

BlueToothRemote 蓝牙遥控器

使用 说 明

版权申明

本手册版权归属塔克创新所有，并保留一切权力。非经(书面形式)同意，任何单位及个人不得擅自摘录或修改本手册部分或全部内容，违者我们将追究其法律责任。

感谢您购买塔克科技产品，在使用产品之前，请仔细地阅读该手册并且确保知道如何正确使用该产品，不合理的操作可能会损坏产品，使用过程中随时参考该手册以确保正确使用。此手册不断更新中，建议您登录论坛、博客、交流群或公众号下载最新版本。

塔克媒体

塔克淘宝店铺	https://shop246676508.taobao.com/
塔克官网社区	www.xtark.cn
塔克技术博客	http://www.cnblogs.com/xtark/
技术商务联系	18660035816 (微信 QQ 同号)
塔克服务邮箱	service@xtark.cn

塔克淘宝，微信公众号二维码：



更多精品 · 欢迎关注塔克淘宝店铺



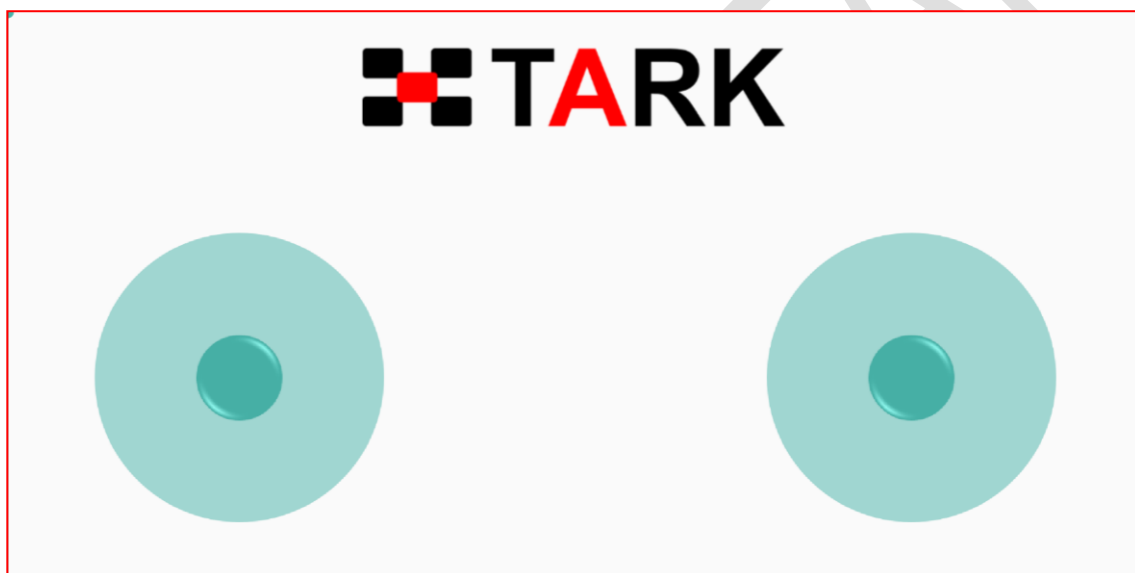
更多资讯 · 欢迎关注塔克创新微信号

1. 软件介绍



BlueToothRemote 蓝牙遥控器软件为塔克创新自主开发的一款用于智能车、机器人制作的安卓手机遥控软件，支持安卓 5.0 以上系统。

软件采用最新的 MD 设计语言，抽屉式框架，横屏完全沉浸式用户体验，具有触觉反馈功能，操控体验由于一般遥控器软件。目前软件支持摇杆操作，按钮操作，体感操作三种模式。



2. 使用说明

2.1. 蓝牙连接

首先蓝牙适配器上电，支持具有 SPP 协议的蓝牙设备，例如常用 HC-05，JDY30 等。

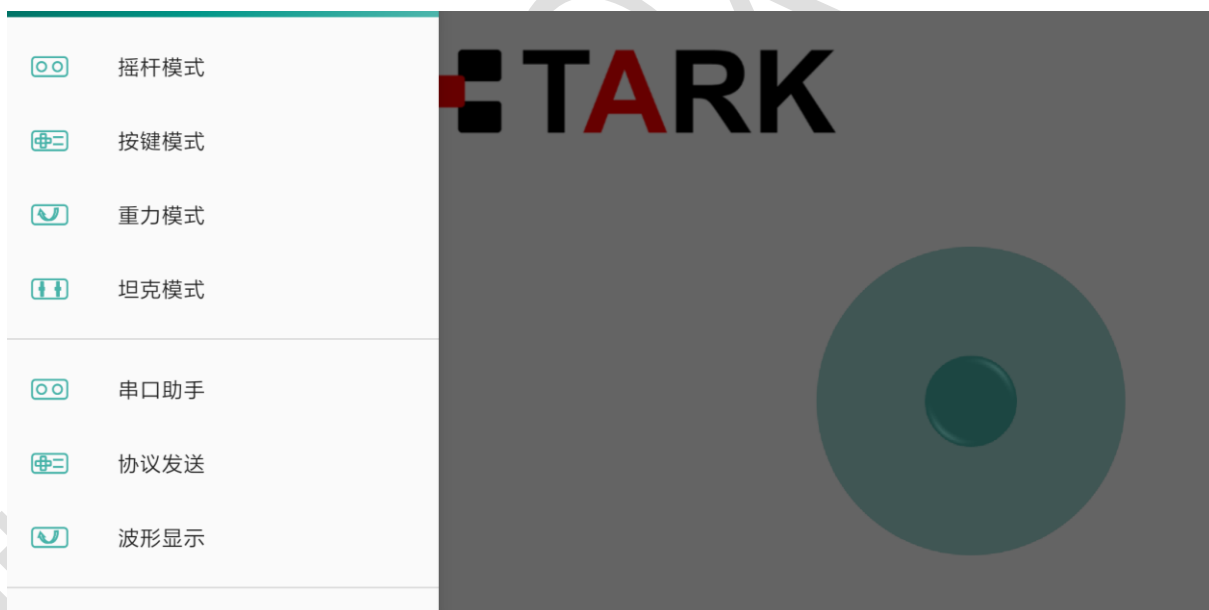
打开手机蓝牙，并搜索到蓝牙设备，例如 JDY30，连接配对，一般默认密码为 0000 或 1234。

然后打开该软件连接蓝牙，例如下图 JDY30。此时软件会提示蓝牙连接进度，连接成功和失败都会有相应提示。如果连接成功，会显现如下。

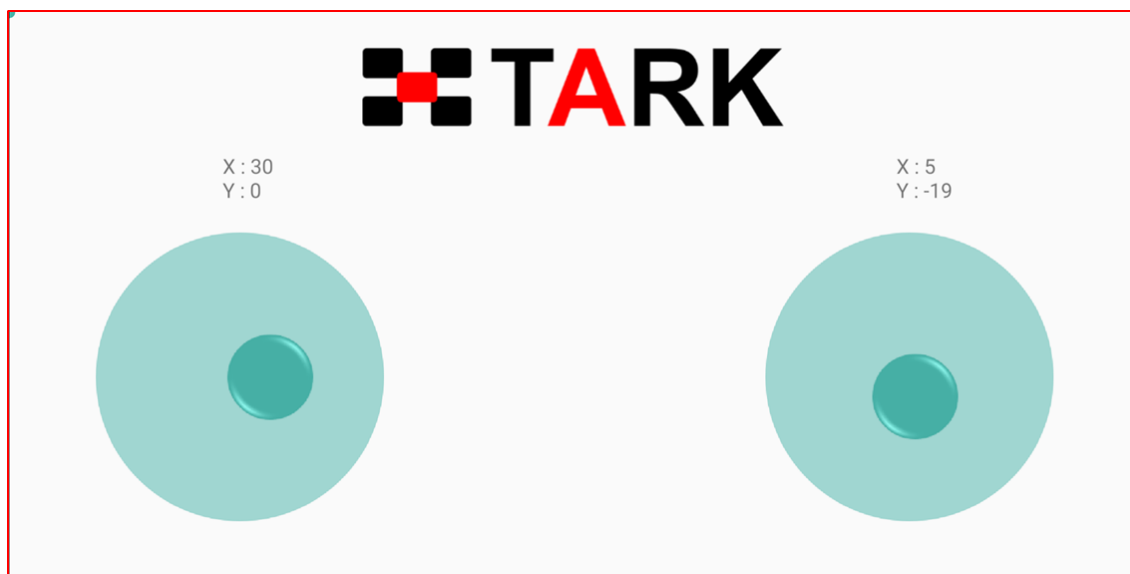


2.2. 功能介绍

软件集成了遥控器和串口助手功能。遥控器支持摇杆模式，按键模式，重力感应模式，坦克模式（待开发）。串口助手功能支持普通串口助手（待开发），协议发送，波形显示（待开发）。



2.3. 摇杆模式



摇杆模式为首界面，界面设计简洁，手动触动摇杆按钮即可通过蓝牙发送摇杆位置数据，随着摇杆滑动，具有震动触觉反馈，操作非常舒服。

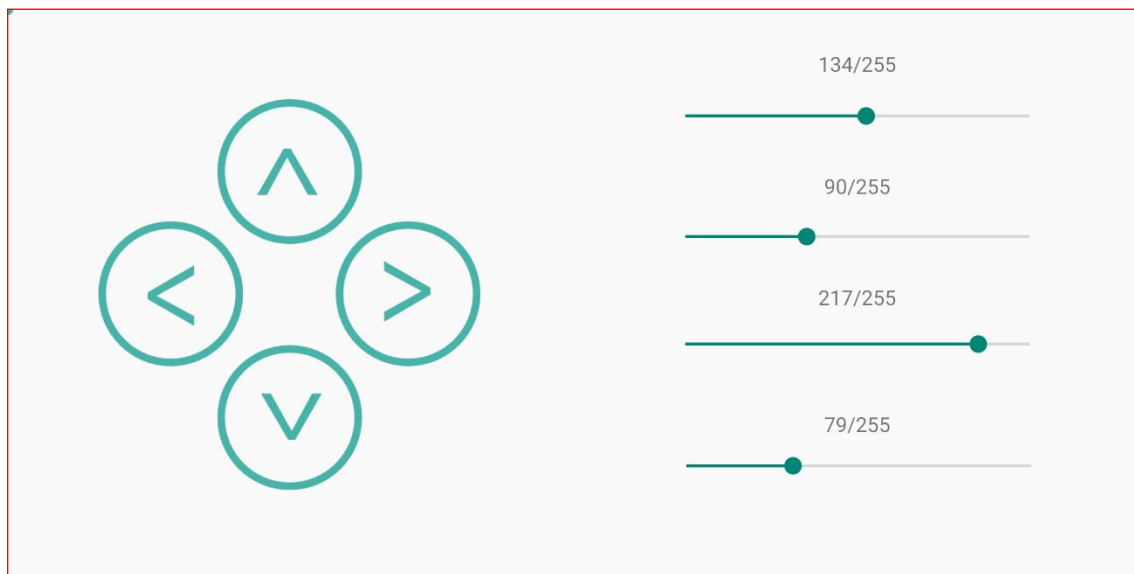
数据发送采用塔克通用 X-Protocol 协议，数据定义如下，数据范围为-100~100。

X-Protocol协议

aa	55	09	31	0d	eb	00	00	31
			ID	左X	左Y	右X	右Y	

备注：摇杆数据范围-100~100

2.4. 按键模式



按键界面，可以操作按钮和滑块，数据发送采用塔克通用 X-Protocol 协议，数据定义如下。

X-Protocol协议

aa 55 0a 32 01 00 00 00 00 3c

ID 按键 滑块1 滑块2 滑块3 滑块4

备注：
键值：上-0x01，下-0x02，左-0x04，右-0x00，
滑块数据范围0~255

2.5. 体感模式



中心小球会随着手机姿态角度变化变动，进入模式后，姿态数据会循环发送。

数据发送采用塔克通用 X-Protocol 协议，数据定义如下。

X-Protocol协议

aa 55 08 33 ef 0b 13 47

ID 航向 横滚 俯仰

备注：
航向范围：-18~18 横滚角度：-90~90 俯仰角：-90~90

2.6. 协议发送

TARK

X-Protocol协议 (变帧长)

AA 55 0B 01 00 01 00 02 00 03 55

帧头 帧长度 帧编码 数据：高位在前，长度可变，自由组合 校验和

注：数据为有符号16进制整数，填写范围-32768~32767

编码		2		数量		8	
数据1	数据2	数据3	数据4	数据5	数据6	数据7	数据8
0	0	0	0	0	0	0	0

发 送

协议发送功能采用 X-Protocol 协议，可进行命令发送和参数调试，例如 PID 参数调试。帧编码范围为 0~255，数量为 1~8，每个数据为有符号 16 进制。

3. X-Protocol 协议

3.1. 协议介绍

为了更好的使用串口进行数据传输，塔克创新设计了一个通用的通信传输协议 X-Protocol 协议，方便塔克创新产品使用，包括软件产品和硬件产品。

X-SOFT通用串口传输协议：X-Protocol协议（变帧长）

AA 55 0B 01 00 01 00 02 00 03 55

帧头 帧长度 帧号 数据：高位在前，长度可变，自由组合 校验和

帧头	双帧头，抗干扰强
帧长度	根据数据长度设定
帧号	用户根据功能设定，标识帧的唯一性
数据	高位在前，长度可变，内容自由组合8位，16位，32位数据
校验和	前面数据累加和的低8位

3.2. 接收参考代码

使用 X-CTR100 控制器的参考接收代码。

```
//接收
static u8 UART_RX_BUF[40]; //接收缓冲，数据内容小于等于32Byte
static u8 UART_RX_CON = 0; //接收计数器
static u8 UART_RX_CHECKSUM; //帧头部分校验和
/**
 * @简 述 串口中断服务程序
 * @参 数 无
 * @返回值 无
 */
void USART1_IRQHandler(void)
{
    uint8_t Res;

    if (USART_GetITStatus(USART1, USART_IT_RXNE) != RESET) //接收中断
    {
        Res = USART_ReceiveData(USART1);

        if (UART_RX_CON < 3) //==接收帧头 + 长度
        {
            if (UART_RX_CON == 0) //接收帧头1 0xAA
            {
                if (Res == 0xAA)
                {
                    UART_RX_BUF[0] = Res;
                    UART_RX_CON = 1;
                }
            }
        }
    }
}
```



```

        else
        {

        }

    }
else if (UART_RX_CON == 1) //接收帧头2 0x55
{
    if (Res == 0x55)
    {
        UART_RX_BUF[1] = Res;
        UART_RX_CON = 2;
    }
    else
    {
        UART_RX_CON = 0;
    }
}
else //接收数据长度
{
    //USART1_RX_LEN = ( Res-1 );
    UART_RX_BUF[2] = Res;
    UART_RX_CON = 3;
    UART_RX_CHECKSUM = 0xFF + Res; //计算校验和, 0xFF为0xAA、0x55校验和
}
}
else //==接收数据
{
    if (UART_RX_CON < (UART_RX_BUF[2] - 1))
    {
        UART_RX_BUF[UART_RX_CON] = Res;
        UART_RX_CON++;
        UART_RX_CHECKSUM = UART_RX_CHECKSUM + Res;
    }
    else //判断最后1位
    {
        //由于没有配套软件, 暂时无需校验
        if (Res == UART_RX_CHECKSUM) //校验正确
        {
            //此处进行数据解析
            UART_Unpack();
        }
        else //校验错误
        {
            //不处理, 输出错误提示

```

```
    }  
  
    //接收完成，恢复初始状态  
    UART_RX_CON = 0;  
}  
}  
    USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);  
}  
}
```

4. 其它说明

暂无。