

实验 11 扩展 IO 功能示例

1. 实验目的

本实验演示了扩展 IO 的相关功能及使用方法。扩展 IO 的功能有普通输入、普通输出、中断输入、AD 输入、PWM 输出和捕获输入的功能。

G 系列串口屏(3.5 寸及以下)支持 6 个扩展 IO，S 系列串口屏(4.3 寸及以上)最多可支持 25 个扩展 IO。

每个 IO 的功能不一样，项目设计的时候，需要提前了解到要用到的 IO 是否支持所需功能，比如如有的 IO 能支持 AD 输入，有的能支持 PWM 输出，具体的 IO 功能介绍参考第七章。

本实验基于 4.3 寸 480*272 分辨率的串口屏，若要验证其他尺寸的串口屏，可在临时修改项目型号(文件->编辑项目)，或者参考本实验重新设计工程。

2. 页面设计

本实验一共 6 个页面，分页演示了不同 IO 功能的用法。



(1)home 页面，一共 5 个按钮，点击对应的按钮可切页到其他页面。



(2) GPIO 页面。

3 个 IO 控件，对应的引脚号是 io7,io8,io9



- 1 个定时器控件，定时 1 秒去读取 io8 引脚的电平状态
- 3 个文本控件，分别显示输出、输入、中断输入的 IO 引脚号
- 4 个整数字控件，分别显示输出值、输入值、中断输入值和中断次数
- 1 个双态按钮控件，控制输入 IO 的电平，弹起是 1
- 1 个按钮控件，点击返回 home 页面



(3) AD 页面

- 1 个 ADC 控件，引脚号 io6
- 1 个定时器控件，定时 1 秒去读取 adc 控件的采样值
- 1 个浮点数字控件，用于显示电压值
- 1 个按钮控件，点击返回 home 页面



(4) PWM 页面

- 1 个 PWM 控件，引脚号 io0
- 1 个键盘控件，用于输入频率和占空比
- 2 个整数字控件，分别显示频率和占空比
- 1 个双态按钮控件，控制输出使能，按下使能，弹起不使能
- 1 个按钮控件，点击返回 home 页面



(5) CAPTURE 页面

- 1 个 Capture 控件，引脚号 io3
- 1 个定时器控件，定时 1 秒去读取 Capture 控件的频率、占空比和脉冲个数
- 3 个整数字控件，分别显示频率、占空比、脉冲个数
- 1 个按钮控件，点击返回 home 页面



(6) ExUART 页面

- 1 个扩展协议解析器控件，默认引脚号 io1,io2
- 2 个文本控件，分别显示接收区和发送区的文本
- 1 个按钮控件，点击“发送”后，扩展串口将发送区的字符串发出去
- 1 个按钮控件，点击返回 home 页面



注意：在页面设计时，提前分配好 IO 的功能，如果同一个 IO 引脚被多次绑定，编译的时候将会报错。
错误提示类型举例：



3. 串口屏脚本处理

本实验的脚本处理很简单，选中控件可以查看相关的脚本。IO 相关控件属性的读和写，列举几个关键的说明。

(1) GPIO 页面

输出电平控制

```
io7.val = 0;  
io7.val = 1;
```

输入电平读取

```
io8num.val = io8.val; //定时 1 秒，读取 io8 的电平状态，0 或者 1
```

IO 中断电平读取和中断处理(num9 显示累加)

```
io9num.val = io9.val;  
num9.val++;
```

(2) AD 页面

读取电压值



```
num1.val = adc.val; //定时 1 秒，读取 IO6 引脚上的模拟电压值
```

(3) PWM 页面

根据键盘输入值设置频率和占空比

```
int val;
val = stringToInt(keyboard4.txt); //将键盘的字符串转换成整数
if(var6.val == 0)
{
    if((val >=1)&&(val <= 1000000)) //频率范围 1Hz~1MHz
    {
        num_freq.val = val;
        pwm3.frequency = num_freq.val; //设置 PWM 输出频率
    }
}
else
{
    if((val >=0)&&(val <= 100)) //占空比范围 0~100
    {
        num_duty.val = stringToInt(keyboard4.txt);
        pwm3.dutyRatio = num_duty.val; //设置 PWM 输出占空比
    }
}
keyboard4.x = -500; //隐藏键盘
```

设置 PWM 输出使能

```
pwm3.enable = 1; //PWM 输出使能
tgbtn1.txt = "enable";
```

设置 PWM 输出不使能

```
pwm3.enable = 0; //PWM 输出不使能
tgbtn1.txt = "disable";
```

(4) CAPTURE 页面

获取捕获参数

```
//定时器去读输入捕获控件的属性，频率、占空比、脉冲个数
num_freq.val = Capture4.frequency;
num_duty.val = Capture4.dutyRatio;
num_cnt.val = Capture4.pulseCount;
```

(5) ExUART 页面

字符串发送

```
//扩展串口发送函数;
//更多的操作函数访问方式为，输入控件名称，然后输入'.', 即 expt14.fun
expt14.uartSend(text4.txt);
```

接收并显示

```
//将收到的串口数组，转换为字符串并赋值给文本显示
text3.txt = bytesToAscii(expt14.rxBuf, 0, expt14.rxBufLen);
```

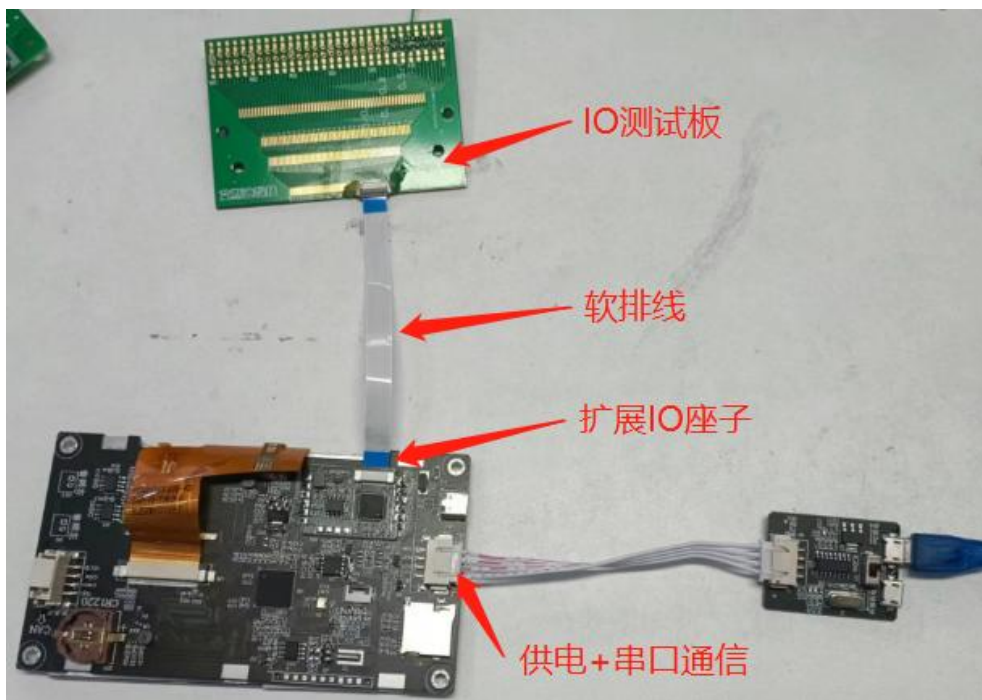


4. 下载验证

将编译后下载到 4.3 寸带显示屏(带扩展 IO 模块), 即可看到如下界面:



为了测试扩展 IO 的功能, 我们需要将扩展 IO 通过软排线接到 IO 测试板上, 如下图:



用万用表、示波器检测 IO 测试板即可。

(1) GPIO 测试。

触摸控制 GPIO 页面双态按钮的状态, 用万用表测试 io7 的输出电压变化, 0V 或者 3.3V;

给 io8 接入 3.3V 或者 GND, 观察显示屏上 io8 的电平值

给 io8 接入 3.3V 或者 GND, 观察显示屏上 io9 的电平值和中断次数

(2) AD 测试。

给 io6 引脚接入 0~3.3V 模拟电压。观察显示屏上 AD 采用出来的电压

(3) PWM 测试。

点击输出使能, 设置 pwm 输出, 可选修改频率和占空比, 用过示波器检测 io0 引脚上的波形。

(4) CAPTURE 测试

将 io0 输出的 pwm 波形接入 io3, 观察显示屏上捕获到的频率、占空比和脉冲个数

(5) ExUART 测试

将 io1 接入 TTL 工具的 rx, io2 接入 TTL 工具的 tx, 然后将 TTL 工具接入电脑, 用串口助手收发测试。测试接线如下: