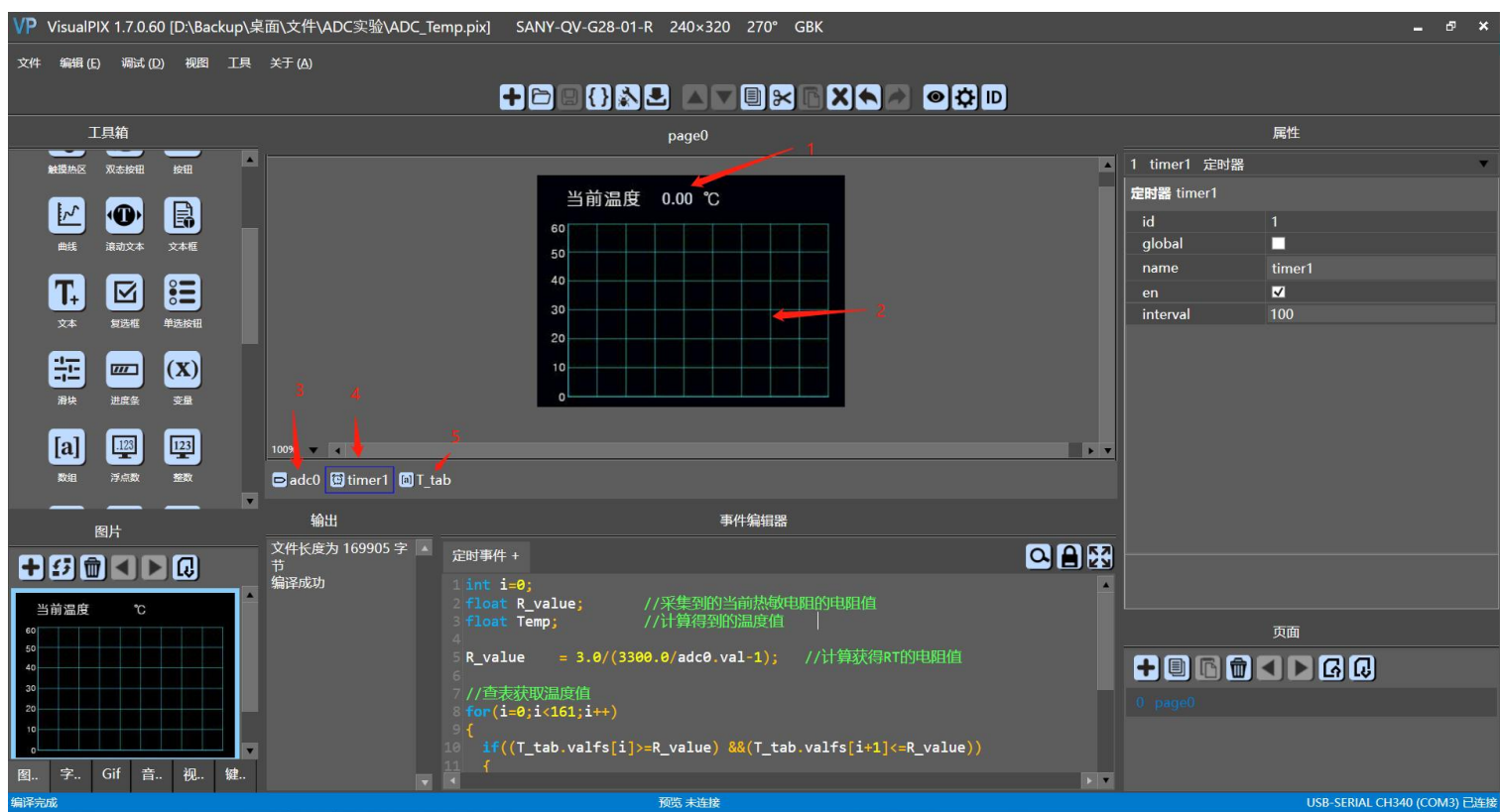


实验 6 ADC 和曲线控件实验

1. 实验目的

使用串口屏自带的 AD 输入引脚采样温度模块的电压，计算出温度，显示当前温度并绘制温度曲线。

2. 页面设计



- (1) 浮点数控件，显示当前温度值
- (2) 曲线控件，显示曲线，温度的变化
- (3) adc 控件，可以获取 IO 电压值
- (4) 定时器，间隔一秒，读取 AD 电压并计算温度，显示曲线
- (5) 数组控件，存放 电阻-温度对应表，计算温度时使用

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

G 系列串口屏支持 3 路 AD 输入。本实验串口屏型号为 SANY-QV-G28-01-R，只用到了 1 路 AD。S 系列串口屏可以支持更多路的 AD 输入。

曲线控件属性配置时，注意通道点数 pCount 的正确设置。Width >= xStep * pCount，否则曲线显示可能达不到预期的效果。

3. 串口屏协议处理

本实验演示 ADC 控件和曲线控件的用法。无通信协议。部分脚本代码如下：



页面的加载事件

//进入这个页面的时候，会执行一次加载页面事件

//温度表初始化

```
int i;
float RT_Tab[161] =
{
    366.4286,344.5753,324.1796,305.1344,287.3413,270.7097,255.1561,240.6035,226.9810,214.2231, // -50℃~-41℃
    202.2693,191.0637,180.5546,170.6944,161.4387,152.7468,144.5807,136.9052,129.6879,122.8985, // -40℃~-31℃
    116.5089,110.4932,104.8272, 99.4883, 94.4558, 89.7101, 85.2332, 81.0082, 77.0194, 73.2523, // -30℃~-21℃
    69.6931, 66.3291, 63.1485, 60.1402, 57.2939, 54.5998, 52.0490, 49.6330, 47.3439, 45.1743, // -20℃~-11℃
    43.1172, 41.1663, 39.3153, 37.5587, 35.8910, 34.3074, 32.8029, 31.3734, 30.0145, 28.7225, // -10℃~-1℃
    27.4936, 26.3245, 25.2119, 24.1527, 23.1442, 22.1835, 21.2682, 20.3959, 19.5644, 18.7714, // 0℃~ 9℃
    18.0151, 17.2935, 16.6048, 15.9475, 15.3198, 14.7203, 14.1475, 13.6003, 13.0772, 12.5771, // 10℃~ 19℃
    12.0988, 11.6413, 11.2037, 10.7848, 10.3839, 10.0000, 9.6324, 9.2802, 8.9428, 8.6195, // 20℃~ 29℃
    8.3096, 8.0124, 7.7275, 7.4541, 7.1919, 6.9403, 6.6987, 6.4669, 6.2442, 6.0304, // 30℃~ 39℃
    5.8250, 5.6276, 5.4380, 5.2557, 5.0804, 4.9119, 4.7498, 4.5939, 4.4439, 4.2995, // 40℃~ 49℃
    4.1605, 4.0268, 3.8980, 3.7739, 3.6544, 3.5393, 3.4284, 3.3215, 3.2185, 3.1191, // 50℃~ 59℃
    3.0234, 2.9310, 2.8419, 2.7559, 2.6729, 2.5929, 2.5156, 2.4410, 2.3690, 2.2994, // 60℃~ 69℃
    2.2322, 2.1673, 2.1046, 2.0440, 1.9854, 1.9288, 1.8740, 1.8211, 1.7699, 1.7204, // 70℃~ 79℃
    1.6725, 1.6262, 1.5813, 1.5379, 1.4959, 1.4553, 1.4159, 1.3778, 1.3408, 1.3051, // 80℃~ 89℃
    1.2704, 1.2368, 1.2043, 1.1728, 1.1422, 1.1126, 1.0839, 1.0560, 1.0290, 1.0028, // 90℃~ 99℃
    0.9774, 0.9527, 0.9288, 0.9055, 0.8830, 0.8611, 0.8399, 0.8193, 0.7992, 0.7798, // 100℃~ 109℃
    0.7609 // 110℃
};

//温度对应表，初始化
for(i=0;i<161;i++)
{
    T_tab.valfs[i] = RT_Tab[i];
}
```

页面加载事件，仅仅在进入页面的时候会执行脚本代码。本实验只有 1 页，页面加载事件只在上电的时候执行一次，所以可以在这里做一些初始化，这里是初始化 电阻-温度表的数组。

定时器事件

```
int i=0;
float R_value; //采集到的当前热敏电阻的电阻值
float Temp; //计算得到的温度值

R_value = 3.0/(3300.0/adc0.val-1); //计算获得 RT 的电阻值①

//查表获取温度值
for(i=0;i<161;i++)
{
    if((T_tab.valfs[i]>=R_value) &&(T_tab.valfs[i+1]<=R_value))
    {
        Temp = i + (R_value - T_tab.valfs[i+1])/(T_tab.valfs[i] - T_tab.valfs[i-1]) - 50;
```



```
}  
}  
  
numf3.valf = Temp; //显示当前温度  
curvedLine2.ch1Val = Temp; //曲线控件添加一个点②
```

①adc0.val 是 AD 引脚的电压值，单位 mV。这里是计算 RT 电阻值

②曲线控件添加一个点，注意曲线控件的数值类型和赋值的类型应该一致。

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。可以看到显示如下显示：

