

汽车电子S32K 系列微控制器

基于ARM Cortex-M4F内核

○ 王宜怀 邵长星 黄熙 著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

汽车电子S32K 系列微控制器

基于ARM Cortex-M4F内核

● 王宜怀 邵长星 黄熙 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书基于恩智浦(NXP)公司于2017年推出的面向汽车电子的S32K系列微控制器,以构件化底层驱动为基础阐述S32K系列微控制器应用程序的设计方法,内容涉及汽车电子技术基础,S32K系列微控制器的基本特性,ARM Cortex-M4F的相关知识,底层驱动构件设计规范,以及S32K系列微控制器的Systick、PDB、LPIT、LTMR、RTC、FTM、Flash、ADC、CMP、SPI、I2C、DMA、CAN等模块的程序设计方法。

本书提供网上光盘,内含与本书相关的文档、工程框架、各章实例源程序及常用软件工具。网上光盘下载地址:<http://sumcu.suda.edu.cn>。

本书适合从事汽车电子及嵌入式技术的研发人员、工程师、高校教师、研究生、高年级本科生使用,也可作为嵌入式技术的培训材料。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

汽车电子 S32K 系列微控制器:基于 ARM Cortex-M4F 内核 / 王宜怀,邵长星,黄熙著. — 北京:电子工业出版社,2018.8

(嵌入式技术与应用丛书)

ISBN 978-7-121-34841-9

I. ①汽… II. ①王… ②邵… ③黄… III. ①汽车—电子设备—微控制器 IV. ①U463.6

中国版本图书馆 CIP 数据核字(2018)第 177371 号

责任编辑:田宏峰

印 刷:三河市双峰印刷装订有限公司

装 订:三河市双峰印刷装订有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1 092 1/16 印张:25 字数:646 千字

版 次:2018 年 8 月第 1 版

印 次:2018 年 8 月第 1 次印刷

定 价:88.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: tianhf@phei.com.cn。

前言

恩智浦(NXP)公司于2017年推出了面向汽车电子的S32K系列微控制器。该微控制器以ARM Cortex-M4F为内核,内核频率为112 MHz,运行温度为 $-40^{\circ}\text{C}\sim 125^{\circ}\text{C}$,具有良好的EMC特性,可应用于汽车电子中的车身中控、信息娱乐系统连接、停车辅助系统、空调控制、车窗/天窗/车门控制、防盗装置、通用传感器节点等。整个S32K系列的引脚兼容,还将与未来Kinetis Auto系列其他产品的引脚兼容,可最大限度地提高硬件与软件的重用性,减少投放市场的时间。相比于其他微控制器,S32K系列微控制器具有更宽的运行温度范围及更高的可靠性,具有性能好、可靠性高、价格低等突出特点,必将成为汽车电子领域的重要成员,也能更好地应用于其他对可靠性要求较高的领域。

本书以构件化底层驱动为基础阐述S32K系列微控制器应用程序的设计方法,具有如下特点。

(1) 按照由浅入深的原则,在简明给出基本软件、硬件要素的基础上,尽快给出符合嵌入式软件工程规范的工程框架、第一个C语言样例、第一个汇编样例及启动过程解析,以便快速规范入门。一些较深的内容,随后给出。

(2) 对底层驱动进行构件化封装。书中的每个模块均根据嵌入式软件工程基本原则并按照构件化封装的要求,编写了面向芯片级的底层驱动程序,同时给出了较为详细、规范的注释及对外接口,为实际应用提供了底层驱动构件,方便移植与复用,可以为读者在实际项目开发时节省大量的时间。面向芯片级底层驱动程序为芯片的实际应用提供了共性技术基础,避免了面向板级驱动给不同用户带来的不便。

(3) 设计合理的测试用例。书中的所有源程序均通过测试,并保留了测试用例,避免了因例程的书写或固有错误给读者带来烦恼。这些测试用例也为读者的验证与理解带来方便。每个测试用例均为独立的工程,包含文档说明,可为读者提供便利。书中附录还提供了S32DS集成开发环境的简明使用方法。

(4) 本书配套的网上光盘提供了所有模块的完整底层驱动构件化封装程序与测试用例。针对需要使用PC程序测试用例的读者,本书还提供了PC的C#源程序。网上光盘的版本将会适时更新。

(5) 本书配套的网上光盘提供了最小硬件系统电路图,可供读者应用参考;同时还提供了含有S32K微控制器的金葫芦IoT-GEC开发套件等硬件评估系统,方便读者进行实践与应用。

本书由苏州大学的王宜怀负责编制提纲和统稿工作,并撰写了第2~6章;中国科技大学的邵长星撰写了第7~11章及附录;NXP公司的黄熙撰写了第1章、第12~14章;苏州大学的博士研究生蒋建武、朱仕浪,硕士研究生孙亚军、程宏玉、黄志贤、刘贤德等协助书稿整理及程序调试工作,他们卓有成效的工作,使本书更加实用;ARM公司、NXP公司

为本书的撰写提供了技术条件；电子工业出版社的编辑为本书的出版做了大量细致的工作。在此一并表示诚挚的谢意。

鉴于作者水平有限，书中难免存在不足和错误之处，恳望读者提出宝贵的意见和建议，以便再版时改进。

苏州大学 王宜怀
2018 年 5 月

网上光盘资源文件夹的结构

- SD-S32K-CD(V1.0)
 - 01-Documant
 - 02-Hardware
 - 03-Software
 - S32K-program
 - ch04-LIGHT
 - ch06-UART
 - ch07-Timer
 - ch08-FTM
 - ch09-Flash
 - ch10-ADC-CMP
 - ch11-SPI-I2C
 - ch12-DMA
 - ch13-FlexCAN
 - S32K共用驱动
 - 01-底层驱动构件
 - 02-工程统一部分
 - 04-Tool
 - 05-Other

该网上光盘可在苏州大学嵌入式学习社区网站下载。下载路径为：“<http://sumcu.suda.edu.cn>” → “教学与培训” → “教学资料” → “S32K 书” → “SD-S32K-CD”。该网上光盘会根据改进情况进行不定期的更新，敬请读者关注。

目录

第 1 章	概述	(1)
1.1	汽车电子技术的基本概念	(1)
1.2	我国汽车电子发展概况	(2)
1.3	恩智浦公司在汽车电子市场中的地位	(3)
1.4	面向汽车电子的 S32K 系列微控制器简介	(3)
1.4.1	S32K 系列微控制器的型号标识	(4)
1.4.2	S32K 系列微控制器的简明特性与结构框图	(4)
1.4.3	S32K 系列微控制器的共性资源列表	(6)
第 2 章	ARM Cortex-M4F 微控制器	(7)
2.1	ARM Cortex-M4F 微控制器简介	(7)
2.1.1	ARM Cortex-M4F 微控制器内部结构概要	(8)
2.1.2	ARM Cortex-M4F 微控制器存储器映像	(10)
2.1.3	ARM Cortex-M4F 微控制器的寄存器	(10)
2.2	指令系统	(14)
2.2.1	指令简表与寻址方式	(15)
2.2.2	数据传送类指令	(16)
2.2.3	数据操作类指令	(18)
2.2.4	跳转控制类指令	(21)
2.2.5	其他指令	(22)
2.3	指令集与机器码对应表	(23)
2.4	GNU 汇编语言的基本语法	(25)
2.4.1	ARM-GUN 汇编语言格式	(26)
2.4.2	常用伪指令简介	(27)
第 3 章	存储映像、中断源与硬件最小系统	(30)
3.1	S32K144 存储映像与中断源	(30)
3.1.1	S32K144 存储映像	(30)
3.1.2	S32K144 中断源	(32)
3.2	S32K144 的引脚功能	(35)
3.3.1	硬件最小系统引脚	(36)
3.2.2	I/O 端口资源类引脚	(36)
3.3	S32K144 硬件最小系统原理图	(37)

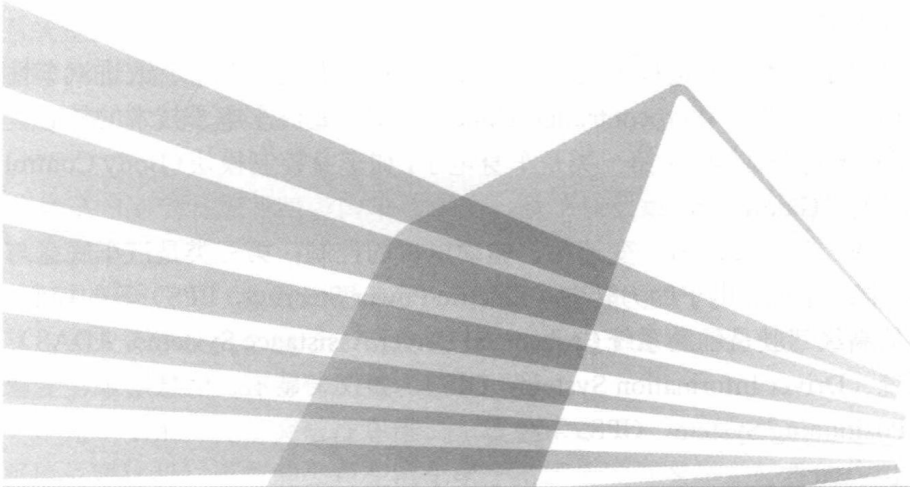
3.3.1	电源及其滤波电路	(37)
3.3.2	复位电路及复位功能	(37)
3.3.3	晶振电路	(38)
3.3.4	SWD 接口电路	(38)
第 4 章	GPIO 及程序框架	(39)
4.1	通用 I/O 接口的基本概念及连接方法	(39)
4.1.1	I/O 接口的概念	(39)
4.1.2	通用 I/O	(40)
4.1.3	上拉/下拉电阻与输入引脚的基本接法	(40)
4.1.4	输出引脚的基本接法	(41)
4.2	端口控制模块与 GPIO 模块的编程结构	(41)
4.2.1	端口控制模块——决定引脚复用功能	(41)
4.2.2	GPIO 模块的对外引脚与内部寄存器	(44)
4.2.3	GPIO 基本编程步骤与基本打通程序	(45)
4.3	GPIO 驱动构件封装方法与规范	(46)
4.3.1	设计 GPIO 驱动构件的必要性及 GPIO 驱动构件封装要点分析	(46)
4.3.2	底层驱动构件封装规范概要与构件封装的前期准备	(48)
4.3.3	S32K144 的 GPIO 驱动构件源码及解析	(49)
4.4	利用构件方法控制 LED 闪烁	(60)
4.4.1	light 构件设计	(60)
4.4.2	light 构件测试工程主程序	(63)
4.5	工程文件组织框架与第一个 C 语言工程分析	(65)
4.5.1	工程框架及所含文件简介	(66)
4.5.2	链接文件常用语法及链接文件解析	(67)
4.5.3	机器码文件解析	(71)
4.5.4	芯片上电启动运行过程解析	(72)
4.6	第一个汇编语言工程：控制小灯闪烁	(75)
4.6.1	汇编工程文件的组织	(76)
4.6.2	汇编语言 GPIO 驱动构件及使用方法	(76)
4.6.3	汇编语言 Light 构件及使用方法	(82)
4.6.4	汇编语言 Light 构件测试工程主程序及汇编工程运行过程	(84)
第 5 章	嵌入式硬件构件与底层驱动构件基本规范	(86)
5.1	嵌入式硬件构件	(86)
5.1.1	嵌入式硬件构件的概念及其分类	(87)
5.1.2	基于嵌入式硬件构件的电路原理图设计简明规则	(87)
5.2	嵌入式底层驱动构件的概念与层次模型	(90)
5.2.1	嵌入式底层驱动构件的概念	(90)
5.2.2	嵌入式硬件构件和软件构件的层次模型	(91)
5.3	底层驱动构件的封装规范	(91)

5.3.1	构件设计的基本思想与基本原则	(92)
5.3.2	编码风格基本规范	(93)
5.3.3	构件公共要素文件	(96)
5.3.4	头文件的设计规范	(98)
5.3.5	源文件的设计规范	(99)
5.4	硬件构件及底层软件构件的重用与移植方法	(100)
5.4.1	硬件构件在实际系统中的应用	(100)
5.4.2	底层驱动构件的移植	(101)
第 6 章	串行通信模块及第一个中断程序结构	(103)
6.1	异步串行通信的通用基础知识	(103)
6.1.1	串行通信的基本概念	(104)
6.1.2	RS-232 总线标准	(105)
6.1.3	TTL 电平到 RS-232 电平转换电路	(106)
6.1.4	串行通信编程模型	(107)
6.2	UART 驱动构件及使用方法	(108)
6.2.1	UART 引脚分析	(108)
6.2.2	UART 驱动构件基本要素分析与头文件	(109)
6.2.3	printf 的设置方法与使用	(113)
6.3	ARM Cortex-M4F 中断机制及 S32K144 中断编程	(113)
6.3.1	关于中断的通用基础知识	(113)
6.3.2	ARM Cortex-M4F 非内核中断编程结构	(115)
6.3.3	S32K144 中断编程步骤 (以串口接收中断为例)	(116)
6.4	UART 驱动构件的设计方法	(118)
6.4.1	UART 模块编程结构	(118)
6.4.2	UART 驱动构件源文件	(123)
第 7 章	Timer 模块	(132)
7.1	ARM Cortex-M4F 内核定时器 (Systick)	(132)
7.1.1	Systick 定时器模块的编程结构	(133)
7.1.2	Systick 驱动构件设计及测试工程	(134)
7.2	延时定时器 (PDB) 模块	(136)
7.2.1	PDB 模块功能概述	(136)
7.2.2	PDB 驱动构件及使用方法	(136)
7.2.3	PDB 驱动构件的设计	(139)
7.3	低功耗中断定时器 (LPIT) 模块	(143)
7.3.1	LPIT 模块功能概述	(143)
7.3.2	LPIT 驱动构件及使用方法	(143)
7.3.3	LPIT 驱动构件设计	(145)
7.4	低功耗定时器 (LPTMR) 模块	(150)
7.4.1	LPTMR 模块功能概述	(150)

7.4.2	LPTMR 驱动构件及使用方法	(150)
7.4.3	LPTMR 驱动构件的设计	(152)
7.5	实时时钟 (RTC) 模块	(155)
7.5.1	RTC 模块功能概述	(155)
7.5.2	RTC 驱动构件及使用方法	(156)
7.5.3	RTC 驱动构件的设计	(160)
第 8 章	PWM 模块和 FTM 模块	(169)
8.1	脉宽调制、输入捕捉与输出比较的通用基础知识	(169)
8.1.1	PWM 的通用基础知识	(169)
8.1.2	输入捕捉与输出比较的通用基础知识	(172)
8.2	FTM 模块的基本知识	(172)
8.2.1	FTM 模块概述	(172)
8.2.2	FTM 模块的技术要点	(173)
8.3	FTM 驱动构件及使用方法	(174)
8.3.1	FTM 模块的脉宽调制、输入捕捉和输出比较的外部引脚	(174)
8.3.2	FTM 驱动构件头文件及使用方法	(175)
8.4	FTM 驱动构件的设计	(183)
8.4.1	FTM 模块编程结构	(183)
8.4.2	FTM 驱动构件设计	(187)
第 9 章	Flash 模块	(196)
9.1	Flash 存储器的基础知识	(196)
9.1.1	Flash 存储器的特性	(196)
9.1.2	Flash 存储器的编程模式	(197)
9.1.3	Flash 存储器的基本操作	(197)
9.2	Flash 驱动构件及其使用方法	(197)
9.2.1	Flash 驱动构件头文件解析	(197)
9.2.2	Flash 驱动构件的使用方法	(200)
9.3	Flash 保护	(201)
9.3.1	Flash 模块保护的含意及保护函数的使用说明	(201)
9.3.2	Flash 模块加密方法与去除密码方法	(202)
9.4	Flash 驱动构件的设计	(203)
9.4.1	Flash 模块编程结构	(203)
9.4.2	Flash 驱动构件设计技术要点	(207)
9.4.3	Flash 驱动构件封装要点分析	(209)
9.4.4	Flash 驱动构件的源文件 (flash.c)	(210)
第 10 章	ADC 模块与 CMP 模块	(218)
10.1	模/数转换器 (ADC) 模块	(218)
10.1.1	ADC 模块的通用基础知识	(218)
10.1.2	ADC 模块驱动构件及使用方法	(221)

10.1.3	ADC 驱动构件的设计	(225)
10.2	比较器 (CMP) 模块	(233)
10.2.1	CMP 模块的通用基础知识	(233)
10.2.2	CMP 驱动构件及使用方法	(234)
10.2.3	CMP 驱动构件的设计	(238)
第 11 章	SPI 模块与 I2C 模块	(244)
11.1	串行外设接口 (SPI) 模块	(244)
11.1.1	SPI 模块的通用基础知识	(244)
11.1.2	SPI 驱动构件及使用方法	(247)
11.1.3	SPI 驱动构件的设计	(252)
11.2	集成电路互连 (I2C) 总线模块	(262)
11.2.1	I2C 模块的通用基础知识	(262)
11.2.2	I2C 驱动构件及使用方法	(267)
11.2.3	I2C 驱动构件的设计	(273)
第 12 章	DMA 模块	(291)
12.1	DMA 模块通用基础知识	(291)
12.1.1	DMA 模块的基本概念	(291)
12.2.1	DMA 模块的基本操作	(292)
12.2	DMA 驱动构件及其使用方法	(293)
12.2.1	DMA 源	(293)
12.2.2	DMA 驱动构件封装要点	(294)
12.2.3	DMA 驱动构件头文件 (dma.h)	(295)
12.2.4	DMA 驱动构件的使用方法	(299)
12.2.5	DMA 驱动构件测试实例	(299)
12.3	DMA 驱动构件的设计	(300)
12.3.1	DMA 模块编程结构	(300)
12.3.2	DMA 驱动构件源程序	(309)
第 13 章	FlexCAN 模块	(314)
13.1	CAN 总线的基础知识	(314)
13.1.1	CAN 总线协议的历史概况	(314)
13.1.2	CAN 总线硬件系统的典型电路	(315)
13.1.3	CAN 总线协议的基本概念	(317)
13.1.4	CAN 总线规范的帧结构	(319)
13.1.5	CAN 总线的位时间	(323)
13.1.6	FlexCAN 模块的操作模式	(324)
13.1.7	FlexCAN 模块的仲裁处理、匹配处理及报文缓冲区管理	(325)
13.2	FlexCAN 驱动构件及使用方法	(326)
13.2.1	FlexCAN 驱动构件头文件的解析	(326)
13.2.2	FlexCAN 驱动构件的头文件	(327)

13.2.3	FlexCAN 驱动构件的使用方法	(330)
13.2.4	FlexCAN 驱动构件的测试实例	(331)
13.3	FlexCAN 驱动构件的设计	(331)
13.3.1	FlexCAN 模块编程结构	(331)
13.3.2	FlexCAN 驱动构件的源程序	(353)
第 14 章	系统时钟与其他功能模块	(361)
14.1	时钟系统模块	(361)
14.1.1	时钟系统概述	(361)
14.1.2	时钟模块概要与编程要点	(364)
14.2	电源模块	(365)
14.2.1	电源模式控制	(365)
14.2.2	电源模式转换	(365)
14.3	校验模块	(366)
14.3.1	CRC 模块简介	(367)
14.3.2	CRC 校验和的生成步骤	(367)
14.3.3	CRC 模块的主要寄存器	(367)
14.4	看门狗模块	(368)
14.5	复位模块	(369)
14.5.1	上电复位 (POR)	(369)
14.5.2	系统复位源	(370)
14.5.3	调试复位	(371)
附录 A	S32K144 芯片引脚复用功能	(372)
附录 B	最小硬件系统原理图	(376)
附录 C	printf 的常用格式	(377)
附录 D	S32DS 集成开发环境简明使用方法	(380)
附录 E	SWD-Programmer 快速指南	(384)
附录 F	术语和缩写	(386)
参考文献		(388)



第1章

概述

本章导读：鉴于本书主要阐述面向汽车电子的微控制器，作为全书导引，本章首先简要阐述汽车电子技术的基本概念、我国汽车电子发展概况，以及恩智浦公司在汽车电子市场中的地位，随后给出恩智浦公司面向汽车电子的微控制器 S32K 系列的型号标识与基本特点。S32K 系列微控制器与一般工业级微控制器的主要区别是提高了运行温度等级和电磁兼容等级，当然也可用于一般工业控制领域。

本章参考资料：1.4 节中关于 S32K 系列微控制器的型号标识参考《S32K 数据手册》的第 3 章；S32K 系列 MCU 的基本特点及简明资源参考《S32K 参考手册》的第 2 章；有关简称含义见本书附录 F。

1.1 汽车电子技术的基本概念

汽车电子是车体汽车电子控制装置和车载汽车电子控制装置的总称，其最重要的作用是提高汽车的安全性、舒适性、经济性和娱乐性。近年来，随着信息技术的快速发展，以及汽车制造业水平的不断提升，汽车电子技术得到了广泛应用和创新，越来越多的电子控制单元（Electronic Control Unit, ECU）被集成在汽车中，电子设备在汽车总成本中所占的比例也随之攀升。据统计，1970 年全球汽车电子设备的成本占比为 2%，1980 年为 5%，1990 年为 15%，2000 年为 20%，2012 年为 25%，2014 年为 27%，2016 年 30%，2017 年 33%，预计到 2030 年将达到 40%。这些汽车电子设备的加入，一方面对提高汽车的动力性、燃油经济性、安全性，以及降低汽车排放污染起到了非常关键的作用；另一方面提供了如仪表显示、信息娱乐、通信互联等丰富的人车交互系统，以及智能化的车身控制模块，大大增强了整车的驾驶舒适性。专家指出，近 10 年来汽车产业 70% 的创新来源于汽车电子技术及其产品的开发应用，

汽车电子技术的应用水平已成为衡量汽车档次水平的主要标志，其应用程度的提高是汽车生产企业提高市场竞争力的重要手段。

汽车电子化被认为是汽车技术发展进程中的一次革命，是用来开发新车型、改进汽车性能最重要的技术措施，而微控制器（Microcontroller Unit, MCU）是汽车技术的核心。按照应用方向，汽车电子技术可分成四类：第一类是车身电子，如车身控制模块（Body Control Module, BCM）、车内网关（Gateway），以及对车灯、车窗、车门等部件进行控制的车身节点；第二类是汽车动力总成，如对发动机、变速箱、驱动电机的控制；第三类是汽车底盘与安全，如对底盘、悬架、制动系统、电子助力转向（Electric Power Steering, EPS）等的控制，以及近期发展势头强劲的高级驾驶员辅助系统（Advanced Driver Assistance Systems, ADAS）；第四类是驾驶员信息系统（Driver Information System, DIS），如仪表显示、信息娱乐、全球卫星定位系统（Global Positioning System, GPS）等。

1.2 我国汽车电子发展概况

我国自 2001 年正式加入 WTO 以来，巨大的市场潜力吸引了全球众多知名汽车厂商的目光，汽车产业因此得到了飞速发展。短短几年，中国一举成为世界最大的汽车生产国和消费国之一。据中国汽车工业协会公布的统计数据显示，2015 年，中国汽车产销双双超过 2450 万辆，再创全球产销最高纪录，其中产量 2450.33 万辆，销量 2459.76 万辆，同比分别增长 3.3% 和 4.7%。预计在未来较长的一段时间，我国都将保持汽车产销大国地位。

汽车产业的迅猛发展为汽车电子及相关半导体产业提供了广阔的市场前景。根据 Strategy Analytics 的最新数据显示，2013 年中国汽车半导体（包括 MCU、Sensor 及 MOSFET、IGBT、三极管、二极管等分立器件）的总需求量为 120.87 亿颗，较 2012 年增长了 26.2%，其中 MCU 需求量为 4.34 亿颗，平均每辆汽车大约包含 19 颗 MCU；2013 年中国汽车半导体的市场规模为 39.99 亿美元，较 2012 年增长了 22.2%，其中 MCU 的市场规模为 10.46 亿美元。我国汽车电子市场及其增长走势如图 1-1 所示。



图 1-1 我国汽车电子市场规模及其增长率走势

未来几年，我国汽车电子及相关的半导体产业将保持稳步增长。其中，随着消费者对汽车安全性关注的日益提高，电子助力转向（EPS）系统、轮胎压力监测系统（Tire Pressure Monitoring System, TPMS），以及基于雷达和视觉的高级驾驶员辅助系统（ADAS）等一系

列安全技术将逐步得到广泛应用,并成为今后一段时间的热点话题。而仪表显示与信息娱乐系统将在图形化仪表与大屏幕显示需求的推动下,继续成为汽车电子市场增长的引擎。随着国家各项节能减排政策的发布和落实,以油电混合动力汽车(Hybrid Electric Vehicle, HEV)应用为主的新能源技术也将成为业界聚焦的热点。

1.3 恩智浦公司在汽车电子市场中的地位

恩智浦公司(收购飞思卡尔公司后)是汽车电子及嵌入式处理解决方案的主要供应商,其主要业务涉及汽车电子、消费电子、工业电子及网络设备等市场。其中,汽车电子类客户遍及全球各大顶尖零部件厂商和整车厂商。

几十年来,凭借突破性思维、工程技术专长、对高品质的追求,以及长久保持的市场领军地位等优势,恩智浦公司在汽车电子市场上发挥着举足轻重的作用。

(1) 北美汽车半导体供应商领域:第一(来源:Strategy Analytics, 2014 年 4 月)。

(2) 全球商用汽车微机电系统(Micro-Electro-Mechanical System, MEMS)传感器领域:第一(来源:IHS, 2014 年 5 月)。

(3) 全球商用汽车加速度传感器领域:第一(来源:IHS, 2014 年 5 月)。

(4) 全球汽车微控制器领域:第二(来源:IHS, 2014 年 3 月)。

(5) 全球车载信息娱乐微处理器领域:第二(来源:Strategy Analytics, 2014 年 4 月)。

(6) 恩智浦公司创造了全球第一款发动机控制芯片,并保持领先至今。

(7) 恩智浦公司是全球首家集成 77 GHz 雷达技术的汽车半导体供应商。

(8) 全球 TOP10 的汽车厂商有 7 家选择 i.MX6 应用微处理器来支持其信息娱乐系统的连接和解决方案。

(9) 全球 6 大顶尖豪华车品牌将在其图形化仪表盘中采用 i.MX 应用微处理器。

(10) 2016 年,在总值 274 亿美元的汽车半导体市场中,恩智浦公司(NXP)已占有 14.2% 的市场份额。

恩智浦公司拥有丰富的微控制器、微处理器、模拟器件和传感器产品线,覆盖了从低端到高端的汽车电子产品,能够帮助客户实现全新的突破性的汽车设计,包括动力总成、车身、底盘与安全性、车载信息娱乐和通信系统,以及车内网络应用。基于世界一流的架构以及业界领先的安防和功能安全技术,恩智浦公司可以帮助客户实现开发的可靠性和自由性。凭借 Power Architecture 技术及 ARM 技术,客户可以获得多种设计选择。

恩智浦公司为汽车电子市场提供了产品长期供货保障,广泛应用的器件都包含在该计划中,可保证至少 15 年的供货期。恩智浦公司全面贯彻高品质企业文化,致力保证产品质量、交货时间和最高服务水平,并成为客户眼中最卓越的半导体公司。

1.4 面向汽车电子的 S32K 系列微控制器简介

S32K 系列 MCU 是恩智浦公司针对汽车市场最新开发的 MCU,在汽车电子领域有着广泛的应用。例如,在信息娱乐系统连接模块、停车辅助系统、普通直流电机/无刷直流电机(Brushless Direct Current Motor, BLDC)控制、电子驻车制动、电池管理、泵/风扇控制器、智能无钥匙进入及启动系统、汽车照明、车身中控/空调、车窗/天窗/车门、座

椅/后视镜/雨刮器、防盗装置、动力总成配套芯片、通用传感器节点等方面，都可以作为目标应用。

1.4.1 S32K 系列微控制器的型号标识

学习一个新的微控制器（MCU）可以从认识型号标识开始，型号标识也是芯片选型与购买的关键知识。

恩智浦公司 S32K 系列 MCU 的型号众多，但同一子系列的中央处理器(Central Processing Unit, CPU) 是相同的，多种型号只是为了适用于不同的应用场合。为了方便选型或订购，恩智浦公司给出了 MCU 型号标识^①。S32K 系列型号标识格式为：“F S32 K I J N X Y F0 M LC R”，各字段说明如表 1-1 所示。

表 1-1 S32K 系列芯片命令字段说明

字 段	说 明	取 值
F/P	产品状态	P 表示原型；F 表示合格订货件号
S32	产品类型/品牌	用于汽车的 32 位 MCU
K/M	生产线	K 表示 ARM Cortex 系列 MCU；M 表示 MagniV/混合信号
I	产品系列	1 表示第一个产品系列；2 表示第二个产品系列
J	核心平台/性能	1 表示 ARM Cortex-M4；4 表示 ARM Cortex-M4F
N	内存大小	M4：2 表示 32 KB，4 表示 64 KB，6 表示 128 KB，8 表示 256 KB。M4F：2 表示 256 KB，4 表示 512 KB，6 表示 1 MB，8 表示 2 MB
X	频率	B 表示不带 DMA 的 48 MHz（仅仅支持 S32K11x）；L 表示带 DMA 的 48 MHz（仅仅支持 S32K11x）；M 表示 64 MHz；H 表示 80 MHz；U 表示 112 MHz
Y	可选功能	N 表示无；R 表示最大内存；F 表示 CAN-FD 和 FlexIO 包括最大内存；S 表示安全包括最大内存；A 表示 CAN-FD、FlexIO 和安全包括最大内存；E 表示以太网和音频，包括最大内存；J 表示 CAN-FD、FlexIO、安全、以太网和音频，包括最大内存
F0	生产信息	Fx 表示 ATMC；Tx 表示 GF；XX 表示 Flex#；x0 表示第 1 次修订，x1 表示第 2 次修订
M	温度范围	C 表示 -40℃~85℃；V 表示 -40℃~105℃；M 表示 -40℃~125℃
LC	封装类型	LC 表示 32LQFP；FM 表示 32QFN；LF 表示 48LQFP；KF 表示 48LQFP-EP；FT 表示 48QFN；LH 表示 64LQFP；KH 表示 64LQFP-EP；LL 表示 100LQFP；MH 表示 100BGA；LQ 表示 144LQFP；LU 表示 176LQFP
R	包装形式	T 表示托盘和管子包装；R 表示卷带包装

本书使用的芯片型号为 FS32K144UAT0VLLA，从该芯片型号标识可以获得如下信息：该芯片为 S32K 系列、ARM Cortex-M4F 内核、CPU 最高频率为 112 MHz、程序 Flash 大小为 512 KB、包含控制器局域网（Controller Area Network，CAN）模块、运行温度范围是-40℃~105℃、100 引脚 LQFP 封装。

1.4.2 S32K 系列微控制器的简明特性与结构框图

学习一个新的 MCU，了解该 MCU 的简明特性与结构框图是十分必要的。
S32K 系列 MCU 基于的 ARM Cortex-M4F 微控制器是目前市场上能效最高的 32 位微控制器，每微安的数据吞吐量在业内居领先水平。S32K 系列 MCU 具有多种灵活的超低功耗模

① 来自《S32K 数据手册》第 3 章，该文档中给出的型号标识字段中含有程序存储器类型，实际出厂的芯片并未使用该字段，默认程序存储器为 Flash。

式, 适合不同的应用情形, 可最大限度地延长电池的使用时间; 在不唤醒内核的情况下, 智能外设在深度睡眠模式下仍然可以工作, 可进行智能决策并处理数据。S32K 系列 MCU 包含了一组功能强大的模拟、通信、定时和控制外设, 提供各种闪存规格和引脚数。

(1) 供电电压、工作频率、温度范围: 该系列 MCU 的工作电压范围为 2.7~5.5V, 内核运行频率最大为 112 MHz, 运行温度范围为-40℃~125℃。

(2) 系统工作时钟源及低功耗特性: 内部含 128 kHz 低功耗振荡器, 支持 32.768 kHz 晶体振荡器、4~40 MHz 外接晶体振荡器或陶瓷谐振器; 在低功耗特性方面, 电源管理模块具有多个电源管理模式, 以支持低功耗应用。

(3) 人机接口、模拟模块、定时器: 最多可达 156 个通用输入/输出接口, 大多支持外部中断; 2 个 32 通道的 12 位模/数转换器, 可在停止模式下运行, 提供硬件触发选项; 提供 4 个独立的 32 位柔性定时器模块, 提供多达 32 个标准通道; 1 个具有灵活唤醒控制功能的 16 位低功耗定时器; 2 个可编程延迟模块, 带有灵活的触发系统; 1 个带 4 个通道的 32 位低功耗中断定时器; 1 个 32 位实时时钟。

(4) 通信接口: 3 个串行通信接口、2 个串行外设接口、2 个集成电路互连 (I²C) 总线接口, 3 个 CAN 总线接口, 均支持直接存储器访问 (Direct Memory Access, DMA) 及低功耗模式。

(5) 可扩展解决方案: 包括高达 2 MB 的闪存 (Flash), 并且可轻松扩展, 以满足不同的汽车应用需求。

S32K 系列 MCU 的结构框图如图 1-2 所示。

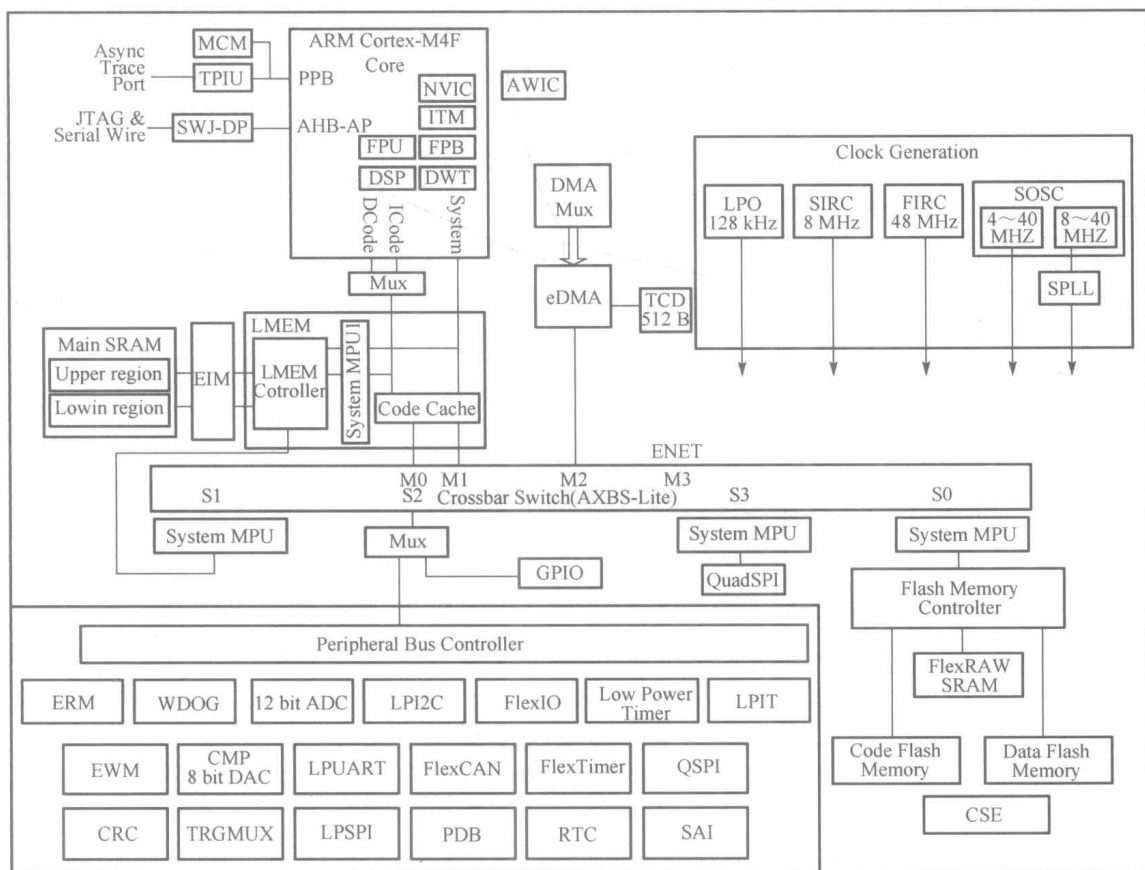


图 1-2 S32K 系列 MCU 的结构框图