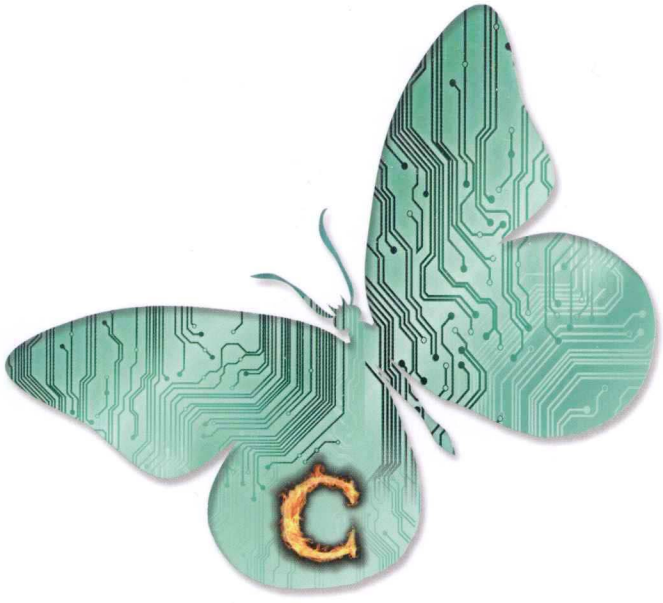
第一本深入剖析STM32官方库及其使用的权威指南
从流水灯剖析到μC/OS-III移植，零死角深入STM32库开发
配套业内最流行的野火STM32开发板，提供完整的工程文件和源代码，极具可操作性

华章科技

单片机与嵌入式



STM32库开发实战指南

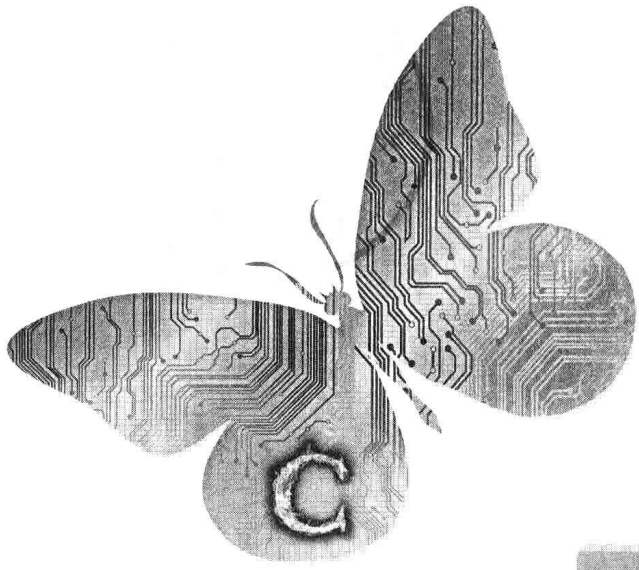
库

刘火良 杨森 编著

附光盘

机械工业出版社
China Machine Press

单片机与嵌入式



库

STM32库开发实战指南

刘火良 杨森 编著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

STM32库开发实战指南 / 刘火良，杨森编著. —北京：机械工业出版社，2013.5

ISBN 978-7-111-42637-0

Ⅰ. S… Ⅱ. ①刘… ②杨… Ⅲ. 微控制器－系统开发－指南 Ⅳ. TP332.3-62

中国版本图书馆CIP数据核字（2013）第109248号

本书基于STM32F103芯片，紧紧围绕“库”的分析和使用展开。在大量实例的基础上，本书对于如何综合运用固件库开发项目给出了具体的范例；在固件库的使用和学习的基础上，又进一步讲解了结合嵌入式实时操作系统、TCP/IP协议栈进行嵌入式系统开发的方法，让读者循序渐进、系统地掌握基于STM32官方库进行开发的方法。

本书内容翔实，案例丰富，操作性极强，可作为高校电子信息、通信工程、信息工程等相关专业的教材，也适合作为从事嵌入式领域科技工作者的参考书。

机械工业出版社（北京市西城区百万庄大街22号　　邮政编码　100037）
责任编辑：余　洁
北京市荣盛彩色印刷有限公司印刷
2013年6月第1版第1次印刷
186mm × 240mm · 31印张
标准书号：ISBN 978-7-111-42637-0
　　ISBN 978-7-89433-959-1（光盘）
定　　价：69.00元（附光盘）

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

前　言

单片机是对8/16位MCU（微控制器）的另外一种叫法。传统的8/16位单片机，久经岁月的洗礼，仍然在工业控制应用中大放光芒。然而，现在的工程师面对的更多的工业控制产品需求是多功能、易用界面、低功耗以及多任务等。基于这样的需求，以往的8/16位单片机已不能满足要求，工程师必须寻找新的符合要求的MCU。工程师虽然可以选择诸如ARM7、ARM9这类速度更快的32位MCU，但是鉴于对成本和开发门槛等种种考虑，它们还是不能满足需求。正是看准了这个市场，ARM公司推出了其全新的基于ARMv7架构的32位Cortex-M3微控制器内核。紧随其后，ST（意法半导体）公司就推出了基于Cortex-M3内核的MCU——STM32。STM32凭借其产品线的多样化、极高的性价比、简单易用的开发方式，迅速在众多Cortex-M3 MCU中脱颖而出，成为最闪亮的一颗新星。STM32一上市就迅速占领了中低端MCU市场，颇有星火燎原之势，这与它倡导的基于固件库的开发方式密不可分。采用库开发的方式可以快速上手，仅通过调用库里面的API（应用程序接口）就可以迅速搭建一个大型的程序，写出各种用户所需的应用，这就大大降低了学习的门槛和开发周期。然而，又因为在开发中只是调用API，而忽略了底层寄存器的操作，库开发被习惯了寄存器开发方式的工程师指为“于浮沙筑高台”，没有学习的价值。这种看法具有一定的片面性，他显然没有意识到这是一种全新的学习方法。试问，对于初学者，面对一个32位且有如此多寄存器的单片机，如果还像我们以往操作8位机，通过配置寄存器的方式来实现，那会是多么繁杂的一项工作？除此之外，库的开发方式自顶向下，它是迈向更高端嵌入式Linux开发的一个垫脚石。

库开发已成主流，这是不争的事实。STM32固件库之所以流行并被大家所喜爱，可以归结为以下两个原因：

（1）技术潮流

1）于个人：库开发大大地降低了学习的门槛，提高了学习的效率，使个人初步了解了大的程序设计，是一种自顶向下的学习方法，可以从上层的API层层跟踪到底层，可以彻彻底底地了解寄存器，了解CPU的内存分布，再到启动代码、开发环境的配置等。如果再深究，还会涉及编译器甚至工具链。库的学习，可不仅仅是简单地调用API，我们需要去分析这个库是如何构建的，是如何从内存到寄存器，寄存器到结构体，结构体到更各层的API，再到层层外设的文件关

联。这里面涉及了太多的C语言的知识，如关键字、宏、结构体、指针、类型转换、条件编译、断言、内联函数等。这些知识的学习，又岂能说是“于浮沙筑高台”。如果你的C语言还停留在基本语法的阶段，那么通过对库的使用和学习，你的C语言将会得到脱胎换骨的提升。学会了库开发还可以快速地迁移到ST其他系列单片机的学习，如STM32F207、STM32F407，这些MCU的固件库基本上都是兼容的。反观如此庞大的固件库，还要相互兼容，细心的人一定可以从中获益良多！

2）于公司：在公司产品开发中，产品上市速度是非常重要的成功因素，库开发可以极大地缩短产品研发周期，以便快速抢占市场。而且库让程序的维护成本更低，程序的升级更快捷。用库来开发，真可谓事半功倍，一箭双雕。

（2）市场趋势

有人曾经质疑STM32的固件库降低了MCU的性能，然而，他却没有考虑到STM32的性能和资源已经不是传统的8/16单片机可比的了。强大的硬件应该与消耗这些资源的软件相匹配，否则资源就被浪费了。硬件和软件是相辅相成、共同促进的。所以硬件改善后，工程师对于MCU的关注应该从全局着眼。

本书采用MDK开发环境，全部例程基于3.5.0版本的固件库讲解，不是简单地调用库，而是试图通过对固件库的使用详细讲解什么是库、为什么使用库、怎样使用库等一系列问题，进而引导读者使用高效率的库开发方法。

本书采用原理分析、代码讲解、实验运用这三点连线的讲解方式，循序渐进，适合在校大学生和科研机构开发人员学习使用。全书分为四个部分，第一部分（第1～5章）是库开发初级篇，涉及入门的两个主题。一个是嵌入式工程师成长之路，属于方法论的问题，涉及了一个工程师从学生时代开始，在每一个不同的阶段应该学习什么、该如何进阶等。另一个是通过对库的了解和GPIO的学习，让读者快速掌握STM32的开发方法，这是入门的第一步。第二部分（第6～16章）是库开发中级篇，讲解了STM32各个外设的使用，是学习的一个进阶阶段，也是STM32学习的重中之重。第三部分（第17～25章）是库开发高级篇，是STM32各个外设的实战演练，如MP3、液晶、摄像头、Wi-Fi等，是属于项目实战的例子，一般可直接用于工程项目的开发。第四部分（第26～28章）是库开发系统篇，这是嵌入式系统开发的必经之路，是区别裸与不裸的分水岭；这部分讲解了μC/OS最新版本μC/OS-Ⅲ在STM32中的移植，通过该移植应用实践，相信可以为以后进阶到WinCE、Linux操作系统的学习打下坚实的基础。

致　　谢

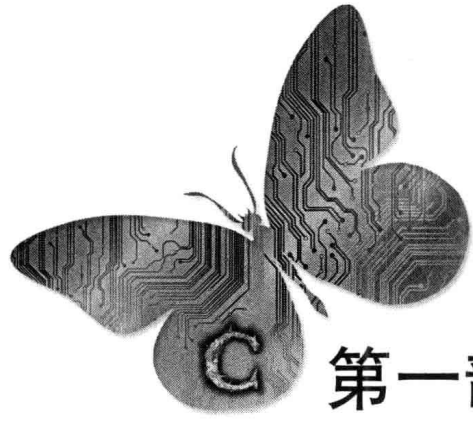
首先要感谢本书的策划编辑张国强先生，是他对STM32的关注促成了这本书的出版，同时在我们撰写书稿时对本书提出了宝贵的写作建议，并对书稿进行了仔细审阅。其次要感谢野火

工作室的成员廖锦松、曾云清等人提供的部分例程及资料；感谢好友何卓波对书稿内容的校正工作。

由于本书涉及的知识面广，时间又仓促，限于笔者的水平和经验，疏漏之处在所难免，恳请专家和读者批评指正，可以发送邮件到 wildfireteam@163.com 与作者进行交流，或者到阿莫论坛野火 M3 专区 http://www.amobbs.com 进行讨论。

刘火良　杨森

目　录



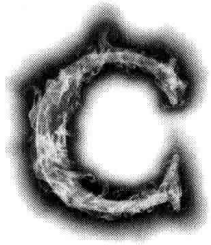
第一部分 库开发初级篇

大多数技术书籍会告诉读者如何掌握某一门技术，但少有讨论与职业规划相关的问题，例如应该学什么以及为什么学。在本书的第一部分我们将与大家共同探讨这些问题，引导读者思考为什么学习 STM32。

初级篇可以帮助初学者快速上手 STM32，以点亮 LED 灯的实例，从软件工程的角度深入剖析什么是固件库、为什么使用固件库和怎样使用固件库；从 STM32 固件库、新建工程、编译和下载程序出发，了解如何操作 GPIO，让新手步步为营，尽享 STM32 的学习乐趣。

我们对初学者的要求是具有基本的单片机基础，如 51、AVR 等，曾使用 C 语言写过单片机程序，但不需精通。读者在学习 STM32 的时候，无需太担心自己的基础，学习这门技术的决心和持之以恒的耐心才是掌握这门技术的法宝。

- 第 1 章　为什么学习 STM32
- 第 2 章　初识 STM32 固件库
- 第 3 章　GPIO 入门之流水灯
- 第 4 章　深入分析流水灯例程
- 第 5 章　调试程序



第 1 章
为什么学习 STM32

本章系统介绍了嵌入式工程师的技术成长路线，并详细介绍技术路线中的岗位设置和知识结构，让读者对于嵌入式工程师有一个全面系统地了解，并在此基础上引导工程师规划自己职业生涯。在本章的最后，作为承上启下的内容，从为什么学习 STM32 和如何学习 STM32 这两个话题入手，引导读者开始对于 STM32 的库开发方式有个初步地理解。

1.1 嵌入式技术知识结构

嵌入式技术是专用计算机系统技术，它以应用为核心，以计算机技术为基础，软硬件均可裁剪，适用在对功能、稳定性、功耗有严格要求的系统之中。嵌入式技术的开发人员需要对整个计算机体系（从底层硬件到软件操作系统）都有了解，而在这个体系之中，每个部分都可以分出一些小领域，因而技术要求很高，见图 1-1。

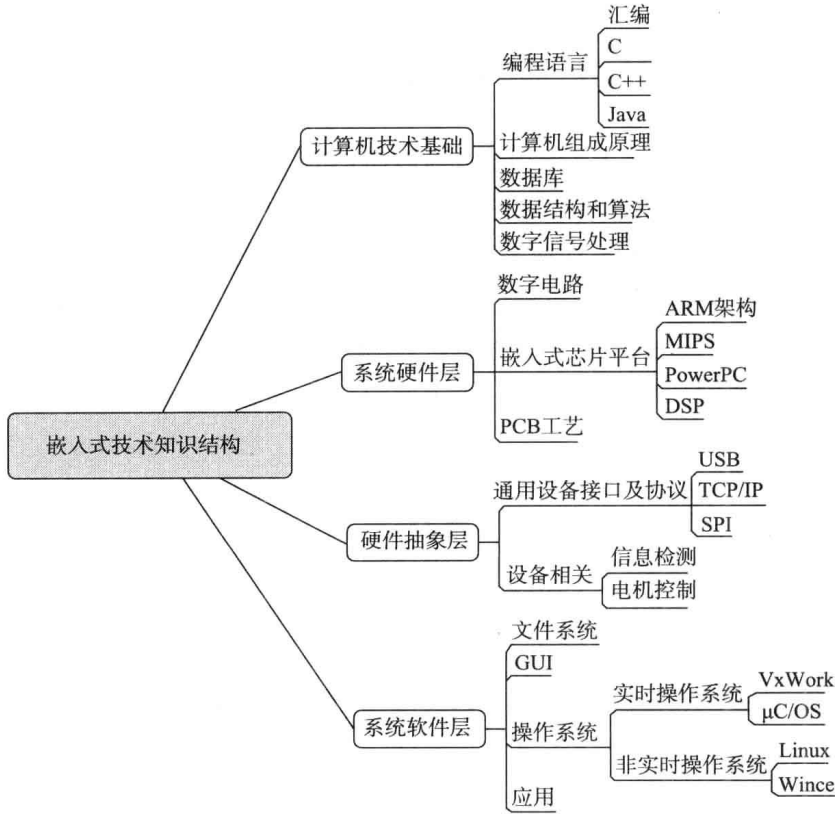


图 1-1　嵌入式技术知识结构

这个图只是粗略地概括了嵌入式技术的知识结构，但从中已经可以看出它涉及的知识面非常广，难怪众多学生甚至技术人员总是“迷茫”。不少电子专业出身的嵌入式技术人员主要从事硬件抽象层（中间层）的开发，这一层是沟通嵌入式系统的硬件层和软件操作系统的桥梁，因而主要的工作是开发驱动程序、板级应用支持、协调软硬件的开发，因而对软硬件都要有深入的了解。

1.2　嵌入式工程师成长之路

1. 从学生成为工程师

若希望从事硬件抽象层的开发，应该如何学习这些知识，才能从学生过渡到工程师呢？见图 1-2，对于希望成为其他方向的嵌入式技术人员也可以参考。

从图 1-2 可以看出，越往上层深入，就越接近于纯软件开发，但这并不代表嵌入式技术人员就不需要了解硬件，相反，上层的知识都是以底层为基础的，很多人说的“做嵌入式软件开发至少要读懂原理图”就是这个道理。

2. 职业规划

在嵌入式技术领域的公司，除了工程师还分很多职业岗位。一般公司的研发部门职位见图 1-3。

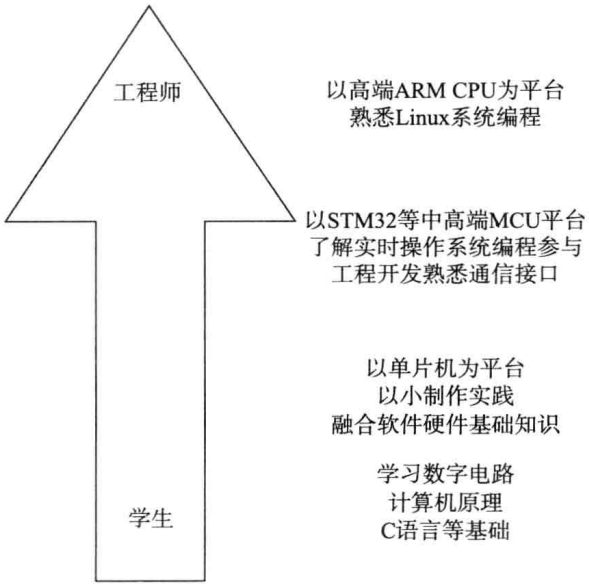


图 1-2　从事硬件抽象层开发的工程师成长之路

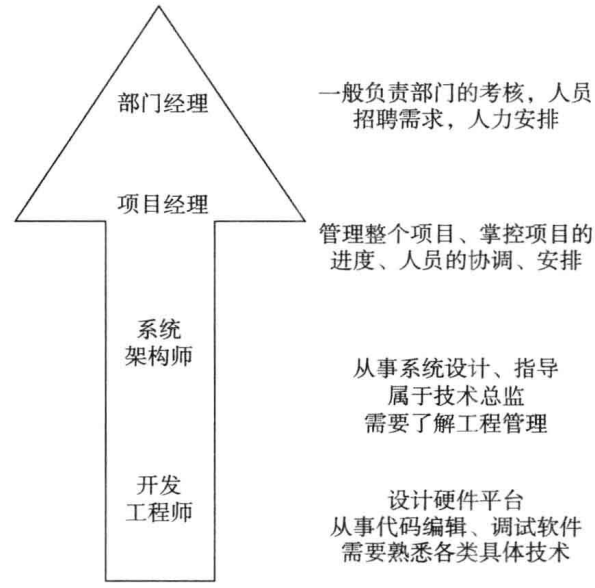


图 1-3　嵌入式工程师的职业成长路线

一般需要 3 ～ 5 年过渡到下一级的岗位，在小公司里项目经理一般也兼任部门经理。部门经理不一定要懂技术，并不是非由项目经理升职而成。直接与技术相关的是开发工程师和系统架构师，开发工程师会针对嵌入式技术的不同领域有不同的区分。在小公司里，熟悉软硬件的跨领域工程师很受欢迎，而大公司则分工明确，更看重在某领域研究得深入的开发工程师。作为系统架构师，则需要熟悉整个嵌入式领域，能够协调不同领域的开发工程师进行项目开发。

对于职业规划，不同的人有不同的见解，情况千差万别，以上所述仅供读者参考。

1.3 为什么学习 STM32

可以发现，在嵌入式领域 STM32 芯片介于低端和高端之间，它相对于普通的 8/16 位机有更多的片上外设，更先进的内核架构，可以运行 μC/OS 等实时操作系统；相对于可运行 Linux 操作系统的高端 CPU，其成本低，实时性强。这个定位使得 STM32 不仅占领了大部分中端控制器的市场，更是成为提升开发者技术的优良过渡平台，为后续的学习打下坚实的基础。

1.4 如何学习 STM32

因为 STM32 的开发方式较普通的单片机开发还是有很大的不同，所以学习时要注意如下几点：

1）转变思维，适应使用固件库的开发方式，加强运用 C 语言的能力，建立工程意识。

2）熟悉 Cortex-M 系列芯片架构，了解 CMSIS 标准，熟悉 STM32 的总线架构。

3）掌握 I^2C、SPI、SDIO、CAN、TCP/IP 等各种通信协议，掌握了这些协议，开发软件驱动就变得相对容易了。

上面有关的内容本书都会详细讲解，但“纸上得来终觉浅，绝知此事要躬行”，读者亲自编程实践是不能少的。

初级篇可以帮助初学者快速上手 STM32，写出自己的应用程序。以点亮 LED 灯的实例，从软件工程的角度深入剖析什么是固件库、为什么使用固件库和怎样使用固件库；从 STM32 固件库、新建工程、编译和下载程序出发，了解如何操作 GPIO，让新手步步为营，尽享 STM32 的学习乐趣。

我们对初学者的要求是具有基本的单片机基础，如 51、AVR 等，曾使用 C 语言写过单片机程序，但不需精通。读者在学习 STM32 的时候，无需太担心自己的基础，我们更需要的是学习的勇气，需要的是拿下 STM32 的决心。试问，我们刚开始学习最简单的单片机的时候，是不是也没基础呢，是不是因此就停止了自己学习的脚步了呢？不是的。我们需要做的是认定一个目标，行动起来，坚持朝向目标的苦行，其中艰辛芳华，唯你自知。