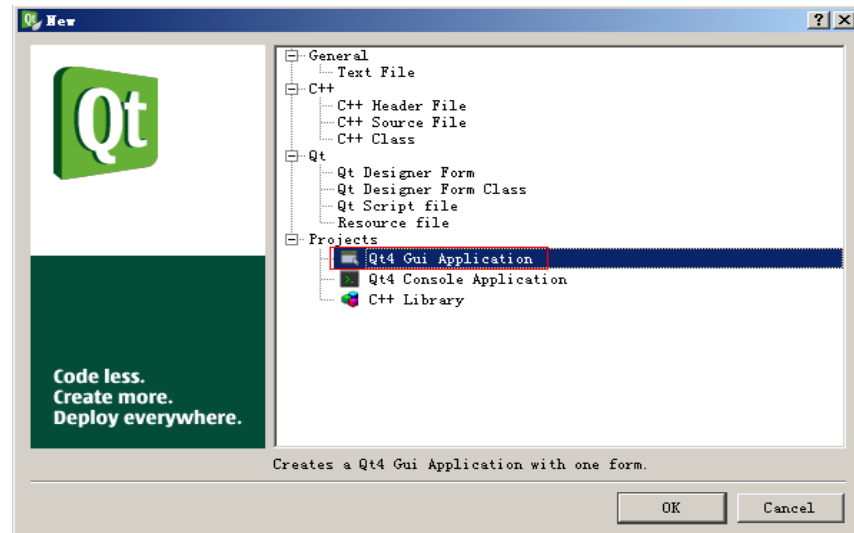
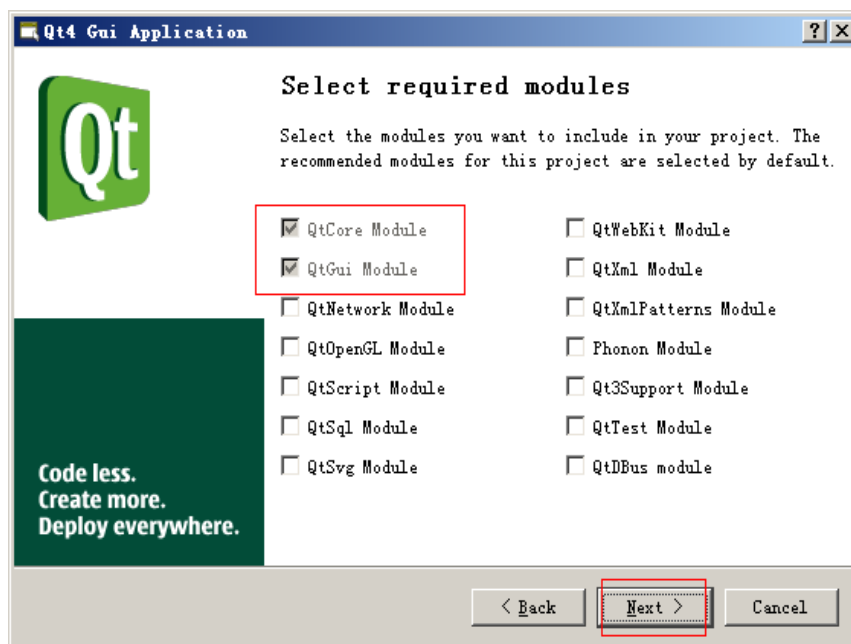
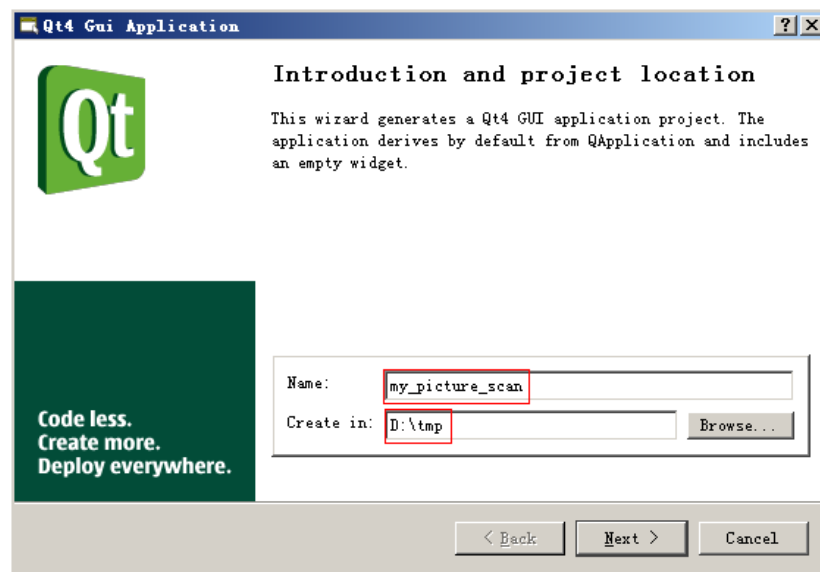


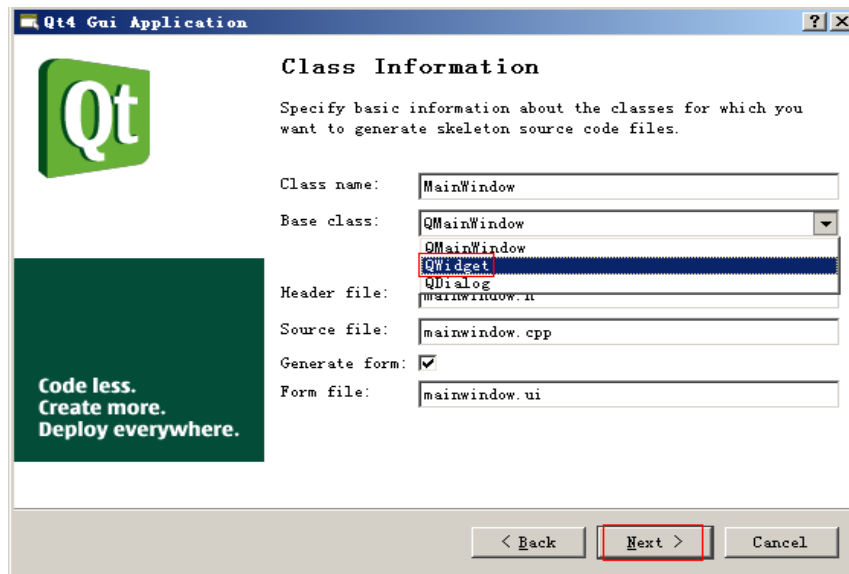
首先打开
QtCreator, 然
后选择
File→New,
出现如图所
示的界面:



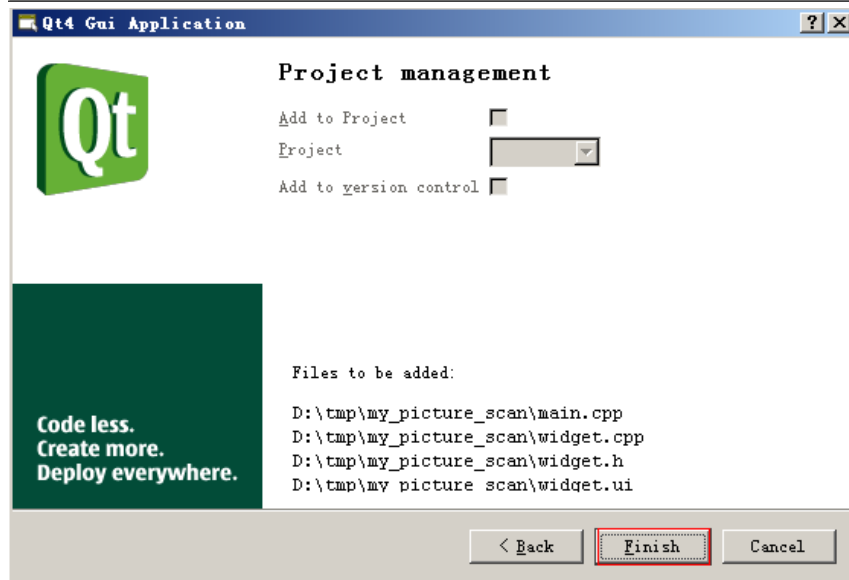
选择 Qt4 Gui
Application, 点
击“OK”, 出现
如下图所示:



选择路
径, 然后
创建项目
名, 点击
“Next”,
出现下
图:



采用默认
设置进入
下一步如
图：



在 Base
class 里面
选择
QWidget,
然后点击
“Next”,
如图：

直接点击 Finish 即可。

此时工程已经创建完毕，双击“widget.ui”，在中间的 Form 里面可以布局你的一些你的控件，然后设计界面。如图所示：

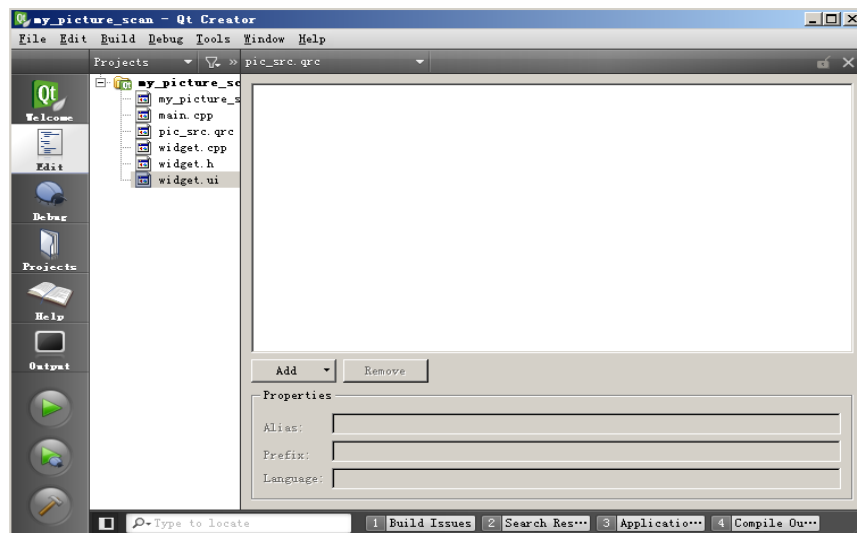


其中包含 8 个 PushButton 控件和一个 Scroll Area 控件，各属性设置如下：

	控件名称	对象名	其他属性
1	打开	btn_open	设置控件大小为 65*25, 并在 icon 里添加打开图标
2	开始	btn_start	设置控件大小为 65*25, 并在 icon 里添加开始图标
3	暂停	btn_stop	设置控件大小为 65*25, 并在 icon 里添加暂停图标
4	上一张	btn_prev	设置控件大小为 65*25, 并在 icon 里添加上一张图标
5	下一张	btn_next	设置控件大小为 65*25, 并在 icon 里添加下一张图标
6	放大	btn_up	设置控件大小为 65*25, 并在 icon 里添加放大图标
7	缩小	btn_down	设置控件大小为 65*25, 并在 icon 里添加缩小图标
8	还原	btn_normal	设置控件大小为 65*25, 并在 icon 里添加还原图标
9	图片显示区域	scrollArea	无

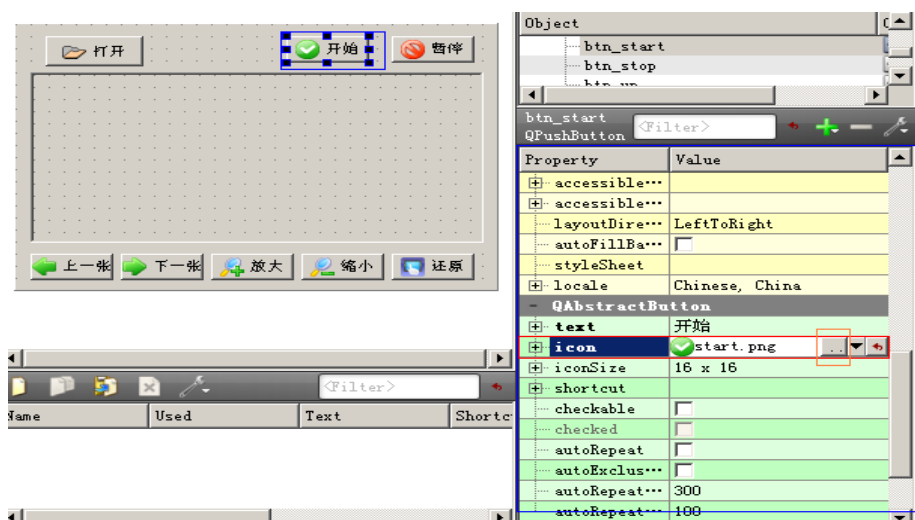
如何给控件等添加小图标呢？操作步骤如下：

点击 File→New→在 Qt 里面选择 Resource file，点击下一步，输入资源名称然后下一步再下一步。出现如图：



点击“Add”，先选择“Add Prefix”，然后在 Prefix 里面只写为“/”即可。然后再点击“Add”，选择“Add Files”选择你的图标所在的位置（应该在你的工程目录下新建一个文件 img_src 用来保存这些图标。然后把所有的这些图标加进来保存即可。）

添加了图标之后，然后对于一个 QPushButton，在它的属性选项里面，设置 icon 点击前面的有两个小点的图标，然后点击刷新按钮，就可以看到之前加载的图标了。



做完界面之后，在每一个按钮控件上面点击右键，选择“go to slot...”。下面是各个模块的具体实现：(黑色表示原来就有，红色表示自己添加，蓝色表示用 go to slot 自动生成的代码)

对于 **main.cpp**: //什么都不用修改

```
#include <QtGui/QApplication>
#include "widget.h"
```

```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    Widget w;
    w.show();
    return a.exec();
}
```

对于 **widget.h**

```
#ifndef WIDGET_H
#define WIDGET_H
```

```
#include <QtGui/QWidget>
#include <QFileDialog>
#include <QString>
#include <QTimer>
#include <QLabel>
```

```
namespace Ui
{
    class WidgetClass;
}
```

```
class Widget : public QWidget
{
    Q_OBJECT
```

```
public:
    Widget(QWidget *parent = 0);
    ~Widget();
```

```
private:
    Ui::WidgetClass *ui;
    QTimer *timer; //定时器
    QPixmap pix; //图像映射
    QLabel *label; //标签
```

```
private slots:
```

```

void on_btn_normal_clicked();
void on_btn_down_clicked();
void on_btn_up_clicked();
void on_btn_next_clicked();
void on_btn_prev_clicked();
void on_btn_stop_clicked();
void on_btn_start_clicked();
void on_btn_open_clicked();

void pic_scan();
};

#endif // WIDGET_H

对于 widget.cpp:

#include "widget.h"
#include "ui_widget.h"

static QStringList::Iterator it;    /* 迭代器保存所有打开加载的图片 */
static QImage image[100];         /* 将图片保存到数组里面 */
static int iCount = 0;             /* 总共有多少张图片 */
static int i = 0;                 /* 当前是第几张图片 */

Widget::Widget(QWidget *parent) : QWidget(parent), ui(new Ui::WidgetClass)
{
    ui->setupUi(this);
    label = new QLabel();          /* 创建显示图片的文本标签 */
    ui->scrollArea->setWidget(label); /* 将显示标签放入滚动条中 */
    ui->scrollArea->setAlignment(Qt::AlignCenter); /* 显示位置，对齐方式 */

    timer = new QTimer;            /* 创建定时器，并与指定的事件关联起来，当时间到了就调用
pic_scan 函数 */
    connect(timer, SIGNAL(timeout()), this, SLOT(pic_scan()));

    label->setAlignment(Qt::AlignCenter);
}

Widget::~Widget()
{
    delete ui;
}

对于每个控件按钮的槽函数：
void Widget::on_btn_open_clicked() /* 按下了打开按钮 */
{

```

```

        timer->stop(); /* 先将定时器暂停 */
        QStringList fileNameList = QFileDialog::getOpenFileNames(this,"file open","d:/","Picture
Files(*.png;*.bmp);;All Files(*)");
        //QStringList fileNameList = QFileDialog::getOpenFileNames(this,"file open","", "Picture
File(*.png)"); //在开发板上运行
        for(it = fileNameList.begin(); it != fileNameList.end(); it++) //返回的文件名 list 是迭代器
        {
            image[iCount] = *it; /* 把打开的图片添加到一个数组里面存起来 */
            iCount ++;
        }
        /* 显示第 i 张图片 */
        pix.load(image[i]);
        label->setPixmap(pix);
    }

void Widget::on_btn_start_clicked() /* 按下了开始按钮 */
{
    timer->start(3000); /* 设置定时器时间间隔 */
}

void Widget::on_btn_stop_clicked() /* 按下了停止按钮 */
{
    timer->stop(); /* 暂停定时器 */
}

void Widget::on_btn_prev_clicked() /* 按下了上一张按钮 */
{
    timer->stop(); /* 暂停定时器 */
    i--; /* 上一张则下标减 1 */
    if(i < 0) /* 如果 i 小于 0 了, 则再从最后一张开始 */
        i = iCount - 1;
    pix.load(image[i]);
    label->setPixmap(pix);
}

void Widget::on_btn_next_clicked() /* 按下了下一张按钮 */
{
    timer->stop(); /* 暂停定时器 */
    i++; /* 下一张则下标加 1 */
    if(i > iCount - 1) /* 如果大于 iCount-1 了, 则再从第一张开始 */
        i = 0;
    pix.load(image[i]);
    label->setPixmap(pix);
}

```

```

void Widget::on_btn_up_clicked() /* 按下了放大按钮 */
{
    timer->stop(); /* 暂停定时器 */
    double x = pix.width(); /* 获得图片的宽 */
    double y = pix.height(); /* 获得图片的高 */
    pix.load(image[i]);
    pix = pix.scaled(x * 1.2, y * 1.2, Qt::KeepAspectRatio);
    /* 将长宽每次放大 1.2 倍 */
    label->setPixmap(pix);
}

void Widget::on_btn_down_clicked() /* 按下了缩小按钮 */
{
    timer->stop(); /* 暂停定时器 */
    double x = pix.width(); /* 获得图片的宽 */
    double y = pix.height(); /* 获得图片的高 */
    pix.load(image[i]);
    pix = pix.scaled(x / 1.2, y / 1.2, Qt::KeepAspectRatio);
    /* 将长宽每次缩小 1.2 倍 */
    label->setPixmap(pix);
}

void Widget::on_btn_normal_clicked() /* 按下了还原按钮 */
{
    timer->stop(); /* 暂停定时器 */

    /* 重新显示图片即可 */
    pix.load(image[i]);
    label->setPixmap(pix);
}

void Widget::pic_scan() /* 开始浏览图片 */
{
    /* 定时器每隔 3 秒执行 pic_scan 函数，i 从 0 开始，当定时器暂停的时候，i 就保存了
    数组中此时显示图片的下标 */
    pix.load(image[i]);
    label->setPixmap(pix);
    i++;
    if(i > iCount - 1) //如果到达最后一张，再从第一张开始
        i = 0;
}

```

8. 找到可执行文件拷贝到开发板运行：./picture_scan -qws