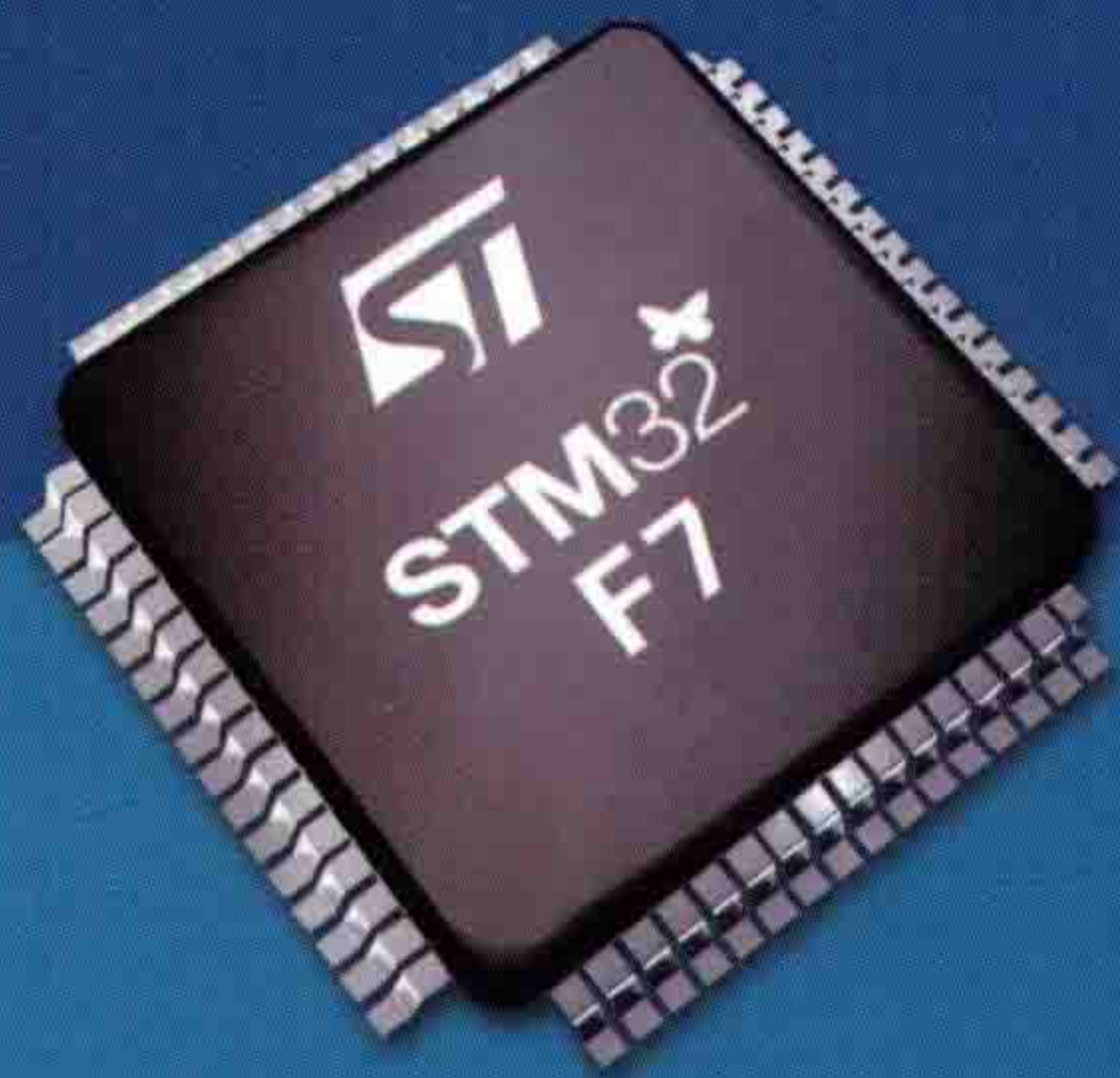


正点原子教你学嵌入式系统丛书

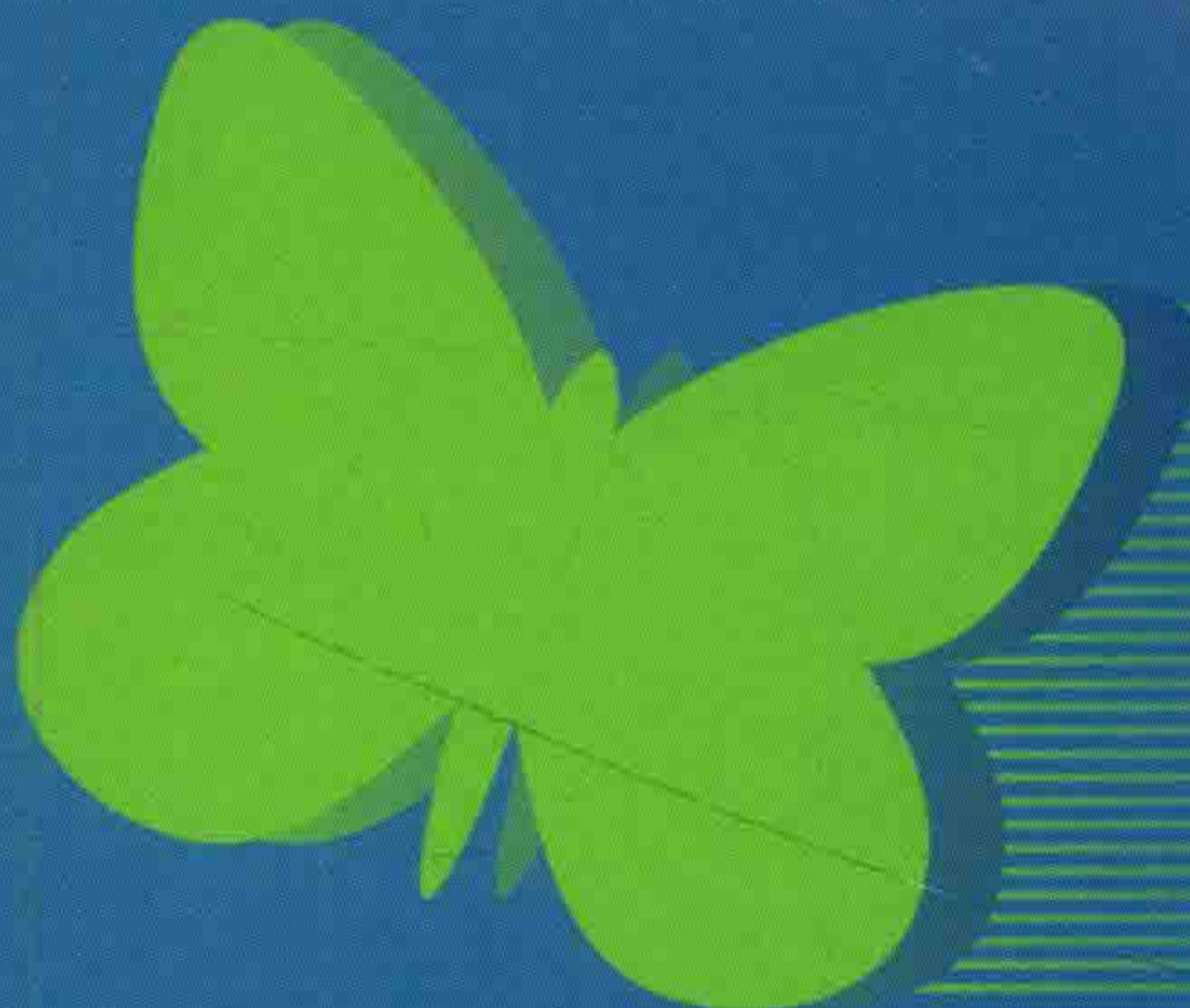
STM32F7

原理与应用

——寄存器版 (下)



刘 军 张 洋 左忠凯 编著



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

STM32F7 原理与应用

——寄存器版(下)

军 张 洋 左忠凯 编著

北京航空航天大学出版社

内 容 简 介

本套书籍以 ST 公司的 STM32F767 为目标芯片,详细介绍了 STM32F7 的特点、片内外资源的使用,并辅以 65 个例程,由浅入深地介绍了 STM32F7 的使用。所有例程都经过精心编写,从原理开始介绍,到代码编写、下载验证,一步步教读者如何实现。所有源码都配有详细注释,且经过严格测试。另外,源码有生成好的 hex 文件,读者只需要通过仿真器下载到开发板即可看到实验现象,亲自体验实验过程。

套书总共分为 4 册:《STM32F7 原理与应用——寄存器版(上)》、《STM32F7 原理与应用——寄存器版(下)》、《STM32F7 原理与应用——HAL 库版(上)》和《STM32F7 原理与应用——HAL 库版(下)》。

本书为《STM32F7 原理与应用——寄存器版(下)》,共 35 章,通过 35 个高级实验例程(对应实验 31~实验 65),带领大家深入了解 STM32F7 的使用。对于没有学过 STM32 的初学者,强烈建议先阅读上册内容,再来学习本书内容。

本书适合 STM32F7 初学者学习参考,对有一定经验的电子工程技术人员也具有实用参考价值。本书也可以作为高等院校电子、通信、计算机、信息等相关专业的教学参考用书。

图书在版编目(CIP)数据

STM32F7 原理与应用:寄存器版,下 / 刘军,张洋,
左忠凯编著. -- 北京:北京航空航天大学出版社,
2017.5

ISBN 978-7-5124-2389-3

I. ①S… II. ①刘… ②张… ③左… III. ①微控制
器 IV. ①TP332.3

中国版本图书馆 CIP 数据核字(2017)第 079251 号

版权所有,侵权必究。

STM32F7 原理与应用——寄存器版(下)

刘 军 张 洋 左忠凯 编著

责任编辑 董立娟

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@buaacm.com.cn 邮购电话:(010)82316936

北京泽宇印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:32.5 字数:731 千字

2017 年 6 月第 1 版 2017 年 6 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-2389-3 定价:79.00 元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

引入寄存器编程方式, 但寄存器编程方式更灵活, 更适合于对硬件资源有较高要求的应用。

本书共分两册, 上册为寄存器版, 下册为 HAL 库版。本书可作为 STM32 开发人员的参考书, 也可作为高等院校相关专业教材。

套书序言

2014 年底, 意法半导体(ST)发布了 STM32F7 系列芯片。该芯片采用 ARM 公司最近发布的最新、最强的 ARM Cortex-M7 内核, 其性能约为意法半导体原有最强处理器 STM32F4(采用 ARM Cortex-M4 内核)的两倍。STM32F7 系列微控制器的工作频率高达 216 MHz, 采用 6 级超标量流水线和硬件浮点单元(Floating Point Unit, FPU), 测试分数高达 1 000 CoreMark。

在 ST MCU 高级市场部经理曹锦东先生的帮助下, 作者有幸于 2015 年拿到了 STM32F7 的样片和评估板。STM32F7 强大的处理能力以及丰富的外设资源足以应付各种需求, 在工业控制、音频处理、智能家居、物联网和汽车电子等领域, 有着广泛的应用前景。其强大的 DSP 处理性能足以替代一部分 DSP 处理器, 在中高端通用处理器市场有很强的竞争力。

由于 STM32F7 和 ARM Cortex-M7 公布都不久, 除了 ST 官方的 STM32F7 文档和源码, 网络上很少有相关的教程和代码, 遇到问题时也很少有人可以讨论。作为 STM32F7 在国内较早的使用者, 作者经过近两年的学习和研究, 将 STM32F7 的所有资源摸索了一遍, 在此过程中, 发现并解决了不少 bug。为了让没接触过 STM32F7 的朋友更快、更好地掌握 STM32F7, 作者设计了一款 STM32F7 开发板(阿波罗 STM32F767 开发板), 并对 STM32F7 的绝大部分资源编写了例程和详细教程。这些教程浅显易懂, 使用的描述语言很自然, 而且图文并茂, 每一个知识点都设计了一个可以运行的示例程序, 非常适合初学者学习。

时至今日, 书已成型, 两年的时间包含了太多的心酸与喜悦, 最终呈现给读者的是包括:《STM32F7 原理与应用——寄存器版(上)》、《STM32F7 原理与应用——寄存器版(下)》、《STM32F7 原理与应用——HAL 库版(上)》和《STM32F7 原理与应用——HAL 库版(下)》共 4 本书的一套书籍。这主要有以下几点考虑:

① STM32F7 的代码编写有两种方式:寄存器和 HAL 库。寄存器方式编写的代码具有精简、高效的特点, 但是需要程序员对相关寄存器比较熟悉;HAL 库方式编写的代码具有简单、易用的特点, 但是效率低, 代码量较大。一般想深入学习了解的话, 建议选择寄存器方式;想快速上手的话, 建议选择 HAL 库方式。实际应用中, 这两种方式都有很多朋友选择, 所以分为寄存器和库函数两个版本出版。

② STM32F7 的功能十分强大, 外设资源也非常丰富, 因此教程篇幅也相对较大, 而一本书的厚度是有限的, 无法将所有内容都编到一本书上, 于是分成上下两册。

由于 STM32F7 的知识点非常多,即便分成上下两册,对很多方面也没有深入探讨,需要后续继续研究,而一旦有新的内容,我们将尽快更新到开源电子网(www.openedv.com)。

STM32F7 简介

STM32F7 是 ST 公司推出的第一款基于 ARM Cortex-M7 内核的微处理器,具有 6 级流水线、硬件单/双精度浮点计算单元、L1 I/D Cache、支持 Flash 零等待运行代码、支持 DSP 指令、主频高达 216 MHz,实际性能是 STM32F4 的两倍;另外,还有 QSPI、FMC、TFTLCD 控制器、SAI、SPDIF、硬件 JPEG 编解码器等外设,资源十分丰富。

套书特色

本套书籍作为学习 STM32F7 的入门级教材,也是市面上第一套系统地介绍 STM32F7 原理和应用的教材,具有如下特色:

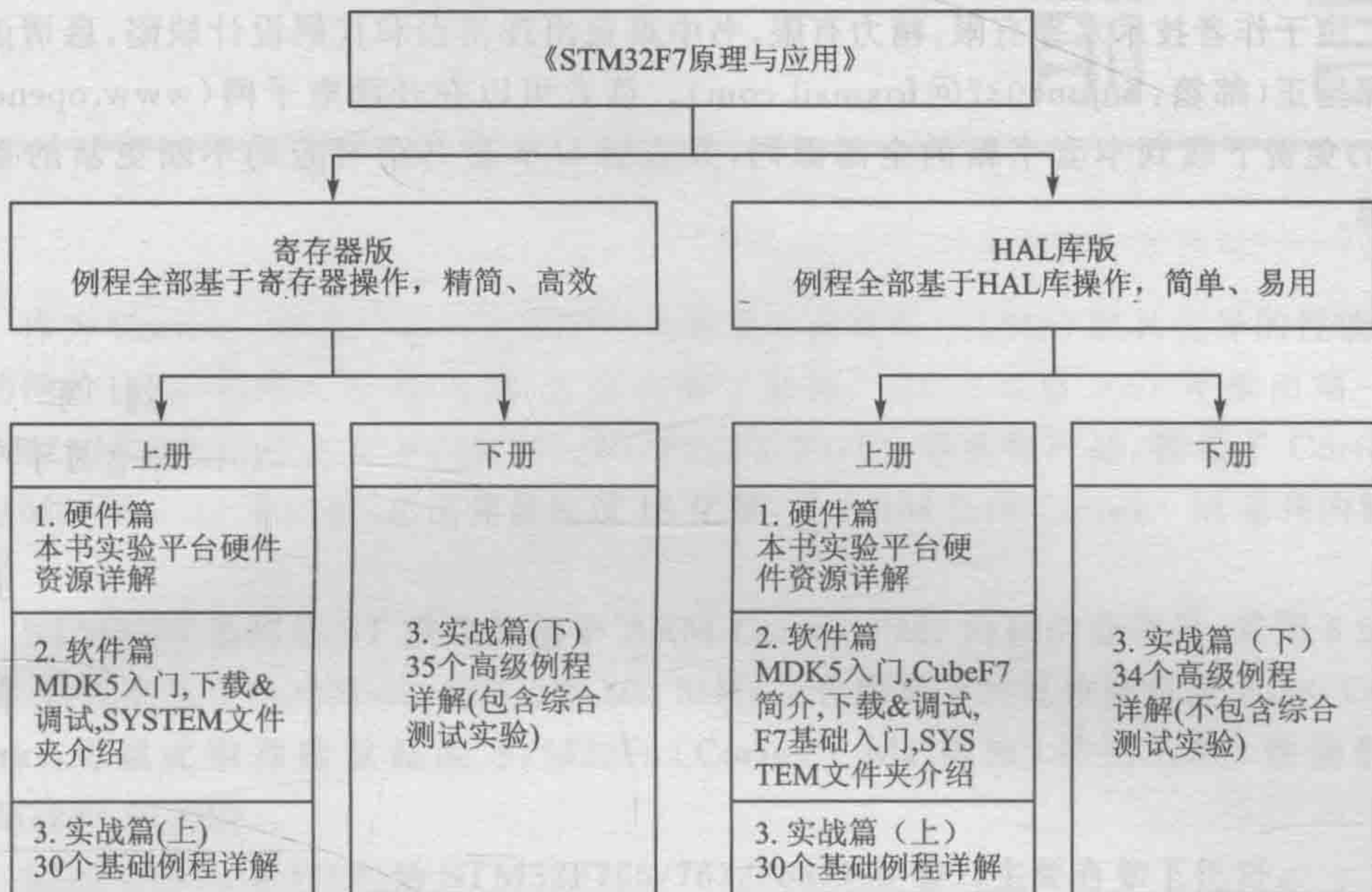
- 最新。新芯片,使用最新的 STM32F767 芯片;新编译器,使用最新的 MDK5.21 编译器;新库,基于 ST 主推的 HAL 库编写(HAL 库版)代码,不再使用标准库。
- 最全。书中包含了大量例程,基本上 STM32F7 的所有资源都有对应的实例,每个实例都从原理开始讲解→硬件设计→软件设计→结果测试,详细介绍了每个步骤,力求全面掌握各个知识点。
- 循序渐进。书本从实验平台开始→硬件资源介绍→软件使用介绍→基础知识讲解→例程讲解,一步一步地学习 STM32F7,力求做到心中有数,循序渐进。
- 由简入难。书本例程从最基础的跑马灯开始→最复杂的综合实验,由简入难,一步步深入,完成对 STM32F7 各个知识点的学习。
- 无限更新。由于书本的特殊性,无法随时更新,一旦有新知识点的教程和代码,作者都会发布在开源电子网(www.openedv.com),读者多关注即可。

套书结构

本套书籍一共分为 2 个版本,共 4 本:《STM32F7 原理与应用——寄存器版(上)》、《STM32F7 原理与应用——寄存器版(下)》、《STM32F7 原理与应用——HAL 库版(上)》和《STM32F7 原理与应用——HAL 库版(下)》。其中,寄存器版本全部基于寄存器操作,精简高效,适合深入学习和研究;HAL 库版本全部采用 HAL 库操作,简单易用,适合快速掌握和使用。上册详细介绍了实验平台的硬件、开发软件的入门和使用、新建工程、下载调试和 30 个基础例程,并且这 30 个基础例程绝大部分都是针对 STM32F7 内部一些基本外设的使用,比较容易掌握,也是灵活使用 STM32F7 的基础。对于想入门,或者刚接触 STM32F7 的朋友,上册版本是您的理想之选。下册则详细介绍了 34/35(寄存器版多了综合实验)个高级例程,针对 STM32F7 内部的一些高级外设和第三方代码(FATFS、Lwip、μC/OS 和音频解码库等)的使用等做了详细介绍,对学

习者要求比较高,适合对 STM32F7 有一定了解、基础比较扎实的朋友学习。

本套书籍的结构如下所示:



使用本套书籍

对于时间充足、有过单片机使用经验、对底层驱动感兴趣的朋友,建议选择寄存器版本学习。因为它全部是基于最底层的寄存器操作,对学习者要求比较高,需要较多的时间来掌握,但是学会之后,编写代码思路会清晰很多,而且代码精简,效率极高。

对于想快速入门、对底层接口兴趣不大,专注应用层软件的朋友,建议选择 HAL 库版本学习。因为它的底层驱动,全部由 ST 官方写好了,读者只须学会函数和参数的使用,就能实现对相关外设的驱动,有利于快速编写驱动代码,无须繁琐地查看寄存器,容易入门,能有更多的时间来实现应用层的功能。

对于没有学习过 STM32F7 的初学者,建议先学习上册的内容,它对 STM32F7 的软硬件开发环境进行了详细的介绍,从新建工程教起,包括 30 个 STM32F7 内部资源使用的基础例程,每个例程都有详细的解说和示例程序,非常适合初学者入门。

对于有一定单片机编程基础、对 STM32F7 有一定了解(最好学过本套书籍上册内容)、想进一步提高的朋友,推荐学习下册内容,它对 STM32F7 的一些高级外设详细介绍和参考代码,并且对第三方代码组件也有比较详细的介绍,非常适合较大工程的应用。

致 谢

感谢北京航空航天大学出版社,它的支持才让本套书籍得以和大家见面。

感谢开源电子网的网友,是他们的支持和帮助才让我一步一步走了下来,其中有一些朋友还参与了本套书籍的审校和代码审核工作,特别感谢:八度空间、春风、jerymy_

2017年2月于广州

前言

作为 Cortex-M 系列通用处理器市场的最大占有者,STM32 以其优异的性能、超高的性价比、丰富的本地化教程,迅速占领了市场。ST 公司自 2007 年推出第一款 STM32 以来,先后推出了 STM32F0/F1/F2/F3/F4/F7 等系列产品,涵盖了 Cortex-M0/M3/M4/M7 等内核,总出货量超过 18 亿颗,是 ARM 公司 Cortex-M 系列内核的霸主。

STM32F7 系列是 ST 推出的基于 ARM Cortex-M7 内核的处理器,采用 6 级流水线,性能高达 5 CoreMark/MHz,在 200 MHz 工作频率下测试数据高达 1 000 CoreMarks,远超此前性能最高的 STM32F4 (Cortex-M4 内核) 系列 (DSP 性能超过 STM32F4 的两倍)。

STM32F76x 系列 (包括 STM32F765/767/768/769 等),主要有如下优势:

- 更先进的内核,采用 Cortex-M7 内核,具有 16 KB 指令/数据 Cache,采用 ST 独有的自适应实时加速技术 (ART Accelerator),性能高达 5 CoreMark/MHz。
- 更丰富的外设,拥有高达 512 KB 的片内 SRAM,并且支持 SDRAM、带 TFTLCD 控制器、带图形加速器 (Chorme ART)、带摄像头接口 (DCMI)、带硬件 JPEG 编解码器、带 QSPI 接口、带 SAI&I²S 音频接口、带 SPDIF RX 接口、USB 高速 OTG、真随机数发生器、OTP 存储器等。
- 更高的性能,STM32F767 最高运行频率可达 216 MHz,具有 6 级流水线,带有指令和数据 Cache,大大提高了性能,性能大概是 STM32F4 的两倍。而且 STM32F76x 自带了双精度硬件浮点单元 (DFFPU),在做 DSP 处理的时候具有更好的性能。

STM32F76x 系列自带了 LCD 控制器和 SDRAM 接口,对于想要驱动大屏或需要大内存的朋友来说,是个非常不错的选择;更重要的是集成了硬件 JPEG 编解码器,可以秒解 JPEG 图片,做界面的时候可以大大提高加载速度,并且可以实现视频播放。本书将以 STM32F767 为例,向大家讲解 STM32F7 的学习。

内容特点

学习 STM32F767 有几份资料经常用到:《STM32F7 中文参考手册》、《STM32F7xx 参考手册》英文版、《STM32F7 编程手册》。

其中,最常用的是《STM32F7 中文参考手册》。该文档是 ST 官方针对 STM32F74x/75x 的一份中文参考资料,里面有绝大部分寄存器的详细描述,内容翔

实,但是没有实例,也没有对 Cortex-M7 构架进行大多介绍,读者只能根据自己对书本的理解来编写相关代码。另外,对 STM32F767 特有的部分外设(比如硬件 JPEG 编解码器、DFSDM 等),则必须参考《STM32F7xx 参考手册》英文版来学习。

《STM32F7 编程手册》文档则重点介绍了 Cortex-M7 内核的汇编指令及其使用、内核相关寄存器(比如 SCB、NVIC、SYSTICK 等寄存器)是《STM32F7 中文参考手册》的重要补充。很多在《STM32F7 中文参考手册》无法找到的内容,都可以在这里找到答案,不过目前该文档没有中文版本,只有英文版。

本书将结合以上 3 份资料,从寄存器级别出发,深入浅出地向读者展示 STM32F767 的各种功能。总共配有 65 个实例,基本上每个实例均配有软硬件设计,在介绍完软硬件之后马上附上实例代码,并带有详细注释及说明,让读者快速理解代码。

这些实例涵盖了 STM32F7 的绝大部分内部资源,并且提供了很多实用级别的程序,如内存管理、NAND Flash FTL、拼音输入法、手写识别、图片解码、IAP 等。所有实例均在 MDK5.21A 编译器下编译通过,读者只须下载程序到 ALIENTEK 阿波罗 STM32 开发板即可验证实验。

读者对象

不管你是一个 STM32 初学者,还是一个老手,本书都非常适合。尤其对于初学者,本书将手把手地教你如何使用 MDK,包括新建工程、编译、仿真、下载调试等一系列步骤,让你轻松上手。本书不适用于想通过 HAL 库学习 STM32F7 的读者,因为本书的绝大部分内容都是直接操作寄存器的;如果想通过 HAL 库学习 STM32F7,可看本套书的 HAL 库版本。

配套资料

本书的实验平台是 ALIENTEK 阿波罗 STM32F7 开发板,有这款开发板的朋友可以直接拿本书配套资料上的例程在开发板上运行、验证。而没有这款开发板而又想要的朋友,可以上淘宝购买。当然,如果已有了一款自己的开发板,而又不想再买,也是可以的,只要你的板子上有和 ALIENTEK 阿波罗 STM32F7 开发板上的相同资源(需要实验用到的),代码一般都是可以通用的,你需要做的就只是把底层的驱动函数(比如 I/O 口修改)稍做修改,使之适合你的开发板即可。

本书配套资料包括 ALIENTEK 阿波罗 STM32F7 开发板相关模块原理图(pdf 格式)、视频教程、文档教程、配套软件、各例程序源码和相关参考资料等,所有这些资料读者都可以在 <http://www.openedv.com/thread-13912-1-1.html> 免费下载。

刘 军

2017 年 2 月于广州



第1章	触摸屏实验	1
第2章	红外遥控实验	21
第3章	数字温度传感器 DS18B20 实验	29
第4章	数字温湿度传感器 DHT11 实验	37
第5章	9轴传感器 MPU9250 实验	44
第6章	无线通信实验	65
第7章	Flash 模拟 EEPROM 实验	78
第8章	摄像头实验	91
第9章	内存管理实验	118
第10章	SD 卡实验	128
第11章	NAND Flash 实验	151
第12章	FATFS 实验	182
第13章	汉字显示实验	196
第14章	图片显示实验	212
第15章	硬件 JPEG 解码实验	224
第16章	照相机实验	247
第17章	音乐播放器实验	262
第18章	录音机实验	290
第19章	SPDIF(光纤音频)实验	303
第20章	视频播放器实验	323
第21章	FPU 测试(Julia 分形)实验	343
第22章	DSP 测试实验	350
第23章	手写识别实验	363
第24章	T9 拼音输入法实验	373
第25章	串口 IAP 实验	384
第26章	USB 读卡器(Slave)实验	398
第27章	USB 声卡(Slave)实验	411
第28章	USB 虚拟串口(Slave)实验	419

第 29 章	USB U 盘(Host)实验	427
第 30 章	USB 鼠标键盘(Host)实验	435
第 31 章	网络通信实验	444
第 32 章	$\mu\text{C}/\text{OS}-\text{II}$ 实验 1——任务调度	463
第 33 章	$\mu\text{C}/\text{OS}-\text{II}$ 实验 2——信号量和邮箱	473
第 34 章	$\mu\text{C}/\text{OS}-\text{II}$ 实验 3——消息队列、信号量集和软件定时器	483
第 35 章	阿波罗 STM32F767 开发板综合实验	501
参考文献		507

第 1 章

触摸屏实验

本章将介绍如何使用 STM32F767 来驱动触摸屏。ALIENTEK 阿波罗 STM32F767 开发板本身并没有触摸屏控制器,但是它支持触摸屏,可以通过外接带触摸屏的 LCD 模块(比如 ALIENTEK LCD 模块)来实现触摸屏控制。本章将介绍 STM32 控制 ALIENTEK LCD 模块(包括电阻触摸与电容触摸)实现触摸屏驱动,最终实现一个手写板的功能。

1.1 触摸屏简介

目前最常用的触摸屏有两种:电阻式触摸屏与电容式触摸屏,下面来分别介绍。

1.1.1 电阻式触摸屏

iPhone 面世之前,几乎清一色的都是使用电阻式触摸屏。电阻式触摸屏利用压力感应进行触点检测控制,需要直接应力接触,通过检测电阻来定位触摸位置。ALIENTEK 2.4/2.8/3.5 寸 LCD 模块自带的触摸屏都属于电阻式触摸屏,下面简单介绍下电阻式触摸屏的原理。

电阻触摸屏的主要部分是一块与显示器表面非常配合的电阻薄膜屏,这是一种多层的复合薄膜,以一层玻璃或硬塑料平板作为基层,表面涂有一层透明氧化金属(透明的导电电阻)导电层,上面再盖有一层外表面硬化处理、光滑防擦的塑料层,它的内表面也涂有一层涂层,它们之间有许多细小的(小于 1/1000 寸)透明隔离点把两层导电层隔开绝缘。当手指触摸屏幕时,两层导电层在触摸点位置就有了接触,电阻发生变化,在 X 和 Y 两个方向上产生信号,然后送触摸屏控制器。控制器侦测到这一接触并计算出(X,Y)的位置,再根据获得的位置模拟鼠标的方式运作。这就是电阻技术触摸屏的最基本原理。

➤ 电阻触摸屏的优点:精度高、价格便宜、抗干扰能力强、稳定性好。

➤ 电阻触摸屏的缺点:容易被划伤、透光性不太好、不支持多点触摸。

从以上介绍可知,触摸屏都需要一个 A/D 转换器,一般来说是需要一个控制器的。ALIENTEK LCD 模块选择的是 4 线电阻式触摸屏,这种触摸屏的控制芯片有很多,包括 ADS7843、ADS7846、TSC2046、XPT2046 和 AK4182 等。这几款芯片的驱动基本上是一样的,也就是说,只要写出了 ADS7843 的驱动,这个驱动对其他几个芯片也是有效

的。而且封装也有一样的,完全 PINTOPIN 兼容,所以替换起来很方便。

ALIEN TEK LCD 模块自带的触摸屏控制芯片为 XPT2046。XPT2046 是一款 4 导线制触摸屏控制器,内含 12 位分辨率、125 kHz 转换速率逐步逼近型 A/D 转换器。XPT2046 支持 1.5~5.25 V 的低电压 I/O 接口。XPT2046 能通过执行两次 A/D 转换查出被按的屏幕位置,除此之外,还可以测量加在触摸屏上的压力。内部自带 2.5 V 参考电压可以作为辅助输入、温度测量和电池监测模式,电池监测的电压范围为 0~6 V。XPT2046 片内集成有一个温度传感器。在 2.7 V 的典型工作状态下关闭参考电压,功耗可小于 0.75 mW。XPT2046 采用微小的封装形式 TSSOP-16、QFN-16(0.75 mm 厚度)和 VFBGA-48,工作温度范围为 -40~+85 °C。

该芯片完全兼容 ADS7843 和 ADS7846,详细使用可以参考这两个芯片的 datasheet。

1.1.2 电容式触摸屏

现在几乎所有智能手机,包括平板计算机,都是采用电容屏作为触摸屏。电容屏是利用人体感应进行触点检测控制,不需要直接接触或只需要轻微接触,通过检测感应电流来定位触摸坐标。

ALIEN TEK 4.3/7 寸 LCD 模块自带的触摸屏采用的是电容式触摸屏,下面简单介绍下电容式触摸屏的原理。

电容式触摸屏主要分为两种:

① 表面电容式电容触摸屏。

表面电容式触摸屏技术是利用 ITO(铟锡氧化物,是一种透明的导电材料)导电膜,通过电场感应方式感测屏幕表面的触摸行为进行。但是表面电容式触摸屏有一些局限性,它只能识别一个手指或者一次触摸。

② 投射式电容触摸屏。

投射式电容触摸屏是传感器利用触摸屏电极发射出静电场线。用于投射电容传感技术的电容类型有两种:自我电容和交互电容。

自我电容又称绝对电容,是最广为采用的一种方法,是扫描电极与地构成的电容。在玻璃表面有用 ITO 制成的横向与纵向的扫描电极,这些电极和地之间构成一个电容的两极。当用手或触摸笔触摸的时候就会并联一个电容到电路中去,从而使该条扫描线上的总体电容量有所改变。在扫描的时候,控制 IC 依次扫描纵向和横向电极,并根据扫描前后的电容变化来确定触摸点坐标位置。笔记本电脑触摸输入板就是采用这种方式,采用 XY 的传感电极阵列形成一个传感格子,当手指靠近触摸输入板时,在手指和传感电极之间产生一个小量电荷。采用特定的运算法则处理来自行、列传感器的信号,从而确定手指的位置。

交互电容又称跨越电容,是在玻璃表面的横向和纵向的 ITO 电极的交叉处形成电容。交互电容的扫描方式就是扫描每个交叉处的电容变化,从而判定触摸点的位置。当触摸时就会影响到相邻电极的耦合,从而改变交叉处的电容量。交互电容的扫描方法可以侦测到每个交叉点的电容值和触摸后电容变化,因而它需要的扫描时间与自我

电容的扫描方式相比要长一些,需要扫描检测 XY 根电极。目前,智能手机、平板计算机等的触摸屏都是采用交互电容技术。

ALIEN TEK 选择的电容触摸屏也采用投射式电容屏(交互电容类型),所以后面仅介绍投射式电容屏。

透射式电容触摸屏采用纵横两列电极组成感应矩阵,从而感应触摸。以两个交叉的电极矩阵(即 X 轴电极和 Y 轴电极)为例来检测每一格感应单元的电容变化,如图 1.1.1 所示。

图中的电极实际是透明的,这里是为了方便读者理解。图中,X、Y 轴的透明电极电容屏的精度、分辨率与 X、Y 轴的通道数有关,通道数越多,精度越高。以上就是电容触摸屏的基本原理,接下来看看电容触摸屏的优缺点:

- 电容触摸屏的优点:手感好、无须校准、支持多点触摸、透光性好。
- 电容触摸屏的缺点:成本高、精度不高、抗干扰能力差。

注意,电容触摸屏对工作环境的要求是比较高的,在潮湿、多尘、高低温环境下都是不适合使用的。

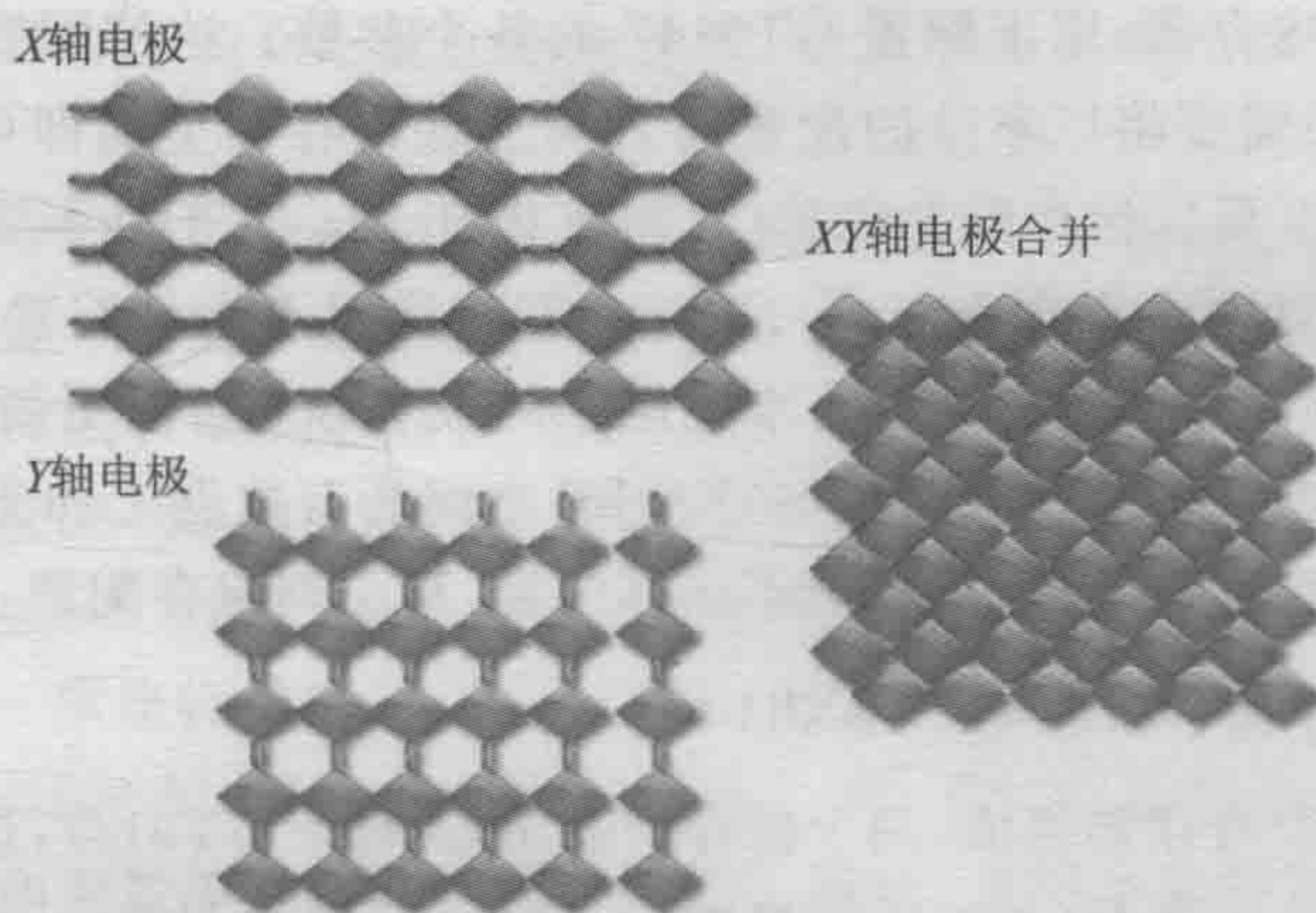


图 1.1.1 投射式电容屏电极矩阵示意图

电容触摸屏一般都需要一个驱动 IC 来检测电容触摸,且一般是通过 I²C 接口输出触摸数据的。ALIEN TEK 7 寸 LCD 模块的电容触摸屏使用 FT5206/FT5426 作为驱动 IC,采用 15×28 的驱动结构(15 个感应通道,28 个驱动通道)。ALIEN TEK 4.3 寸 LCD 模块则使用 GT9147/OTT2001A 作为驱动 IC,采用 17×10 的驱动结构(10 个感应通道,17 个驱动通道)。

这两个模块都只支持最多 5 点触摸,本例程除 CPLD 方案的 V1 版本 7 寸屏模块不支持外,其他所有 ALIEN TEK 的 LCD 模块都支持。对于电容触摸驱动 IC,这里只介绍 GT9147 的驱动,OTT2001A、FT5206 和 FT5426 的驱动同 GT9147 类似,读者参考着学习即可。

下面简单介绍 GT9147。该芯片是深圳汇顶科技研发的一颗电容触摸屏驱动 IC,支持 100 Hz 触点扫描频率、5 点触摸、18×10 个检测通道,适合小于 4.5 寸的电容触摸

屏使用。

GT9147 与 MCU 的连接是通过 4 根线实现的,即 SDA、SCL、RST 和 INT。其中,SDA 和 SCL 是 I²C 通信用的,RST 是复位脚(低电平有效),INT 是中断输出信号。I²C 的详细介绍可参考《STM32F7 原理与应用——寄存器版(上)》第 30 章的内容。

GT9147 的 I²C 地址可以是 0X14 或者 0X5D,在复位结束后的 5 ms 内,如果 INT 是高电平;则使用 0X14 作为地址;否则,使用 0X5D 作为地址,具体的设置过程参见“GT9147 数据手册.pdf”文档。本章使用 0X14 作为器件地址(不含最低位,换算成读/写命令,则是读:0X29,写:0X28),接下来介绍 GT9147 的几个重要寄存器。

1. 控制命令寄存器(0X8040)

该寄存器可以写入不同值,从而实现不同的控制。一般使用 0 和 2 这两个值,写入 2 即可软复位 GT9147,在硬复位之后,一般要往该寄存器写 2 来实行软复位。然后,写入 0,即可正常读取坐标数据(并且会结束软复位)。

2. 配置寄存器组(0X8047~0X8100)

这里共 186 个寄存器,用于配置 GT9147 的各个参数。这些配置一般由厂家提供(一个数组),所以只需要将厂家给的配置写入到这些寄存器里面即可完成 GT9147 的配置。由于 GT9147 可以保存配置信息(可写入内部 Flash,于是不需要每次上电都更新配置),这里有点须注意的地方:①0X8047 寄存器用于指示配置文件版本号,程序写入的版本号,必须大于等于 GT9147 本地保存的版本号才可以更新配置。②0X80FF 寄存器用于存储校验和,使得 0X8047~0X80FF 之间所有数据之和为 0。③0X8100 用于控制是否将配置保存在本地,写 0 则不保存配置,写 1 则保存配置。

3. 产品 ID 寄存器(0X8140~0X8143)

这里总共由 4 个寄存器组成,用于保存产品 ID。对于 GT9147,这 4 个寄存器读出来就是 9、1、4、7 这 4 个字符(ASCII 码格式)。因此,可以通过这 4 个寄存器的值来判断驱动 IC 的型号,从而判断是 OTT2001A 还是 GT9147,以便执行不同的初始化。

4. 状态寄存器(0X814E)

该寄存器各位描述如表 1.1.1 所列。

这里仅关心最高位和最低 4 位,最高位用于表示 buffer 状态,如果有数据(坐标/按键),buffer 就会是 1;最低 4 位用于表示有效触点的个数,范围是 0~5,0 表示没有触摸,5 表示有 5 点触摸。最后,该寄存器在每次读取后,如果 bit7 有效,则必须写 0,清除这个位,否则不会输出下一次数据。这个要特别注意!

表 1.1.1 状态寄存器各位描述

寄存器	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0X814E	buffer 状态	大点	接近有效	按键	有效触点个数			

5. 坐标数据寄存器(共 30 个)

这里共分成 5 组(5 个点),每组 6 个寄存器存储数据,以触点 1 的坐标数据寄存器组为例,如表 1.1.2 所列。

表 1.1.2 触点 1 坐标寄存器组描述

寄存器	bit7~0	寄存器	bit7~0
0X8150	触点 1 的 X 坐标低 8 位	0X8151	触点 1 的 X 坐标低高位
0X8152	触点 1 的 Y 坐标低 8 位	0X8153	触点 1 的 Y 坐标低高位
0X8154	触点 1 的触摸尺寸低 8 位	0X8155	触点 1 触摸尺寸高 8 位

一般只用到触点的 X、Y 坐标,所以只需要读取 0X8150~0X8153 的数据组合即可得到触点坐标。其他 4 组分别是 0X8158、0X8160、0X8168 和 0X8170 等开头的 16 个寄存器组成,分别针对触点 2~4 的坐标。GT9147 支持寄存器地址自增,只需要发送寄存器组的首地址,然后连续读取即可。GT9147 会自动地址自增,从而提高读取速度。

GT9147 相关寄存器的介绍就介绍到这里,更详细的资料可参考“GT9147 编程指南.pdf”文档。

GT9147 只需要经过简单的初始化就可以正常使用了,初始化流程:硬复位→延时 10 ms→结束硬复位→设置 I²C 地址→延时 100 ms→软复位→更新配置(需要时)→结束软复位。此时 GT9147 即可正常使用了。

然后,不停地查询 0X814E 寄存器,判断是否有有效触点,如果有,则读取坐标数据寄存器,得到触点坐标。注意,如果 0X814E 读到的值最高位为 1,就必须对该位写 0;否则,无法读到下一次坐标数据。

特别说明:FT5206 和 FT5426 的驱动代码一模一样,只是版本号读取的时候稍有差异,读坐标数据和配置等操作完全是一样的。所以,这两个电容屏驱动 IC 可以共用一个.c 文件(ft5206.c)。电容式触摸屏部分就介绍到这里。

1.2 硬件设计

本章实验功能简介:开机的时候先初始化 LCD,读取 LCD ID。随后,根据 LCD ID 判断是电阻触摸屏还是电容触摸屏,如果是电阻触摸屏,则先读取 24C02 的数据,判断触摸屏是否已经校准过。如果没有校准,则执行校准程序,校准过后进入电阻触摸屏测试程序;如果已经校准了,就直接进入电阻触摸屏测试程序。

如果是 4.3 寸电容触摸屏,则先读取芯片 ID,判断是不是 GT9147,如果是,则执行 GT9147 的初始化代码;如果不是,则执行 OTT2001A 的初始化代码。如果是 7 寸电容触摸屏(不支持采用 CPLD 驱动的 7 寸 V1 屏),则执行 FT5206 的初始化代码(兼容 FT5426);在初始化电容触摸屏完成后,进入电容触摸屏测试程序(电容触摸屏无须校准)。

电阻触摸屏测试程序和电容触摸屏测试程序基本一样,只是电容触摸屏支持最多

5 点同时触摸,电阻触摸屏只支持一点触摸,其他一模一样。测试界面的右上角会有一个清空的操作区域(RST),单击这个地方就会将输入全部清除,恢复白板状态。使用电阻触摸屏的时候,可以通过按 KEY0 来实现强制触摸屏校准,只要按下 KEY0 就会进入强制校准程序。

所要用到的硬件资源如下:指示灯 DS0、KEY0 按键、LCD 模块(带电阻/电容式触摸屏)、24C02。

所有这些资源与 STM32F767 的连接在本书上册都已经介绍了(下同),这里只针对 LCD 模块与 STM32F767 的连接端口再说明一下。LCD 模块的触摸屏(电阻触摸屏)总共有 5 根线与 STM32F767 连接,连接电路图如图 1.2.1 所示。

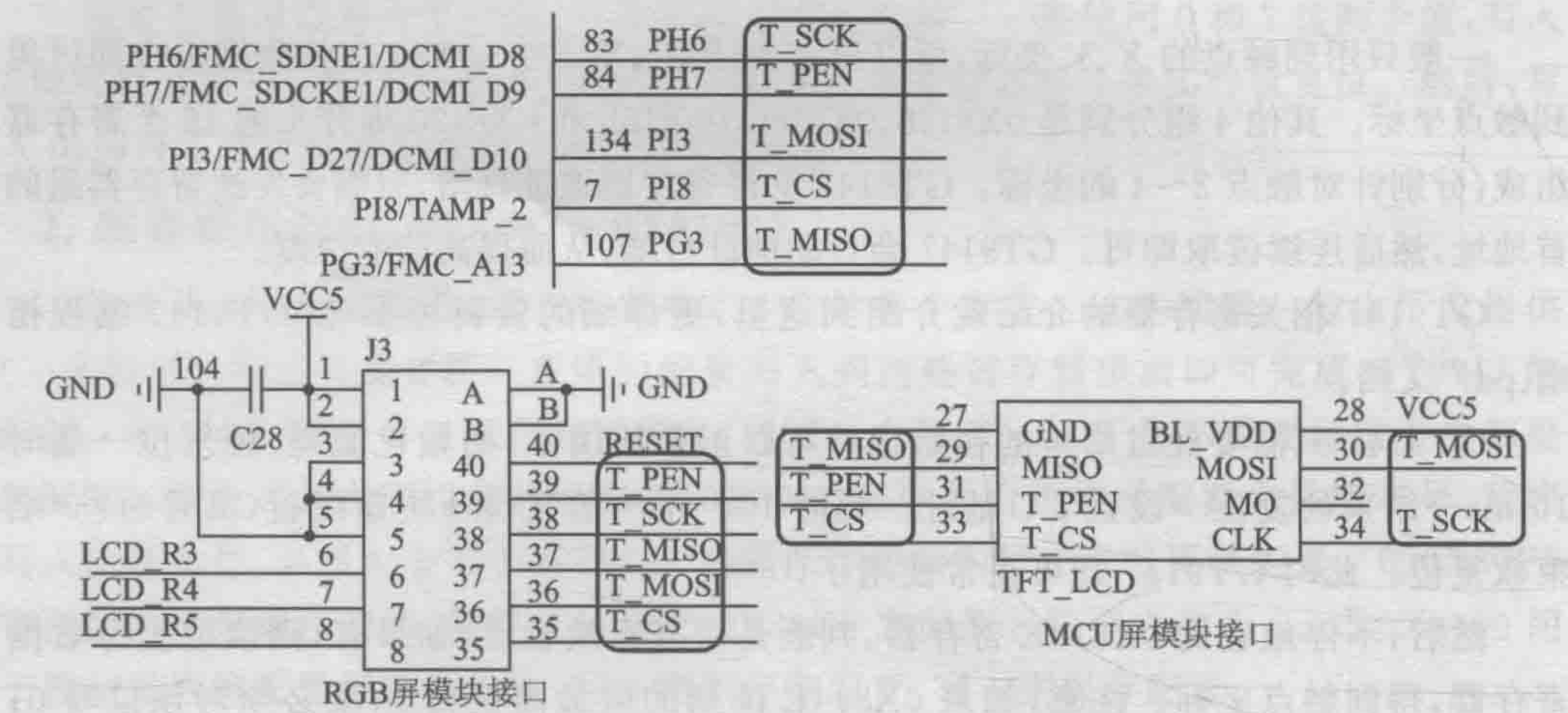


图 1.2.1 触摸屏与 STM32F767 的连接图

可以看出,T_MOSI、T_MISO、T_SCK、T_CS 和 T_PEN 分别连接在 STM32F767 的 PI3、PG3、PH6、PI8 和 PH7 上。另外,阿波罗 STM32F767 开发板有 2 种屏幕接口:RGB 屏和 MCU 屏,它们共用触摸屏接口(须分时复用)。

如果是电容式触摸屏,则接口和电阻式触摸屏一样(图 1.2.1 右侧接口),只是没有用到 5 根线了,而是 4 根线,分别是 T_PEN(CT_INT)、T_CS(CT_RST)、T_CLK(CT_SCL)和 T_MOSI(CT_SDA)。其中,CT_INT、CT_RST、CT_SCL 和 CT_SDA 分别是 OTT2001A、GT9147、FT5206 和 FT5426 的中断输出信号、复位信号,I²C 的 SCL 和 SDA 信号。这里用查询的方式读取数据,因为 OTT2001A/FT5206/FT5426 没有用到中断信号(CT_INT),所以同 STM32F767 的连接只需要 3 根线即可;不过 GT9147 还需要用到 CT_INT 做 I²C 地址设定,所以需要 4 根线连接。

1.3 软件设计

打开《STM32F7 原理与应用——寄存器版(上)》第 35 章的工程,由于本章用不到 USART 和 CAN 相关代码,所以,先去掉 USART 相关代码和 can.c(此时 HARD-