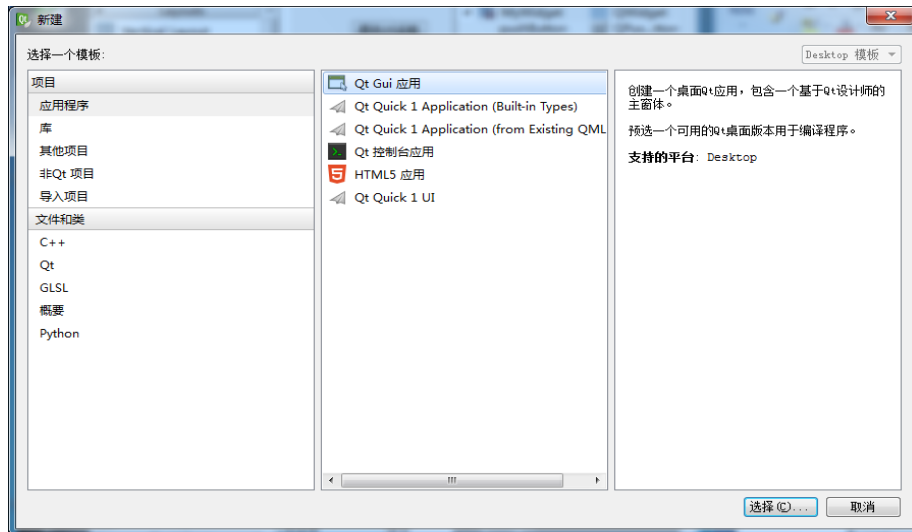


控制 LED 灯实验

1. 新建工程

新建一个 Qt 的 GUI 工程，如图



选择工程路径和工程名为 qt-led:

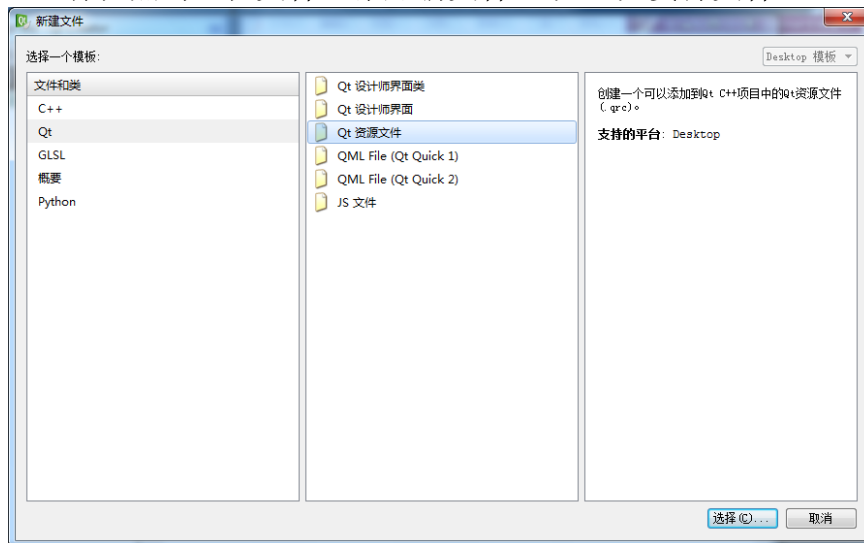


类信息选择：类名 MyLed 基类 QWidget

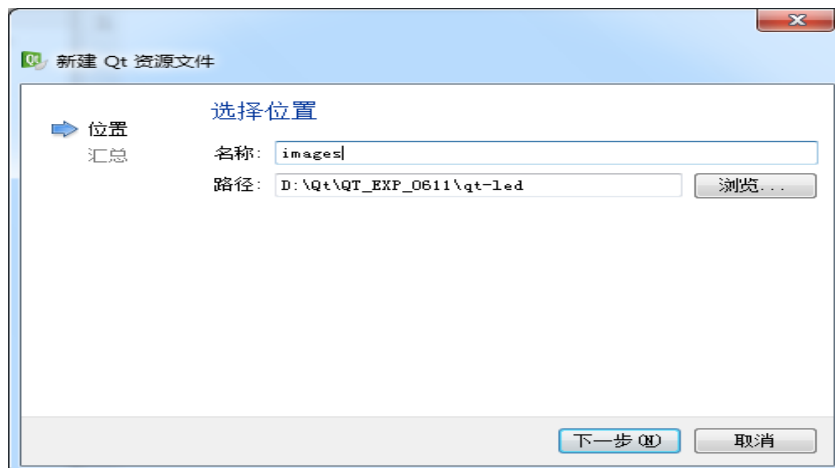


2. 添加资源文件（图像）

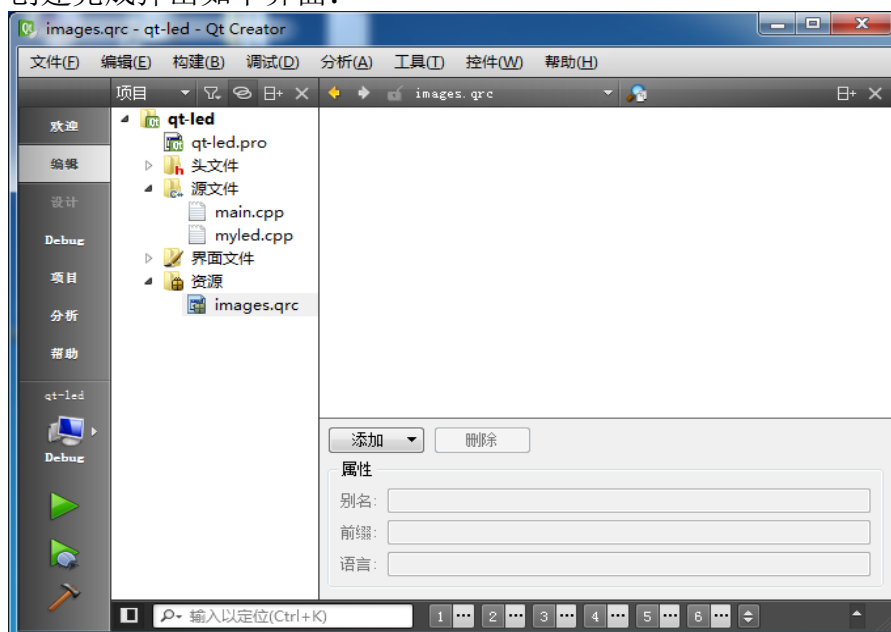
有键点击工程文件->添加新文件->Qt->Qt 资源文件



指定保存位置 及文件名 images，如下示：



创建完成弹出如下界面：



点击下方的添加→添加前缀，将默认值/new/prefix1 改为/，如下：

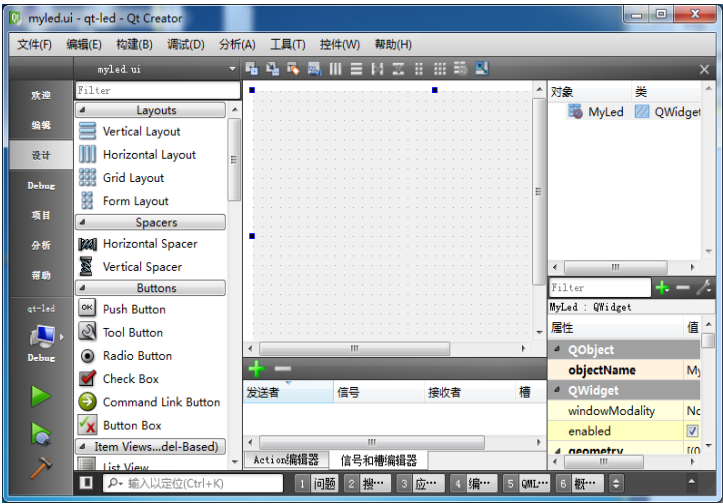


点击下方的添加→添加文件，选择 images 文件下的所有文件，如下：



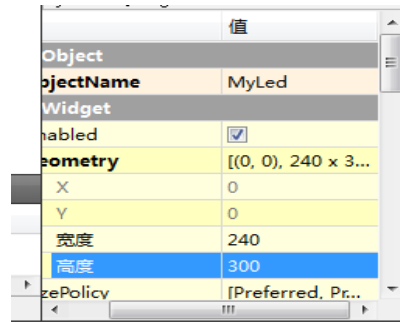
添加完成后，如下显示

：“ctrl+s”保存该文件

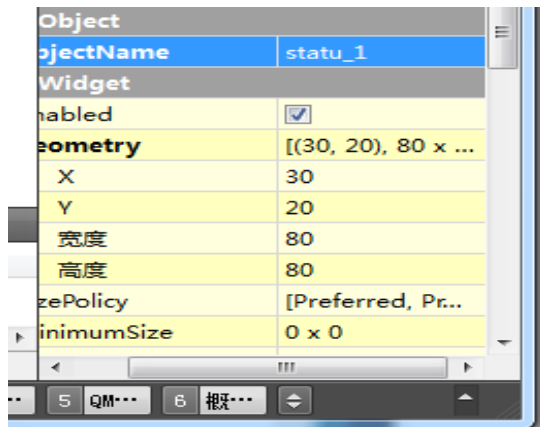


3. 编辑 UI 界面

双击 myled.ui 文件进入界面设计器

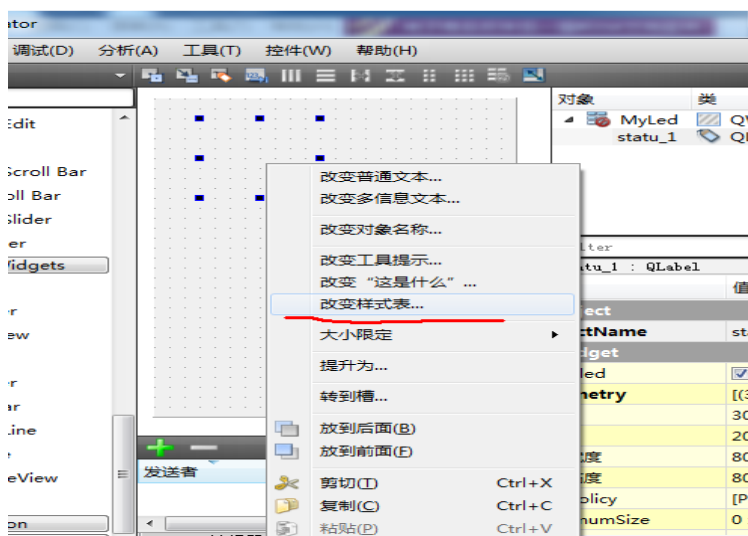


选中设计界面，设置长宽为： 240x300 ；



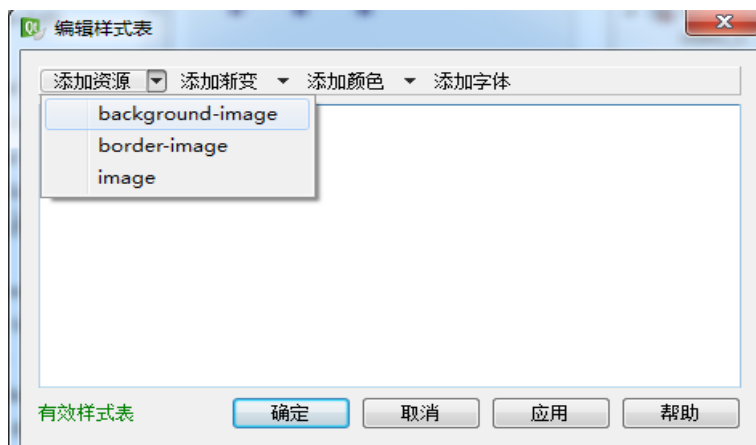
添加标签，文本为：led 灯控制实例，objectName 为 label；

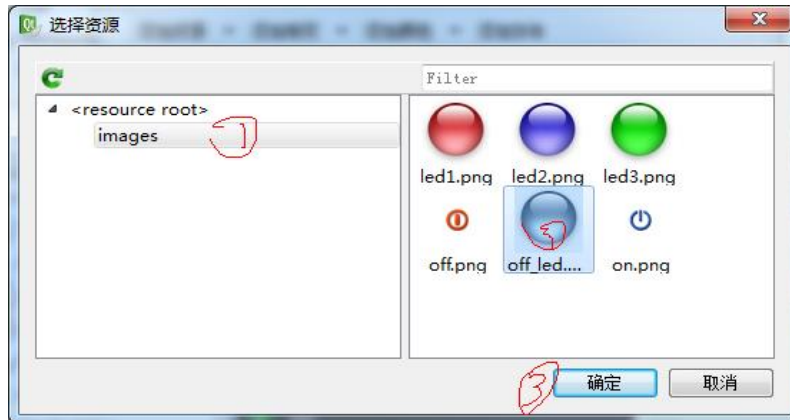
添加标签，清空文本，objectName 为 statu_1，设置大小为 80*80，如左所示：



为标签添加背景图片，右键点击标签，选择 ChangestyleSheet...，如下所示：

在弹出的对话框中选择背景图片

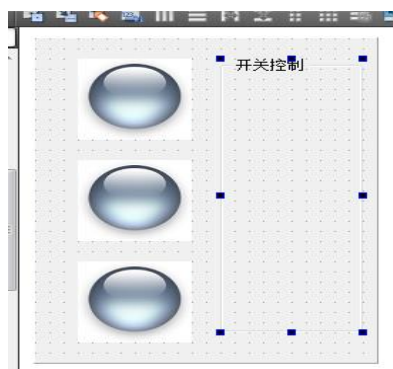
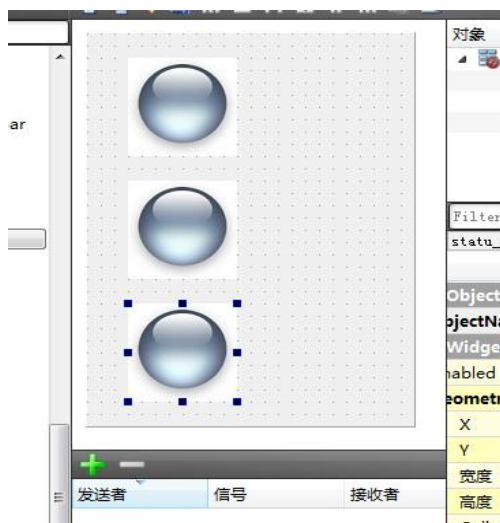




添加完成以后，标签背景已经改变了，如下：

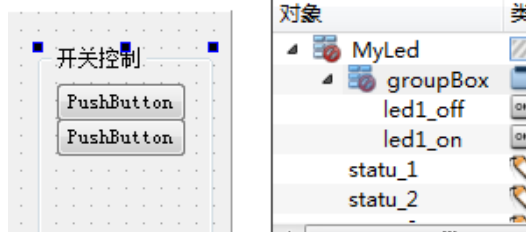


按照相同的方法实现标签 statu_2 和 statu_3，如下图：



添加一个 GroupBox 控件， 文本：开关控制 ，高 270，宽 100，摆放在界面的右侧，如下：

添加一个 QPushButton 按钮控件，文本为：点亮，objectName 为 led1_on, 点击按钮，在右侧属性栏中找到 icon 项，添加图标为 on.png，添加后如下：



添加 pushbutton，并改名字

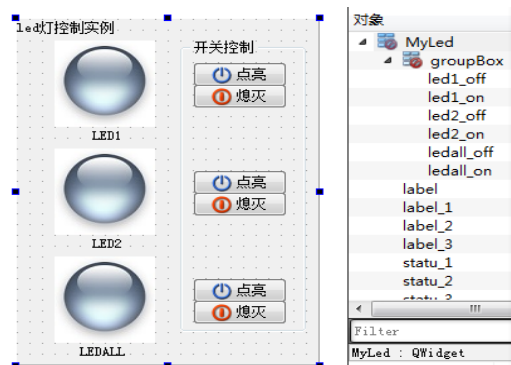
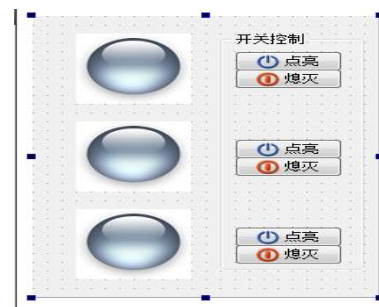


更改按键显示文本为： 点亮和熄灭

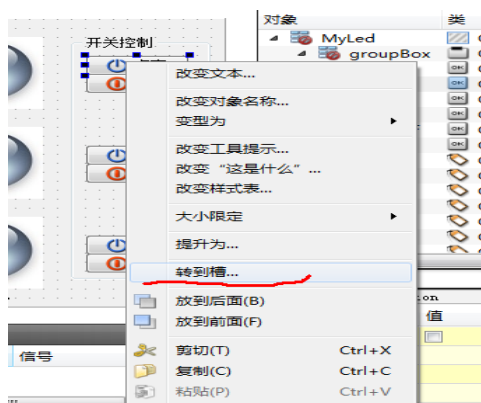


添加上 icon 选项后

按照相同的方法添加按钮控件，分别命名为 led2_on led2_off ledall_on ledall_off，添加完如下：

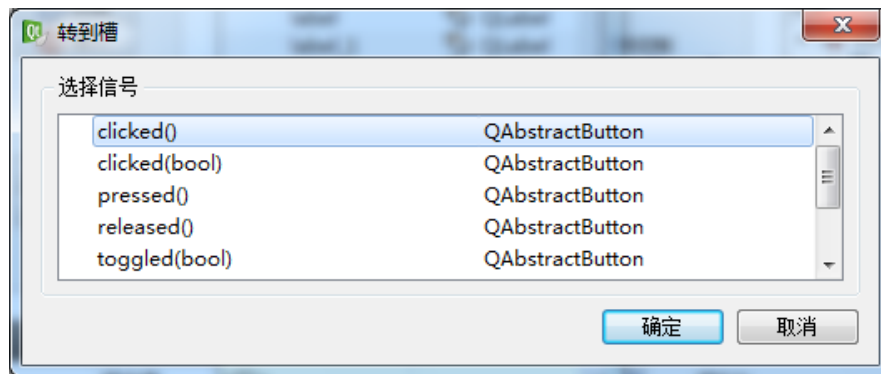


在每个图标下面分别添加一个标签，起名为 label_1 label_2 label_3, 如下：



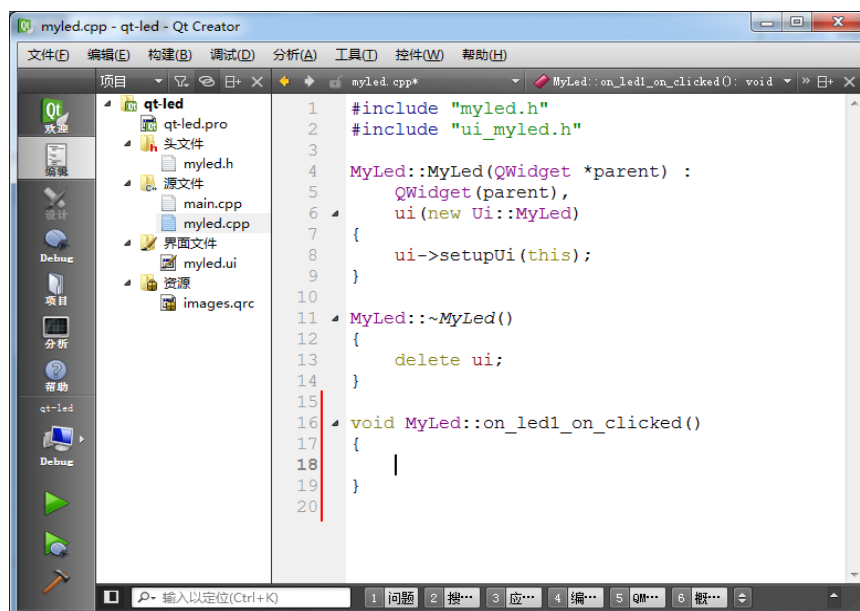
完成界面的设计，保存 UI 文件，接下来进行槽函数的构建：
选中名为 led1_on 的按钮，右击：→ 转到槽

在弹出的对话框选择要连接的信号：



这时自动切换到编辑代码模式，实现槽函数的代码：

（这种建立信号和槽的机制叫做自动关联，把关联函数整合到槽命名中，注意要使用 Qt 部件已经提供的信号。在该处槽函数的名字由字符“on”和发射信号的部件对象名还有信号名组成）



添加如下代码到槽函数：

```
ioctl(fd, 1, 0);  
ui->statu_1->setStyleSheet(QString::fromUtf8("background-image:  
url(/images/led1.png);"));
```

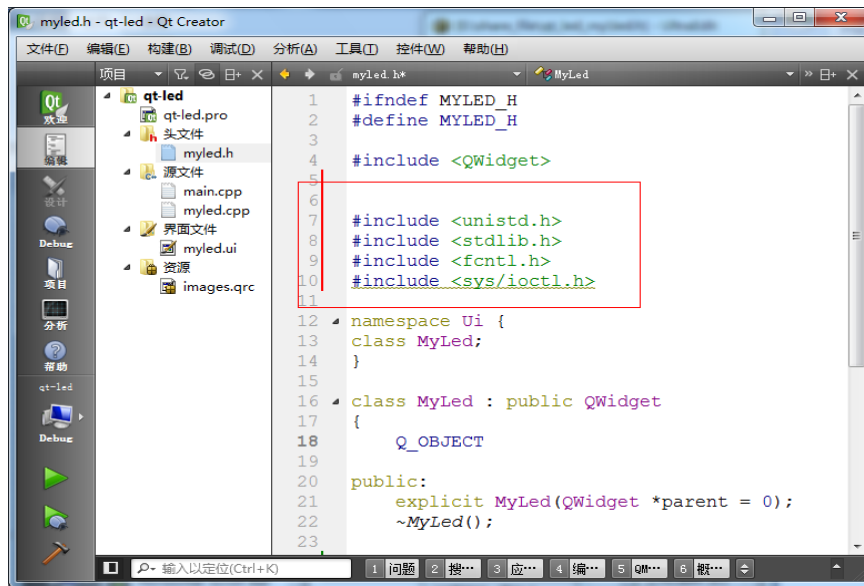
```
20 void MyLed::on_led1_on_clicked()  
21 {  
22     ioctl(fd, 1, 0);  
23     ui->statu_1->setStyleSheet(QString::fromUtf8("b:  
24 }
```

注意槽名字

按照同样的方法给其他五个按钮控件添加和实现槽函数：

过程略

添加完成后在 myled.h 中添加如下语句：



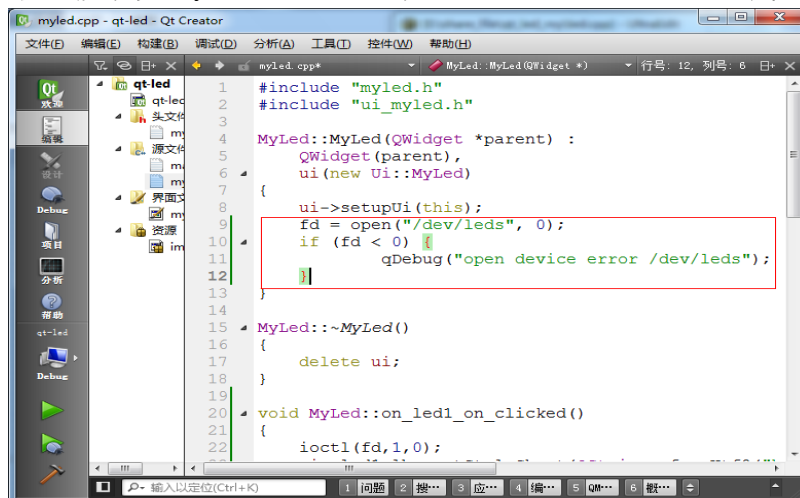
再在类声明中添加私有数据成员 fd，如下：

```

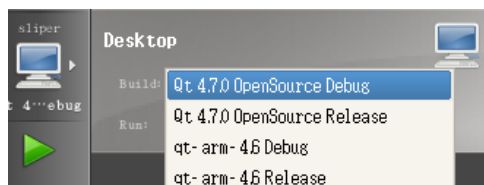
28     void on_led2_on_clicked();
29     void on_ledall_on_clicked();
30     void on_ledall_off_clicked();
31 private:
32     Ui::MyLed *ui;
33     int fd;
34 };
35
36 #endif // MYLED_H

```

最后修改类 MyLed 的构造函数，添加打开 led 设备的代码：



完成以上编辑工作后，选择 qt-arm-4.6 Debug 编译器，重新编译，



然后在 /root/qtproject 中找到相应的工程文件夹 qt-led-build-desktop 下的可执行文件 qt-led，拷贝到开发板中，先加载以前写过的 led 驱动，然后执行 ./qt-led -qws