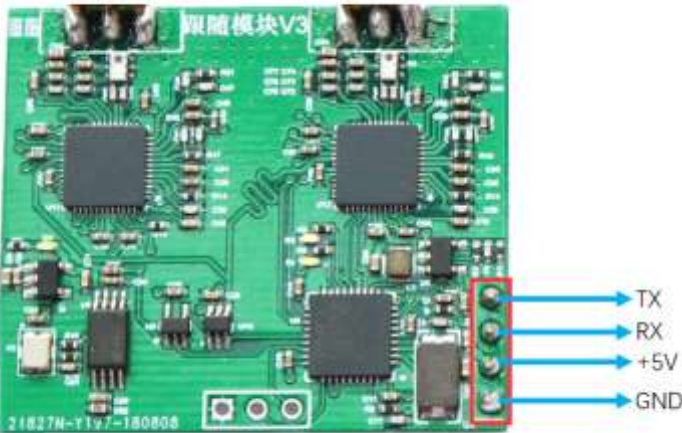
1. 产品图片



跟随模块



标签



DC5V供电口 电源开关

2. 跟随模块技术参数

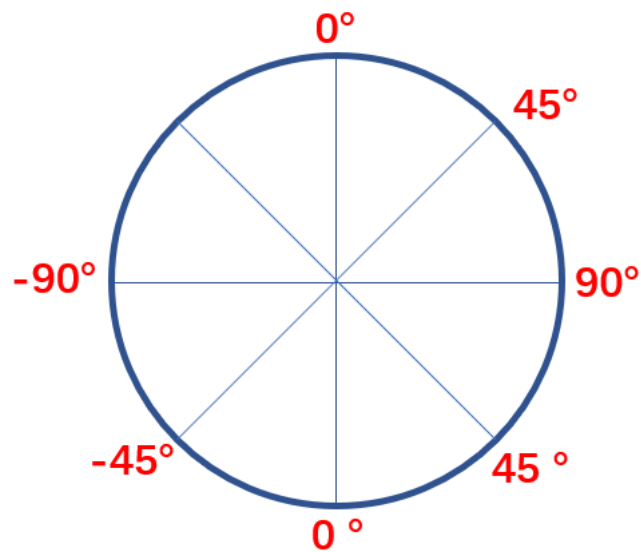
型号	HXZK-GS2
尺寸	37.5*32 毫米
天线长	39 毫米
重量	6.2 克
供电电压	DC5V
天线	全向天线
工作频段	3-7GHz
工作温度	-20°C~65°C
工作湿度	0%~90%无凝结
测距频率	1-30Hz 可调
测距精度	10-20 厘米 (无遮挡)
角度精度	±10 度
测距距离	±50 米
数据输出	RS232 串口
波特率	115200
数据位	8
停止位	1

3. 标签参数

型号	HXZK-bq2
尺寸	21*59 毫米
重量	4.5 克
供电电压	DC5V
天线	全向天线
工作频段	3-7GHz
工作温度	-20°C~65°C
工作湿度	0%~90%无凝结
测距频率	1-30Hz 可调
测距精度	10-20 厘米 (无遮挡)
角度精度	±10 度
测距距离	±50 米

4. 角度说明

跟随模块正前方为 0 度，左边 0 到-90 度，右边 0 到+90 度，前后对称。跟随模块本身无法区分前后，但是当标签处于基站背面时，小车转向时会触正反馈，加速转到正面。



5. 跟随模式输出数据格式说明

串口格式：波特率 115200，数据位 8 位，停止位 1 位。

数据输出格式见下表

包头 2 字节	距离 2 字节	角度 2 字节	标签 ID2 字节	校验 2 字节
0x6DD6	单位 cm	单位度		

注：1.数据低位在前高位在后，16 进制。

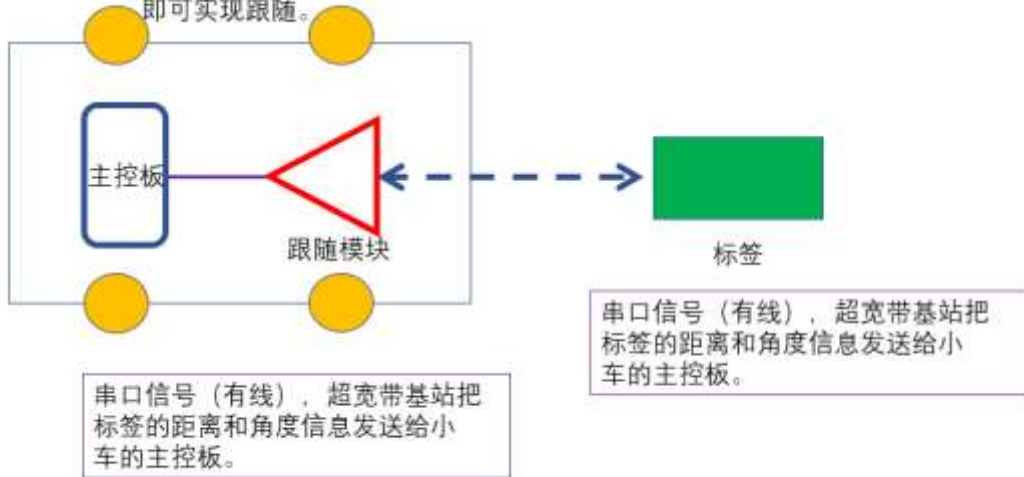
校验不包括包头。

使用 check sum 校验，校验程序如下：

```
uint16_t Checksum_u16(uint8_t* pdata, uint32_t len)
{
    uint16_t sum = 0;
    uint32_t i;
    for(i=0; i<len; i++)
        sum += pdata[i];
    sum = ~sum;
    return sum;
}
```

6. 跟随系统结构

小车的主控板在收到标签的距离和角度信息后。
当标签在小车左边时，小车向左转。
当标签在小车右边时，小车向右转。
当标签在小车正前方时，小车向前走。
到达设定好的跟随距离时，小车停止运动。
即可实现跟随。



通过串口读取到标签的距离和角度信息后，只要可以控制车体实现前进、左转、右转，即可实现跟随功能。

7. 实例代码

```
#include "Include.h"
#include <Math.h>

#define FollowDistance 120 // 跟随距离, 单位cm, 超过跟随距离小车启动跟随, 到达距离内小车停止跟随, 这个距离时标签到基站的位置
// 如果数值过小, 小车到人脚下还是会一直走。
#define FollowSpeed 1000 // 跟随速度, 最大1000, 过小电机不转。
#define RotationThreshold 30 // 触发小车转向标签的角度, 数值过小小车可能会频繁转向, 大大小车可能会翻向不转。

void IMU_Handler(void)
{
}

/**
 * Brief This function handles Hard Fault exception.
 * #param None
 * #retval None
 */
void HardFault_Handler(void)
{
    /* Go to infinite loop when Hard Fault exception occurs */
    while (1)
    {
    }
}

+ 函数名: main
* 描述 : 程序入口函数
* 输入 : 无
* 输出 : 无
* 调用 : 无
*/
extern float distanceL, distanceM, distanceR;
extern int speed_diff, speed;
extern int angle, angle2, dd;
uint16_t ii;

void main(void)
{
    /* 设置系统时钟为72M */
    SystemInit();

    SysTick_Configuration(); //SysTick中断配置, 主要用来定时

    //Delay_s(38); //等延时38秒等待WIFI模块启动

    GPIO_ALL_Config(); //GPIO初始化

    Init_Steer(); //舵机角度初始化
}
```