

串行通信 C 程序员指南

(第二版)

[美] Joe Campbell 著

徐国定 廖卫东 张 庆 译

吴洪来 赵 军 审校

清华大学出版社

目 录

序 言	(1)
-----------	-----

第一部分 串行通信基础

第 1 章 ASCII 字符集 (3)

1.1 上下文意义	(3)
1.1.1 机器的上下文意义:指令集	(4)
1.1.2 人类上下文意义:字符集	(4)
1.2 ASCII 字符集的位	(4)
1.2.1 “ASCII”的歧义性	(5)
1.2.2 ASCII 码表	(5)
1.3 图形字符	(7)
1.3.1 数字字符	(8)
1.3.2 拉丁字母表	(9)
1.3.3 特殊字符	(10)
1.3.4 ASCII 码排序序列	(13)
1.4 控制字符	(13)
1.4.1 物理设备控制字符	(15)
1.4.2 逻辑通信控制字符	(17)
1.4.3 物理通信控制字符	(18)
1.4.4 信息分隔符	(19)
1.4.5 用于代码扩展的控制字符	(20)
1.4.6 控制字符的灾难	(21)
1.4.7 控制字符的图案表示	(22)
1.4.8 ANSI X3.64:控制代码扩展	(22)
1.4.9 ANSI X3.64 控制代码格式	(26)
1.5 控制序列前导符	(26)
1.5.1 用 ANSI X3.64 编程	(27)
1.6 换 行	(28)
1.7 一个合理的建议	(29)
1.7.1 与行结束相关的词汇表	(29)

第 2 章 异步通信技术基础 (30)

2.1 电子通信的历史	(30)
2.1.1 早期的并行系统	(30)
2.1.2 串行二进制系统	(31)

2.1.3	早期的打印电报	(32)
2.1.4	五位代码	(33)
2.1.5	机器自动编码和解码	(34)
2.1.6	同步化	(36)
2.1.7	为什么要用五位代码?	(40)
2.1.8	ASCII 码的传送	(41)
2.1.9	串行术语	(41)
2.2	通信线路的用法	(43)
2.2.1	同步与异步串行通信的比较	(43)
2.3	小 结	(44)
第 3 章	错误及错误检测	(45)
3.1	错误起源	(45)
3.2	错误检测	(45)
3.2.1	冗余位	(46)
3.2.2	块冗余:奇偶校验	(47)
3.3	循环冗余校验(CRC)	(49)
3.3.1	模-2 算术运算	(49)
3.3.2	普通写法的模-2 除法	(50)
3.3.3	模-2 除法与硬件	(51)
3.3.4	清除余数寄存器	(54)
3.3.5	典型的 CRC 电路	(55)
3.3.6	CRC 和多项式	(55)
3.3.7	选择生成器多项式(除数)	(56)
3.3.8	获得零余数	(57)
3.3.9	对累加器清零的另一种考虑	(57)
3.4	CRC 的各种变形	(59)
3.4.1	典型 CRC	(60)
3.4.2	前导零	(60)
3.4.3	一字节数据的 CRC	(60)
3.5	小 结	(62)
第 4 章	信息传输	(63)
4.1	流控制	(63)
4.2	软件流控制过程	(63)
4.2.1	逐个字符的流控制过程	(64)
4.2.2	行流控制	(66)
4.3	流控制协议	(67)
4.3.1	用于硬件设备的流控制协议	(67)
4.3.2	字符协议	(67)
4.3.3	整行协议	(69)
4.3.4	文件传送协议	(69)

4.4 自动重复请求(ARQ)协议	(70)
4.4.1 发送并等待 ARQ	(70)
4.4.2 连续 ARQ	(71)
4.5 信息包	(71)
4.5.1 分隔信息包的控制字符	(71)
4.5.2 限定信息包的区段长度	(72)
4.5.3 数据区段固定长度的信息包	(72)
4.6 XMODEM 协议	(73)
4.6.1 XMODEM 的技术说明	(73)
4.6.2 XMODEM 发送	(73)
4.6.3 XMODEM 接收	(75)
4.6.4 XMODEM-CRC	(77)
4.6.5 超时处理	(78)
4.6.6 XMODEM 的有关问题	(79)
4.6.7 多文件 XMODEM	(80)
4.7 Kermit 简介	(81)
4.7.1 Kermit 协议	(83)
4.7.2 在 Kermit 信息包内控制字符的编码	(85)
4.7.3 在 Kermit DATA 区段中数据字节高阶位的编码	(87)
4.7.4 在 Kermit DATA 区段中重复计数的编码	(87)
4.7.5 Kermit 信息包中的区段	(88)
4.7.6 Kermit 信息包的类型	(89)
4.7.7 一个 Kermit 通信会话范例	(91)
4.7.8 终止一个传送过程	(91)
4.7.9 Kermit 的扩充	(93)
4.7.10 连续 ARQ	(94)
4.7.11 加长信息包	(94)
4.7.12 如何选择	(96)
4.8 局部链路协议	(96)
4.9 小 结	(97)
第 5 章 调制解调器及其控制	(99)
5.1 调制解调器	(99)
5.1.1 调制解调器基础	(99)
5.1.2 调制	(100)
5.1.3 通信方式或带宽用法	(102)
5.1.4 频率调制	(104)
5.1.5 带宽限制	(106)
5.1.6 相位调制	(107)
5.1.7 微分相移键控	(109)
5.1.8 正交调幅	(110)
5.1.9 握手:建立数据链路	(110)
5.1.10 带宽增加	(114)

5.1.11	ECM:回波抵消多工系统	(114)
5.1.12	网格编码	(115)
5.1.13	V.32	(115)
5.1.14	V.32bis	(117)
5.1.15	V.FAST	(117)
5.2	调制解调器控制	(117)
5.3	RS-232 标准	(118)
5.3.1	交换电路的法定功能说明	(120)
5.3.2	电信号特征	(122)
5.3.3	接口电路的机械描述	(123)
5.4	实际的 RS-232	(124)
5.4.1	调制解调器和 RS-232 接口	(125)
5.4.2	微机接口	(129)
5.5	灵巧型调制解调器	(130)
5.5.1	调制解调器和流控制	(131)
5.6	RS-232 的非标准用法	(134)
5.6.1	空(Null)调制解调器	(134)
5.6.2	连接非调制解调器设备	(135)
5.7	小 结	(136)

第 6 章 UART:一个概念上的模型 (138)

6.1	软件异步 I/O	(138)
6.1.1	软件异步输出	(138)
6.1.2	软件异步输入	(139)
6.2	UART 的介绍	(142)
6.2.1	串行数据时钟	(142)
6.2.2	UART 发送器	(143)
6.2.3	UART 接收器	(145)
6.2.4	错误检测	(145)
6.2.5	接收器同步	(147)
6.3	数据格式	(150)
6.3.1	奇偶校验	(150)
6.3.2	数据位个数	(151)
6.3.3	停止位个数	(152)
6.3.4	“送 BREAK”位	(152)
6.4	RS-232 接口	(152)
6.4.1	RS-232 输出寄存器	(153)
6.4.2	RS-232 输入	(153)
6.4.3	握手信号	(154)
6.4.4	RS-232 状态寄存器	(154)
6.4.5	RS-232 输出控制寄存器	(154)
6.4.6	RS-232 反相逻辑	(155)
6.5	UART 中断	(155)

6.5.1 生成中断	(156)
6.5.2 中断向量的确定	(156)
6.6 FIFO 式 UART	(157)
6.6.1 过速/欠载	(157)
6.7 中断等待	(158)
6.7.1 中断饱和	(158)
6.8 块结束问题	(159)
6.9 一个理想 FIFO 的 UART	(159)
6.10 小结	(159)
第 7 章 实际的 UART	(160)
7.1 National 8250/16450	(160)
7.2 8250 硬件基础	(161)
7.2.1 8250 时钟和同步	(162)
7.3 8250 内部结构	(162)
7.3.1 8250 内部寄存器寻址	(162)
7.3.2 8250 寄存器用法概要	(164)
7.3.3 8250 上的中断	(169)
7.3.4 中断服务例行程序	(170)
7.4 National 16550 UART	(171)
7.4.1 16550 中断标识寄存器	(172)
7.4.2 线路状态寄存器	(173)
7.4.3 FIFO 控制寄存器	(173)
7.4.4 FIFO 中断模式操作	(173)
7.4.5 接收 FIFO	(174)
7.4.6 发送中断操作	(174)
7.4.7 查询操作中 16550 的使用	(174)
7.5 Zilog Z80SIO 串行输入/输出控制器	(174)
7.6 Z80SIO 与 8250 比较	(175)
7.6.1 寄存器寻址	(175)
7.6.2 FIFO	(175)
7.6.3 DELTA 状态位锁存	(176)
7.6.4 向量中断	(176)
7.6.5 Z80SIO 附加特征	(176)
7.7 Z80SIO 硬件基础	(176)
7.7.1 数据寄存器	(178)
7.7.2 控制/状态口寻址	(178)
7.7.3 Z80SIO 中断	(180)
7.7.4 其余寄存器概述	(184)
7.8 小结	(186)
第 8 章 baseline 灵巧型调制解调器	(187)
8.1 灵巧型调制解调器的灵巧之处何在?	(188)

8.2	Hayes Smart 调制解调器的简史	(189)
8.3	调制解调器状态	(189)
8.3.1	命令状态	(189)
8.3.2	拨号状态	(190)
8.3.3	握手状态	(190)
8.3.4	在线状态	(190)
8.3.5	在线命令状态	(190)
8.3.6	哑终端方式	(190)
8.4	命令语法和调制解调器响应	(190)
8.4.1	命令语法	(190)
8.4.2	命令执行时间	(191)
8.4.3	按键退出	(192)
8.5	软挂起:在线命令状态	(192)
8.5.1	在线转义的危险	(193)
8.6	对命令的响应	(193)
8.7	硬件问题	(194)
8.7.1	自动波特率调整	(194)
8.7.2	速率和数据格式	(195)
8.7.3	非易失性存储器	(195)
8.7.4	调制解调器的 RS-232 接口	(196)
8.7.5	前面板	(197)
8.7.6	后面板	(197)
8.7.7	配置开关	(197)
第 9 章	调制解调器命令	(200)
9.1	存根命令	(200)
9.2	调制解调器命令	(200)
9.3	方式命令	(201)
9.3.1	用户接口命令组	(201)
9.3.2	基本拨号和应答命令组	(204)
9.3.3	拨号修改标志组	(205)
9.3.4	拨号变量命令组	(208)
9.3.5	其他命令组	(210)
9.3.6	Profile 管理命令组	(213)
9.3.7	电话硬件控制命令组	(213)
9.3.8	RS-232 命令组	(215)
9.3.9	连接性选择命令组	(216)
9.4	数字变量命令	(217)
9.4.1	S 寄存器字符变量	(218)
9.4.2	计数/定时用 S 寄存器变量	(219)
第 10 章	协议调制解调器	(226)
10.1	调制解调器协议和协议调制解调器	(226)

10.1.1	调制解调器协议	(226)
10.1.2	背景	(226)
10.1.3	调制解调器协议的简短回顾	(227)
10.1.4	网络模型	(228)
10.1.5	OSI 七层网络模型	(228)
10.2	究竟什么是一个链路协议?	(230)
10.2.1	同步还是异步?	(230)
10.2.2	体系结构	(230)
10.2.3	数据透明性	(231)
10.2.4	数据等待时间	(231)
10.2.5	协议效率	(233)
10.2.6	BREAK 处理	(233)
10.3	MNP 协议	(234)
10.3.1	标题	(234)
10.3.2	信息	(234)
10.3.3	帧校验序列	(235)
10.3.4	BREAK 处理	(235)
10.4	MNP 异步协议:1类和2类 MNP	(235)
10.4.1	信息包引导符	(235)
10.4.2	标题字段	(235)
10.4.3	信息字段	(235)
10.4.4	信息包终结符序列	(235)
10.4.5	帧校验序列	(235)
10.4.6	协议效率	(236)
10.5	MNP 同步协议:3类和4类 MNP	(236)
10.5.1	信息包引导符/终结符序列	(236)
10.5.2	标题字段	(237)
10.5.3	帧校验序列	(237)
10.5.4	协议效率	(237)
10.6	LAPM(V.42)	(237)
10.6.1	信息包引导符/终结符序列	(238)
10.6.2	地址字段	(238)
10.6.3	控制字段	(238)
10.6.4	信息字段	(238)
10.6.5	BREAK 处理	(238)
10.7	压缩理论	(238)
10.7.1	重复	(240)
10.7.2	统计重复:哈夫曼编码	(241)
10.7.3	历史重复:Lempel-Ziv	(242)
10.7.4	5类 MNP 压缩规程	(242)
10.7.5	V.42bis BTLZ 压缩规程	(243)
10.7.6	V.42bis 和 5类 MNP 之间的比较	(247)
10.8	流控制	(247)

10.8.1 流控制与数据压缩	(248)
10.9 协议调制解调器及其命令	(250)
10.10 &Q 协议启动	(250)
10.11 S46:协议及压缩规程选择	(251)
10.12 S48:特性协商	(251)
10.13 S36:协商退却(Fallback)	(252)
10.14 &K:流控制	(252)
10.15 线路速度控制	(253)
10.16 Wn 和 S95:扩展响应	(253)

第 11 章 传真机 (256)

11.1 T.30:传真通信协议	(258)
11.2 HDLC 信息包	(259)
11.3 传真字段	(261)
11.4 成串信息包	(262)
11.5 同步线路控制	(262)
11.5.1 轮询	(262)
11.5.2 X 位(X-Bit)	(263)
11.6 传真的五个阶段	(263)
11.6.1 阶段 A:呼叫建立	(263)
11.6.2 阶段 B:识别和协商	(264)
11.6.3 阶段 C:数据(“报文”)传输	(265)
11.6.4 阶段 D:页后过程	(265)
11.6.5 阶段 E:呼叫释放	(266)
11.7 传真过程的实例描述	(266)
11.7.1 实例描述一:单页传真	(266)
11.7.2 实例描述二:规格相同的多页传真	(268)
11.7.3 实例描述三:不同规格的多页传真	(268)
11.7.4 实例描述四:轮询方式单页传真	(271)
11.7.5 实例描述五:在 G3 训练期间降低速度	(273)
11.7.6 实例描述六:在 G3 训练期间提高速度	(274)
11.8 DIS/DCS 位映象	(275)
11.8.1 向后兼容性和可扩展性	(275)
11.8.2 新的 FCF	(275)
11.8.3 最小性能集合	(276)
11.8.4 DIS/DCS 信息包的逐位解释	(279)
11.9 T.4:传真图象协议	(282)
11.10 分辨率	(283)
11.10.1 1992 年的 T.4 建议	(284)
11.11 文件尺寸	(285)
11.11.1 页面尺寸	(285)
11.12 传真编码	(289)
11.12.1 一维编码(改进型哈夫曼编码)	(289)

11.12.2 二维编码(READ 编码)	(294)
11.12.3 编码方式综述	(297)
11.12.3 线终码	(297)
11.12.4 页编码	(299)
11.13 差 错	(300)
11.13.1 纠错	(301)
11.14 EIA 传真调制解调器	(302)
11.15 EIA 578(Class1)	(304)
11.15.1 Clas-1 服务	(305)
11.15.2 命令综述	(306)
11.16 会话实例	(309)

第二部分 用 C 语言编写异步通信程序

第 12 章 设计一个基本的串行 I/O 库 (314)

12.1 Microsoft C 编译器	(314)
12.1.1 SIOLOCAL.H 文件	(314)
12.2 串行 I/O 库	(317)
12.2.1 函数库的层次结构——分级	(319)
12.2.2 用于屏蔽状态寄存器的常量	(321)
12.2.3 UI6x50.LIB	(322)
12.2.4 2 级库;BUOS.LIB	(325)
12.2.5 3 级库;SIO.LIB	(326)
12.3 终端模拟程序的第一个版本 TERM 0	(327)
12.3.1 控制台 I/O	(328)
12.4 TERM0	(329)
12.5 连接 TERM0	(331)
12.5.1 0 级模块;IBMPC.C	(331)
12.6 小结	(332)

第 13 章 程序的可移植性 (333)

13.1 1 级函数	(333)
13.1.1 结构中的函数指针	(334)
13.1.2 含有指向 UART 读/写函数的指针的结构 sio	(335)
13.1.3 修改后的 UART.LIB 函数	(336)
13.1.4 内存映像系统中的指针	(337)
13.2 SIO 数据类型	(337)
13.2.1 管理多个 SIO	(338)
13.3 修改 1 级函数	(339)
13.3.1 寄存器存储类型 register	(341)
13.4 修改 2 级函数	(341)

13.4.1	进一步讨论函数指针	(341)
13.4.2	BUOS.LIB 中修改后的 2 级函数	(342)
13.4.3	声明和初始化 SIO	(343)
13.5	为内存映象 UART 提供的 SIO 结构	(346)
13.5.1	定时函数	(346)
13.6	定时函数的类型	(346)
13.6.1	延时	(346)
13.6.2	超时功能	(347)
13.7	系统定时器和时间保持器	(347)
13.7.1	系统节拍器	(347)
13.7.2	系统节拍器的软件接口	(348)
13.8	设计一个虚拟的定时系统	(348)
13.8.1	IBM PC 机的定时程序	(349)
13.8.2	0 级定时函数 delay	(350)
13.8.3	“等待字符”函数	(351)
13.9	为定时参数定义常量	(352)
13.10	UART 清除器	(353)
13.10.1	测试定时函数的程序	(354)
13.11	小 结	(355)
第 14 章 波特率和数据格式函数		(356)
14.1	设计目标	(356)
14.1.1	用户准则	(357)
14.1.2	一般假设	(357)
14.1.3	虚拟寄存器	(358)
14.2	IBM PC 机的数据格式	(361)
14.2.1	16X50 的虚拟寄存器数组	(362)
14.2.2	位操作的通用结构(vregbits...)	(363)
14.2.3	IBM PC 机上的数据格式	(364)
14.2.4	通过 SIO 指针访问虚拟寄存器	(365)
14.2.5	为停止位和数据长度声明的结构	(366)
14.2.6	多寄存器操作	(367)
14.2.7	最终版的 vsetbits 函数	(369)
14.2.8	16X50 的 1 级函数(_vsetbits)	(371)
14.2.9	3 级数据格式函数	(371)
14.2.10	结构定义的位置	(373)
14.3	波特率函数	(373)
14.3.1	vbaud_数据结构	(374)
14.3.2	IBM PC 的波特率数据结构	(375)
14.3.3	设置波特率的 2 级函数	(376)
14.3.4	用于 16X50 的 1 级波特率函数(_vsetbr)	(377)
14.3.5	设置波特率的 3 级函数	(378)
14.4	配置和恢复	(379)

14.4.1 用于 16X50 的设置和恢复函数:s_config 和 s_restore	(379)
14.4.2 用于数据格式、波特率和检错的字符串数组	(381)
14.5 TERM2	(382)
14.6 小 结	(387)
第 15 章 RS-232 输入控制	(388)
15.1 RS-232 输出控制	(388)
15.1.1 RS-232 UART 输出	(388)
15.1.2 TERM3	(397)
15.2 RS-232 输入控制	(400)
15.2.1 VIN232_结构	(400)
15.2.2 RS-232 输入的常量	(401)
15.2.3 VIN232_结构的说明	(401)
15.2.4 第 2 级 RS-232 输入函数(VSTAT232)	(402)
15.2.5 新的 SIO 结构成员	(403)
15.2.6 RS-232 输入状态的第 1 级函数	(404)
15.2.7 第 3 级 RS-232 输入函数	(405)
15.2.8 SIO 说明和初始化	(406)
15.2.9 TERM 的若干改动	(407)
15.3 小 结	(408)
第 16 章 流控制和 SIO 管理	(409)
16.1 流控制	(409)
16.1.1 虚拟流控制	(409)
16.2 端口管理	(413)
16.2.1 打开串行 I/O 设备	(415)
16.2.2 第 3 级 open 函数	(418)
16.2.3 第 3 级关闭函数	(420)
16.3 发送器和接收器函数	(421)
16.3.1 第 3 级中的其他函数	(423)
16.4 TERM 的修订版	(424)
16.5 小 结	(425)
第 17 章 格式输出	(426)
17.1 格式输出	(427)
17.1.1 soctl_结构的说明和初始化	(429)
17.1.2 SIO 结构的新结构成员	(430)
17.2 输出控制函数	(430)
17.2.1 速度考虑	(433)
17.2.2 s_fputc 概述	(435)
17.2.3 “原始”模式	(435)
17.2.4 字节的转换和删除	(436)

17.2.5 控制字符删除	(437)
17.2.6 行结束转换	(437)
17.2.7 被发送字节的处理	(438)
17.3 格式 I/O 的第 3 级函数	(440)
17.3.1 逐个字符的字符串输出	(440)
17.3.2 操作 <code>soctl_</code> 结构成员的函数	(441)
17.3.3 输出控制结构的临时存储	(444)
17.4 TERM4	(446)
17.4.1 传送文件的其他设计方法	(449)
17.4.2 改进建议	(449)
17.5 小 结	(450)
第 18 章 格式输入	(451)
18.1 格式输入函数	(451)
18.1.1 <code>sictl_</code> 结构	(451)
18.1.2 <code>sictl_</code> 结构的说明和初始化	(453)
18.1.3 SIO 结构的新结构成员	(453)
18.2 <code>s_fgetc</code> 概述	(454)
18.2.1 原始处理	(457)
18.2.2 控制逻辑	(458)
18.2.3 超时操作	(458)
18.2.4 行结束符的转换	(458)
18.2.5 “回读”机制	(458)
18.2.6 有关速度的进一步讨论	(459)
18.2.7 回送	(460)
18.3 格式输入的第 3 级函数	(460)
18.3.1 输入控制结构的临时存储	(461)
18.4 带来方便的第 3 级输入函数	(462)
18.4.1 <code>s_fgets</code> 函数	(462)
18.5 TERM5	(463)
18.5.1 数据转储函数 <code>dnload</code> 的设计	(465)
18.5.2 数据获取函数的设计	(468)
18.5.3 读取函数 1: <code>cbuff</code>	(468)
18.5.4 读取函数 2: <code>locbuff</code>	(472)
18.6 小 结	(474)
第 19 章 中断 I/O 导论	(475)
19.1 中断的基础概念	(475)
19.2 中 断	(475)
19.2.1 中断和异步 I/O	(476)
19.3 IBM PC 机的中断结构	(476)
19.3.1 8259 外围中断控制器(PIC)	(477)

19.4 系统中断函数	(478)
19.4.1 SIO 的新结构成员和常量	(478)
19.4.2 主中断处理程序	(480)
19.4.3 整体系统结构	(482)
19.4.4 __sys_introff 函数	(485)
19.5 UART 中断函数	(486)
19.5.1 URAT 中断管理程序	(486)
19.5.2 SIO 中断管理程序 s_iadmin	(487)
19.5.3 UART 中断的再向量化	(488)
19.5.4 摘要	(489)
19.6 对第 3 级函数的改动	(490)
19.6.1 对 s_open 的改动	(490)
19.6.2 对 s_close 的改动	(492)
19.7 小 结	(493)
第 20 章 中断子处理程序	(494)
20.1 串行化出错中断	(497)
20.1.1 SIO 中的串行化出错	(497)
20.1.2 串行化出错子处理程序	(497)
20.1.3 串行化出错的高层接口	(498)
20.1.4 串行化出错的中断控制	(499)
20.2 RS-232 状态中断	(500)
20.2.1 RS-232 中断子处理程序	(501)
20.2.2 RS-232 中断控制	(502)
20.3 检测 16550	(503)
20.3.1 s_config 函数的修改	(504)
20.4 接收器中断	(505)
20.4.1 接收器中断缓冲区的设计	(505)
20.4.2 接收器中断子处理程序	(506)
20.4.3 接收器中断到 SIO 的接口	(508)
20.4.4 接收器中断控制	(509)
20.5 发送器中断	(511)
20.5.1 发送器中断缓冲区的设计	(511)
20.5.2 发送器中断子处理程序	(512)
20.5.3 发送器中断控制	(515)
20.6 发送器中断到 SIO 的接口	(518)
20.6.1 成块写接口	(518)
20.7 TERM6	(522)
第 21 章 灵巧型调制解调器程序设计	(524)
21.1 基本设计标准	(524)
21.1.1 用户接口	(524)

21.1.2	RS-232 控制	(525)
21.1.3	兼容性	(525)
21.1.4	通用性	(525)
21.2	调制解调器的结构和常量	(525)
21.3	调制解调器命令	(528)
21.3.1	命令和响应的结构成员	(528)
21.3.2	发送调制解调器命令:m_cmd	(529)
21.4	获取调制解调器的应答	(530)
21.4.1	取应答函数:s_fgetln	(530)
21.4.2	取应答函数:m_getstr	(532)
21.5	对调制解调器的应答求值	(533)
21.5.1	验证应答函数:m_ismessage	(534)
21.6	高层调制解调器函数	(536)
21.6.1	发送命令,读取应答的函数:m_cmd_str	(536)
21.6.2	发送命令,验证其有效性:m_cmd_OK	(537)
21.6.3	强制调制解调器转入命令状态:m_gocmd	(538)
21.7	重置函数	(539)
21.7.1	重置函数所需的调制解调器结构成员	(539)
21.7.2	重置调制解调器的函数:m_reset	(539)
21.7.3	识别调制解调器:m_whoru	(543)
21.7.4	用于识别所需的结构成员	(543)
21.8	对基线调制解调器进行配置	(545)
21.8.1	配置:m_config 和 m_unconfig	(546)
21.9	协议调制解调器	(548)
21.9.1	波特率转换	(549)
21.9.2	回归到无协议情况	(549)
21.9.3	CONNECT 消息	(550)
21.9.4	配置	(550)
21.9.5	协议配置的结构成员	(551)
21.9.6	配置协议调制解调器:m_protoconfig	(552)
21.9.7	分析调制解调器应答的函数:m_parsernsg	(554)
21.9.8	有关 m_parsemsg 的说明	(556)
21.10	拨号和回答函数	(557)
21.10.1	返回呼叫—进行应答:m_wait4dcd	(557)
21.10.2	自动拨号函数:m_dial	(559)
21.10.3	重拨上一次的号码:m_redial	(561)
21.10.4	回答电话:m_answer	(561)
21.10.5	终止连接的函数:m_hup	(562)
21.10.6	考察数据载波检测(DCD):m_warndcd	(563)
21.10.7	调制解调器函数的字符串数组	(564)
21.11	TERM 6	(565)
21.11.1	TERM6 的一个解释	(568)
21.12	小结	(569)

第 22 章 XMODEM 文件传送	(570)
22.1 文件传送的设计考虑	(570)
22.2 异常处理	(571)
22.3 XMODEM 传送	(578)
22.4 XMODEM 接收	(590)
22.5 校验和出错检测函数	(599)
22.6 结 论	(601)
第 23 章 循环冗余校验 CRC 计算	(603)
23.1 多项式与 CRC 函数	(603)
23.1.1 用硬件和软件实现多项式除数	(603)
23.1.2 模拟 CRC 硬件:crchware	(606)
23.1.3 用查表法计算 CRC	(608)
23.2 为 x_snd 和 x_rcv 提供的 CRC 函数	(611)
23.2.1 发送 CRC 的函数:x_sndcrc	(611)
23.2.2 接收 CRC:x_rcvcrc	(612)
23.3 CRC-16 的计算	(613)
23.3.1 CRC 函数的 CRC 测试程序: CRC.C	(616)
23.4 小 结	(617)
第 24 章 Group 3 传真图象的编码和解码	(618)
24.1 创建一个测试位图	(619)
24.2 T.4 编码	(621)
24.2.1 RUN2T4:把行程编码转换为修订的哈夫曼编码	(624)
24.3 T.4 解码	(630)
24.3.1 T4DECODE:对修订的哈夫曼编码进行解码	(631)
24.3.2 T4TORUN:修正的哈夫曼行程编码	(635)
24.3.3 行程编码转换为位图图象	(636)
参考书目	(641)
1. 期刊	(641)
2. 书籍	(641)
3. 标准及参考文献	(643)
美国国家标准协会的文件	(643)
电子/远程通信工程协会的文件	(644)
CCITT	(644)
附录 A 附带函数	(646)
BIGBUFF	(646)
HAKTC	(647)
PUTBIN	(648)

RFOPEN	(648)
STR_ATOI	(649)
WFOPEN	(649)
附录 B 汇编程序接口和内存模式	(651)
B.1 汇编代码	(651)
B.1.1 汇编程序体:CPGSC.ASM	(651)
B.2 主 ISR:COM1ISR.ASM	(653)
B.3 宏:CPGSC.MAC	(655)
附录 C IBMPC.C	(657)
C.1 SIO 初始化程序:COM1SIO.C	(659)
附录 D 头文件	(667)
ASCII.H	(667)
BUOS.H	(668)
CRC.H	(668)
IBMPC.H	(669)
KEY.H	(670)
LEVELO.H	(670)
MISE.H	(670)
MODEM.H	(671)
SIOCTL.H	(673)
SIODEF.H	(675)
SIOLOCAL.H	(683)
T4.H	(683)
TERM.H	(684)
UART.H	(684)
U16X50.H	(684)
XMOD.H	(687)
附录 E TERM7	(690)
附录 F 传真编码和解码表	(695)
附录 G 函数索引	(706)