目录

1. R150 简介

极飞科技 R150 农业无人车（以下简称“无人车”）拥有精简的模块化设计、稳定的运载平台、强大的扩展功能、多样的工作模式，可配载 JetSprayer ™ 气流喷雾系统、XIoT ™ 农业物联系统等智能农业设备，提供精准植保、智能巡田、农资运输等农事服务，在满足不同需求的同时，通过更好的作业效果、更高的作业效率，实现生态、经济、社会效益“三赢”格局，诠释智慧农业的丰富内涵。

R150 无人车主要设备包括有：行走平台系统，人机交互设备，业务设备，配套设备。

A:行走平台系统：车身结构，控制系统，通信系统，电力系统，动力系统

B:人机交互设备：手机 APP，单手遥控器。

C:业务设备：根据不同的业务配套不同的作业设备。

D:配套设备：电池，充电器，发电机，RTK 移动基站等，根据需求选择。

图 1 底盘爆炸图

图 2 结构尺寸图

电器连接如图 3 所示

各模块设备均通过保险接线盒连接，接线盒内部有多路的保险丝，其中小功率模块保险能力为 5A（控制器，声光指示模块等），两路动力的保险分别是 50A，尾部设备保险为 60A。

注意：按下急停开关只切断车辆动力部分电源，车身尾部的设备取电点的电源；对任务设备做任何接线操作时，务必关闭智能电池电源后进行。

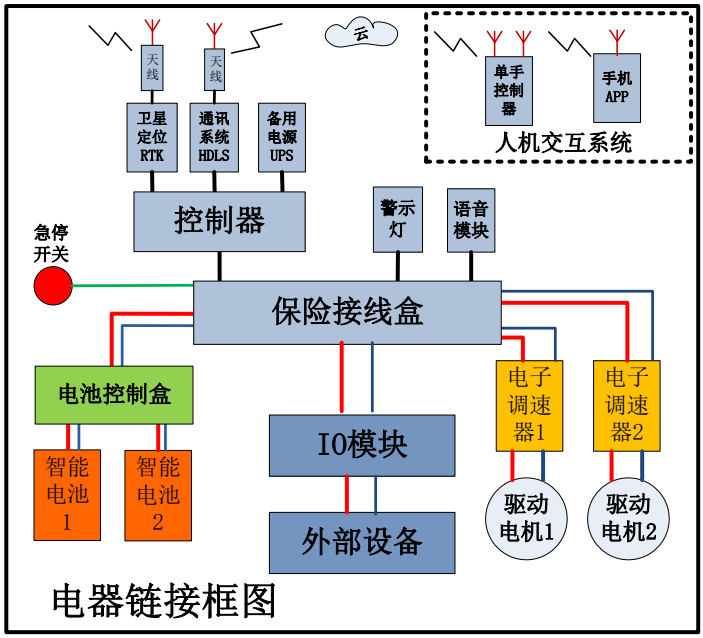

图 3 电器连接框图

2. 使用 ACS2 遥控器

无人车需要配合专业遥控器和 APP 使用，支持多种操控方式，目前支持的操控设备（组合）有：ACS2 车控模块 + 智能手机

无人车配对完成后，可根据以下的按键介绍对无人车进行相应的操控。

a）遥控器开 / 关机：在遥控器关闭 / 开启状态下，长按电源键（ 约 2 秒 ）直至 6 个指示灯同时亮起并闪烁时松手，立刻再次长按电源键（ 约 2 秒 ）直至遥控器开 / 关机。

b)无人车解锁：同时长按遥控器后方两个按键，可解锁无人车，再次短按可上锁。在手动模式下解锁后，长按上方按键可解除驻车，解除驻车后可以自由推动无人车。

c)自主 / 遥控切换：长按中间空白键，切换自主模式。按任意运动控制键则切换成遥控模式。在自主模式时可以通过向 IO 模块发送控制指令来直接控制车辆的两个电机来控制车辆的运动。

d)运动控制：短按向前 / 向后（ / ）按键，控制无人车向前 /向后行驶。短按左转 / 右转（ / ）按键，控制无人车原地左转 / 右转。

3. IO 扩展模块简介

此模块为 R150 扩展版的 IO 模块。将车身尾部的电源与通信 CAN 都接到此模块，即可实现通讯控制功能。该模块的功能有：

a) 1 路 48V 输出，2000W（带输出过流短路保护）

b) 4 路（A、B、C、D）12V 输出，A+B 最高输出 250W，C+D 最高输出 250W。

c) 2 路 5V 输出，共 10W。

d) 4 路 PWM 输入，电压范围 0-15V；L：0 到 0.8V，H：2 到 15V。频率可达 20KHz。

e) 4 路 PWM 输出（开漏输出），电压范围：0.6-24V，频率可达 20KHz。

f)一路 RS232 串口通信接口。

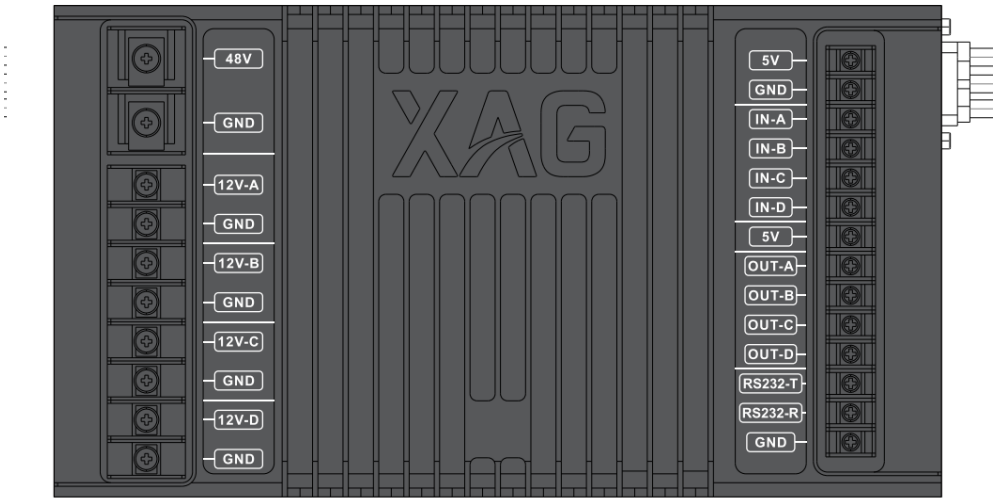

图 4 IO 模块接口示意图

注意事项：

a) 每路 48V、12V 输出端均有各自的 GND 搭配，尽量不要混合使用，避免产生不必要的干扰。

b) 48V，2000W 输出短路保护电流阈值取得比较大，必须小心使用，尽量不要短路。

c) 两路 5V，输出电流不要超过 2A，否则会触发保护。

d) PWM 输入，外设需每路有 1mA 以上的驱动能力保证波形稳定。

e) PWM 输出，是以达林顿开漏形式输出，最低电压为 0.64V（不是 0V）。电压范围：0.6-24V。外设需串联合适限流电阻再接到此输出端，每路灌电流在 10mA 到 300mA 均可。峰值超过 500mA 有损坏风险，注意取合适的限流电阻。

f) 外壳是铝壳，接线时导线不要与外壳接触。

4. IO 扩展模块 RS232 接口传输协议

4.1. 通讯参数

波特率：115200bps，数据位：8，奇偶校验：无，停止位：1，数据流控：无

无特定说明情况下，所有字节大于 1 的数据均采用小端模式。

4.2. 通讯数据格式

帧头		参数长度	命令	参数	CRC
0xAA	0x55	u8	u8	*	u16

帧头：固定为 0xAA 0x55。

参数长度：1 字节，参数的长度，若没有参数，则长度为 0。

命令：1 字节，具体详见通讯命令。

参数：0-255 字节，不同的命令其参数所代表的意义不一样。

CRC:从命令到参数的 CRC 校验，2 字节，详见附录 CRC 算法。

4.3. 通讯命令

在通信系统中，从机向主机发送命令称为**请求**；主机接收到从机的请求后，返回处理结果称为**响应**。每次通信都是“从机请求-主机响应”的方式。

本文中，xx 表示任意字符，也可以表示没有字符。用以下几种数据类型描述各字段：bool、s8、u8、s16、u16、s32、u32。对于 bool 类型，它只有 true 和 false 两个值，本文取其意义为成功和失败。1 表示成功，0 表示失败。下文对 bool 类型的字段如无特别说明，不再赘述。

4.3.1. 电源控制（0x02）

请求：u32 status;//按位设定

响应：bool;

说明：status:

Bit[0]:12V　A 路电源输出状态

Bit[1]:12V　B 路电源输出状态

Bit[2]:12V　C 路电源输出状态

Bit[3]:12V　D 路电源输出状态

Bit[4]:48V　电源输出状态

Bit[5]:5V　电源输出状态

Bit[6:31]:保留位

4.3.2. 四路 PWM 输出控制（0x03）

请求：

u16 pwmDuty[4];//0.1%,0-1000，由低到高依次为 A、B、C、D

响应：bool；

4.3.3. 获取四路 PWM 输出的占空比（0x04）

请求：xx

响应：u16 pwmDuty[4];//0.1%,0-1000，由低到高依次为 A、B、C、D

4.3.4. 获取四路 PWM 输入状态（0x05）

请求：xx

响应：

u32 capPeriod[4]；//捕获 A 输入周期，单位 Hz，范围 10-20000Hz

//由低到高依次为 A、B、C、D

u16 capDuty[4]; //捕获 A 输入占空比，0.1%，0-1000
//由低到高依次为 A、B、C、D

4.3.5. 获取系统信息（0x06）

请求：xx；

响应：

u32 voltage12v[4]; //四路 12v 输出电压，单位 mV，由低到高依次为 A、B、C、D

u32 voltage48v; //主电源电压，单位 mV

s32 current12v[4]; //四路 12v 输出电流，单位 mA，由低到高依次为 A、B、C、D

s32 current48v; //主电源电流，单位 mA

s32 temp12v[2]; //12V 输出温度，单位 0.1 度，由低到高依次为 AB 路温度，CD 路温度

u32 sysStatus; //系统状态，按位设定

u8 powerCtrlMode; //电源控制模式，0:RS232 控制，1:app 控制。

u8 pwmCtrlMode; //pwm 控制模式，0:RS232 控制，1:app 控制

u8 powerStatus; //电源输出状态，按位设定

u8 pwmStatus; //pwm 输出状态，按位设定

u8 reserve[20]; //保留位

说明：sysStatus：

bit[0] ： 12v 电源 AB 路过流

bit[1] ： 12v 电源 CD 路过流

bit[2] ： 48v 主电源过流

bit[3] ： 5v 电源过流

bit[4] ： 12v 电源 AB 路高温

bit[5] ： 12v 电源 CD 路高温

bit[6] ： 电源模块异常

bit[7:31]: 保留

说明：powerStatus：

Bit[0] ： A 路 12V 电源输出状态，0：关闭，1：输出

Bit[1] ： B 路 12V 电源输出状态，0：关闭，1：输出

Bit[2] ： C 路 12V 电源输出状态，0：关闭，1：输出

Bit[3] ： D 路 12V 电源输出状态，0：关闭，1：输出

Bit[4] ： 48V 主电源电源输出状态，0：关闭，1：输出

Bit[5] ： 5V 电源输出状态，0：关闭，1：输出

Bit[6:7]: 保留。

说明：pwmStatus：

Bit[0] ： A 路 pwm 输出状态，0：占空比为 0；1：占空比大于 0

Bit[1] ： B 路 pwm 输出状态，0：占空比为 0；1：占空比大于 0

Bit[2] ： C 路 pwm 输出状态，0：占空比为 0；1：占空比大于 0

Bit[3] ： D 路 pwm 输出状态，0：占空比为 0；1：占空比大于 0

Bit[4:7]: 保留。

4.3.6. 心跳包（0x07）

请求：xx；

响应：bool；

说明：超时 3 秒，其他指令也可作为心跳包

4.3.7. 获取版本号（0xf0）

请求：xx

响应：

u32　HardwareVersion　　//硬件版本号

u32　SoftwareVersion　　//软件版本号

说明：版本号格式：

1byte	1byte	1byte	1byte
主版本号	副版本号	地区号	build号

4.3.8. 设置电机 PWM（0x12）

请求：

s16 rightPWM;　//右侧电机 pwm 值，范围±1000，正值向前，负值向后

s16 leftPWM;　//左侧电机 pwm 值，范围±1000，正值向前，负值向后

响应：

s16 rightMotorSpeed;　//右侧电机转速，单位转/分钟，减速比 28.3

s16 leftMototrSpeed;　//左侧电机转速，单位转/分钟，减速比 28.3

u16 soc;　//剩余电量百分比，单位 0.1%

u8　reserved[2];　//保留位

说明：最快响应频率 20Hz，指令超时时间为 500ms，即 500ms 没有接收到该指令，电机自动停转。当 PWM 最大时，整车速度约为 1.4m/s。

5. 附录

CRC 算法

const static u16 fcsTab[256] = {

0x0000, 0x1189, 0x2312, 0x329b, 0x4624, 0x57ad, 0x6536, 0x74bf,

0x8c48, 0x9dc1, 0xaf5a, 0xbed3, 0xca6c, 0xdbe5, 0xe97e, 0xf8f7,

0x1081, 0x0108, 0x3393, 0x221a, 0x56a5, 0x472c, 0x75b7, 0x643e,

0x9cc9, 0x8d40, 0xbfdb, 0xae52, 0xdaed, 0xcb64, 0xf9ff, 0xe876,

0x2102, 0x308b, 0x0210, 0x1399, 0x6726, 0x76af, 0x4434, 0x55bd,

0xad4a, 0xbcc3, 0x8e58, 0x9fd1, 0xeb6e, 0xfae7, 0xc87c, 0xd9f5,

0x3183, 0x200a, 0x1291, 0x0318, 0x77a7, 0x662e, 0x54b5, 0x453c,

0xbdcb, 0xac42, 0x9ed9, 0x8f50, 0xfbef, 0xea66, 0xd8fd, 0xc974,

0x4204, 0x538d, 0x6116, 0x709f, 0x0420, 0x15a9, 0x2732, 0x36bb,

0xce4c, 0xdfc5, 0xed5e, 0xfcd7, 0x8868, 0x99e1, 0xab7a, 0xbaf3,

0x5285, 0x430c, 0x7197, 0x601e, 0x14a1, 0x0528, 0x37b3, 0x263a,

0xdecd, 0xcf44, 0xfddf, 0xec56, 0x98e9, 0x8960, 0xbbfb, 0xaa72,

0x6306, 0x728f, 0x4014, 0x519d, 0x2522, 0x34ab, 0x0630, 0x17b9,

0xef4e, 0xfec7, 0xcc5c, 0xddd5, 0xa96a, 0xb8e3, 0x8a78, 0x9bf1,

0x7387, 0x620e, 0x5095, 0x411c, 0x35a3, 0x242a, 0x16b1, 0x0738,

0xffcf, 0xee46, 0xdcdd, 0xcd54, 0xb9eb, 0xa862, 0x9af9, 0x8b70,

0x8408, 0x9581, 0xa71a, 0xb693, 0xc22c, 0xd3a5, 0xe13e, 0xf0b7,

0x0840, 0x19c9, 0x2b52, 0x3adb, 0x4e64, 0x5fed, 0x6d76, 0x7cff,

0x9489, 0x8500, 0xb79b, 0xa612, 0xd2ad, 0xc324, 0xf1bf, 0xe036,

0x18c1, 0x0948, 0x3bd3, 0x2a5a, 0x5ee5, 0x4f6c, 0x7df7, 0x6c7e,

0xa50a, 0xb483, 0x8618, 0x9791, 0xe32e, 0xf2a7, 0xc03c, 0xd1b5,

```
0x2942, 0x38cb, 0x0a50, 0x1bd9, 0x6f66, 0x7eef, 0x4c74, 0x5dfd,
0xb58b, 0xa402, 0x9699, 0x8710, 0xf3af, 0xe226, 0xd0bd, 0xc134,
0x39c3, 0x284a, 0x1ad1, 0x0b58, 0x7fe7, 0x6e6e, 0x5cf5, 0x4d7c,
0xc60c, 0xd785, 0xe51e, 0xf497, 0x8028, 0x91a1, 0xa33a, 0xb2b3,
0x4a44, 0x5bcd, 0x6956, 0x78df, 0x0c60, 0x1de9, 0x2f72, 0x3efb,
0xd68d, 0xc704, 0xf59f, 0xe416, 0x90a9, 0x8120, 0xb3bb, 0xa232,
0x5ac5, 0x4b4c, 0x79d7, 0x685e, 0x1ce1, 0x0d68, 0x3ff3, 0x2e7a,
0xe70e, 0xf687, 0xc41c, 0xd595, 0xa12a, 0xb0a3, 0x8238, 0x93b1,
0x6b46, 0x7acf, 0x4854, 0x59dd, 0x2d62, 0x3ceb, 0x0e70, 0x1ff9,
0xf78f, 0xe606, 0xd49d, 0xc514, 0xb1ab, 0xa022, 0x92b9, 0x8330,
0x7bc7, 0x6a4e, 0x58d5, 0x495c, 0x3de3, 0x2c6a, 0x1ef1, 0x0f78
};

u16 fcs_CRC16(u8 *cp, u16 len)
{
u16 fcs = 0xffff;
    while (len--)
    {
        fcs = (fcs >> 8) ^ fcsTab[(fcs ^ *cp++) & 0xff];
    }
return (fcs^0xffff);
}
```