

DIANQI KONGZHI YU
PLC ZONGHE YINGYONG JISHU

电气控制与 PLC 综合应用技术

赵江稳 主编



中国电力出版社
CHINA ELECTRIC POWER PRESS

DIANQI KONGZHI YU
PLC ZONGHE YINGYONG JISHU

电气控制与 PLC 综合应用技术

主编 赵江稳

参编 吕增芳 杜相如



中国电力出版社
CHINA ELECTRIC POWER PRESS

内 容 简 介

本书介绍了通用电气公司 (General Electric Company, GE) 可编程自动化控制器 (PAC) 技术的相关基础知识, 内容包括电气控制基础、PLC 基础知识、PAC RX3i 硬件、PME 软件、PAC 指令系统、人机交互——触摸屏、RX3i 控制系统基础实践与综合应用。附录给出了 GE 智能平台的系统变量表、GE 智能平台 PAC 指令一览表和 AWG 电缆标准单位换算。

本书可作为普通高等学校电气与自动化技术、自动化、机电一体化等相关专业教材, 也可作为相关工程技术人员的参考书。

图书在版编目 (CIP) 数据

电气控制与 PLC 综合应用技术 / 赵江稳主编. —北京: 中国电力出版社, 2014.3

ISBN 978-7-5123-5551-4

I. ①电… II. ①赵… III. ①电气控制②plc 技术 IV. ①TM571.2②TM571.6

中国版本图书馆 CIP 数据核字 (2014) 第 026116 号

中国电力出版社出版、发行

(北京市东城区北京站西街 19 号 100005 <http://www.cepp.sgcc.com.cn>)

汇鑫印务有限公司印刷

各地新华书店经售

*

2014 年 3 月第一版 2014 年 3 月北京第一次印刷

787 毫米×1092 毫米 16 开本 16 印张 389 千字

印数 0001—3000 册 定价 29.80 元

敬告读者

本书封底贴有防伪标签, 刮开涂层可查询真伪
本书如有印装质量问题, 我社发行部负责退换

版 权 专 有 翻 印 必 究

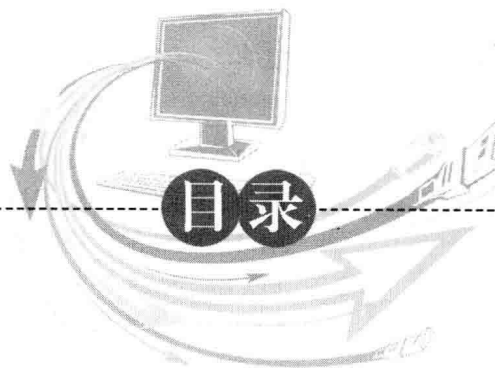
目前市面上关于 GE 可编程控制器的资料不多；而且很多是英文编写的，非常不方便国内读者阅读。为了帮助读者能够快速掌握 GE 可编程控制器的基本知识，作者在参阅大量关于 GE 原文资料和各种文献之后编写了此书。本书总共分为 7 章。

第 1 章介绍了电气控制基础知识；第 2 章介绍了 PLC 基础知识；第 3 章介绍了 PAC RX3i 硬件系统；第 4 章介绍了 PME 软件；第 5 章介绍了 PAC 的指令系统；第 6 章介绍了触摸屏——人机交互；第 7 章介绍了 RX3i 控制系统基础实践与综合应用；附录给出了 GE 智能平台的系统变量表、GE 智能平台 PAC 指令一览表和 AWG 电缆标准单位换算。本书第 1.2 节由吕增芳编写，第 1.3 节由杜相如编写，其余章节由赵江稳编写。全书由赵江稳统稿。在编写本书的过程中得到了马宁、孔红、吕增芳、杨国生、弓宇等的指导，有一些项目的参考程序由吕增芳提供，全书的参考程序由敖马泽一一验证，在此一并表示感谢。

在编写过程中，作者参阅和引用了通用电气公司的最新技术资料和相关院校、企业、科研院所的一些教材、文献和资料。有些正式出版的文献已在书的参考文献中列出，有些难免遗漏，对未能列出的文献和资料，作者向其作者表示诚挚的感谢。

由于 GE 在不断发展，组织结构、经销商、新产品等都在不断变化；再加上作者学识有限、编写时间仓促，因此书中难免出现一些错误和问题，希望阅读此书的读者和相关专家不吝赐教，以便再版时改正。

作 者



前言

第1章 电气控制基础	1
1.1 数字电路基础知识	1
1.1.1 数制与码制	1
1.1.2 十进制	1
1.1.3 二进制	1
1.1.4 八进制	2
1.1.5 十六进制	2
1.1.6 进制间的转换	2
1.1.7 BCD 码	5
1.1.8 ASCII 码	5
1.1.9 Unicode 码	7
1.1.10 中文编码	8
1.1.11 逻辑代数	10
1.1.12 电气控制线路的逻辑代数表示法	12
1.2 电气控制基础知识	14
1.2.1 三相笼型异步电动机全压启动控制电路（也有资料写成“起动”）	15
1.2.2 三相笼型异步电动机减压启动控制电路（也有写为降压启动）	23
1.2.3 电动机的制动控制电路	27
1.3 变频器基础	30
1.3.1 转速和频率的关系	31
1.3.2 电动机要求主磁通量不变的原因	31
1.3.3 变频器的控制电路	31
1.3.4 变频器的额定数据	34
1.3.5 变频器的发展	34
1.3.6 MM440 通用型变频器	35
思考题	37
第2章 PLC 基础	38
2.1 PLC 概述	38
2.1.1 PLC 的产生和发展	38

2.1.2	PLC 的特点	41
2.1.3	PLC 的性能指标和分类	42
2.1.4	PLC 的应用	45
2.2	PLC 的组成	45
2.2.1	PLC 的结构及各部分的作用	45
2.2.2	PLC 的配置	55
2.3	PLC 的工作原理	56
2.3.1	PLC 的等效电路	56
2.3.2	PLC 的工作过程	57
2.4	PLC 的软件基础	58
2.4.1	PLC 的分类	58
2.4.2	PLC 的编程语言	59
	思考题	61
第 3 章	PAC RX3i 硬件	62
3.1	PAC 概述	62
3.1.1	PAC 概念	62
3.1.2	PAC 的发展历史	62
3.1.3	PAC 的特征	63
3.1.4	PAC 的开发和功能优势	64
3.1.5	PAC 与 PLC 的区别	64
3.1.6	PAC 与 PC Control 的区别	65
3.1.7	GE 智能平台自动化设备一览	65
3.1.8	PAC Systems 解决方案	66
3.2	PAC Systems RX3i 硬件	67
3.2.1	PAC Systems RX3i 的背板	69
3.2.2	PAC Systems RX3i 的电源 (POWER)	84
3.2.3	PAC Systems RX3i 的 CPU (IC695CPU310)	93
3.2.4	PAC Systems RX3i 特殊功能模块	96
3.2.5	PAC Systems RX3i 数字量输入模块 (digital input, DI)	106
3.2.6	PAC Systems RX3i 数字量输出模块 (digital output, DO)	109
3.2.7	PAC Systems RX3i 数字量混合模块	112
3.2.8	PAC Systems RX3i 模拟量输入模块 (Analog Input, AI, A/D 转换)	115
3.2.9	PAC Systems RX3i 模拟量输出模块 (Analog Output, AO, D/A 转换)	120
	思考题	122
第 4 章	PME 软件	123
4.1	Proficy 软件家族	123
4.2	PME 简介	125
4.2.1	PME 的特点	126
4.2.2	PME 的缺点或者不足	128

4.2.3	PME 组件	128
4.3	PME 的安装	128
4.4	PME 工具介绍	132
4.4.1	工具栏	133
4.4.2	浏览工具	133
4.4.3	属性检查工具	135
4.4.4	在线帮助	135
4.4.5	反馈信息工具	135
4.4.6	数据监视工具	136
4.4.7	工具箱	138
4.4.8	编辑器窗口	138
4.5	PME 使用	139
4.5.1	创建工程	140
4.5.2	硬件配置	142
4.5.3	编写程序	147
	思考题	156
第 5 章	PAC 指令系统	157
5.1	GE PAC 的数据类型、数据存储和变量	157
5.1.1	PAC 支持的数据类型	157
5.1.2	PAC 的存储区域	159
5.2	GE PAC 基本指令系统	162
5.2.1	继电器指令	163
5.2.2	定时器指令	167
5.2.3	计数器指令	170
5.2.4	数学运算指令	172
5.2.5	比较运算指令	176
5.2.6	位操作	177
5.2.7	数据传送指令	179
5.2.8	数据表指令	181
5.2.9	数据转换指令	183
5.2.10	控制指令	184
5.2.11	程序流程指令	185
	思考题	186
第 6 章	人机交互——触摸屏	187
6.1	GE 触摸屏硬件	187
6.1.1	QuickPanel View/Control 6" TFT 硬件组成	189
6.1.2	QuickPanel View/Control 6" TFT 的基本安装	191
6.1.3	QuickPanel View/Control 6" TFT 启动设置	192
6.1.4	QuickPanel View/Control 6" TFT 以太网设置	192

6.1.5	QuickPanel View/Control 6" TFT 串行数据通信端口设置	193
6.1.6	QuickPanel View/Control 6" TFT USB 接口设置	194
6.2	GE 触摸屏操作界面开发设计	194
6.2.1	新建 Quick Panel 界面	194
6.2.2	创建触摸屏	194
6.2.3	创建驱动	195
6.2.4	触摸屏地址配置	196
6.2.5	PAC 关联地址配置	196
6.2.6	触摸屏界面创建	197
6.2.7	触摸屏界面制作	197
6.3	GE 触摸屏操作界面调试	200
	思考题	201
第 7 章	RX3i 控制系统基础实践与综合应用	203
7.1	输出互锁控制	203
7.2	3 灯 3 开关控制	204
7.3	双灯单按钮控制	204
7.4	通电延时控制	206
7.5	断电延时控制	206
7.6	脉冲方波的产生	207
7.7	计数通断控制	208
7.8	交叉计数控制	209
7.9	超时报警控制	211
7.10	小型三相异步电动机的控制	212
7.10.1	电动机点动、自锁控制	212
7.10.2	电动机正、反转控制	214
7.10.3	电动机禁止多次启动控制	214
7.10.4	电动机自锁运行控制	215
7.10.5	电动机 Y- Δ 启动控制	217
7.10.6	触摸屏设定运转速度控制 (模拟量输出控制)	217
7.10.7	触摸屏设定运转速度控制 (模拟量输入控制)	218
7.10.8	传送指令控制	220
7.11	LED 灯轮换显示	221
7.12	移位寄存器、数据传输和块清除指令控制	226
	思考题	228
附录 A	GE 智能平台的系统变量表	229
附录 B	GE 智能平台 PAC 指令一览表	233
附录 C	AWG 电线标准单位换算	245
	参考文献	248

电气控制基础

本章主要介绍与 PLC 相关的基础知识,包括数字电路的基础知识、电气控制的基础知识和变频器基础知识。

1.1 数字电路基础知识

1.1.1 数制与码制

数制是人们按某种进位规则进行计数的科学方法。生活中常见的数制有十进制、七进制(星期)、十二进制(一年有 12 个月)、二十四进制(每天 24 个小时)、六十进制(时、分、秒)等。计算机中常用二进制、八进制、十六进制。在通信、情报等领域还涉及到编码的问题。

1.1.2 十进制

数值的基值(基数)决定了计数中不同符号或者数字的总数。最常用的十进制的基值(基数)是 10,也就是说在十进制中只会出现 0、1、2、…、9 这 10 个数码。十进制数的值取决于它的位数以及每一位的值。分配(权)值被赋值到各个位置,数字从右往左置于相应的位置中。在十进制中,最右边第一位为 0,第二位为 1,第三位为 2……依此类推持续到最后一位。每个位的权值能够通过基值 10 及其幂的组合形式来表达。例如:

$$\begin{aligned} 2013 &= 2 \times 10^3 + 0 \times 10^2 + 1 \times 10^1 + 3 \times 10^0 \\ (3176.54)_{10} &= 3 \times 10^3 + 1 \times 10^2 + 7 \times 10^1 + 6 \times 10^0 + 5 \times 10^{-1} + 4 \times 10^{-2} \end{aligned}$$

十进制数字可以在数字的下标位置写上数码 10 来标注,即 2013_{10} ,或者省略不写。由于十进制系统的英文是 Decimal System,因此很多时候也用字母 D 表示十进制: $(2013)_D$ 。所以 $2013 = (2013)_{10} = (2013)_D$,这三种表示形式都是可行的。

十进制的计数原则是“逢十进一,借一当十”。十进制的加法和乘法我们都很熟悉。其他进制的表示与十进制类似。

1.1.3 二进制

二进制以数字 2 为基值,在此系统中只存在 0、1 两个数码。由于在数字系统中可以很容易地辨别出和二进制 0、1 相关联的两种不同状态,因此二进制系统能够非常方便地用于 PLC 和计算机系统。大多数的 PLC 定时器和计算器工作在二进制计数模式下,如二进制数 0、

1、00、01、11 等。

很多时候用下标数字 2 或者字母 B (binary) 表示二进制系统。二进制的计数原则是“逢二进一，借一当二”。二进制的加法和乘法关系为

$$0+0=0, 0+1=1+0=1, 1+1=10$$

$$0\times 0=0, 0\times 1=1\times 0=0, 1\times 1=1$$

1.1.4 八进制

八进制由 0、1、2、3、4、5、6、7 这 8 个数码组成，其基数为 8。

很多时候用字母 O (Octal) 表示八进制系统，但是字母 O 非常容易和数字 0 混淆，有些场合下用下标 8 或者 Q 表示八进制系统。

1.1.5 十六进制

十六进制由 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 这 16 个数码组成，其基数为 16。

很多时候用下标数字 16 或者字母 H (Hexadeximal) 表示十六进制系统。

1.1.6 进制间的转换

进制转换是人们利用符号来计数的方法，包含很多种数字转换。进制转换由一组数码符号和两个基本因素（“基”与“权”）构成。

1. 其他进制转换为十进制

其他进制转换为十进制比较容易，例如：

$$(2013)_D = 2 \times 10^3 + 0 \times 10^2 + 1 \times 10^1 + 3 \times 10^0$$

二进制数转换为十进制数的规律：把二进制数按位权形式展开为多项式和的形式，求其最后的和，就是所对应的十进制数，简称“按权求和”。例如：

$$(10011.11)_B = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = (19.75)_D$$

用同样的方法可以得出八进制数 $(45.2)_Q$ 转换为十进制数的方法：

$$(45.2)_Q = 4 \times 8^1 + 5 \times 8^0 + 2 \times 8^{-1} = (37.25)_D$$

其他进制的数转化为十进制数的方法和上面类似。

2. 十进制转换为二进制

十进制转换为二进制，分为整数部分和小数部分。

(1) 整数部分。

方法：除 2 取余法，即每次将整数部分除以 2，余数为该位权上的数，而商继续除以 2，余数又为上一个位权上的数，这个步骤一直持续下去，直到商为 0 为止，最后读数时候，从最后一个余数读起，一直到最前面的一个余数。

【例 1-1】 将十进制的 53 转换为二进制数。

第一步，53 除以 2，商 26，余数为 1。

第二步，26 除以 2，商 13，余数为 0。

第三步，13 除以 2，商 6，余数为 1。

第四步, 6 除以 2, 商 3, 余数为 0。

第五步, 3 除以 2, 商 1, 余数为 1。

第六步, 1 除以 2, 商 0, 余数为 1。

第七步, 读数, 因为最后一位是经过多次除以 2 才得到的, 因此它是最高位, 读数字从最后的余数向前读, 即 110101, 记作 $(53)_D = (110101)_B$ 。

$n = \text{权}$

先写商再写余, 无余数写零。

除 (权) 2	53	1(余)	得: 110101 ↑ 反向写出
权 2	26(商)	0	
取 2	13	1	
余 2	6	0	
	3	1	
	1	1	
	0		

(2) 小数部分。

方法: 乘 2 取整法, 即将小数部分乘以 2, 然后取整数部分, 剩下的小数部分继续乘以 2, 然后取整数部分, 剩下的小数部分又乘以 2, 一直取到小数部分为零为止。如果永远不能为零, 就同十进制数的四舍五入一样, 按照要求保留多少位小数, 就根据后面一位是 0 还是 1 取舍: 如果是零, 舍掉; 如果是 1, 进入一位。换句话说, 就是 0 舍 1 入。读数要从前面的整数读到后面的整数。

【例 1-2】 将 0.125 换算为二进制数。

第一步, 0.125 乘以 2, 得 0.25, 其整数部分为 0, 小数部分为 0.25。

第二步, 小数部分 0.25 乘以 2, 得 0.5, 其整数部分为 0, 小数部分为 0.5。

第三步, 小数部分 0.5 乘以 2, 得 1.0, 其整数部分为 1, 小数部分为 0.0。

第四步, 读数。从第一位读起, 读到最后一位, 即为 0.001。

【例 1-3】 将 0.45 转换为二进制数 (保留到小数点后第四位)。

第一步, 0.45 乘以 2, 得 0.9, 其整数部分为 0, 小数部分为 0.9。

第二步, 小数部分 0.9 乘以 2, 得 1.8, 其整数部分为 1, 小数部分为 0.8。

第三步, 小数部分 0.8 乘以 2, 得 1.6, 其整数部分为 1, 小数部分为 0.6。

第四步, 小数部分 0.6 乘以 2, 得 1.2, 其整数部分为 1, 小数部分为 0.2。

第五步, 小数部分 0.2 乘以 2, 得 0.4, 其整数部分为 0, 小数部分为 0.4。

第六步, 小数部分 0.4 乘以 2, 得 0.8, 其整数部分为 0, 小数部分为 0.8。

第七步, 小数部分 0.8 乘以 2, 得 1.6, 其整数部分为 1, 小数部分为 0.6。

从上面步骤可以看出, 当第五次做乘法时, 得到的结果是 0.4, 那么小数部分继续乘以 2, 得 0.8, 0.8 又乘以 2 的, 到 1.6 这样一直乘下去, 最后不可能使得小数部分为零。因此, 这个时候只好借鉴十进制的方法进行四舍五入了, 但是二进制只有 0 和 1 两个数字, 于是就出现 0 舍 1 入。因此在这种转换运算中计算机会产生误差, 但是由于保留位数很多, 精度很高, 所以可以忽略不计。那么, 我们可以得出结果将 0.45 转换为二进制约等于 0.0111。

将十进制的数转化为其他进制的数, 可以仿照上面的方法。例如, 要将十进制的 120 转

换成八进制数。

被除数	计算过程	商	余数
120	120/8	15	0
15	15/8	1	7
1	1/8	0	1

因此得到： $(120)_D=(170)_8$ 。

显然，将十进制数转换成八进制的方法和转换为二进制的方法类似，唯一的变化就是除数由 2 变成 8。

3. 二进制与八进制、十六进制之间的转换

原则上可以先将二进制数转化为十进制数（此时十进制可以称为中间进制），然后把得到的十进制数转化为八进制数、十六进制数即可（中间进制也可以选取其他）。显然这种方法慢且易出错。

还有一种简便的方法。注意到数学关系 $2^3=8$ ， $2^4=16$ ，我们可以利用八进制和十六进制的这种关系来转换。即用三位二进制表示一位八进制数字，用四位二进制表示一位十六进制数字。

（1）二进制转换为八进制。例如，要将二进制数 $(11001.101)_B$ 转化为八进制数，简便方法如下。

整数部分：从右往左每三位一组，缺位处用 0 填补，然后按十进制方法进行转化，则有： $011=3, 001=1$ ，然后将结果按从左往右的顺序书写就是 31，那么这个 31 就是二进制 11001 的八进制形式。

小数部分：从左往右每三位一组，缺位处用 0 填补，然后按十进制方法进行转化，则有： $101=5$ ，然后将结果部分按左往右的顺序书写就是 5，那么这个 5 就是二进制 0.101 的八进制形式。

所以 $(11001.101)_B=(31.5)_8$ 。

（2）二进制转换为十六进制。仿照上面的方法，我们很容易能将一个二进制数转换为十六进制数，只不过这种情况下是每四位一组。例如， $(10100101)_B=(A5)_{16}$ 。

（3）将八进制转换为二进制。

方法：取一分三法，即将一位八进制数分解成三位二进制数，用三位二进制按权相加去凑这位八进制数，小数点位置照旧。具体如下。

首先，将八进制按照从左到右，每位展开为三位，小数点位置不变。

然后，按每位展开为 2^2 、 2^1 、 2^0 （即 4、2、1）三位去做凑数，即 $a \times 2^2 + b \times 2^1 + c \times 2^0 = \text{该位上的数}$ （ $a=1$ 或者 $a=0$ ， $b=1$ 或者 $b=0$ ， $c=1$ 或者 $c=0$ ），将 abc 排列就是该位的二进制数。接着，将每位上转换成的二进制数按顺序排列。

最后，就得到了八进制转换成二进制的数字。

（4）八进制与十六进制的转换。

方法：一般不直接转换，而是选取二进制为中间进制，相应的转换请参照上面二进制与八进制的转换和二进制与十六进制的转换。

至于各个进制数之间的加减乘除，还需要掌握原码、反码、补码、移码等概念，这里略去，读者可从一般的计算机基础、计算机组成原理、数字电路或者数学教材中获得。

1.1.7 BCD 码

BCD (Binary-Coded Decimal) 码亦称二进码十进数或二-十进制代码, 用 4 位二进制数来表示 1 位十进制数中的 0~9 这 10 个数码。它是一种二进制的数字编码形式, 即用二进制编码的十进制代码。这种编码形式利用了四个位来储存一个十进制的数码, 使二进制和十进制之间的转换得以快捷的进行。这种编码技巧最常用于会计系统的设计里, 因为会计制度经常需要对很长的数字串做准确的计算。相对于一般的浮点式记数法, 采用 BCD 码既可保证数值的精确度, 又可免去使计算机做浮点运算时所耗费的时间。此外对于其他需要高精度的计算, BCD 编码亦很常用。由于十进制数共有 0、1、2、...、9 十个数码, 因此, 至少需要 4 位二进制码来表示 1 位十进制数。4 位二进制码共有 $2^4(=16)$ 种码组。在这 16 种代码中, 可以任选 10 种来表示 10 个十进制数码, 因此可以选择编码的方案种类很多。常用的 BCD 代码可分为有权码和无权码两类: 有权 BCD 码有 8421 码、5421 码、2421 码等, 其中 8421 码是最常用的; 无权 BCD 码有余 3 码、格雷码 (严格讲格雷码并不是 BCD 码) 等。

8421BCD 码是最基本和最常用的 BCD 码, 它和四位自然二进制码相似, 各位的权值为 8、4、2、1, 故称为有权 BCD 码。和四位自然二进制码不同的是, 它只选用了四位二进制码中的前 10 组代码, 即用 0000~1001 分别代表它所对应的十进制数, 余下的六组代码不用, 也就是说四位二进制数的其余六个编码 1010、1011、1100、1101、1110、1111 不是有效编码。见表 1-1。

表 1-1 常用二-十进制数 (BCD 码)

比 8421BCD 码多余 3

十进制数	有权码				无权码
	8421 码	5421 码	2421 (A)	2421 (B)	余 3 码
0	0000	0000	0000	0000	0011
1	0001	0001	0001	0001	0100
2	0010	0100	0010	0010	0101
3	0011	0101	0011	0011	0110
4	0100	0111	0100	0100	0111
5	0101	1000	0101	1011	1000
6	0110	1001	0110	1100	1001
7	0111	1100	0111	1101	1010
8	1000	1101	1110	1110	1011
9	1001	1111	1111	1111	1100

权为 8、4、2、1

1.1.8 ASCII 码

ASCII (American Standard Code for Information Interchange, 美国信息互换标准代码) 是基于拉丁字母的一套计算机编码系统。它主要用于显示现代英语, 而其扩展版本 EASCII 则可以部分支持其他西欧语言, 并等同于国际标准 ISO/IEC 646。ASCII 第一次以规范标准的形态发表是在 1967 年, 最后一次更新则在 1986 年, 至今共定义了 128 个字符; 其中 33 个字符无法显示 (这是以现今操作系统为依归, 但在 DOS 模式下可显示出一些诸如笑脸、扑克牌花

ASCII 字 符

表 1-2

ASCII 非打印控制字符										ASCII 打印字符																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
0000										0001																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
0										1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
高四位	十进制				十进制				低四位	十进制				十进制				十进制																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
	0	1	2	3	4	5	6	7		8	9	10	11	12	13	14	15																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
0000	0	BLANK NULL	^@	NUL	空	字符 解释	十进制	字符	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符 解释	十进制	Ctrl	代码	字符

式等 8 位符号)，且这 33 个字符多数已是陈废的控制字符。控制字符的用途主要是操控已经处理过的文字。在 33 个字符之外的是 95 个可显示的字符。用敲下键盘空格键所产生的空白字符也算 1 个可显示字符（显示为空白）。见表 1-2。

万维网使得 ASCII 广为通用，但是 ASCII 的局限在于只能显示 26 个基本拉丁字母、阿拉伯数字和英式标点符号，因此只能用于显示现代美国英语（而且在处理英语当中的外来词（如 naïve、café、élite 等）时，所有重音符号都不得不去掉，即使这样做会违反拼写规则）。

EASCII（Extended ASCII，延伸、扩展美国标准信息交换码）是将 ASCII 由 7 位扩充为 8 位而成。EASCII 的内码是由 0~255 共 256 个字符组成的。EASCII 比 ASCII 扩充出来的符号包括表格符号、计算符号、希腊字母和特殊的拉丁符号。ISO/IEC 8859 是最常见的 8 位字符编码。除此之外，不同的操作系统都会有它的 8 位字符编码。见表 1-3。

表 1-3 扩展 ASCII 码

128	Ç	144	É	160	á	176	☒	192	ℒ	208	⊥	224	α	240	≡
129	ù	145	æ	161	í	177	☒	193	⊥	209	⊥	225	β	241	±
130	é	146	Æ	162	ó	178	■	194	⊥	210	⊥	226	Γ	242	≥
131	â	147	ó	163	ú	179		195	⊥	211	ℒ	227	π	243	≤
132	ä	148	ö	164	ñ	180	⊥	196	—	212	ℒ	228	Σ	244	∫
133	à	149	ò	165	Ñ	181	⊥	197	+	213	ℒ	229	σ	245	∫
134	å	150	û	166	ª	182	⊥	198	⊥	214	ℒ	230	μ	246	÷
135	ç	151	ù	167	º	183	⊥	199	⊥	215	+	231	τ	247	≈
136	ê	152	ÿ	168	¿	184	⊥	200	ℒ	216	+	232	Φ	248	°
137	ë	153	Ö	169	Γ	185	⊥	201	ℒ	217	⊥	233	⊙	249	.
138	è	154	Ü	170	¬	186		202	⊥	218	ℒ	234	Ω	250	.
139	ï	155	ø	171	½	187	⊥	203	⊥	219	■	235	δ	251	√
140	î	156	£	172	¼	188	⊥	204	⊥	220	■	236	∞	252	°
141	ì	157	¥	173	ı	189	⊥	205	=	221	■	237	φ	253	²
142	Ä	158	§	174	《	190	⊥	206	+	222		238	ζ	254	■
143	Å	159	ƒ	175	》	191	⊥	207	⊥	223	■	239	∩	255	

1.1.9 Unicode 码

EASCII 虽然解决了部分西欧语言的显示问题。但对更多其他语言依然无能为力，同时由于互联网的普及，强烈要求推出一种统一的编码方式。

20 世纪 80 年代末，位于美国加利福尼亚州的 Unicode 组织联合了主要的计算机软硬件厂商，如奥多比系统、苹果、惠普、IBM、微软、施乐等公司，同时国际标准化组织（ISO）因为计算机普及和信息国际化的要求，分别各自成立了 Unicode 组织和 ISO 10646 工作小组。他们不久便发现对方的存在，大家为着相同的目的而工作，于是两个组织便共同合作开发了适用于各国语言的通用码，而且“相当有默契地”各自发表 Unicode 和 ISO 10646 字集。虽然实际上两者的字集编码相同，但实质上两者确实为两个不同的标准。目前 Unicode 发展由

非营利机构统一码联盟 (<http://www.unicode.org/>) 负责, 该机构致力于使 Unicode 方案取代既有的字符编码方案, 因为既有的方案往往空间非常有限, 亦不适用于多语环境。直到 2007 年 12 月, ASCII 逐渐被 Unicode 取代。

Unicode 伴随着通用字符集的标准而发展, 同时也以书本的形式对外发表。统一码联盟在 1991 年首次发布了 The Unicode Standard。Unicode 的开发结合了国际标准化组织所制定的 ISO/IEC 10646, 即通用字符集。自 1991 年 10 月发布 Unicode 1.0 至 2013 年 9 月 30 日, 版本已经达到 Unicode 6.3, 收入的字符超过十万个 (第十万个字符在 2005 年获采纳)。Unicode 涵盖的数据除了视觉上的字形、编码方法、标准的字符编码外, 还包含了字符特性, 如大小写字母。

UTF-8 编码是在互联网上使用最广的一种 Unicode 实现方式。其他实现方式还包括 UTF-16 (字符用两个字节或四个字节表示) 和 UTF-32 (字符用四个字节表示)。由于多种优势集合在一起, 导致以 UTF-8 编码的 Unicode 广泛地应用于电脑软件的国际化与本地化过程。有很多新科技, 如可扩展置标语言、Java 编程语言以及现代操作系统, 都采用 Unicode 编码。

1.1.10 中文编码

在联合国的 6 种工作语言 (汉语、英语、法语、俄语、阿拉伯语与西班牙语) 中, 除了汉语是笔画文字外, 其余都是字母文字。前面说到 ASCII 以及扩展 ASCII 编码可以表示的最大字符数是 256, 其实英文字符并没有那么多, 一般只用前 128 个 (最高位为 0), 其中包括了控制字符、数字、大小写字母和其他一些符号。但是面对中文等复杂的文字, 255 个字符显然不够用。于是, 各个国家纷纷制定了自己的文字编码规范。

据德国出版的《语言学及语言交际工具问题手册》说, 现在世界上查明的有 5651 种语言。2009 年 2 月 19 日联合国教科文组织发布的《世界濒危语言图谱 (第三版)》显示, 全世界约有 7000 种语言, 但是全世界 80% 的人讲 83 种主要语言, 剩下 6000 多种语言绝大多数从没有过文字记载。使用人数超过一亿的语言有 12 种, 它们是: 汉语、英语、印地语、西班牙语、阿拉伯语、孟加拉语、葡萄牙语、法语、俄语、印度尼西亚语、德语、日语。使用这十几种语言的人占世界人口的近 60%。

汉语言是全球当之无愧的第一使用语言, 以汉语为主语言的人大约有 14 亿 (3 千万人作为第二语言)。据 2010 年第六次全国人口普查主要数据公报 (第 1 号) (2011 年 4 月 28 日发布) 的内容, 全国总人口为 1370536875 人 (含港、澳、台地区); 据 2011 年 11 月 30 日在上海举行的第二届中国侨务论坛公布的一项研究成果表明, 当时海外华人约为 5000 万人 (当然也有一些华人使用的语言不是汉语); 2011 年 10 月 26 日联合国人口基金发表的《2011 年世界人口状况报告》中提到世界人口 5 天后将达到 70 亿; 除此之外还有很多国家针对母语是非汉语的人员开设了汉语课程、留学生等等。因此使用汉语言人群的数量约占世界人口的 1/5。

根据以英语作为母语的人数计算, 英语是世界上最广泛的第二语言, 但它可能是世界上第二大或第四大语言 (1999 年统计为 3.8 亿人使用英语)。而且世界上 60% 以上的信件是用英语书写的, 50% 以上的报纸杂志是英语的。

印地语和乌尔都语加起来是世界上第二大语言, 使用人口超过 5 亿人。

西班牙语是世界第三大语言, 约有 3.52 亿人使用。

阿拉伯语使用人口 3 亿以上。

孟加拉语使用人口2亿以上。

葡萄牙语使用人口近2亿。

法语使用人口约2亿。

全球以俄语为母语的使用人数超过1.4亿，当做第二语言使用的则有近4500万人。

印度尼西亚语：全世界约有1700万到3000万人将印度尼西亚语作为他们的母语，还有大约1.4亿人将印度尼西亚语作为第二语言。

德语使用人口超过1.1亿。

日语使用人口近1.1亿。

很显然，汉语的计算机信息交换、交流是一个必须解决的大问题。

对于中文输入，由于历史的原因，在Unicode之前一共存在过3套中文编码标准。Big5是中国台湾使用的编码标准，编码了中国台湾使用的繁体汉字，大概有8千多个。HKSCS，是中国香港使用的编码标准，字体也是繁体，但跟Big5有所不同。中国大陆使用的是GB 2312—1980。

1979年，电子工业部华北计算技术研究所（现中国电子科技集团公司第十五研究所）根据国家标准总局下达的关于制定国家标准汉字信息交换码的任务，会同国内15个从事计算机研制、教学、生产、应用和文字研究的单位，在华北计算技术研究所已有工作的基础上，经过两年的努力，由陈耀星等人制定，于1981年5月1日公布了国家标准“GB 2312—1980《信息交换用汉字编码字符集·基本集》”，又称GB0。该项成果获1985年国家科技进步奖一等奖。它是汉字信息处理领域里最重要的基础标准，规定了汉字信息交换用的基本图形字符及其二进制编码表示，适用于一般汉字处理、汉字通信系统间的信息交换。GB 2312编码通行于中国大陆，中国大陆几乎所有的中文系统和国际化的软件都支持GB 2312；新加坡等地也采用此编码。该标准共收录6763个汉字，其中一级汉字3755个，二级汉字3008个；同时收录了包括拉丁字母、希腊字母、日文平假名及片假名字母、俄语西里尔字母在内的682个字符。GB 2312的出现，基本满足了汉字的计算机处理需要，它所收录的汉字已经覆盖中国大陆99.75%的使用频率。

GB 2312—1980是和ASCII兼容的一种编码规范，它利用扩展ASCII没有真正标准化这一点，把一个中文字符用两个扩展ASCII字符来表示。但是这个方法最大的问题就是中文字没有真正属于自己的编码，因为扩展ASCII码虽然没有真正的标准化，但是PC里的ASCII码还是有一个事实标准的（存放着英文制表符），所以很多软件利用这些符号来画表格。这样的软件应用到中文系统中，这些表格符就会被误认作中文字，破坏版面。而且，统计中英文混合字符串中的字数，也是比较复杂的，必须先判断一个ASCII码是否扩展，以及它的下一个ASCII是否扩展，然后才“猜”那可能是一个中文字。

除了GB 2312—1980，其余两套编码标准也都采用了扩展ASCII的方法，但是这些编码互不兼容，而且编码区间也各有不同。因为不兼容，在同一个系统中同时显示GB（国家标准）和Big5基本上是不可能的。我们在早期的一些计算机输入软件中能发现，它们在自动识别中文编码、自动显示正确编码方面都做了很多努力。

由于GB 2312—1980只收录了6763个汉字，有不少汉字如人名（“镨”）、古汉语等方面出现的罕用字、部分在GB 2312—1980推出以后才简化的汉字（如“啰”）、中国台湾及中国香港使用的繁体字、日语及朝鲜语等，并未收录在内。