

嵌入式软件开发

版本：v1.5.2

Crifan Li

摘要

此文主要介绍的嵌入式软件开发的背景知识，基本逻辑和流程，基本名词、概念和理念，嵌入式开发环境的基本组成，交叉编译和交叉编译器，烧录，工具，系统设计，嵌入式CPU，嵌入式资源，常见问题及解答。



本文提供多种格式供：

在线阅读	HTML ¹	HTMLs ²	PDF ³	CHM ⁴	TXT ⁵	RTF ⁶	WEBHELP ⁷
下载（7zip压缩包）	HTML ⁸	HTMLs ⁹	PDF ¹⁰	CHM ¹¹	TXT ¹²	RTF ¹³	WEBHELP ¹⁴

HTML版本的在线地址为：

http://www.crifan.com/files/doc/docbook/embedded_soft_dev/release/html/embedded_soft_dev.html

有任何意见，建议，提交bug等，都欢迎去讨论组发帖讨论：

http://www.crifan.com/bbs/categories/embedded_soft_dev/

修订历史		
修订 1.5.2	2015-05-25	crl
<div>1. 增加嵌入式软件开发的背景知识</div> <div>2. 增加嵌入式基本的组成和流程的介绍</div> <div>3. 增加嵌入式行业从业人员概览</div> <div>4. 把portfolio移到软件通用知识中</div> <div>5. 添加解释Android和Linux是不同的</div> <div>6. 更新嵌入式开发工具，添加嵌入式CPU的应用领域</div>		

¹ http://www.crifan.com/files/doc/docbook/embedded_soft_dev/release/html/embedded_soft_dev.html
² http://www.crifan.com/files/doc/docbook/embedded_soft_dev/release/htmls/index.html
³ http://www.crifan.com/files/doc/docbook/embedded_soft_dev/release/pdf/embedded_soft_dev.pdf
⁴ http://www.crifan.com/files/doc/docbook/embedded_soft_dev/release/chm/embedded_soft_dev.chm
⁵ http://www.crifan.com/files/doc/docbook/embedded_soft_dev/release/txt/embedded_soft_dev.txt
⁶ http://www.crifan.com/files/doc/docbook/embedded_soft_dev/release/rtf/embedded_soft_dev.rtf
⁷ http://www.crifan.com/files/doc/docbook/embedded_soft_dev/release/webhelp/index.html
⁸ http://www.crifan.com/files/doc/docbook/embedded_soft_dev/release/html/embedded_soft_dev.html.7z
⁹ http://www.crifan.com/files/doc/docbook/embedded_soft_dev/release/htmls/index.html.7z
¹⁰ http://www.crifan.com/files/doc/docbook/embedded_soft_dev/release/pdf/embedded_soft_dev.pdf.7z
¹¹ http://www.crifan.com/files/doc/docbook/embedded_soft_dev/release/chm/embedded_soft_dev.chm.7z
¹² http://www.crifan.com/files/doc/docbook/embedded_soft_dev/release/txt/embedded_soft_dev.txt.7z
¹³ http://www.crifan.com/files/doc/docbook/embedded_soft_dev/release/rtf/embedded_soft_dev.rtf.7z
¹⁴ http://www.crifan.com/files/doc/docbook/embedded_soft_dev/release/webhelp/embedded_soft_dev.webhelp.7z

嵌入式软件开发:

Crifan Li

版本 : v1.5.2

出版日期 2015-05-25

版权 © 2015 Crifan, <http://crifan.com>

本文章遵从 : [署名-非商业性使用 2.5 中国大陆\(CC BY-NC 2.5\)](#)¹⁵

¹⁵ http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#cc_by_nc

目录

正文之前	viii
1. 此文目的	viii
2. 联系方式	viii
1. 嵌入式软件开发相关背景知识	1
1.1. 嵌入式行业和领域	1
1.2. 嵌入式产业链	4
1.3. 嵌入式行业概览	4
1.3.1. 2014年嵌入式行业从业人员：工作经验	5
1.3.2. 2014年嵌入式行业从业人员：学历要求	6
1.3.3. 2014年嵌入式行业从业人员：地域分布	6
1.3.4. 2014年嵌入式行业从业人员：嵌入式开发前的研发方向	7
1.3.5. 2014年嵌入式行业从业人员：薪资水平	7
1.3.6. 2014年嵌入式行业从业人员：软件开发语言的选择	8
1.3.7. 2014年嵌入式行业从业人员：所属行业分布情况	8
1.3.8. 2014年嵌入式行业从业人员：公司规模	9
1.3.9. 2014年嵌入式行业从业人员：软件开发平台的选择	9
1.3.10. 2014年嵌入式行业从业人员：软件开发调试工具的选择	10
1.3.11. 2014年嵌入式行业从业人员：硬件开发平台的选择	10
1.3.12. 2014年嵌入式行业从业人员：处理器芯片的选择	11
1.3.13. 2014年嵌入式行业从业人员：软硬件人员的安排	11
1.3.14. 2014年嵌入式行业从业人员：未来嵌入式操作系统首选	12
1.3.15. 2014年嵌入式行业从业人员：最想选择的嵌入式应用方向	12
1.3.16. 2014年嵌入式行业从业人员：获取专业知识的途径	13
1.3.17. 2014年嵌入式行业从业人员：首选搜索引擎	13
1.3.18. 2014年嵌入式行业从业人员：企业人才需求现状	14
1.3.19. 2014年嵌入式行业从业人员：对工作薪资的满意度	14
1.3.20. 2014年嵌入式行业从业人员：如何看待培训	15
1.4. 嵌入式软硬件厂商的合作关系	15
2. 嵌入式软件开发的基本逻辑	17
2.1. 嵌入式软件开发的基本逻辑	17
2.1.1. 按照是否有操作系统	17
2.1.2. 按照操作系统是否是私有的	18
2.2. 嵌入式软件开发内容分类	19
2.3. 嵌入式软件期间会涉及哪些内容	19
2.4. 嵌入式软件开发需要学习哪些知识	23
2.4.1. 嵌入式领域内相对通用的知识和技能	24
2.4.1.1. 嵌入式相对通用的知识：计算机编程语言	24
2.4.1.1.1. 嵌入式相对通用语言：C语言	24
2.4.1.1.2. 嵌入式相对通用语言：汇编语言	24
2.4.1.1.3. 嵌入式相对通用语言：C++语言	24
2.4.1.2. 嵌入式相对通用的知识：芯片架构	24
2.4.1.3. 嵌入式相对通用的知识：嵌入式操作系统	26
2.4.1.3.1. 嵌入式常用操作系统：嵌入式Linux	26
2.4.1.3.2. 嵌入式常用操作系统：μC/OS-II	26
2.4.1.3.3. 嵌入式常用操作系统：ThreadX	26
2.4.1.3.4. 嵌入式常用操作系统：VxWorks	26
2.4.1.3.5. 嵌入式常用操作系统：WinCE	26
2.4.1.3.6. Android不是传统意义上的嵌入式操作系统	26
2.4.1.3.6.1. Android和嵌入式Linux的区别	26
2.4.1.4. 嵌入式相对通用的知识：嵌入式开发环境搭建和使用	27
2.4.1.4.1. 嵌入式开发环境：交叉编译器	28
2.4.1.4.2. 嵌入式开发环境：固件的烧写	28
2.4.1.4.3. 嵌入式开发环境：调试	28
2.4.1.4.3.1. 嵌入式开发调试方法和调试工具	28
2.4.1.4.4. 嵌入式开发环境：嵌入式IDE	29

2.4.2. 嵌入式领域内某领域内相关特定知识	29
2.4.2.1. 嵌入式领域相关知识之消费类数码	30
2.4.2.2. 嵌入式领域相关知识之工业自动化控制	30
2.4.2.3. 嵌入式领域相关知识之汽车电子	30
2.4.2.4. 嵌入式领域相关知识之医疗器械	31
2.4.2.5. 嵌入式领域相关知识之航空航天	31
2.5. 嵌入式系统中从底层硬件到上层软件之间的逻辑层次和映射关系	31
3. 嵌入式软件开发的基本名词和概念	33
3.1. 开发套件==开发组件==develop suite==develop toolkit=StartKit=DevBoard	33
3.2. Part Number==模块编号==产品编号	35
3.3. fabless==无厂半导体公司==无工厂的IC设计公司	35
3.4. foundry==晶圆厂==晶圆代工厂==代工厂	36
3.5. IP Core==IP核==软核	36
3.6. 量产	36
3.6.1. 批量生产芯片	37
3.6.2. 重新烧录U盘的固件	37
3.7. VFP 浮点处理器	37
3.8. 代码空间占用大小：footprint	37
4. 嵌入式开发环境的基本构成	39
4.1. 嵌入式Linux开发环境组成	39
4.2. WinCE开发环境组成	39
5. 嵌入式开发之交叉编译和交叉编译器	40
5.1. 什么是交叉编译	40
5.2. 如何获得交叉编译器	40
5.3. 如何使用交叉编译器	40
5.4. 嵌入式编译相关知识	40
6. 嵌入式开发之烧录	41
6.1. 什么是烧录	41
6.2. 如何进行烧录	41
6.2.1. 常见的烧录所用工具	41
6.2.2. 常见的烧录方式	41
7. 嵌入式开发之开发工具	43
7.1. 嵌入式开发的IDE	43
7.1.1. 嵌入式IDE：Keil	43
7.1.2. 嵌入式IDE：ARM相关的ADS，RVDS等	43
7.1.3. 嵌入式IDE：IAR	43
7.2. 协议分析工具	43
7.3. 交叉编译器/工具链	44
7.4. 嵌入式开发的调试工具	45
8. 嵌入式开发之系统设计	46
8.1. 嵌入式系统设计之硬件选型	46
8.2. 嵌入式系统设计流程	47
9. 嵌入式开发之嵌入式CPU	48
9.1. 什么是嵌入式CPU	48
9.2. 嵌入式CPU核心参数	48
9.3. 嵌入式CPU的应用领域	48
10. 嵌入式软件开发的外围设备	50
10.1. 嵌入式外设之LCD显示屏	50
10.2. 嵌入式外设之图像传感器	50
11. 嵌入式软件开发相关硬件知识	51
11.1. 芯片封装方式	51
12. 嵌入式软件开发相关的厂商	52
12.1. 嵌入式软件厂商	52
12.1.1. 嵌入式软件厂商：Keil	52
12.2. 嵌入式硬件厂商	52
12.2.1. 嵌入式硬件开发板厂商	52
13. 嵌入式软件开发相关接口和协议	54
13.1. 嵌入式总线之内部总线	54

13.1.1. 嵌入式内部总线之MIPI	54
13.1.2. 嵌入式内部总线之AMBA	54
13.2. 嵌入式总线之外部总线	55
14. 嵌入式软件开发常见问题与解答	56
常见问题	56
14.1. 学习嵌入式C语言要掌握到什么程度？	56
15. 嵌入式软件开发方面的值得借鉴的帖子	58
16. 嵌入式软件开发相关资源	59
16.1. 嵌入式软件开发书籍	59
16.1.1. Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers	59
17. 如何学习嵌入式软件开发	60
参考书目	63

插图清单

1.1. 嵌入式应用之便携式导航	1
1.2. 嵌入式应用之智能家居	2
1.3. 嵌入式应用之安全防护	3
1.4. 嵌入式应用之电力系统	3
1.5. 2014年嵌入式行业从业人员：工作经验	5
1.6. 2014年嵌入式行业从业人员：学历要求	6
1.7. 2014年嵌入式行业从业人员：地域分布	6
1.8. 2014年嵌入式行业从业人员：嵌入式开发前的研发方向	7
1.9. 2014年嵌入式行业从业人员：薪资水平	7
1.10. 2014年嵌入式行业从业人员：软件开发语言的选择	8
1.11. 2014年嵌入式行业从业人员：所属行业分布情况	8
1.12. 2014年嵌入式行业从业人员：公司规模	9
1.13. 2014年嵌入式行业从业人员：软件开发平台的选择	9
1.14. 2014年嵌入式行业从业人员：软件开发调试工具的选择	10
1.15. 2014年嵌入式行业从业人员：硬件开发平台的选择	10
1.16. 2014年嵌入式行业从业人员：处理器芯片的选择	11
1.17. 2014年嵌入式行业从业人员：软硬件人员的安排	11
1.18. 2014年嵌入式行业从业人员：未来嵌入式操作系统首选	12
1.19. 2014年嵌入式行业从业人员：最想选择的嵌入式应用方向	12
1.20. 2014年嵌入式行业从业人员：获取专业知识的途径	13
1.21. 2014年嵌入式行业从业人员：首选搜索引擎	13
1.22. 2014年嵌入式行业从业人员：企业人才需求现状	14
1.23. 2014年嵌入式行业从业人员：对工作薪资的满意度	14
1.24. 2014年嵌入式行业从业人员：如何看待培训	15
2.1. ARM的Cortex-M3架构概述	25
2.2. 传统Linux和Android的不同的开发所处层次关系对比图	27
3.1. PIC32入门套件	33
3.2. PIC32开发工具	34
7.1. SD原始时序图	44
7.2. 结果解码后的SD时序效果图	44
8.1. PIC32系列产品中有各种不同资源	46
8.2. 嵌入式开发流程图	47
13.1. 用于取代其他内部总线的MIPI总线	54

表格清单

2.1. 嵌入式Linux驱动开发期间相关的内容	20
--------------------------------	----

正文之前

1. 此文目的

此文主要介绍的嵌入式软件开发的基本概念和组成，以及大概包含哪些东西，如何开发的基本流程等等内容，目的是为了搞懂如何入手嵌入式软件开发。

2. 联系方式

有任何意见、建议，都欢迎[联系我Crifan](http://www.crifan.com/about/me/)¹

¹ <http://www.crifan.com/about/me/>

第 1 章 嵌入式软件开发相关背景知识

嵌入式是一个很大的概念，从不同的侧面去介绍，可以有不同的解读。

1.1. 嵌入式行业和领域

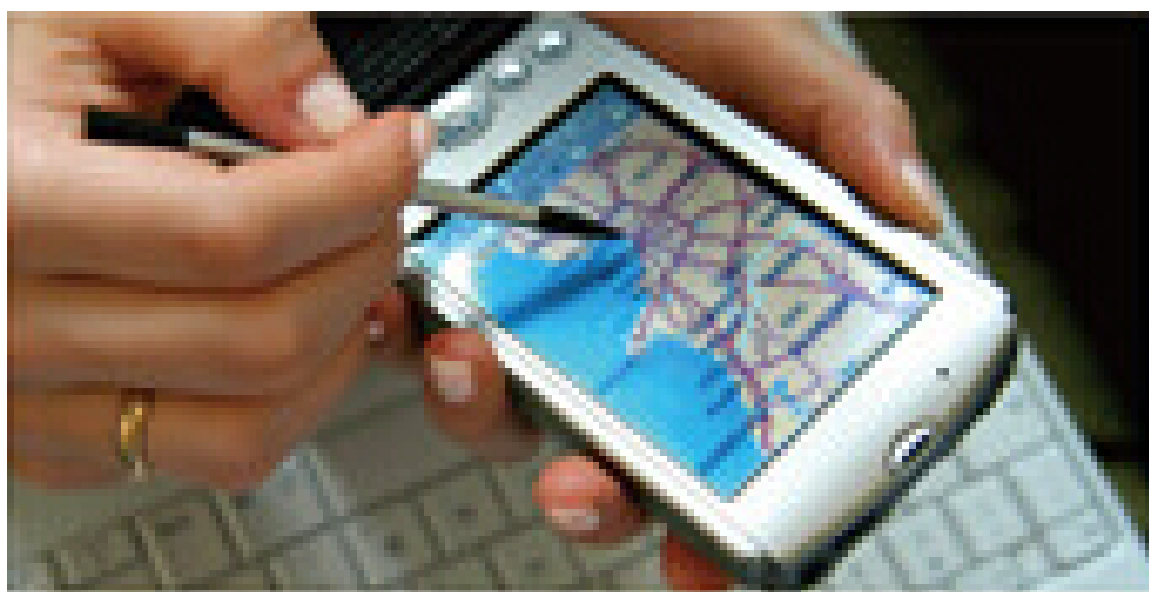
下面就先来说说，嵌入式行业，嵌入式的应用领域，都有哪些：

- 消费类数码产品
各种解决方案，各种嵌入式的CPU（MCU，SOC，DSP等等），对应的各种软件开发。

对应着各种产品，比如：

- MP3
- 音响
- 手机
- PAD
- 相机
- 便携式导航

图 1.1. 嵌入式应用之便携式导航



- 游戏机外设
- 打印机
- 工业控制和控制自动化
 - 生产线上的机械手臂，细分叫做PLC
 - 测量设备
如流量仪，阀门控制器等等，涉及到现场总线等技术

- 工业触摸屏控制系统
- 医疗电子器械
医疗行业内的嵌入式设备的软件开发
- 汽车电子
除了技术，还有很多标准，比如SPICE等
- 仪器仪表
- 智能家居

图 1.2. 嵌入式应用之智能家居



- 智能交通
- 车载设备
- 物联网终端
- 安防监控

图 1.3. 嵌入式应用之安全防护



- 远程信息处理
- 教学实验设备
- 监控设备与各种人机界面
- 节能环保
- 能源节能
- 电力系统

图 1.4. 嵌入式应用之电力系统



- 通讯系统

- 纺织行业
- 数控行业
- 机器人视觉
- 机器人视觉
- 媒体处理无线应用
- 数字家电
- 通信设备
- 网络终端

注意的是：其中，具体开发时，涉及的技术和任务不同，有的严格意义上来说，并不属于，偏底层的嵌入式，比如，android的app开发，其实是主要是用java写上层应用，不是那种，为某特定开发板，比如把android移植到三星手机的硬件平台，涉及到的底层驱动开发，这才是严格意义上的底层驱动开发。关于Android不是传统意义上的嵌入式的解释详见：

1.2. 嵌入式产业链

下面以一个典型的例子，来阐述，什么是嵌入式产业链。只有明白了嵌入式产业链的基本概念，才能真正理解后面要介绍到的嵌入式软件开发所处的层次和地位。

从芯片研发、量产、方案提供商、产品，中间会涉及到：

- arm专门设计cpu的ip core软核
- 三星制作cpu sc2440
- 某公司提供方案
用该芯片，加上外设，提供方案，包括cpu,各种外设，以及嵌入式操作系统，整套的软硬件方案
- 然后卖方案给某品牌商
比如爱国者，用此方案去，找设计人员设计外壳
- 再找模具厂生产模具
- 找代工厂生产整套的核心电路板
最后组装成
- 消费者
看见的，成品，电子书。

大概过程就是如此。

而你做的事情就是，在该卖方案的公司中，为那套方案中的软件部分，即嵌入式linux写各种外设和接口的驱动，比如nand flash驱动，sd卡驱动，i2c驱动等。这才是一种典型的嵌入式开发。然后你才知道你处在整个产业链中的什么位置，知道你在干什么。

1.3. 嵌入式行业概览

关于嵌入式行业的从业情况，包括公司，人员，技术，工具等等方面的概况，详细很多人和我一样，都是希望有个宏观的了解的。

后来巧了，发现华清远见每年都这方面的调查，此处摘录当前最新的2014年的[2013-2014年中国嵌入式开发从业人员大调查](http://www.farsight.com.cn/subject/survey2013-2014.htm)¹

¹ <http://www.farsight.com.cn/subject/survey2013-2014.htm>

的结果：[2013-2014（第六届）中国嵌入式开发从业人员调查报告隆重推出](http://www.farsight.com.cn/subject/report2013-2014.htm)²

对应报告是：[2013-2014年（第六届）中国嵌入式开发从业人员调查报告](http://download.farsight.com.cn/download/pdf/EmbeddedReport2013-2014.pdf)³

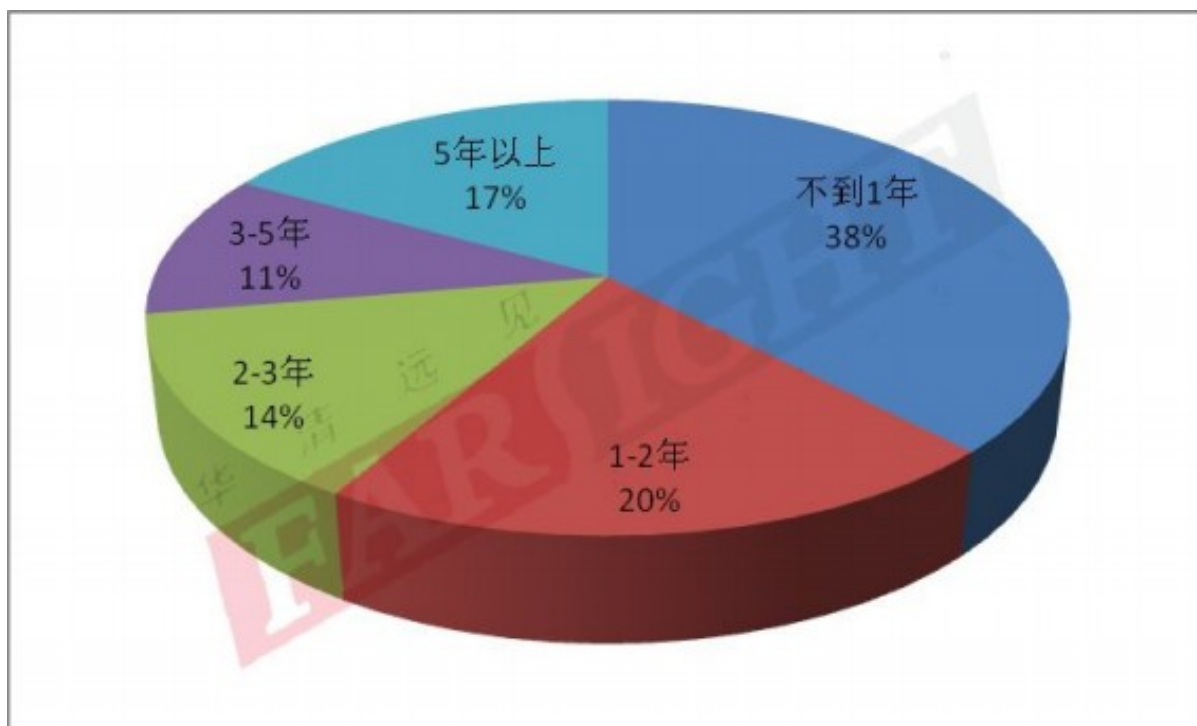
顺带列上之前几年的调查结果：

- [2012-2013年中国嵌入式开发从业人员调查报告](http://download.farsight.com.cn/download/pdf/EmbeddedReport2012-2013.pdf)⁴
- [2011-2012年中国嵌入式开发从业人员调查报告](http://download.farsight.com.cn/download/pdf/EmbeddedReport2011-2012.pdf)⁵
- [2010-2011年中国嵌入式开发从业人员调查报告](http://download.farsight.com.cn/download/pdf/EmbeddedReport2010-2011.pdf)⁶
- [2009-2010年中国嵌入式开发从业人员调查报告](http://download.farsight.com.cn/download/pdf/Embedded-Report2.pdf)⁷
- [2008-2009年中国嵌入式开发从业人员调查报告](http://download.farsight.com.cn/download/pdf/Embedded-Report.pdf)⁸

下面列出2014年的结果如下，供大家参考：

1.3.1. 2014年嵌入式行业从业人员：工作经验

图 1.5. 2014年嵌入式行业从业人员：工作经验



² <http://www.farsight.com.cn/subject/report2013-2014.htm>

³ <http://download.farsight.com.cn/download/pdf/EmbeddedReport2013-2014.pdf>

⁴ <http://download.farsight.com.cn/download/pdf/EmbeddedReport2012-2013.pdf>

⁵ <http://download.farsight.com.cn/download/pdf/EmbeddedReport2011-2012.pdf>

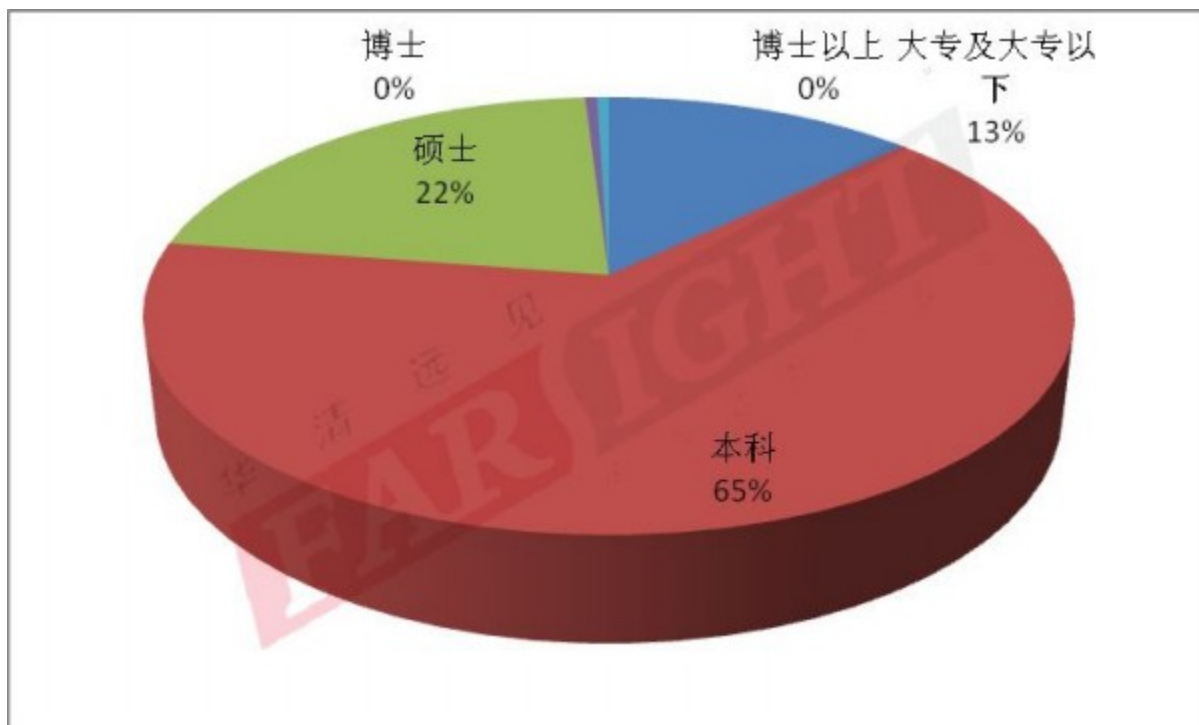
⁶ <http://download.farsight.com.cn/download/pdf/EmbeddedReport2010-2011.pdf>

⁷ <http://download.farsight.com.cn/download/pdf/Embedded-Report2.pdf>

⁸ <http://download.farsight.com.cn/download/pdf/Embedded-Report.pdf>

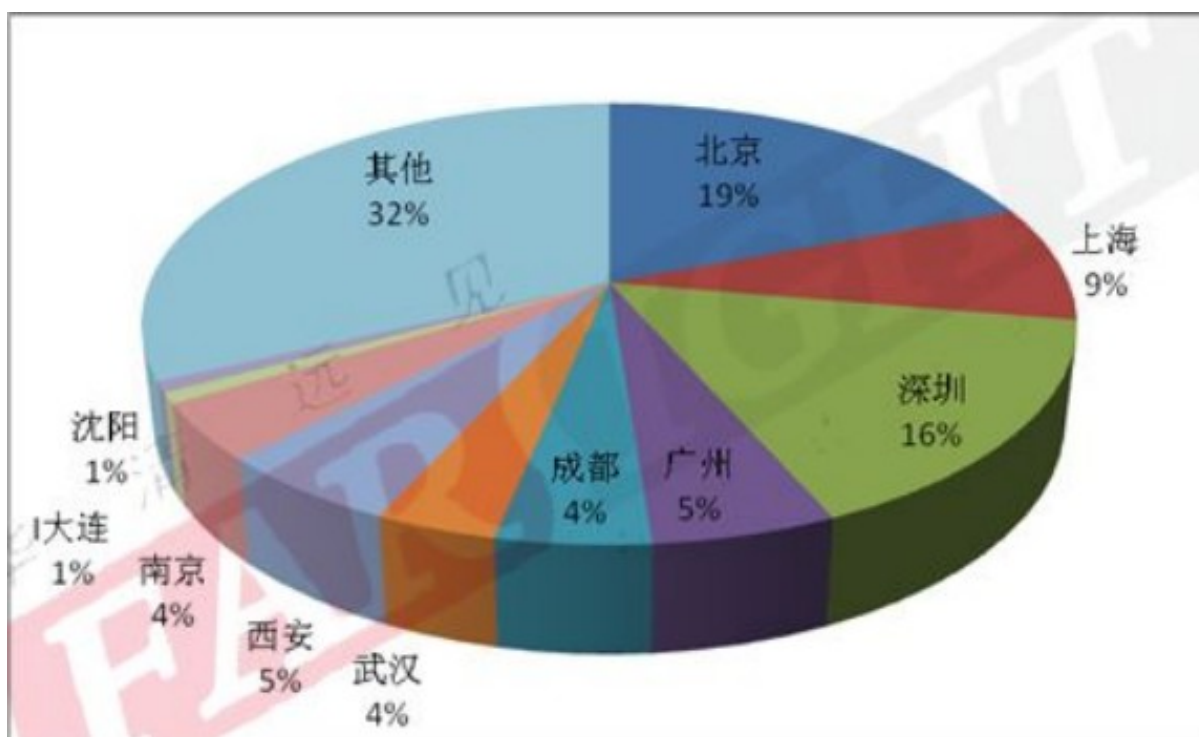
1.3.2. 2014年嵌入式行业从业人员：学历要求

图 1.6. 2014年嵌入式行业从业人员：学历要求



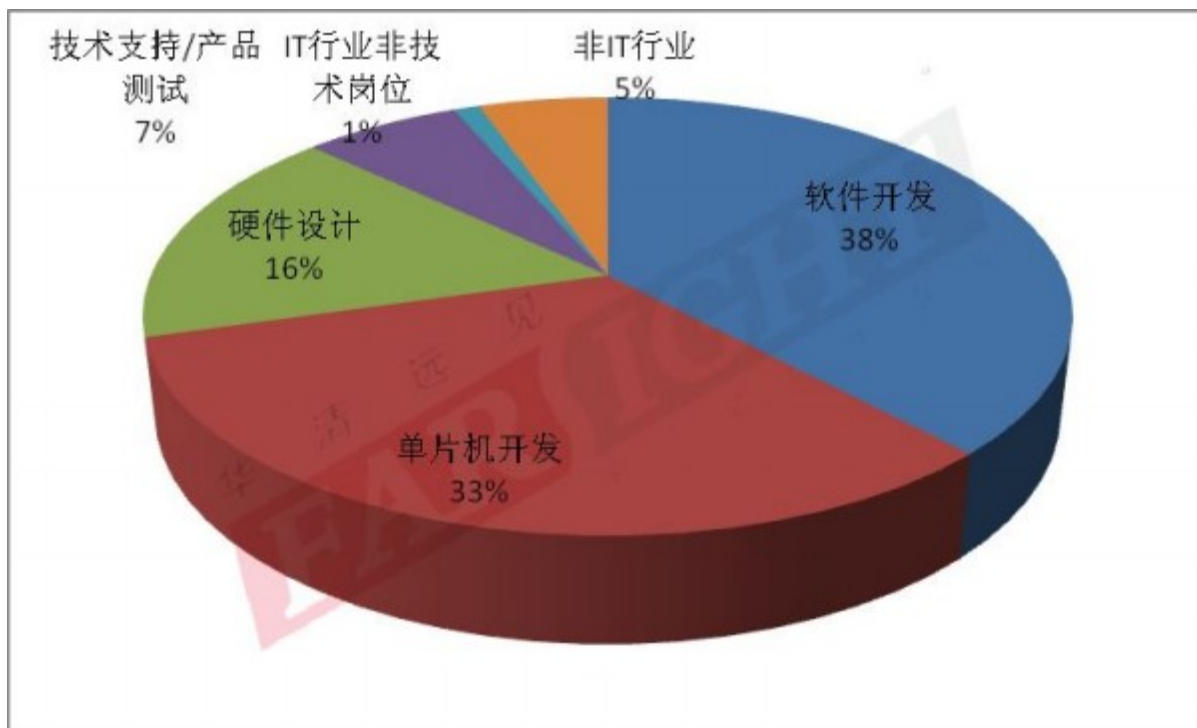
1.3.3. 2014年嵌入式行业从业人员：地域分布

图 1.7. 2014年嵌入式行业从业人员：地域分布



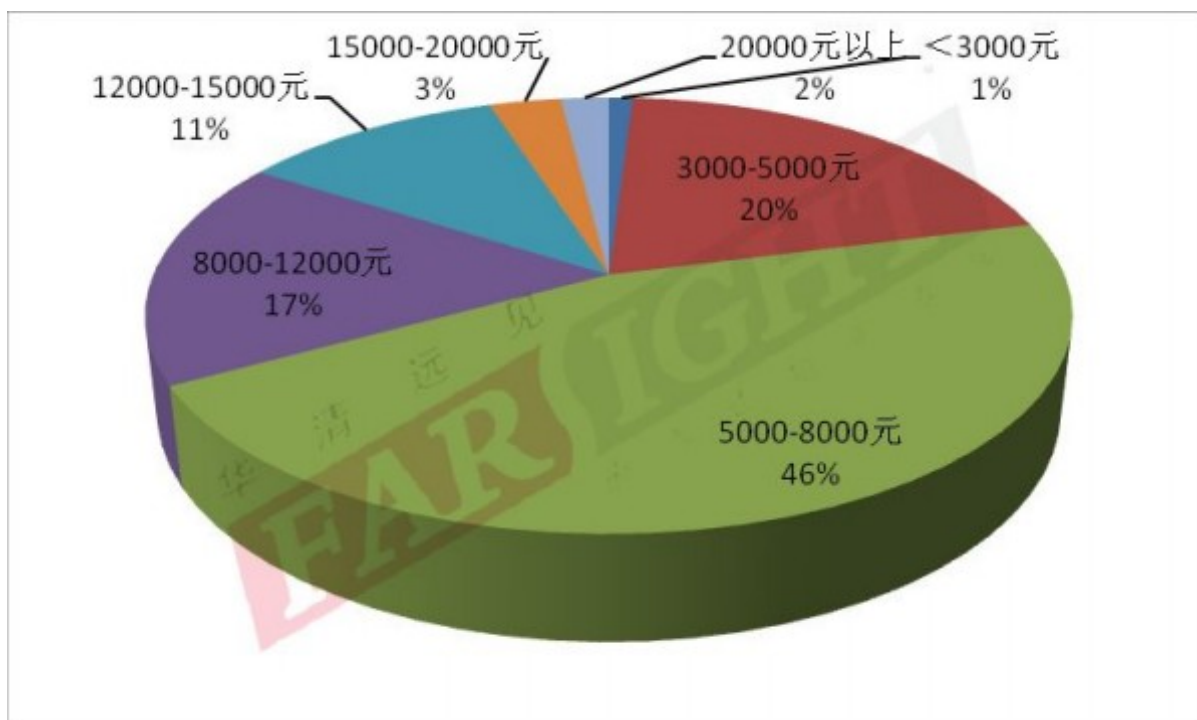
1.3.4. 2014年嵌入式行业从业人员：嵌入式开发前的研发方向

图 1.8. 2014年嵌入式行业从业人员：嵌入式开发前的研发方向



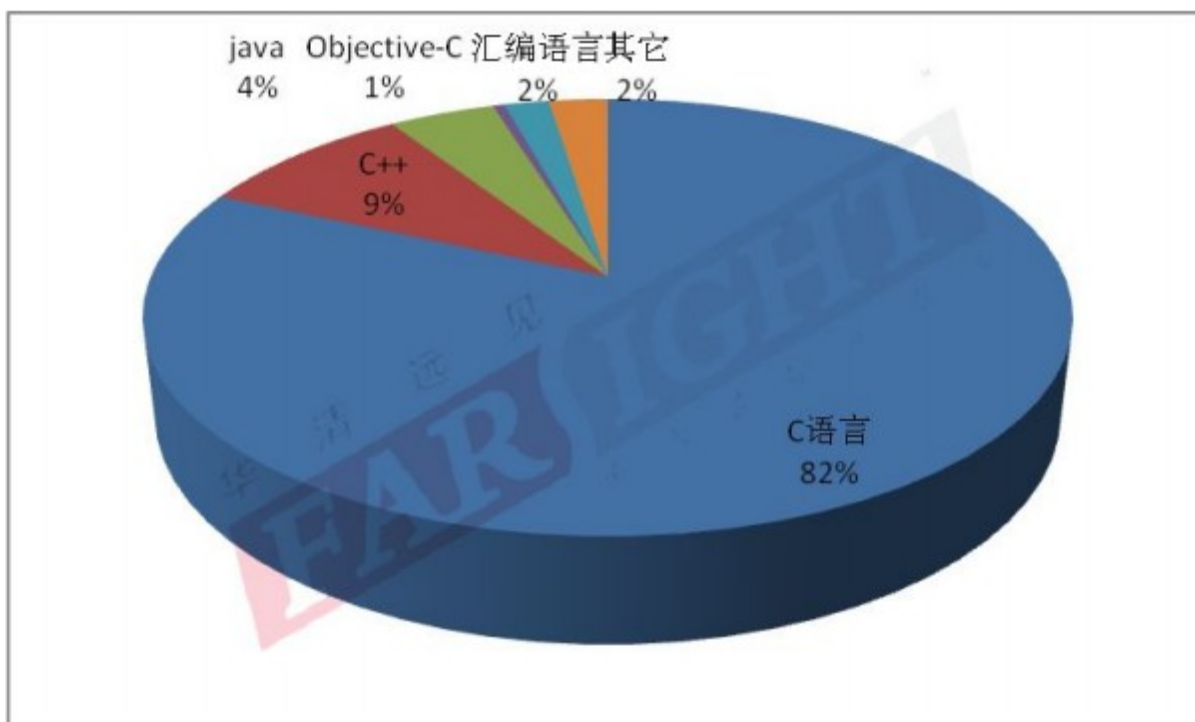
1.3.5. 2014年嵌入式行业从业人员：薪资水平

图 1.9. 2014年嵌入式行业从业人员：薪资水平



1.3.6. 2014年嵌入式行业从业人员：软件开发语言的选择

图 1.10. 2014年嵌入式行业从业人员：软件开发语言的选择



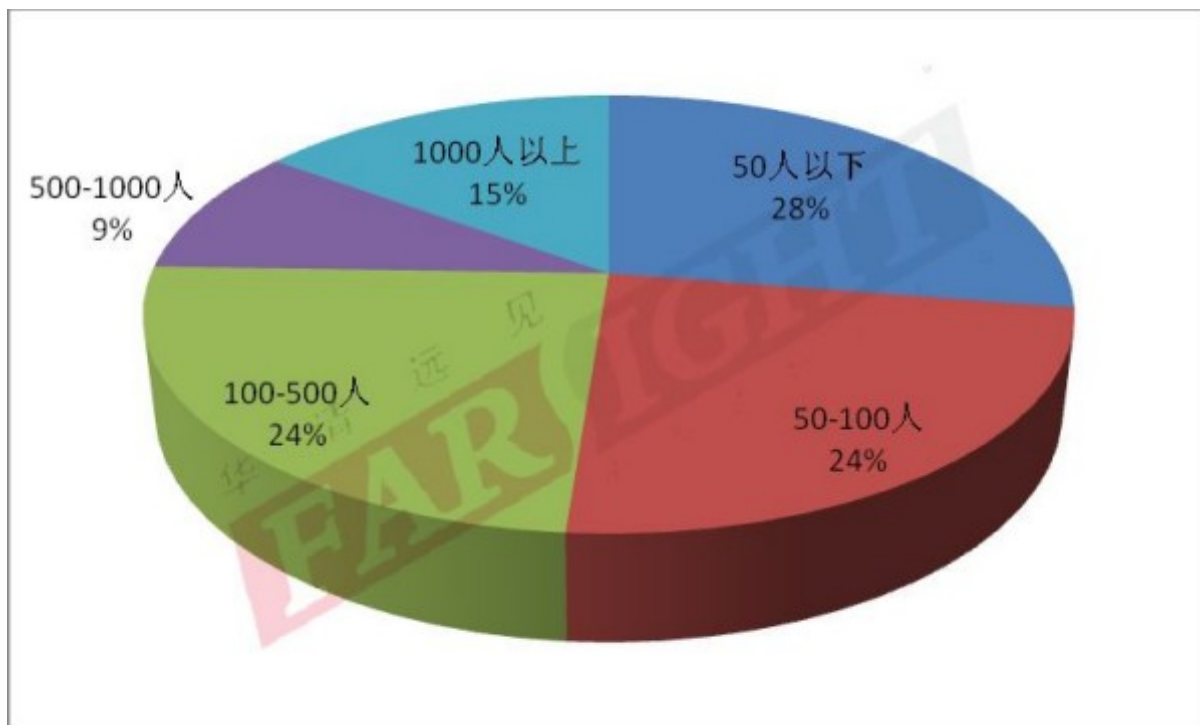
1.3.7. 2014年嵌入式行业从业人员：所属行业分布情况

图 1.11. 2014年嵌入式行业从业人员：所属行业分布情况



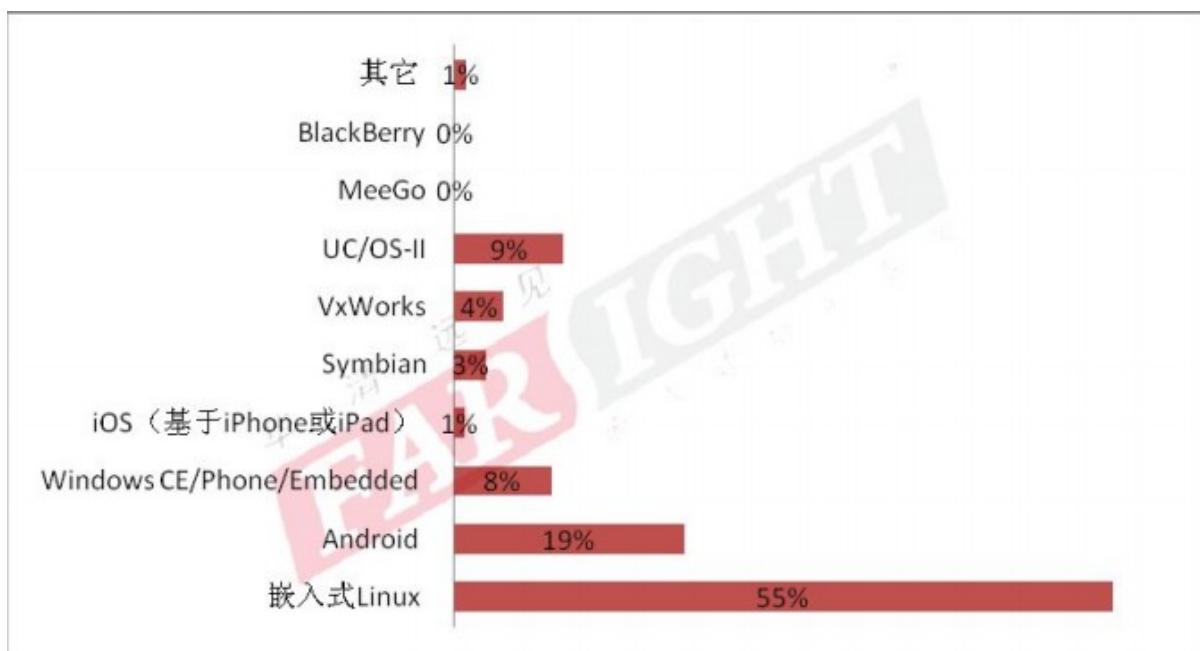
1.3.8. 2014年嵌入式行业从业人员：公司规模

图 1.12. 2014年嵌入式行业从业人员：公司规模



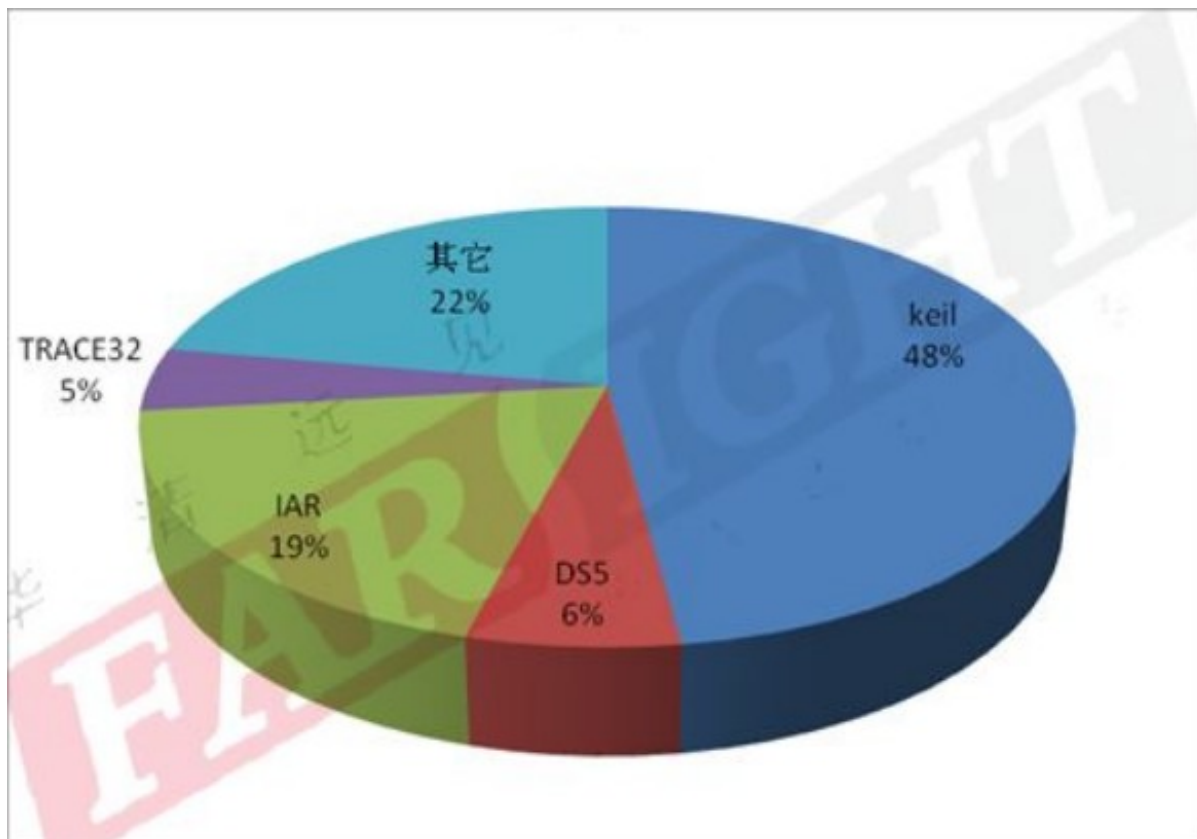
1.3.9. 2014年嵌入式行业从业人员：软件开发平台的选择

图 1.13. 2014年嵌入式行业从业人员：软件开发平台的选择



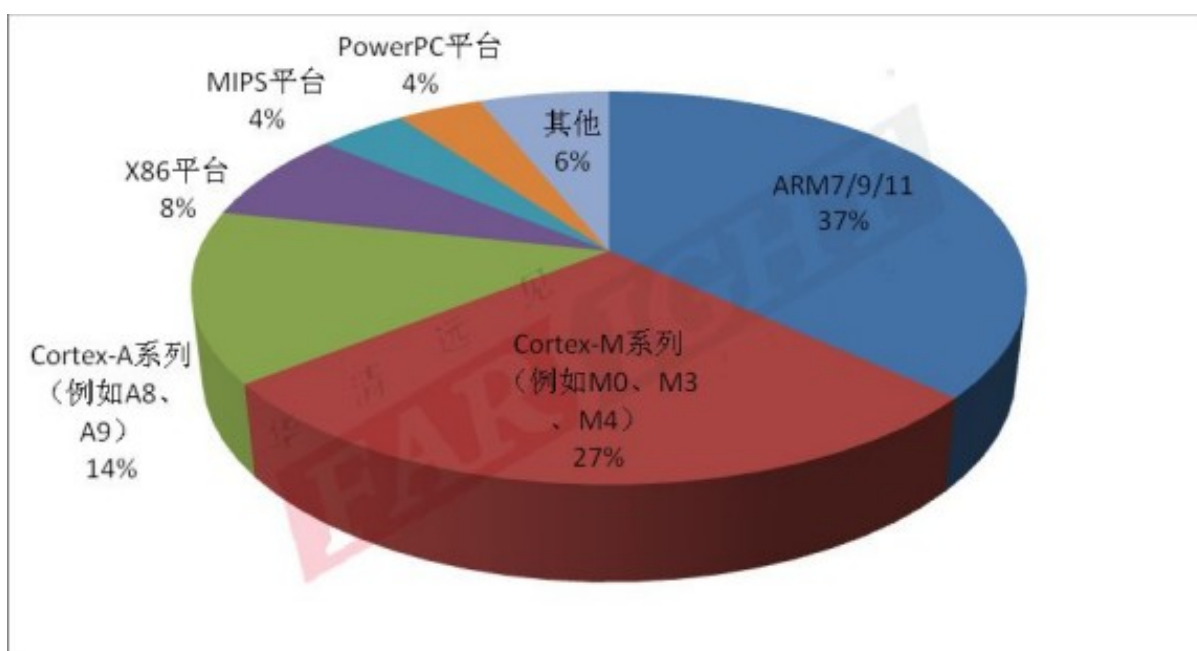
1.3.10. 2014年嵌入式行业从业人员：软件开发调试工具的选择

图 1.14. 2014年嵌入式行业从业人员：软件开发调试工具的选择



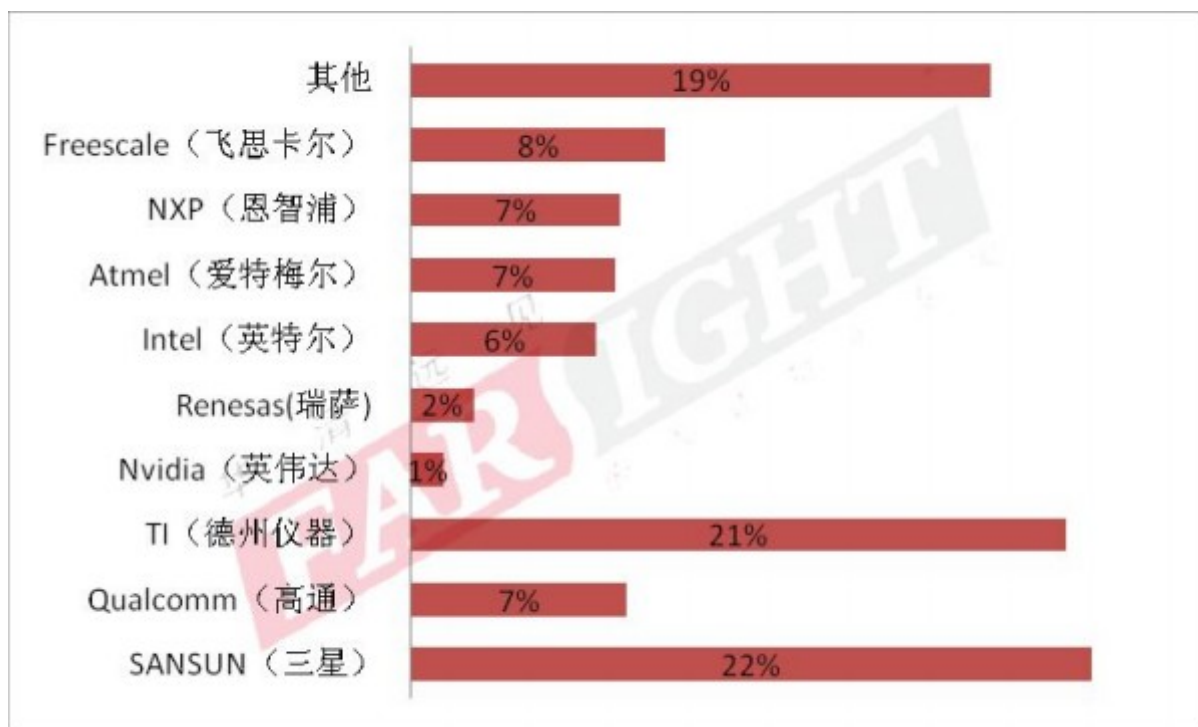
1.3.11. 2014年嵌入式行业从业人员：硬件开发平台的选择

图 1.15. 2014年嵌入式行业从业人员：硬件开发平台的选择



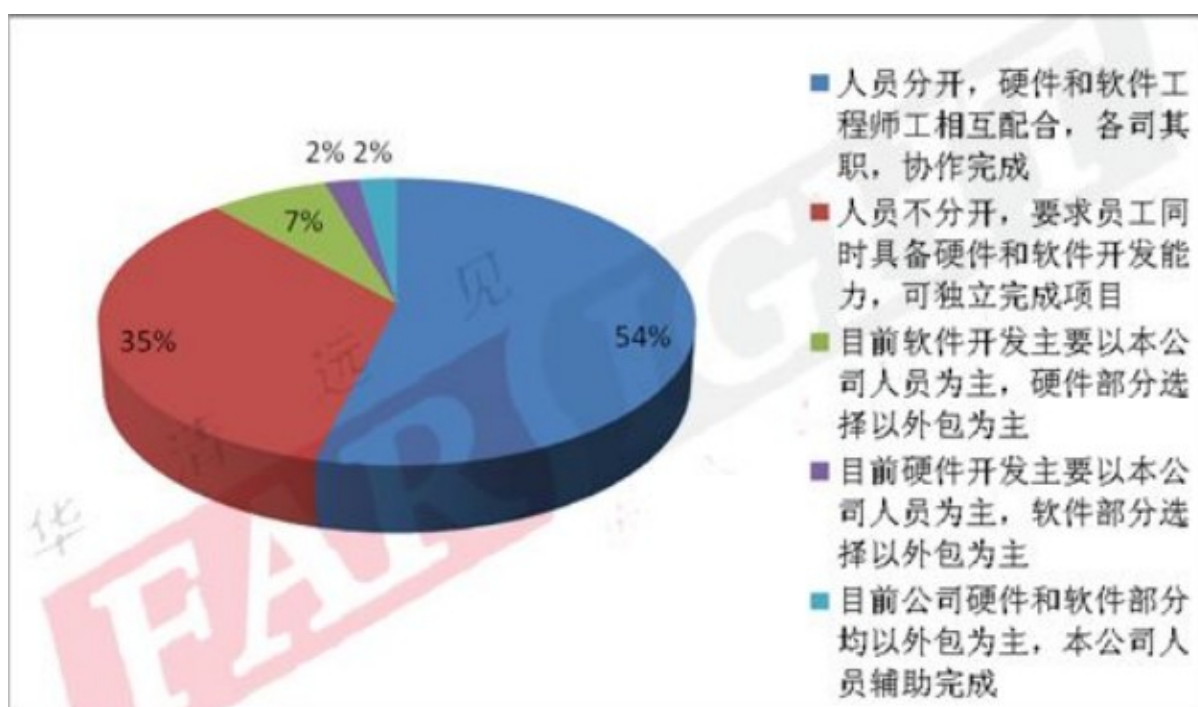
1.3.12. 2014年嵌入式行业从业人员：处理器芯片的选择

图 1.16. 2014年嵌入式行业从业人员：处理器芯片的选择



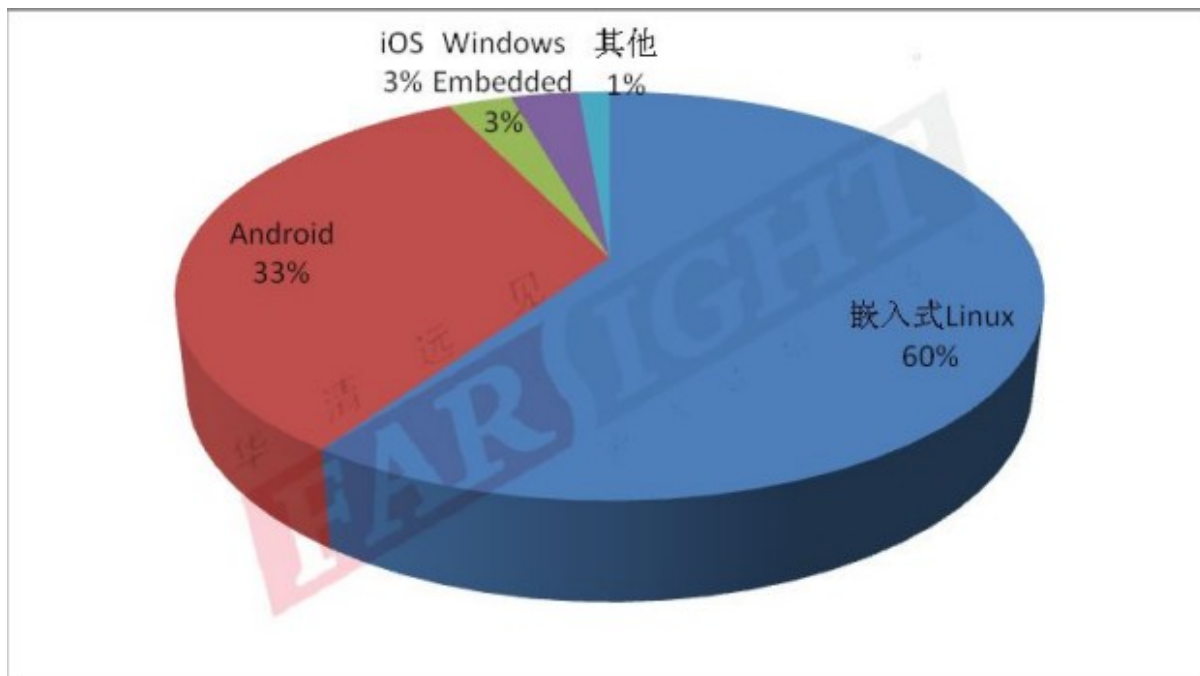
1.3.13. 2014年嵌入式行业从业人员：软硬件人员的安排

图 1.17. 2014年嵌入式行业从业人员：软硬件人员的安排



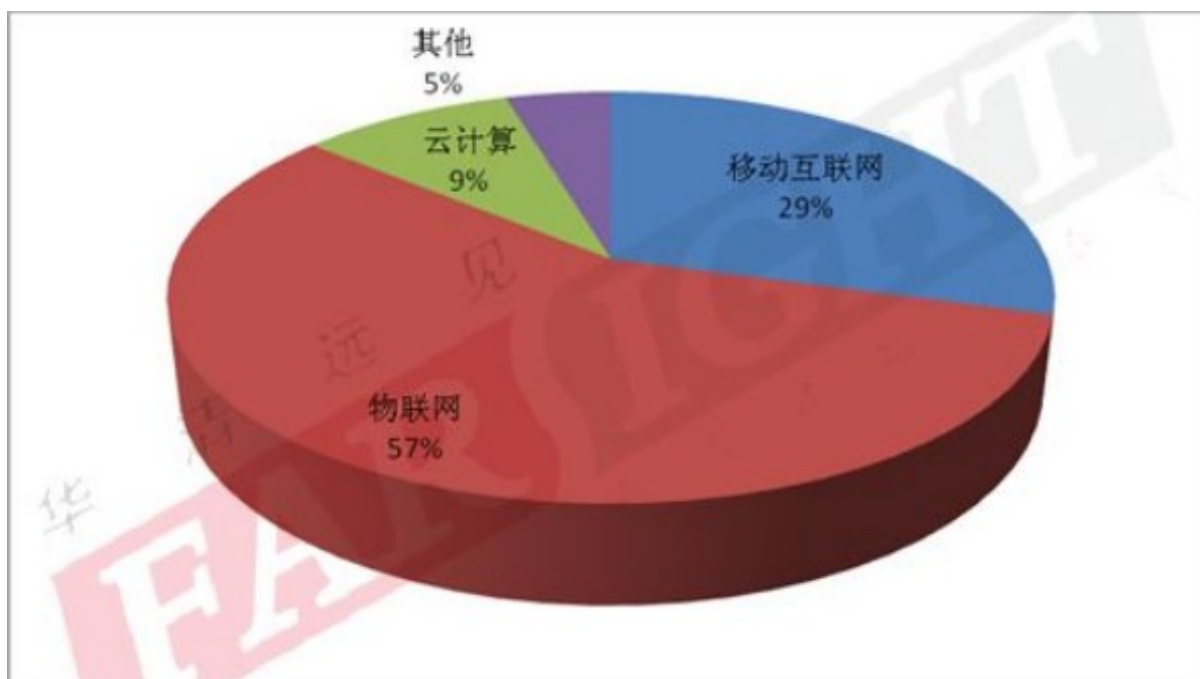
1.3.14. 2014年嵌入式行业从业人员：未来嵌入式操作系统首选

图 1.18. 2014年嵌入式行业从业人员：未来嵌入式操作系统首选



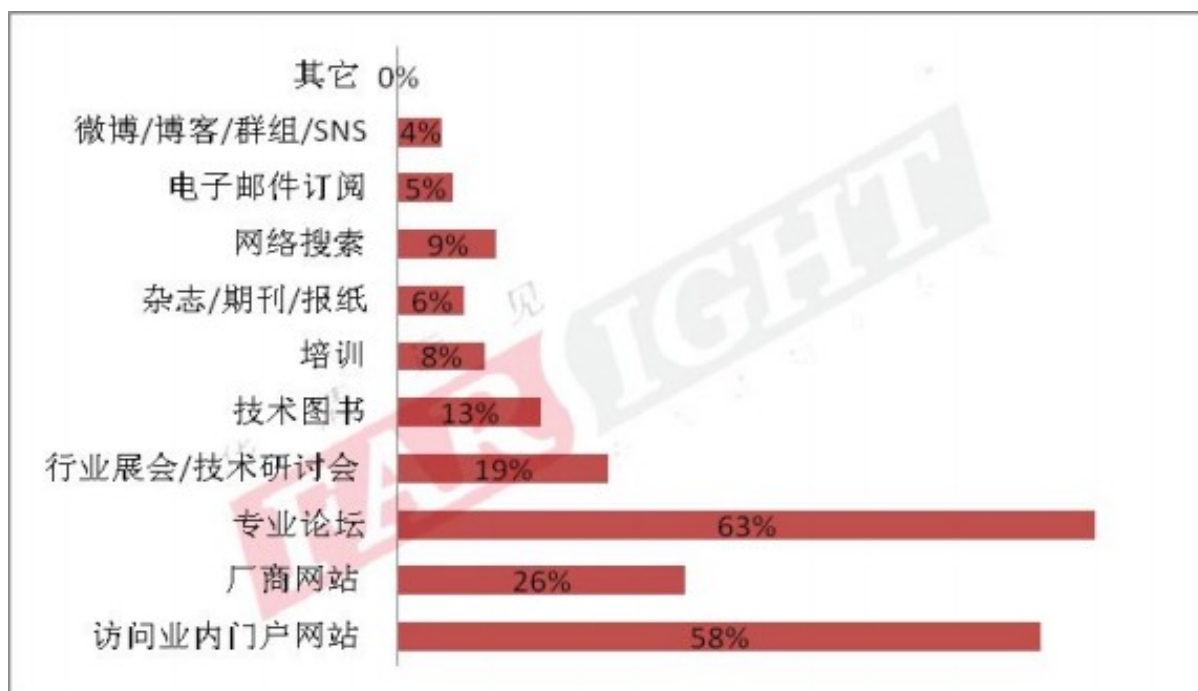
1.3.15. 2014年嵌入式行业从业人员：最想选择的嵌入式应用方向

图 1.19. 2014年嵌入式行业从业人员：最想选择的嵌入式应用方向



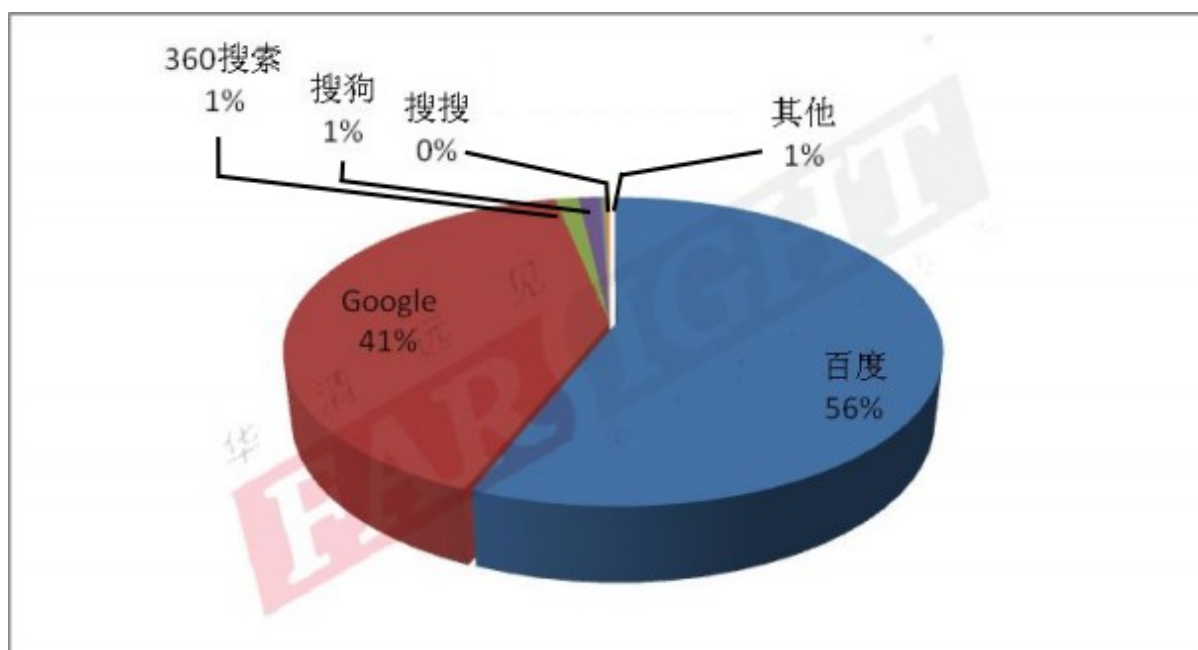
1.3.16. 2014年嵌入式行业从业人员：获取专业知识的途径

图 1.20. 2014年嵌入式行业从业人员：获取专业知识的途径



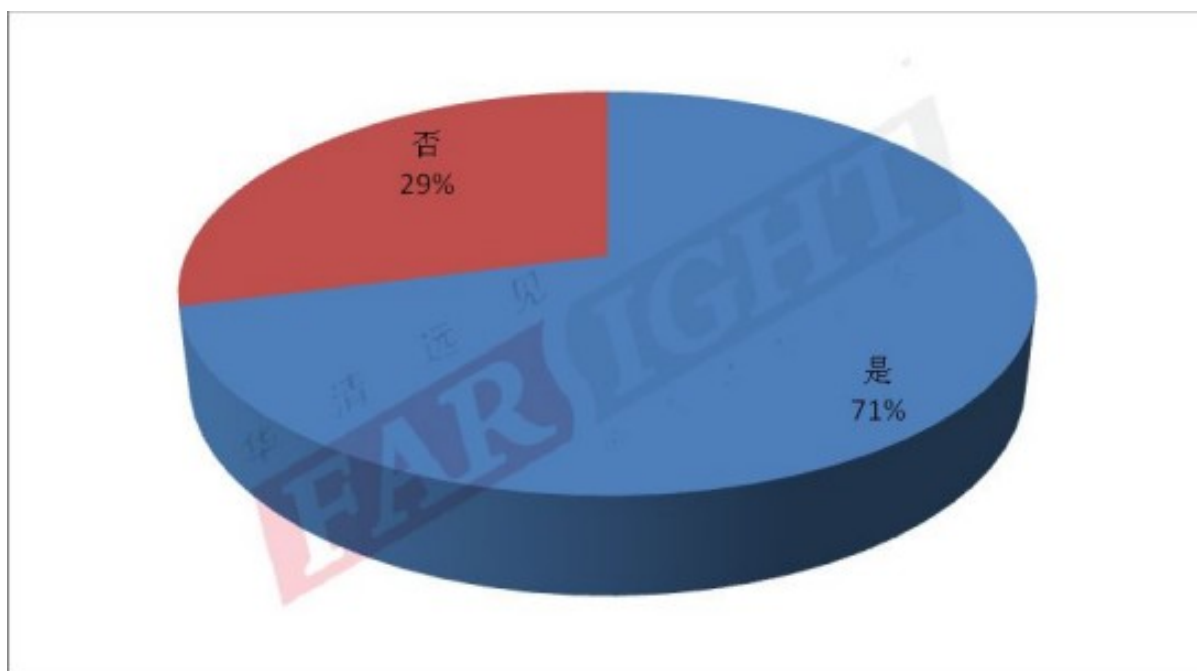
1.3.17. 2014年嵌入式行业从业人员：首选搜索引擎

图 1.21. 2014年嵌入式行业从业人员：首选搜索引擎



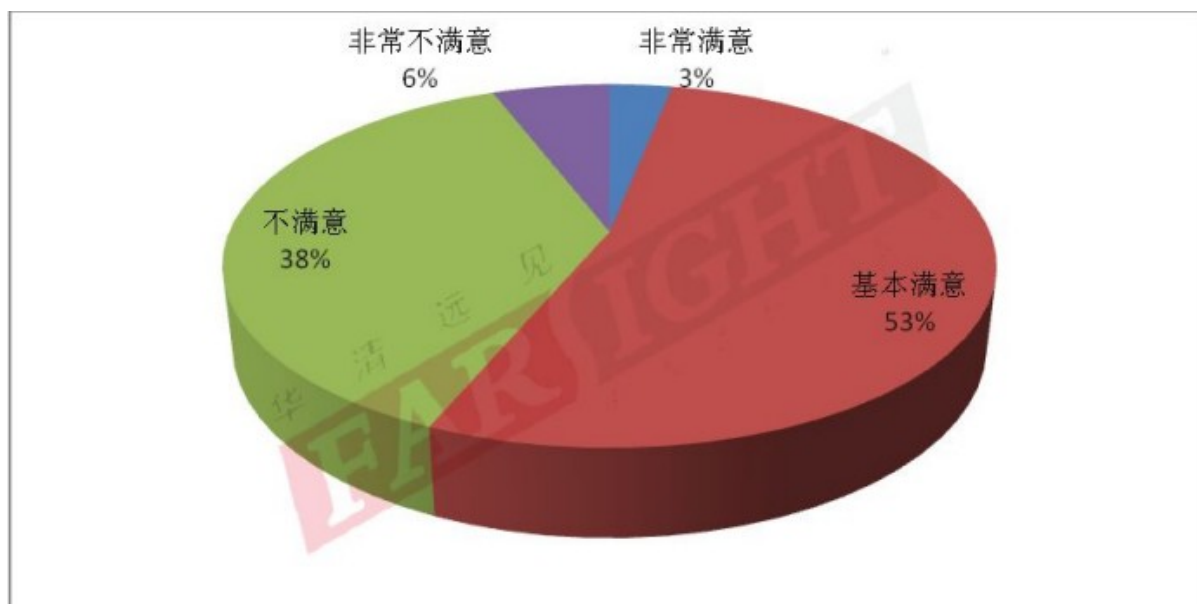
1.3.18. 2014年嵌入式行业从业人员：企业人才需求现状

图 1.22. 2014年嵌入式行业从业人员：企业人才需求现状



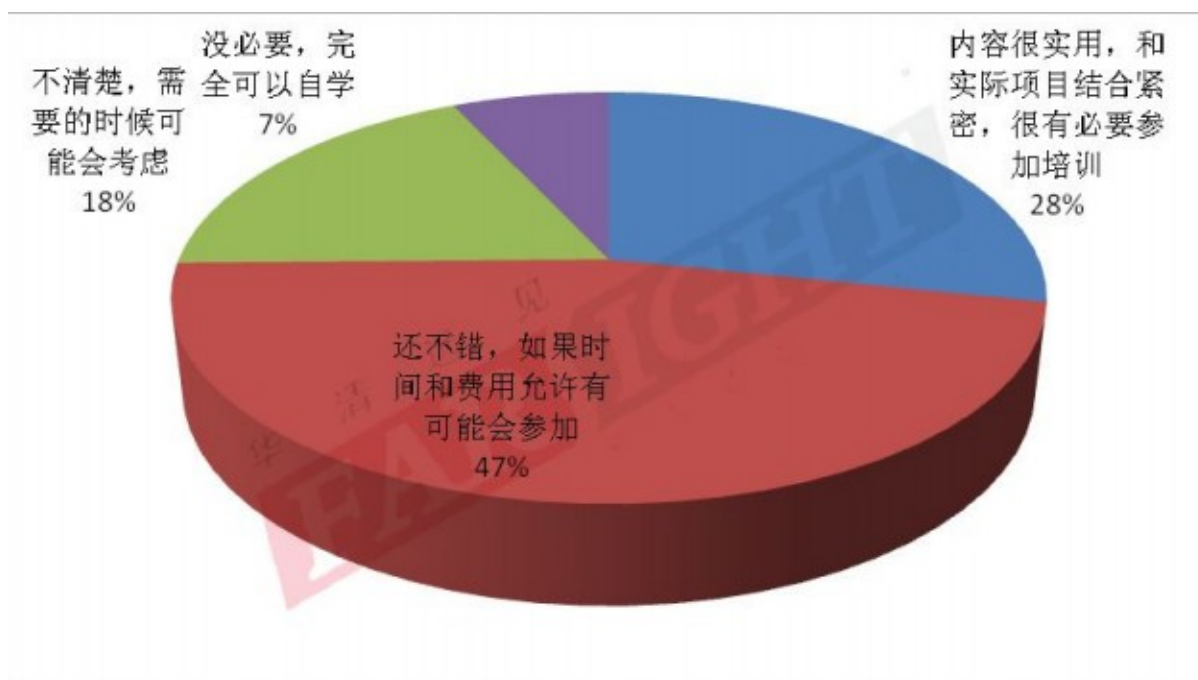
1.3.19. 2014年嵌入式行业从业人员：对工作薪资的满意度

图 1.23. 2014年嵌入式行业从业人员：对工作薪资的满意度



1.3.20. 2014年嵌入式行业从业人员：如何看待培训

图 1.24. 2014年嵌入式行业从业人员：如何看待培训



1.4. 嵌入式软硬件厂商的合作关系

搞嵌入式开发，对于一些背景知识，也需要了解一下，比如：

嵌入式领域内的软硬件厂商是互相合作的，以及各种不同公司之间的合作关系

比如：之前去购买蓝牙HART猫：

[【记录】用PayPal从国外在线网站上购买蓝牙HART猫](#)⁹

卖家是<http://www.mactekcorp.com/>的Garry Cusick，其是销售总监

Garry Cusick针对订单的情况，去找公司内部Thomas Holmes核实之后，Thomas Holmes再去安排发货的事情。

但是去发货时，[【记录】查询Fedex的国际快递](#)¹⁰，

却是从该公司的合作伙伴<http://www.avid-tech.com/>的Erica Deliere去给我发货的。

由此可见MACTeck和Avid-tech应该是合作伙伴，并且很可能是：

作为研发蓝牙HART猫的MACTeck，虽然具有对应的技术，但是没有生产部门，也不负责生产，所以最终量产时找的是Avid-tech这家代工厂

所以，最终普通用户去购买产品时，最终是从代工厂Avid-tech去发的货。

然后Avid-tech再去选择Fedex这个国际通用的物流公司，去把东西寄给国外的用户

相应的，作为买家的我会收到：

⁹ http://www.crifan.com/record_use_paypal_buy_bluetooth_hart_modem_from_foreign_online_website/

¹⁰ http://www.crifan.com/query_fedex_global_delivery/

- A：来自Avid-tech的邮件，告诉我说已经用Fedex发货给我了，且给出了订单号
- B：我也会收到从Fedex系统中发出的发货通知，也可以用订单号去查询和最终货物的物流详情。

总之：

搞软件开发，搞嵌入式软件开发，对于各种相关公司的关系和背景，也要有所了解，否则都弄不懂东西是谁做的，而无法及时有效的沟通。

另外一个证明各个厂商之间的合作关系的例子：

segger的[emWin的升级](#)¹¹，就以二进制库的形式，支持合作客户去升级，包括：

- ARM/KEIL as part of the MDK Pro package
- Cypress as download from cypress.com
- Infineon as part of the Dave3 installation
- NXP as download from lpcware.com
- Silicon Labs as part of the Simplicity Studio for EFM32 devices
- ST as part of the STemWin graphic software

等等。

其他还有：Segger是ARM的合作伙伴

通过http://www.arm.com/community/partners/display_product/rw/ProductId/2210/的Partners¹²可以看到：[SEGGER Microcontroller GmbH & Co. KG](#)¹³的[embOS, Real-Time Operating System](#)¹⁴

从中可以看出segger是ARM的合作伙伴

然后通过http://www.arm.com/community/partners/all_partners.php发现，ARM的合作伙伴，有一堆，非常多。。。

¹¹ <http://www.segger.com/emwin-source-code-upgrade.html>

¹² <http://www.arm.com/community/partners/>

¹³ http://www.arm.com/community/partners/display_company/rw/company/segger-microcontroller-gmbh-co-kg

¹⁴ http://www.arm.com/community/partners/display_product/rw/ProductId/2210/#

第 2 章 嵌入式软件开发的基本逻辑

2.1. 嵌入式软件开发的基本逻辑

包括嵌入式软件开发的知识体系和背景：

先说背景：

计算机，领域内，知识体系，按层次分，可以分为：

- 上层：纯软件
- 中间：嵌入式
- 底层：硬件

其中：

- 纯软件：无需实际接触硬件，写代码，也不会涉及到操作硬件
->当然你对硬件的工作方式，工作机制，特点等内容有了解，自然更容易设计出，写出高质量的软件代码
- 嵌入式：全称算是嵌入式软件
注意：对应的，很多时候，也有把嵌入式硬件，简称为嵌入式的。
- 嵌入式软件方面，更多的时候，写的代码，都是和硬件打交道的。

此处，暂且不讨论上层的纯软件，以及底层的硬件，只是简要概述一下，中间的嵌入式，尤其是嵌入式软件部分的知识体系，基本架构：

2.1.1. 按照是否有操作系统

按照是否有操作系统，可分为：

- 无OS的嵌入式系统
其中，无OS的，相对比较简单：一个大循环，

http://www.segger.com/admin/uploads/productDocs/UM01001_embOS_Generic.pdf

提到的那个Super Loop

- 有OS的嵌入式系统
有OS的，往往架构都和OS有关：

- 应用程序
 - 应用程序
 - 库函数
- 操作系统内核
 - 各个功能模块：网络，文件系统，进程通信，进程管理，存储管理
 - 硬件抽象层HAL
- 硬件

典型的嵌入式Linux软件系统，根据不同的阶段和模块分，又可以分为：

- Bootloader：最常见的就是Uboot了
- Rootfs：根文件系统
- Kernel：嵌入式Linux内核

更多详细的解释，请参见：[嵌入式Linux软件开发详解](#)¹

2.1.2. 按照操作系统是否是私有的

其中有OS的，按照是否是开源的，是否是常用的，通用的，还是私有分，可以分为：

- 开源的/常见的/通用的 嵌入式操作系统
开源的/常见的/通用的，并不是等价的，但是只是此处相对于私有的来说，从概念上和逻辑上，是等价的。

常见的有：嵌入式Linux（及其衍生物Android），QNX，VxWorks，ucOS/II，TRON（ITRON），WinCE等等。

- 私有的嵌入式操作系统

什么叫做，写的代码，是和硬件打交道的？

意思是：对于物理上，真实存在的硬件设备，写代码，操作对应的硬件设备，往往是写硬件的驱动程序之类的

什么叫做，写代码操作硬件设备？

本质上就是：写代码，设置，配置硬件

往往都是：读写硬件设备的寄存器，从而达到使得硬件可以工作，可以正常工作的目的。

什么叫做用代码驱动硬件工作，即什么叫做硬件设备驱动？

背景：

硬件有各种功能，特性；

目的：

希望硬件按照我们所希望的方式，去工作；

需要做的事情：

大体分两步：

1. 初始化硬件
2. 实现硬件数据的读写

举例：

¹ http://www.crifan.com/files/doc/docbook/embedded_linux_dev/release/html/embedded_linux_dev.html

一个典型的，在嵌入式Linux系统中，实现对应的SD卡的

背后的逻辑

我们的目标

以及具体如何去写SD卡驱动的大概步骤和逻辑就是：

SD卡，插入到开发板的SD卡插槽上，然后开发板中正在运行的嵌入式Linux系统，可以识别出该SD卡，并且可以实现SD的数据的读写：

- 读：把里面的数据拷贝出来，比如里面有张图片，可以把图片这个文件的数据拷贝出来
- 写：往里面写数据，比如想要拷贝一首歌曲到SD中

那么，我们就需要：

在嵌入式Linux系统中，为SD卡写驱动

其大概的逻辑和步骤是：

1. SD的初始化部分
去设置对应的SD卡的寄存器，申请各种相关的软硬件资源（此部分相关细节很多，待以后详解）
即把SD卡配置成你需要的工作模式了，然后再去实现后续的数据读写。

2. 让SD卡工作的部分：实现数据的读写
然后再去（Linux的相关的驱动框架下）写相关的SD的数据的读写的功能

操作对应的寄存器

把数据从SD的相关的数据寄存器中读出来

或者是

把要写入的数据，写入到SD的相关的数据寄存器中

其他相关的内容还有：为了提高数据读写速度，可以去利用DMA，如果有DMA可用的话

2.2. 嵌入式软件开发内容分类

根据开发内容的所属架构层次分，至少可以分为：

- 嵌入式驱动开发
比如在Linux系统中，开发各种设备驱动。
- 嵌入式应用开发
比如在Linux系统上面写各种应用。典型的例子是，在拿到了某芯片的原厂的SDK，比如已实现了各种驱动的嵌入式Linux，基于此，去写Linux的应用。

2.3. 嵌入式软件期间会涉及哪些内容

对于嵌入式领域中会听到，见到的，涉及到如此众多的领域，技术，芯片，公司，开发工具，软件平台等等名称，很多人在开始接触的时候，都会很晕。

而这里，尝试用初学者所能听得懂的方式去解释一些常见的名词，所处的地位和作用

->由此也验证了那句：搞嵌入式开发，尤其是底层开发，相对来说，难度还是要比上层软件要难的

接下来，我熟悉的嵌入式Linux中Nand Flash驱动开发为例来解释说明具体过程和所涉及到的内容：

假如你是个普通的嵌入式驱动软件开发者，你的老板给你安排一个嵌入式Linux中Nand Flash驱动开发的任务的时候，就会直接或间接接触到很多东西：

表 2.1. 嵌入式Linux驱动开发期间相关的内容

嵌入式Linux中Nand Flash驱动开发期间直接涉及到的	嵌入式Linux中Nand Flash驱动开发期间间接相关的，类似的，类比的	备注
首先，此处所从事的开发领域，是一个消费类数码产品，比如电子书	于此对应的，同级别的，其他开发领域有： 工业控制 ， 医疗器械 ， 汽车电子 等等行业 ->其他不同行业内，所要接触到的开发任务，往往又是完全不同的任务 ->比如工业控制领域所涉及到的 现场总线Fieldbus ，对于搞消费类数码的人来说，甚至可以说很多名词和概念都没听过。 ->更别说，让你去折腾 HART ， FF (Foundation Fieldbus) ， Profibus ，了，其和消费类数码领域，很多技术都是完全不同和不相关的	相关内容： 现场总线Fieldbus简析²
其次，此处的操作系统用的是嵌入式Linux系统 ->目前在嵌入式领域内使用可以说是最广泛的	与此对应的，同级别的，其他的，嵌入式操作系统也有很多，比如： ucOS/II ， VxWorks ，等等。	
而对于Nand Flash，作为其中一种设备，一种接口，与此对应的也有其他很多协议/设备/接口，比如： Nor Flash ， Bluetooth ， USB ， RS232 ， SD/MMC 等等；	->换句话说，有可能过些天，你的老板让你去实现Linux中的SD/MMC驱动。	
而接着你就开始去折腾Linux下面的Nand Flash芯片驱动的开发了，然后你可能就会接触到很多很多相关知识了：假如所用的硬件，先是嵌入式开发板，所用 SoC 芯片是 S3C2440 ，是基于 ARM 的 ARM920T 内核的芯片：		
嵌入式开发板	->网上也有很多其他 开发板 ，就是用来卖给你们，用来折腾的，用来搞嵌入式开发的->比如你可以花点钱，买个对应到板子，用于练手，练习各种驱动的或应用的开发；	
SoC	->>此处的嵌入式CPU，很多时候，用的是另外一个标准叫法，SoC，而与SoC类似（但不相同的）概念还有 MCU ， DSP ， ASIC ， FPGA 等等；	
S3C2440	->此处是三星的芯片S3C2440，三星自己家的其他基于ARM内核的芯片还有很多，比如 S3C2410 ， S3C6410 等等。而与此类似的	

² http://www.crifan.com/files/doc/docbook/fieldbus_intro/release/html/fieldbus_intro.html

嵌入式Linux中Nand Flash驱动开发期间直接涉及到的	嵌入式Linux中Nand Flash驱动开发期间间接相关的，类似的，类比的	备注
	->类似三星的，也还有其他厂商，可以有能力自己去设计MCU，SoC，比如常见的意法半导体ST Microelectronics的STM32系列的芯片，比如STM32F103VGT6，其是ARM的Cortex-M3内核的，所以全称一般叫做：STM32F103VGT6 ARM Cortex-M3 MCU	
ARM	->ARM是一家IP公司，专门设计CPU内核，所谓的IP Core，但不生产。 ->其他芯片设计公司，比如三星 Samsung，飞思卡尔Freescale等等去用ARM的内核设计自己的SoC或MCU。 ->与此相对的，另外也有一家做IP的公司叫MIPS	
ARM920T	->类似的内核系列还有ARM922T，ARM926EJS等等，以及后来出来的ARM Cortex系列，Cortex-M3是比较流行的一个系列	
		现在很多开发板，都是采用比较新的，流行的SoC，而也是基于ARM内核的，比如最新一点的ARM11，ARM Cortex-M3等等；
然后再接着说工作细节中涉及到的Nand Flash芯片方面的知识，此处假设用的Nand Flash芯片是三星的K9GAG08U0M		
Nand Flash	->和Nand Flash对应的Flash技术还有Nor Flash，以及其他更进一步的封装后的LBA Nand Flash，eMMC等等。	
三星的Nand Flash	->与此对应的是，其他还有很多Nand Flash芯片厂商，比如Micron镁光，东芝Toshiba，Hynix海力士，闪迪SanDisk等等，Intel英特尔	
K9GAG08U0M	->其他还有很多常用的三星的Nand Flash芯片，比如K9LBG08UXD等等。	
然后再说开发所涉及到的软件，包含很多方面，一个个说：交叉编译环境，嵌入式Linux环境构成等等		
交叉编译环境	使用交叉编译工具链：已有交叉编译器，配置好环境就可以用来交叉编译了；	相关内容： 交叉编译详

³ http://www.crifan.com/files/doc/docbook/cross_compile/release/html/cross_compile.html

⁴ http://www.crifan.com/files/doc/docbook/crosstool_ng/release/html/crosstool_ng.html

嵌入式Linux中Nand Flash驱动开发期间直接涉及到的	嵌入式Linux中Nand Flash驱动开发期间间接相关的，类似的，类比的	备注
	制作交叉工具链：没有交叉编译器，则需要自己去制作交叉编译器，比如用Buildroot，crosstool-ng等等去制作	解³ ， crosstool-ng详解⁴
整套的嵌入式Linux环境：uboot+kernel+rootfs	典型的嵌入式Linux开发环境主要构成就是：uboot+kernel+rootfs , ->与嵌入式Linux对应的有，WinCE，其又是不一样的，自己一套的开发环境 , ->期间就可能会涉及到，用ARM自己的开发工具去在Windows主机上面交叉编译了，ARM的相关开发工具有ADS，RVDS等等	
Uboot	叫做bootloader，类似Uboot的还有vivi，Redboot，（WinCE下的）Eboot等等	
Kernel==Linux Kernel	指的是Linux的内核，类似的，就是完全另外一套的WinCE环境了。	
Rootfs	rootfs内容：各种制作rootfs的工具，手工制作，用工具制作（比如用Buildroot生成的rootfs）rootfs格式：用哪种文件系统，此处假如用yaffs2	
Yaffs2	类似的，适用于Nand Flash的文件系统还有JFFS2，YAFFS，UBIFS 对应着，还要有制作yaffs2的rootfs的工具：mkysaffs2image	
真正开始涉及到嵌入式Linux底层的Nand Flash驱动的开发的话，又会接触到很多技术和框架，名词		
Linux驱动框架	不论你是实现哪个设备的驱动，都需要先搞懂本身Linux系统下的驱动框架和模型是什么样的 ->然后才能搞懂，框架帮你实现了哪些通用的功能，需要自己实现余下的设备相关的功能	
MTD	Linux中对于Nand Flash等设备所设计的框架，如果写Nand Flash驱动，就是基于该框架下去实现自己芯片的驱动 ->于此相对的，其他Linux中还有其他很多技术框架，比如针对声卡音频的ALSA等等。	
Nand Flash的技术细节：Block，Page，OOB，ECC，Wear Leveling	真正开始学习和了解Nand Flash驱动的话，会涉及到很多技术细节，比如Nand Flash的内部结构的Block，Page，OOB，数据需要用ECC算法校验保证数据无误（硬件还是软件ECC，ECC算法是BCH的话支持	相关内容： 【详解】如何编写Linux下

嵌入式Linux中Nand Flash驱动开发期间直接涉及到的	嵌入式Linux中Nand Flash驱动开发期间间接相关的，类似的，类比的	备注
	4位还是8位)，为了延长使用寿命而去实现 负载均衡Wear Leveling ，等等需要考虑很多	Nand Flash驱动⁵
Linux的使用和操作	在具体驱动开发期间，不可避免的要涉及到常见的Linux的使用，包括各种命令的用法，Linux系统基本原理和目录结构，都是避不开，要了解的。 ->比如常见的 ls,mkdir,rm,pwd,chmod,chown,ln 等等。	
Makefile	在开发期间，也会遇到用make命令，执行Makefile文件 ->嵌入式Linux中，往往都是用 Makefile 去管理项目的 ->为了了解项目代码的编译过程，也需要学习和了解make的机制 ->于此对应的，Windows系统下的很多软件，项目，代码的开发，往往都是集成了 IDE ，可以使很多人避开Makefile	
开发辅助工具： SourceInsight，BeyondCompare，虚拟机VirtualBox，Ubuntu等等	典型的开发期间，也往往会用到，用于查看Linux内核源码的 SourceInsight ，用于比较文件差别的 Beyond Compare ，用于搭建编译环境的虚拟机 VirtualBox+Ubuntu ，等等等等。	

总结：

- 搞嵌入式开发，更多的时候，需要先搞懂所涉及的技术所处的层次：
哪个行业，哪套硬件和软件平台，哪个软件平台下的什么框架，最终才是某个具体的技术。

只有这样，才能知道自己在干嘛，而不是完全一头雾水的埋头写代码。
- 正因为嵌入式开发涉及到的概念，软件，硬件，平台，比较多，需要的背景知识比较多，所以才会有那个结论
->总体来说，搞嵌入式软件开发，入门比较难

->至少比其他上层纯软件的入门要难不少

2.4. 嵌入式软件开发需要学习哪些知识

由于嵌入式本身是个复杂的概念，为了便于理解，此处用类比的手段去好好解释一下，嵌入式开发和从事建筑行业的关系，以便解释清楚，嵌入式学习，需要掌握哪些知识：

举个例子：就像，你从事的是建筑行业

有些是相对通用的基本知识和技能：（不论你是干体力活的各种工种的工人，还是大楼的整个架构的设计师）进入建筑工地，都要戴安全帽

不论你是干哪个工种的，比如瓦工，水电工，木工等等，那基本上都要具备的基本素质都是：

⁵ http://www.crifan.com/files/doc/docbook/linux_nand_driver/release/html/linux_nand_driver.html

身体素质本身要好，要能吃苦

但是，很明显，有些知识和技能，就属于领域相关的知识：

- 瓦工
你干瓦工的，要能看懂施工图，要会砌砖，砌各种类型的砖，等等；
- 木工
你搞木工，那锯子，刨子，锤子，斧头，那要用的熟，能玩出花样来，才能做出好的柜子，椅子，桌子啥的；

如此等等。但如果是你瓦工，则不需要关心木工需要了解的技能，反之亦然。

而搞嵌入式，这上面这几点上，有点类似于搞建筑行业：

- 有些知识，也是属于基本的，相对通用的知识，算是属于嵌入式行业内的基础知识和基本功
- 有些知识，是属于，具体干哪行，哪个领域，所要涉及到的领域相关知识

2.4.1. 嵌入式领域内相对通用的知识和技能

2.4.1.1. 嵌入式相对通用的知识：计算机编程语言

比如C语言，汇编语言，C++语言等等

2.4.1.1.1. 嵌入式相对通用语言：C语言

2.4.1.1.2. 嵌入式相对通用语言：汇编语言

2.4.1.1.3. 嵌入式相对通用语言：C++语言

2.4.1.2. 嵌入式相对通用的知识：芯片架构

硬件采用何种芯片，比如是ARM内核还是MIPS内核

是ARM内核的话，具体是哪种内核，比如ARM9TDMI还是Cortex-M3等等

不过，关于芯片架构的事情，要好好解释解释：

经常看到，嵌入式方面，介绍arm的架构

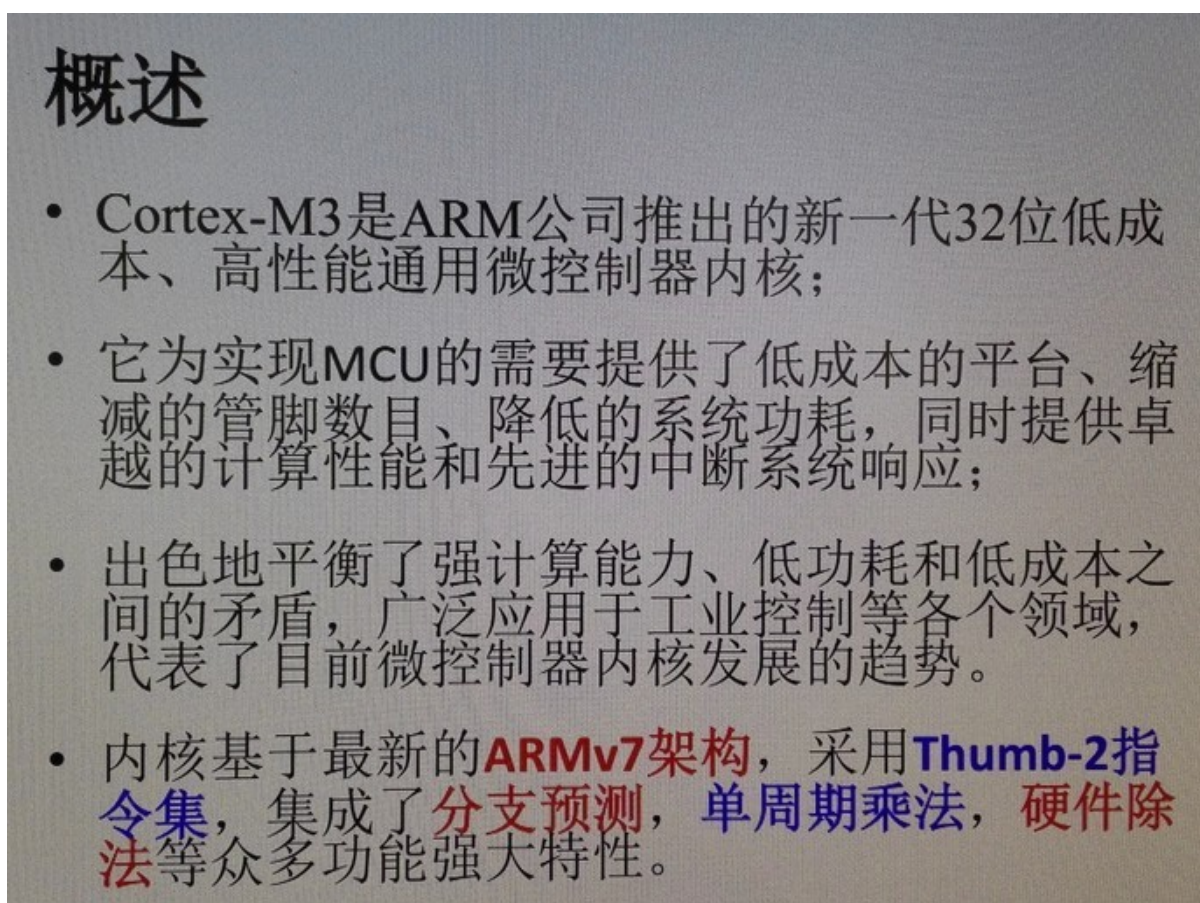
但是却又感觉学了也没啥用，和工作的内容也对不上号，纯粹就是在死读书，读死书

至少我最开始的感觉就是这样

现在其实也还是没有透彻理解，不过有一点点心得了，总结如下

比如以ARM Cortex-M3为例：

图 2.1. ARM的Cortex-M3架构概述



解释我所能理解的，架构中的内容，和你后续嵌入式Linux开发之间的关系：

- ARM架构中的AMBA总线
里面有AHB和APB

对应Linux中就是有对应的AMBA的概念的定义的，好像是在设备注册期间也有涉及到，比如probe的时候

- ARM核是否带MMU
决定了是用uclinux还是linux

- 采用哪款ARM内核
比如说Cortex-M3采用了ARMv7内核

则意味着你选择交叉编译器（甚至是自己用crosstool-ng去制作交叉编译器⁶）的时候内核选择是armv7。

- 指令集
比如说Cortex-M3是Thumb2的指令集

意味着，比如后期制作或用交叉编译器时，编译代码、生成什么样的汇编指令，都是要设置为thumb2

⁶ http://www.crifan.com/files/doc/docbook/crosstool_ng/release/html/crosstool_ng.html

如此，一点点地，把芯片架构是什么，以及不同的值的真正含义是什么，对后续开发有何影响，都说清楚了

这样才能说，明白了芯片的架构

2.4.1.3. 嵌入式相对通用的知识：嵌入式操作系统

相对通用的操作系统：嵌入式Linux，ucOS/II，VxWorks等等

2.4.1.3.1. 嵌入式常用操作系统：嵌入式Linux

此处的嵌入式Linux，是有别于普通桌面电脑所使用的普通桌面Linux系统。

2.4.1.3.2. 嵌入式常用操作系统：μC/OS-II

2.4.1.3.3. 嵌入式常用操作系统：ThreadX

2.4.1.3.4. 嵌入式常用操作系统：VxWorks

TODO：<http://www.windriver.com/evaluations/gpp-ve/>可以下载试用包，有空去试试。

2.4.1.3.5. 嵌入式常用操作系统：WinCE

2.4.1.3.6. Android不是传统意义上的嵌入式操作系统

此处需要专门解释一下，被很多人归类为嵌入式操作系统的Android，实际上不是传统意义上的嵌入式操作系统。

2.4.1.3.6.1. Android和嵌入式Linux的区别

典型的Android系统中，根据开发所用语言不同和开发环境不同，可以划分为：

- 上层应用：用java语言在Eclipse+ADT或者Android Studio的IDE中开发Android的app应用程序
- 嵌入式底层：用C，C++等语言实现底层的，以Linux为基础的，在Android框架下的，设备的驱动相关的开发，其中更多的是涉及到Linux驱动开发，Java的JNI，Android的驱动框架模型等等内容。

此时，对于传统意义上的，嵌入式软件开发，尤其是嵌入式Linux驱动开发，基本上是和，上述的Android开发中的，嵌入式底层开发中的Linux驱动开发，是基本上一样的。

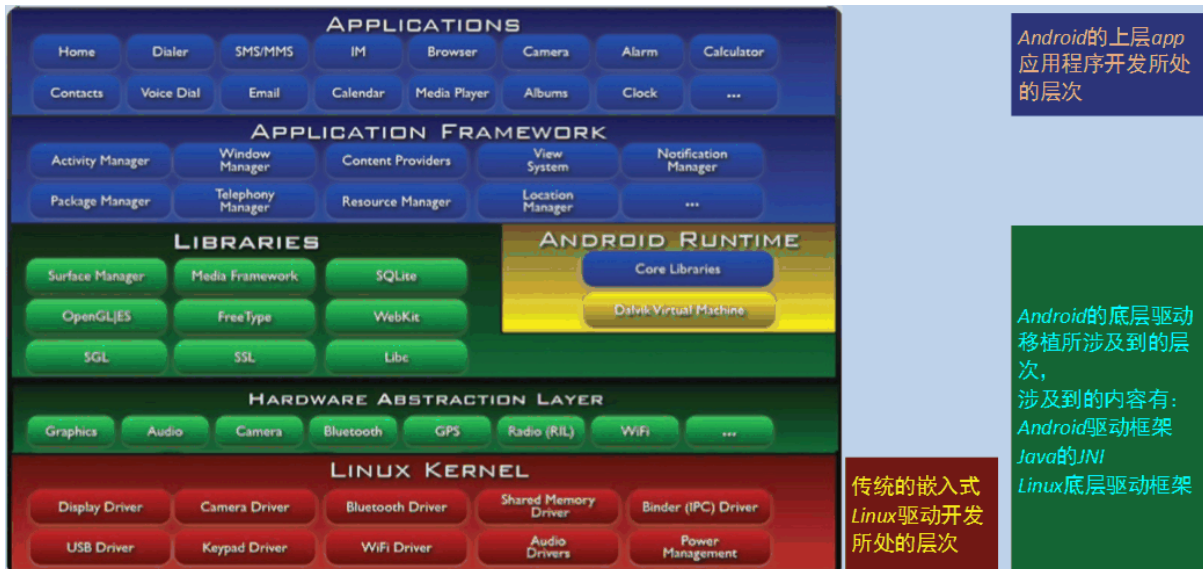
而Android系统中的其他内容，包括嵌入式底层的Java的JNI，Android驱动模型，以及上层应用的Java的Android的app开发，都是和传统的Linux的驱动开发，没什么关系，也是差别很大的，包括所使用的语言，以及所使用的开发环境。

即，Android的开发，基本上都是基于app那层去开发app的，都是用Java语言去写代码的。且底层有Android的运行时作为支持

而Linux的开发，对于Linux驱动开发，都是基于底层的内核去开发的，用的都是C语言，甚至偶尔还会涉及到汇编代码；对于Linux的应用层面的开发，都是Application Framework级别的，应用程序的开发，往往都是去写C语言代码的。

对应的传统Linux嵌入式驱动开发，以及Android中的app开发，以及Android中的驱动移植，所处于不同层次的关系，可以用下图表示：

图 2.2. 传统Linux和Android的不同的开发所处层次关系对比图



而对于Android的中间的Libraris，Android Runtime，以及Application Framework，很多时候，是各种（C，C++，Java等）不同的语言的混合体，包括中间的Android Runtime是利用到了Java的JNI，为Java和C或C++语言之间，提供一个接口，可以互相通讯。

而对于Android这套框架来说：

- 如果是你做Application，即做Android的应用，现在流行称其为app，的话则是去开发上层的应用->比如基于Android做一个收音机，做一个音乐播放器等等的应用
 - >都是在Android的框架之下（用Android相关的工具，比如Eclipse+ADT）去写Java代码去实现对应的功能
- 如果你是做Android底层嵌入式开发的话
 - >那就是在Linux内核框架下，去实现各种功能和模块
 - >典型的工作要属，你是一个做硬件的公司，做了块开发板，为了使得其支持Android系统，需要在开发板这个硬件的基础上，去为开发板中的各个硬件模块，比如Nand Flash存储芯片，SD卡芯片，网卡芯片等等，去写对应的模块的驱动
 - >此时主要就是在Linux框架下写C（或C++）代码去实现模块驱动
 - >当然有时候为了支持Android，还需要了解Android本身的框架，可能还需要加上一些JNI的代码支持

2.4.1.4. 嵌入式相对通用的知识：嵌入式开发环境搭建和使用

对于嵌入式开发来说，不论你是用何种嵌入式操作系统，都会涉及到，嵌入式环境的搭建和如何使用，如何去搞嵌入式开发。

比如，如果是嵌入式Linux的话，则要去搞懂常见的uboot+kernel+rootfs开发环境

2.4.1.4.1. 嵌入式开发环境：交叉编译器

相关内容：[交叉编译详解](#)⁷

相关内容：[crosstool-ng详解](#)⁸

2.4.1.4.2. 嵌入式开发环境：固件的烧写

不论是嵌入式Linux，还是WinCE等环境，都会涉及到，将程序交叉编译成可执行文件后，要下载到目标环境中。

这个动作，往往被叫做burn，烧录，烧写。

比如嵌入式Linux中，有很多种烧写方式：tftp，通过uboot，专门的烧写工具等等

如何用tftp下载文件

WinCE平台，也有特定的烧写工具

相关内容：[【详解】嵌入式开发中固件的烧录方式](#)⁹

2.4.1.4.3. 嵌入式开发环境：调试

此处的嵌入式环境中的调试，包括不同的调试方法和用不同的调试工具去调试。

2.4.1.4.3.1. 嵌入式开发调试方法和调试工具

- 软件：
 - printf
嵌入式Linux中经典的printf
 - 利用NFS实现快速测试
如何利用NFS实现快速的驱动的开发测试：方便的下载新编译好的驱动.ko文件并insmod去测试
- 硬件
 - JTAG
利用硬件的支持，比如JTAG实现调试
 - MultiICE
 - Jlink
TODO：把之前折腾过的Jlink的帖子都整理过来

⁷ http://www.crifan.com/files/doc/docbook/cross_compile/release/html/cross_compile.html

⁸ <http://www.crifan.com/files/doc/docbook/crosstool-ng/release/html/crosstool-ng.html>

⁹ http://www.crifan.com/files/doc/docbook/firmware_download/release/html/firmware_download.html

2.4.1.4.4. 嵌入式开发环境：嵌入式IDE

详见：[第 7.1 节 “嵌入式开发的IDE”](#)

换句话说，不论你在嵌入式行业内，具体做什么“工种”，以我之前所遇到，见到，听到的为例：

- 所在公司是做消费类数码产品研发，比如做Linux底层驱动开发
- 比如工业控制领域内的现场总线开发
- 汽车电子行业内的嵌入式开发

那么C语言，基本都是必备的，基本功，因此：不论你做相关的哪种领域的嵌入式开发，那么都要努力学好C语言



学好C语言是什么意思

学好C语言的意思是：不单单掌握C语言的语法，更要：

- 有良好的编程思想
 - 良好的编写高质量代码的编码习惯
 - 良好的处理问题的思路
 - 良好的代码风格
 - 了解写出高质量C语言代码需要注意哪些细节
- 等等。

另外，由于一些嵌入式操作系统，比如嵌入式Linux系统，ucOS/II（以及其他很多常见的嵌入式操作系统，比如VxWorks等）都是相对通用的。

所以，为了学好嵌入式，也要尽量找机会去多学习和了解这些系统的特点和使用方法，最好找机会多去实际动手练习。

2.4.2. 嵌入式领域内某领域内相关特定知识

同理：有些是Domain knowledge，即领域相关的知识

关于这部分，之前见过的，一个老外写的，不错的嵌入式的教程：

[第 16.1.1 节 “Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers”](#)

中所给出的例子，就是个很好的，domain knowledge的例子：

如果打算从事嵌入式机顶盒方面的嵌入式开发的话，那么往往又要涉及到

DVB，MHP，EGB，和其他的协议，架构，软件库，等等等等

所以说对于不同的领域，其所相关的东西，光是新名词就一堆一堆的

更别说，每个名字后面都对应着一堆规范说明，都N多页N多页

往往还都是英文的

此时你所面对这些技术（名字和规范说明）的心情

有点类似于：你是干木匠的，然后让你去把瓦工干的活，不仅仅要把名词搞懂了，还要搞懂如何实现的，并且还要把活干得好

这时候，往往就是：需要你有足够的背景知识：对于该领域所涉及的到的技术，即使没实际经验，也要最好有个概念

加上良好的学习能力：有了其他领域的背景知识，以便于稍微帮助你学习新领域的知识

以及良好的悟性：很多时候，学习新东西，出了基础和经验，剩下就是悟性了

加上：足够的耐心，毅力

以及实际动手能力：把所理解的内容，能转化有有效的代码，

最终：

真正去用高质量的代码，实现软件的功能

由此，算是：

在建筑领域内混的一个木工，靠着基础知识，努力学习瓦工方面的知识，加上良好的悟性和动手能力，

最终把瓦工的活干了，而且干得还不错

当然老板最希望你是：干的比原来的瓦工干的还好

如此，正所谓：嵌入式开发中

最先要，也最重要的，是把基础知识学好了，基础打牢了；

然后是，针对所要学习和工作的相关领域的知识，去用最短的时间，最大的学习热情，加上悟性，和学习方法，最终掌握该领域的知识，并且实际动手去锻炼，最终实现目标：

用高质量的代码实现嵌入式领域内的相关的软件功能

2.4.2.1. 嵌入式领域相关知识之消费类数码

如果做消费类数码方面的嵌入式研发，往往会涉及到嵌入式开发板

以某个SoC芯片为核心的开发板，然后上面有各种接口

为各种接口去开发不同平台下，比如嵌入式Linux，ucOS/II等等，的驱动，或者上层应用层的软件。

2.4.2.2. 嵌入式领域相关知识之工业自动化控制

其实严格区分的工业自动化控制的话，算是两个方向，工业控制和工业自动化，这两个领域，也是不同，各有侧重点的。

但是此处也还是算是工业方面的领域，所以就放在一起说了。

可能会涉及的技术：现场总线

2.4.2.3. 嵌入式领域相关知识之汽车电子

汽车电子领域内，在协议和接口方面，相对用的最多的是CAN总线协议，也常会涉及到MOST (Media Oriented Systems Transport) 总线

相对来说其他行业就很少用到这类技术。

除了汽车电子领域相关的技术，往往还会涉及到相关的标准，比如SPICE等等。

2.4.2.4. 嵌入式领域相关知识之医疗器械

2.4.2.5. 嵌入式领域相关知识之航空航天

这个领域不熟悉。但是了解过一些相关知识。比如美国的好奇号火星探测器，里面用的操作系统就是VxWorks。

2.5. 嵌入式系统中从底层硬件到上层软件之间的逻辑层次和映射关系

嵌入式教程，关于嵌入式硬件，尤其是嵌入式开发，尤其是上面的CPU：

从底层硬件实体，到上层软件层面的抽象，到底是如何映射的

抽空写教程，详细介绍

此处，先简要介绍一下：

1. 找个开发板，真正的截图，能清晰的显示，开发板上面的CPU是什么芯片
2. 找个对应或类似CPU芯片的，物理排版图，BOM图，能否显示出，片内的RAM，CPU核心，外围设备的，那种图
3. 然后说明，CPU部分，片内RAM等部分，对应的逻辑抽象，分别叫做什么
4. 然后再去对照着，对应芯片的datasheet，找到对应的芯片参数表，对应着CPU core，多少Hz，什么核心的
比如ARM核的片内RAM，比如128K，等等内容
5. 最后，其实，CPU内部的架构，再详细剖析一下
包括CPU内部是对应的ALU，寄存器，等等
6. 然后再去提到，相关的CPU内部指令执行，分别3部，5步，之类的，比如取指，译指，执行，写回等等
包括内部寄存器，分为几种类型，比如arm的shadow寄存器，在某些FIP等特殊情况下，无需保存现场，而直接实现中断时快速跳转等功能
7. 然后ALU的话，内部原理是如何实现的，主要是硬件的门电路gate等实现的
8. 对于门电路，再往下分，最后就是对应的与非门，逻辑与，逻辑或等等门电路了。
9. 而对于门电路，物理硬件的实现，好像也是有多种可能的。
具体需要再去学习[硬件电路基础知识](http://www.play-hookey.com/)¹⁰中提到的那个教程：<http://www.play-hookey.com/>，才能具体解释清楚的。

¹⁰ http://www.crifan.com/files/doc/docbook/hardware_basic/release/htmls/

10. 再往下，好像就是：晶体二极管，三极管了。
都是这些晶体管，慢慢的组合，构成不同的逻辑电路的，包括什么D寄存器，K寄存器（好像是）之类的。
- 11.

第 3 章 嵌入式软件开发的基本名词和概念

嵌入式领域内，有很多常见的基本概念和名词，对于初学者来说，往往是听都没听过这些词，更别说明白其含义了。

此处就整理一下，我所知道的一些嵌入式领域内的常见的概念和名词的含义。

3.1. 开发套件==开发组件==develop suite==develop toolkit=StartKit=DevBoard

比如：[具备USB，CAN和以太网功能的PIC32微控制器系列](http://ww1.microchip.com/downloads/en/DeviceDoc/39904k_tw.pdf)¹中的：

图 3.1. PIC32入门套件



使用 **PIC³²** 微控制器進行開發

Microchip是唯一提供8位元、16位元和32位元全產品線微控制器的晶片供應商，這三種微控制器由一致的開發環境支援。MPLAB® IDE為免費提供，且易於使用。

PIC32入門工具套件

使用高整合度的PIC32入門工具套件、入門指南和PIC32入門工具套件電路板（具有可方便與PC連接的USB介面）可輕鬆入門，該套件包括：

- MPLAB IDE和MPLAB C32 C編譯器†
- 整合了燒錄器和除錯器的PIC32入門工具套件電路板
- 範例程式、說明文件、入門教學指南以及專案範例，選購I/O擴充電路板允許PICtail™ Plus子卡訊號突破及連接

†Lite版無程式大小的限制，且具有所有最佳化功能。一些最佳化功能將於60日後被停用。

¹ http://ww1.microchip.com/downloads/en/DeviceDoc/39904k_tw.pdf

图 3.2. PIC32开发工具

PIC32開發工具

選擇平臺：

入門工具套件平臺			或	Explorer 16平臺	
<p>PIC32 入門工具套件 (DM320001)</p>	<p>PIC32 USB 入門工具套件II (DM320003-2)</p>	<p>PIC32乙太網 入門工具套件 (DM320004)</p>		<p>Explorer 16 開發電路板 (DM240001)</p>	<p>PIC32接插模組 (MA320001) (MA320002) (MA320003)</p>
選購				+	
<p>PIC32 I/O 擴充電路板 (DM320002)</p>				及	
<p>多媒體 擴充電路板 (DM320005)</p>					
				<p>MPLAB® ICD 3 線上除錯器 (DV164035)</p>	<p>MPLAB® REAL ICE™ 線上模擬系統 (DV244005)</p>
				或	
<p>兩種開發平臺通用的PICtail™電路板</p>					
<p>配備3.2英吋顯示幕 的圖形子板套件 (AC164127-3)</p>	<p>ECAN/LIN子板 (AC164130)</p>	<p>MRF24WB0MA Wi-Fi子板 (AC164136-4)</p>	<p>MRF24J40MA 2.4 GHz RF卡 (AC164134)</p>		
				...還有更多!	

就是：

开发套件

= 硬件+软件

= 硬件开发板 + 集成的IDE 或 单独提供的编译器，示例代码等内容

其他的，各种平台的，往往也会提供对应的，硬件（开发板，模块等等）+软件（软件源码，示例代码，驱动等等）的组合，叫做对应的Toolkit，即工具套件

成套提供，以套餐方式的提供

类似的词还有：StartKit，DevBoard

也都是，方便用户进行快速上手开发的开发板

目的是：

方便其用户，作为嵌入式开发的你，尽快的上手，能够跑通这个平台，能用于构建你的嵌入式系统，能够进行接下来的，你所需要的相关的软硬件方面的开发。

其背景是：

如果芯片厂家，直接把一款芯片扔给你，给你用；

不仅是新手，即使是老手，嵌入式有经验的，也要相对很耗精力，才能搞懂如何初始化，如何让其运行起来，即跑起来，需要花不少的精力

因为嵌入式领域，芯片不同，细节会千差万别，所需要了解的细节也很多。

所以，才有：

卖芯片的厂家，除了设计，量产，销售芯片给作为客户当你之外，

往往还有对应的所谓的开发套件，

即硬件（芯片或开发板）加上对应的软件（驱动，示例代码）

然后让芯片尽快的，正常的跑起来

之后才是作为嵌入式开发者的你，继续后续折腾，实现对应的你自己的功能。

举例：

- [MSP430微控制器 | 16 位微控制器 | 16位MCU – 概述](#)²中就有"MSP430 套件与评估模块"，指的就是可以去用于开发工作的一堆成套的相关工具。
- <http://www.microchip.com/pagehandler/en-us/family/32bit/>中就有对应的StartKit + Development Board的组合。

3.2. Part Number == 模块编号 == 产品编号

不仅各种硬件模块有Part Number

软件（或库，组件等）也有

比如：[MPLAB Integrated Development Environment](#)³的"Part Number: SW007002"表示：

Part Number == 作为该软件或硬件的标示 == 就像人的身份证

-> 你想要买人家的产品（某软件或硬件），只需要告诉人家对应的Part Number即可

-> 人家就知道你要买的，具体是哪个东西了。

3.3. fabless == 无厂半导体公司 == 无工厂的IC设计公司

简单说就是：专门设计芯片的，但是没有自己的制造工厂。

是自己设计出芯片后，找其他的，专门制造生产芯片的公司去生产，对应的生产芯片的公司，也是下面将要介绍的，Foundry。

比如：[威盛电子 - 维基百科，自由的百科全书](#)⁴就是一个fabless

全球范围内的fabless，非常非常多。

² http://www.ti.com.cn/lscs/ti_zh/microcontroller/16-bit_msp430/overview.page

³ http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=cn019469

⁴ <http://zh.wikipedia.org/wiki/%E5%A8%81%E7%9B%9B%E9%9B%BB%E5%AD%90>

维基百科有些统计，可供参考：

[Category:Fabless semiconductor companies - Wikipedia, the free encyclopedia](http://en.wikipedia.org/wiki/Category:Fabless_semiconductor_companies)⁵

[Fabless manufacturing - Wikipedia, the free encyclopedia](http://en.wikipedia.org/wiki/Fabless_manufacturing)⁶

其中里面比较典型的一些企业有：

AMD，ARM，ATI，Broadcom#，Marvell，MediaTek，Qualcomm，Realtek，Rockchip，Silicon Labs，Tensilica，VIA，Xilinx

3.4. foundry == 晶圆厂 == 晶圆代工厂 == 代工厂

简单说就是：专门生产芯片的公司

详细的说：专业的叫法叫做，专业集成电路制造服务类的企业

详见：[Foundry - Wikipedia, the free encyclopedia](http://en.wikipedia.org/wiki/Foundry)⁷

世界上最大的、最有名的要数，台积电，了。

详见：

[台湾集成电路制造公司 - 维基百科，自由的百科全书](http://zh.wikipedia.org/wiki/台湾集成电路制造公司)⁸

另外其他还有一些小的晶圆厂，比如AMS==奥地利微电子，也有自己的晶圆厂。

详见：[AMS公司投2500万欧元建三维堆叠集成电路制造能力 全球军情 全球防务频道 新闻中心](http://www.cnn.com/2013/09/06/45621.shtml)⁹，[奥地利计划购买200毫米晶圆代工厂以加强晶圆制造能力](http://www.cnn.com/2013/09/06/45621.shtml)¹⁰

其他相关资料：

[混合集成电路的Fabless之路-《电子元器件应用》2003年02期-中国知网](http://www.cnki.net/magazine/article/YQJY200302000.htm)¹¹

3.5. IP Core == IP核 == 软核

谈到fabless，往往就会涉及到一个概念：IP Core。

因为，这类fabless，就只是设计芯片，对应的产出就是对应的IP Core。

举个类似的例子：fabless设计出IP Core或者叫做芯片，拿去给Foundry去生产

就类似于，建筑设计院设计出一栋大楼的图纸，拿去给建筑商去建造出来，是一个道理。

fabless有设计能力，foundry有生产能力

3.6. 量产

此处的量产，有两种含义：

⁵ http://en.wikipedia.org/wiki/Category:Fabless_semiconductor_companies

⁶ http://en.wikipedia.org/wiki/Fabless_manufacturing

⁷ <http://en.wikipedia.org/wiki/Foundry>

⁸ <http://zh.wikipedia.org/wiki/%E5%8F%B0%E7%81%A3%E7%A9%8D%E9%AB%94%E9%9B%BB%E8%B7%AF%E8%A3%BD%E9%80%A0%E5%85%AC%E5%8F%B8>

⁹ <http://www.cnn.com/2013/09/06/45621.shtml>

¹⁰ <http://express.cetin.net.cn:8080/cetin2/servlet/cetin/action/HtmlDocumentAction;jsessionid=AF2B8EDC2B2181874CCB65508F6F20FB?baseid=108&docno=152977>

¹¹ <http://mall.cnki.net/magazine/article/YQJY200302000.htm>

3.6.1. 批量生产芯片

一般指的是：某芯片设计公司fabless设计出来了芯片，然后拿去让代工厂foundry生产真正的，物理上的，你能看得见摸得着的，芯片。

3.6.2. 重新烧录U盘的固件

烧录固件到已有的嵌入式设备（尤其是U盘）上

不过，量产还有另外一种意思，就是已经有了对应的可以使用的某设备。

尤其是U盘，然后重新用烧录工具，将另外一个新版本的程序，一般叫做固件，烧录进去

实现功能的改进，比如将原先的U盘，弄成CDROM+U盘的形式。

相关细节可参考我之前的一些折腾：

[【教你从不懂U盘量产到量产成功 -> 宇瞻AH320微笑碟8G详细量产过程】](#)¹² [【记录】Transcend创见16GB USB3.0 U盘 第二次重新量产为USB+CDROM](#)¹³ [【记录】第四次折腾U盘量产以制作Win7启动U盘](#)¹⁴

3.7. VFP 浮点处理器

用于专门处理浮点数据的处理器。

嵌入式领域内，谈到浮点处理时，往往会提到这个：VFP，浮点处理器

比如：

[http://www.freescale.com/zh-Hans/webapp/sps/site/prod_summary.jsp?code=i.MX6D#prettyPhoto\[iframes\]/0/](http://www.freescale.com/zh-Hans/webapp/sps/site/prod_summary.jsp?code=i.MX6D#prettyPhoto[iframes]/0/)

中提到的：ARM的NEON的VFP

3.8. 代码空间占用大小：footprint

footprint的英文本意是：脚印

放在嵌入式领域内，指的是：软件类的东西（比如一个操作系统，某个软件，某个协议栈等等），所占用的存储空间的大小。

可以理解为：嵌入式系统中，把软件，看成一个人，此人在RAM或ROM等存储空间上，所踩出来的脚印。

与此相关的，在介绍某软件，组件的优点时，说的更多的是“small footprint”这个词。

随便举个例子：

[RTOS – Real-Time Operating Systems for Embedded Development, Real Time System By Express Logic](#)¹⁵

¹² http://www.crifan.com/teach_you_to_understand_u_disk_from_mass_production_to_mass_production_success_gt_apacer_ah320_smile_dish_8g_detailed_production_process/

¹³ http://www.crifan.com/sencond_time_mass_production_transcend_16gb_usb_3-0_disk_to_usb_cdrom/

¹⁴ http://www.crifan.com/again_make_bootable_win7_usb_disk/

¹⁵ <http://rtos.com/products/threadx/>

中，ThreadX这个嵌入式操作系统，在解释其优势的时候，其中一条就是：“Small Footprint”
指的就是：

此ThreadX嵌入式操作系统，所需要的RAM或ROM等存储空间相对较小



嵌入式的存储空间往往都有限所以占用空间小就省空间就是优势

此处有个背景知识：

嵌入式系统中，往往是RAM或ROM等存储空间很有限，所以需要运行在其中的嵌入式操作系统，以及程序，要尽量少的占用这些空间，

这所谓，放得下，跑得起来，才行。所以，占用相对较小的存储空间，这点对于嵌入式系统来说，就是个很大的优势了。

所以ThreadX在介绍其自己的特点时，才拿Small footprint来说事的。

第 4 章 嵌入式开发环境的基本构成

嵌入式开发，有很多主流的操作系统，对应着各自不同的开发环境。

常见的有：

- 嵌入式Linux
- WinCE

下面分别来解释：

4.1. 嵌入式Linux开发环境组成

先要搞懂嵌入式Linux开发的常见的，典型的环境的基本组成：uboot+kernel+rootfs

- bootloader是啥
为何要bootloader
- 什么是kernel
- rootfs
为何非要有个rootfs。
- 以及，常见的，典型的，嵌入式开发流程是什么样的
为何非要这么开发，换个方式行不行，有没有更好的方式。
- 主机Host
- 一套交叉工具链，也叫交叉编译器
大了说，是一套编译开发环境
- uboot
- kernel
- rootfs
- 烧录工具
- 开发板

抽空再按照，硬件有哪些，软件需要哪些去划分，可能更容易理解。

关于嵌入式Linux开发环境的解释，详见：

[嵌入式Linux软件开发](http://www.crifan.com/files/doc/docbook/embedded_linux_dev/release/html/embedded_linux_dev.html) ¹

4.2. WinCE开发环境组成

¹ http://www.crifan.com/files/doc/docbook/embedded_linux_dev/release/html/embedded_linux_dev.html

第 5 章 嵌入式开发之交叉编译和交叉编译器

5.1. 什么是交叉编译

详见：

[交叉编译简介](#) ¹

5.2. 如何获得交叉编译器

详见：

[如何得到交叉编译器](#) ²

5.3. 如何使用交叉编译器

简单说就是：直接把交叉工具链所在的路径，放到环境变量PATH中，使得可以在命令行环境中正常调用即可

5.4. 嵌入式编译相关知识

嵌入式编译，除了会涉及编译原理本身的东西之外，还会很多不同体系架构方面特定的内容。

比如，ARM的scatter file的概念。

TODO：详见：<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0474c/CHDDDBGE.html>

¹ http://www.crifan.com/files/doc/docbook/cross_compile/release/html/cross_compile.html#crosscompile

² http://www.crifan.com/files/doc/docbook/cross_compile/release/html/cross_compile.html#how_to_get_crosscompiler

第 6 章 嵌入式开发之烧录

6.1. 什么是烧录

烧录，英文叫做burn。

也常被叫做下载，download

指的是，把嵌入式开发期间，编译好的文件，往往又被称为固件，firmware。从你的当前搞嵌入式开发的PC机，也叫做主机端host，传输到你的目标平台，即开发板，真正用来存储和运行你的固件程序的地方。这个过程，就叫做烧录，下载。

6.2. 如何进行烧录

6.2.1. 常见的烧录所用工具

下面是一些常见的，用的比较广泛的烧录工具：

- JLink
Keil公司在[Keil Flash Utility](http://www.keil.com/flash/utilities.asp)¹就列举了很多，其中就有这个JLink
之前也折腾过很多：[使用破解版的JLink实现对开发板上的外部Nor Flash的烧写 + JLink V4.08k 下载地址](http://www.crifan.com/use_the_cracked_version_of_jlink_to_achieve_the_development_of_the_external_board_of_nor_flash_programming_jlink_v408k_download/)²
- 串口
- USB接口类的工具
比如e8a的开发板，自带USB接口的专门用于烧录e8a板子的特殊硬件

其他还有一些小众的烧录工具，很多都是针对自己的开发板去烧录程序的工具：

- LPC2000 Flash Utility
[【记录】安装Philips Flash Utility的Laubch LPC210x ISP V2.2.3.zip中的Philips Flash Utility Installation.exe](http://www.crifan.com/record_philips_flash_utility_lpc2000_isp/)³

6.2.2. 常见的烧录方式

其中比较常见的做法是：

- 用JLink等工具，烧录uboot
- 在uboot中，通过串口，网卡等烧录kernel和rootfs

详见：

¹ <http://www.keil.com/flash/utilities.asp>

² http://www.crifan.com/use_the_cracked_version_of_jlink_to_achieve_the_development_of_the_external_board_of_nor_flash_programming_jlink_v408k_download/

³ http://www.crifan.com/record_philips_flash_utility_lpc2000_isp/

[【详解】嵌入式开发中固件的烧录方式](#)⁴

⁴ http://www.crifan.com/files/doc/docbook/firmware_download/release/html/firmware_download.html

第 7 章 嵌入式开发之开发工具

7.1. 嵌入式开发的IDE

TODO :

[【整理】各种常见的嵌入式IDE](#) ¹

[【整理】嵌入式IDE : Renesas的HEW\(High-performance Embedded Workshop\)](#) ²

[【整理】嵌入式IDE : IAR Embedded Workbench IDE简介](#) ³

嵌入式开发领域内也有很多常见的，相对通用的IDE：

[【整理】各种常见的嵌入式IDE | 在路上](#) ⁴

7.1.1. 嵌入式IDE : Keil

7.1.2. 嵌入式IDE : ARM相关的ADS , RVDS等

ARM自家有一些常见的工具，之前叫做ADS，后来升级到RVDS，后来还收购了Keil的RealView MDK

ADS最常用的版本就是ADS 1.2

7.1.3. 嵌入式IDE : IAR

[【整理】嵌入式IDE : IAR Embedded Workbench IDE简介 | 在路上](#) ⁵

7.2. 协议分析工具

协议分析工具很方便很强大 -> 有机会还是多多利用协议分析工具 -> 提高工作效率

比如之前的，就见过USB协议分析工具，即[USB基础知识概论](#) ⁶中[其他一些USB测试和协议分析等软件](#)提到的：

“Ellisys的USB Explorer 260硬件，加上对应的USB软件Ellisys USB Analysis Software，实现USB数据抓包和分析”和：“Catalyst Enterprises公司的硬件，加上对应的软件SBAE USB，实现USB数据捕获和分析”

SBAE的效果可参见：[举例详解USB的枚举过程](#) ⁸

现在又在：<http://www.zlgmcu.com/tools/LogicAnalyser/LA1032/LA1032-SD.pdf>中看到有个原始数据：

¹ http://www.crifan.com/summary_common_embedded_ide/

² http://www.crifan.com/intro_embedded_ide_renesas_hew_high_performance_embedded_workshop/

³ http://www.crifan.com/embedded_ide_iar_embedded_workbench_intro/

⁴ http://www.crifan.com/summary_common_embedded_ide/

⁵ http://www.crifan.com/embedded_ide_iar_embedded_workbench_intro/

⁶ http://www.crifan.com/files/doc/docbook/usb_basic/release/html/usb_basic.html

⁷ http://www.crifan.com/files/doc/docbook/usb_basic/release/html/usb_basic.html#usb_analysis_tool

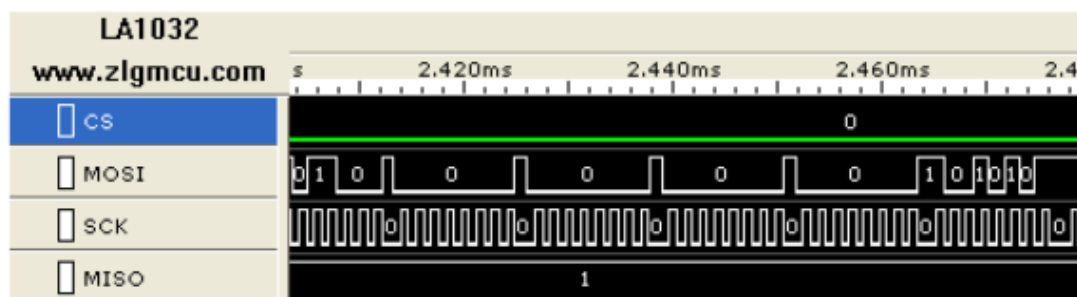
⁸ http://www.crifan.com/files/doc/docbook/usb_basic/release/html/usb_basic.html#usb_emulation_example

图 7.1. SD原始时序图

1.4 解码的效果图

通过采集 SPI 总线接口对 SD 卡操作的时序图，并对其进行时序分析解码的效果。以下为一个写单块、设备响应参数有错误的操作。

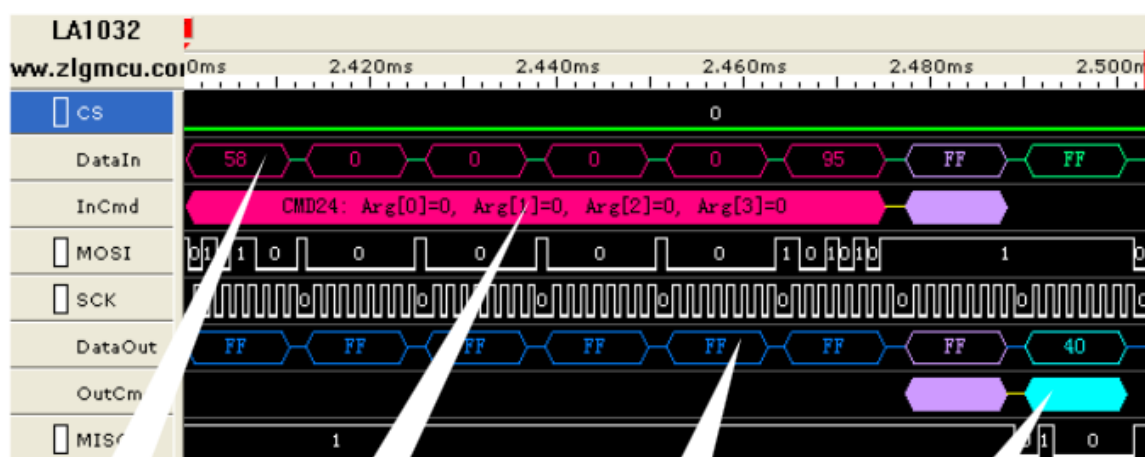
1、原始时序图。



用 (SD) 协议分析工具分析后的结果：

图 7.2. 结果解码后的SD时序效果图

2、通过解码后的效果图



输出数据字节格式解码

输出数据命令格式解码

输入数据字节格式解码

输入数据命令格式解码

很明显：

在SD协议中对应的物理的信号所表示的含义，不用你很笨的手动一点点的去分析了，因为人家的分析工具直接帮你分析好了。相对来说，算是极大的提高了工作效率。

所以，有机会，有条件（毕竟有些协议没有这么好的分析工具，或者即使有未必是免费，未必你能得到该工具）的话，还是好好利用工具，提高做事的效率。

7.3. 交叉编译器/工具链

市面上有很多不同公司提供的，各种类型的编译器和其他相关开发工具。

比如⁹[raisonance](http://www.raisonance.com/)提供的：C编译器，编程工具，调试工具，评估板，支持来自不同厂商，包括NXP，STMicroelectronics，TI，Silicon Laboratories等，基于ARM，8051，STM8，ST7，PowerPC，CoolRISC为微处理器内核的快速开发。

7.4. 嵌入式开发的调试工具

嵌入式调试工具¹⁰[TDS560 Plus仿真器](http://www.armkits.com/product/tds560.asp)，<http://www.armkits.com/product/tds560.asp>，用于调试TI的DSP（也支持ARM）的开发。

⁹ <http://www.raisonance.com/>

¹⁰ <http://www.timll.com/chinese/product/TDS560.asp>

第 8 章 嵌入式开发之系统设计

8.1. 嵌入式系统设计之硬件选型

硬件选型时需要根据所需的硬件资源情况去选择合适芯片

比如：

图 8.1. PIC32系列产品中有各种不同资源

PIC32產品系列																														
元件	快閃KB +啟動快閃	SRAM KB	接腳數	MHz	SPI	IC™	UART	DMA通道 一般/專用	USB	10/100乙太網路	CAN 2.0b	IC/OC/PWM	10位元ADC 1 Msps	類比比較器	計時器 16b/32b	RTCC	並列式主控埠	JTAG程式設計、 除錯、掃描												
PIC32MX320F032H	32 + 12	8	64	40	2	2	2	0/0	N	N	N	5/5/5	16通道	2	5/1	1	Y	Y												
PIC32MX320F064H	64 + 12	16	64	40																										
PIC32MX320F064H			80																											
PIC32MX320F128H	128 + 12	16	64	80																										
PIC32MX320F128L			100																											
PIC32MX340F128H			64																											
PIC32MX340F128L			100																											
PIC32MX340F256H	256 + 12	32	64	80																										
PIC32MX360F256L			100																											
PIC32MX340F512H			64																											
PIC32MX360F512L			100																											
PIC32MX420F032H	32 + 12	8	64	40	1	2	2	0/2	Y	N	N	5/5/5	16通道	2	5/1	1	Y	Y												
PIC32MX440F128H	128 + 12	32	64	80	1																									
PIC32MX440F128L			100	2																										
PIC32MX440F256H	256 + 12	32	64	80	1																									
PIC32MX460F256L			100	2																										
PIC32MX440F512H			64	80	1																									
PIC32MX460F512L			100	2																										
PIC32MX534F064H	64+12	16	64	80	3														4	4/4	Y	N	1	5/5/5	16通道	2	5/1	1	Y	Y
PIC32MX534F064L					100														4											
PIC32MX564F064H			32		64	80	3	4																						
PIC32MX564F064L					100	4	5																							
PIC32MX564F128H	128+12	32	64	80	3	4	6	Y	N	1	5/5/5	16通道	2	5/1	1	Y	Y													
PIC32MX564F128L			100		4	5																								
PIC32MX575F256H	256 + 12	64	64	80	3	4												8/4												
PIC32MX575F256L			100		4	5																								
PIC32MX575F512H			64		80	3													4											
PIC32MX575F512L			100		4	5																								

就列出来了，不同芯片，所具体的不同资源：

是否有所需要的功能模块

以及部分功能模块的数量：根据需要选择满足你的需求的芯片

比如：

需要性能相对高的，所要CPU的频率相对高 -> 才能流畅的跑很多复杂的，资源消耗多的应用

其他比如，SPI等接口的个数，决定了，你如果外接某些设备，比如SPI的Wifi芯片，至少占用1个SPI接口，

如果还有其他的模块需要SPI接口，则需要考虑SPI接口总数是否够用

类似的，还有I2C，UART等其他接口，都需要根据自己的实际需求，决定所需要的接口的个数，是否够用

另外还有类似的DMA：

有些Nand Flash，SD卡等模块，为了提高数据传输速度，则可以用DMA来实现

所以，如果系统中Nand Flash，SD卡，都需要DMA，那么此时至少需要2个DMA通道。所以，需要选择DMA通道个数大于2的芯片。

其中，关于DMA，不了解的可参考：[详解ARM的AMBA设备中的DMA设备PL08X的Linux驱动](http://www.crifan.com/files/doc/docbook/dma_pl08x_analysis/release/html/dma_pl08x_analysis.html)¹

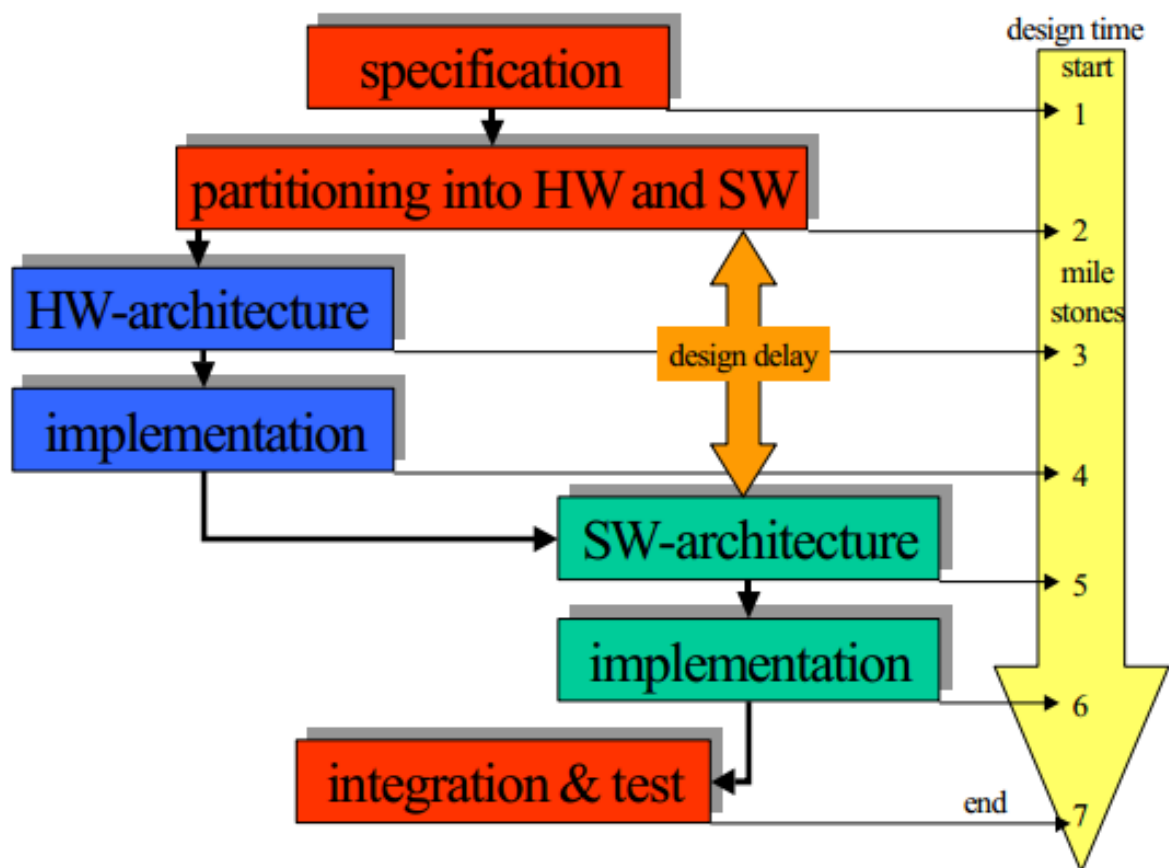
8.2. 嵌入式系统设计流程

说实话，自己没有真正独立的从无到有的设计一个嵌入式系统，所以整理出来的内容，也是参考别人写的内容。

截至目前，尤其是也没发现，有解释的够详细的，暂时看到了点资料，整理过来，仅供参考：

http://www.uml.org.cn/embedded/pdf/embed_design.pdf

图 8.2. 嵌入式开发流程图



¹ http://www.crifan.com/files/doc/docbook/dma_pl08x_analysis/release/html/dma_pl08x_analysis.html

第 9 章 嵌入式开发之嵌入式CPU

9.1. 什么是嵌入式CPU

[【整理】嵌入式CPU, MCU, SoC, DSP之间的区别和联系 | 在路上](#)¹

[【整理】用汽车去类比详解嵌入式的SoC | 在路上](#)²

[【整理】常见的嵌入式CPU, MCU, SoC, CPU Core的公司, 品牌, 厂家 | 在路上](#)³

[【整理】嵌入式软硬件概览 v2011-10-13 | 在路上](#)⁴

9.2. 嵌入式CPU核心参数

嵌入式CPU/SoC/MCU/DSP等核心参数：主要还是性能：MHz，存储：RAM，外设多少等等

比如，除了之前[【整理】嵌入式开发经验和心得总结](#)⁵

中提到的PIC32的那个图，现在还有个：

http://www.analog.com/zh/processors-dsp/sharc/products/selection-tables/sharc_Selection_Table/resources/fca.html

的产品型号参数表，也可以提现出这点，主要参数还是性能：

- CPU频率：多少MHz
- 存储：片上RAM，内存多少
- 外设：外设的类型和个数

9.3. 嵌入式CPU的应用领域

嵌入式的CPU的应用领域，其实和本身嵌入式的领域的划分：[第 1.1 节 “嵌入式行业和领域”](#)是差不多的。

下面总结一下，不同的嵌入式CPU的主要特点，应用领域：

- SHARC处理器
TODO：[SHARC处理器的应用](#)⁶

特点：

在浮点DSP市场占据主导地位

拥有出色的内核和存储器性能

以及优异的I/O吞吐能力

¹ http://www.crifan.com/summary_embedded_cpu_dsp_mcu_soc_relation_comparation/

² http://www.crifan.com/explain_soc_by_analogy_car/

³ http://www.crifan.com/embedded_common_used_soc_mcu_ip_core_company_brand/

⁴ http://www.crifan.com/order_embedded_software_and_hardware_overview_v2011-10-13/

⁵ http://www.crifan.com/summary_embedded_development_experience_and_note/

⁶ <http://www.analog.com/zh/processors-dsp/sharc/products/index.html>

主要应用领域：

- 音频
- 安全与监控
- 汽车电子
- 测试测量与控制
- 过程控制
- ARM
ARM本身应用极其广泛。此处只总结部分的典型应用。

TODO：[ARM内核的应用](#)⁷

ARM946E-S 处理器可用于众多先进的数字产品：

- 消费品：智能手机、PDA、机顶盒、PMP、电子玩具、数码相机、数码摄像机等
- 联网：无线局域网、802.11、蓝牙、Firewire、SCSI、2.5G/3G基带等
- 汽车：电力火车、ABS、车身系统、导航、信息娱乐等
- 嵌入式：USB 控制器、蓝牙控制器、医用扫描仪等
- 存储：硬盘控制器、固态驱动器等

ARM还有Corte系列的内核, 比如常用的Cortex M3/M4

- Renesas
Renesas的芯片在汽车电子领域中应用较广
比如某汽车内部用到的芯片就有Renesas V850/K0R
- Freescale
Freescale的芯片在汽车电子领域中应用较广

⁷ <http://www.arm.com/zh/products/processors/classic/arm9/arm946.php>

第 10 章 嵌入式软件开发的外围设备

TODO :

[【整理】嵌入式系统的各种常见外设](#)¹

10.1. 嵌入式外设之LCD显示屏

一些LCD控制器，可以参考：

[Segger - emWin - List of supported LCD controllers](#)²

10.2. 嵌入式外设之图像传感器

图像传感器，Image Sensor，之前所接触到的，都只是放在CPU内部的。

后来才看到，还有外部接口接Image Sensor的：

<http://www.armkits.com/download/at91sam9g45.pdf>中的：ITU-R BT. 601/656 Image Sensor Interface

是专门的接口，用于连接外部图像传感器的。

¹ http://www.crifan.com/summary_embedded_system_various_peripherals

² http://www.directinsight.co.uk/products/segger/emwingsc_driver.html

第 11 章 嵌入式软件开发相关硬件知识

嵌入式软件开发期间，也会涉及到很多硬件相关的知识。

比如ARM的LEON3：想要在LEON3基础上添加更多模块，则可参考：[LEON3 Integration](http://people.xiph.org/~j/bzr/theora-fpga/doc/leon3_integration/)¹

11.1. 芯片封装方式

搞嵌入式软件开发，往往会涉及很多芯片，其中芯片的外形和大小，和其封装方式有关。

芯片封装方式，有很多种，比如：

- SIP==系统式封装
- MCP==多芯片封装

TODO：抽空了解封装方面的知识：

<http://www.spansion.com/CN/Products/diewafer/Pages/DieWafer-NOR-Flash.aspx>

¹ http://people.xiph.org/~j/bzr/theora-fpga/doc/leon3_integration/

第 12 章 嵌入式软件开发相关的厂商

嵌入式软件方面的开发期间，会遇到各种不同的软件的、硬件的、开发板的等等方面的厂商、公司。

对此需要有一定了解，以便于熟悉本行业的情况，便于后续开发。

下面来总结一下常见的各方面的公司、厂商：

12.1. 嵌入式软件厂商

12.1.1. 嵌入式软件厂商：Keil

<http://www.keil.com/>

中的介绍，支持很多类似的CPU：

- [MDK-ARM Microcontroller Development](#)¹
- [8051 Development Tools](#)²
- [C166 Development Tools](#)³
- [DS-5 Development Studio](#)⁴

另外，注意到，Keil也支持AMS的芯片：

<http://www.keil.com/dd/>中的：[AustriaMicroSystems AS3525](#)⁵和：[List of ARM7/ARM9/Cortex Devices Available from AustriaMicroSystems](#)⁶

需要提醒的是：Keil后来被ARM买了，现在属于ARM旗下的品牌了。

12.2. 嵌入式硬件厂商

12.2.1. 嵌入式硬件开发板厂商

嵌入式软件开发期间，常会遇到很多嵌入式开发板。

下面就来整理一些嵌入式开发板：

- 天嵌科技
之前用过ARM9的TQ2440的开发板，觉得还不错，资料算很齐全。

各种软硬件支持的也不错。

¹ <http://www.keil.com/arm/>

² <http://www.keil.com/c51/>

³ <http://www.keil.com/c166/>

⁴ <http://www.keil.com/ds5>

⁵ <http://www.keil.com/dd/chip/4389.htm>

⁶ <http://www.keil.com/dd/chips/austriamicrosystems/arm.htm>

- 安赛卓尔电子科技的AM335X开发板EZ335X-EVB
看到一个觉得算特别的点：

双MAC的MCU：唯一一个集成2个MAC的MCU

其他还有其他特点：

支持多种操作系统

已在多个领域内应用

可编程实时单元和工业用通信子系统(PRU-ICSS)从ARM内核分离，从而实现了针对更大效率和灵活性的独立运行和时钟控制。

PRU-ICSS支持附加外设接口和诸如EtherCAT，PROFINET，EtherNet/IP，PROFIBUS，以太网POWERLINK，串行实时通信协议(Sercos)的实时协议和其它协议。

第 13 章 嵌入式软件开发相关接口和协议

嵌入式软件方面的开发期间，会遇到各种常见的协议、接口、总线等等。

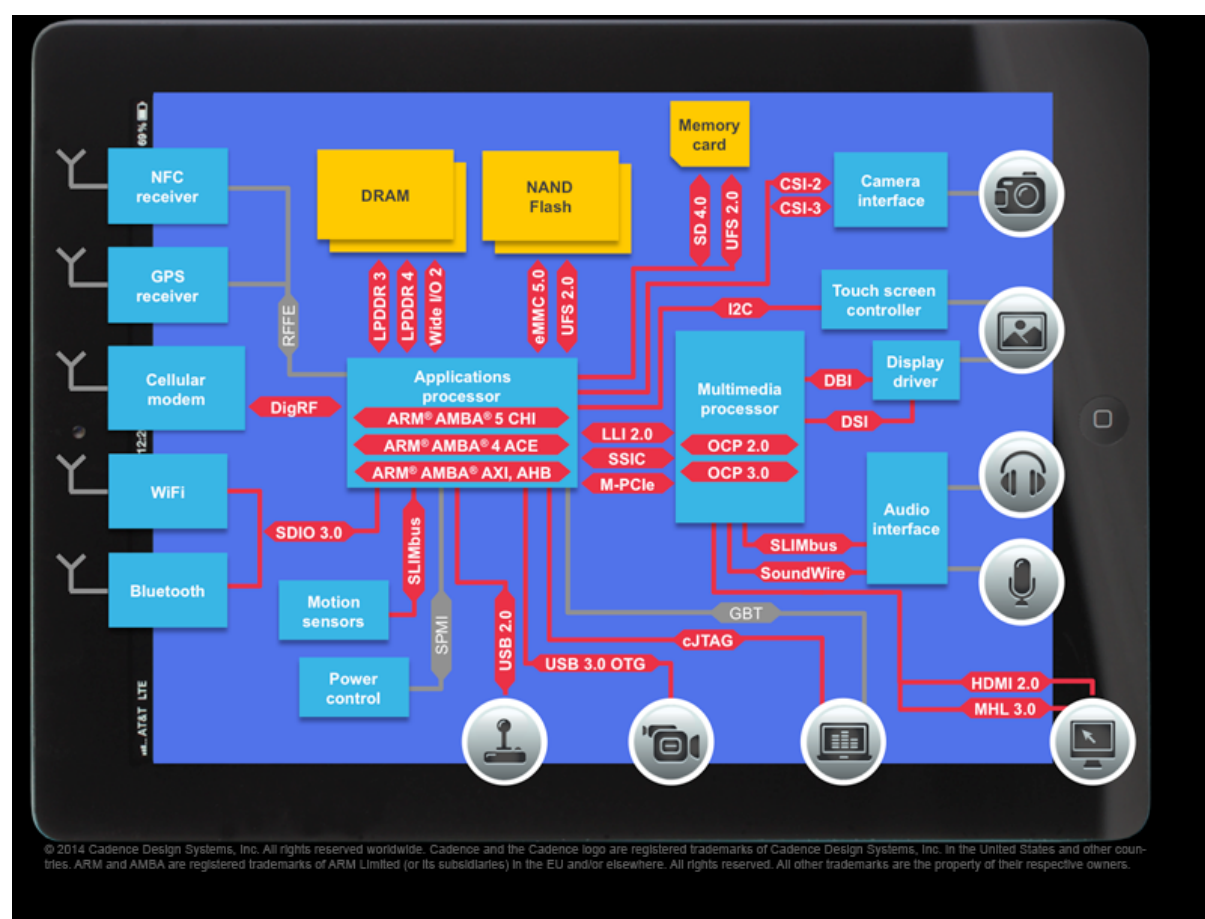
下面就来总结一下，常见的协议和接口，总线。

13.1. 嵌入式总线之内部总线

13.1.1. 嵌入式内部总线之MIPI

以前都没听说过MIPI。后来无意间在cadence的一个资料中看到了MIPI，说是目的是以后代替其他所有的内部总线：

图 13.1. 用于取代其他内部总线的MIPI总线



貌似上述的内部总线，都会被MIPI取代，好像是包含C-PHY, D-PHY, M-PHY, and UniPro

13.1.2. 嵌入式内部总线之AMBA

13.2. 嵌入式总线之外部总线

第 14 章 嵌入式软件开发常见问题与解答

嵌入式软件开发领域内，有很多常见的疑问。此处试图解答清楚这类疑问。

14.1. 学习嵌入式C语言要掌握到什么程度？

比如要求C语言能编多少行的程序，还是看懂多么复杂的程序，还是只要会修改就行。

主要是针对：[学习嵌入式C语言要掌握到什么程度？ - 知乎¹](http://www.zhihu.com/question/29590155/answer/44887185)的回答。

其实嵌入式，尤其是嵌入式底层驱动开发，对于写代码的多少和价值，用一句经典的话来描述就是：

如果写一个嵌入式驱动，一共100行，

写出来那100行代码，本身可能只值1块钱

而搞懂如何写，值99元

->

嵌入式，尤其是底层驱动开发，其实代码量相对（尤其是上层软件）都不是很大，但是要搞懂背后的逻辑，协议，接口，框架，平台等内容，然后才能写出来代码。

而搞懂背后的逻辑，往往才是难点和重点。

->

所以，往往也不能简单说就是，需要你懂多么多么复杂的逻辑的代码，而是：

看似代码量不多，但是寥寥几行代码背后，所涉及到的软件和硬件的知识，系统方面的知识，框架方面的知识，才是对你来说，最复杂的。

->

当然，如果搞懂了应该如何写，往往也就会修改对应的代码了。

总的来说：

嵌入式开发，需要的背景知识：

- 行业背景知识
- 操作系统，平台
- 平台下的（驱动）框架
- （驱动的）具体实现细节
- 看懂设备本身的手册搞懂如何配置操作

等等，很多很杂的内容。而在这些都搞懂的前提下，对于C语言本身的语言方面的要求，其实相对来说并不高：只需要有一定的编码经验，懂C语言语法，即可去写代码，修改代码。

简言之：

- 搞懂如何写：大概占70%的技术比重

¹ <http://www.zhihu.com/question/29590155/answer/44887185>

- 而具体写出来（高质量的）C语言代码：可能只占30%左右的技术比重
- 而关于嵌入式的，行业背景知识，系统平台和框架，驱动框架等等方面的东西：

1. 先看：[嵌入式软件开发](#)²搞懂嵌入式软件。
2. 嵌入式领域内的，各种驱动共性的东西：[嵌入式驱动开发](#)³
3. 单独针对Linux系统下面的软件开发：[嵌入式Linux软件开发](#)⁴
4. 针对Linux下面的，驱动方面的开发：[嵌入式Linux驱动开发](#)⁵

² http://www.crifan.com/files/doc/docbook/embedded_soft_dev/release/html/embedded_soft_dev.html

³ http://www.crifan.com/files/doc/docbook/embedded_drv_dev/release/html/embedded_drv_dev.html

⁴ http://www.crifan.com/files/doc/docbook/embedded_linux_dev/release/html/embedded_linux_dev.html

⁵ http://www.crifan.com/files/doc/docbook/embedded_linux_drv_dev/release/html/embedded_linux_drv_dev.html

第 15 章 嵌入式软件开发方面的值得借鉴的帖子

在嵌入式开发领域，有很多的别的走过的路，做过的事情，值得参考和借鉴的。整理如下：

[我希望成为一个优秀的IT人，我已经努力了七年](#)¹

后，感悟：也是可以去继续学习硬件电路设计的大不了，靠自己自学。毕竟现在算是有了一定的基础，足够的网络资源，足够的学习方法，

善于组织和利用的话，还是可以靠自己，一点点积累出来对应的硬件机能的。如此，加上自己对软件的理解，才能最终算是对于计算机的软件和硬件，都有自己的理解，才可能设计出好的东西

这句我觉得说的不错：<http://bbs.csdn.net/topics/40212259?list=lz>

to DancingCalf(未出手的飞刀)：其实并没有本质上的改变，我一直在IT行业学习工作。让我作出改变的原因是我想做的更加深入，可以说 一个对操作系统不了解的人很难写出优秀的应用系统，对硬件不了解的人也不可能深刻理解操作系统，同样对IC不了解的人也无法真正理解硬件，这决定了我的发展方向。当然这样是无穷无尽的，这也应了那句古言“学无止境”。一个人不可能学会全部的知识，但我不想用这个理由为自己的停滞不前做解释。

¹ <http://bbs.csdn.net/topics/40212259>

第 16 章 嵌入式软件开发相关资源

16.1. 嵌入式软件开发书籍

关于嵌入式软件开发，有一些还不错的值得推荐的书：

16.1.1. Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers

这本书，以一个实际的嵌入式系统开发的例子去解释了，嵌入式开发期间到底会涉及到哪些知识，具体的系统架构是什么样的，嵌入式硬件和软件分别有哪些内容等等。

从整体嵌入式框架，到嵌入式的Host端和嵌入式开发板端，感觉就是无所不包。

所以可以说，该书最大的特点，算是内容非常翔实。

比如，在“3.7 Putting It All Together: The Integrated Circuit (IC)”中，把嵌入式中涉及到的电阻，电容，封装形式等等，都简单解释了基本概念。这类知识的介绍，对于嵌入式入门和开发，很有帮助。

如此，等你看完了该书，就能真正的对于真实的嵌入式系统，有了很好的概念了。

网上找到的免费的PDF版本的下载地址：[\[PDF\] Embedded Systems Architecture A Comprehensive Guide for Engineers and Programmers](http://www.csit-sun.pub.ro/~cpop/Documentatie_SM/Sisteme_incorporate/Embedded%20Systems%20Architecture%20-%20A%20Comprehensive%20Guide%20for%20Engineers%20and%20Programmers.pdf)¹

亚马逊也有卖的：[\[Amazon\] Embedded Systems Architecture A Comprehensive Guide for Engineers and Programmers](http://www.amazon.com/Embedded-Systems-Architecture-Second-Edition/dp/0123821967)²

¹ http://www.csit-sun.pub.ro/~cpop/Documentatie_SM/Sisteme_incorporate/Embedded%20Systems%20Architecture%20-%20A%20Comprehensive%20Guide%20for%20Engineers%20and%20Programmers.pdf

² <http://www.amazon.com/Embedded-Systems-Architecture-Second-Edition/dp/0123821967>

第 17 章 如何学习嵌入式软件开发

要写优质的教程，让别人看得懂的教程，才有价值。

而关于如何写教程，举个例子：

看了这个：

[基于S3C2410的SD卡linux驱动工作原理](#)¹

后，所想到的：

如果我去写对应的该部分的内容，那么应该是：

1. 先去解释说明，当没有Linux，Uboot等各种框架时，裸的驱动，是如何实现的
最好把全部代码贴出来

或者至少把驱动核心部分框架和核心代码贴出来。
2. 然后再去说，对于这样的裸驱动来说，其全部的活，被（Linux，Uboot等）框架，做了哪些
只有搞懂了，框架帮你做了哪些了

然后你才懂，余下来的事情，就是自己需要实现的了。
3. 搞懂需要实现哪些接口函数
再去一点点，搞懂函数的接口和内部实现的逻辑

然后再去写代码，一点点实现

再去边调试边写驱动的代码

网上有很多嵌入式教程写的都不够透彻，没将明白要修改文件背后的逻辑，为何要这么修改。

比如：[mini2440 u-boot-2009.03 移植最详细攻略](#)²

说实话，已经写的，算是步骤相对比较详细了。

但是：

最核心的，最根本的：

如何移植，为何要改这些文件

以及背后的逻辑，都没有解释

导致的结果：

除非本身对这方面的移植已经有概念的

否则，很容易看着一头雾水：

感觉就是：

这个文件改改

¹ http://www.crifan.com/based_on_s3c2410_linux_driver_sd_card_works/

² <http://jesserei.blog.163.com/blog/static/1214116892009931103415765/>

那个文件改改

而且为何这么修改，往往解释的也不够清楚和详细

使得：

如果自己拿到对应的源码，对应的硬件

对于如何移植，还是一头雾水

所以，抽空自己去：

亲自折腾一遍移植

并且详细记录步骤

如果要修改文件或其他地方

一定要解释清楚背后的逻辑

然后等移植完毕

再花时间，搞懂uboot的框架

搞懂如果从头到尾，自己亲力亲为，从无到有的，去移植Uboot

背后的逻辑是啥

需要关心哪些功能

这些功能Uboot中已经有了什么（框架，驱动）了。

还需要我们去实现哪些东西

以及，具体再解释，针对每个要改的功能模块

自己应该如何去改

所以要：

思路清晰

再去下手

否则：

永远都是：只能照葫芦画瓢，而且画的还不像

只有：

自己搞懂逻辑和概念了，

才能：

不仅可以快速高效的，照葫芦画瓢，而且如果需要，也可以自己从无到有的去涉及一个葫芦或一个瓢（可供别人当参考）

结论：

抽空先自己搞懂如何移植Uboot，然后再抽空整理出思路和有参考价值的教程。

参考书目

- [1] [关于嵌入式驱动的学习？ - 知乎](#)¹
- [2] [关于软件工程和嵌入式的一些问题，嵌入式及软件的前辈请进 百度知道](#)²
- [3] [android framework 图解 - kaixinbingju的专栏 - 博客频道 - CSDN.NET](#)³
- [4] [MIPI VIP Resources | Cadence IP](#)⁴
- [5] [2013-2014 年（第六届）中国嵌入式开发从业人员调查报告](#)⁵
- [6] [硬件行业知识体系概要-CSDN论坛-CSDN.NET](#)⁶
- [7] [Senior Embedded Software Engineer招聘 Senior Embedded Software Engineer招聘信息-猎聘网](#)⁷
- [8] [嵌入式开发之嵌入式背景知识](#)⁸
- [9] [嵌入式开发之基本概念和开发逻辑](#)⁹
- [10] [【问题解答】用一个嵌入式开发板去实现数据采集的基本思路和逻辑](#)¹⁰
- [11] [【整理】嵌入式领域的名词概念解释](#)¹¹
- [12] [【整理】嵌入式开发经验和心得总结](#)¹²
- [13] [广州天嵌科技 - TQ_AM335x+7寸电容屏 - 产品详细](#)¹³
- [14] [AM335X开发板 百度百科](#)¹⁴
- [15] [【整理】类比解释：嵌入式开发需要掌握哪些知识](#)¹⁵
- [16] [【整理】嵌入式软件知识体系概述](#)¹⁶
- [17] [【整理】嵌入式领域中常见的名词和概念的解释 | 在路上](#)¹⁷

¹ <http://www.zhihu.com/question/21457018/answer/19517807>

² <http://zhidao.baidu.com/question/559495337288649284.html>

³ <http://blog.csdn.net/kaixinbingju/article/details/8835744>

⁴ <http://ip.cadence.com/knowledgecenter/know-vip/vip-resource-pages/mipi-vip-resources>

⁵ <http://download.farsight.com.cn/download/pdf/EmbeddedReport2013-2014.pdf>

⁶ <http://bbs.csdn.net/topics/390487802>

⁷ http://a.liepin.com/4442902/job_6186910.shtml?sfrom=recom-PC_JD-supply_14

⁸ http://www.crifan.com/embedded_development_background/

⁹ http://www.crifan.com/embedded_development_basic_concept_develop_logic/

¹⁰ http://www.crifan.com/qanda_embedded_data_acquisition_system_implement_process/

¹¹ http://www.crifan.com/summary_embedded_concept_name_explanation/

¹² http://www.crifan.com/summary_embedded_development_experience_and_note/

¹³ <http://www.embedsky.com/index.php?s=/Product/show/id/49.html>

¹⁴ <http://baike.baidu.com/view/10170430.htm>

¹⁵ http://www.crifan.com/summary_use_similar_building_field_explain_embedded_system_need_learn_which_kind_of_knowledge/

¹⁶ http://www.crifan.com/embedded_system_software_field_knowledge_overview/

¹⁷ http://www.crifan.com/summary_embedded_concept_name_explanation/