

英 汉 双 解

# 嵌入式系统词典

Embedded Systems Dictionary

[美] Jack Ganssle Michael Barr 著

马广云 潘琢金 彭甫阳 译

CMPBooks



北京航空航天大学出版社

英汉双解  
嵌入式系统词典  
Embedded Systems Dictionary

[美] Jack Ganssle 著  
Michael Barr

马广云 潘琢金 彭甫阳 译

北京航空航天大学出版社

## 内 容 简 介

本书是一本非常实用的嵌入式技术词典,为美国 CMP Books 出版的 *Embedded Systems Dictionary* 的中译本。书中保留了原书英文内容,即一书中、英文两版。全书收录了当今嵌入式系统领域的最新词汇近 3 000 多条,内容涉及硬件、软件、Internet、数据存储、信息处理、无线通信等领域,以及各种组织机构和标准,反映了嵌入式系统领域的最新进展。作者用通俗的语言解释专业性词汇,可让读者快速领悟与掌握嵌入式系统领域的专业术语及行话。

本词典词汇量大,覆盖面广,编选科学,实用性强,是一本嵌入式系统领域的专业技术人员、高等院校师生、业余爱好者、投资者、管理人员、市场营销人员和翻译人员等的必备工具书。

### 图书在版编目(CIP)数据

英汉双解嵌入式系统词典/(美)甘斯勒(Ganssle, J.), (美)博尔(Barr, M.)编著;马广云等译.  
北京:北京航空航天大学出版社,2006.1

书名原文: Embedded Systems Dictionary

ISBN 7-81077-359-3

I. 英… II. ①甘…②博…③马… III. 微型计算机—系统设计—词典—英、汉 IV. TP360.21-61

中国版本图书馆 CIP 数据核字(2006)第 004522 号

本书英文版原名: Embedded Systems Dictionary

Copyright © 2003 by CMP Books, except where noted otherwise.

Published by CMP Books, CMP Media LLC.

600 Harrison Street

San Francisco, CA 94107, USA

All rights reserved.

本书中文简体字版由美国 CMP Books 公司授权北京航空航天大学出版社在中华人民共和国境内(不包括香港特别行政区)独家出版发行。版权所有。

北京市版权局著作权登记号:图字:01-2003-4690

### 英汉双解嵌入式系统词典 Embedded Systems Dictionary

[美] Jack Ganssle 著  
Michael Barr

马广云 潘琢金 彭甫阳 译

责任编辑 刘瑞月

\*

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(100083) 发行部电话:010-82317024 传真:010-82328026

http://www.buaapress.com.cn E-mail: bhpress@263.net

涿州市新华印刷有限公司印装 各地书店经销

开本: 787×960 1/16 印张: 36 字数: 1 083 千字

2006 年 1 月第 1 版 2006 年 1 月第 1 次印刷 印数: 5 000 册

ISBN 7-81077-359-3 定价: 59.00 元

# 译者序

人类已开始全面步入信息化社会,嵌入式系统作为信息化建设的重要技术支撑,呈现出前所未有的强劲发展势头。当前,嵌入式技术的发展日新月异,其应用已渗透到人类社会与生活的各个领域,对人们日常生活、经济建设乃至军事领域等产生了重大而深远的影响。

伴随着嵌入式系统软件、硬件、网络、系统集成、信息处理、无线通信等技术的不断发展,随之产生出许多新知识、新概念和新名词术语;同时,一些过时的词汇逐渐被淘汰,还有一些词汇被赋予了新的含义。

当人们在阅读中英文专业图书、文献和资料,浏览网上信息,以及查看软件联机帮助信息时,常常会对那些陌生的名词、术语、缩略语和“行话”困惑不解。因此,有一本内容新、实用性强、涵盖嵌入式技术领域的词典摆在案头,应是许多人心中迫切的愿望。

## 关于本书

CMP Books 出版的 *Embedded Systems Dictionary* 正是这样一本能反映当今嵌入式技术和概念的实用词典。对需要理解嵌入式专业术语和行话的人们来说,这是一本不可或缺的指南。这本权威著作解释了嵌入式系统领域近 3 000 条最常用术语的含义及用法,适合专业和非专业的读者。每个词条的主要释义简明陈述了该词条的含义,以满足专业知识较少的读者需要。每个释义的扩展内容十分丰富,包括:发音,缩写词,可替代术语,交叉引用,示例,示意图、电路原理图和图片,公式,程序清单,以及进一步阅读的提示。

鉴于此,我们觉得非常有必要将此书译成中文版介绍给国内广大读者。

考虑到国内众多专业与非专业读者的需要,在筹备此书的翻译过程中,译者与出版社达成共识:将此书做成英汉双解版本,在将英文译成中文的同时,也将原文呈献给读者。在翻译过程中,我们力求中译文准确、科学和规范,但由于嵌入式系统及应用涉及诸多技术领域,加之译者的学识和水平所限,会有译得不准确甚至错译的地方。采用英汉双解形式出版,对于译得不妥之处,读者则可对照原文来获得正确信息,并可告知译者和出版社进行修正。这样可做到真正对读者负责。

对于广大读者来说,该译本是一书两版:对于英语较好的读者,这是一本原版书;对于与嵌入式系统有关的读者,这又是一本很好的专业英汉对照读物;同时,对于嵌入式领域的专业读者,除了作为词典查阅之外,还是一本值得“通读”的专业书籍,对于拓宽与完善相关的专业知识大有裨益。

但对于译者来说,以双解形式出版外文版图书需要一定的勇气,因为这会将译者的专业知识和翻译水平毫无保留地暴露在读者面前,也对译者的工作提出了更高的要求。

翻译词典是一项极为艰苦细致的工作和挑战。嵌入式技术飞速发展,涉及众多技术领域,对于同一英文术语,可能会对应有多种中文叫法;有些英文新术语很难理解;不少术语至今还没有一个为大家所接受的中文译名,要为一些英文新、难术语赋予准确、恰当的中文译名却是颇费功夫的。就译者而言,要做到准确、科学和规范地翻译,实在不是一件容易的事情,译者常常为一个术语的译名而翻阅很多辞书与相关专业书籍,反复推敲和印证。

对于大部分词汇,原则上按照国家自然科学技术名词审定委员会推荐的标准名称赋予其中文译名;对于尚未见诸于标准的新术语之中文定名,若其中文名称已出现在科学杂志和图书中,则参考定名;若未曾在任何中文资

料中出现,则按译者的理解,在多方征求有关专家意见的基础上,尽量给出比较恰当和专业的译法;还有一些表示专有产品或体系结构的名词,则保留原名,不予译出。所有词条的中文定名都经过了多位专家的仔细审订。

### 几点编排说明

本书原英文版中,每个词条都有释文。词条释文分一个释义和多个释义2种情况。释文放在词条之后,有多个释义的释文则按顺序编号。

本书在英文词条和释文后增加中译文后,对于只有一个释义的词条,编排格式比较简单,即:

- 英文词条的中文译名放在其右边,用空格分开。
- 英文词条对应有意义相近、可互相替换使用的不同中文译名时,则用逗号“,”分开。
- 词条中文译名含圆括号表示圆括号中的内容可以取舍。例如,“相移键控(技术,法)”既可表示“相移键控”,也可表示“相移键控技术”和“相移键控法”。
- 词条中文译名含方括号表示方括号中的内容可以替换方括号左边的内容。例如,“宏汇编器[程序]”既可为“宏汇编器”,又可为“宏汇编程序”。“互联网[因特网]校验和”既可为“互联网校验和”,又可为“因特网校验和”。
- 词条的中文释文紧跟其英文释文。

但对于具有多个释义的词条,同一英文词条的不同释义对应的中文译名有的意义相同,有的意义相差较远。为便于阅读,本书编排时根据不同情况作了如下处理:

(1) 若同一英文词条的不同释义所对应的中文译名意义相同,则中文译名的格式与只有一个释义的情况相同。不同释义的释文仍按原英文顺序编号放在词条后面。每个释义的中文释文紧跟其英文释文。例如,词条 **kernel** 的编排格式为:

#### **kernel** 内核

1. *n.* A minimalist operating system.  
最小限度的操作系统。
2. *n.* The core of a microkernel architecture operating system like Mach.  
微内核架构操作系统(如 Mach)的核心。
3. *n.* An essential part of any real-time operating system, the kernel consists of the scheduler and context switch routine.  
任何实时操作系统的基本部分,内核包括任务调度程序和上下文切换例程。

(2) 若同一英文词条的不同释义对应的中文译名意义相差较远,且不可互相替换,则依据不同的中文译名,将该词条按释义拆分成多个词条。将原英文释文的顺序编号改放在英文词条的右上角。例如:词条 **junction** 的原英文版编排格式为:

#### **junction**

1. *n.* The interface between an N- and a P-type of silicon on a diode, transistor, or other semiconductor.
2. *n.* The connection between two nodes on a schematic or in an electrical circuit.

本书中将其拆分成2个词条,编排格式为:

#### **junction<sup>1</sup>** (PN)结

- n.* The interface between an N- and a P-type of silicon on a diode, transistor, or other semicon-

ductor.

二极管、晶体管或其他半导体器件中, N型硅与P型硅之间的连接面。

### junction<sup>2</sup> 结点

*n.* The connection between two nodes on a schematic or in an electrical circuit.

原理图或电气线路中两个节点的连接。

又例如: 词条 package 的原英文版编排格式为:

### package

1. *n.* The plastic, metal, or ceramic container that encloses an IC. Packages protect the chip and bring the leads from the outside world to the chip.

2. *n.* In Java, a collection of classes.

3. *n.* In UML, a model element that contains other model elements.

词条 package 的第2和第3个释义对应的词条中文译名相同, 但与第1个释义的不同; 故本书也将其拆分成2个词条, 编排格式为:

### package<sup>1</sup> 封装

*n.* The plastic, metal, or ceramic container that encloses an IC. Packages protect the chip and bring the leads from the outside world to the chip.

封住 IC 所用的塑料、金属或陶瓷包装。封装可以保护芯片, 并使芯片与外部世界连接。

### package<sup>2,3</sup> 包

2. *n.* In Java, a collection of classes.

在 Java 语言中, 类的集合。

3. *n.* In UML, a model element that contains other model elements.

在 UML 中, 一种包含其他模型元素的模型元素。

## 致 谢

本书内容涉及领域众多, 译者除了马广云、潘琢金和彭甫阳之外, 还有许多来自不同专业领域的其他专家参与了本书的翻译工作, 他们是: 阎永红、骆丽、田泽、闫晓莺、车晓萍、万永波、杨峰、吴双力、张伯岭、张烁, 以及凌阳公司的黄鹤九、王立平、潘秀利、李静、何春娟, 周立功公司的肖典宇; 阎宝明和王瑛担任了本书的审校工作, 逐字逐句地审阅了本书的全部内容; 何立民教授、张炯博士、沈建华博士、陈渝博士和马忠梅老师对译者的一些疑惑问题给予了细致的解答, 并提出了许多宝贵的修改意见; 黄莹为本书一些新难术语进行了网上查询工作, 在此我们表示衷心的感谢。

由于时间紧迫和译者水平有限, 书中译法难免有不妥之处, 我们愿虚心接受读者和专家的批评指正。恳切希望读者能将您的意见和发现的翻译不妥之处转达给我们, 以便在本书重印时及时更正。译者的联系方式如下:

电话/传真: 010-82317022

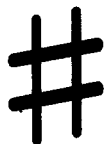
EMAIL: mgy@263.com.cn

译 者

2005 年 12 月

# 目 录

#	1
A	13
B	45
C	75
D	126
E	166
F	188
G	216
H	231
I	250
J	280
K	287
L	293
M	315
N	355
O	372
P	388
Q	430
R	433
S	461
T	502
U	524
V	532
W	540
X	548
Y	550
Z	552
附录	553
作者简介	555



## # # 符号,井号

(pound) 1. *n.* A spare key on a phone's tiny keypad. In that capacity, often used as a data delimiter ("Enter your account number followed by the pound sign."). *See also* dual-tone multiple frequency.

电话小键盘上的一个备用键,常用做数据定界符(如“输入您的账号,以“#”号键结束”)。参阅 dual-tone multiple frequency.

2. *n.* An abbreviation for the English-system unit of weight that's also abbreviated "lb".

英制重量单位“磅”的缩略语,亦可缩略为“lb”。

## #define #define 定义语句

(pound define) *res.* In C and C++, a preprocessor command used to define constants and macros.

C 和 C++ 语言中的预处理器命令,用于定义常量和宏。

## #include #include 包含语句

(pound include) *res.* In C and C++, a preprocessor command that incorporates the named file that follows at the current location of the current source file, prior to compilation of the current file.

C 和 C++ 语言中的预处理器命令。它指示预处理器在当前文件被编译之前,将所指定的文件插入当前源文件的当前位置。

## #pragma #pragma 编译指示语句

(pragma) *res.* In C and C++, a preprocessor command used to invoke compiler-specific functionality.

C 和 C++ 语言中的预处理器命令,用于调用编译器的特定功能。

## ! 逻辑非运算符

(not) *res.* The logical-NOT operator in C and various related languages. In C and C++, !foo is 0 if foo is nonzero; 1 if foo is 0.

C 和各种相关语言中的逻辑非运算符。在 C 和 C++ 语言中,如果 foo 为非零,则!foo 为零;如果 foo 为零,则!foo 为 1。

## ^ 按位“异或”运算符

(ex or) *res.* The bitwise-XOR operator in C and various related languages.

C 和各种相关语言中的按位“异或”运算符。

## | 按位“或”运算符

(or) *res.* The bitwise-OR operator in C and various related languages. The | operator is easily confused with the || operator. However, their behavior is in fact very different. 0x55 | 0x0F performs a binary OR resulting in 0x5F. 0x55 || 0x0F performs a logical OR that says true (nonzero) OR true, therefore true. Beware that the compiler won't warn you if you choose the wrong one.



#

C和各种相关语言中的按位“或”运算符。该运算符易与逻辑“或”运算符“||”相混淆。但两者的行为实际上相差甚远。“0x55 | 0x0F”执行二进制“或”运算,结果为0x5F。“0x55 || 0x0F”执行逻辑“或”运算,即“真”(非零)与“真”相“或”,结果为“真”。注意:如果选用了错误的运算符,编译器不会发出警告。

## || 逻辑“或”运算符

(or) *res.* The logical-OR operator in C and various related languages. Often confused with |.

C和各种相关语言中的逻辑“或”运算符,常易与按位“或”运算符“|”相混淆。

## ~<sup>1</sup> 按位“反”运算符

(not) *res.* The unary bitwise-NOT operator in C and various related languages. Sometimes confused with the logical NOT operator !, though their functions are very different. Make sure you use the right one; your compiler won't know to warn you of the error.

**Example:** If 16-bit  $x$  is 0x0123,  $\sim x$  is 0xFEDC;  $!x$  is 0 (Boolean false).

C和各种相关语言中的单目按位“反”运算符。有时易与逻辑非运算符“!”相混淆,尽管两者的功能相差甚远。应确认使用了正确的运算符。如果选用了错误的运算符,编译器不会发出警告。

**示例:**如果16位数 $x$ 的值为0x0123,则 $\sim x$ 的值为0xFEDC,而 $!x$ 的值为0(布尔“假”)。

## ~<sup>2</sup> 前缀符号

*symbol.* A prefix used on schematic diagrams and in logic equations to indicate active low signals.

一种前缀符号,用于原理图和逻辑方程中,表示信号低电平有效。

## %<sup>1</sup> 取余运算符

(mod) *res.* The modular division operator in C, C++, and Java. The result of  $a \% b$ , where  $a$  and  $b$  are both integers, is the integer remainder of  $a$  divided by  $b$ . The result is, by definition, between 0 and  $b-1$ .

C、C++和Java语言中的取余运算符。两个整数 $a$ 和 $b$ 的取余运算为 $a \% b$ ,其结果是 $a$ 除以 $b$ 后的整数余数。根据此定义,结果应在 $0 \sim b-1$ 之间。

## %<sup>2</sup> 百分号

*n.* Percentage.

百分号。

## &<sup>1</sup> 按位“与”运算符

(and) *res.* The bitwise-AND operator in C and various related languages. The & operator is easily confused with the && operator. However, their behavior is in fact very different.  $0x55 \& 0x0F$  performs a bitwise-AND resulting in 0x05.  $0x55 \&\& 0x0F$  performs a logical-AND resulting in true, since both are non-zero. Beware; the compiler won't warn you if you choose the wrong one.

C和各种相关语言中的按位“与”运算符。运算符“&”易与运算符“&&”相混淆。但两者的行为实际上相差甚远。“ $0x55 \& 0x0F$ ”执行按位“与”运算,结果为0x05;而“ $0x55 \&\& 0x0F$ ”执行逻辑“与”运算,由于两个操作数均为非零,结果为“真”。注意,如果选用了错误的运算符,编译器不会发出警告。

## &<sup>2</sup> 取地址运算符

*res.* In another context, the unary address-of operator in C and C++. The & is interpreted as an address-of operator when it precedes the name of a variable in a statement like  $p = \&a$  or  $\&b +$

sizeof(b).

在另一场合,符号“&”为 C 和 C++ 语言中取操作数地址的单目运算符。如果在一个语句中,符号“&.”位于变量名的前面,例如“p=&a”或“&b + sizeof(b)”,则“&.”为取地址运算符。

### &<sup>3</sup> 引用类型声明符

*res.* In the declaration of a variable, an & preceding a variable's name tags it a reference type in C++.

在 C++ 语言的变量声明语句中,符号“&.”位于变量名前面则表明该变量为引用类型。

### && 逻辑“与”运算符

(and) *res.* The logical-AND operator in C and various related languages. Often confused with &.

C 和各种相关语言中的逻辑“与”运算符,常易与“&.”相混淆。

### > 大于(符号)

(greater-than) *res.* The greater than operator in C and various related languages. If *a* is greater than *b*, the test *a > b* is true.

C 和各种相关语言中的“大于”运算符。如果 *a* 大于 *b*,则测试 *a > b* 的值为“真”。

### >> 右移运算符

(right shift) *res.* The right-shift operator in C and various related languages. Shifting an integer right by *n* bits is equivalent to dividing its value by 2<sup>*n*</sup>. Any bits shifted out the right end are discarded. In C and C++, the bits shifted in from the left depend on the type of the operand; if it is unsigned, 0s are always shifted in from the left; if it is signed, however, it is up to the implementer (of the compiler) to decide whether the sign bit in the MSb position is duplicated or 0s are shifted in. In Java, this situation was corrected, and the >> op-

erator always handles signed operands portably; a second >>> operator is used to explicitly ignore the sign bit during right shifts.

**Example:** If variable `uint8_t x` contains the value 01010101b, the result of `x >> 1` will be 00101010b.

C 和各种相关语言中的右移运算符。将一个整数右移 *n* 位,等效于将该整数除以 2<sup>*n*</sup>。右端移出的任何位都会被丢弃。在 C 和 C++ 语言中,从左端移入的位取决于操作数的类型;如果操作数为无符号数,则左端总是移入 0;如果操作数为带符号数,那么是复制最高有效位上的符号位还是移入 0,取决于(编译器的)实现者。在 Java 语言中,这种情形得以纠正,运算符“>>”总是处理带符号操作数,便于移植。另一个运算符“>>>”用于右移时明确地忽略符号位。

**示例:** 如果变量 `uint8_t x` 的值为 01010101b,则 `x >> 1` 的结果为 00101010b。

### >>> 无符号右移运算符

(unsigned right shift) *res.* Right-shift with zero extension. In Java, there are separate signed and unsigned right-shift operators. This is the unsigned one, which does not copy the sign bit into the uppermost bit as the pattern is shifted to the right. Instead, 0s are shifted in from that end. *See also* >>.

零扩展右移。在 Java 语言中,分别有无符号右移和带符号右移两个运算符。“>>>”为无符号右移运算符。当右移时,并不将符号位复制到最高位,而是从左端移入 0。参阅 >>。

### < 小于(符号)

(less than) *res.* The less-than operator in C and various related languages. If *a* is less than *b*, the test *a < b* is true.

C 和各种相关语言中的“小于”运算符。如果 *a* 小于 *b*,则测试 *a < b* 的值为“真”。

### << 左移运算符

(left shift) *res.* The left-shift operator in C and vari-

#



ous related languages. Shifting an integer left by  $n$  bits is equivalent to multiplying its value by  $2^n$ . However, be forewarned that it could result in an overflow. Any bits shifted out the left end are discarded; 0s are shifted in from the right.

C和各种相关语言中的左移运算符。将一整数左移  $n$  位等效于将该整数乘以  $2^n$ 。但须注意,该运算可能会导致溢出。从左端移出的任何位均会被丢弃,并从右端移入 0。

### $\mu\text{A}$ 微安, $10^{-6}$ 安(培)

(microamp) *abbr.* See(见)microamp.

### $\mu\text{C}/\text{OS}$ $\mu\text{C}/\text{OS}$ 实时操作系统

(micro-kos) *N.* A real-time operating system for small microcontrollers. Derived from a two-part expository article in *Embedded Systems Programming* in May-June 1992, this little RTOS took on a life of its own with the publication of a book-length treatment of the subject by creator Jean Labrosse. Now in the operating system's second release ( $\mu\text{C}/\text{OS-II}$ ), the revised book includes complete source code and explanations of everything it does.

用于小型微控制器的实时操作系统。 $\mu\text{C}/\text{OS}$  实时操作系统源于 *Embedded Systems Programming* 杂志在 1992 年 5 月和 6 月分两部分发表的一篇文章。随着其创建者 Jean Labrosse 关于该主题长篇论文的发表,这一小小的实时操作系统开始流行。现在该操作系统的第二版( $\mu\text{C}/\text{OS-II}$ )已经发布,修订后的书中包括了完整的源代码和详细的说明。

### $\mu\text{Clinux}$ $\mu\text{Clinux}$ 操作系统

See(见)uClinux.

### $\mu\text{F}$ 微法, $10^{-6}$ 法(拉)

(microfarad) *abbr.* See(见)microfarad,  $10^{-6}$  法(拉)

### $\mu\text{s}$ 微秒, $10^{-6}$ 秒

(microsecond) *abbr.* See(见)microsecond.

### $\mu\text{V}$ 微伏, $10^{-6}$ 伏(特)

(microvolt) *abbr.* See(见)microvolt.

### $\pi$ 圆周率

(pie) *n.* A mathematical constant representing the relationship between a circle's diameter and circumference. Approximately equal to 3.141 592 653 589.

**Usage:** Programmers often overspecify  $\pi$  when defining it in their programs. Even with double-precision data types, specifying the value to more than 16 decimal digits is a waste of effort. The following listing shows an appropriate C/C++ declaration.

This declaration for  $\pi$  is precise enough for most applications.

```
const double pi = 3.1415926;
```

一个表示圆的直径与周长关系的数学常数,约等于 3.141 592 653 589。

**用法:** 程序员经常在其程序中过于精确地定义圆周率。即使使用双精度数据类型,指定圆周率的值超过 16 个小数位也是徒劳。以下代码是一个合适的 C/C++ 声明。

圆周率  $\pi$  的如下声明对于绝大多数应用来说已足够精确

```
const double pi = 3.1415926;
```

### \* 星号

1. *res.* In C and various related languages, an operator used to declare a variable or parameter of pointer type.

**Example:** To declare a pointer  $p$  to integer  $x$ :

```
int *p = &x;
```

C和各种相关语言中,用于声明指针型变量或参数的运算符。

**示例:** 声明一个指针  $p$ , 指向整型变量  $x$ 。

```
int *p = &x;
```

2. *res.* In C and various related languages, an operator used to dereference a pointer.

**Example:** If pointer *p* points to integer *x*, you can read and write *x* via *\*p*.

C和各种相关语言中的一个运算符,用于访问一个指针所指向的地址。

**示例:** 如果指针 *p* 指向整型变量 *x*,则可通过 *\*p* 来读/写 *x*。

3. *n.* A spare key on a phone's tiny keypad. As in, "To disable call waiting, dial \* 70, then the number." *See also* dual-tone multiple frequency.

电话小键盘上的一个备用键。例如,“要取消呼叫等待,按 \* 70,然后拨号码”。参阅 dual-tone multiple frequency。

## ++ 增量运算符

(plus plus) *res.* The increment operator in C and various related languages. *See also* preincrement, postincrement.

C和各种相关语言中的增量运算符。参阅 preincrement, postincrement。

## -- 减量运算符

(minus minus) *res.* The decrement operator in C and various related languages. *See also* predecrement, postdecrement.

C和各种相关语言中的减量运算符。参阅 predecrement, postdecrement。

## .bss 大容量存储段

*See*(见) BSS segment.

## .data 数据段

*See*(见) data segment.

## .HEX Intel Hex 格式

*See*(见) Intel Hex format.

## .text 正文段,代码段

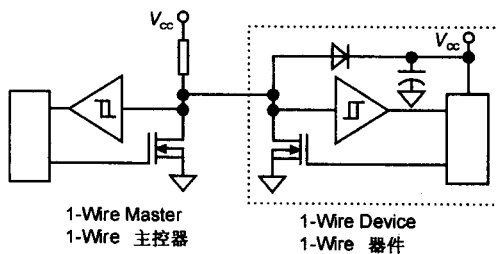
*See*(见) text segment.

## 1-Wire 单总线,1-Wire 总线

*N.* A low-cost bidirectional serial communications bus requiring just a single wire to implement. Developed by Dallas Semiconductor, 1-Wire is an asynchronous, half-duplex, single-master component bus. 1-Wire minimizes device pin count by using extra on-chip circuitry to enable each 1-Wire component to communicate serially (over a single shared connection) with other 1-Wire devices. Compatible serial EEPROMs, A/D converters, temperature sensors, and other compatible devices are available. Electrical features include low data rates (about 16 kbps, max), flexible voltage and timing requirements, long maximum distances (up to 1 000 feet), and optional parasitic powering of slaves. Each 1-Wire device has a unique 48-bit serial number, so multiple slaves can share the same bus. *See also* I<sup>2</sup>C, SPI.

**History:** The 1-Wire protocol was invented by Dallas Semiconductor, which holds a related patent.

**Further Reading:** Willey, H. Michael. "One Cheap Network Topology," *Embedded Systems Programming*, January 2001.



A representation of the circuitry used to connect 1-Wire devices. 用于连接 1-Wire 器件的电路示意图

一种低成本的双向串行通信总线,只需一根线即可实现。1-Wire 总线由 Dallas 半导体公司开发,为

#

异步、半双工、单主器件总线。1-Wire 总线能使器件引脚数量最少,这是通过利用附加的片上电路,来使每个 1-Wire 器件能够与其他 1-Wire 器件(经由单个共享连接)进行串行通信实现的。可用的 1-Wire 器件包括兼容的串行 EEPROM、A/D 转换器、温度传感器及其他兼容器件。其电气特性包括较低的数据传输速率(最高约为 16 kbps)、灵活的电压和时序要求、较长的最大传输距离(可达 1 000 英尺),以及可选的从器件寄生供电。每个 1-Wire 器件都有惟一的 48 位序列号,因此,多个从器件可共享同一总线。参阅 I<sup>2</sup>C, SPI。

**历史:** 1-Wire 协议由 Dallas 半导体公司发明,它还拥有一项相关的专利。

**相关阅读:** Willey, H. Michael. "One Cheap Network Topology," *Embedded Systems Programming*, January 2001.

## 100Base-T 100Base-T 接口

(hundred base tee) *abbr.* The most popular and cost-effective interface circuitry for Fast Ethernet. Short for 100 Mbps, BASEband, Twisted pair. Though competitive Fast Ethernet interfaces over fiber and four-pair cable are defined and used, 100Base-T is popular because it's based on the successful 10Base-T standard.

一种最流行且性价比高的快速以太网接口电路。100Base-T 为 100 Mbps, BASEband, Twisted pair 的缩略语。尽管使用光纤和四对线缆的快速以太网接口是其竞争对手,并已被定义和采用,但由于 100Base-T 是基于非常成功的 10Base-T 标准,因此仍然很流行。

## 10Base-T 10Base-T 接口

(ten base tee) *abbr.* An IEEE standard for sending information at 10 Mbps on ordinary unshielded twisted-pair cable. Short for 10 Mbps, BASEband, Twisted pair. The standard defines connector types (typically RJ-45), pin connections, voltage levels, and noise immunity requirements.

See also 100Base-T.

**Usage:** Often misused to mean any 10 Mbps Ethernet connection or connector. However, the standard for running Ethernet LANs over 10Base-T-compliant cables and connectors is actually distinct and part of IEEE 802.3.

在普通非屏蔽双绞线上以 10 Mbps 速率传送信息的 IEEE 标准。10Base-T 为 10 Mbps, BASEband, Twisted pair 的缩略语。该标准定义了连接器的类型(通常为 RJ-45)、引脚连接方式、电平及抗噪声要求等。参阅 100Base-T。

**用法:** 10Base-T 常误指任何 10 Mbps 的以太网连接或连接器。但实际上,在符合 10Base-T 的线缆和连接器上运行的以太网局域网的标准与 IEEE 802.3 是有区别的,10Base-T 是 IEEE 802.3 标准的一部分。

## 1488 1488 芯片

(fourteen eighty-eight) *num.* A now-obsolete driver chip for RS-232 circuits. The 1488, introduced by Motorola in the 1970s, converted TTL logic levels (0 to 5 V) to the plus/minus 15 V used by RS-232. No longer used because it required external power supplies to provide these levels; Maxim and others today produce driver chips that internally create the needed RS-232 levels.

一种 RS-232 电路的驱动芯片,现已过时。1488 芯片由 Motorola 公司在 20 世纪 70 年代推出,能将 TTL 逻辑电平(0~5 V)转化为 RS-232 所使用的 ±15 V 电平。不再使用 1488 芯片的原因是,它需要外部电源来提供 RS-232 电平。现在 Maxim 及其他一些公司生产可内部产生所需 RS-232 电平的驱动芯片。

## 1489 1489 芯片

(fourteen eighty-nine) *num.* An obsolete chip used to receive RS-232 signals. A 1970s part originally from Motorola, the 1489 was a line receiver that translated the ±15-V RS-232 levels to TTL signals.

Now replaced by combined receiver/transmitter chips such as the MAX232 from Maxim.

一种用于接收 RS-232 信号的过时芯片。1489 芯片最初由 Motorola 公司在 20 世纪 70 年代推出, 是一种线路接收器, 能将  $\pm 15\text{ V}$  的 RS-232 电平转换为 TTL 信号。该芯片现已被组合式接收/发送芯片(如 Maxim 公司的 MAX232)所取代。

## 16-bit 16 位

1. *adj.* Generally used to denote a processor that uses a data bus 2 bytes wide. Note that this always refers to the processor's data bus width; address buses are typically unrelated to CPU size. Confusing the issue more, many embedded processors have an internal data bus of one size, but an external bus of quite a different size. The 188, for instance, is 16 bits internally but has only eight external data lines.

通常用于表示数据总线为 2 字节宽的处理器。注意, 这种表示法总是指处理器数据总线的宽度, 地址总线通常与 CPU 的大小无关。更使人迷惑的是, 许多嵌入式处理器的内部数据总线是一个宽度, 而外部数据总线又是另一个不同的宽度。例如, 188 处理器的内部数据总线为 16 位, 而外部数据线只有 8 位。

2. *adj.* Any circuit or algorithm that operates on data 2 bytes wide. Examples include 16-bit counter/timers and 16-bit A/D converters.

描述任何对 2 字节宽的数据进行操作的电路或算法。例如 16 位计数器/定时器, 16 位 A/D 转换器。

## 186 186 处理器

(one eighty-six) *num.* A 16-bit processor from Intel. The 186 instruction set is virtually the same as the 8088/8086, but the part packs a number of on-board peripherals, such as timers, DMA controllers, and serial channels. The 186 is part of a family of CPUs from Intel aimed specifically at the embedded market.

**History:** The 8088/8086 were wildly successful in the PC market but struggled in the embedded space because of the extensive support circuits needed. The 186 and its 188 baby brother also were licensed to AMD and others, who created their own variants with the same instruction set but a different mix of on-board peripherals.

Intel 公司的 16 位处理器。其指令集实际上与 8088/8086 处理器的相同, 但集成了若干片内外设, 如定时器、DMA 控制器和串行通道。186 处理器是 Intel 公司专门针对嵌入式市场而推出的 CPU 家族中的一员。

**历史:** 8088/8086 处理器在 PC 机市场上取得了巨大的成功; 但由于需要大量的支持电路, 它在嵌入式领域却一直在挣扎。186 处理器及其小兄弟 188 处理器被授权给了 AMD 及其他一些公司。这些公司推出了各自的变种。所有这些变种处理器与 186 具有相同的指令系统, 但片上外设却各不相同。

## 188 188 处理器

(one eighty-eight) *num.* A peer of the 186 with an 8-bit external data bus. The narrower bus reduces memory bandwidth (every 16-bit data transfer takes twice as long) but also reduces memory cost and the number of traces on the PCB. Highly integrated microcontroller versions of the 188 from Intel and AMD are popular in embedded designs.

一种与 186 对等的处理器, 只是外部数据总线为 8 位。其较窄的总线减少了内存带宽(每 16 位数据的传输时间增加了 1 倍), 但同时也降低了内存成本和印制板上的走线数量。Intel 公司和 AMD 公司推出的高度集成的 188 微控制器版本在嵌入式设计中很流行。

## 1s complement 1 的补码, 二进制反码

See(见)one's complement notation.

## 2s complement 2 的补码, 二进制补码

See(见)two's complement notation.

#



### 3-dB 3 分贝

*adj.* Represents halving the power of a signal. The decibel is  $10 \log(P_1/P_2)$ , where  $P_1$  and  $P_2$  are two power levels. If  $P_1$  is the power of a frequency component input to a filter and  $P_2$  the power output at that frequency,  $P_1/P_2$  is 2.  $10 \log 2$  is about 3, so this half-power point of the amplitude response will show as -3 dB of attenuation at that frequency.

Decibels are often used because the logarithmic scale compresses huge ranges of data to smaller, more manageable, values. For example, the human ear responds to inputs ranging from  $10^{-16}$  to  $10^{-4} \text{ W/cm}^2$ , a range of 1 trillion. Using logs compresses the range to 12.

表示将信号功率减半。分贝的定义是  $10 \log(P_1/P_2)$ , 其中  $P_1$  和  $P_2$  为功率。如果  $P_1$  为滤波器输入的某一频率分量的功率,  $P_2$  为该频率分量的输出功率, 则  $P_1/P_2$  等于 2。  $10 \log 2$  大约为 3, 因此幅度响应的半功率点表现为该频率下的 -3 dB 衰减。

之所以经常使用分贝, 是因为对数尺度能将很大范围的数据压缩为较小且更易处理的数值。例如, 人耳能响应的声强输入范围为  $10^{-16} \sim 10^{-4} \text{ W/cm}^2$ , 即 1 万亿( $10^{12}$ )的范围。使用对数可将范围压缩到 12。

### 32-bit 32 位

1. *adj.* Generally used to denote a processor that uses a data bus 4 bytes wide. Note that this always refers to the processor's data bus width. However, many 32-bit CPUs do use a 32-bit address bus as well. Confusing the issue more, many embedded processors have an internal data bus of one size, but an external bus of a different size. The 386EX, for instance, is 32 bits internally but has only 16 external data lines.

通常用于表示数据总线为 4 字节宽的处理器。注意, 这种表示法总是指处理器数据总线的宽度。但是, 许多 32 位 CPU 也采用 32 位地址总线。使人

迷惑的是, 许多嵌入式处理器的内部数据总线是一个宽度, 而外部数据总线又是另一个不同的宽度。例如, 386EX 处理器的内部数据总线为 32 位, 而外部数据总线只有 16 位。

2. *adj.* Any circuit or algorithm that operates on data 4 bytes wide. Examples include 32-bit CRCs and 32-bit address buses.

描述任何对 4 字节宽的数据进行操作的电路或算法。例如 32 位循环校验码和 32 位地址总线。

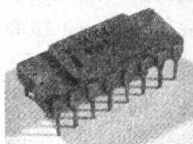
### 386EX 386EX 微控制器

*n.* A microcontroller variant of the 80386 processor from Intel. Designed specifically for use in embedded systems, the 386EX processor included numerous on-chip peripherals. This part was a hermaphrodite, with a 32-bit internal data bus but only a 16-bit external bus.

Intel 公司 80386 处理器的微控制器变种。它专门为嵌入式系统而设计, 包含大量的片上外设。386EX 有两种宽度的数据总线, 内部数据总线为 32 位, 而外部数据总线只有 16 位。

### 4004 4004 微处理器

(four thousand four) *num.* The first microprocessor. Invented by Intel (a memory company at the time) in 1971, the 4004 was the world's first general-purpose microprocessor. A far cry from modern processors, the 4004 had a 12-bit instruction pointer, forty six 8-bit opcodes, and sixteen 4-bit registers. The 4004 was a 16-pin DIP part, containing about 2 000 transistors. See also *Busicom*.



The world's first single-chip processor. 世界首款单片处理器

Intel 公司 1971 年推出的首款微处理器, 当时 Intel 公司是一家存储器生产商。4004 是世界上第一个通用微处理器。4004 微处理器与现代处理器有很

大的不同,它有 12 位指令指针、46 个 8 位操作码和 16 个 4 位寄存器。4004 采用 16 引脚的双列直插式封装,包含大约 2 000 个晶体管。参阅 Busicom。

### 555 555 集成电路

(five five five) *num.* An historically popular 8-pin integrated circuit that can be used to generate a one-shot pulse or periodic clock signal. Also called a 555 timer.

一种历史上曾很流行的 8 引脚集成电路,可用于产生单脉冲或周期性时钟信号。也称为 555 定时器。

### 56000 56000 系列处理器

*num.* A family of digital signal processors from Motorola.

Motorola 公司的数字信号处理器系列。

### 68000 68000 微处理器

*num.* The first 16-bit microprocessor from Motorola. The first member of the 68k family. The 68000 is a 16-bit CPU with separate address and data buses, a single-phase clock, and no on-chip peripherals. The device offered a very orthogonal instruction set reminiscent of the PDP-11s. Its flat address space allowed programmers to address up to 16 MB of memory directly. See also 680x0.

**History:** Reputedly named for the number of transistors on the die, the 68000 was Motorola's attempt to create a PC processor for IBM. The 68000 was technically superior to the 8088/8086, which required cumbersome segmentation for addresses beyond 64 KB. Unable to produce the part on time, however, Motorola lost the PC business to Intel. Apple used the 68000 and later successors in the Mac.

Motorola 公司的首款 16 位微处理器,68k 家族的第一个成员。68000 是 16 位 CPU,具有分离的地址总线和数据总线以及单相时钟,无片上外设。它具有正交指令集,会让人回想起 PDP-11 的指令系

统。其线性地址空间允许程序员直接访问多达 16 MB 的内存空间。参阅 680x0。

**历史:** 据说 68000 是因其芯片上晶体管的数目而得名,它是 Motorola 公司为 IBM 公司生产 PC 处理器的一个尝试。68000 在技术上优于 8088/8086,后者要求在地址空间超过 64 KB 时采用很麻烦的分段技术。然而,由于不能按时生产出芯片,Motorola 公司在 PC 市场上输给了 Intel 公司。Apple(苹果)公司则在 Mac(麦金托什)计算机上使用了 68000 及其后续产品。

### 680x0 680x0 系列处理器

*num.* The family of 68000-series processors comprising the 68010, 68020, 68030, 68040, and beyond. Also called the 68k family. The 680x0 family was the basis of the original Apple and Macintosh computers.

由 68010、68020、68030、68040 及后继型号组成的 68000 系列处理器,也称为 68k 系列。680x0 系列是最初的 Apple 和后来的 Mac 计算机的基础。

### 683xx 683xx 系列微控制器

*num.* Motorola's family of 68000-like microcontrollers targeted specifically at embedded systems. The 683xx family primarily comprises the CPU32 series of processors, which use essentially the 68000 instruction set. Each of the family members has a different mix of on-chip peripherals.

Motorola 公司类似于 68000 的微控制器系列,专门针对嵌入式系统而设计。683xx 系列主要由 CPU32 系列处理器组成,基本上采用 68000 处理器的指令集。系列中的每一个成员都具有不同的片上外设。

### 68k 68k 系列处理器

(sixty-eight kay) *abbr.* See(见)680x0.

### 68xx 68xx 系列微控制器

*num.* A family of 8- and 16-bit microcontrollers



#

from Motorola. Family members include the 68HC05, 68HC11, and 68HC12.

Motorola 公司的 8 位和 16 位微控制器系列,其成员包括 68HC05、68HC11 和 68HC12。

## 6DOF 6 自由度

(six doff) *n.* Able to move in either direction along three different axes, typically in the 3-D world. Short for Six Degrees Of Freedom.

**Example:** A helicopter can move up or down, left or right, and forward or backward; thus, it is the quintessential example of the 6DOF capability.

指能够沿着 3 个不同轴的任一方向移动,通常是指在三维世界中。为 Six Degrees Of Freedom 的缩略语。

**示例:** 直升机可上或下、左或右、前或后运动。因此,它是一个精萃的 6 自由度例子。

## 7-segment display 7 段显示

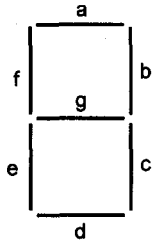
*n.* A device that displays all numbers and many letters and symbols via just seven linear segments. Most also have one or two decimal points. The LED version of an 7-segment display comes in both common anode and common cathode configurations, indicating whether each diode's anode or cathode is connected together. 7-segment LCD displays are also widely used.

There's no standard for driving LEDs. Some designers drive them directly from the processor's PIO pins, connecting each segment to an output pin. This uses more I/O pins but lets the programmer generate any of the possible characters. Others use a driver IC (e. g., 74HC4511), which accepts a 4-bit input, creates a character, amplifies the output, and drives the segments. These limit the possible characters that can be displayed.

To support multidigit displays, most vendors supply banked 7-segment displays, in which the anodes or cathodes are brought out individually to pins, and the other terminals — those connected to

segments — are all tied together. This supports multiplexing of the display via dedicated hardware or smart software.

一种仅用 7 条直线段显示所有数字、许多字母和符号的器件。大多数 7 段显示器件还有 1 个或 2 个小数点。7 段 LED 显示有共阳极和共阴极两种配置,表示将所有二极管的阳极或阴极连接在一起。7 段 LCD 显示也得到了广泛使用。



Layout of a 7-segment display. 7 段显示布局图

没有驱动 LED 的标准。有些设计者直接从处理器的 PIO 引脚驱动发光二极管,将每个段都连接到一个输出引脚。这要占用较多的 I/O 引脚,但程序员可生成任何可能的字符。另外一些设计者则采用驱动集成电路(如 74HC4511)。这样的驱动电路接受 4 位输入,生成字符,并对输出放大,最后驱动显示段。这样所能显示的字符就受到限制。

为了支持多位数字显示,大多数厂家提供 7 段显示器模组,其各个阳极或阴极单独引接到引脚上,而其他极(那些连接到段的极)都连接在一起。这种组合方式通过专用的硬件或智能软件可支持多路动态显示。

## 8-bit 8 位

1. *adj.* Generally used to denote a processor that uses a data bus 1 byte wide. Note that this always refers to the processor's data bus width. Typical 8-bit CPUs have 16 or 20 address lines.

通常用于表示使用 1 字节宽数据总线的处理器。注意,这种表示法总是指处理器的数据总线宽度。通常 8 位 CPU 有 16 或 20 根地址线。

2. *adj.* Any circuit or algorithm that operates on data 1-byte wide. Examples include 8-bit buffers and 8-bit D/A converters.

指任何对 1 字节宽数据进行操作的电路或算法。例如,8 位缓冲器和 8 位 D/A 转换器。