

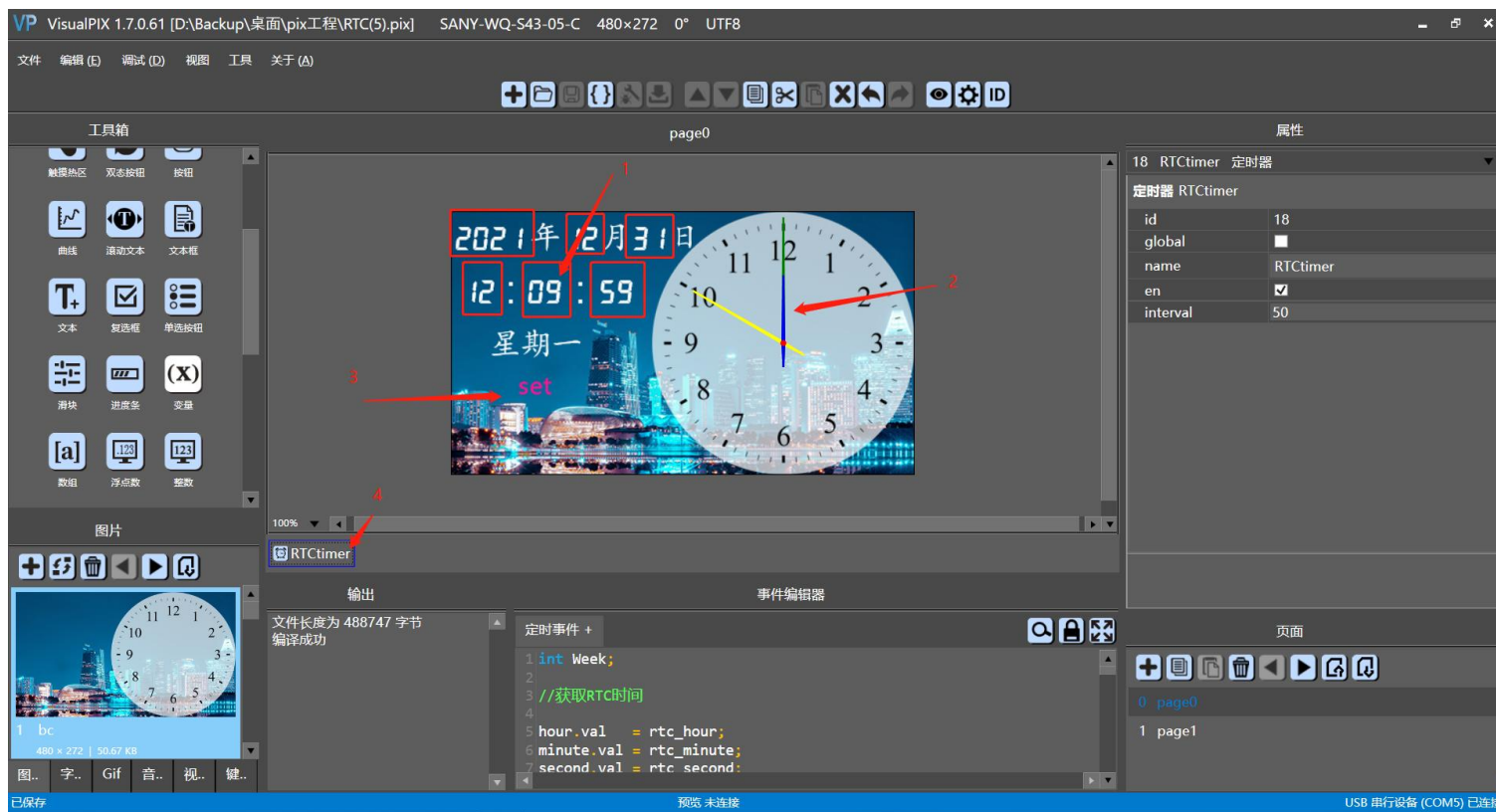


实验 9 RTC 显示和设置

1. 实验目的

使用串口自带的 RTC 功能，显示时间，设置时间。。

2. 页面设计



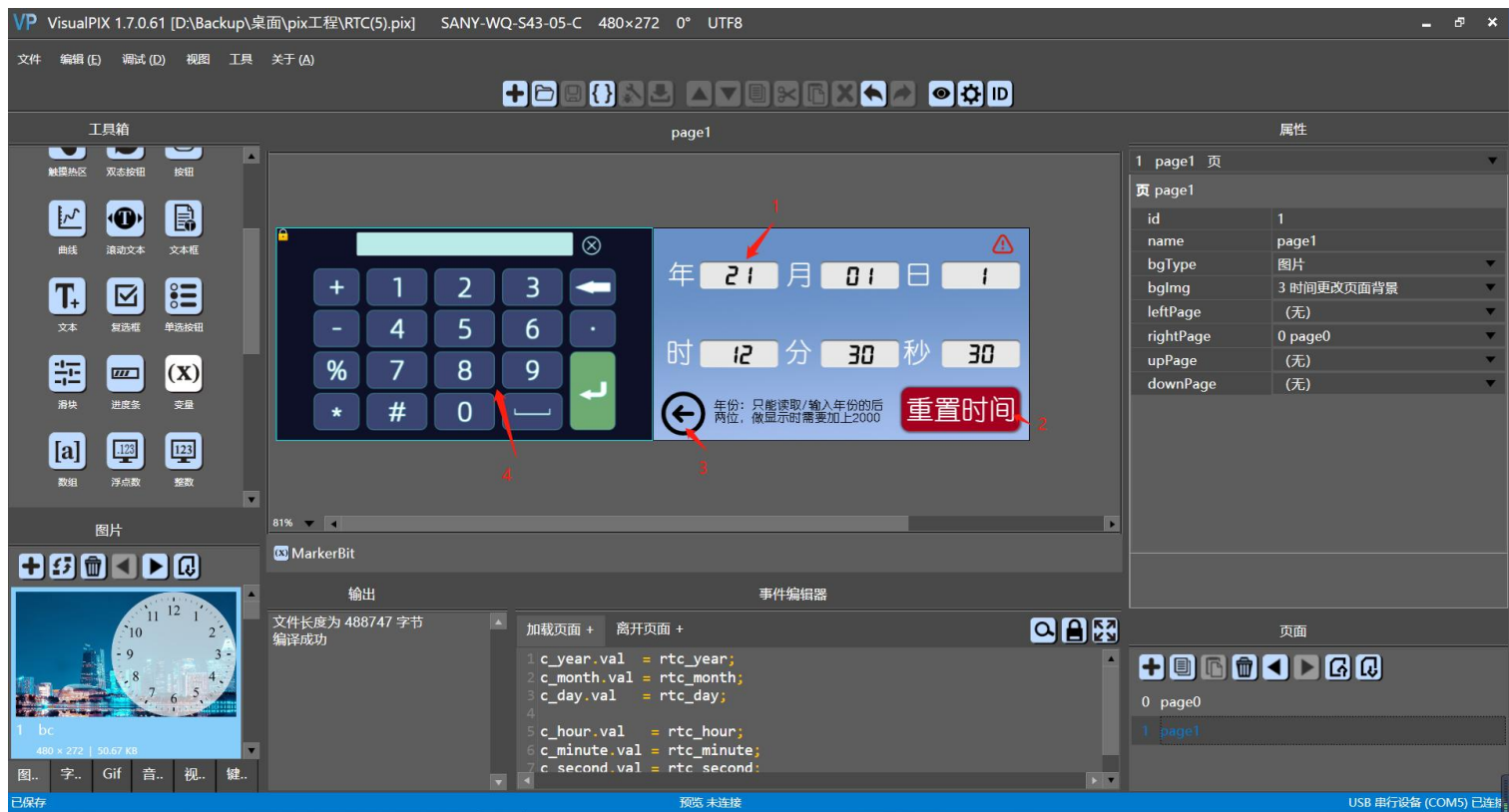
(1) 整数控件，显示年月日时分秒

(2) 指针控件，显示时间

(3) 按钮，翻页到 page1

(4) 定时器控件，间隔 1 秒去读取 RTC 值，并赋值给控件显示

选中控件，可以在右侧属性栏查看控件的各个属性和事件代码。详细属性说明可参考 第四章 控件的介绍和用法。



- 1、整数字件，显示年月日时分秒
- 2、按钮，设置 RTC 参数，返回 page0
- 3、按钮，返回 page0
- 4、键盘控件，用于设置时间

选中控件，可以在右侧属性栏查看控件的各个属性和事件代码。详细属性说明可参考 第四章 控件的介绍和用法。

3.串口屏脚本处理

Page0 的定时器 RTCTimer 用于间隔 1 秒获取 RTC 的值，并赋值给控件显示。实际使用时，可能只需要简单的显示时间数字即可，这里为了演示，通过 RTC 时间值，计算指针的角度，实现表盘显示时间。

选中定时器控件，可以看到脚本代码如下：

```
int Week;

//获取 RTC 时间
hour.val = rtc_hour;
minute.val = rtc_minute;
second.val = rtc_second;
year.val = 2000 + rtc_year; //年份偏移是 2000，显示时需要加上 2000
month.val = rtc_month;
date.val = rtc_day;
Week = rtc_week;

//根据时间，计算指针的位置
if(hour.val == 0 || hour.val == 12)
{
```



```
    Phour.angle = 0 + 90 + minute.val/10*6-6;
}else if(hour.val == 1 || hour.val == 13)
{
    Phour.angle = 30 + 90 + minute.val/10*6-6;
}else if(hour.val == 2 || hour.val == 14)
{
    Phour.angle = 60 + 90 + minute.val/10*6-6;
}else if(hour.val == 3 || hour.val == 15)
{
    Phour.angle = 90 + 90 + minute.val/10*6-6;
}else if(hour.val == 4 || hour.val == 16)
{
    Phour.angle = 120 + 90 + minute.val/10*6-6;
}else if(hour.val == 5 || hour.val == 17)
{
    Phour.angle = 150 + 90 + minute.val/10*6-6;
}else if(hour.val == 6 || hour.val == 18)
{
    Phour.angle = 180 + 90 + minute.val/10*6-6;
}else if(hour.val == 7 || hour.val == 19)
{
    Phour.angle = 210 + 90 + minute.val/10*6-6;
}else if(hour.val == 8 || hour.val == 20)
{
    Phour.angle = 240 + 90 + minute.val/10*6-6;
}else if(hour.val == 9 || hour.val == 21)
{
    Phour.angle = 270 + 90 + minute.val/10*6-6;
}else if(hour.val == 10 || hour.val == 22)
{
    Phour.angle = 300 + 90 + minute.val/10*6-6;
}else if(hour.val == 11 || hour.val == 23)
{
    Phour.angle = 330 + 90 + minute.val/10*6-6;
}
//时针的角度计算，角度 = 整点角度 + 90 度偏移 + 分钟数在一小时刻度中的细分

Pminute.angle = minute.val*6+90;
Psecond.angle = second.val*6+90;
//分针和秒针的计算

//显示星期
if(Week == 1)
{
    week.txt = "星期一";
}else if(Week == 2)
{
    week.txt = "星期二";
```



```
}else if(Week == 3)
{
    week.txt = "星期三";
}else if(Week == 4)
{
    week.txt = "星期四";
}else if(Week == 5)
{
    week.txt = "星期五";
}else if(Week == 6)
{
    week.txt = "星期六";
}else if(Week == 0)
{
    week.txt = "星期日";
}
```

Page2 的键盘控件，用于手动输入时间，输入完成事件中，会将键盘的输入赋值给整数字控件。选择键盘控件，可以看到脚本代码。

```
if(MarkerBit.val == 1)    //年
{
    c_year.val = stringToInt(keyboard1.txt);
}else if(MarkerBit.val == 2)    //月
{
    c_month.val = stringToInt(keyboard1.txt);
}else if(MarkerBit.val == 3)    //日
{
    c_day.val = stringToInt(keyboard1.txt);
}else if(MarkerBit.val == 4)    //时
{
    c_hour.val = stringToInt(keyboard1.txt);
}else if(MarkerBit.val == 5)    //分
{
    c_minute.val = stringToInt(keyboard1.txt);
}else if(MarkerBit.val == 6)    //秒
{
    c_second.val = stringToInt(keyboard1.txt);
}

keyboard1.txt = "";

keyboard1.x = -480;
keyboard1.y = 0;
```



Page2 的重置时间按钮，主要是设置 RTC 的值，脚本代码如下

```
rtc_year = c_year.val;    //RTC 只能读出后两位数，设置也只能设置后面两位
rtc_month = c_month.val;
rtc_day = c_day.val;

rtc_hour = c_hour.val;
rtc_minute = c_minute.val;
rtc_second = c_second.val;

showPage(0);
```

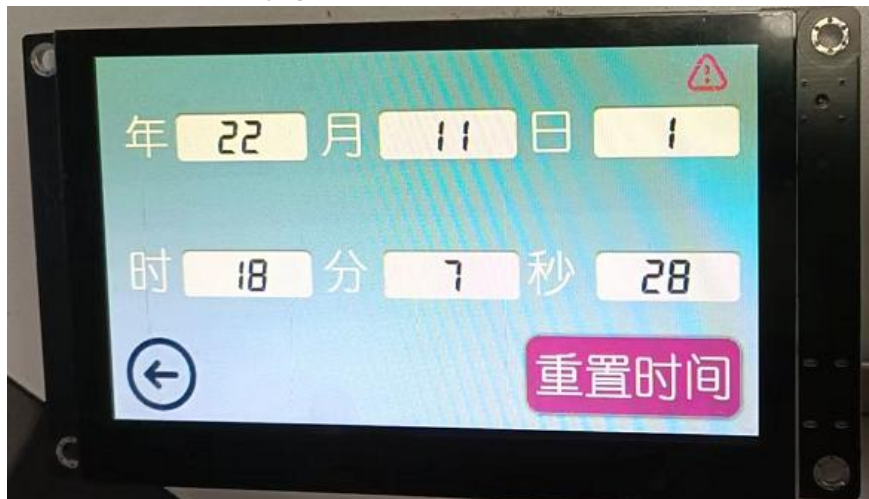
说明：RTC 可以读取和设置 年、月、日、时、分、秒。星期不用设置，设置好年月日后，系统会自己调整是星期几。年 `rtc_year` 的范围是 0~99，显示的时候需要加上 2000 才是正确的年份。

4. 下载验证

编译成功后，点击下载按钮，选择正确的端口号和波特率，下载到串口屏。可以看到如下界面



点击 set 按钮,切换到 page1。点击数字，可以手动设置时间



修改 RTC 时间也能通过串口修改。指令格式可以串口屏系统指令 `sset`，系统指令用法可以参考实验 1。也可以通过自定义协议，在脚本代码里修改 RTC 系统变量的值，自定义协议用法可以参考实验 2。