

轻装版四驱小车 V2 开发手册

1. 简单入门与 APP 在线调参
2. CAN 控制
3. CAN 采集数据
4. 串口控制
5. 串口采集数据与 OLED
6. 运动学分析
7. 注意事项
8. 如何给小车下载程序

1. 简单入门与 APP 调参

收到小车后，在确认外观没有受损的情况下，就可以测试一下了。首先插上电池红色接头，小车不设开关的。此时，红色的电源指示灯亮起，系统供电正常。与此同时，蓝灯也开始闪烁，代表着单片机已经正常启动了。资料里面有测试视频教程。小车需要用户旋转右轮选择 APP、PS2、线性 CCD、ELE 电磁巡线等控制方式，并单击用户按键确认选择，超过 8 秒不做任何操作，系统会自动确认选择，其中 APP 控制方式会自动使能电机，另外 3 种控制方式需要进入该模式之后单击用户按键使能电机。

注：为保证测试效果，请测试之前务必清除小车轮胎和地面的灰尘。

接下来请把最新版的 MiniBalance APP 安装到安卓手机上，然后根据相应的视频教程操作即可遥控小车或者进行参数在线调节等。具体每个通道的名称可以根据下图设置，点击右上方的菜单可以自定义。



另外，在调节 PID 参数之前，我们需要点击【获取设备参数】(右上角菜单按钮调出)，把小车的 PID 参数更新到 APP 上面，然后拖动滑块，当我们松手的时候，APP 即可把参数发送到小车上面。

2.CAN 控制

CAN (Controller Area Network)即控制器局域网，因为具有高性能、高可靠性以及独特的设计而越来越受到关注, 现已形成国际标准，被公认为几种最有前途的现场总线之一。小车提供 CAN 接口，可以通过 CAN 总线输入控制指令，下面是通信协议说明。

2.1 CAN 使能和失能

默认小车接收蓝牙指令，如果需要让小车接收 CAN 指令，需要先向 CAN 口发送使能指令。波特率是 1M。

使能指令格式如下：

标识符 ID:0x121

帧类型：标准帧

帧格式：DATA

DLC:8 个字节

数据域	tx[0]	tx[1]	tx[2]	tx[3]	tx[4]	tx[5]	tx[6]	tx[7]
内容	10	12	15	19	24	30	37	Flag

Flag=1 时，CAN 控制使能，之后主板不再接收蓝牙模块的指令。此时，我们可以发送 CAN 指令对小车进行控制。如果在使能 CAN 控制之后要失能，也可发送上面的指令，但是 Flag=0；

2.2 CAN 控制指令说明

标识符 ID:0x121

帧类型：标准帧

帧格式：DATA

DLC:8 个字节

数据域	tx[0]	tx[1]	tx[2]	tx[3]	tx[4]	tx[5]	tx[6]	tx[7]
内容	控制量	控制量	控制量	控制量	预留	预留	预留	预留

具体到下面是：

数据域	tx[0]	tx[1]	tx[2]	tx[3]	tx[4]	tx[5]	tx[6]	tx[7]
内容	前后速度大小控制	前后方向控制	转向速度大小控制	转向方向控制	预留	预留	预留	预留

其中 tx[0]控制前后运动速度的大小，tx[1]控制前后运动的方向，tx[2]控制转向运动的速度大小，tx[3]控制转向运动的方向。可以参考下面的例程：

```
txbuf[0]=32;//设置前后速度的大小
```

```
txbuf[1]=1;//设置前后运动的方向 0 是正，也就是前进，其他是负。
```

```
txbuf[2]=40;//设置转向速度的大小
```

```
txbuf[3]=0;设置转向运动的方向 0 是逆时针，其他是负。
```

```
txbuf[4]=0;
```

```
txbuf[5]=0;
txbuf[6]=0;
txbuf[7]=0;
CAN1_SEND(0X121, txbuf); //CAN 发送
```

速度的单位和后面串口控制指令的单位是一样的, 在串口控制章节统一解释尺寸。

3.CAN 数据采集

小车在接受 CAN 控制指令的同时, 也可发送小车自身的传感器数据。因为数据比较多, 所以分 2 次传输, 通过标识符区分。

第一次数据传输如下:

标识符 ID:0x100

帧类型: 标准帧

帧格式: DATA

DLC:8 个字节

数据域	rx[0]	rx[1]	rx[2]	rx[3]	rx[4]	rx[5]	rx[6]	rx[7]
内容	X 轴角 度高 8 位	X 轴角 度低 8 位	Y 轴角 度高 8 位	Y 轴角 度低 8 位	Z 轴角 度高 8 位	Z 轴角 度低 8 位	Z 轴角 速度高 8 位	Z 轴角 速度低 8 位

接收之后需要对数据进行解包:

```
Pitch= (float)((rx[0]*256+rx[1])-30000)/100; //单位是°
```

```
Roll = (float)((rx[2]*256+rx[3])-30000)/100; //单位是°
```

```
Yaw = (float)((rx[4]*256+rx[5])-30000)/100; //单位是°
```

```
GZ= rx[6]*256+rx[7]-32768; //MPU6050 原始角速度数据
```

第二次数据传输如下:

标识符 ID:0x101

帧类型: 标准帧

帧格式: DATA

DLC:8 个字节

数据域	rx[0]	rx[1]	rx[2]	rx[3]	rx[4]	rx[5]	rx[6]	rx[7]
内容	电机 A 速度大小	电机 A 速度方向	电机 B 速度大小	电机 B 速度方向	电机 C 速度大小	电机 C 速度方向	电机 D 速度大小	电机 D 速度方向

其中 rx[1]、rx[3]、rx[5] 代表方向，0 代表正，1 代表 0，2 代表负。

接收之后需要对数据进行解包：

$\text{Encoder_A} = (1 - \text{rx}[1]) * \text{rx}[0]$; // 每 10ms 采集得到的脉冲计数

$\text{Encoder_B} = (1 - \text{rx}[3]) * \text{rx}[2]$; // 每 10ms 采集得到的脉冲计数

$\text{Encoder_C} = (1 - \text{rx}[5]) * \text{rx}[4]$; // 每 10ms 采集得到的脉冲计数

$\text{Encoder_D} = (1 - \text{rx}[7]) * \text{rx}[6]$; // 每 10ms 采集得到的脉冲计数

发送的数据不包括电机位置信息，如果需要使用到位置信息，可以在接收数据之后，单位时间内对速度进行积分得到位置。

4. 串口控制

串行接口（Serial Interface）是指数据一位一位地顺序传送，其特点是通信线路简单，只要一对传输线就可以实现双向通信。小车可以通过串口控制，这大大降低了大家的使用难度，下面是通信协议说明。

4.1 串口控制使能

首先需要说明的是，小车上提供了 3 个串口，串口 2 用于连接蓝牙了，波特率 9600。串口 3 可以发送里程计或者接收控制指令，波特率是 115200。当然。其他控制器向小车发送数据控制小车的频率原则上没有限制，建议不大于 200hz。串口 3（控制器上面引出的 4pin 排针，有丝印“USART3”）是 TTL 电平的，我们可以通过杜邦线连接到另外的控制器，比如 STM32，Arduino 等进行通信。串口 5 是预留没使用的（PD2 PC12）。

APP 模式下，默认小车接收蓝牙遥控指令，如果要其他的设备通过串口对小车进行控制，需要先使能串口控制。为了降低使用难度，串口使能机制比较简单，小车只要超过 10 次进入串口接收中断即可使能串口接收标志位，换一种理解就是：向小车发送 10 个任意字节即可使能串口控制。此时，我们就可以使用其他的单片机，比如 Arduino，通过串口发送指令对小车进行控制了。如何判断串口已经使能了呢？使用 APP 遥控一下，如果小车不再接收 APP 的遥控指令，即可说明串口已经使能。需要说明的是串口（包括 CAN）接收数据都是不需要使能的，上电进入 APP 模式就可以接收小车的数

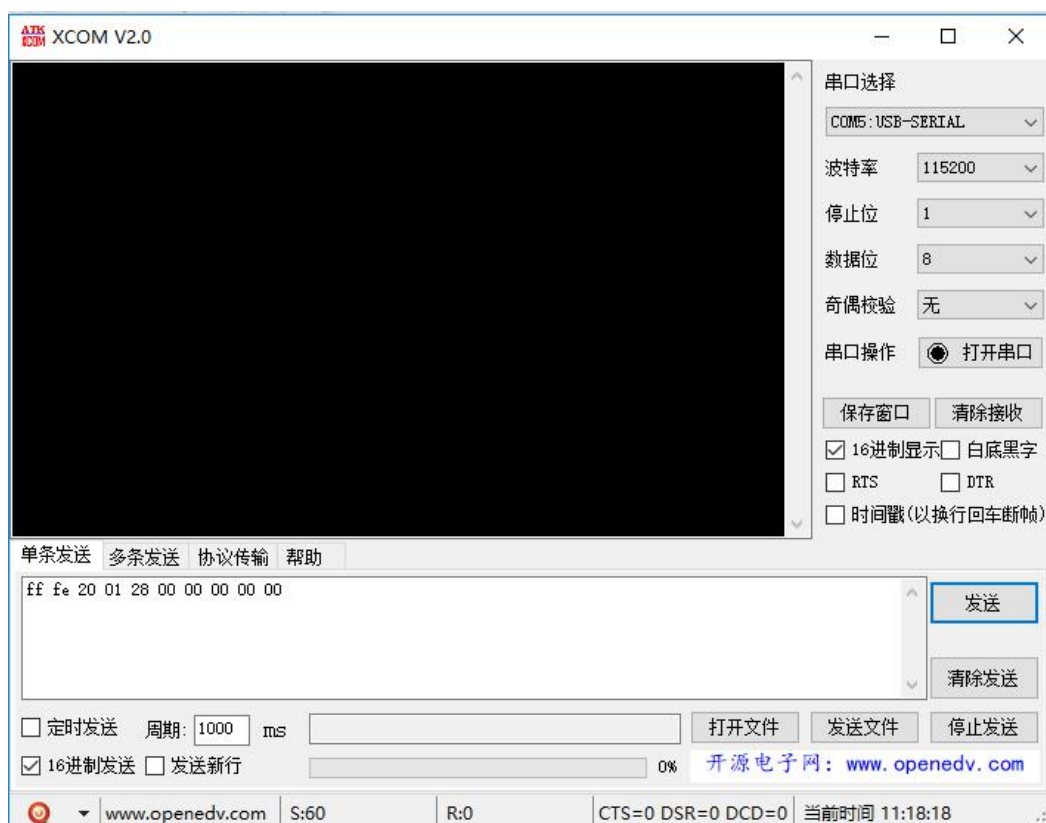
4.2 串口控制指令说明

数 据 域	帧头	帧头	tx[0]	tx[1]	tx[2]	tx[3]	tx4]	tx[5]	tx[6]	tx[7]
内容	0xff	0xfe	控制量	控制量	控制量	控制量	预留	预留	预留	预留

串口控制的指令和 CAN 控制完全一致，具体请参考 2.2 节。

4.3 例程说明

我们可以通过 MicroUSB 数据线把小车连接到电脑。打开我们常用的串口调试助手。可以很方便的使用串口调试助手测试，如下图：



这样相当于控制小车以 0x20 (32) 的速度向后运动，同时以 0x28 (40) 的速度逆时针转向。下面解释一下这个 32 的单位（其他的同理分析），10ms 转 32 个脉冲，1s 转 3200 个脉冲，车轮转一圈，输出(编码器线数 13*减速比 30*4 倍频)=1560。也就是小车以每秒转 3200/1560 圈的速度运行，再结合轮胎的直径信息，就可以得到小车的运行速度，转向计算同理，得到的是线速度。

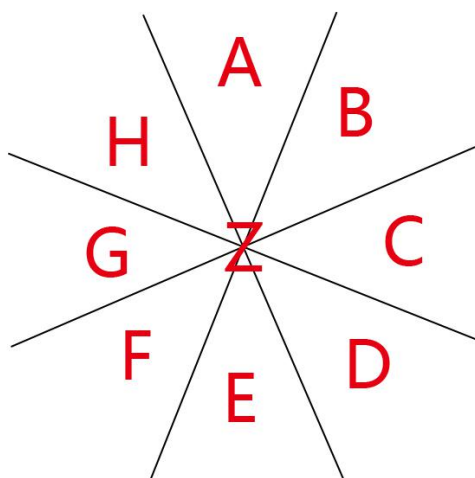
另外也可通过 TTL 电平的串口 3 控制小车。比如我们使用 STM32 控制器，发送 10 个字节就可以构成一个完整的控制指令：

```
Usart_Send(0xff);
Usart_Send(0xfe);
Usart_Send(0x20);
Usart_Send(0x01);
Usart_Send(0x28);
Usart_Send(0x00);
Usart_Send(0x00);
Usart_Send(0x00);
Usart_Send(0x00);
Usart_Send(0x00);
```

这个指令实现的控制效果和刚才使用电脑发送的指令一致。需要注意的是，发送的数据结构是 10 个字节，不是字符串。

4.4 蓝牙串口控制协议简单说明

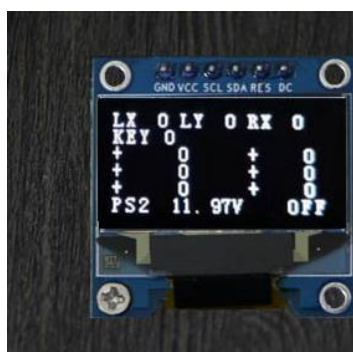
通过蓝牙可以进行简单的遥控控制，是通过单字符指令控制的，A~H 个字符对应不同的运动方位，比如 A 对应前进，E 对应后退，Z 对应停止，根据 ASCII 码，发字节形式的指令也可以，A 对应的是 0X41。



5. 串口采集数据与 OLED

小车默认是通过 Android 手机的 APP 查看数据和进行遥控的，使用我们的最新的 MiniBalance APP 即可，OLED 显示屏也可以显示小车的信息，如下图所示：





PS2模式下OLED显示内容：

第一行：PS2方向和速度控制指令

第二行：PS2手柄的键值

第三行：B电机目标速度和测量速度

第四行：C电机目标速度和测量速度

第五行：D电机目标速度和测量速度

第六行：模式、电池电压，电机状态



CCD模式下OLED显示内容：

第一行：CCD图像显示

第二行：CCD的中值和阈值

第三行：B电机目标速度和测量速度

第四行：C电机目标速度和测量速度

第五行：D电机目标速度和测量速度

第六行：模式、电池电压，电机状态



ELE模式下OLED显示内容：

第一行：左中右三个电感的数据

第二行：提取的算法中值

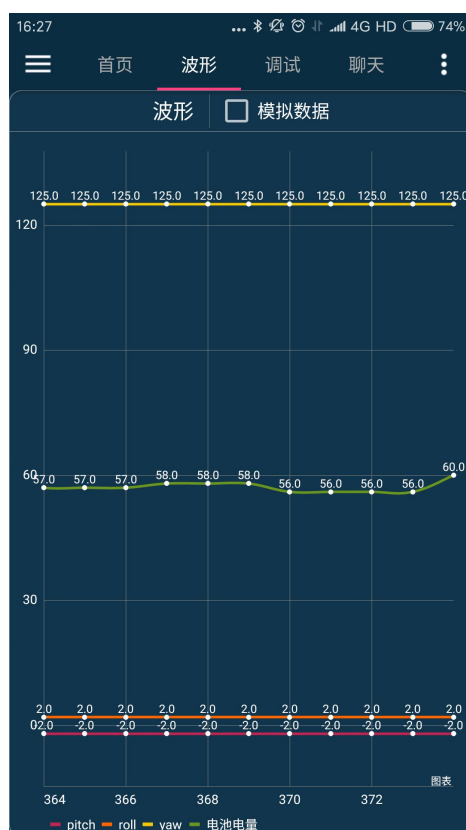
第三行：B电机目标速度和测量速度

第四行：C电机目标速度和测量速度

第五行：D电机目标速度和测量速度

第六行：模式、电池电压，电机状态

使用 APP 也可监控小车的参数，在波形界面，会通过波形显示小车的 3 轴姿态和电池电量（百分比）。



我们根据上图可以对 APP 每个通道的名称重命名，以便更好的区分每个通道的数据。点击右上方的菜单按钮重命名。

另外，通过小车上面的串口 3 采集 IMU 和里程计。一帧数据有 12 个字节，其中 0xff、0xfe 是帧头，详细说明如下：

数据域	帧头	帧头	tx[0]	tx[1]	tx[2]	tx[3]	tx[4]	tx[5]	tx[6]	tx[7]	tx[8]	tx[9]
内容	0xf	0xfe	A 电机速度大小	A 电机速度方向	B 电机速度大小	B 电机速度方向	C 电机速度大小	C 电机速度方向	D 电机速度大小	D 电机速度方向	Z 轴陀螺仪高 8 位	Z 轴陀螺仪低 8 位

速度信息很容易理解，单位的解释参考控制部分。方向信息的解释：0 是正，2 是负，1 就代表数据为 0。其中 Z 轴陀螺仪的数据是+32768 再发过来的，所以，处理之后的真实 Z 轴陀螺仪数据应该是：

$$\text{GyroZ} = \text{tx}[6] * 256 + \text{tx}[7] - 32768$$

6.运动学分析

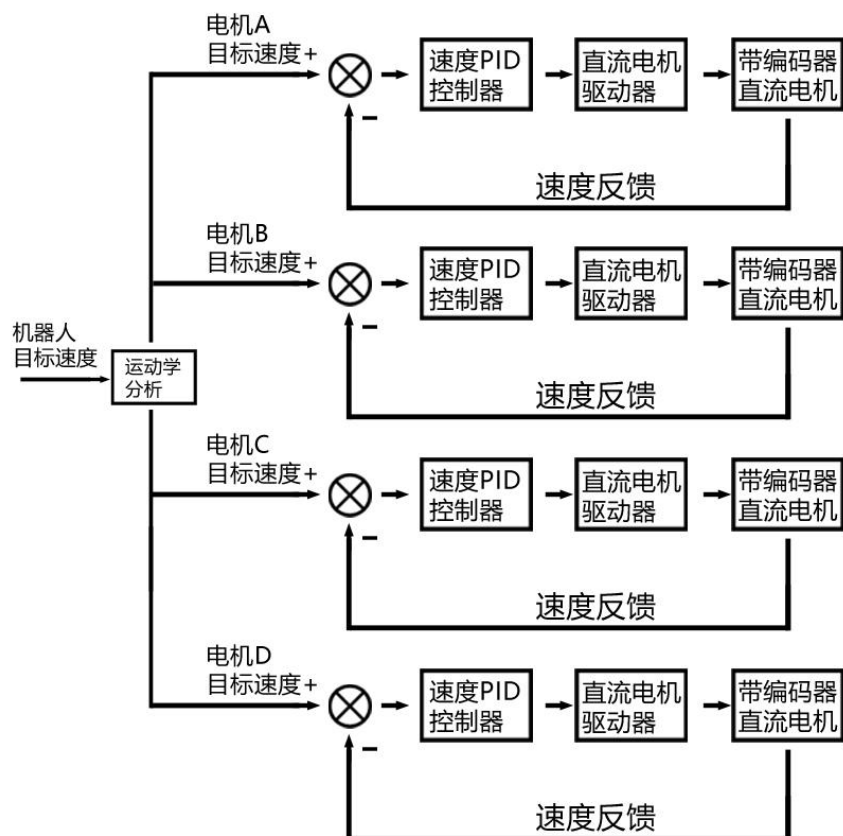
我们可以容易使用 C 语言编程。

```
#define a_PARAMETER          (0.079f)

void Kinematic_Analysis(float Vy,float Vz)
{
    Target_A   = Vy-Vz*(a_PARAMETER);
    Target_B   = Vy-Vz*(a_PARAMETER);
    Target_C   = Vy+Vz*(a_PARAMETER);
    Target_D   = Vy+Vz*(a_PARAMETER);
}
```

入口参数为 X Y Z 轴速度，解算出四个电机的目标转速，送入 PID 控制器进行控制。

以下分别是速度控制的原理图。



四驱小车速度控制原理图

7.注意事项

1. 轮胎或者地面上面的灰尘会减小摩擦力，测试或者演示之前，建议清除轮胎和地面的灰尘，轮胎上面使用半湿的毛巾擦拭即可

2. 务必时刻注意电池电压，建议低于 11.5V 就开始充电，锂电池过放必然会导致电池永久损坏。显示屏有显示电池电压。

3. 通过 CAN 或者串口控制的时候，在加减速的时候建议使用平滑变化的曲线，以减小小车急停或者高速启动带来的滑动摩擦，需要知道的是，编码器对滑动摩擦是无法感知的。

4. 主板上面的 5V 和 3.3V 引脚可以对外供电，但是不可带太大电流的负载，其中 5V 输出不建议带超过 1.0A 的负载，3.3V 输出不建议带超过 200mA 的负载。

5. 使用的过程中要避免电机堵转，否则很容易烧毁主板。

8. 如何给小车下载程序

小车可以通过 SWD 接口下载程序，在主板上上面有标识，分别是 PA13 和 PA14。

1. 硬件准备

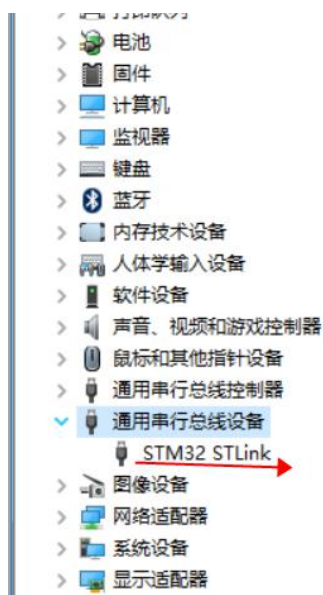
1. 小车

2. STlink(Jlink 也可)

2. 软件准备

对应的 STlink 或者 Jlink 驱动的安装。

安装成功后可以打开设备管理器看看



可以看到驱动已经安装成功！

3. 接线

STlink -----小车

SWDIO-----PA13

SWCLK-----PA14

GND-----GND

4. OK, 一切准备就绪，确保小车已经上电了。

点击下图箭头所指的按钮，程序就可以下载了！因为勾选了编程后执行，所以程序下载完后，会自动运行。默认程序的配置是针对 STlink 的，如需配置 Jlink 或者其他的调试器下载，需要修改 MDK 设置。

