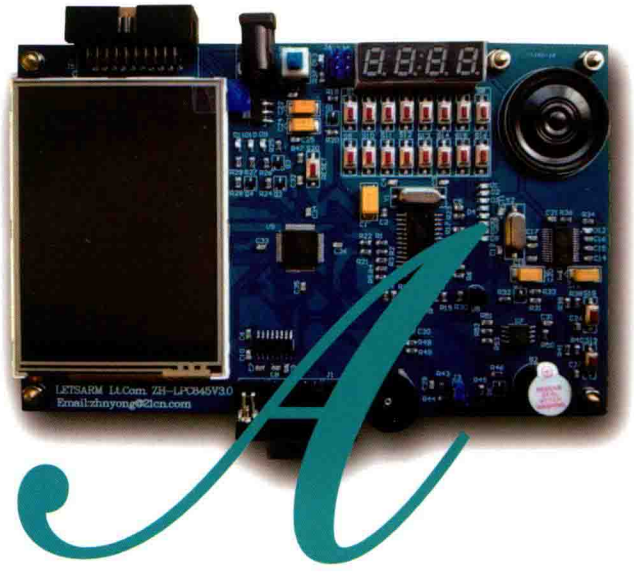
清華 开发者书库



Principles and Applications of ARM Embedded Microprocessor

Based on Cortex-M0 + Core LPC84X and μC / OS-III Operating System, Second Edition

ARM嵌入式微控制器原理与应用

基于Cortex-M0+内核LPC84X与μC/OS-III操作系统

（第2版）

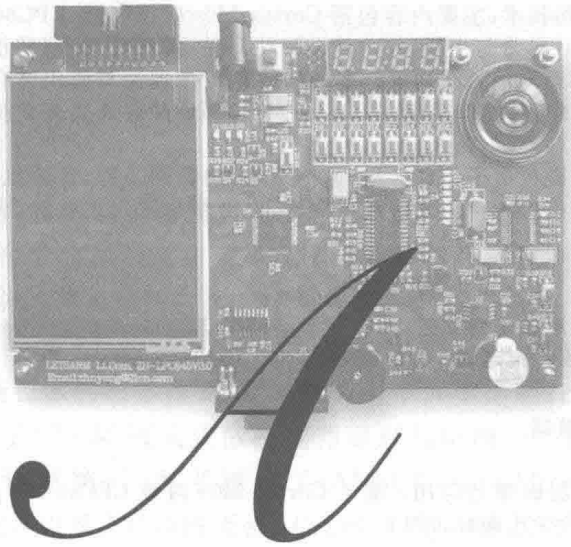
张 勇◎编著

ZhangYong

清華大學出版社

清華

开发者书库



Principles and Applications of ARM Embedded Microprocessor

Based on Cortex-M0 + Core LPC84X and μC / OS-III Operating System, Second Edition

ARM嵌入式微控制器原理与应用

基于Cortex-M0+内核LPC84X与μC/OS-III操作系统

（第2版）

张 勇◎编著

清華大學出版社

北京

内 容 简 介

ARM Cortex-M0+内核微控制器以其高性能、极低功耗和易用性等特点成为替代传统 8051 架构单片机的首选微控制器，其中以 NXP 公司 LPC84X 系列微控制器因其处理速度快、存储空间大和片内外设资源丰富而最有代表性。Micrium 公司μC/OS-Ⅱ/Ⅲ系统软件是在全球范围内被广泛加载到微控制器上的嵌入式实时操作系统。本书结合微控制器 LPC84X 与嵌入式实时操作系统μC/OS-Ⅱ/Ⅲ详细讲述了 ARM 微控制器原理与应用技术，主要内容包括 Cortex-M0+微控制器、LPC84X 硬件电路系统、Keil MDK 集成开发环境、Cortex-M0+异常与中断、片内外设驱动编程、μC/OS-Ⅱ/Ⅲ移植、μC/OS-Ⅱ/Ⅲ任务、信号量与互斥信号量以及消息邮箱与消息队列等。本书的特色在于理论与应用结合紧密且实例丰富，对学习基于 Cortex-M0+微控制器和实时操作系统μC/OS-Ⅱ/Ⅲ等领域的嵌入式开发技术，都具有很强的指导意义和参考价值。

本书可作为普通高等院校电子信息工程、通信工程、物联网工程、计算机工程、软件工程、自动控制和智能仪器等相关专业的高年级本科生或研究生教材，也可作为嵌入式系统设计爱好者和工程开发人员的参考用书。

图书在版编目(CIP)数据

ARM 嵌入式微控制器原理与应用：基于 Cortex-M0+内核 LPC84X 与μC/OS-Ⅲ操作系统/张勇编著. —2 版. —北京：清华大学出版社，2019
(清华开发者书库)
ISBN 978-7-302-52705-3

Ⅰ. ①A…　Ⅱ. ①张…　Ⅲ. ①微处理器—系统设计　Ⅳ. ①TP332.3

中国版本图书馆 CIP 数据核字(2019)第 063140 号

责任编辑：赵　凯
封面设计：李召霞
责任校对：徐俊伟
责任印制：宋　林

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：186mm×240mm　　**印　　张**：29　　**字　　数**：687 千字
版　　次：2018 年 6 月第 1 版　2019 年 6 月第 2 版　　**印　　次**：2019 年 6 月第1次印刷
定　　价：79.00 元

产品编号：083029-01

第2版前言

PREFACE

物联网技术与互联网+技术的迅猛发展，促使电子设计与智能控制领域发生了一次新的技术革命，这场技术革命的典型特征在于 ARM 微处理器和微控制器的普及应用及嵌入式实时操作系统(ERTOS)的普及应用。国内各高等院校与时俱进，在电子通信与智能控制等相关专业开设了 ARM 与 ERTOS 方面的多门课程，以培养高质量的嵌入式技术工程人员。为了适应高等院校新技术的教学需要，同时作为清华大学出版社“开发者书库”系列教材的出版计划之一，编写了《ARM 嵌入式微控制器原理与应用——基于 Cortex-M0+内核 LPC84X 与 μC/OS-Ⅲ操作系统》。本书涵盖了 ARM Cortex-M0+极低功耗微控制器 LPC84X 的设计与应用技术以及 ERTOS 系统 μC/OS-Ⅱ/Ⅲ的实战应用技术。

本书出版一年以来，受到了国内广大师生和嵌入式爱好者的喜爱，在此作者表示由衷的感谢。在过去的一年里，收到了大量读者的宝贵反馈意见，同时结合在物联网专业本科和研究生教学中遇到的问题，修订了本书第 1 版中出现的一些小问题。特别是由于 Keil MDK 最新版开发软件中芯片支撑库结构的大幅调整，使得本书原版中全部例程均需作重大修改才能运行在新版 Keil MDK 下。在这种情况下，对原书中的工程例程进行了全面的修订，形成了本书的第 2 版。同时，基于 IAR EWARM 开发环境，也编写了以 LPC84X 与 ERTOS 应用为核心的嵌入式教材《ARM Cortex-M0+嵌入式微控制器原理与应用——基于 LPC84X、IAR EWARM 与 μC/OS-Ⅲ操作系统》，即将由清华大学出版社出版，该书更适合那些习惯于借助 EWARM 进行嵌入式开发的教研人员。

本书第 2 版与第 1 版在内容安排上相同，同样具有概念表述准确、硬件方案开源、工程代码完备、应用实例丰富等特点，适用于课内教学与课外实验相结合的教学方法，也适用于结合 MOOC 技术和微课技术进行新型教学范式改革，同样适用于结合电子设计大赛进行赛课结合教学。本书配套的 ARM 学习电路板可以在教学过程中设计制作，可极大地提高本书的学习成就感和学习乐趣。对于本科二年级学生，适用内容为根据第 3 章内容制作 ARM 学习板和第 1～4 章；对于本科三年级学生，根据学习基础可选用第 1～8 章或第 1～12 章；对于研究生，适用第 9～16 章。结合作者的教学经验，针对本科三年级学生，本书的课内教学宜为 32 学时，实验教学不少于 32 学时，相关的开放实验学时为 64～96 学时。

本书由江西省学位与研究生教育教学改革研究项目(编号：JXYJG-2018-074)资助出版，特此感谢。同时，感谢恩智浦(NXP)中国公司辛华峰经理对本书编写的关心与支持；感谢北京博创智联科技有限公司陆海军总经理对本书编写的关心与支持；感谢广州天嵌计算

机科技有限公司梁传智总经理对本书编写的关心与支持；感谢清华大学出版社赵凯编辑的辛勤工作；感谢我的爱人贾晓天老师在资料检索和LPC845学习板焊装调试方面所做的大量工作；感谢阅读了作者已出版的教材并反馈了宝贵意见的读者们。本书的编写通俗易懂，其自学门槛较以往的教材大大降低。

由于作者水平有限，书中难免会有纰漏之处，敬请同行专家和读者朋友批评指正(联系方式：15270015009@qq.com)。

张　勇

2019年5月

第1版前言

PREFACE

当前，ARM 微控制器正在逐步替代传统 8051 架构单片机而成为嵌入式系统的核心控制器。2010 年以后，ARM 公司主推 Cortex 系列内核，Cortex 系列分为 R 系列、A 系列和 M 系列，其中，A 系列是高性能内核，用于基于 Android 操作系统的智能手机和平板电脑，支持 ARM、Thumb 和 Thumb-2 指令集；R 系列为微处理器内核，支持 ARM、Thumb 和 Thumb-2 指令集；M 系列为低功耗微控制器内核，仅支持 Thumb-2 指令集，诞生于 2004 年，最早推出的内核为 Cortex-M3，目前有 Cortex-M0、M0+、M1、M3、M4 和 M7 等，用于支持快速中断的嵌入式实时应用系统中。在 Cortex 系列中，M 系列芯片的应用量最大，每年的应用量为几十亿块。

在 Cortex-M 系列中，M0 和 M0+内核都是极低功耗内核，M0+内核的功耗比 M0 内核更低（ARM 公司公布的功耗数据为 11.2 μW/MHz），被誉为全球功耗最低的微控制器内核，主要应用在控制和检测领域，涵盖了传统 8051 单片机的应用领域，比传统 8051 单片机在处理速度、功耗、片上外设灵活多样性、中断数量与中断反应能力、编程与调试等诸多方面都有更大优势，M0+内核的代表芯片如 NXP 公司的 LPC845 微控制器。

基于 ARM Cortex-M0+微控制器的软件开发有两种方式，即传统的芯片级别的应用软件开发和加载嵌入式实时操作系统的应用软件开发。芯片级别的应用软件开发方式直接使用 C 语言函数管理硬件外设驱动和实现用户功能，称之为面向函数的程序设计方式；加载嵌入式实时操作系统的应用软件开发使用嵌入式操作系统管理硬件外设和存储资源，借助于用户任务实现用户功能，称之为面向任务的程序设计方式。由于 Cortex-M0+微控制器片内 RAM 空间丰富，一般在 8KB 以上，适宜加载嵌入式实时操作系统（RTOS）μC/OS-Ⅱ或 μC/OS-Ⅲ。在 Cortex-M0+微控制器上加载了 RTOS 后，将显著加速项目的开发进度。

本书主要以 Cortex-M0+内核 LPC845 微控制器为例，在介绍了 Cortex-M0+内核组成原理和 LPC84X 微控制器芯片结构后，详细介绍了 LPC845 典型硬件系统及其片上外设的驱动方法，基于面向函数的程序设计方法介绍了 LED 灯、蜂鸣器、按键、数码管、温度显示（DS18B20）、串口通信、模数转换器（ADC）、存储器访问、LCD 屏显示和触摸屏输入等外设驱动程序设计技术；然后，详细介绍了嵌入式实时操作系统 μC/OS-Ⅱ和 μC/OS-Ⅲ在 LPC845 微控制器上的移植与应用技术，包括用户任务、信号量与互斥信号量、消息邮箱与消息队列等组件应用程序设计方法，重点在于阐述面向任务的程序设计方法及其优越性。

本书讲义经过多名教师的使用，理论学时宜为 32 学时，实验学时为 32 学时。建议讲述

内容为第1～12章(第一篇与第二篇),选学内容为第13～16章(第三篇),按书中章节顺序讲述。作者巧妙地组织了书中的全部实例,使得全部实例代码均是完整的。因此,要求读者必须在掌握了前面章节实例的基础上,才能学习后面章节的实例。对于自学本书的嵌入式爱好者而言,要求至少具有数字电路、模拟电路、C语言程序设计等课程的基础知识,并建议使用LPC845学习板辅助学习,以增加学习乐趣。

本书具有以下三个方面的特色:

1. 公布了基于LPC845微控制器为核心的开源硬件平台,对嵌入式硬件开发具有很强的指导作用。

2. 全书工程实例丰富,通过完整的工程实例详细讲述了函数级别与任务级别的程序设计方法,对于嵌入式系统应用软件开发具有颇强的指导意义。

3. 结合LPC845硬件平台,详细讲述了嵌入式实时操作系统μC/OS-Ⅱ/Ⅲ的任务管理和系统组件应用方法,对学习和应用μC/OS-Ⅱ/Ⅲ具有良好的可借鉴性。

本书由江西省学位与研究生教育教学改革研究项目(编号:JXYJG-2018-074)资助出版,特此感谢。同时,感谢恩智浦(NXP)中国公司辛华峰经理对本书编写的关心与支持;感谢北京博创智联科技有限公司陆海军总经理对本书编写的关心与支持;感谢广州天嵌计算机科技有限公司梁传智总经理对本书编写的关心与支持;感谢清华大学出版社的辛勤工作;感谢我的爱人贾晓天在资料检索和LPC845学习板焊装调试方面所做的大量工作;感谢阅读了作者已出版的教材并反馈了宝贵意见的读者们。本书的编写通俗易懂,其自学门槛较以往的教材大大降低。

由于作者水平有限,书中难免会有纰漏之处,敬请同行专家和读者朋友批评指正(联系方式:15270015009@qq.com)。

张　勇

2018年4月于

江西财经大学枫林园

目录

CONTENTS

第一篇 LPC84X 典型硬件系统与芯片级软件设计

第一篇 LPC84X典型硬件系统与芯片级软件设计

本篇包括第1～8章，为全书的硬件系统与芯片级软件设计部分，将依次介绍ARM Cortex-M0+内核、LPC84X微控制器、LPC845典型硬件系统电路和面向函数的程序设计方法。在本篇中，借助于实例和Keil MDK集成开发环境，先后详细阐述以下板级外设的芯片级工程程序设计方法：

(1) LED灯与蜂鸣器控制；

(2) 按键与数码管显示(包含DS18B20温度传感器访问技术)；

(3) 串口通信(包含SYN6288声码器访问技术)；

(4) 模数转换器(ADC)和存储器访问技术；

(5) 240×320像素分辨率彩色LCD屏与触摸屏控制。

需要说明的是，全书中使用了Keil MDK 5.26版本，该版本为2018年10月发布的最新版本。建议读者使用最新的Keil MDK版本(由于版本升级而导致的工程兼容性问题，请与作者联系)。

第1章

ARM Cortex-M0+内核

ARM是Advanced RISC Machine(高级精简指令集机器)的缩写,现为ARM公司的注册商标。ARM Cortex-M0+内核属于ARM公司推出的Cortex-M系列内核之一,相对于高性能的Cortex-M3内核而言,它具有体积小、功耗低和控制灵活等特点,主要针对传统单片机的控制与显示等嵌入式系统应用。本章将介绍Cortex-M0+内核特点、架构、存储器配置和内核寄存器等内容。

1.1 ARM Cortex-M0+内核特点

Cortex-M0+内核基于ARMv6-M体系结构,使用ARMv6-M汇编语言指令集,它具有以下特点:

(1) Cortex-M0+内核包含极低数量的门电路,是目前全球功耗最低的内核,特别适用于对功耗要求苛刻的嵌入式系统应用场合。

(2) 支持32位的Thumb-2扩展指令集和16位的Thumb指令集,代码的执行效率远远高于8位的单片机汇编指令。

(3) 支持单周期的I/O(输入/输出)口访问,对外设的控制速度快。

(4) 具有低功耗工作模式,在内核空闲时可以使其进入低功耗模式,从而极大地节约电能;当需要内核工作时,通过与内核紧耦合的快速中断唤醒单元使其进入正常工作模式。

(5) 内核中的各个组件采用模块化结构,通过精简的高性能总线(AHB-Lite)连接在一起,内核中的功耗管理单元可以动态配置各个组件的工作状态,可根据需要使某些空闲的组件处于掉电模式,以尽可能地减少功耗。

(6) 可执行代码保存在Flash存储区中,Cortex-M0+内核支持从Flash中以极低的功耗快速读取指令,并以极低的功耗(工作在一个相对较低的CPU时钟下)在内核中高速执行代码。

(7) Cortex-M0+内核支持硬件乘法器,硬件乘法器最早出现在DSP(数字信号处理器)芯片中,与加法器协同工作,并称为乘加器,是指在一个CPU时钟周期内,用硬件电路直接实现A×B+C的三操作数运算。这里Cortex-M0+支持的硬件乘法器可以在一个CPU时

钟周期内实现 A×B 的二操作数运算。

(8) Cortex-M0+内核的每条汇编指令的执行周期是确定的；中断处理的时间是确定的、高效的，且具有快速中断处理能力，特别适用于对实时性要求苛刻的智能控制场合。

(9) 支持二线的串行调试接口(SWD)，只需要使用芯片的两根引脚就可以实现对Cortex-M0+内核芯片的在线仿真与调试，通过 SWD 可以向芯片的 Flash 存储器固化程序代码，且具有指令跟踪执行功能。而绝大多数的传统单片机是不能在线仿真调试的，因此基于单片机的工程测试复杂且周期漫长。

(10) Cortex-M0+内核不是物理形态的微控制器芯片，而是属于知识产权(IP)，所以常被称为 IP 软核。目前全球至少有 200 多家大型半导体公司购买了 ARM 公司的 IP 软核，生产集成了 IP 软核的微控制器芯片(称为流片，流片测试成功后进入芯片量产阶段)。所有集成了 Cortex-M0+内核的微控制器芯片，均可使用相同的集成开发环境(如 Keil 公司的 MDK 和 IAR 公司的 EWARM 等)和相同的仿真器(如 ULink2 和 JLink V8 等)进行软件开发。事实上，几乎全部的 ARM 芯片都使用相同的开发环境和仿真器，但是对于传统的单片机而言，不同半导体厂商生产的单片机所用的开发环境和编程下载器往往不相同。

1.2 ARM Cortex-M0+内核架构

相对于 8 位的传统 8051 单片机而言，Cortex-M0+内核是 32 位的微控制器内核，其内部总线宽度为 32 位，指令和数据传输能力大大提升。Cortex-M0+内核架构如图 1-1 所示。

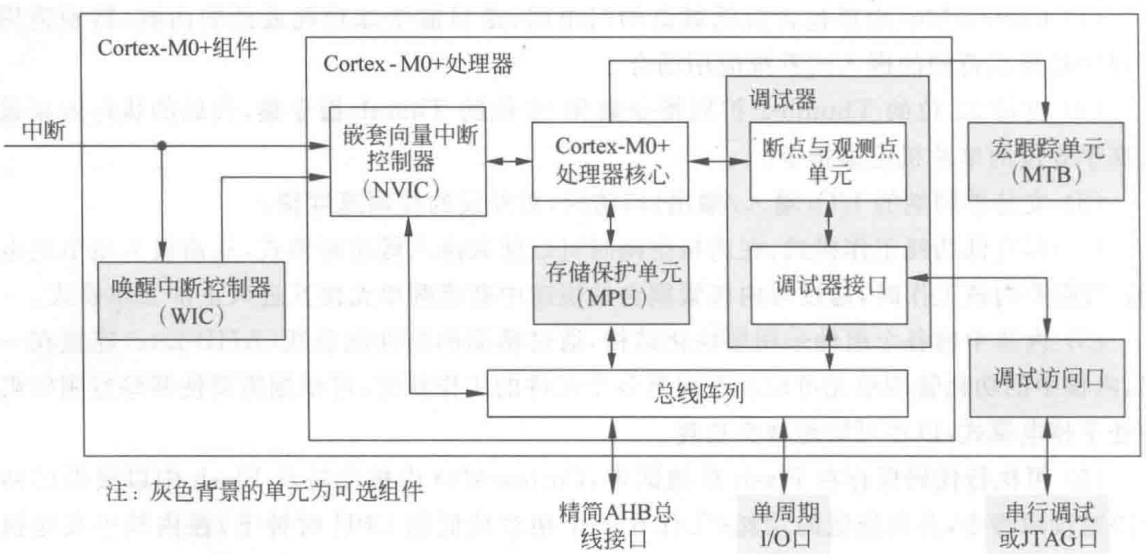


图 1-1 Cortex-M0+内核架构

由图 1-1 可知，Cortex-M0+内核由 Cortex-M0+处理器和三个可选的组件，即唤醒中断控制器(WIC)、宏跟踪单元(MTB)和调试访问口组成，Cortex-M0+处理器包括嵌套向量中

断控制器(NVIC)、Cortex-M0+处理器核心和两个可选的组件,即存储保护单元(MPU)和调试器组成,其中调试器又包括断点与观测点单元和调试器接口。Cortex-M0+内核与外部通过总线阵列和中断进行通信,其中中断为单向输入口,与总线阵列相连接的精简高性能总线(AHB)接口以及可选的单周期 I/O 口和串行调试或 JTAG 口均为双向口。

32 位的 Cortex-M0+处理器核心是计算和控制中心,采用了两级流水的冯·诺依曼结构,执行 ARMv6-M 指令集(即 16 位的 Thumb-2 指令集,含 32 位的扩展指令),集成了一个单周期的乘法器(用于高性能芯片中)或一个 32 周期的乘法器(用于低功耗芯片中)。

Cortex-M0+内核具有很强的中断处理能力,一个优先级可配置的嵌套向量中断控制器(NVIC)直接与 Cortex-M0+处理器核心相连接,通过这种紧耦合的嵌套向量中断控制器,可以实现不可屏蔽中断、中断尾链、快速中断响应、睡眠态唤醒中断和四级中断优先级。这里的"中断尾链"是指当有多个中断被同时触发时,优先级高的中断响应完成后,不需要进行运行环境的恢复,而是直接运行优先级次高的中断,全部中断响应完后,再恢复运行环境。如果没有中断尾链功能,则处理器在响应一个中断前,先进行入栈操作,保存当前中断触发时的运行环境,然后处理器暂停当前程序的执行去响应中断,响应完中断后,进行出栈操作,恢复响应中断前的环境(即使程序计数器指针 PC 指向被中断的程序位置处)继续执行原来的程序,接着重复这些操作响应下一个中断。因此,没有中断尾链功能时,两个连续响应的中断间需要插入一次出栈和一次入栈操作(即恢复前一个运行环境和保存后一个运行环境),而具有中断尾链功能时,这两次堆栈操作均被省略掉了。

Cortex-M0+内核具有很强的调试能力,通过调试访问口,外部的串行调试或 JTAG 调试口与 Cortex-M0+处理器的调试器相连接,调试器直接与 Cortex-M0+处理器核心连接,还通过它与宏跟踪单元相连接。因此,通过串行调试或 JTAG 调试口可以访问 Cortex-M0+内核的全部资源,包括调试或跟踪程序代码的执行,还可以检查代码的执行结果。

存储保护单元(MPU)可以对存储器的某些空间设定访问权限,使用特定的程序代码才能访问这些空间,普通的程序代码则无权访问,这样可以有效地保护关键的程序代码存储区或数据区不受意外访问(例如病毒)的侵害。

在图 1-1 中,相对于 Cortex-M0+处理器核心而言,其余的组件均称为 Cortex-M0+内核的外设,所有这些组件均为 IP 核。半导体厂商在这个 IP 核的基础上添加中断发生器、存储器、与精简 AHB 总线(通过高级外设总线 APB)相连接的多功能芯片外设和 I/O 口等,即可以得到特定功能的微控制器芯片。

1.3 ARM Cortex-M0+存储器配置

Cortex-M0+存储空间的最大访问能力为 2^{32} 字节,即 4GB。对于 Cortex-M0+而言,8 位(8 bit)为一个字节,16 位称为半字,32 位称为字。以字为单位,Cortex-M0+的存储空间的最大访问能力为 2^{30} 字,即从 $0 \sim 2^{30}-1$ 字。一般地,访问地址习惯采用字节地址,此时 Cortex-M0+的存储空间配置如图 1-2 所示。