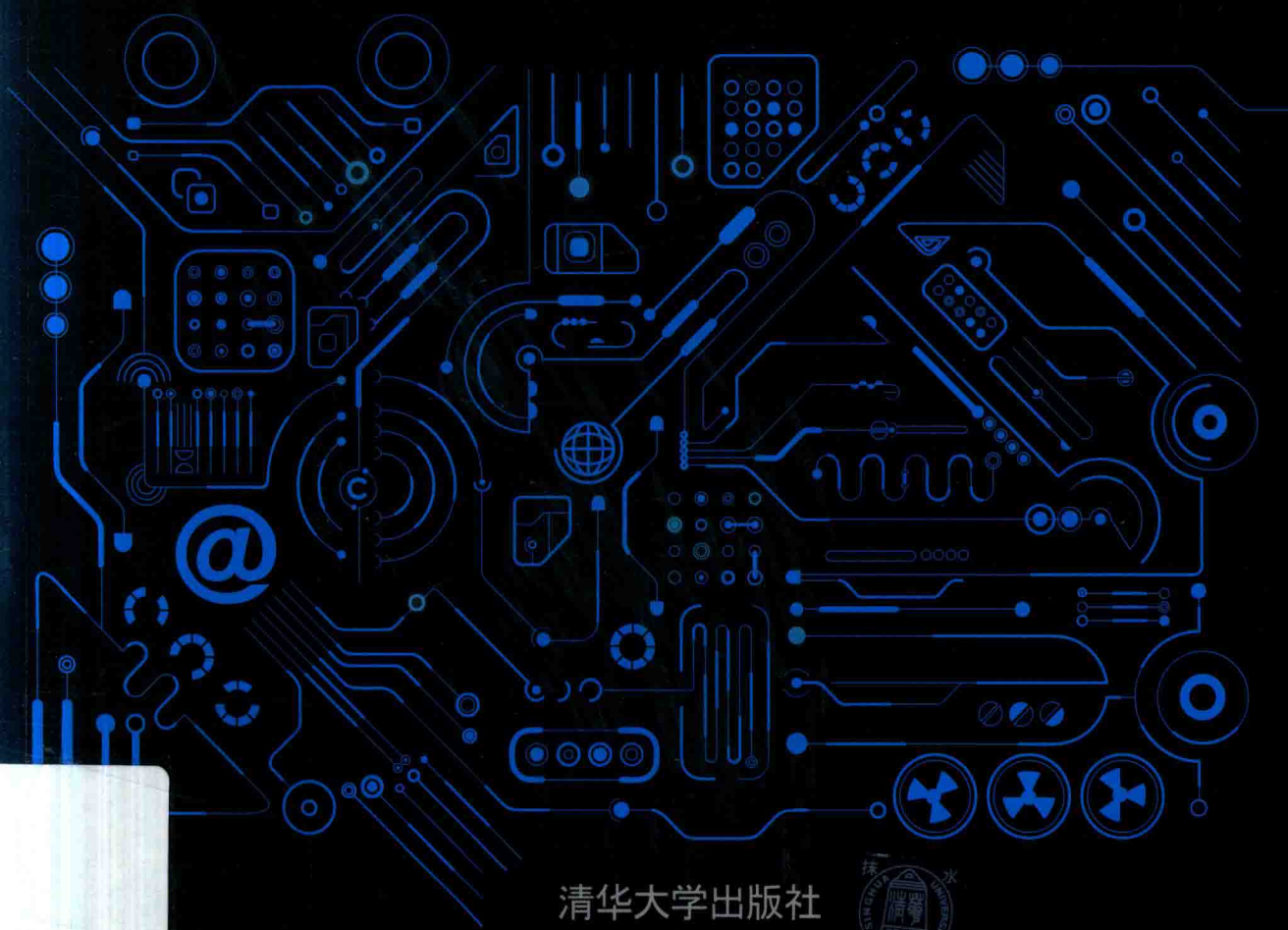
细说 Cortex-M0开发

以LPC1114为例

冯迅 编著

学习ARM Cortex-M0开发，从这里起航

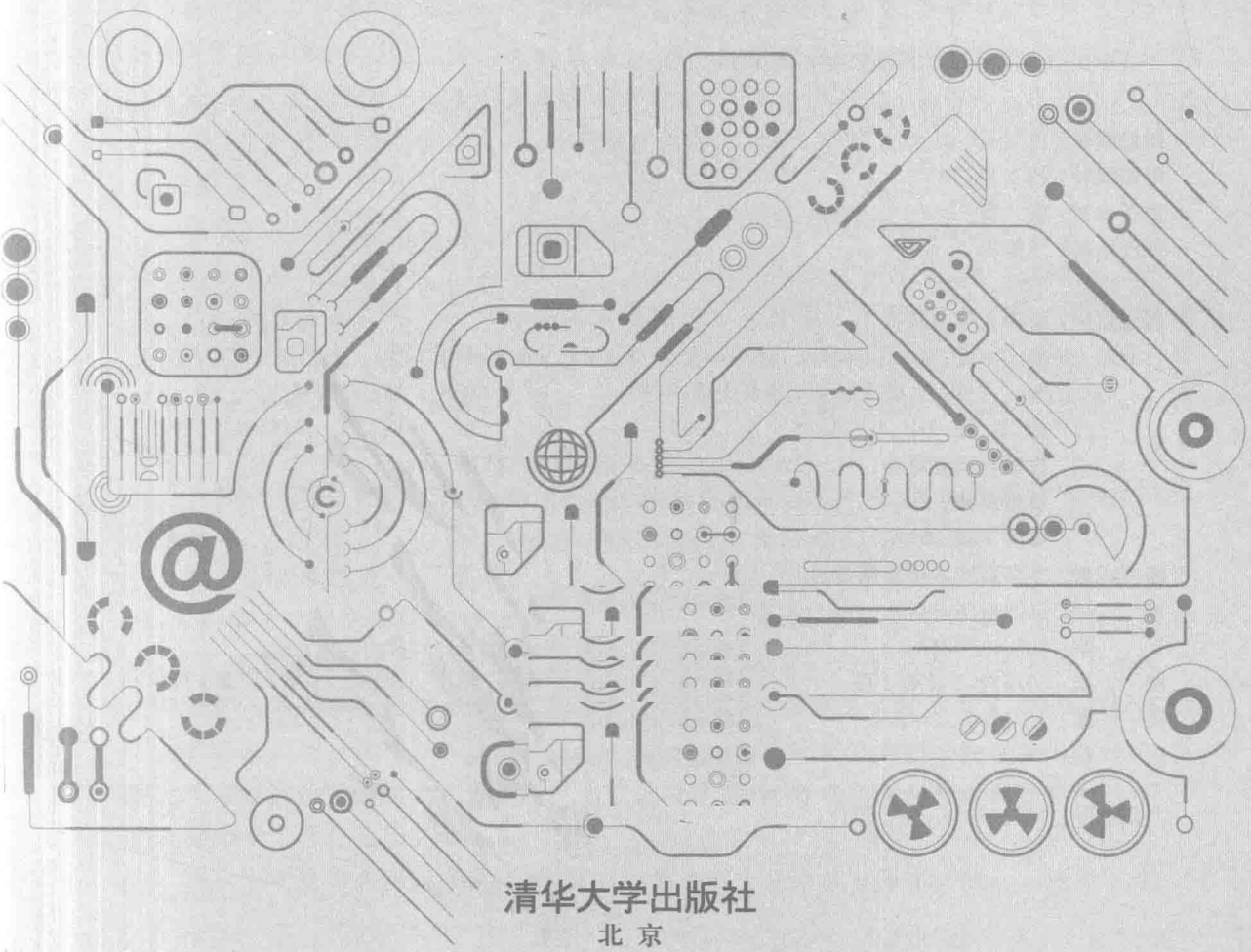


清华大学出版社

细说 Cortex-M0开发

以LPC1114为例

冯迅 编著



清华大学出版社
北京

内 容 简 介

本书以恩智浦公司(NXP)的 LPC1114 芯片为例，详细讨论了 ARM Cortext-M0 处理器的开发过程。主要讲述了开发环境 MDK-ARM 的配置及 LPC1114 内部各个组成模块的结构及使用方法，包含 GPIO 端口、时钟源、NVIC 中断系统、系统定时器、定时器/计数器、串口、A/D 转换、I^2C 接口、SPI 接口、看门狗、功耗管理及 Flash 编程固件等内容。全书以应用为主，针对 LPC1114 开发中可能遇到的问题都进行了详细的讨论。

书中所有代码无须任何修改就可直接使用，所有程序代码均在 MDK-ARM4.22a 环境下调试通过。

本书可作为高校及相关培训机构的教材，也可作为爱好者自学使用的书籍，还可作为嵌入式系统工程开发人员的参考手册。

图书在版编目(CIP)数据

细说 Cortex-M0 开发：以 LPC1114 为例/冯迅编著. —北京：清华大学出版社，2018
ISBN 978-7-302-48998-6

Ⅰ. ①细… Ⅱ. ①冯… Ⅲ. ①微处理器－系统设计 Ⅳ. ①TP332

中国版本图书馆 CIP 数据核字(2017)第 294406 号

责任编辑：白立军
封面设计：杨玉兰
责任校对：梁　毅
责任印制：王静怡

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：北京嘉实印刷有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：20.25　　**字　　数**：481 千字
版　　次：2018 年 1 月第 1 版　　**印　　次**：2018 年 1 月第1次印刷
印　　数：1～1500
定　　价：59.00 元

产品编号：070806-01

前言

在当前嵌入式系统的低端应用领域，以 ARM Cortext-M0 为核心的芯片有着良好的市场前景。世界各大半导体厂商（如 NXP、Freescale、Atmel 等）都陆续推出了基于 Cortext-M0 内核的系列芯片。随着此类芯片的大量普及，其价格也越来越低，让广大开发者以 8 位芯片的价格获得了 32 位芯片的性能，大有取代传统低端芯片之势，所以，学习和使用 Cortext-M0 开发的人也越来越多，市场潜力较大。

2010 年，恩智浦（NXP）公司推出的基于 ARM Cortex-M0 内核的 LPC1100 系列芯片具有非常出色的性价比，随后又不断扩展升级，迅速引起业界的广泛关注。本书选取目前较为常用的 LPC1114 芯片，并把它当作一个 32 位的高级单片机来进行分析和讨论，本书在结构上也采用了传统单片机教材的形式。在编写过程中，均以官方提供的英文版数据手册为依据，书中所有引用的表格、图片、数据等都来自官方手册，但引用并非简单地对手册进行罗列翻译，而是在理解内容的基础上，结合初学者的认知规律和作者多年的教学实践经验，通过学习者易于接受的形式进行讲述，可提高学习兴趣，减小挫折感。书中讲述的内容符合初学者的认知规律，采取循序渐进的方式，力求解决在学习中可能遇到的问题，突出实际应用。

本书选取国内流行的 MDK-ARM 作为 LPC1114 的开发环境，用实例的方式讲述了开发中程序结构之间的关系，并详细剖析了头文件的用法，让读者不仅知其然还知其所以然。然后着重讲述 LPC1114 内部各组成模块的功能及实际应用，包含 GPIO 端口、时钟源、NVIC 中断系统、系统定时器、定时器/计数器、串口、A/D 转换、I^2C 接口、SPI 接口、看门狗、功耗管理及 Flash 编程固件等内容。在每个章节的结尾，都会给出一个与该部分内容密切相关的实例，让学习者容易理解和掌握所讲授的内容，以此来提升学习效果。全书面向实际开发应用，注重对细节的分析，力求让读者能容易上手，快速掌握 Cortex-M0 的开发过程。

对于处理器的学习，从本质上来说就是学习处理器中各个寄存器的原理及其使用方法，为此本书讨论了 LPC1114 所用到的几乎所有寄存器，但并没有直接罗列所有的寄存器，而是把寄存器的学习放到了实际的例子当中，用到哪个就讨论哪个，这样能有效地避免单纯学习寄存器带来的枯燥。书中对寄存器的描述都先引用官方原文，然后再对其内容进行详细讨论。这样做不仅能提高资料的权威性，还可以避免翻译和理解上的歧义，有利于读者进行学习和参考。即便对于开发人员，本书也可作为资料手册进行查询，而不必再去翻阅官方数据手册。

书中所有程序代码及教学课件，都可以到清华大学出版社的官方网站上进行下载。

本书由云南师范大学信息学院冯迅编写，希望能对有志于学习 Cortex-M0 开发的读者起到积极的帮助作用。在此，对清华大学出版社提供的大力支持表示由衷的感谢！由于编者知识局限和时间仓促，书中不足之处在所难免，恳请广大读者批评指正！

冯　迅

2017 年 6 月

目 录

第1章

LPC1114及其开发环境简介

本章学习目标

(1) 了解LPC111x系列微控制器的特性。

(2) 掌握MDK-ARM开发环境及配置。

1.1 LPC1114及其主要特性

LPC111x系列微控制器是恩智浦(NXP)公司推出的基于ARM Cortex-M0内核的低功耗、32位微控制器,具有高性能、低功耗、简单指令集、统一编址寻址等优点,相对于当前市场上的8位、16位架构微控制器来说,它能有效地降低代码长度。

LPC111x系列微控制器具有以下特点。

(1) ARM Cortex-M0处理器,工作频率高达50MHz。

(2) ARM Cortex-M0内嵌向量中断控制器(NVIC)。

(3) 32KB(LPC1114)、4KB(LPC1113)、16KB(LPC1112)或8KB(LPC1111)片上Flash可编程存储器。

(4) 高达8KB的SRAM。

(5) 通过片上引导(BootLoader)软件实现现场编程(ISP)和在线编程(IAP)。

(6) 串行接口。

① 有分数波特率发生器,内部FIFO,支持RS-485/EIA-485总线和Modem控制的UART。

② 最多可有两个具有SSP特性的、带FIFO的SPI控制器,具有很好的多协议兼容性(第二个SPI只在LQFP48和PLCC44封装中存在)。

③ I^2C总线接口支持全速I^2C总线规格和增强快速模式(速率可达到1Mb/s,具有多地址识别监听模式)。

(7) 其他外设。

① 多达42个带有可配置上拉/下拉电阻的GPIO引脚。

② 每个引脚有大电流(20mA)驱动输出。

③ 两个I^2C总线引脚在增强快速模式时,为大电流灌入驱动(20mA)。

④ 4个通用定时器(共4个捕获输入和13个比较输出)。

⑤ 看门狗定时器(WDT)。

⑥ 系统滴答定时器。

(8) 串行线调试(SWD)。

本书以 LPC1114 作为 ARM Cortex-M0 的学习选型。LPC1114 是恩智浦公司推出的基于 ARM Cotex-M0 内核的一个 32 位低功耗微控制器，其内部拥有 32KB 的 Flash ROM、8KB 的 RAM，低电压供电(1.8～3.6V)，速度快(0.9DMIPS/MHz)，功能强大，价格低廉，应用非常广泛，其 LQFP48 封装的实物外形如图 1-1 所示。

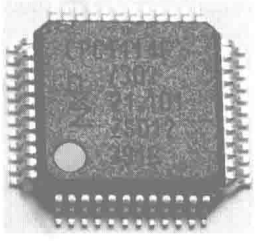


图 1-1 LQFP48 封装的 LPC1114

LPC1114 芯片一共有 48 根引脚，其中 42 根是通用输入输出引脚(GPIO)，分为 PIO0 口(PIO0_0～PIO0_11)12 根脚，PIO1 口(PIO1_0～PIO1_11)12 根脚，PIO2 口(PIO2_0～PIO2_11)12 根脚，PIO3 口(PIO3_0～PIO3_5)6 根脚。余下的 6 根引脚分别是两个晶振引脚、两个正电源引脚(系统、A/D)和两个地引脚(系统、A/D)。在要求不严格的情况下，系统电源和 A/D 电源可以接在一起。图 1-2 给出了一个 LQFP48 封装的 LPC1114 微控制器引脚分布情况。

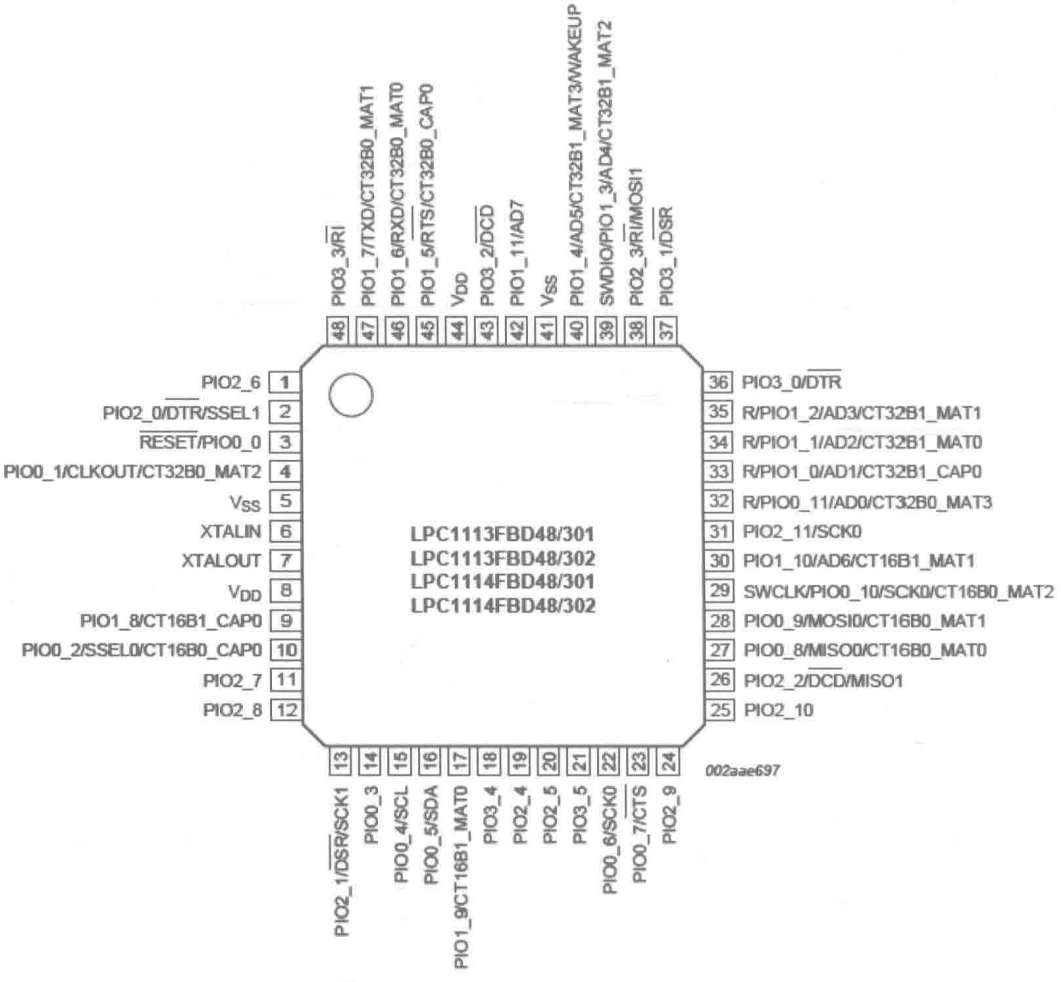


图 1-2 LPC1114 引脚图(LQFP48)

从图 1-2 中可以看到，LPC1114 的大多数引脚都具有复用功能，甚至芯片的复位端(RESET)都与引脚 PIO0_0 复用在一起了(第 3 脚)，所以 LPC1114 芯片没有独立的复位引脚，不过在默认状态下，第 3 脚是复位端口(即第一功能为复位功能)。

LPC1114 系列内部自带有 12MHz 的 RC 振荡器，并且在其上电(或复位)时默认使用片内的 RC 为主时钟源。所以 LPC1114 在默认情况下，上电就可运行。不过内部 RC 振荡器的精度并不高(一般为 1%)，所以在要求较高时还是要使用外部晶振来作为振荡时钟。图 1-3 给出了一个使用外部晶振的最小系统。

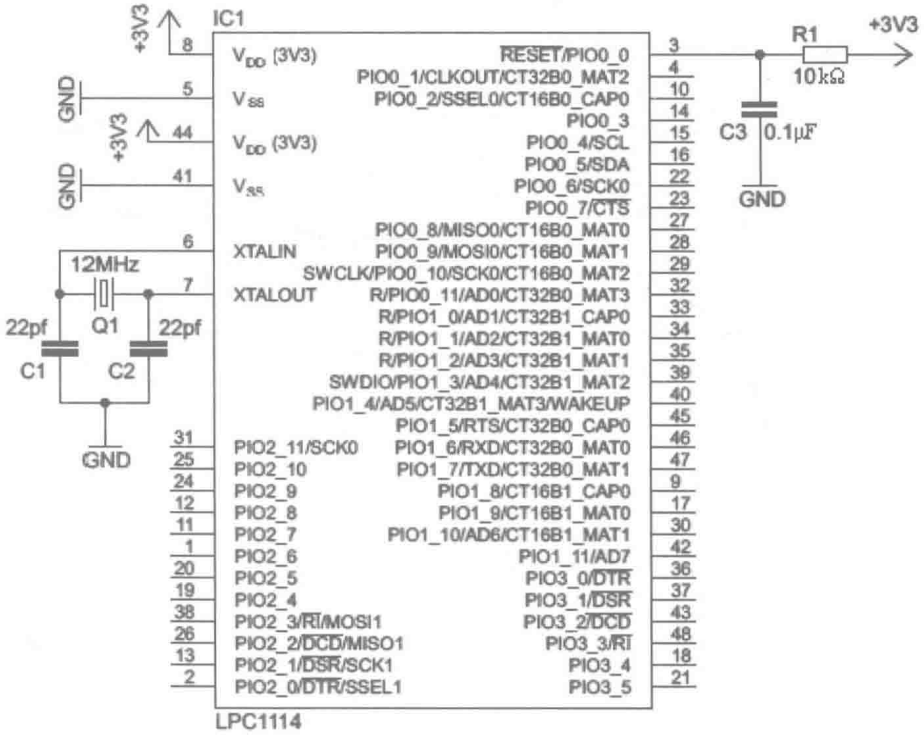


图 1-3　LPC1114 最小系统原理图

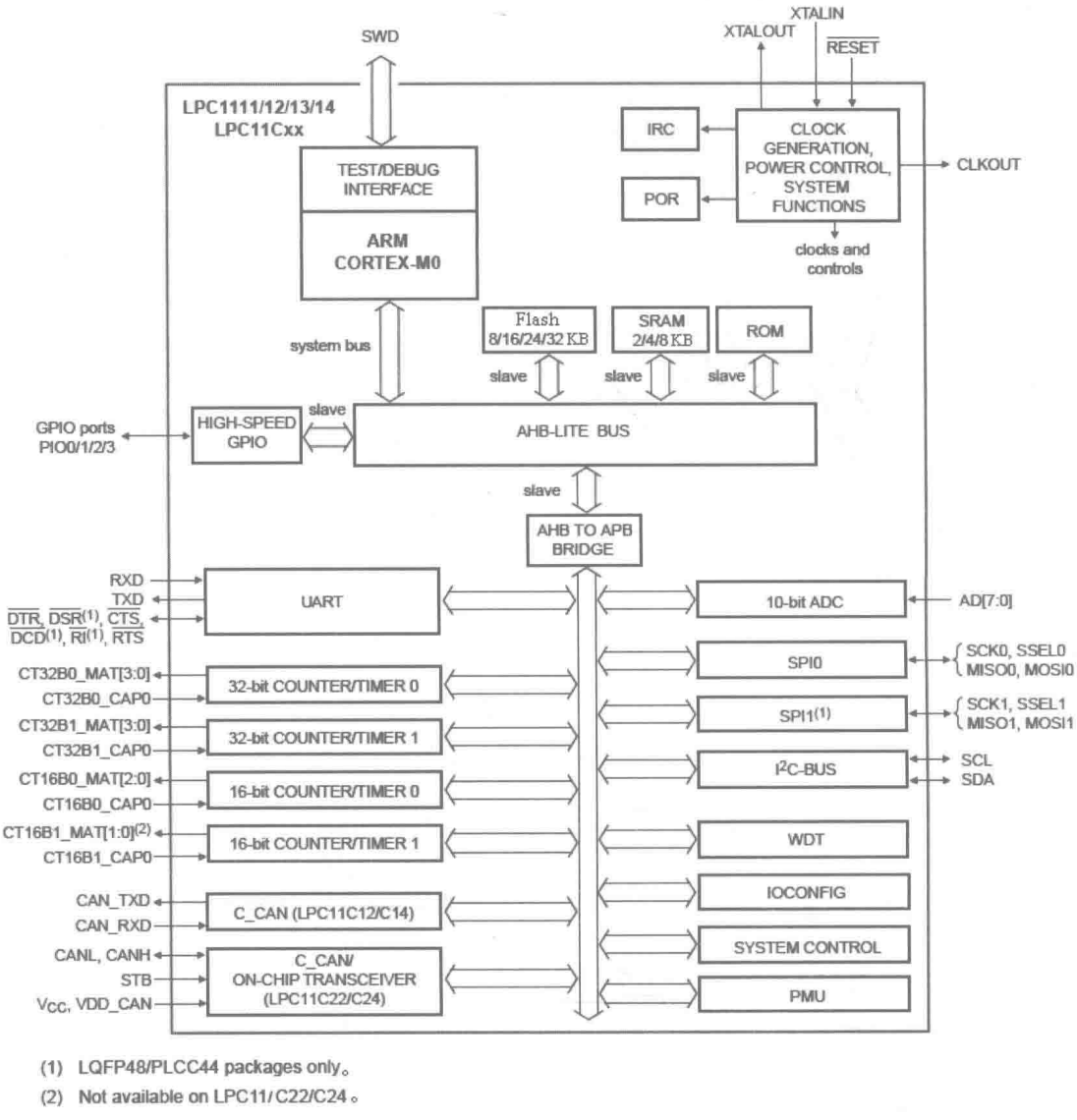
图 1-4 给出了 LPC1114 芯片的内部结构。

从图 1-4 中可以看到，在 LPC1114 微控制器芯片中，最核心的部分是一个 Cortex-M0 的 ARM 内核。

Cortex-M0 内核是 ARM 公司为适应低端市场的需求而专门设计的一个入门级的 32 位 ARM 处理器，它目前被广泛地应用于各类嵌入式系统中。该处理器具有结构简单、容易学习和编程、功耗极低、运算效率高、代码密度出色、高性能中断处理、向上兼容 Cortex-M 系列处理器等特性。

Cortex-M0 采用 ARMv6-M 结构，基于 16 位 Thumb 指令集，并包含 Thumb-2 技术。所以能提供一个现代 32 位体系结构处理器所希望的出色性能，代码密度比其他 8 位和 16 位微控制器都要高。

ARM Cortex-M0 处理器基于一个高集成度、低功耗的 32 位处理器内核，采用 3 级流水线的冯·诺依曼结构。通过简单、功能强大的指令集以及全面优化的设计(提供包括一



(1) LQFP48/PLCC44 packages only。

(2) Not available on LPC11/C22/C24。

图 1-4　LPC1114 内部结构图

个单周期乘法器在内的高端处理硬件),ARM Cortex-M0 处理器可实现极高的能效。

NXP 公司基于 ARM Cortex-M0 内核设计了 LPC11xx 系列微控制器,并加入了大量功能丰富的外设和通信接口,给嵌入式开发者提供了极大的便利。

1.2　MDK 开发环境及其配置

开发基于 LPC1114 的系统要选择一个合适的开发工具。对于 NXP 系列 CPU,恩智浦公司专门委托 code_red 公司为其量身定制了一款基于开源项目 Eclipse 的开发工具,称为 LPCXpresso。该软件可以通过 code_red 公司网站注册后免费下载并激活。其他的开发工具,基本上都是基于 ARM 的通用开发环境,常见的有 ARM 公司著名的 Real

View MDK(MDK-ARM)、IAR 公司的 EWARM 等。本书选用国内使用非常广泛的 MDK-ARM 作为 LPC1114 的开发工具,其他开发工具读者可自行研究使用,它们都有各自的特点。

Real View MDK 的前身其实就是著名的 Keil μVision,后来它被 ARM 公司收购后就结合 ARM 系统推出了 MDK-ARM。熟悉 Keil μVision 的用户会发现两者的界面基本上是一样的,使用起来非常方便。

本书使用的是 MDK-ARM4.22a 版本,它结合了程序编辑、编译、查错、调试、仿真等功能为一体,功能强大,使用方便,熟悉 51 单片机开发的读者可以很快上手。图 1-5 是它的运行界面,可看出它基本上就是 Keil C51 的界面。

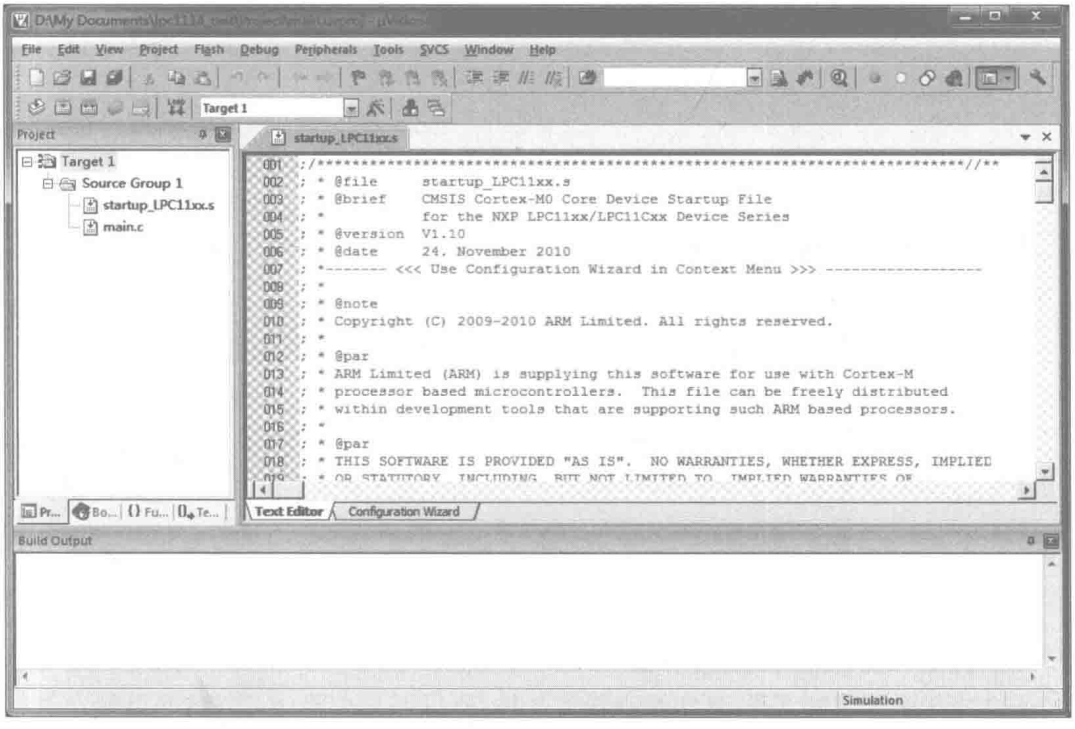


图 1-5 MDK-ARM 的运行界面

MDK-ARM 开发环境的配置非常简单,下面就用它来具体配置一个 LPC1114 的开发环境。

配置步骤如下。

(1) 安装 MDK-ARM4.22a 版本,输入许可证码将其激活(详略)。

(2) 新建一个工程,单击 Project→New μVision Project,在弹出的对话框中选择一个路径并输入一个工程名称(如 test,默认扩展名为 μvproj),单击“保存”按钮。

(3) 在弹出的器件选择对话框中选择 NXP 下的 LPC1114/301,并单击 OK 按钮(见图 1-6)。

(4) 接下来会弹出一个如图 1-7 所示的对话框,询问是否要把启动代码(Startup Code)加入到工程中,单击“是”按钮。

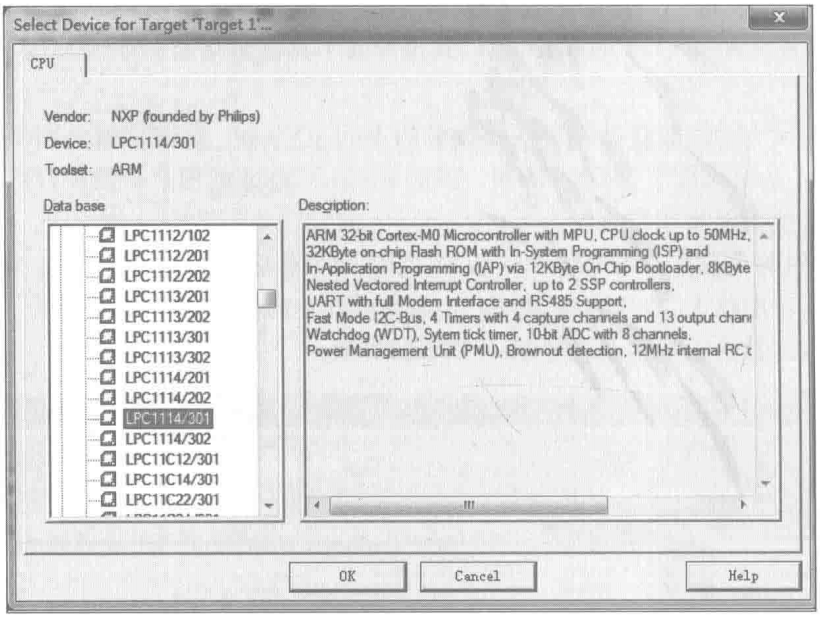


图 1-6　器件选择对话框

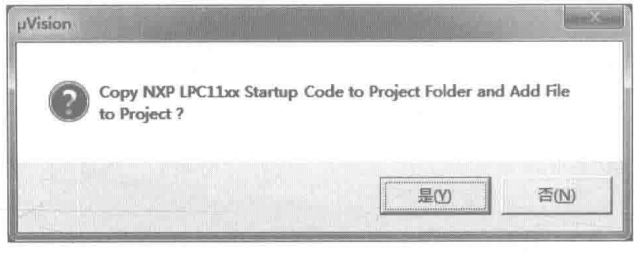


图 1-7　复制启动代码对话框

(5) 完成后就可进入到工作界面，如图 1-8 所示。

(6) 由于是新建的工程，其中没有任何文件，所以下一步要加入一些必要的文件(如头文件等)，同时要新建主程序文件并把它也添加到工程中来。

(7) 新建主程序文件，单击 File→New，可看到在工作区域内新建了一个默认名称为 Text1 的文本文件，如图 1-9 所示。

(8) 接下来保存该主程序文件，单击 File→Save，在弹出的对话框中选择好保存路径，然后在文件名中为主程序文件取一个名称，注意名称一定要加上扩展名，如本例中为 c，单击“保存”按钮。

(9) 最后一步要把刚才新建的主程序文件添加到工程中来，单击左侧 Target1 中的加号把它展开，然后在其下面的 Source Group 1 上右击，在弹出的快捷菜单中选择 Add Files to Group 'Source Group 1'一项，如图 1-10 所示。

(10) 在弹出的对话框中找到刚才保存的主程序文件，单击 Add 按钮将其添加到工程中(见图 1-11)。这里要注意，若找不到主程序文件可能有以下几个原因：一是刚才保存时没有为主程序文件加上扩展名 c；二是对话框中的筛选条件不对(文件类型应为 C

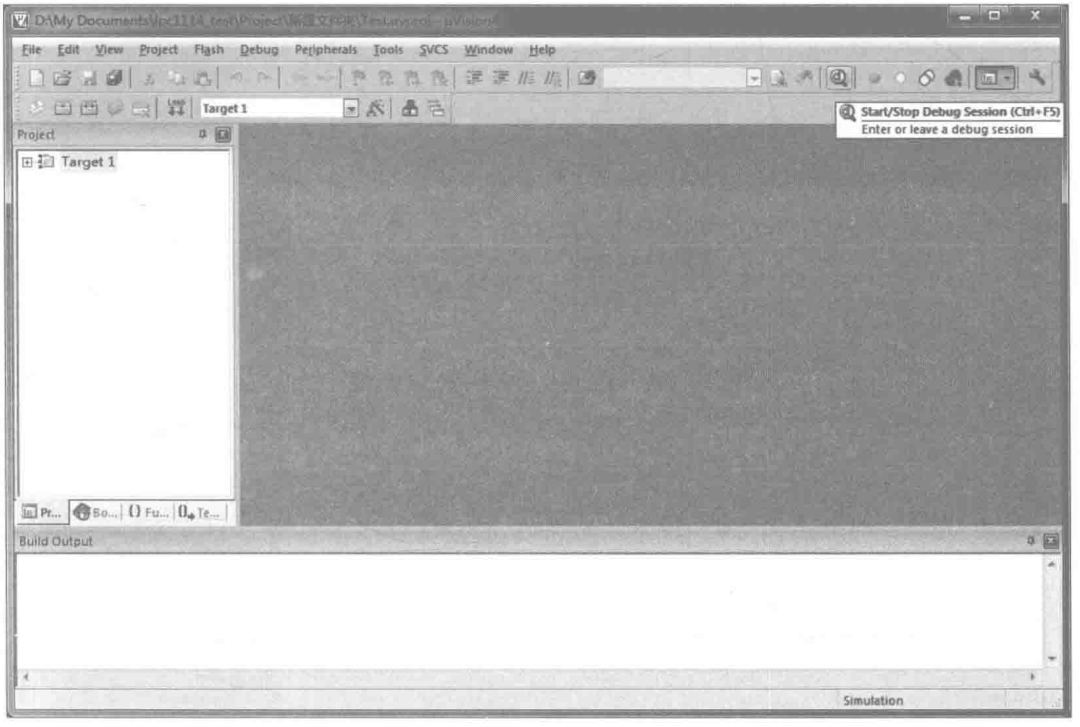


图 1-8　工作界面

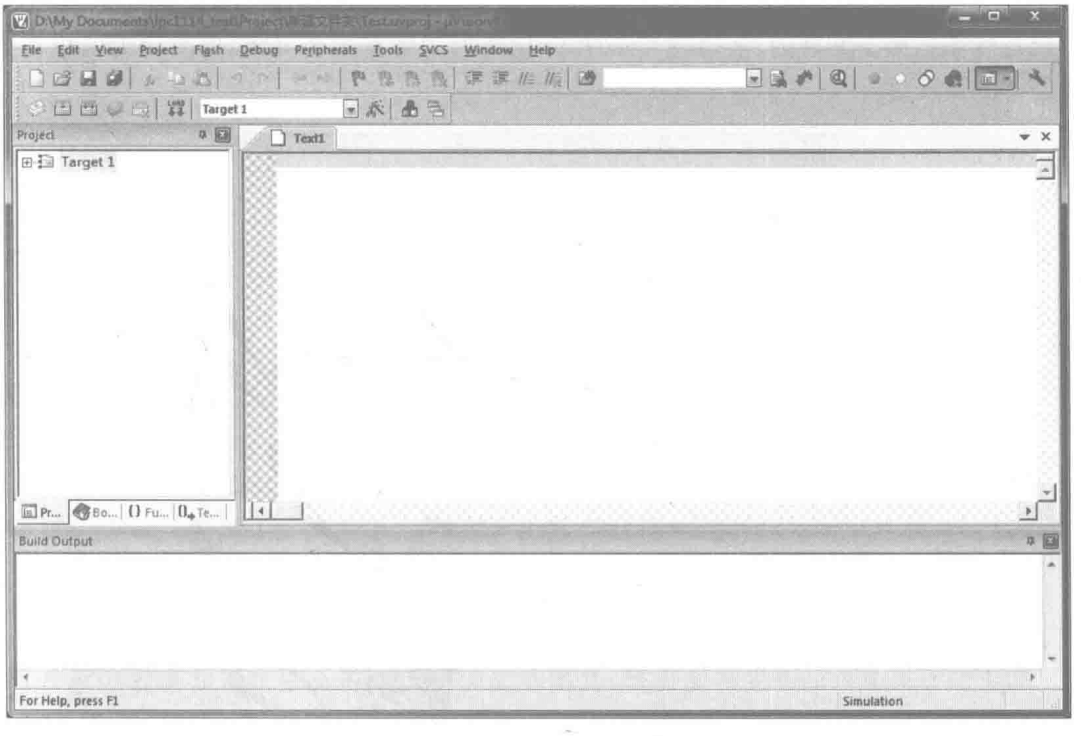


图 1-9　新建主程序文本