



嵌入式GUI编程

Version 1.0

西安电子科技大学

本章重点

- 嵌入式GUI开发库（类库、函数库）简介
- Qt编程入门
- 信号和槽机制
- 所见即所得WYSIWYG的GUI设计（Qt Designer）
- Qt程序的编译方法
- Qt开发环境的搭建
- 如何在嵌入式系统上部署Qt运行环境

西安电子科技大学

嵌入式GUI开发库

1. 为什么需要GUI开发库:

- 屏蔽底层显示驱动接口 (FrameBuffer驱动)
- 渲染的实现 (矢量—>栅格)
- 常用界面元素 (窗口、按钮、文本框、滚动条等) 的实现
- 多个界面间的覆盖和刷新问题
- 事件捕获、分发, 事件驱动编程模型
- 所见即所得的开发工具
- 应用程序生命周期管理
- 字体管理、语言管理
- 其他常用功能 (字符串处理、网络通信、多媒体等)
-

3

西安电子科技大学

常用的开源GUI开发库

1. Qt for Embedded Linux
2. MiniGUI
3. Android
4. Microwindows
5. Tiny X

4

西安电子科技大学

Qt 编程入门

1. 一个Hello World例子

Qt的信号和槽机制

1. Qt对C++语言的扩展
2. 一种强大的对象间通信机制
 - 对象间松耦合（互相不知道对方的信息）
 - 对象间的通信关系可以由第三方灵活定义
3. “信号signal”相当于一个对象的“输出管脚”
4. “槽slot”相当于一个对象的“输入管脚”
5. 用connect函数将“信号”和“槽”关联起来
6. 用disconnect函数解除关联
7. 一个信号可以关联多个槽，一个槽也可以关联多个信号

Qt的信号和槽机制

```
bool QObject::connect (const QObject* sender, const char *  
signal,const QObject* receiver, const char * member) [static]
```

```
bool QObject::disconnect (const QObject* sender, const char *  
signal,const QObject* receiver, const char * member) [static]
```

Qt Designer

1. 一个非所见即所得的例子:

Qt程序的编译

1. 先用qmake –project命令生成工程文件
2. 再用qmake命令生成标准的Makefile文件
3. 最后用任意一个c++编译器 (gcc、VS2010等) 编译
4. 上述过程也可由集成开发环境 (IDE) 代劳。