

轮 趣 科 技

MiniBalanceV3.5 上位机 使用说明

推荐关注我们的公众号获取更新资料



版本说明:

版本	日期	内容说明
V1.0	2021/10/28	第一次发布

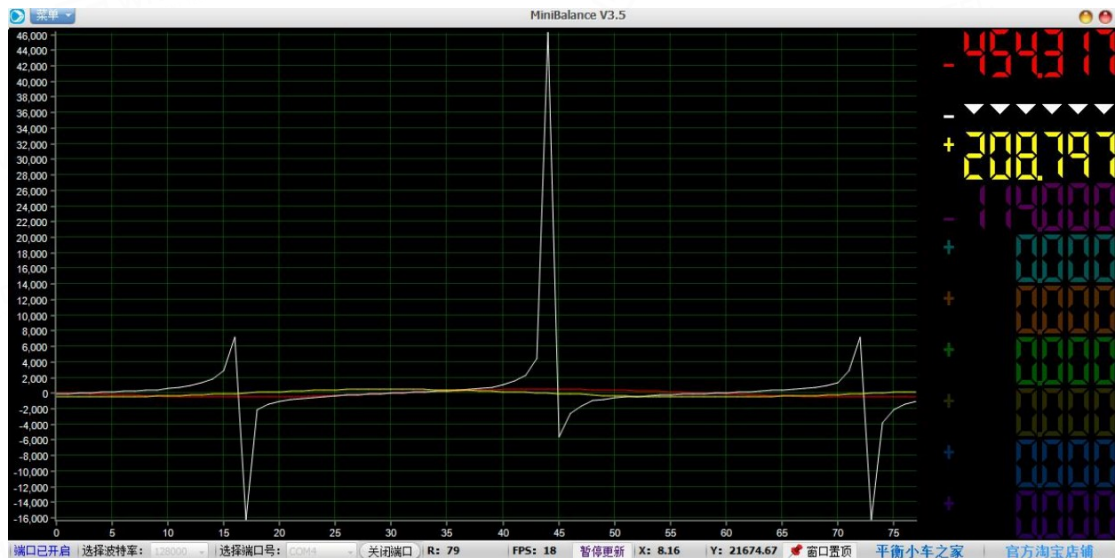
网址: www.wheeltec.net

目录

1. 简单使用.....	3
2. 注意事项.....	3
3. 拓展功能.....	4
4. 移植教程.....	5

1. 简单使用

MiniBalance 为免安装版，直接右键以管理员身份运行即可（如果打开失败，重启一次电脑就行）。使用资料包里面的测试例程下载到 STM32F103C8 或者 Arduino 里面，然后运行上位机，波特率 128000；做如下设置即可显示波形：



（购买套件的同学无需重新下载代码，可直接显示波形）

2. 注意事项

与上位机通信的数据帧长度固定为 $4*N+1$ ，帧数据字节之间传输延时不可超过 $1ms$ ，否则将认为当前帧结束。UI 刷新没有限制显示延时。这也意味着 UI 刷新速率直接取决于 PC 的图形、计算性能以及通讯链的可靠。如若出现通讯不稳定等等因素导致漏、错、字节或延迟、丢帧的现象，都将无法通过上位机的协议解析。则在图表中当前数据无法显示，即可能呈现出线条不完整，数据缺失等现象。

3. 拓展功能

不选择打开扩展功能子功能时，虚拟示波器主界面为默认更新

10 个通道的数值

扩展功能总共有 4 个，分别为：

- 1、3D 立方体 （观察物体立体姿态）
- 2、地平仪 （观察物体姿态角）
- 3、3D 图表 （查看物体 3 维运动轨迹）
- 4、通道独立显示 （观察数据直方图及各通道独立图表显示）

扩展功能的数据传递是复用了部分通道数据，具体为：

一、3D 立方体（此时通道 1、2、3 用来传递立方体的姿态数据）

立方体俯仰角（Pitch） 映射至 CH1

立方体横滚角（Roll） 映射至 CH2

立方体航向角（Yaw） 映射至 CH3

取值范围：

Pitch: ± 90

Roll : ± 180

Yaw : 0-360

二、地平仪（此时通道 1、2、3 用来传递立方体的姿态数据）

Pitch -> CH1

Roll -> CH2

Yaw -> CH3

取值范围:

Pitch: ± 90

Roll : ± 180

Yaw : 0-360

三、3D 图表（此时通道 1、2、3 用来传递 3 维空间内点的坐标数据）

X -> CH1

Y -> CH2

Z -> CH3

四、通道独立显示

顾名思义，将各个通道数据分别显示在独立的图表里

4. 移植教程

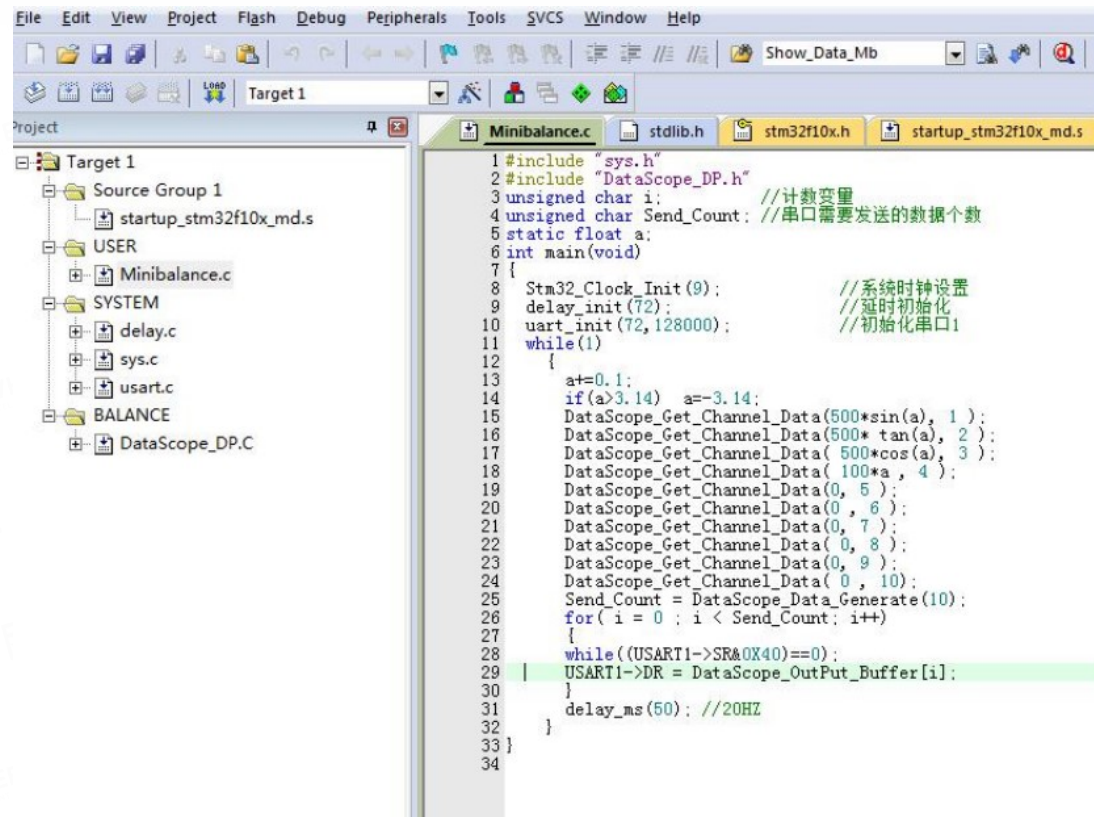
我们提供的例程是 STM32F103 和 Arduino 的，如果使用其他单片机，可以很容易移植过去的。首先把资料包中的上位机库文件 `#include` 到工程里面并在主函数添加如下代码:

(其中红色代码替换为您的单片机的底层代码即可)

```
#include "sys.h"
#include "DataScope_DP.h"
unsigned char i;    //计数变量
unsigned char Send_Count; //串口需要发送的数据个数
static float a;
int main(void)
{
```

```
Stm32_Clock_Init(9); //系统时钟设置
delay_init(72); //延时初始化
uart_init(72,128000); //初始化串口 1
while(1)
{
    a+=0.1;
    if(a>3.14) a=-3.14;
    DataScope_Get_Channel_Data(500*sin(a), 1 );
    DataScope_Get_Channel_Data(500* tan(a), 2 );
    DataScope_Get_Channel_Data( 500*cos(a), 3 );
    DataScope_Get_Channel_Data( 100*a , 4 );
    DataScope_Get_Channel_Data(0, 5 );
    DataScope_Get_Channel_Data(0 , 6 );
    DataScope_Get_Channel_Data(0, 7 );
    DataScope_Get_Channel_Data( 0, 8 );
    DataScope_Get_Channel_Data(0, 9 );
    DataScope_Get_Channel_Data( 0 , 10);
    Send_Count = DataScope_Data_Generate(10);
    for( i = 0 ; i < Send_Count; i++)
    {
        while((USART1->SR&0X40)==0);
        USART1->DR = DataScope_OutPut_Buffer[i];
    }
    delay_ms(50); //20HZ
}
}
```

下面是完整的 STM32 工程截图



```
1#include "sys.h"
2#include "DataScope_DP.h"
3unsigned char i; //计数变量
4unsigned char Send_Count; //串口需要发送的数据个数
5static float a;
6int main(void)
7{
8    Stm32_Clock_Init(9); //系统时钟设置
9    delay_init(72); //延时初始化
10    uart_init(72,128000); //初始化串口1
11    while(1)
12    {
13        a+=0.1;
14        if(a>3.14) a=-3.14;
15        DataScope_Get_Channel_Data(500*sin(a), 1);
16        DataScope_Get_Channel_Data(500* tan(a), 2);
17        DataScope_Get_Channel_Data( 500*cos(a), 3);
18        DataScope_Get_Channel_Data( 100*a , 4);
19        DataScope_Get_Channel_Data(0, 5);
20        DataScope_Get_Channel_Data(0, 6);
21        DataScope_Get_Channel_Data(0, 7);
22        DataScope_Get_Channel_Data( 0, 8);
23        DataScope_Get_Channel_Data(0, 9);
24        DataScope_Get_Channel_Data( 0, 10);
25        Send_Count = DataScope_Data_Generate(10);
26        for( i = 0 ; i < Send_Count; i++)
27        {
28            while((USART1->SR&0X40)==0);
29            USART1->DR = DataScope_OutPut_Buffer[i];
30        }
31        delay_ms(50); //20HZ
32    }
33}
34
```

如果移植成功，可以出现本文页首的波形。