

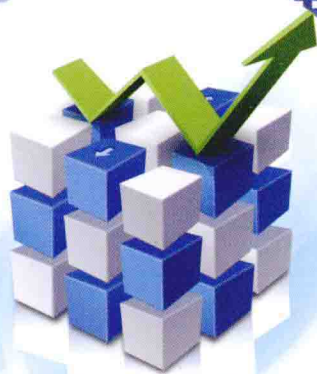


普通高等教育卓越工程能力培养规划教材

# 基于ARM的 单片机应用及实践

## ——STM32案例式教学

武奇生 白璘 惠萌 巨永锋 © 编著



机械工业出版社  
CHINA MACHINE PRESS

普通高等教育卓越工程能力培养规划教材

# 基于 **ARM** 的单片机应用及实践 ——**STM32** 案例式教学

武奇生 白 璘 惠 萌 巨永锋 编著

---

机械工业出版社

本书的内容涵盖了基于 ARM 的 STM32 系统的基本概念、原理、技术和应用案例,结合计算机的发展史说明了单片机技术的最新进展和发展趋势。本书按照“卓越工程师教育培养计划”的理念,以案例式教学为主,培养学生的工程实践能力。

本书论述严谨、内容新颖、图文并茂,注重基本原理和基本概念的阐述,强调理论联系实际,突出应用技术和实践,并安排了丰富的教学实验和实际场景训练。

本书可作为高等院校自动化及相关专业大学本科的教材或参考教材,也可作为从事检测、自动控制等工作的工程技术人员的参考用书。

## 图书在版编目(CIP)数据

基于 ARM 的单片机应用及实践: STM32 案例式教学/武奇生等编著. —北京:机械工业出版社, 2014. 3

普通高等教育卓越工程能力培养规划教材

ISBN 978-7-111-45803-6

I. ①基… II. ①武… III. ①单片微型计算机—高等学校—教材  
IV. ①TP368. 1

中国版本图书馆 CIP 数据核字(2014)第 026094 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:于苏华 责任编辑:于苏华

版式设计:常天培 责任校对:陈延翔

封面设计:张静 责任印制:李洋

北京瑞德印刷有限公司印刷(三河市胜利装订厂装订)

2014 年 6 月第 1 版第 1 次印刷

184mm × 260mm · 21.25 印张 · 519 千字

标准书号:ISBN 978-7-111-45803-6

定价:42.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010)88361066

教材网:<http://www.cmpedu.com>

销售一部:(010)68326294

机工官网:<http://www.cmpbook.com>

销售二部:(010)88379649

机工官博:<http://weibo.com/cmp1952>

读者购书热线:(010)88379203

封面无防伪标均为盗版

# 前 言

在长安大学自动化专业 2010 年入选国家第一批“卓越工程师教育培养计划”试点专业之后，我们一直在思考这样的问题：学生达到怎样的工程实践能力，才能去联合企业实施校企联合培养？企业对学生的要求是什么？在校内学习阶段对学生如何培养？

对此，我们一方面请部分企业的总工和一线工程技术人员来学校研讨，另一方面去相关企业进行调研。通过交流我们了解到，企业对学生的实践能力，基础知识、专业知识的掌握，研发能力，团队合作精神等方面都有要求。其中，有些要求是很具体的。例如，在讨论中，企业尤其谈到了生产对工业机器人的需求。我国正处在经济转型的关键时期，更多的行业将使用工业机器人替代人力从事危险、单调、重复的劳动，以提高生产工艺的柔性、生产效率和产品质量。工业机器人是自动化技术高度发展的产物，它综合了计算机、控制论、机构学、信息和传感技术、人工智能、仿生学等诸多学科，是先进制造技术领域不可缺少的自动化设备。而且，工业机器人的应用是一个国家工业自动化水平的重要标志，更是自动化学科的重要研究领域。因此，未来的企业将需要大量的自动化专业工程技术人员去研发、维护这些用于生产的工业机器人。

第二方面，我们在参加了多次全国性教改会议，深入了解了当前高等教育改革的要求后，深切地体会到，提高人才培养质量是高等教育内涵式发展的关键所在，是教师义不容辞的责任。

为了满足国家经济转型发展的需求，使得学生在校内学习阶段具备工程实践能力，适应企业开展校企联合培养的要求，通过对学生认知规律的研究，按照“卓越工程师教育培养计划”要求设计了“一级项目”，对应“一级项目”，我们组织研发、制作了“CHD1807—STM32F103 开发系统”实验装置，用于“STM32 案例式教学”，并编写了这本教材和实验指导书。本书旨在从大学低年级开始，通过完成项目达到培养目标，对学生的基础知识、专业知识的掌握，研发能力，团队合作精神等方面进行培养。以案例式的教学为主，在实验中开展对学生工程实践能力的培养，提升他们的专业兴趣，使得学生更加理解专业的内涵，走内涵式发展之路。

本书由武奇生、白璘、惠萌、巨永锋四人编著，武奇生负责统稿。全书编写的具体分工为：武奇生（第 1、2 章）、巨永锋（第 3 章）、惠萌（第 4、5、6 章）、白璘（第 7、8、9 章），第 10 章由白璘、武奇生共同完成。孙眉浪等研究生绘制了书中的部分插图，对本书的初稿进行了阅读和校对。

在本书即将出版之际，回顾近一年的编写、试用过程，时常想起以下需要感谢的人和事。

2011 年 11 月 6 日，我们受邀参加了上海庆科公司（MXCHIP）在西安举行的 STM 技术讲座，施海工程师给予我们大力的技术支持，使得我们在教学改革方面得到了许多帮助；

2011 年 12 月在上海，总经理王永虹邀请我们参观了上海庆科公司，详细介绍了公司研发的物联网技术产品，向我们赠送了样片，获益匪浅。

大暑节气后的西安，骄阳似火，参与制作 CHD1807—STM32F103 实验装置的长安大学 2010 级“卓越工程师教育培养计划”试点班的同学们，仍然在闷热的实验室用电烙铁焊接电路板，而且还要戴着保护口罩，汗水浸透了衣服。此情此景，记忆犹新；换位体验，令人深深感动。同学们任劳任怨、不计名利、团结协作的态度，充分体现出了以祖国需要为己任、坚定科学报国的信念。这些同学是王爱民、何运来、冯仰刚、曹清源、马旭攀、景首才、郝熠、朱进玉、谢乾坤等，在此，对这些同学的辛勤劳动表示感谢。

本书的完成获得了汪贵平教授的大力帮助及长安大学国家级自动化特色专业建设点专项经费、2013 年陕西省高等教育教学改革研究一般项目（陕教高 [2013] 45 号 - 13BY28）的资助。

在本书的写作过程中参阅了许多资料，在此对编写本书时所参考书籍的作者一并表示诚挚的感谢。本书编写过程中引用了互联网上最新资讯及报道，在此向原作者和刊发机构表示诚挚的感谢，并对不能一一注明来源深表歉意。对于收集到的共享资料没有标明出处或找不到出处的，以及对有些资料进行加工、修改后纳入本书的，我们在此郑重声明，其著作权属于原作者，并向他们表示致敬和感谢。

由于水平和时间有限，书中难免存在错误和不妥之处，恳请同行专家和读者批评指正。

作 者

# 目 录

## 前言

第1章 概述 .....	1	2.3.7 存储器的默认访问许可 .....	47
1.1 计算机发展史 .....	1	2.4 工作模式 .....	47
1.1.1 计算机的诞生 .....	1	2.5 异常与中断 .....	49
1.1.2 计算机的发展 .....	5	2.5.1 中断号与优先级 .....	50
1.2 计算机的体系结构 .....	6	2.5.2 向量表 .....	54
1.2.1 冯·诺依曼架构模型 .....	6	2.5.3 中断输入及挂起 .....	55
1.2.2 面向嵌入式应用的架构改进 .....	9	2.5.4 Fault 类异常 .....	56
1.3 单片机发展史 .....	10	2.5.5 中断的具体行为 .....	57
1.3.1 计算机及早期单片机 .....	10	2.5.6 中断嵌套控制 .....	59
1.3.2 单片机的发展趋势——走向集成、 嵌入式 .....	11	2.5.7 高级中断操作 .....	59
1.4 ARM、Cortex 和 STM32 简介 .....	12	2.5.8 异常返回值 .....	61
1.4.1 ARM 系列内核 .....	12	2.6 堆栈 .....	62
1.4.2 Cortex 系列内核 .....	16	2.6.1 堆栈的基本操作 .....	62
1.4.3 STM32F103 系列微控制器 .....	17	2.6.2 Cortex-M3 堆栈操作 .....	63
1.5 计算机发展的趋势和工程设计开发 .....	21	2.6.3 Cortex-M3 的双堆栈机制 .....	64
1.5.1 计算机发展的趋势 .....	21	2.7 小结 .....	64
1.5.2 嵌入式系统的工程设计和开发 .....	22	习题 .....	64
1.6 小结 .....	25	第3章 STM32 最小系统的设计 .....	65
习题 .....	25	3.1 STM32F103 最小系统的设计方案 .....	65
第2章 Cortex-M3 处理器 .....	26	3.2 最小系统设计的要素 .....	67
2.1 Cortex-M3 内核 .....	26	3.2.1 STM32 晶体振荡器 .....	68
2.1.1 内核体系结构 .....	26	3.2.2 复位电路 .....	69
2.1.2 系统总线结构 .....	29	3.2.3 LED、Key 及 BOOT 跳线 .....	70
2.2 寄存器 .....	30	3.2.4 稳压电源及 ISP 下载口 .....	71
2.2.1 通用寄存器 .....	31	3.2.5 I/O 端口 .....	74
2.2.2 特殊功能寄存器 .....	33	3.3 PCB 图设计 .....	75
2.3 存储器管理 .....	36	3.4 小结 .....	76
2.3.1 寄存器空间分配 .....	37	第4章 MDK-ARM 软件入门 .....	77
2.3.2 位带操作 .....	38	4.1 MDK-ARM 4.70 简介 .....	77
2.3.3 互斥访问 .....	43	4.2 新建 MDK 工程 .....	78
2.3.4 端模式 .....	45	4.2.1 下载外设库 .....	78
2.3.5 存储保护单元 .....	46	4.2.2 建立新工程 .....	79
2.3.6 存储器访问属性 .....	46	4.2.3 工程编译环境设置 .....	87
		4.3 RVMDK 使用技巧 .....	91
		4.3.1 文本美化 .....	92

4.3.2 代码编辑技巧 .....	92	5.4.1 移位运算 .....	124
4.3.3 其他小技巧 .....	98	5.4.2 按位与运算 .....	126
4.4 调试与下载 .....	98	5.4.3 按位或运算 .....	127
4.4.1 STM32 软件仿真 .....	98	5.4.4 取反运算 .....	127
4.4.2 STM32 程序下载 .....	106	5.4.5 异或运算 .....	128
4.5 固件函数库函数命名规则 .....	108	5.5 GPIO 控制实例 .....	128
4.6 小结 .....	110	5.5.1 实例 1——控制 LED 闪烁 .....	128
习题 .....	110	5.5.2 实例 2——跑马灯 .....	130
<b>第 5 章 GPIO 及外部中断的使用</b> .....	<b>111</b>	5.5.3 实例 3——按键输入 1 .....	133
5.1 综述 .....	111	5.6 外部中断和中断控制器 .....	137
5.1.1 通用 I/O .....	112	5.6.1 嵌套向量中断控制器 .....	137
5.1.2 单独的位设置或位清除 .....	112	5.6.2 外部中断/事件控制器 .....	141
5.1.3 外部中断/唤醒线 .....	112	5.6.3 NVIC 库函数介绍 .....	142
5.1.4 复用功能 .....	112	5.6.4 外部中断控制器库函数介绍 .....	146
5.1.5 软件重新映射 I/O 复用功能 .....	112	5.6.5 外部中断实例——按键输入 2 .....	149
5.1.6 GPIO 锁定机制 .....	112	5.7 小结 .....	150
5.1.7 输入配置 .....	113	习题 .....	151
5.1.8 输出配置 .....	113	<b>第 6 章 定时器的使用</b> .....	<b>152</b>
5.1.9 复用功能配置 .....	113	6.1 STM32F 的定时器简介 .....	152
5.1.10 模拟输入配置 .....	113	6.1.1 高级控制定时器 TIM1 的 TIM8 .....	152
5.2 库函数 .....	114	6.1.2 通用定时器 TIMx .....	153
5.2.1 函数 GPIO_Init .....	114	6.1.3 基本定时器 TIM6 和 TIM7 .....	153
5.2.2 函数 GPIO_SetBits .....	116	6.2 通用定时器功能描述 .....	154
5.2.3 函数 GPIO_ResetBits .....	116	6.2.1 时基单元 .....	154
5.2.4 函数 GPIO_WriteBit .....	117	6.2.2 计数器模式 .....	155
5.2.5 函数 GPIO_Write .....	117	6.2.3 时钟选择 .....	161
5.2.6 函数 GPIO_ReadOutputDataBit .....	117	6.2.4 捕获/比较通道 .....	163
5.2.7 函数 GPIO_ReadOutputData .....	118	6.2.5 输入捕获模式 .....	164
5.2.8 函数 GPIO_ReadInputDataBit .....	118	6.2.6 PWM 输入模式 .....	165
5.2.9 函数 GPIO_ReadInputData .....	119	6.2.7 强置输出模式 .....	166
5.3 I/O 端口的外设映射 .....	119	6.2.8 输出比较模式 .....	166
5.3.1 将 OSC_32 IN/OSC_32 OUT 作为 .....	120	6.2.9 PWM 模式 .....	167
PC14/PC15 端口 .....	120	6.2.10 定时器和外部触发的同步 .....	169
5.3.2 将 OSC_IN/OSC_OUT 作为 .....	120	6.2.11 定时器同步 .....	171
PD0/PD1 端口 .....	120	6.3 定时器固件库函数介绍 .....	171
5.3.3 CAN 复用功能重映射 .....	120	6.3.1 函数 TIM_DeInit .....	171
5.3.4 JTAG/SWD 复用功能重映射 .....	120	6.3.2 函数 TIM_TimeBaseInit .....	172
5.3.5 ADC 复用功能重映射 .....	121	6.3.3 函数 TIM_OC1Init .....	173
5.3.6 定时器复用功能重映射 .....	121	6.3.4 函数 TIM_OC2Init .....	175
5.3.7 USART 复用功能重映射 .....	122	6.3.5 函数 TIM_OC3Init .....	175
5.3.8 I <sup>2</sup> C1 复用功能重映射 .....	123	6.3.6 函数 TIM_OC4Init .....	176
5.3.9 SPI1 复用功能重映射 .....	123	6.3.7 函数 TIM_ICInit .....	176
5.4 位运算 .....	124	6.3.8 函数 TIM_BDTRConfig .....	178

6.3.9	函数 TIM_Cmd	179	<b>第 8 章 STM32 显示模块操作</b>	219
6.3.10	函数 TIM_CtrlPWMOutputs	179	8.1 OLED 显示	219
6.3.11	函数 TIM_ITConfig	180	8.1.1 OLED 的驱动方式	219
6.3.12	函数 TIM_SelectInputTrigger	181	8.1.2 硬件设计	221
6.3.13	函数 TIM_EncoderInterfaceConfig	181	8.1.3 软件设计	225
6.3.14	函数 TIM_ARRPreloadConfig	182	8.2 TFT LCD 显示	226
6.3.15	函数 TIM_CCPreloadControl	182	8.2.1 TFT LCD 的特点	226
6.3.16	函数 TIM_OC1PreloadConfig	183	8.2.2 硬件设计	227
6.3.17	函数 TIM_OC2PreloadConfig	183	8.2.3 软件设计	228
6.3.18	函数 TIM_OC3PreloadConfig	184	8.3 图片显示实例	230
6.3.19	函数 TIM_OC4PreloadConfig	184	8.3.1 图片显示原理简介	230
6.3.20	函数 TIM_SelectOutputTrigger	184	8.3.2 硬件设计	231
6.3.21	函数 TIM_SelectSlaveMode	185	8.3.3 软件设计	232
6.3.22	函数 TIM_SelectMasterSlaveMode	186	8.4 汉字显示实例	261
6.3.23	函数 TIM_SetCounter	186	8.4.1 汉字显示原理简介	262
6.3.24	函数 TIM_SetAutoreload	187	8.4.2 硬件设计	265
6.3.25	函数 TIM_GetCounter	187	8.4.3 软件设计	265
6.3.26	函数 TIM_GetPrescaler	188	8.5 小结	267
6.3.27	函数 TIM_GetFlagStatus	188	习题	267
6.3.28	函数 TIM_ClearFlag	189	<b>第 9 章 STM32 外设接口模块</b>	268
6.3.29	函数 TIM_GetITStatus	189	9.1 USART 模块	268
6.3.30	函数 TIM_ClearITPendingBit	190	9.1.1 USART 功能描述	268
6.4	定时器实训	190	9.1.2 USART 寄存器简介	270
6.4.1	定时器控制跑马灯	190	9.1.3 USART 操作实例	273
6.4.2	PWM 电动机控制	192	9.2 SPI 模块	274
6.5	小结	197	9.2.1 SPI 简介	275
习题		198	9.2.2 SPI 功能描述	275
<b>第 7 章 STM32 的 A/D 转换模块</b>		199	9.2.3 SPI 配置简介	279
7.1	ADC 的主要技术指标及选型	199	9.2.4 SPI 操作实例	280
7.1.1	ADC 的主要技术指标分析	200	9.3 I <sup>2</sup> C 模块	281
7.1.2	ADC 的选型技巧及注意事项	201	9.3.1 I <sup>2</sup> C 简介	281
7.2	STM32 的 ADC	203	9.3.2 I <sup>2</sup> C 功能描述	282
7.2.1	ADC 功能描述	203	9.3.3 I <sup>2</sup> C 配置简介	283
7.2.2	ADC 寄存器描述	209	9.3.4 I <sup>2</sup> C 操作实例	283
7.2.3	ADC 中断	214	9.4 CAN 总线模块	286
7.3	STM32 内部温度传感器	214	9.4.1 CAN 简介	286
7.3.1	硬件设计	214	9.4.2 CAN 功能描述	287
7.3.2	软件设计	215	9.4.3 CAN 配置简介	288
7.4	STM32 的 ADC 实验	217	9.4.4 CAN 操作实例	290
7.4.1	硬件设计	217	9.5 USB 模块	292
7.4.2	软件设计	217	9.5.1 USB 简介	293
7.5	小结	218	9.5.2 USB 功能描述	293
习题		218	9.5.3 USB 配置简介	294



9.5.4	USB 操作实例	295	10.4.2	储能式光电寻迹车软硬件设计	313
9.6	小结	297	10.4.3	储能式光电寻迹车的实现	316
	习题	297	10.5	MP3 播放器	317
<b>第 10 章</b>	<b>STM32 综合实验</b>	<b>298</b>	10.5.1	MP3 播放器设计要求	317
10.1	STM32 开发板	298	10.5.2	MP3 播放器软硬件设计	317
10.1.1	STM32 开发板资源介绍	298	10.5.3	MP3 播放器的实现	319
10.1.2	STM32 开发板硬件设计	300	10.6	基于 GPRS 的电热水器控制系统	323
10.2	追光系统	304	10.6.1	基于 GPRS 的电热水器控制系统 设计要求	323
10.2.1	追光系统设计要求	304	10.6.2	基于 GPRS 的电热水器控制系统 软硬件设计	323
10.2.2	追光系统软硬件设计	305	10.6.3	基于 GPRS 的电热水器控制系统 实现	326
10.2.3	追光系统实现	307	10.7	小结	329
10.3	倒立摆	308		习题	329
10.3.1	倒立摆设计要求	308	<b>参考文献</b>	<b>331</b>	
10.3.2	倒立摆软硬件设计	309			
10.3.3	倒立摆的实现	311			
10.4	储能式光电寻迹车	312			
10.4.1	储能式光电寻迹车设计要求	312			

# 第 1 章

## 概 述

信息技术发展到今天，离不开计算机技术的发展，计算机技术的发展，走过了几十年的历程。本章首先回顾计算机的发展过程，介绍计算机系统和单片机系统，为后面单片机的学习打下基础。

### 1.1 计算机发展史

#### 1.1.1 计算机的诞生

什么是计算机？现代意义上的计算机，与古代的计算辅助工具，如中国的算盘和欧洲中世纪的莱布尼兹计算器有何本质不同？对后一问题的回答，归根到底还是要谈到图灵计算机模型。该模型的强大计算能力取决于两点：存储程序及其动态修改能力，而这恰恰是一切“计算器”设备所缺乏的。而“计算机”，就可以认为是这一理论模型的物理实现，而且不论该物理实现是采用机械装置、电子管技术、晶体管和集成电路技术、光计算器件还是生物分子技术。

但是，计算机的物理实现受限于每个时代所能提供的技术手段。由此不难理解，英国数学家巴贝奇设计的机械式通用计算机尽管具有和现代计算机一样的程序存储和自动执行等一系列先进思想，但是从 1837 年提出设计方案，到 1871 年巴贝奇去世，这台机器一直没有最终完成。过去，大多数人认为第一台计算机是 1946 年 2 月由宾夕法尼亚大学的莫奇利和艾克特研制成功的电子数字积分计算机 ENIAC (Electronic Numerical Integrator and Calculator)，它从 1946 年 2 月投入使用，到 1955 年 10 月最后切断电源，服役 9 年多。虽然它每秒只能进行 5000 次加、减运算，但它预示了科学家将从奴隶般的计算中解脱出来。但是 ENIAC 本身存在两大缺点：一是没有严格意义上的存储器；二是用布线接板进行控制，非常麻烦，计算速度也被这一人工操作所抵消。所以，ENIAC 是否被认为是一台计算机也引起了一些争议。

有意思的是，计算机设计制造技术的突破几乎是在同一时期由不同的人 and 机构分别独立实现，并共同推动了计算机的早期发展。这也反映了科技上的重大发明从来都是时代进步的结果，并不完全取决于某个个人的努力。

美国衣阿华州立大学的数学物理教授阿塔纳索夫与同事和研究生贝利，用 500 美元的资助和自己的工资，在 1941 年最早采用电子管技术设计了 ABC 计算机。设计始于 1935 年，

于 1939 年完成。阿塔纳索夫等人研制完成了控制器等一些关键部件，但却由于战争期间转入军队服务未全部完工。莫奇利曾亲自到衣阿华州立大学所在地住了 5 天，仔细了解了 ABC 的设计细节和内部工作原理，1941 年将曾阅读过的阿塔纳索夫关于 ABC 计算机设计的笔记内容，运用到之后 ENIAC 的设计中。1973 年，美国明尼苏达地区法院经过数年调查，确认莫奇利的设计是来自与衣阿华州立大学阿塔纳索夫的交谈和其笔记，深受阿塔纳索夫设计的 ABC 计算机的影响，因此 ENIAC 不能作为一项独立发明，故最终正式宣判取消了莫奇利等的计算机专利，肯定了 ABC 的设计者阿塔纳索夫才是真正的现代计算机的发明人。作为一段小插曲，阿塔纳索夫终于得到人们的承认，不过，这也不妨碍 ENIAC 在计算机发展历史上的地位。

### 1. 从数值计算到通用信息处理和智能计算

下面从计算机和计算机的基本模型谈起。在计算机诞生之前，“计算”主要是指数值计算，即使是在计算机发展的早期，计算机本质上也只不过是个体积巨大的机器，主要用于科学研究和军事领域，用于执行数值计算任务。例如，由美国陆军兵器局出资，数学家冯·诺依曼主持设计的 ENIAC 作为电子计算机最早期的代表，主要用于弹道计算，该机在 30s 内即可完成弹道计算，在当时被称为“比子弹还快”的超人。这一发明实现了计算机科学家的第一个设想：自动化的计算。ENIAC 是早期的电子计算机之一，但就自动化的计算或者是机械辅助的计算机这一主题，人类早已开始了各种探索。英国工程师巴贝奇（1791—1871）在 1834 年设计了一台完全用程序控制的机械计算机，通过齿轮旋转来进行计算，用齿轮和杠杆传送数据，用穿孔卡片输入程序和数据，用穿孔卡片和打印机输出计算结果。限于当时的技术条件，这台机器未能制造出来，但巴贝奇的设计思想是不朽的，他与现代电子计算机的设计完全吻合。

伴随着电子技术特别是微电子技术的发展，计算机本身的成本得以大幅度降低，使得其应用范围真正地从小众走向普通大众，PC 一度成为计算机的代名词。而在成本降低的同时，计算机本身的性能却在大幅度提高，特别是存储器容量的增加、工作速度的提高和外围接口（I/O）设备的增加，使得计算机有能力将物理世界中的大量模拟信息，如文字、语音、图片和视频，转换成二进制数字格式进行存储、处理、传输和展示，使得计算机能够处理的数据范围远远扩大，计算机的主要用途也终于由传统的单纯的数值计算演化为通用的信息处理，实现了早期计算机科学家所梦想的第二个目标：通用信息处理，而不是仅仅局限在数值计算。“计算”这个术语的内涵也随之同步扩大，在今天应理解为“信息处理”而不同于“科学计算”，计算机本身，更准确地说，也应称之为“通用信息处理机”。在这一历史趋势下，诞生了 IBM 的 PC286、Apple 的图形用户界面、Microsoft 的 MSDOS 和 Windows 操作系统等一系列优秀的产品，它们使得计算机进入了人们日常的生活，成为日常工作、交流、娱乐的核心平台，这一趋势即使至今也没有改变。

尽管计算机作为通用信息处理机，在过去 30 年里，成功地改造了我们的世界，但是仍应看到，绝大多数情况下，计算机是在人的操作和控制下机械式地处理数据，虽然性能比较高，但是就其智力程度而言，还不如人类中的一个 3 岁小孩。而计算机科学家的第三个梦想，就是希望计算机能够成为机械脑，能够像人脑那样处理输入的数据，如语音和图像，并能自主运行，与人类协作。这一目标也就是智能计算机的终极目标。

但是，在追求智能终极目标的过程中，在许多具体的应用中，却出现了计算能力相对过

剩的问题，许多实际系统的设计更看重功能的完整性、可靠性及成本等非性能问题。因此，计算机工程应用中的一个重要问题是：如何在保证功能完整性、满足用户需求的前提下，综合考虑功能、性能、成本、可靠性多种因素，实现平衡设计，以及如何借助网络通信实现分布式计算。对平衡设计的追求，最终以嵌入式系统的具体形式体现出来，比如一部手机、一台洗衣机、一个机器人等。可以这么说，如同 PC 在 20 世纪 80 年代成为计算机技术的代名词一样，单片机技术 + 智能是目前这个时代计算机技术的主流。

## 2. 计算的基本模型：图灵机理论模型

为什么计算机具有近乎无限的处理能力而不是像人类发明的其他工具那样在诞生之后功能即被固定化？计算的能力边界在何处？它可以应付今天还没有出现的未来问题吗？究竟应该如何理解“计算”的内涵？

为了回答这些最基本问题，计算机理论界的先驱者阿兰·图灵（Alan Turing）提出了图灵机理论模型。阿兰·图灵，1912 年 6 月 23 日出生于英国伦敦，他被认为是 20 世纪最著名的数学家之一和计算机科学的先驱。1936 年，年仅 24 岁的图灵在其著名论文《论可计算数在判定问题中的应用》（On Computable Numbers with an Application to the Entscheidungsproblem）一文中，以布尔代数为基础，将逻辑中的任意命题（即可用数学符号）用一种通用的机器来表示和完成，并能按照一定的规则推导出结论。这篇论文被誉为现代计算机原理开山之作，它描述了一种假想的可实现通用计算机的机器，后人称之为“图灵机”。

图灵的基本思想是用机器来模拟人用纸笔进行数学运算的过程，他把这样的过程看作下列两种简单的动作：

- 1) 在纸上写上或擦除某个符号。
  - 2) 把注意力从纸的一个方向移动到另一个方向。
- 而在每个阶段，人要决定下一步的动作，依赖于此人当前所关注的纸上某个位置的符号，即此人当前思维的状态。

图灵计算机模型的构造如图 1.1 所示。图灵假想的抽象机器包括这样几部分：

1) 一条无限长的纸带 TAPE。纸带被划分为一个接一个小格子，每个格子上包含一个来自有限字母表的符号，字母表中有一个特殊的符号表示空白。纸带上的格子从左到右依次被编号为 0, 1, 2, 3, …，纸带的右端可以无限伸展。

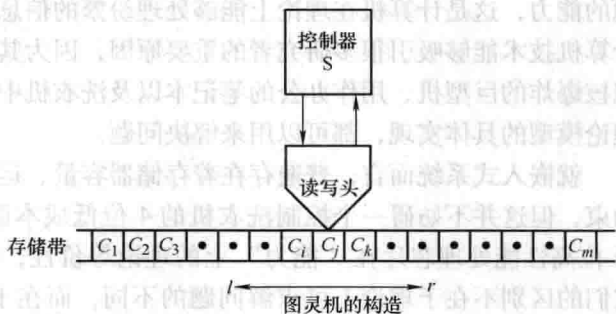


图 1.1 图灵计算机模型的构造

2) 一个读写头 HEAD。该读写头可以在纸带上左右移动，它能读出当前所指的格子上的符号，并能改变当前格子上的符号。

3) 一套控制规则 TABLE。它根据当前机器所处的状态以及当前读写头所指的格子上的符号来确定读写头下一步的动作，并改变状态寄存器的值，令机器进入一个新的状态。

4) 一个状态寄存器。它用来保存机器当前所处的状态。机器的所有可能状态的数目是有限的，并且有一个特殊的状态，称为停机状态。

注意，这个机器的每一部分都是有限的，但它有一个潜在的无限长的纸带，因此这种机器只是一个理想设备。图灵认为这样的一台机器就能模拟人类所能进行的任何计算过程。对于任意一个图灵机，因为它的描述是有限的，因此总可以用某种方式将其编码成一个长字符串，这里用  $\langle M \rangle$  表示图灵机  $M$  的编码。可以构造出一个特殊的图灵机，它接受任意一个图灵机  $M$  的编码  $\langle M \rangle$ ，然后模拟  $M$  的运作，这样的图灵机就称为通用图灵机 (Universal Turing Machine)。现代电子计算机本质上就是一种通用图灵机，它能接受一段描述其他图灵机的程序，并运行程序实现该程序所描述的算法。

图灵说明了这种机器能进行多种运算并可用于证明一些著名的定理。这就是最早给出的通用计算机模型，尽管遵照这一思想设计的具体机器还要再经过 10 年左右才能问世，所谓图灵机设计还是一纸空文，但其思想奠定了整个现代计算机发展的理论基础。

图灵机模型的贡献突出表现在下面几个方面：

1) 它回答了计算的能力范围。这是实现通用信息处理机的必备理论基础。作为计算机领域中的最基本模型，它从最抽象的层次回答了最基本的计算机系统是什么样子，以及这一系统为什么具备“完成通用计算”的能力而不是像历史上其他人类发明的工具那样仅具有少数特征化的功能。换言之，它回答了计算机为什么可以是一台通用信息处理机而不是专用信息处理机。图灵机模型对于一大类有限步数可计算的问题给出了一个普适性的定义，每一个这样的问题都存在一个图灵机可对其进行计算和给出答案。换言之，一个现实问题，不论其多么复杂，如果可以抽象为这样一个有限步数的计算问题，那么它一定是图灵可解的。对这些问题的讨论导致了可计算性和计算复杂度领域的诞生。

2) 符合图灵机原理的不同技术实现在理论上具有相同的计算原理。图灵机模型并没有限定用什么技术来实现它，可以用电子管、晶体管、集成电路等实现，甚至用机械装置实现也没有问题，只要符合图灵机原理，这些装置的计算能力在本质上就是相同的。因此，任何一个符合图灵机模型的计算机系统，不论其简单或复杂，都具备了在理论上处理一切可解问题的能力，这是计算机在理论上能够处理纷繁的信息并得到结果的理论保证。而且，这也是计算机技术能够吸引很多研究者的重要原因，因为其能力是无限的。用于计算天气预报和模拟核爆炸的巨型机、用作办公的笔记本以及洗衣机中控制电机转动的微控制器，都是图灵机理论模型的具体实现，都可以用来解决问题。

就嵌入式系统而言，普遍存在着存储器容量、运算速度、电源、尺寸、成本等各方面的约束，但这并不妨碍一个控制洗衣机的 4 位低成本微处理芯片和一个用于高速图像处理的 64 位高性能处理芯片在“能力”上的理论等价性，因为它们都是图灵机模型的具体实现。它们的区别不在于理论上可求解问题的不同，而在于解决问题的快慢，即所谓的“性能”。一个问题在巨型机上可解，那么换成笔记本或微控制器，理论上也是一定可解的，只不过计算的过程慢许多而已。而这个区别在汉语中常常被混淆，例如人们在评价某人说他很有能力的时候，往往隐含着两重含义，一是他可以解决未知问题和疑难问题，这是他的能力，另一重含义是他做事做得又快又好，这其实是它的效率问题。而图灵机模型中的“能力”(Capability) 是指前者，后者应属于“性能”(Performance) 范畴。今天的计算机，尽管形态各异，本质上都是图灵机模型的一个个技术实现，因此它们都具有相同的理论计算能力。

3) 它在理论上规范了计算机的实现思路。图灵机模型并没有说明如何设计和实现一个计算机系统，但是它已经隐含地说明了一个计算装置应该至少包含存储器（代替图灵机中

的纸带)、运算器和控制器(代替图灵机的读写头和控制器)。这只要再配上输入输出设备就几乎是冯·诺依曼模型了。

1945年图灵结束了战争期间的密码服务工作,来到英国国家物理实验室。他结合自己多年的理论讨论研究和战时制造密码破译机的经验,起草了一份关于研制自动计算机器(Automatic Computer Engine, ACE)的报告,以期实现他曾提出的通用计算机的设计思想。通过长期研究和深入思考,图灵预言,总有一天计算机可通过编程获得能与人类竞争的智能。1950年10月,图灵发表了题为《机器能思考吗?》的论文,在计算机科学界引起了巨大震动,为人工智能学的创立奠定了基础。同年,图灵花费4万英镑、用了约800个电子管的ACE样机研制成功,ACE被认为是当时世界上速度最快、功能最强的计算机之一。图灵还设计了著名的“模仿游戏试验”,后人称之为“图灵测试”。该实验把被提问的一个人和一台计算机分别隔离在两间屋子,让提问者用人和计算机都能接受的方式来进行问题测试。如果提问者分不清回答者是人还是机器,那就证明计算机已具备人的智能。

现代计算机之父冯·诺依曼生前曾评价说:如果不考虑巴贝奇等人早先提出的有关思想,现代计算机的概念当属于阿兰·图灵。冯·诺依曼能把“计算机之父”的桂冠戴到自己小10岁的图灵头上,足见图灵对计算机科学影响之巨大。为了纪念图灵在计算机学科的开创性贡献,计算机领域的最高奖命名为图灵奖。

### 1.1.2 计算机的发展

计算在本质上就是信息处理。人类对信息处理的需求自古就存在,最早的结绳计数和古老的算盘都可以认为是计算的具体形式之一。但是,现代意义上的信息处理,主要是指基于电子计算机的信息处理,是开始于20世纪40年代,基于第三次工业革命,即电气革命的技术和物理成就,在军事、科学计算等领域的需求推动下发展起来的。它大致上可以概括为这样三个趋势:

1) 从人动计算迈向机动计算——追求更快的计算。从20世纪40年代计算机诞生之初到80年代末,计算机界的主流工作是如何设计制造更高性能的计算机,以拓展计算机的应用范围,使机器可以代替人,让以人为主的工作变成以机器为主的工作,典型的代表就是军事弹道轨道计算、科学研究、财务处理、办公文字处理、计算机辅助设计制造(CAD/CAM)等。

2) 从科学计算迈向智能计算——追求最好的计算。事实上,对智能计算的期待,即让计算机能够像人一样思考和工作并替代人是计算机科学发展的基本目标,但是,限于这一问题的难度,特别是早期计算机无法在性能上提供有效的支持,真正与智能相关的工程实践主要是萌芽于20世纪70年代,并在20世纪80~90年代达到第一个小高峰。这一阶段的主要工作以人工智能和专家系统、神经网络、模糊计算、遗传与进化计算、统计学习、复杂自适应系统、自然语言处理、图形处理和模式识别等为典型代表,它们的成果部分地应用在高级工业过程控制、监控、故障监测与诊断、语音识别和自动输入等领域,解决了传统方法不能解决的一大批问题。许多方面的工作在今天依然是研究的热点。

3) 从集中计算迈向普适计算——计算无处不在。在学术界集中大部分精力处理智能问题的时候,工业界也在为降低计算机的成本、提高计算机的性能做着长期的努力。特别是自20世纪90年代以来,计算机的性能已经可以满足许多领域的需求,计算机核心芯片的生产

成本也已经降低到可被许多系统采用之后,计算机应用的范围也进一步扩大,计算机系统本身也从传统的温室般的机房走向恶劣的应用现场,与物理环境的融合趋势更加明显。这一趋势从而引导和推动了嵌入式系统的发展,例如,仅 ARM 公司的 ARM7 类芯片,在全球就运行于 60 多亿个设备中。

导致出现这一趋势的另外一个重要原因是网络的迅速普及,而大量联网的设备比少数孤立设备在很多应用中更能发挥作用。所以,今天的重要趋势就是计算无处不在,分布化,网络化,嵌入化,而这一技术趋势对传统的一些计算理论和方法也提出了新的挑战。

计算机的历史沿革如图 1.2 所示。

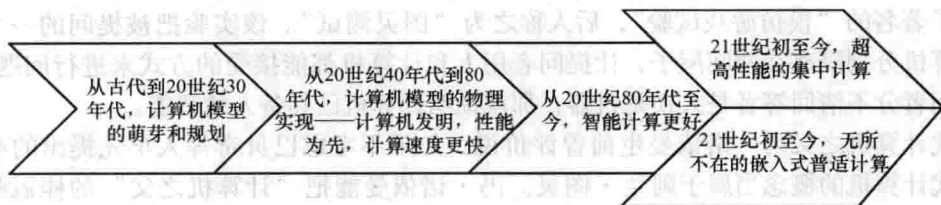


图 1.2 计算机的历史沿革

## 1.2 计算机的体系结构

### 1.2.1 冯·诺依曼架构模型

事实上,图灵机模型已经包含了如何设计并实现一台计算机的基本思路,图灵机包含三个基本的组成模块,分别是纸带、读写头和控制电路,它们反映到计算机设计中,分别就是存储器、运算器和控制器。冯·诺依曼意识到这一点,进一步扩展了输入设备和输出设备,并在莫奇利建造的 ENIAC 基础上,对计算机组织结构进一步规范化,总结出了指导计算机设计的早期的冯·诺依曼计算机架构,如图 1.3 所示。

在冯·诺依曼架构模型中,完整的计算机系统被认为应包含这样五部分:存储器,运算器,控制器,输入设备和输出设备。其中,运算器作为计算环节需要处理好操作数的输入(从哪里来)和输出(到哪里去)问题,因此自然地被作为整个系统的中心。但是,这种架构很快就暴露出其弱点,就是运算器的数据吞吐能力十分有限,会成为系统的瓶颈,因此很快演化为以存储器为中心的改进型冯·诺依曼架构,如图 1.4 所示。这样在各个模块的高速数据交换中心就可以利用存储器这个大容量中介,极大地提高了效率。

冯·诺依曼架构的价值在于它首次规范了计算机系统的具体设计技术,回答了“应如何构建一台计算机”的问题。在冯式模型诞生之前,历史上也曾出现过许多个具有计算能

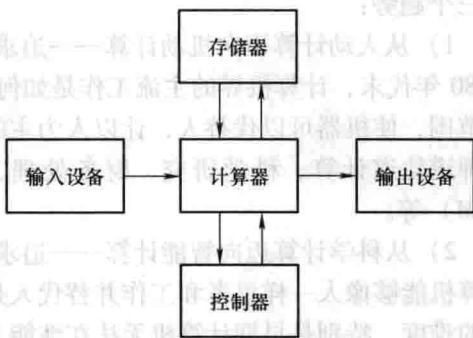
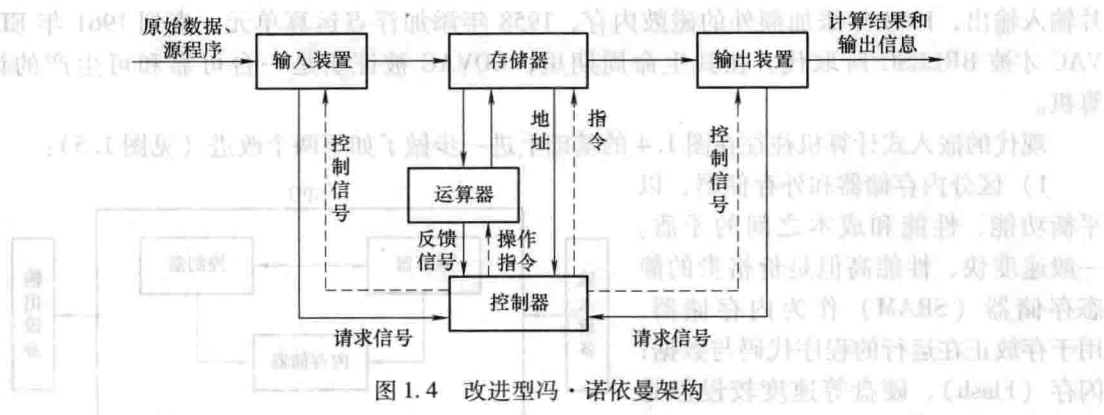


图 1.3 早期的冯·诺依曼架构



力的设备，从中国早期的算盘，到欧洲的水力计算机，达芬奇的计算机，但是所有这些更多的是依赖设计者本人的巧妙构思，并未上升到通用的层次。冯·诺依曼模型清楚地说明：只要分别设计存储器、运算器、输入设备、输出设备和控制器五大部件，然后把它们连接到一起，就组成了计算机。至于这些部件采用何种方式实现，使用人力驱动、水力驱动还是电力驱动，是采用原始的石头摆放、穿孔卡片存储、磁记录方式存储还是触发器电路存储都没有关系。诚然，电子技术的发展在竞争中提供了最有力、最方便的实现手段，并在计算机发展中成为主流，直到今天。

冯·诺依曼清楚地意识到 ENIAC 的设计不足，并加入到 EDVAC 的设计群体中。1945 年 6 月，他在内部发布了 EDVAC 设计初稿《关于 EDVAC 的报告草案》(First Draft of a Report the EDVAC)，报告提出的体系结构一直延续至今，即冯·诺依曼架构。长达 101 页的 EDVAC 最终版设计方案明确指出了新机器有五个构成部分，即计算机 CA、逻辑控制装置 CC、存储器 M、输入 I、输出 O，并描述了这五个部分的职能和相互关系。这份报告也因此成为一份划时代的文献，它奠定了现代计算机的设计基础，直接推动了 20 世纪 40 年代末数十种早期计算机的诞生。EDVAC 方案有两个非常重大的改进：一是为了充分发挥电子元件的高速度而采用了二进制；二是提供了“存储程序”，可以自动地从一个程序指令进入到下一个程序指令，其作业顺序可以通过一种称为“条件转移”的指令而自动完成。“指令”包括数据和程序，把它们用码的形式输入到机器的记忆装置中，即用记忆数据的同一记忆装置存储执行运算的命令，这就是所谓存储程序的新概念。这个概念也被誉为计算机史上的一个里程碑。EDVAC 的发明才是真正为现代计算机在体系结构和工作原理上奠定了基础。

EDVAC 于 1949 年 8 月交付给弹道研究实验室，它使用了大约 6000 个电子真空管和 12000 个电子二极管，占地 45.5m<sup>2</sup>，重达 7850kg，消耗电力 56kW，具有加、减、乘和除的功能。整个系统包括一个使用汞延迟线容量为 1000 个字的存储器（每个字 44bit），一个磁带记录仪，一个连接示波器的控制单元，一个分发单元、用于从控制器和内存接收指令并分发到其他单元，一个运算单元及一个定时器。

在发现和解决许多问题之后，EDVAC 直到 1951 年才开始运行，而且局限于基本功能。延迟的原因是莫奇利和艾克特从宾夕法尼亚大学离职并带走了大部分高级工程师，开始组建莫奇利-艾克特电子计算机公司，由此与宾夕法尼亚大学产生了专利纠纷。到 1960 年，EDVAC 每天运行超过 20h，平均 8h 无差错时间。EDVAC 的硬件不断升级，1953 年添加穿孔卡



片输入输出, 1954 年添加额外的磁鼓内存, 1958 年添加浮点运算单元, 直到 1961 年 EDVAC 才被 BRLESC 所取代。在其生命周期里, EDVAC 被证明是一台可靠和可生产的计算机。

现代的嵌入式计算机往往在图 1.4 的基础上进一步做了如下两个改进 (见图 1.5):

1) 区分内存储器 and 外存储器, 以平衡功能、性能和成本之间的矛盾。一般速度快、性能高但是价格贵的静态存储器 (SRAM) 作为内存储器, 用于存放正在运行的程序代码与数据; 闪存 (Flash)、硬盘等速度较慢但是单位存储成本较低的器件作为外存储器, 用于脱机断电期间提供程序和数据存储。这种存储层次在嵌入式系统中经常体现为高速 SRAM 和大容量 Flash 的区别。

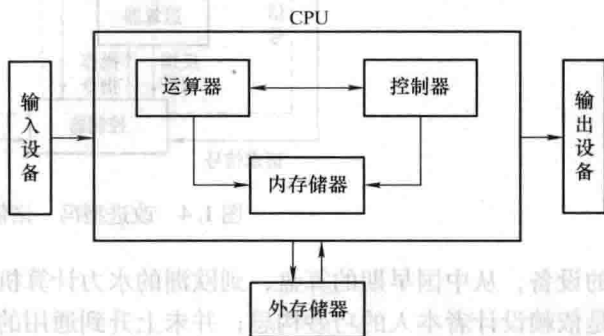


图 1.5 冯·诺依曼架构的扩展

2) 区分指令存储器和数据存储器, 并分别设置指令总线和数据总线进行存取。这样可以进一步提高 CPU 访问的性能, 这种体系结构被称为哈佛架构。这一设计在高性能芯片如 TI 和 ADI 公司的各种数字信号处理芯片中广泛存在; 而在低成本微控制器应用中, 出于降低成本和复杂度的需要, 大多只提供一条总线通向存储器。一个折中的方案是, 总线仍然只是一条, 但是允许程序代码和数据分开存储在不同的存储器区域中, 这样就可以根据不同存储器的性能来分配指令存储器和数据存储器从而达到较优的性能。ARM 和 Cortex 都支持存储器重映射以提供上述功能。

图 1.6 所示是冯·诺依曼架构和哈佛架构的比较。

与阿塔纳索夫等设计 ABC、莫奇利等设计 ENIAC、冯·诺依曼设计 EDVAC 同时期, 世界上其他大学和科研机构也纷纷展开这方面的工作。德国的许莱尔、朱斯合作, 计划制造一台有 1500 个电子管、每秒能运行 10000 次

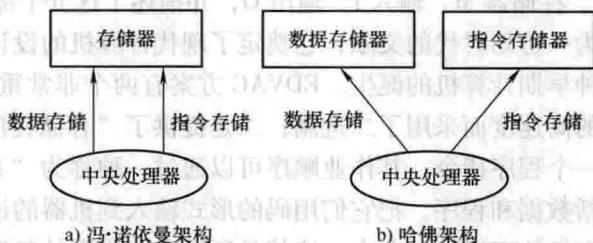


图 1.6 冯·诺依曼架构和哈佛架构的比较

的通用机, 这台机器的运算部件于 1942 年完成, 但整个计划由于遭到政府的拒绝而夭折。图灵在二战期间曾参与英国军方破译德国密码的工作, 并在战争结束后于 1945 年 2 月向英国国家物理实验室 (NPL) 执行委员会提交了一份详细文档, 给出了存储程序式计算机的第一份完全可行性设计。但是, 由于图灵和他最初的工程师朋友都已签署了保密协议, 图灵在 NPL 的同事不了解图灵先期工作的成果, 认为建造完整 ACE 的工作量太大, 无法完成。在图灵离开 NPL 后, 威尔金森接受整个项目, 建造了 ACE 的一个简化版本, 也是第一台 ACE 的实现——Pilot ACE, 并于 1950 年 5 月 10 日运行了第一个程序。它比图灵先前设计的规模要小, 使用了大约 80 个电子管, 存储器是汞延迟线, 有 12 个汞延迟线, 每个包含 32 条 32 位元的指令或数据, 时钟频率为 1MHz, 这在当时的电子计算机中是最快的。但由于 Pilot