学会助手 APP

使用手册

版本 1.4.1

2023/05/16

目录

一、连接设备

支持蓝牙模块（HC-05、HC-06、HC-08、BLE、BT04 等）与 WiFi 模块（ESP8266、ESP32 等）。

- 推荐使用经典蓝牙（蓝牙 2.0 如 HC-05、HC-06、BT04 等）
- 低功耗蓝牙高速传输数据可能导致数据丢失（如 115200bit）

1.1 连接

- 蓝牙：工具->蓝牙->单击连接
- Wifi：打开手机 wifi 连接到模块热点，工具->wifi->右上角->添加 wifi->输入 IP 地址、端口号，确定->单击连接
- Wifi 长按删除设备

- 双模蓝牙优先连接到经典蓝牙模式
- Wifi 只能用作 TCP 客户端

1.2 自动连接

- 软件打开后首次连接需手动
- 蓝牙模块与 wifi 模块连接独立运行
- 进入连接界面自动连接将暂停，期间若无设备成功连接，退出时，自动连接继续
- 程序被后台杀死，自动连接停止

1.3 断开连接

- 单击已连接设备，选择断开连接。

二、串口

图 2.1 串口

- 接收区接收连续大量数据会导致卡顿甚至 APP 卡死，仅用于功能测试与少量数据的接收。
- 使用经典蓝牙时，由于中文由多个字节组成，且分包发送，可能出现乱码。
- 在设置中选择字符编码格式，编码格式对整个 APP 均有效

三、图像上位机

支持彩色图像（RGB565）、灰度图像（256）与二值化图像（一个字节 8 个像素）。

3.1 窗口

图 3.1 图像上位机窗口

- 图片宽高：不超过 999*999，太小可能导致丢帧。
- 点击阈值修改，阈值在 L 于 H 之间显示为白，不符合为黑
- 保存图片：彩色图片格式*.color 灰度图片格式*.grey 二值化图片格式*.bin
- 彩色图像格式：低八位在前，高八位在后（先发送低八位，后发送高八位）
- 灰度图像亮度取反：关闭时值越大越亮，开启时值越小越亮
- 二值化图像字节顺序：正序为高位在前，倒序为低位在前
- 单击小窗口切换视图
- 点击游标两端拖动游标

3.2 下位机上传图像程序

➢ 数据格式：0x01 0x09 0x08 0x07 ...图像数据...

```
/*************************************************************
 *函数名称：XHZS_SendImage
 *简　　介：学会助手 APP 图像下位机程序
 *输　　入：imgaddr:数组起始地址 imgsize:图像对应数组大小
 *输　　出：无
 *注意事项：无
 *************************************************************/
void XHZS_SendImage(uint8_t *imgaddr, uint32_t imgsize)
{
    uint8_t cmdf[4] = {0x01, 0x09,0x08, 0x07};    //帧头

    Uart_SendArray(USART1,cmdf,sizeof(cmdf));   //发送帧头
    Uart_SendArray(USART1,imgaddr,imgsize);     //发送数据
}
```

四、虚拟示波器

4.1 窗口

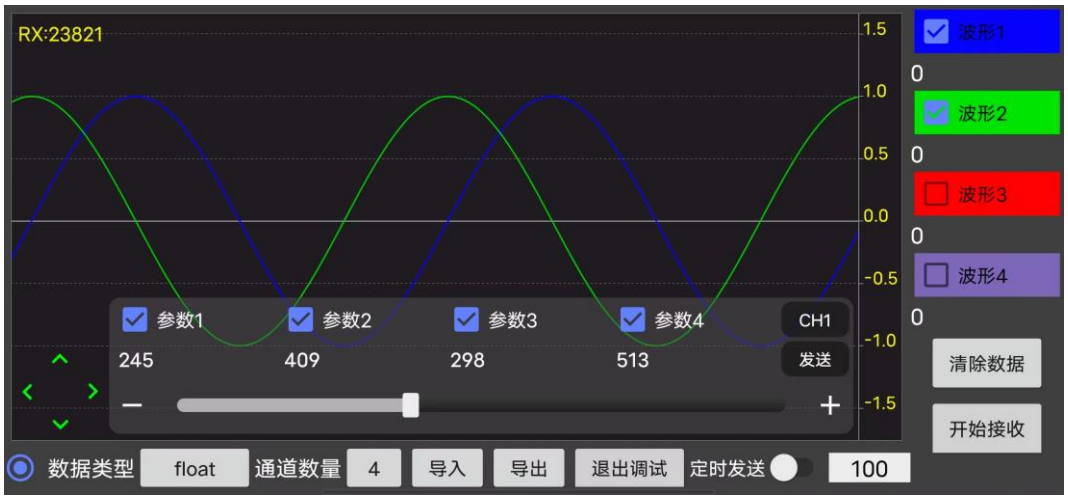

图 4.1 虚拟示波器窗口

- 单击波形名称设置波形是否可见，长按修改波形名称
- 点击”CH1”选择参数组，点击发送发送当前参数组
- 单击发送发送当前参数组到下位机
- 单击参数名称选择是否加入发送队列，长按修改参数相关参数
- 开启定时发送，以设置时间为单位定时发送当前参数组
- 未开启定时发送时，点击定时时间可修改时长
- 保存文件后缀*.wave
- 最多仅可接收 300 万数据
- 手势无法缩放时，尝试使用左下角方向按键缩放

4.2 APP 下发参数格式

- APP 下发参数格式：
  参数 1+参数 2+参数 3+参数 4+”\r\n”；参数 x=”c”+参数所在组+x+”:”
- 未勾选参数不发送
- 例：参数组选择 CH1，参数 1 为 0，参数 2 不勾选，参数 3 为 3.1，参数 4 为 3.14
  下发数据：c11:0,c13:3.1,c14:3.14,\r\n（字符串，注意手动补零）
- 可在下位机例程中参考数据提取方法。

4.3 下位机上传波形程序

4.3.1 使用字符串发送

- 使用方法：点击数据格式选择字符串
- 下位机上传字符串格式：”w:”+数据 1+’,’+数据 2+’,’+数据 3+’,’+数据 4+”\r\n”，数据数量与上位机通道数量对应
  例：printf("w:1987,37,3.141\r\n",num);为发送三个数据 1987,73,3.141 到上位机

4.3.2 使用发送函数

➢ 数据格式：0x03 0xFC 通道 1 通道 2... 0xFC 0x03

```
/*************************************************************
 *函数名称：XHZS_SendWave
 *简    介：学会助手 APP 虚拟示波器下位机程序（兼容山外）
 *输    入：wareaddr:数组地址  waresize:数组大小
 *输    出：无
 *注意事项：无
 *************************************************************/
void XHZS_SendWave(uint8_t *waveaddr, uint16_t wavesize)
{
    uint8_t cmdf[2] = {0x03, 0xFC};    //帧头
    uint8_t cmdr[2] = {0xFC, 0x03};    //帧尾
    Uart_SendArray(USART1,cmdf, sizeof(cmdf));     //发送帧头
    Uart_SendArray(USART1,waveaddr, wavesize);    //发送数据
    Uart_SendArray(USART1,cmdr, sizeof(cmdr));     //发送帧尾
}
/*************************************************************
 *函数名称：SendWave
 *简    介：发送数据到学会助手 APP 虚拟示波器
 *输    入：无
 *输    出：无
 *注意事项：将要发送的数据赋值给 Wave 数组即可
 *************************************************************/
void SendWave()
{
    uint32_t Wave[4]={0};    //注意修改数据类型；数组大小即为通道数量，最多四条通道
    //需要发送的变量赋值给 Wave 数组即可
    XHZS_SendWave((uint8_t *)Wave,sizeof(Wave));
}
```

五、UI

5.1 控件简介

- TextView 文本框
- Button 按键
- EditText 文本输入框
- Switch 开关
- SeekBar 拖动条（进度条）
- LineChart 折线图
- JoyStick 摇杆

图 5.1 UI 控件

5.2 文件

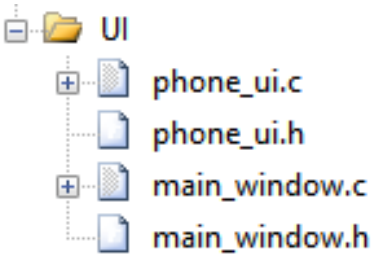

图 5.2 UI 文件

UI 底层文件：
phone_ui.c　UI 底层文件
phone_ui.h　UI 底层文件头文件
UI 编写文件：
main_window.c　可在此文件中构建 UI
main_window.h　UI 头文件

5.3 文件移植

（1）添加 UI 相关文件 phone_ui.c、phone_ui.h、main_window.c、main_window.h 到工程中。
（2）添加串口收发相关头文件。在 phone_ui.c 中添加串口发送所在头文件；将 phone_ui.h 添加到串口接收中断函数所在文件中。
（3）修改 phone_ui.c 中串口发送函数

```
//串口发送一个字节
void UI_SendByte(unsigned char byte)
{
    Uart_SendByte(USART1,byte); //修改此处为工程中串口发送一个字节函数
}
```

（4）在串口接收中断中添加 UI_ByteDeal(Res)，Res 为串口接收到的一个字节

（5）在 phone_ui.h 中宏定义 REAL_TIME_PROCESSING 的值选择实时处理（接收一帧后在串口中断中直接处理）与在主程序中处理。

```
#define REAL_TIME_PROCESSING 0 //（1）实时处理（0）在主程序中处理
```

如选择在主程序中处理，在主程序中运行 UI_BufferDeal()即可。

5.4 控件使用

以下操作在 main_window.c 以及 main_window.h 中构建，
或参考 main_window.c 与 main_window.h 另外新建文件构建页面。

5.4.1 窗口设置

```
Window_Clear(); //清空屏幕
Window_SetBackground(0XFFFFFFFF); //设置窗口颜色
Window_SetSize(1000,2000,0);//设置窗口宽高
```

5.4.2 添加一个 TextView 控件

（1）在 main_window.c 中创建一个 TextView 的结构体变量

```
TextView_TypeDef    main_tv;
```

（2）在 main_window.h 中将控件声明为全局变量

```
extern TextView_TypeDef    main_tv;
```

（3）调用 TextView 初始化函数

```
TextView_Init(main_tv,50,50,500,500);
```

（4）TextView 显示文本

```
TextView_SetText(main_tv,"文本");
```

5.4.3 添加一个 Button 控件

（1）创建一个 Button 结构体变量

```
Button_TypeDef    main_bt;
```

（2）在.h 中将控件声明为全局变量

```
extern Button_TypeDef    main_bt;
```

（3）调用 Button 初始化函数

```
Button_Init(main_bt,50,50,300,150);
```

（4）查询 Button 状态

```
main_bt.isPressed    //(=0)按钮抬起，(=1)按钮按下
main_bt.isUpdated;    //(=1)数值已更新，需手动置 0
```

5.4.4 添加一个 EditText 控件

（1）创建一个 EditText 结构体变量

```
EditText_TypeDef    main_et;
```

（2）在.h 中将控件声明为全局变量

```
extern    EditText_TypeDef    main_et;
```

（3）调用 EditText 初始化函数

```
EditText_Init(main_et,50,50,700,160);
```

（4）查询 EditText 输入字符串

```
main_et.text[EditText_MaxNum];  //输入字符串，有效字符串以'\0'结尾
main_et.isUpdated;  //(=1)数值已更新，需手动置 0
```

5.4.5 添加一个 Switch 控件

（1）创建一个 Switch 结构体变量

```
Switch_TypeDef  main_sw;
```

（2）在.h 中将控件声明为全局变量

```
extern  Switch_TypeDef  main_sw;
```

（3）调用 Switch 初始化函数

```
Switch_Init(main_sw,50,50,150,150);
```

（4）查询 Switch 状态

```
main_sw.isChecked;   //(=0)开关关闭，(=1)开关打开
main_sw.isUpdated;   //(=1)数值已更新，需手动置 0
```

5.4.6 添加一个 SeekBar 控件

（1）创建一个 SeekBar 结构体变量

```
SeekBar_TypeDef  mian_sb;
```

（2）在.h 中将控件声明为全局变量

```
extern  SeekBar_TypeDef  mian_sb;
```

（3）调用 SeekBar 初始化函数

```
SeekBar_Init(mian_sb,50,50,700,100);
```

（4）SeekBar 相关用法

```
mian_sb.progress;   //当前值
mian_sb.isUpdated;  //(=1)数值已更新，需手动置 0
SeekBar_SetProgress(main_sb,50); //设置 SeekBar 当前值，用作进度条
```

5.4.7 添加一个 LineChart 控件

（1）创建一个 LineChart 结构体变量

```
LineChart_TypeDef  mian_lc;
```

（2）在.h 中将控件声明为全局变量

```
extern  LineChart_TypeDef  mian_lc;
```

（3）调用 LineChart 初始化函数

```
LineChart_Init(mian_lc,50,50,700,350);
```

（4）LineChart 添加数据

```
LineChart_AddData(main_lc,50); //曲线添加一个函数
```

5.4.8 添加一个 JoyStick 控件

（1）创建一个 JoyStick 结构体变量

```
JoyStick_TypeDef  mian_js;
```

（2）在.h 中将控件声明为全局变量

```
extern  JoyStick_TypeDef  mian_js;
```

（3）调用 JoyStick 初始化函数

```
JoyStick_Init(mian_js,50,50,400,400);
```

（4）查询 JoyStick 的值

```
mian_js.xValue; //X 的值（0-200）
mian_js.yValue;//Y 的值（0-200）
mian_js.isUpdated;//(=1)数值已更新，需手动置 0 其他方法参考例程
```

5.4.9 其他

- 其他方法在 phone_ui.h 最下方查看。
- 当需要多个样式类似控件时，可以使用 UI 中控件 Style 结构体为控件快速设置样式。

5.5 其他操作

- 蓝牙连接：点击右上角三个点，可快速进入蓝牙连接界面
- 设置壁纸：点击右上角三个点->设置->更换壁纸
- 设置屏幕方向：点击右上角三个点->设置->切换屏幕方向
- 当使用拖动条、摇杆这类频繁传输数据的控件时，建议使用经典蓝牙（蓝牙 2.0），并将波特率设置为 115200，否则数据传输可能会有延迟

六、遥控器

6.1 遥控器窗口

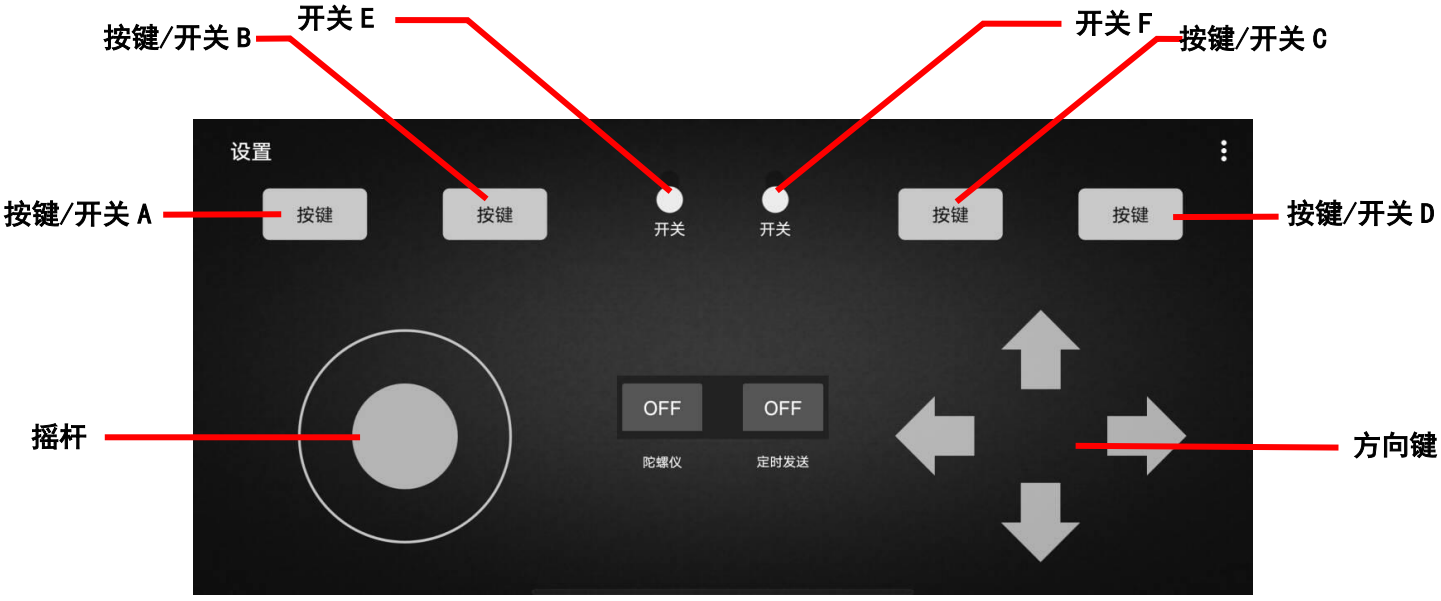

图 6.1 遥控器界面

- 开启设置可选择摇杆与方向按键
- 开启设置后点击按键或开关名称可自定义按键与开关的名称，发送内容不可修改
- 开启设置后上划按键可将按键切换为开关
- 开启设置后点击时间可自定义发送间隔，默认发送间隔 100ms
- 下位机发送的数据将在屏幕中心显示，以”\r\n”检测结尾，超过 64 字节将直接显示

6.2 数据格式

6.2.1 按扭

表 6.1 实时发送模式按钮数值（16 进制）

	按下（开）	抬起（关）
左上	11	10
左下	21	20
左左	31	30
左右	41	40
右上	51	50
右下	61	60
右左	71	70
右右	81	80
按键/开关 A	A1	A0
按键/开关 B	B1	B0
按键/开关 C	C1	C0
按键/开关 D	D1	D0
开关 E	E1	E0
开关 F	F1	F0

6.2.2 摇杆

- 范围：X 0-200,Y 0-200
- 实时发送模式下数据格式：高 4 位为命令，低 4 位为数据
- 左摇杆：1X/16 2X%16 3Y/16 4Y%16
- 右摇杆：5X/16 6X%16 7Y/16 8Y%16
- 例：在左摇杆为 X=50=3*16+2，Y=160=A*16+0 时 发送数据：13 22 3A 40（16 进制）

6.2.3 定时发送

左X+左Y（6字节）或 上+下+左+右（4字节） + 右X+右Y（6字节）或 上+下+左+右（4字节） + 按键A或开关A + 按键B或开关B + 按键C或开关C + 按键D或开关D + 开关E+开关F+"\r\n"

图 6.2 遥控器定时发送数据格式

- 例：0501601000100010\r\n（字符）
  表示左 X 为 50，左 Y 为 160，右上方向键按下，按键 A 按下，开关 E 打开

七、秒表

图 7.1 秒表

通过发送十六进制数据操作秒表

- 启动：0x31
- 计次：0x32
- 停止：0x33
- 复位：0x34

➢ 至多计次 500 次

八、时钟

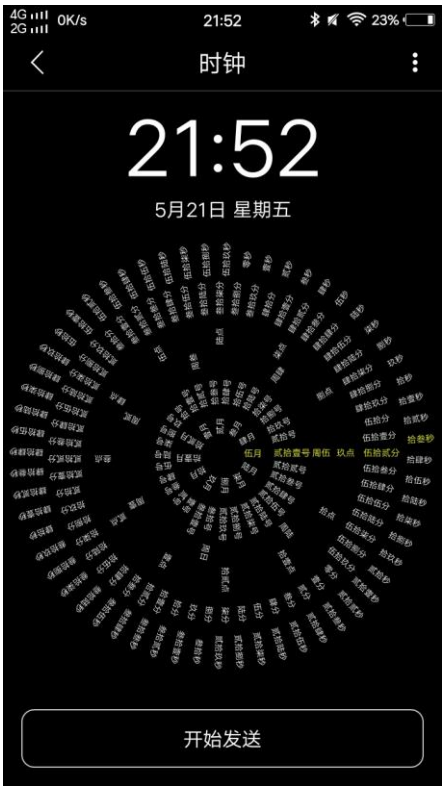

图 8.1 时钟

➢ 数据格式：
例：2022 年 5 月 21 日周五 21 点 35 分 24.1 秒
数据：20220521052135241\r\n

➢ 数据每隔 100ms 发送一次，精度为 100ms

➢ 周日为 00，周一为 01，依次类推

九、语音播报

图 9.1 语音播报

- 播放一句话：语音内容+"\r\n"（0x0D 0x0A）

例：播放电子学会，下位机调用 printf("电子学会\r\n");
或发送数组 uint8_t string[10]="电子学会\r\n";
或使用字符串发送函数发送

- 播方音频：

  例：播方音频 1，下位机发送 01 0D 0A（16 进制）

  - 最多能添加 30 个音频
  - 同时支持中文，英文。若无法播方可能是手机不支持
  - 中文至多 50 字，或 100 字符

十、GPS

10.1 窗口

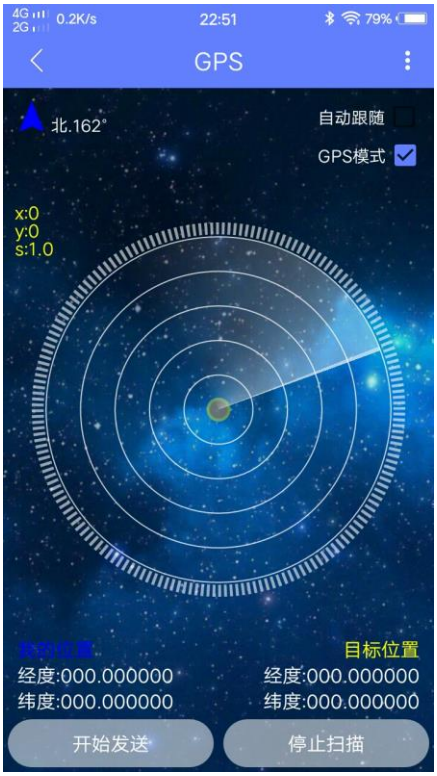

图 10.1 GPS

- 功能分为 GPS 模式与自定义模式
- 自定义模式下点击“我的位置”可自定义我的位置
- GPS 模式下发送手机位置到下位机

  例：经度 115.546627 纬度 35.37776

  数据为：115546627035377760\r\n（字符）

  - 使用时打开 GPS 定位
  - 室内可能无法获取定位
  - 发送周期为 1s

10.2 下位机发送位置到上位机

```
/*************************************************************
 *函数名称：XHZS_SendWave
 *简    介：学会助手 APP 虚拟示波器下位机程序（兼容山外）
 *输    入：wareaddr:数组地址  waresize:数组大小
 *输    出：无
 *注意事项：无
 *************************************************************/
void XHZS_SendWave(uint8_t *waveaddr, uint16_t wavesize)
{
    uint8_t cmdf[2] = {0x03, 0xFC};     //帧头
    uint8_t cmdr[2] = {0xFC, 0x03};    //帧尾
    Uart_SendArray(USART1,cmdf, sizeof(cmdf));      //发送帧头
    Uart_SendArray(USART1,waveaddr, wavesize);    //发送数据
    Uart_SendArray(USART1,cmdr, sizeof(cmdr));     //发送帧尾
}

/*************************************************************
 *函数名称：SendLocation
 *简    介：发送经纬度到学会助手 APP
 *输    入：longitude:经度  latitude:纬度
 *输    出：无
 *注意事项：无
 *************************************************************/
void SendLocation(float longitude,float latitude)
{
    float Wave[2]={0};

    Wave[0] =longitude;
    Wave[1] =latitude;
     XHZS_SendWave((uint8_t *)Wave,sizeof(Wave));
}
/*************************************************************
 *函数名称：Send_XY
 *简    介：发送坐标到学会助手 APP
 *输    入：x ,y
 *输    出：无
 *注意事项：无
 *************************************************************/
void SendXY(int x,int y)
{
    int Wave[2]={0};
    Wave[0] =x;
    Wave[1] =y;
     XHZS_SendWave((uint8_t *)Wave,sizeof(Wave));
}
```

十一、画板

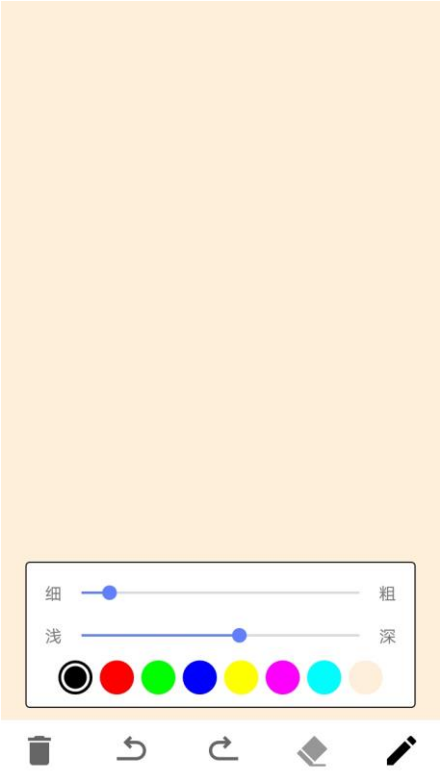

图 11.1 画板

- APP 实时发送数据到下位机，X 与 Y 数据长度均为 4

例：X:568 Y: 627

数据格式为 05680627\r\n

十二、提示

- 使用前可将数据发送至电脑端串口查看，了解数据格式。
- 可在工具->设置->资料下载中下载相关资料