



实验 10 物理按键实验

1. 实验目的

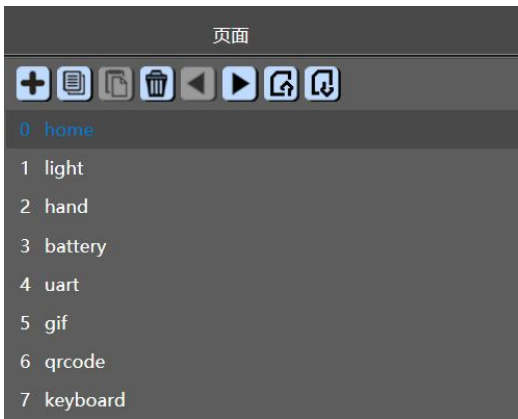
本实验演示了，不用屏幕自带的触摸板时，使用外接的串口按键板来控制串口屏的显示。如使用按键板的上、下、左、右，确认键实现切页、参数配置、参数输入等功能。

本实验配套的例程有两个工程，【物理按键_三个键值.pix】演示了按键板使用左、右和确认三个按键的实现方式；【物理按键_五个键值.pix】演示了按键板使用左、右、确认、上、下五个按键的实现方式，本实验以【物理按键_三个键值.pix】详细说明。本实验的这种物理按键实现方式比较灵活，弄清楚本实验的实现方式后，用户可根据这种方式自行调整以达到实现用户自己项目的预期效果。

本实验基于 2.4 寸串口屏，若要验证其他尺寸的串口屏，可在 VP 软件里修改项目型号（文件->编辑项目）。

2. 页面设计

本实验为演示物理按键的控制功能，一共有 8 个页面可供切换，如下：



(1) home: 主页面。一共 6 个按钮，点击每个按钮可切换到相应的页面。



(2) light:亮度控制页面。



3 个按钮，“-”按钮(控制系统亮度-10)，“+”按钮(控制系统亮度+10)，“回到主页面”按钮(点击时返回主界面)。

1 个整数控件(显示具体的亮度值)。

1 个滑块控件(用滑块的位置来指示亮度值)。



(3) hand:汽车速度指针页面。

3 个按钮，“打开”按钮(启动定时器 2，指针开始顺时针转动)，“关闭”按钮(指针指向零刻度)，“回到主页面”按钮(点击时返回主界面)。

3 个指针控件，分别用于指示转速、速度、油表。



(4) battery:电池充放电模拟页面。

3 个按钮，“充电”按钮(启动定时器，电池和环形充电电池的电量值增长)，“放电”按钮(启动定时器，电池和环形充电电池的电量值减少)，“回到主页面”按钮(点击时返回主界面)。

1 个进度条控件，指示电池电量。

1 个环形进度条控件，指示环形充电电池电量。



(5) uart: 串口发送页面。

3 个按钮，“上发送”按钮(发送整数)，“下发送”按钮(发送浮点数)，“回到主页面”按钮(点击时返回主界面)。

1 个整数控件。

1 个浮点数控件。



(6) gif: 动图显示页面。

3 个按钮，“播放”按钮(播放动图)，“暂停”按钮(暂停动图)，“回到主页面”按钮(点击时返回主界面)。

1 个 GIF 控件，显示动图。

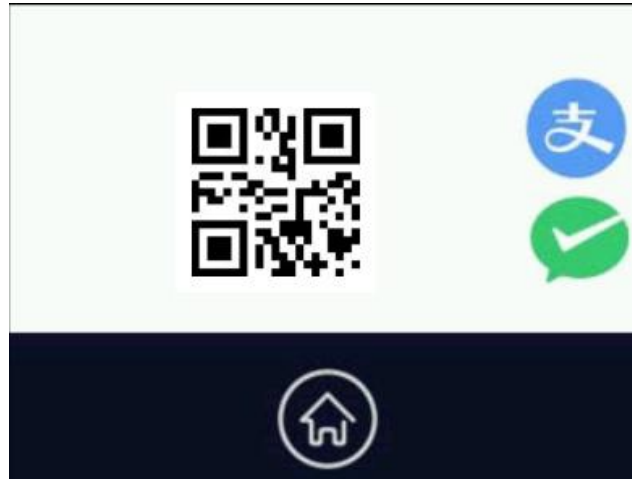




(7) qrcode:二维码显示页面。

3 个按钮，“支付宝”按钮(设置二维码为 zhifubao)，“微信”按钮(设置二维码为 weixin)，“回到主页面”按钮(点击时返回主界面)。

1 个二维码控件。



(8) keyboard:键盘页面。

15 个按钮，0~9 和小数点为普通功能按钮，“←”为退格功能，“清除”为清空输入框，“退出”为退出键盘，“确认”按键将输入框的内容赋值到指定控件。

1 个文本控件，表现为输入框。



3. 串口屏脚本处理

工程【物理按键_三个键值.pix】一共需要处理三个键值 右(AA 01 BB),左(AA 02 BB),确认(AA 03 BB)。

工程【物理按键_五个键值.pix】一共需要处理五个键值 右(AA 01 BB),左(AA 02 BB),确认(AA 03 BB),下(AA 04 BB),上(AA 05 BB)。

脚本代码主要在协议解析器控件的数据接收事件里，选中 home 页面的协议解析器控件 proc，可以看到处理的脚本。

【物理按键_三个键值.pix】如下：

```
/*  
接收处理按键板发过来的键值，一共三个键值 右(AA 01 BB),左(AA 02 BB),确认(AA 03 BB)  
cursor 变量表示当前光标的位置，通过判断光标的位置，来设置按钮的背景图片，实现左右选中的效果。  
注意:跨页访问控件属性时，需要指明页名称，并将要访问的控件设置成全局。  
*/
```

```
if((proc.rxBuf[0] == 0xAA)&&(proc.rxBuf[2] == 0xBB)) //检测键值 帧头 帧尾
{
    if(sys_pid == home.id) //当前处于 home 页面
    {

        if((proc.rxBuf[1] == 1) || (proc.rxBuf[1] == 2))
        {
            //根据旧光标位置，设置按钮的背景图片 为 常态背景图片
            if(cursor.val == 1) button1.bgImg = user.image.Main;
            else if(cursor.val == 2) button2.bgImg = user.image.Main;
            else if(cursor.val == 3) button3.bgImg = user.image.Main;
            else if(cursor.val == 4) button4.bgImg = user.image.Main;
            else if(cursor.val == 5) button5.bgImg = user.image.Main;
            else if(cursor.val == 6) button6.bgImg = user.image.Main;

            //根据键值改变光标的位置
            if(proc.rxBuf[1] == 1) //右
            {
                if(++cursor.val > 6) cursor.val = 1;
            }
            else if(proc.rxBuf[1] == 2) //左
            {
                if(--cursor.val < 1) cursor.val = 6;
            }

            //根据新光标的位罝，设置按钮的背景图片为深色类型，实现选中的效果
            if(cursor.val == 1) button1.bgImg = user.image.Main2;
            else if(cursor.val == 2) button2.bgImg = user.image.Main2;
            else if(cursor.val == 3) button3.bgImg = user.image.Main2;
            else if(cursor.val == 4) button4.bgImg = user.image.Main2;
            else if(cursor.val == 5) button5.bgImg = user.image.Main2;
            else if(cursor.val == 6) button6.bgImg = user.image.Main2;
        }
        else if(proc.rxBuf[1] == 3)
        {
            //确认键 执行按钮的点击事件
            if(cursor.val == 1) click(button1.x, button1.y);
            else if(cursor.val == 2) click(button2.x, button2.y);
            else if(cursor.val == 3) click(button3.x, button3.y);
            else if(cursor.val == 4) click(button4.x, button4.y);
            else if(cursor.val == 5) click(button5.x, button5.y);
            else if(cursor.val == 6) click(button6.x, button6.y);
        }
    }
    else if(sys_pid == light.id)
    {
        if((proc.rxBuf[1] == 1) || (proc.rxBuf[1] == 2))
        {
```



```
//根据旧光标位置，设置按钮的背景图片 为 常态背景图片
if(cursor.val == 1) light.button1.bgImg = user.image.light;
else if(cursor.val == 2)light.button2.bgImg = user.image.light;
else if(cursor.val == 3)light.button3.bgImg = user.image.light;

//根据键值改变光标的位置
if(proc.rxBuf[1]== 1) //右
{
    if(++cursor.val > 3)cursor.val = 1;
}
else if(proc.rxBuf[1]== 2) //左
{
    if(--cursor.val < 1)cursor.val = 3;
}

//根据新光标的位罝，设置按钮的背景图片为深色类型，实现选中的效果
if(cursor.val == 1)light.button1.bgImg = user.image.light2;
else if(cursor.val == 2)light.button2.bgImg = user.image.light2;
else if(cursor.val == 3)light.button3.bgImg = user.image.light2;
}
else if(proc.rxBuf[1] == 3)
{
    //确认键 执行按钮的点击事件
    if(cursor.val == 1)click(light.button1.x,light.button1.y);
    else if(cursor.val == 2)click(light.button2.x,light.button2.y);
    else if(cursor.val == 3)click(light.button3.x,light.button3.y);
}

}

else if(sys_pid == hand.id)
{
    if((proc.rxBuf[1] == 1) || (proc.rxBuf[1] == 2))
    {
        //根据旧光标位置，设置按钮的背景图片 为 常态背景图片
        if(cursor.val == 1) hand.button1.bgImg = user.image.hand;
        else if(cursor.val == 2)hand.button2.bgImg = user.image.hand;
        else if(cursor.val == 3)hand.button3.bgImg = user.image.hand;

        //根据键值改变光标的位置
        if(proc.rxBuf[1]== 1) //右
        {
            if(++cursor.val > 3)cursor.val = 1;
        }
        else if(proc.rxBuf[1]== 2) //左
        {
            if(--cursor.val < 1)cursor.val = 3;
        }
    }
}
```



```
}

//根据新光标的位置，设置按钮的背景图片为深色类型，实现选中的效果
if(cursor.val == 1)hand.button1.bgImg = user.image.hand2;
else if(cursor.val == 2)hand.button2.bgImg = user.image.hand2;
else if(cursor.val == 3)hand.button3.bgImg = user.image.hand2;
}
else if(proc.rxBuf[1] == 3)
{
    //确认键 执行按钮的点击事件
    if(cursor.val == 1)click(hand.button1.x,hand.button1.y);
    else if(cursor.val == 2)click(hand.button2.x,hand.button2.y);
    else if(cursor.val == 3)click(hand.button3.x,hand.button3.y);
}
}

else if(sys_pid == battery.id)
{
    if((proc.rxBuf[1] == 1) || (proc.rxBuf[1] == 2))
    {
        //根据旧光标位置，设置按钮的背景图片 为 常态背景图片
        if(cursor.val == 1) battery.button1.bgImg = user.image.cell;
        else if(cursor.val == 2)battery.button2.bgImg = user.image.cell;
        else if(cursor.val == 3)battery.button3.bgImg = user.image.cell;

        //根据键值改变光标的位置
        if(proc.rxBuf[1]== 1) //右
        {
            if(++cursor.val > 3)cursor.val = 1;
        }
        else if(proc.rxBuf[1]== 2) //左
        {
            if(--cursor.val < 1)cursor.val = 3;
        }
    }

    //根据新光标的位置，设置按钮的背景图片为深色类型，实现选中的效果
    if(cursor.val == 1)battery.button1.bgImg = user.image.cell2;
    else if(cursor.val == 2)battery.button2.bgImg = user.image.cell2;
    else if(cursor.val == 3)battery.button3.bgImg = user.image.cell2;
}
else if(proc.rxBuf[1] == 3)
{
    //确认键 执行按钮的点击事件
    if(cursor.val == 1)click(battery.button1.x,battery.button1.y);
    else if(cursor.val == 2)click(battery.button2.x,battery.button2.y);
    else if(cursor.val == 3)click(battery.button3.x,battery.button3.y);
}
}
```



```
else if(sys_pid == uart.id)
{
    if((proc.rxBuf[1] == 1) || (proc.rxBuf[1] == 2))
    {
        //根据旧光标位置，设置按钮的背景图片 为 常态背景图片
        if(cursor.val == 1) uart.num1.bgImg = user.image.uart;
        else if(cursor.val == 2)uart.button2.bgImg = user.image.uart;
        else if(cursor.val == 3)uart.numf3.bgImg = user.image.uart;
        else if(cursor.val == 4)uart.button4.bgImg = user.image.uart;
        else if(cursor.val == 5)uart.button5.bgImg = user.image.uart;

        //根据键值改变光标的位置
        if(proc.rxBuf[1]== 1) //右
        {
            if(++cursor.val > 5)cursor.val = 1;
        }
        else if(proc.rxBuf[1]== 2) //左
        {
            if(--cursor.val < 1)cursor.val = 5;
        }

        //根据新光标的位置，设置按钮的背景图片为深色类型，实现选中的效果
        if(cursor.val == 1)uart.num1.bgImg = user.image.uart2;
        else if(cursor.val == 2)uart.button2.bgImg = user.image.uart2;
        else if(cursor.val == 3)uart.numf3.bgImg = user.image.uart2;
        else if(cursor.val == 4)uart.button4.bgImg = user.image.uart2;
        else if(cursor.val == 5)uart.button5.bgImg = user.image.uart2;
    }
    else if(proc.rxBuf[1] == 3)
    {
        //确认键 执行按钮的点击事件
        if(cursor.val == 1)click(uart.num1.x,uart.num1.y);
        else if(cursor.val == 2)click(uart.button2.x,uart.button2.y);
        else if(cursor.val == 3)click(uart.numf3.x,uart.numf3.y);
        else if(cursor.val == 4)click(uart.button4.x,uart.button4.y);
        else if(cursor.val == 5)click(uart.button5.x,uart.button5.y);
    }
}

else if(sys_pid == gif.id)
{
    if((proc.rxBuf[1] == 1) || (proc.rxBuf[1] == 2))
    {
        //根据旧光标位置，设置按钮的背景图片 为 常态背景图片
```



```
if(cursor.val == 1) gif.button1.bgImg = user.image.GIF;
else if(cursor.val == 2)gif.button2.bgImg = user.image.GIF;
else if(cursor.val == 3)gif.button3.bgImg = user.image.GIF;

//根据键值改变光标的位置
if(proc.rxBuf[1]== 1) //右
{
    if(++cursor.val > 3)cursor.val = 1;
}
else if(proc.rxBuf[1]== 2) //左
{
    if(--cursor.val < 1)cursor.val = 3;
}

//根据新光标的位置，设置按钮的背景图片为深色类型，实现选中的效果
if(cursor.val == 1)gif.button1.bgImg = user.image.GIF2;
else if(cursor.val == 2)gif.button2.bgImg = user.image.GIF2;
else if(cursor.val == 3)gif.button3.bgImg = user.image.GIF2;
}
else if(proc.rxBuf[1] == 3)
{
    //确认键 执行按钮的点击事件
    if(cursor.val == 1)click(gif.button1.x,gif.button1.y);
    else if(cursor.val == 2)click(gif.button2.x,gif.button2.y);
    else if(cursor.val == 3)click(gif.button3.x,gif.button3.y);
}
}

else if(sys_pid == qrcode.id)
{
    if((proc.rxBuf[1] == 1) || (proc.rxBuf[1] == 2))
    {
        //根据旧光标位置，设置按钮的背景图片 为 常态背景图片
        if(cursor.val == 1)qrcode.button1.bgImg = user.image.qrcode;
        else if(cursor.val == 2)qrcode.button2.bgImg = user.image.qrcode;
        else if(cursor.val == 3)qrcode.button3.bgImg = user.image.qrcode;

        //根据键值改变光标的位置
        if(proc.rxBuf[1]== 1) //右
        {
            if(++cursor.val > 3)cursor.val = 1;
        }
        else if(proc.rxBuf[1]== 2) //左
        {
            if(--cursor.val < 1)cursor.val = 3;
        }
    }
}
```



```
//根据新光标的位置，设置按钮的背景图片为深色类型，实现选中的效果
if(cursor.val == 1)  qrcode.button1.bgImg = user.image.qrcode2;
else if(cursor.val == 2)qrcode.button2.bgImg = user.image.qrcode2;
else if(cursor.val == 3)qrcode.button3.bgImg = user.image.qrcode2;
}
else if(proc.rxBuf[1] == 3)
{
    //确认键 执行按钮的点击事件
    if(cursor.val == 1)click(qrcode.button1.x,qrcode.button1.y);
    else if(cursor.val == 2)click(qrcode.button2.x,qrcode.button2.y);
    else if(cursor.val == 3)click(qrcode.button3.x,qrcode.button3.y);
}
}

else if(sys_pid == keyboard.id)
{
    if((proc.rxBuf[1] == 1) || (proc.rxBuf[1] == 2))
    {
        //根据旧光标位置，设置按钮的背景图片 为 常态背景图片
        if(cursor.val == 1)  keyboard.button1.bgImg = user.image.key;
        else if(cursor.val == 2)keyboard.button2.bgImg = user.image.key;
        else if(cursor.val == 3)keyboard.button3.bgImg = user.image.key;
        else if(cursor.val == 4)keyboard.button4.bgImg = user.image.key;
        else if(cursor.val == 5)keyboard.button5.bgImg = user.image.key;
        else if(cursor.val == 6)keyboard.button6.bgImg = user.image.key;
        else if(cursor.val == 7)keyboard.button7.bgImg = user.image.key;
        else if(cursor.val == 8)keyboard.button8.bgImg = user.image.key;
        else if(cursor.val == 9)keyboard.button9.bgImg = user.image.key;
        else if(cursor.val == 10)keyboard.button10.bgImg = user.image.key;
        else if(cursor.val == 11)keyboard.button11.bgImg = user.image.key;
        else if(cursor.val == 12)keyboard.button12.bgImg = user.image.key;
        else if(cursor.val == 13)keyboard.button13.bgImg = user.image.key;
        else if(cursor.val == 14)keyboard.button14.bgImg = user.image.key;
        else if(cursor.val == 15)keyboard.button15.bgImg = user.image.key;

        //根据键值改变光标的位置
        if(proc.rxBuf[1]== 1) //右
        {
            if(++cursor.val > 15)cursor.val = 1;
        }
        else if(proc.rxBuf[1]== 2) //左
        {
            if(--cursor.val < 1)cursor.val = 15;
        }

        //根据新光标的位置，设置按钮的背景图片为深色类型，实现选中的效果
        if(cursor.val == 1)keyboard.button1.bgImg = user.image.key2;
```



```

else if(cursor.val == 2)keyboard.button2.bgImg = user.image.key2;
else if(cursor.val == 3)keyboard.button3.bgImg = user.image.key2;
else if(cursor.val == 4)keyboard.button4.bgImg = user.image.key2;
else if(cursor.val == 5)keyboard.button5.bgImg = user.image.key2;
else if(cursor.val == 6)keyboard.button6.bgImg = user.image.key2;
else if(cursor.val == 7)keyboard.button7.bgImg = user.image.key2;
else if(cursor.val == 8)keyboard.button8.bgImg = user.image.key2;
else if(cursor.val == 9)keyboard.button9.bgImg = user.image.key2;
else if(cursor.val == 10)keyboard.button10.bgImg = user.image.key2;
else if(cursor.val == 11)keyboard.button11.bgImg = user.image.key2;
else if(cursor.val == 12)keyboard.button12.bgImg = user.image.key2;
else if(cursor.val == 13)keyboard.button13.bgImg = user.image.key2;
else if(cursor.val == 14)keyboard.button14.bgImg = user.image.key2;
else if(cursor.val == 15)keyboard.button15.bgImg = user.image.key2;
}
else if(proc.rxBuf[1] == 3)
{
    //确认键 执行按钮的点击事件
    if(cursor.val == 1)click(keyboard.button1.x,keyboard.button1.y);
    else if(cursor.val == 2)click(keyboard.button2.x,keyboard.button2.y);
    else if(cursor.val == 3)click(keyboard.button3.x,keyboard.button3.y);
    else if(cursor.val == 4)click(keyboard.button4.x,keyboard.button4.y);
    else if(cursor.val == 5)click(keyboard.button5.x,keyboard.button5.y);
    else if(cursor.val == 6)click(keyboard.button6.x,keyboard.button6.y);
    else if(cursor.val == 7)click(keyboard.button7.x,keyboard.button7.y);
    else if(cursor.val == 8)click(keyboard.button8.x,keyboard.button8.y);
    else if(cursor.val == 9)click(keyboard.button9.x,keyboard.button9.y);
    else if(cursor.val == 10)click(keyboard.button10.x,keyboard.button10.y);
    else if(cursor.val == 11)click(keyboard.button11.x,keyboard.button11.y);
    else if(cursor.val == 12)click(keyboard.button12.x,keyboard.button12.y);
}
}

```

【物理按键_三个键值.pix】处理 右(AA 01 BB),左(AA 02 BB)两个键值时, home 页面上的变量控件 cursor 用于指示光标的位置。通过判断光标的位置, 来设置按钮的背景图片为深色类型, 实现左右移动的效果。如 home 页面的光标位置有 0(不显示光标),

- 1(光标在 button1 按钮上)
- 2(光标在 button2 按钮上)
- 3(光标在 button3 按钮上)
- 4(光标在 button4 按钮上)
- 5(光标在 button5 按钮上)
- 6(光标在 button6 按钮上)

每一页光标的值默认是 0, 光标值随键值而变化, 如:

home 页面

循环按右键(AA 01 BB)时, 光标值的变化为

0→1→2→3→4→5→6→1→2→3→4→5→6→1→2→3→4→5→.....

循环按左键(AA 02 BB)时, 光标值的变化为

0→6→5→4→3→2→1→6→5→4→3→2→1→6→5→4→3→2→.....

light 页面



循环按左键(AA 01 BB)时，光标值的变化为

0→1→2→3→1→2→3→1→2→3→1→2→.....

循环按左键(AA 02 BB)时，光标值的变化为

0→3→2→1→3→2→1→3→2→.....

【物理按键_五个键值.pix】与【物理按键_三个键值.pix】的差异主要是体现在光标值的变化上。

【物理按键_三个键值.pix】只处理左右两个键值，因此键值的变化比较简单，cursor 加或者减即可。

【物理按键_五个键值.pix】同时需要考虑左右两个键值，还要考虑上下两个键值。上下两个键值的变化不是简单的加减，需要自行定义，实验中每一页定义了两个数组 arrh, arrv。其中 arrh 用于保存左右键值变化的规律，arrv 用于保存上下键值变化的规律。数组的初始化，写在 home 页面的加载页面事件里，并用 startflag 变量标记，只需初始化一次即可。如后面页面的数组初始化如下：

```
//home 页
arrh.valbs[0] = 0;
arrh.valbs[1] = 1;
arrh.valbs[2] = 2;
arrh.valbs[3] = 3;
arrh.valbs[4] = 4;
arrh.valbs[5] = 5;
arrh.valbs[6] = 6;

arrv.valbs[0] = 0;
arrv.valbs[1] = 1;
arrv.valbs[2] = 3;
arrv.valbs[3] = 5;
arrv.valbs[4] = 2;
arrv.valbs[5] = 4;
arrv.valbs[6] = 6;
```

每一页光标的值默认是 0，根据页面上的按钮，光标值随键值而变化，如：

home 页面

循环按右键(AA 01 BB)时，光标值的变化为

0→1→2→3→4→5→6→1→2→3→4→5→6→1→2→3→4→5→.....

循环按左键(AA 02 BB)时，光标值的变化为

0→6→5→4→3→2→1→6→5→4→3→2→1→6→5→4→3→2→.....

循环按下键(AA 01 BB)时，光标值的变化为

0→1→3→5→2→4→6→1→3→5→2→4→6→1→3→5→2→4→.....

循环按上键(AA 02 BB)时，光标值的变化为

0→6→4→2→5→3→1→6→4→2→5→3→1→6→4→2→5→3→.....

【物理按键_五个键值.pix】部分脚本代码如下

```
if((proc.rxBuf[0] == 0xAA)&&(proc.rxBuf[1] <= 5)&&(proc.rxBuf[2] == 0xBB)) //检测键值 帧头 帧尾
{
    if(sys_pid == home.id) //当前处于 home 页面
    {
        if(proc.rxBuf[1] != 3)
        {
            //根据旧光标位置，设置按钮的背景图片 为 常态背景图片
            if(cursor.val == 1) button1.bgimg = user.image.Main;
```



```
else if(cursor.val == 2)button2.bgImg = user.image.Main;
else if(cursor.val == 3)button3.bgImg = user.image.Main;
else if(cursor.val == 4)button4.bgImg = user.image.Main;
else if(cursor.val == 5)button5.bgImg = user.image.Main;
else if(cursor.val == 6)button6.bgImg = user.image.Main;

//根据键值改变光标的位置
get_new_cursor(); //自定义函数，工具->函数编辑器 里可以查看和修改

//根据新光标的位罝，设置按钮的背景图片为深色类型，实现选中的效果
if(cursor.val == 1)button1.bgImg = user.image.Main2;
else if(cursor.val == 2)button2.bgImg = user.image.Main2;
else if(cursor.val == 3)button3.bgImg = user.image.Main2;
else if(cursor.val == 4)button4.bgImg = user.image.Main2;
else if(cursor.val == 5)button5.bgImg = user.image.Main2;
else if(cursor.val == 6)button6.bgImg = user.image.Main2;
}
else
{
    //确认键 执行按钮的点击事件
    if(cursor.val == 1)click(button1.x,button1.y);
    else if(cursor.val == 2)click(button2.x,button2.y);
    else if(cursor.val == 3)click(button3.x,button3.y);
    else if(cursor.val == 4)click(button4.x,button4.y);
    else if(cursor.val == 5)click(button5.x,button5.y);
    else if(cursor.val == 6)click(button6.x,button6.y);
}
}
```

可以看到改变光标位置的时候，调用了—个函数 `get_new_cursor()`，这个函数是自定义的，可在工具->函数编辑器 里可以查看和修改，主要是处理五个键值时，光标位置的变化相对复杂一些，这时候自定义一个函数，可使脚本代码更加清晰。

`get_new_cursor()`功能大概是，根据页面和键值的类型，找到当前光标的位置；再根据 `arrh` 或者 `arrv` 数组中定义的变化规则，改变光标的位置。

处理 确认(AA 03 BB)键值时，根据光标的所在位置，调用 `click` 函数，执行对应按钮的脚本，实现切页、控制功能。

注意：访问其他页面的控件时，需要勾选访问的控件 `global` 属性，并在访问时加上页名称前缀，如 `hand.button2.bgImg`

本实验由于演示的页面比较多，为了使脚本代码更清晰，脚本要访问控件的名字务必定义清晰，增加代码的易读性，可按照下面的规则：

- 1、名称简短
- 2、最好能体现实际意义
- 3、相同属性的控件，最好带有一定顺序的编号



4. 物理按键板单片机程序

单片机的程序很简单，检测到按钮按下并发送键值即可。

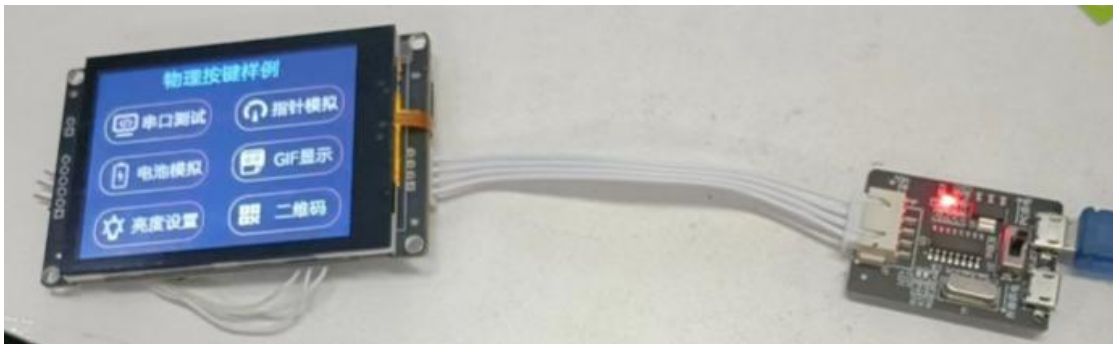
```
char uart_buf[3] = {0xAA,0x00,0xBB};
int main(void)
{
    u8 key_value;

    Sys_SetRcc();                //设置系统使用内部时钟
    delay_init(64);              //延时函数初始化
    usart1_init();               //初始化串口
    KEY_Init();                  //按键初始化
    while(1)
    {
        key_value = KEY_Scan(0); //得到键值
        if(key_value) //上 7 左 8 右 6 下 11 中 5
        {
            if(key_value == J6_PRES) //按键 右
            {
                uart_buf[1] = 0x01;
                USART_OUT(USART1, (uint8_t*)uart_buf,3);
            }
            else if(key_value == J8_PRES) //按键 左
            {
                uart_buf[1] = 0x02;
                USART_OUT(USART1, (uint8_t*)uart_buf,3);
            }
            else if(key_value == J5_PRES) //按键 OK
            {
                uart_buf[1] = 0x03;
                USART_OUT(USART1, (uint8_t*)uart_buf,3);
            }
            else if(key_value == J11_PRES) //按键 下
            {
                uart_buf[1] = 0x04;
                USART_OUT(USART1, (uint8_t*)uart_buf,3);
            }
            else if(key_value == J7_PRES) //按键 上
            {
                uart_buf[1] = 0x05;
                USART_OUT(USART1, (uint8_t*)uart_buf,3);
            }
        }
    }
}
```

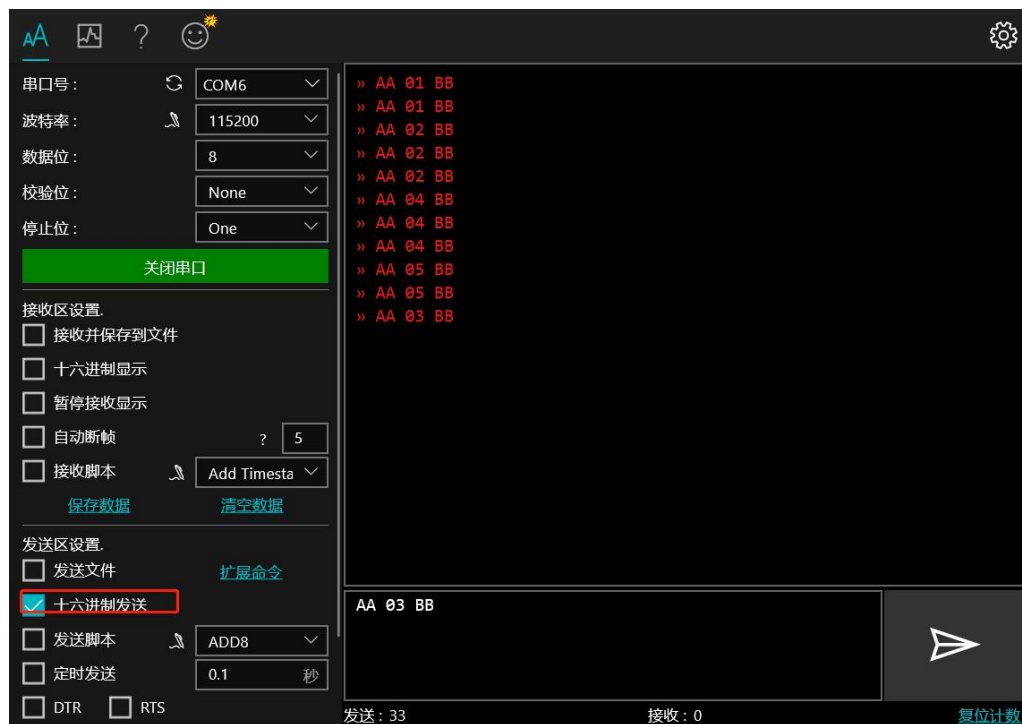


5. 下载验证

将【物理按键_三个键值.pix】或者【物理按键_五个键值.pix】编译后下载到 2.4 寸显示屏，即可看到如下界面：



(1)调试时先将屏的串口连接到电脑，用电脑端的串口助手模拟给屏发键值，确认屏能正确处理键值：



(2)再将 STM32 单片机的键值输出到电脑端的串口助手上，确认 STM32 单片机发出的键值是正确的。

(3)最后将 STM32 单片机的串口接到屏的串口上，可以验证到，上、下、左、右、确认键都是正常的

