

浅谈嵌入式的学习步骤及方法

一：嵌入式 c 语言

C 语言是嵌入式领域最重要也是最主要的编程语言，通过大量编程实例重点理解 C 语言的基础编程以及高级编程知识。包括：基本数据类型、数组、指针、结构体、链表、文件操作、队列、栈等。

二：Linux 基础

Linux 操作系统的概念、安装方法，详细了解 Linux 下的目录结构、基本命令、编辑器 VI，编译器 GCC，调试器 GDB 和 Make 项目管理工具，Shell Makefile 脚本编写等知识，嵌入式开发环境的搭建。

三：Linux 系统编程

重点学习标准 I/O 库，Linux 多任务编程中的多进程和多线程，以及进程间通信(pipe、FIFO、消息队列、共享内存、signal、信号量等)，同步与互斥对共享资源访问控制等重要知识，主要提升对 Linux 应用开发的理解和代码调试的能力。

四：Linux 网络编程

计算机网络在嵌入式 Linux 系统应用开发过程中使用非常广泛，通过 Linux 网络发展、TCP/IP 协议、socket 编程、TCP 网络编程、UDP 网络编程、Web 编程开发等方面入手，全面了解 Linux 网络应用程序开发。重点学习网络编程相关 API，熟练掌握 TCP 协议服务器的编程方法和并发服务器的实现，了解 HTTP 协议及其实现方法，熟悉 UDP 广播、多播的原理及编程方法，掌握混合 C/S 架构网络通信系统的设计，熟悉 HTML，Javascript 等 Web 编程技术及实现方法。

五：数据结构与算法

数据结构及算法在嵌入式底层驱动、通信协议、及各种引擎开发中会得到大量应用，对其掌握的好坏直接影响程序的效率、简洁及健壮性。此阶段的学习要重点理解数据结构与算法的基础内容，包括顺序表、链表、队列、栈、树、图、哈希表、各种查找排序算法等应用及其 C 语言实现过程。

六：C++、QT

C++是 Linux 应用开发主要语言之一，本阶段重点掌握面向对象编程的基本思想以及 C++的重要内容。图形界面编程是嵌入式开发中非常重要的一个环节。

由于 QT 具有跨平台、面向对象、丰富 API、支持 2D/3D 渲染、支持 XML、多国语等强大功能，在嵌入式领域的 GUI 开发中得到了广范的应用，在本阶段通过基于 QT 图形库的学习使学员可以熟练编写 GUI 程序，并移植 QT 应用程序到 Cortex-A8 平台。

包括 IDE 使用、QT 部件及布局管理器、信息与槽机制的应用、鼠标、键盘及绘图事件处理及文件处理的应用。

七：Cortex A8 、Linux 平台开发

通过基于 ARM Cortex-A8 处理 s5pv210 了解芯片手册的基本阅读技巧，掌握 s5pv210 系统资源、时钟控制器、电源管理、异常中断控制器、nand flash 控制器等模块，为底层平台搭建做好准备。Linux 平台包括内核裁减、内核移植、交叉编译、GNU 工具使用、内核调试、Bootloader 介绍、制作与原理分析、根文件系统制作以及向内核中添加自己的模块，并在 s5pv210 实验平台上运行自己制作的 Linux 系统，集成部署 Linux 系统整个流程。同时了解 Android 操作系统开发流程。Android 系统是基于 Linux 平台的开源操作系统，该平台由操作系统、中间件、用户界面和应用软件组成，是首个为移动终端打造的真正开放和完整的移动软件，目前它的应用不再局限于移动终端，还包括数据电视、机顶盒、PDA 等消费类电子产品。

八：驱动开发

驱动程序设计是嵌入式 Linux 开发工作中重要的一部分，也是比较困难的一部分。本阶段的学习要熟悉 Linux 的内核机制、驱动程序与用户级应用程序的接口，掌握系统对设备的并发操作。熟悉所开发硬件的工作原理，具备 ARM 硬件接口的基础知识，熟悉 ARM Cortex-A8 处理器 s5pv210 各资源、掌握 Linux 设备驱动原理框架，熟悉工程中常见 Linux 高级字符设备、块设备、网络设备、USB 设备等驱动开发，在工作中能独立胜任底层驱动开发。