一本好书，重要的不只是知识本身，还有通灵的思想与方法

疯狂

STM32
实战讲学录

欧阳骏 李英芬 王小强 等编著
粟思科 审

独辟蹊径，按照**“沿着时钟树，结合固件库”**的思路，带您轻松入门。
一切从简单开始，从最简单的**STM32**处理器**时钟系统**入手，
对功能进行**逐步扩展**，最终实现较为复杂的系统。
模块化设计与**系统设计**相结合。尽最大限度实现**代码**的**复用**。

中国水利水电出版社
www.waterpub.com.cn

疯狂 STM32 实战讲学录

欧阳骏　李英芬　王小强　等编著

粟思科　审

中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书面向立志于进行STM32处理器开发的初学者以及从单片机向STM32处理器转型的工程师，依照理论与实践相结合的思想，介绍了STM32开发过程中的基础理论，并给出了具体的实例。

本书针对STM32处理器裸机开发过程中的重点、难点问题，特别是库函数的使用方法，既有基础知识的讲述，又有相关配套实验，使读者能容易、快速、全面地掌握STM32处理器开发。

本书循序渐进、内容完整、实用性强，以教材方式组织内容，可作为高等院校电子、通信、自动控制等专业的学习用书，也可供广大嵌入式工程师作为参考。

书中所用源代码下载地址：http://www.smartmaker.cn/bbs/forum.php。

图书在版编目（CIP）数据

疯狂STM32实战讲学录 / 欧阳骏等编著. -- 北京 : 中国水利水电出版社, 2013.12
ISBN 978-7-5170-1398-3

Ⅰ. ①疯… Ⅱ. ①欧… Ⅲ. ①微控制器 Ⅳ. ①TP332.3

中国版本图书馆CIP数据核字(2013)第271816号

策划编辑：周春元　　责任编辑：张玉玲　　加工编辑：李　燕　　封面设计：李　佳

书　　名	疯狂 STM32 实战讲学录
作　　者	欧阳骏　李英芬　王小强　等编著 粟思科　审
出版发行	中国水利水电出版社 （北京市海淀区玉渊潭南路 1 号 D 座　100038） 网址：www.waterpub.com.cn E-mail：mchannel@263.net（万水） sales@waterpub.com.cn 电话：（010）68367658（发行部）、82562819（万水）
经　　售	北京科水图书销售中心（零售） 电话：（010）88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排　　版	北京万水电子信息有限公司
印　　刷	北京蓝空印刷厂
规　　格	185mm×240mm　16 开本　15.25 印张　350 千字
版　　次	2013 年 12 月第 1 版　2013 年 12 月第 1 次印刷
印　　数	0001—4000 册
定　　价	38.00 元

I

前 言

众所周知，ARM Cortex-M 是基于 ARM7v 架构的 ARM 内核，因此，ST（ST Microelectronics）公司的 STM32 系列处理器是基于 ARM Cortex-M 内核 32 位 RISC 系列处理器，一般应用在家电、工业控制等领域。

由于 ST 公司提供的固件库较大，初学者在入门阶段会遇到各种问题，然而，市面上很多书籍开篇大论各种编程实例，针对各种接口、各种模块，如无线通信模块、GPS 模块，等等，这在很大程度上提高了 STM32 处理器入门的门槛。

基于上述原因，本书本着“简单就是美”的原则，取其精华，去其糟粕，在繁杂的 STM32 处理器固件库和众多的开发实例中，忽略那些令人眼花缭乱的东西，删掉那些影响开发者入门的东西，从 STM32 处理器时钟树讲起，正所谓“沿着时钟树，结合固件库，开启入门之路”，这也正是笔者写作本书的出发点，在本书的各个章节中，“沿着时钟树”的学习思路将会得到淋漓尽致的体现，希望在这众多的开发书籍中，本书能给初学者照亮一条前进的道路。

笔者深信“一本好书表达的不仅仅是知识，更是一种知识探索的方法”。作为一名初学者，学习 STM32 处理器时，很难掌握 STM32 固件库的来龙去脉，以及固件库的使用方法。但是，请读者明白，固件库是为了方便用户进行程序设计而开发的，一旦掌握了固件库的使用方法，对应用程序的开发将起到巨大的推动作用。从另一方面讲，任何数字系统都是按照特定的时钟来运行的，所以，只需要弄清楚时钟系统，然后沿着时钟系统的路线学习，很快就可以入门。因此，本书的主线是：从 STM32 处理器时钟系统入手，沿着时钟学习各个功能模块的工作原理及使用方法，即本书强调的“沿着时钟树，结合固件库”的思路。

本书的特点

- 独辟蹊径。本书以按照“沿着时钟树，结合固件库”的写作思路，使读者尽快入门。
- 模块化设计与系统设计相结合。本书秉承了 STM32 处理器固件库的模块化设计风格，力图使程序模块化，尽最大限度实现代码的复用。

笔者努力使本书成为一本 STM32 处理器开发的纪实手册，尽力展现出开发过程中的问题及其解决方法，为给读者提供参考，使读者少走弯路，因此，笔者选择用通俗的语言来叙述，而并不想用艰深晦涩难懂的语言来迷惑读者。

本书内容概述

第 1 章讲述了 STM32Fxx 系列处理器的发展历史以及相关的背景知识。

第 2 章通过一个流水灯实例对 STM32F103VET6 处理器开发进行了具体讲解，给读者展现出 STM32 处理器开发的基本方法，以及固件库的概念及使用方法。

第 3 章对 ARM C 语言程序开发过程中的基础知识进行了讲解，重点分析了位运算的定义和具体应用实例。

第 4 章对 GPIO 编程进行了具体讲解。

第 5 章主要讲解了 STM32 处理器中断系统，首先讲解了中断的基本概念，然后重点分析了中断响应的过程。

第 6 章重点对系统时钟系统和定时器进行了讲解，以理论为指导，配合了恰当的实验，向读者展示了系统时钟初始化以及定时器的具体应用。

第 7 章对 STM32F103VET6 处理器的 UART 进行了讲解，在本章并没有涉及较多的寄存器操作，对常用的几个固件库函数进行了讲解。此外本章还扩展了可变参数函数、UART 输出重定向功能进行了讲解并给出了具体的操作方法。

第 8 章对 STM32F103VET6 处理器的 DMA 进行了讲解，对 DMA 存储器到存储器的数据传输、存储器到外设的数据传输以及外设到存储器的数据传输功能进行了讲解，并给出了具体的操作方法。

第 9 章主要讨论了启动代码的组成部分及各个部分的组成原理，同时给出了简化版的启动代码。

第 10 章主要讨论了 STM32F103VET6 处理器 ADC 的使用。

第 11 章主要讨论了 SST25VF016B 存储器的使用方法，重点讲解了 STM32F103VET6 处理器硬件 SPI 的使用方法。

第 12 章主要讨论了 STM32F103VET6 处理器 DAC 的使用。

第 13 章讨论了 STM32F103VET6 处理器内部 Flash 存储器的使用方法。

第 14 章主要讨论了 STM32F103VET6 处理器 LCD 显示器的初始化使用方法以及使用 LCD 显示图片和汉字的原理。

此外，本书只是对 STM32 处理器裸机开发进行了讲解，熟悉裸机开发是进行操作系统开发的基础，秉承本书的风格，一切从简单开始，对功能进行逐步扩展，最终实现较为复杂的系统，在后续编写计划中，笔者将对嵌入式实时操作系统 μC/OS-II 以及基于开源 TCP/IP 协议栈 LwIP 的网络程序设计进行讲解，敬请期待。

适用对象

- 高等院校电子、通信、自动控制等专业学生；
- 从单片机开发向 ARM 嵌入式开发转型的工程师；
- 从事 ARM 嵌入式开发的相关技术人员。

编者与致谢

本书由欧阳骏、李英芬、王小强等编著，粟思科审。参与本书编写的还有：王治国、钟晓林、王娟、胡静、杨龙、张成林、方明、王波、陈小军、雷晓、李军华、陈晓云、方鹏、龙帆、刘亚航、凌云鹏、陈龙、曹淑明、徐伟、杨阳、张宇、刘挺、单琳、吴川、李鹏、李岩、朱榕、陈思涛和孙浩，在此一并表示感谢。

配套服务

我们为 STM32 读者和用户尽心服务，围绕 STM32 技术、产品和项目市场，探讨 STM32 应用与发展，发掘热点与重点，开展 STM32 教学。STM32 俱乐部 QQ: 183090495，电子邮件 hwhpc@163.com，欢迎 STM32 爱好者和用户联系。由于作者水平有限，书中难免有不恰当的地方，恳请广大读者批评指正。

II

目　录

1 开场白

最初的 ARM 处理器由英国剑桥的 Acorn 计算机公司设计。ARM 公司成立于 1990 年，该公司是知识产权（IP）提供商（不生产芯片）。目前，ARM 架构处理器已在高性能、低功耗、低成本的嵌入式应用领域中占据了领先地位。

ARM 公司作为嵌入式 RISC 处理器的知识产权 IP 供应商，公司本身并不直接从事芯片生产，而是将设计许可授权给合作公司，合作公司添加自己的外设，进而生产各具特色的 SoC 芯片，利用这种合伙关系，ARM 很快成为许多全球性 RISC 标准的缔造者。

目前，全世界有几十家大的半导体公司都使用 ARM 公司的授权，其中包括 Intel、IBM、Samsung、LG 半导体、NEC、SONY、PHILIP 等公司。因此，采用 ARM 处理器进行嵌入式系统开发时，开发者可以获得更多的第三方工具和技术支持，进而从一定程度上降低整个系统的研发成本，缩短研发周期，从而使产品更具市场竞争力。

STM32 系列处理器是 ST 公司基于 ARM Cortex-M 内核而专门开发的。由于采用了 Cortex 内核，因此，其中断响应速度得到了大幅度的提升，此外由于 ST 公司提供了大量的固件库，使得其开发流程得到了简化，因此 STM32 系列处理器得到了广大工程师和科研院所学生们的广泛关注。

所谓“长江后浪推前浪，前浪死在沙滩上”，对应用工程师来讲，单纯的寄存器编程略显枯燥，更严重的是，严重影响产品设计周期，但是对于 STM32 系列处理器则只需要了解固件库中的常用函数即可，更换不同系列的处理器，无非是存储器大小不同、各个模块略有不同，但是总体上固件库函数的接口风格都是一样的，因此读者掌握固件库开发的基本技巧，犹如掌握了“独孤九剑”，遇招拆招，看剑破剑。

1.1 STM32 处理器是什么

ARM 公司致力于开发新型处理器的内核，例如 ARM7TDMI 内核、ARM920T 内核等，其中

LPC2138 处理器是基于 ARM7TDMI 内核，而 S3C2440 处理器则是基于 ARM920T 内核。经过十几年的发展，造就了一系列的 ARM 架构，如图 1-1 所示。

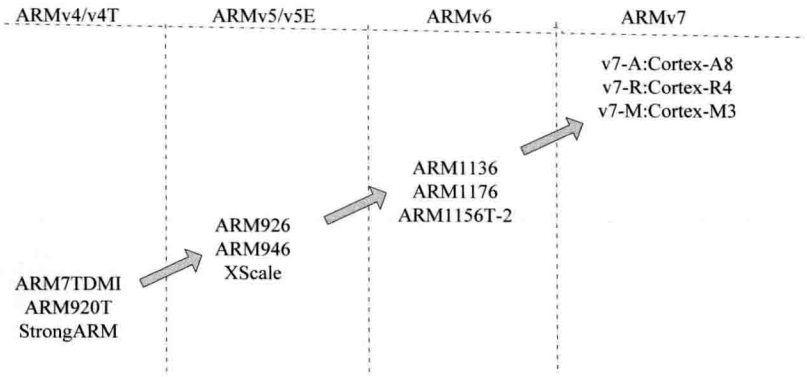

图 1-1　ARM 架构演进史

ARMv7 架构采用了新的设计理念，该架构首次演进成了三种款式，以适应不同的场合，这也是由于应用需求的多元化而采取的应对措施，毕竟，各种应用环境对处理器性能要求的侧重点不同。

- v7-A（Application）系列：侧重于高性能的应用场景，例如平板电脑、智能手机的处理器等；
- v7-R（Real Time）系列：侧重于嵌入式实时系统应用，特别是系统要求对实时性要求较高的场合；
- v7-M（Microcontroller）系列：侧重于一般的嵌入式系统应用。

ST 公司推出的 STM32 系列处理器是基于 ARM Cortex-M 内核，目前其产品阵容主要包括 STM32F 系列和 STM32L 系列，各个系列都有相应的产品来满足不同应用场景的需要，如图 1-2 所示。

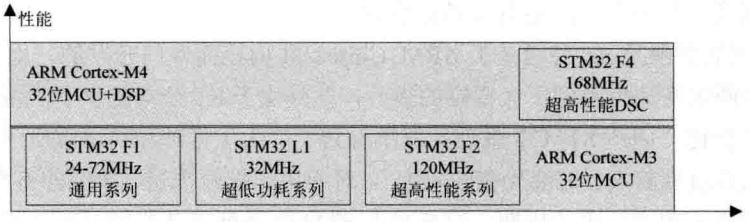

图 1-2　STM32 处理器产品系列图

1.2　数据长度

通常而言，ARM 处理器支持的数据类型有：

- 字节型数据（Byte）：数据宽度为 8bits；
- 半字数据类型（HalfWord）：数据宽度为 16bits，必须以 2 字节对齐的方式存取；

- 字数据类型（Word）：数据宽度为 32bits，必须以 4 字节对齐的方式存取。

1.3 地址空间

学习 STM32 处理器有很多优点，其中之一就是由于 Cortex-M3 内核定义了存储空间的大框架，用户学习一款处理器以后，很容易掌握其他的基于 Cortex-M3 内核的处理器，而无需重新学习存储空间的分配。基于其他架构的 ARM 处理器，其存储器映射由半导体厂商来确定存储器的具体分配，不同的厂商使用不同的存储器分配方式，因此不利于用户对存储器的选型。

1.3.1 大端和小端的故事

端模式（Endian）的这个词出自 Jonathan Swift 在 1726 年写的一篇讽刺小说《格利佛游记》。书中根据将鸡蛋敲开的方法不同，将所有的人分为两类，从圆头开始将鸡蛋敲开的人被归为 Big Endian，从尖头开始将鸡蛋敲开的人被归为 Little Endian。“端模式”的故事大意是：吃鸡蛋前，原始的方法是打破鸡蛋较大的一端。可是当时皇帝小时候吃鸡蛋，一次按古法打鸡蛋时碰巧将一个手指弄破了，因此，他的父亲就下了一道敕令，命令全体臣民吃鸡蛋时打破鸡蛋较小的一端，违令者重罚。老百姓们对这项命令极为反感……历史告诉我们，由此曾发生过六次叛乱，其中一个皇帝送了命，另一个丢了王位……

在各种计算机体系结构中，对于字节、字等的存储机制有所不同，到底是将数据的高位存放在高地址端还是存放在低地址端呢？至今也没有统一的定论。因而在计算机通信领域中存在一个很现实的问题，即通信双方传输的信息单元应该以什么样的顺序进行传送。如果不达成一致的规则，通信双方将无法进行正确的编/译码从而导致通信失败。目前在各种体系的计算机中通常采用的存储机制主要有两种：大端（big-endian）和小端（little-endian），当不同端模式的计算机进行通信时，需要进行相应的转换。

- 大端：数据的高位存放在存储器低地址端，数据的低位存放在存储器高地址端；
- 小端：数据的高位存放在存储器高地址端，数据的低位存放在存储器低地址端。

例如：字型变量 A：A=0xFF7744CC，在内存中的起始地址为 0x30000000，在内存中的起始地址为 0x30000000，其在内存中的存放格式如图 1-3 所示。

图 1-3 数据在内存中的存放格式

1.3.2 STM32Fxx 系列处理器存储空间布局

要谈及 STM32 系列处理器的存储空间布局，还得从其内核架构说起，Cortex-M3 架构对存储器映射进行了最大限度的划分，即使允许半导体厂商进行特定功能存储区域的重新定义，总体的存储器映射大框架也是不变的，这无疑对用户来说是个好消息，更换处理器时，存储器映射了然于胸的感觉还是对开发很有用的。

总体而言，Cortex-M3 支持最大 4GB 的存储空间，其地址映射关系如图 1-4 所示。

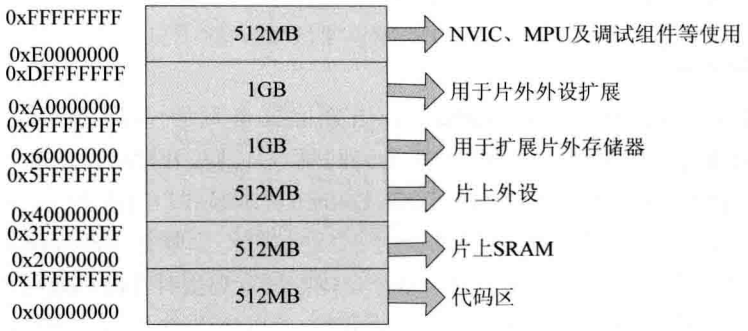

图 1-4 Cortex-M3 存储器映射

下面结合一个具体的实例分析一下存储器映射在Keil开发环境中的使用方法。对STM32F103VET6处理器而言，片上 SRAM 大小为 64KB，起始地址为 0x20000000，代码区程序存储器大小为 512KB，起始地址为 0x08000000，在 Keil 开发环境中，设置方法如图 1-5 所示。

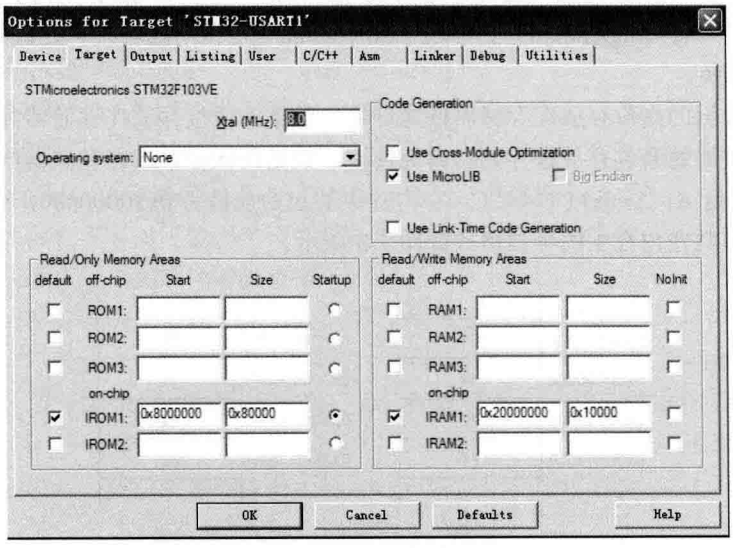

图 1-5 存储器映射实例

需要注意的是，IROM1 即为片上程序存储器，在此，即片上集成的 Flash 存储器，对应 STM32F103VET6 处理器，片上集成的 Flash 存储器大小为 512KB，即：0x80000；IRAM1 即为片上数据存储器，即片内 SRMA，对于 STM32F103VET6 处理器而言，内部 SRAM 的大小是 64KB，即：0x10000。

通过上述实例可以得出如下结论：

1）片上 SRAM 的起始地址是 0x20000000，如图 1-4 所示，这是由 Cortex-M3 内核决定的，从 0x20000000 开始的 512MB 存储空间内都可以接 SRAM，即所谓的“大框架”，但是具体 SRAM 的大小是由具体的半导体厂商来决定的，例如对 ST 公司的 STM32F103VET6 处理器而言，只扩展了 64KB 的 SRAM，说句题外话，SRAM 成本较高，因此，一般容量不是很大。

2)代码区的地址范围为 0x00000000～0x1FFFFFFF，地址空间大小为 512MB，这是由 Cortex-M3 内核决定的，即大的“大框架”，但是具体的程序存储器挂接的起始地址和程序存储器的大小是由具体的半导体厂商决定的，例如，ST 公司的 STM32F103VET6 处理器挂接的程序存储器首地址是 0x08000000，大小为 512KB，即 0x80000。

3）通过上述分析，可以很容易的得出基于 Cortex-M3 内核处理器的特征，即到“大框架”规定的存储区域去找对应的存储介质即可，这里的存储介质可以是 Flash 存储器、SRAM 等。

面对 STM32 系列处理器众多的家族成员，入门级读者往往无从选择，将面对例如不同处理器之间有什么区别与联系，如何快速的掌握一款处理器，如何进行开发等一系列的问题。

本书并不是致力于回答上述问题，而是近乎“武断性”的选择了 STM32F103VET6 进行了讲解，当读者不知道如何进行选择时，笔者推荐用户选择一款处理器，对比相应的实验资料进行学习，一个猛子扎进去，如饥似渴的学习，当你再次露出水面时，相信你已经可以从容的进行处理器开发及选型了。

当你一个猛子扎进去时，希望能看到水底下正是那闪烁着耀眼光芒的这本书，以及书中那些值得学习与推敲的开发技巧与思路。人们常说，成功的原因是因为自己站在了巨人的肩上，那么，笔者希望本书可以给读者提供一个可以站立的肩膀。

1.4 STM32F103VET6 处理器开发平台

STM32F103VET6-EV 开发板是成都智造者科技有限公司设计的针对 ST 公司 STM32F103VET6 处理器的开发板。开发板上提供了按键、LED、IIC 接口等常用的功能部件，还拥有 RS-232、TFT LCD、串行 Flash 存储器接口电路等。具体外设情况请读者参见相关开发板手册，如图 1-6 所示仅仅是给出了本书所涉及的部分模块。

注意：各个模块与处理器的接口方式并没有详细给出，在后面实验部分会给出详细的解释。

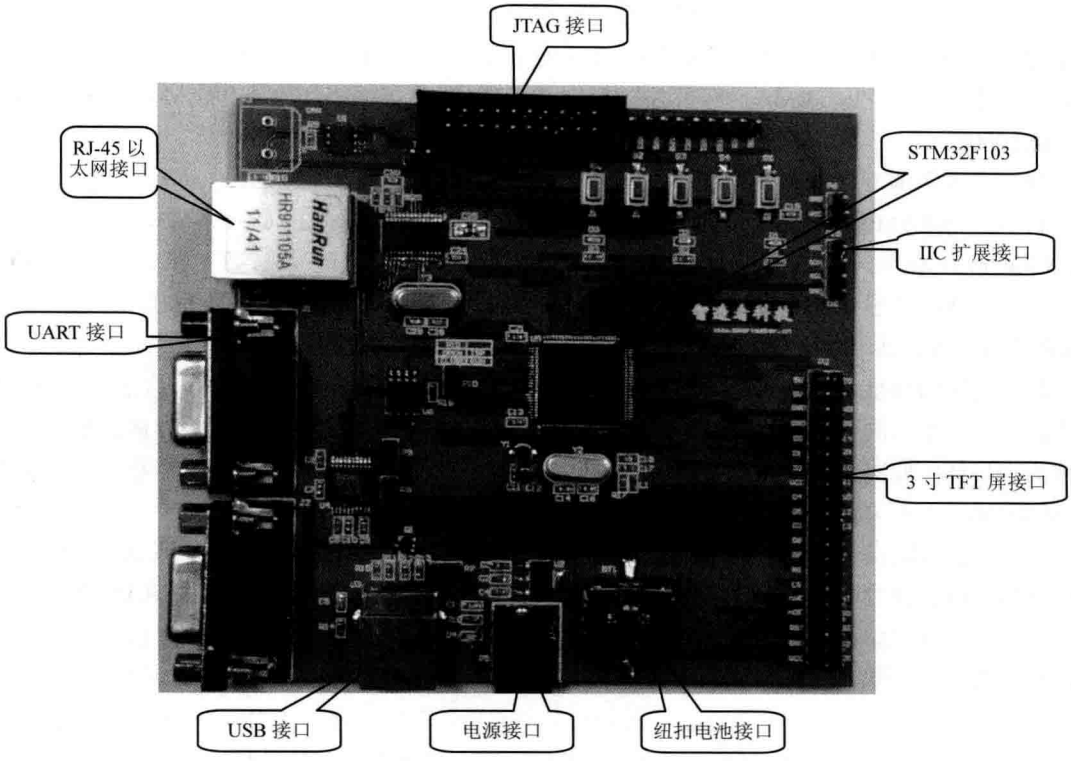

图 1-6　STM32F103VET6-EV 开发板

该开发平台主要包含了如下功能模块：

- 1 个 USB 2.0 Slave 接口；
- 3 个 LED 灯；
- 2 个 UART 接口；
- 1 个 3 寸触摸屏接口；
- 1 个复位按键；
- 4 个普通 IO 口按键；
- 1 个纽扣电池卡座；
- 1 个网络接口模块 ENC28J60；
- 2 路 AD 输入接口；
- 1 路 IIC 接口；
- 1 路 SPI 接口；
- 1 路 PWM 输出接口；
- 1 块 SPI 接口的串行 Flash 存储器。

目前，ARM 处理器已经被广泛应用于智能手机和平板电脑中。在 Windows 系统支持 ARM 处理器之后，ARM 将进军 PC 市场。不过，ARM 仍将把关注重点放在智能手机和平板电脑市场。可见，ARM 处理器已经在挑战传统的 X86 构架的处理器了。

对于初学者，面对众多的板载资源可能会有这样的疑问：

- 为什么需要专门的电源电路呢？
- 什么是 SRAM 呢?
- 在 ARM 系统中，程序放在什么地方呢?
- 系统上电后从哪里执行第一条指令呢？
- STM32Fxx 系列处理器中断系统工作原理是什么呢？
- STM32Fxx 系列处理器固件库是什么，如何使用固件库进行程序开发呢？
- STM32Fxx 系列处理器如何响应中断呢？
- STM32Fxx 系列处理器发生中断时，如何保存寄存器内容呢？
- STM32Fxx 系列处理器时钟系统是怎么产生的呢？
- STM32Fxx 系列处理器各个外设时钟和系统时钟有什么关系呢？

基于上述问题，本书展开了讨论，在阅读过程中，读者完全可以自己提出类似的一系列问题，去查阅资料，逐个解决，当问题渐渐变少时，相信读者的水平也在无形中得到了逐步的提高。

此外，市面上有很多针对 ST 公司 STM32Fxx 系列处理器的开发板，经过对比会发现，板载资源几乎是差不多的，这又是为什么呢？

本书试图从初学者的角度去理解 STM32 系列处理器，忽略那些“迷惑”初学者的技术术语，试图使用朴实的语言来理解各个功能模块，在每个功能模块的讲解过程中，都给出了具体的实例，并对其进行了分析。本书秉承“沿着时钟树，结合固件库，开启入门之路”的学习路线，希望在众多的书籍中，本书能为读者提供一条简洁的入门之路。

因此，想要了解上面问题的根源，请读者慢慢阅读本书。

1.5 本章小结

本章主要讲述了 STM32Fxx 系列处理器的基础知识，此外还给出智造者科技有限公司 STM32F103VET6-EV 开发板的部分功能模块图，使读者对硬件模块有一个整体的概念。本章最后，提出了一些常见的问题，在本书后续章节中将会对此问题进行剖析。

1.6 扩展阅读之 CISC 处理器和 RISC 处理器的关系

在过去相当长的一段时间里，计算机性能的提高往往是通过不断增加系统硬件的复杂性来实现的。但是随着集成电路技术的迅速发展，特别是超大规模集成电路（VLSI）技术的发展，为了提高软件编程的灵活性和提高程序的运行速度，硬件工程师采用的办法是：不断增加可实现复杂功能