《嵌入式系统》
（第十四讲）

厦门大学信息学院软件工程系 曾文华

2024年12月10日

- 第1章：嵌入式系统概述
- 第2章：ARM处理器和指令集
- 第3章：嵌入式Linux操作系统
- 第4章：嵌入式软件编程技术
- 第5章：开发环境和调试技术
- 第6章：Boot Loader技术
- 第7章：ARM Linux内核
- 第8章：文件系统
- 第9章：设备驱动程序设计基础
- 第10章：字符设备和驱动程序设计
- 第11章：Android操作系统（增加）
- 第12章：块设备和驱动程序设计
- 第13章：网络设备驱动程序开发
- 第14章：嵌入式GUI及应用程序设计

第14章 嵌入式GUI及应用程序设计

- 14.1 嵌入式GUI设计概述
- 14.2 嵌入式GUI体系结构设计
- 14.3 基于主流GUI的应用程序设计

14.1 嵌入式GUI设计概述

• 14.1.1 嵌入式GUI简介

– GUI：Graphical User Interface，图形用户界面。

– 嵌入式GUI设计包括以下3个方面的内容：

① 硬件设计：通过LCD控制器，将LCD显示器与开发板连接起来。

② 驱动程序设计：为LCD设计驱动程序，并移植嵌入式GUI系统，为上层应用程序设计提供图形函数库。

③ 用户界面程序设计：使用嵌入式系统提供的函数库，进行图形化应用程序设计。

– 嵌入式GUI分为以下3大类：

① 与操作系统结合的GUI：这些GUI一般由有操作系统开发实力的大公司开发，如微软的Windows Phone（其前身是Windows CE和Windows Mobile）、苹果公司的iOS等。

② 外挂GUI：这些GUI通常基于操作系统运行，向应用层提供开发接口，如Android、Qt/E、MiniGUI、Microwindows等。

③ 简单GUI：这些GUI通常与应用程序结合在一起，可重用性较差。

• 14.1.2 嵌入式GUI设计需求

– GUI系统需要完成的主要功能是：

① 提供桌面和窗口管理功能。可同时运行多个应用程序，创建多个窗口。可对创建的窗口进行显示、隐藏、移动、改变大小等操作。

② 提供多种窗口组件界面。如光标、菜单、按钮、编辑框、列表框、静态控制框、滚动条、对话框和默认窗口等多种窗口界面对象。

③ 提供图形操作。编写的应用程序能够绘制各种复杂图形，还可以填充任何闭合区域，如绘制直线、圆、曲线、矩形等图形。

④ 支持基本的输入输出硬件设备。能够通过各种输入设备，如鼠标、键盘等对窗口进行控制或输入。

⑤ 提供资源管理的功能。支持当今大多数流行的通用图形格式，如BMP等，支持多字符集和多字体，支持汉字输入法。

- **14.1.3 嵌入式GUI设计原则**

  - 1、可移植性。嵌入式系统发展迅速，嵌入式硬件平台和操作系统的种类繁多、更新速度快、系统特点不一，为了支持在不同的嵌入式平台中运行，嵌入式GUI系统应具备良好的移植性。

  - 2、较高的稳定性和可靠性。嵌入式系统运行环境大都较差，而且一旦崩溃就可能导致无法挽回的严重后果。因此，嵌入式GUI系统要求有较高的稳定性和可靠性。

  - 3、系统开销少。嵌入式系统的硬件资源大都受限，处理器频率较低、RAM和Flash容量较小等。而且在嵌入式系统中，通常还运行着比GUI系统更为重要的系统软件和应用软件。因此，嵌入式GUI系统不能占用过多的系统资源，运行开销要小。

  - 4、较高可配置性。嵌入式GUI系统的可配置性通常包括功能配置、界面特性配置、皮肤和主题配置等方面。不同的嵌入式应用对嵌入式GUI系统配置有不同的要求，因此，嵌入式GUI系统应具有一定的可配置性，从而适应不同系统的需求和不同用户体验的选择。

- # 14.1.4 主流嵌入式GUI简介

  - **1、Qt/E。Qt/Embedded是面向嵌入式系统的Qt版本。 Qt/Embedded是一个C++函数库。是一个多平台的C++图形用户界面应用程序框架，能给用户提供精美的图形用户界面所需要的所有元素。**

  - **2、MiniGUI。MiniGUI是一个自由软件项目，其目标是为基于Linux的实时嵌入式系统提供一个轻量级的图形用户界面支持系统，比较适合工控领域的应用。MiniGUI具有：方便的编程接口、使用了图形抽象层和输入抽象层、多字体和多字符集支持、多线程机制的特点。**

  - **3、MicroWindows。Qt/Embedded是一个开放源码的项目。是一个基于典型客户/服务器体系结构的GUI系统。 Qt/Embedded有三层：最底层是面向图形输出和键盘、鼠标或触摸屏的驱动程序；中间层提供底层硬件的抽象接口，并进行窗口管理；最高层分别提供兼容于X Window和Windows CE的API。**

- 4、Tiny-X。Tiny-X是标准X Window系统的简化版，去掉了许多对设备的检测过程，无须设置显示卡驱动，很容易对各种不同硬件进行移植。其设计目的是为了在小容量内存的环境下运行，非常适合用作嵌入式Linux的GUI系统。

- 5、Android。Android的应用程序都是使用Java来编写的，因此很容易移植到新的硬件平台上，用户可以使用Google提供的SDK平台来设计与开发Android周边应用，此外，Android平台还包括3D图形加速引擎、SQLite支持、Webkit支持等特性。

- 6、Windows CE。是Microsoft针对嵌入式产品的一套模块化设计的操作系统。为用户提供良好的GUI。 Windows CE的基本GUI模块包括：窗口管理模块、COM组件、窗口控制组件。

- 7、Palm。Palm OS是Palm公司研制的专门用于其掌上电脑产品Palm的操作系统。Palm操作系统的特点是简单易用，运行需要的内存与处理器资源较小，速度也很快，Palm操作系统不支持多线程。

- 8、iOS。iOS是以Darwin为基础的，属于类UNIX的商业操作系统。iOS用户界面的创新设计是多点触控。控制方法包括滑动、轻触开关和按键，交互方式包括滑动、轻按、挤压和旋转等。

14.2 嵌入式GUI体系结构设计

- **14.2.1 嵌入式GUI体系结构**
  - 嵌入式GUI体系结构一般都采用分层设计（通常包括：抽象层、核心层、接口层），以便简化整个GUI系统的设计。
  - 分层设计的GUI系统具有清晰的层次结构，层与层之间的接口定义相对简单，可增强整个GUI系统的可靠性和稳定性。
  - 典型的嵌入式GUI体系结构包括：抽象层、图形设备接口、窗口管理、消息管理、内存管理和通信管理等模块。

- **14.2.2 抽象层**

  - 包括操作系统抽象层、硬件输入抽象层、图形输出抽象层。
  - 1、操作系统抽象层：主要用来隔离具体的操作系统。
  - 2、硬件输入抽象层：主要用来实现硬件输入功能。
  - 3、图形输出抽象层：主要用来实现图形输出功能。

- **14.2.3 核心层**

  - 1、消息管理。其主要任务是保证消息能够正常地发送、传递、捕获和处理。大部分GUI系统采用事件和消息驱动机制作为系统的基本通信机制。
  - 2、内存管理。在GUI初始化之初就申请一块连续的共享内存，用链表把此块内存管理起来，避免在应用程序中频繁动态地申请和释放内存时造成大量的内存碎片。
  - 3、窗口管理。负责窗口的分类、窗口树和Z序的管理、窗口剪切域的管理，以及窗口绘制和跟GDI模块的交互等。
  - 4、资源管理。资源是指GUI中所使用到的图片、字体库等。GUI把所有需要用到的图片数据进行预处理，调用时可大大提高效率。
  - 5、定时器管理。根据操作系统时钟，为GUI提供定时服务。
  - 6、图形设备接口。完成点、线、矩形、椭圆、多边形等绘制的基本操作。

- **14.2.4 接口层**

  - 接口层提供各种GUI对象（窗口、控件）的数据结构、应用编程接口以及绘制接口。

  - 数据结构包括各种图形设备接口对象的数据结构，如画笔、画刷、背景、位图、字体等。

  - 接口（应用编程接口、绘制接口）包括设备上下文的操作、图形设备接口对象的操作、坐标系统转换、图形绘制单元的操作等。

  - 所有的接口一般以封装的方式提供，不仅可以提高代码的可重用性，还便于开发人员对已有的窗口或控件对象进行扩展。

14.3 基于主流GUI的应用程序设计

- **14.3.1 MiniGUI开发环境搭建**
  - MiniGUI是一个自由软件项目，其目标是为基于Linux的实时嵌入式系统提供一个轻量级的图形用户界面支持系统，比较适合工控领域的应用。MiniGUI具有：方便的编程接口、使用了图形抽象层和输入抽象层、多字体和多字符集支持、多线程机制的特点。
  - MiniGUI可以运行在Linux系统上，也可以运行在Windows系统上。在Linux系统上搭建MiniGUI开发环境的步骤是：
  - 1、安装MiniGUI相关程序
  - 2、配置MiniGUI环境
  - 3、MiniGUI的使用

- # 14.3.2 基于MiniGUI的应用程序设计

  - MiniGUI是一个典型的GUI图形界面支持系统，GUI应用程序通过监控单击鼠标、按键等输入设备事件，再通过GUI内部处理，把相对应的响应反馈传递到图形的窗口上，通过局部或整体重绘，达到图形界面交互的效果。

  - 1、编程环境介绍
    - MiniGUI完全由C语言编写。

  - 2、MiniGUI框架介绍
    - MiniGUI采用了基于线程的体系结构，并在此基础上，架构起较为完备的消息传递与多窗口处理机制。

  - 3、基础编程
    - MiniGUI编程过程与传统的GUI编程方式十分相似。

• 14.3.3 Android开发环境搭建

- 1、Android开发环境介绍
  - 常用的有JBuilder、Eclipse、NetBeans、Android Studio等。

- 2、环境搭建步骤
  - 教材上介绍的是Eclipse环境搭建步骤。
  - 实验七给出了Android Studio环境的搭建步骤。

- 3、Android SDK介绍
  - Android SDK是Google为Android应用开发者提供的开发工具包，包括文档、API、工具以及相关的例程。
  - Android工具包括：模拟器、调试桥、SD卡工具、编译器、批处理脚本等。
  - API集是Android SDK包中的核心功能组件，它提供了如函数、方法、属性、类库等一系列在用户开发应用过程中所必需的应用编程接口集合。

- **14.3.4 基于Android的应用程序设计**

  - **1、创建Android工程**
    - 在创建Android工程的过程中，Eclipse（Android Studio）与Android SDK为用户自动生成了支持Android与工程所需的相关文件，其中部分在用户开发时需要修改或重写，而另外一部分则不能被修改。

  - **2、基础UI设计**
    - Android系统拥有自带的视图系统与窗口管理系统，针对基于Android的相关应用，用户可以为其设计UI交互，降低人机交互成本，提升应用的用户体验。

  - **3、扩展性设计**
    - 在Android系统架构中包含GPS、通信等相关的功能模块，开发者可以根据实际应用情况进行扩展性的开发与设计。

小结

- 针对特定的嵌入式硬件设备或环境，嵌入式GUI设计不同的用户图形界面系统应用，使嵌入式软件与硬件完美结合。
- 首先从嵌入式GUI的设计内容、设计需求、设计原则等角度，介绍了嵌入式GUI设计的基础知识。
- 然后详细分析了典型的嵌入式GUI体系结构和组成部分。
- 接着详细介绍了Qt/E、MiniGUI和Android等主流嵌入式GUI的设计方案。
- 最后介绍了MiniGUI、Android的开发环境搭建，以及基于MiniGUI、Android的GUI应用程序设计。

进一步探索

- 针对无操作系统的嵌入式开发平台，寻找一些图形库，编写GUI应用程序，并比较与基于MiniGUI平台的GUI应用开关的区别。

- 搭建iOS手机应用开发平台，并尝试编写一些简单应用，比较与在Android平台上编写方式的区别。

Thanks