CLIPPER 丛书

扩充函数库——通讯篇

(台湾)TSID 工作室 编著

电子工业出版社
天津科学技术出版社
广东科技出版社

(京)新登字 055 号

内 容 提 要

本书作为一本程序设计的书，介绍了 CLIPPER 函数的名称、功能、语法、参数、传回值并展示了其范例程序及源程序。在介绍 CLIPPER 函数及其高级应用的同时，还为您介绍了通讯技术的基本原理，从调制解调器、Cable 线到通讯协定，都循序地做了解剖说明及图示介绍。

为求完整书中所有程序，我们都附上了原始程序，读者可根据需要运用或修改使用。

本书适用于计算机及通讯专业技术人员、大专院校学生阅读。

CLIPPER 丛书

CLIPPER 扩充函数库——通讯篇

(台湾)TSID 工作室　　编著

责任编辑　崔围荣

电 子 工 业 出 版 社

百通公司(集团)　　天津科学技术出版社　　出版发行

广 东 科 技 出 版 社

各地新华书店经销

电子工业出版社计算机排版室排版

北京科技大学印刷厂印刷

*

开本：850×1168 毫米　1/32　印张：12.875　字数：343 千字

1995 年 5 月第一版　1995 年 5 月第一次印刷

印数：3500　定价：20.00 元

ISBN 7-5053-2867-0/TP・941

出版说明

CLIPPER 语言是 dBASE 数据库语言的换代语言之一，该语言兼容 dBASE 数据库语言的绝大部分命令和函数，增加了一批命令和函数，还特别建立了 C 语言的接口，用户可在 CLIPPER 语言程序内用类似函数调用的方式，调用 C 语言程序，使编出的程序兼具 dBASE 数据库语言程序和 C 语言程序两方面的优点，是商业、管理方面多媒体软件开发的实用性语言。

CLIPPER 语言建立了 C 语言的接口，使编程者可调用现成的 C 语言程序，将数据库语言以外的全部操作交由 C 语言去完成，为用户提供了一个吸收其他软件精华的窗口，也为用户编程技巧的发挥提供了广阔的空间。有了这一接口，采用 CLIPPER 语言程序调用 C 语言程序的方法，可以 CLIPPER 语言扩充多种功能，如：屏幕绘图、硬件控制、网络、计算机通讯等。

CLIPPER 语言系统小巧、灵活，计算机只要达到基本配置（286 机型或以上，内存 640KB 或以上，有硬盘）即可使用。这种语言灵活性强、可扩充的功能多样，很适合作为商业、管理方面专项软件的编程语言。在台湾，约有九成应用软件是用 CLIPPER 语言编写的，这些应用软件遍及金融、会计、贸易、仓库管理、进销存系统、人事工资系统、零售和录像带出租管理等方面。

CLIPPER 语言摆脱了传统 dBASE 数据库语言的“运行环境”约束，通过编译和连接，直接将用户程序转换为 EXE 文件，使编写出来的程序速度快、保密性非常好。原来使用 dBASE 数据库语言的读者转用 CLIPPER 语言后，将会发现这样更灵活、更方便。

由电子工业出版社、广东科技出版社、天津科学技术出版社共同合作，从台湾碁峰图书股份有限公司引进本套“CLIPPER丛书”，

包括：

1. CLIPPER 入门与应用
2. CLIPPER 扩充函数库——绘图篇
3. CLIPPER 扩充函数库——硬件篇
4. CLIPPER 扩充函数库——通讯篇
5. CLIPPER 扩充函数库——网络篇(上)
6. CLIPPER 扩充函数库——网络篇(下)

其中第 1 册将 CLIPPER 语言与 dBASE 数据库语言作比较，介绍 CLIPPER 语言的基本概念，CLIPPER 语言程序的编写、编译、运行方法；第 2 册至第 6 册分 4 个专题，分别介绍如何在 CLIPPER 语言程序中调用 C 语言程序，使编出的程序具有屏幕图形、硬件控制、通讯和网络方面的功能。本套图书还可供读者在编写其他语言程序时参考。

我们希望，本套图书能满足广大读者学习 CLIPPER 语言的需要，并将促进海峡两岸计算机软件技术的交流。

电 子 工 业 出 版 社
广 东 科 技 出 版 社
天 津 科 学 技 术 出 版 社

1994 年 10 月

改编者的话

当前微机数据库管理系统 dBASE 由于实用性强、易学、易用等特点，已在我国得到广泛应用。但随着时间的推移，其缺点也越来越明显，主要有计算能力差、处理速度慢等。这些缺点不但限制了 dBASE 的应用范围，也难以满足高水平管理信息系统的开发需要。CLIPPER 作为 dBASE 语言的最新编译型版本，自问世以来，对微机数据库管理系统产生了深远影响。CLIPPER 是一个开发工具，它利用 dBASE Ⅲ Plus 的扩充作为其标准命令集，其命令与函数是 dBASE Ⅲ Plus 的超集，其扩展系统则允许用户存取用 C 与汇编语言编制的例程与函数。

本书对通信基础知识及如何控制和运用调制解调器(MODEM)作了介绍，并提供现成的范例和通信控制函数供读者参考使用。

本书由刘志强负责改编，任天电子信息技术研究所王东、刘得利负责全书的审校。

改编者

1994 年 6 月

前　言

在这个信息爆炸的时代里，信息的接收是人们日常生活中极其重要的一件事。例如，股票的买进卖出就需要随时注意上市厂商的经营动向；又如，现在十分热门的 BBS，在其网络上我们不但可以获取想要的信息，而且还可以作为信息传输的中转站，甚至可用 BBS 来取代目前的邮政系统。但这一切都需要有精密的软硬件配合才能符合人们的日常需要。

如果想要传输或接收信息，电话是最快速也是最传统的工具。因此通信与信息具有密不可分的关系。不论是 BBS 或股票的信息传输，电话线和调制解调器(MODEM)都是必备的工具。本书除了介绍基本的通信知识外，还介绍如何控制和运用调制解调器(MODEM)，并提供现成的范例供读者测试使用。

CLIPPER 已广泛应用于商业和管理领域，并获得了巨大成功，但它不适合于工具软件的开发，而且本身在通信控制方面很欠缺。为满足广大 CLIPPER 使用者的需要，我们特将与通信相关的这类程序以函数形式介绍给读者。

TSID 工作室

目　录

第 1 章　浅谈调制解调器

当前由于使用网络连线增加，电传视频和 BBS 站的兴起，调制解调器的重要性也与日俱增，只要拥有一台个人计算机和一部调制解调器，利用现有的电话线，只要付出与电话费相同的代价就可获取想要的信息，再加上调制解调器价格大幅度降低，个人拥有调制解调器的趋势也就越来越大，下面针对调制解调器的种类、选购以及使用时应该注意的事项，提供一些建议：

■速度

市面上大多数调制解调器都是以速度来分类，从 1200 BPS、2400 BPS 到 9600 BPS 的都有，其中 2400 BPS 最为普遍，因为 1200 BPS 速度太慢，而 9600 BPS 在价格上又太高，在成本与速度上作一选择，所以 2400 BPS 使用最广，以下范例就是以 2400 BPS 的调制解调器为标准。

此外，如果只是用于电子布告栏或电子邮件等少量的信息传递，则低速调制解调器即可，但是如果大量用于发送图形文件、会计统计数据、进出货等数据，尤其是需要与国外连线时，速度就相当重要了，其节省的电话费就等于选购高速调制解调器。

■传输协议

传输协议中最为通用的是 CCITT 和 BELL 两大类，在北美地区

大部份使用 BELL 标准，而欧洲地区则使用 CCITT 标准，一般的调制解调器所提供的传输协议标准都不太一样，采购时应注意其规格。

■传输数据方式

传输数据的方式有同步与异步两种，一般讲 2400 BPS 以下多为同步，4800 BPS 以上必为同步传输，如果用户的系统是两台计算机之间大量文件发送，则最好选用同步方式，若只传输少量数据或需要作交互式通讯，则可考虑异步方式。

■线路类型

一般在线路使用类型上有拨号式(Dial-up)和租用专线式(Leased Line)两种，家用电话就是拨号式，而租用专线式则是两台计算机或工作站之间直接连线，而不需通过电信局。

■外型分类

从外观上来分类，可分为内接式、外接式和掌上型(pocket)，内接式只是一块卡直接插入计算机的扩充槽，其缺点是没有任何指示灯。而外接式则需由 RS-232 连接，并有显示灯可看出调制解调器的状态。掌上型调制解调器则随 NOTEBOOK 和 LAPTOP 等便携式计算机潮流，选购时可根据不同的需要而作决定。

■MNP 功能

另外值得一提的是 MNP 功能，它多了数据压缩和错误更正，在高速传输或电话线质量不是非常好的情况下，有必要选择 MNP 功能以增加信息的正确性和传输效率，MNP 可分为 1 到 7 级，各有不同

的协议，等级越高效率越好。

■COM PORT 问题

使用调制解调器时最常出现的问题是 COM PORT 相冲突，在个人计算机上可能接了鼠标和打 J机等其它的外部设备，所以在通讯软件和调制解调器的 COM PORT 选择一致的情况下，还要注意不要与其它设备冲突。此外，有些内接式调制解调器卡提供了 4 个 COM PORT 以供选择。最后提到一点，既然是两端通讯，则两页的速度、错误检查位、数据长度、停止位等选择必须一致，而这些在通讯软件上都很容易找到并加以设置。

现在，我们来研究一下计算机如何与调制解调器连线，计算机通过标准 RS-232 串行口控制调制解调器的通讯能力，通常数据的发送有两种方式，一是并行的，需要有 8 条数据线及两条控制线，数据传递在同一时间将数据线上所有的位同时送出，属于并行通信，这种方式需要很多条线；若将数据及控制信息一次一个位发送，就只需要一条线及一条地线，也就是所谓的“串行通信”，其速度较并行通信慢得多。使用串行通信可省下很多线的设备，利用串行通讯协议和接口，任两个设备就可利用已有的电话线进行交谈。

串行通讯协议最重要的问题在于所有发送和接受一个字符信息所需的数据和控制信号，都是通过一条数据线由一端发送到另一端，传输速率是以每秒发送多少位来计算，单位为“波特率”。

数据发送以位为单位，时间长度则由波特率决定，当前调制解调器的速度则从 300，1200，4800，9600 BPS 不等，即每秒钟可发送 300 至 9600 位。

通信线上没有数据发送时，此线维持在逻辑 1 的状态或称为“记号状态”。当我们准备要发送一个字符数据时，第一个发送的位称为“起始位”，联机起始位是用逻辑 0 的状态或“间隔状态”来代表，起始位的时间长度则由波特率来决定。当接收器发现联机信号由记号状

态变成间隔状态，即由 1→0，即接收到起始位时，它就准备接收一个完整的字符信息。

紧接在起始位之后的是组成正在传输的字符信息的“数据位”，数据位的数目可以设置为 7 或 8 个，但在相同的传输中，每个字符都须包含相同的数据位数目。由于多数情况下，发送的字符信息都少于 8 位。通常 ASCII 类型的文件，在发送时可设置 7 个字符，而 BINARY 类型的文件，则必须设置为 8 个字符。因为数据位数并不固定，而利用不固定的数据位数，数据传输的速率可以达到最佳，数据位先从最小有效位开始发送，到达接收端再重新组合成所发送过来的字符数据。

一个可有可无的校验位(parity bit)紧接在数据位之后，整个传输过程中所采用的校验位类型必须一致，如果选用偶校验位(even parity)，则数据位和检验位中逻辑 1 的位总数必须为偶数；如果用奇校验位(odd parity)，则逻辑 1 的位总数必须为奇数 1 而校验位也可不设置此校验位，这样能使接收器检测出某类型的传输错误。

在校验位(假如没有设置，即在数据位)之后，还有 1 或 2 位的记号(MARK)，这些位称为“终止位”(STOP BITS)。终止位代表此线在下一个起始位出现之前必须处于记号状态的最短时间。在整个传输过程中，终止位的数目也必须一致，如果有另一个字符要接着发送，则其起始位必须接在终止位之后；如果没有其它的字符需要发送，则此线将一直维持在记号状态，直到有其它的字符要发送为止。

因为在前面所讨论的串行通信协议中，数据位可以在任何时间开始或停止到达，因此便称之为“异步传输协议”(Asynchronus Protocol)，就是在上一个字符的终止位接收到之后，接收器要等到下一个字符的起始位到达之后才开始接收其它更多的数据，在此之前，通信线一直假定是在记号状态，这和“同步串行通信传输协议”(Synchronous serial communication protocol)是不同的，因为在同步协议中，数据字符一直在通信线上发送。

在异步协议中，接收器在接收到新字符之后就必须和发射器同

步，由于接收器和发射器的波特率是相同的，因此接收器可以将起始位、数据位、校验位、终止位等当作时间函数从通信线上拾取出来，而起始位的开始便可用来当作同步之用，假设发射器和接收器的时钟脉冲有一些误差，可能就会因为发射器和接收器在每一个字符的开头都重新同步一次而不致发生错误，因此若要保持数据接收成功，发射器和接收器的数据位数、校验位的类型和终止位的数目都必须相同才行。

■UART

若要在异步串行通信环境下，编写一个程序来发送或接收数据字符是一件相当复杂的工作，但幸运的是，有一个叫做“通用异步接收发送器”(Universal Asynchronous Receiver Transmitter)的微处理机或UART专门用来执行大部份串行通信协议的转换(conversion)工作。计算机AT，AT里面所用的UART就叫8250异步通信元件(Asynchronous communication Element)。

UART的功能标在图1中，UART可以做“并行/串行”或“串行/并行”的转换，并且直接发送和接收串行数据。在使用UART之前，必须先告诉它(利用80x86 CPU里的OUT指令)所要的传输速度、数据位数、校验位的类型以及终止位的数目，只是UART以所要的串行协议规划好之后，即可以用来发送和接收串行数据。

发送一个字符的数据时，先检查UART的状态(利用80x86 CPU里的IN指令)，看看“发送保持寄存器”(transmitting holding register)是否空(位5是否为1)，如果空，就可以输出一个字节的数据到UART，此字节即放在发送保持寄存器之中。当UART已经发送完上一个字符或当前并不发送任何的数据时(输出线为逻辑1)，发送保持寄存器的内容即可置入“发送器位移寄存器”(transmitter shift register)之中，此时发送保持寄存器可以再从计算机输入另一个字符。如果协议只要求数据位数目比8还少时，则仅使用发送寄存器的低

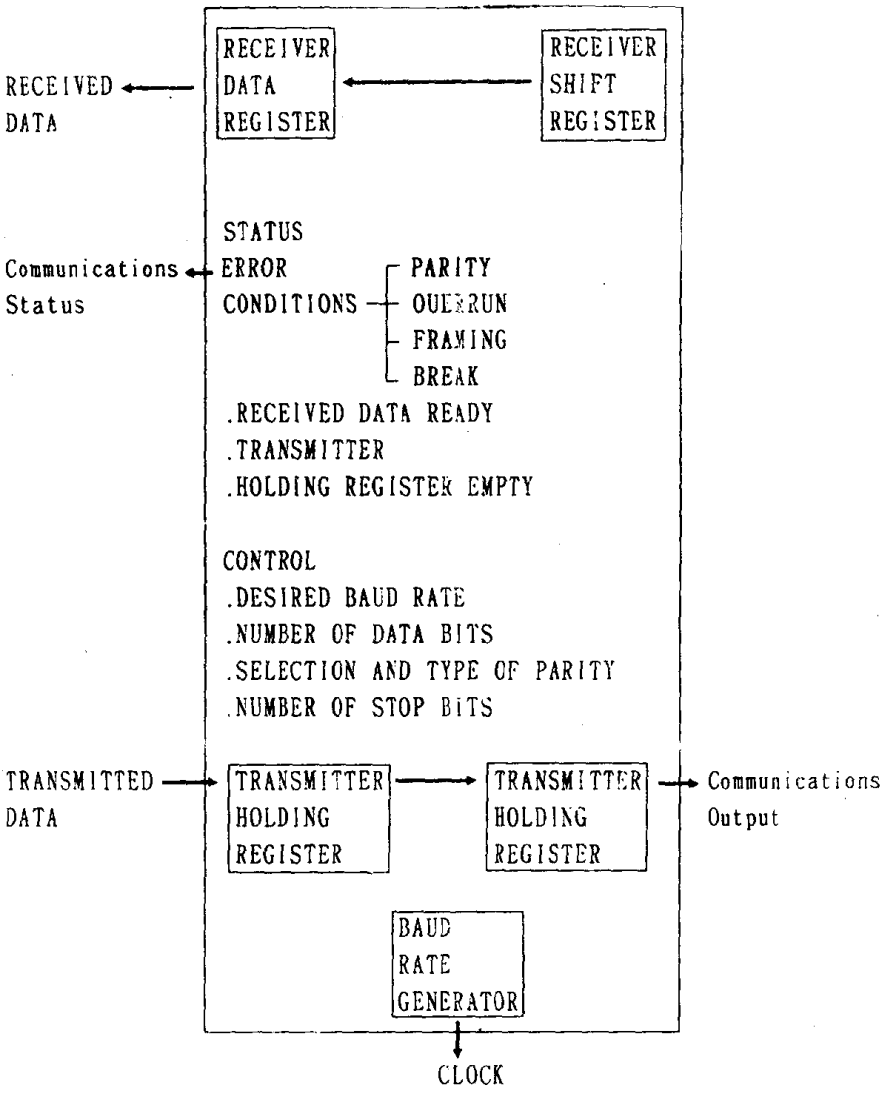


图 1　UART 的功能

位部份，UART 会自动给发送器移位寄存器里的字符加上正确的起始位、校验位以及终止位，然后再将整位发送到串行通信线上。

因此，只要 UART 已经设置好，我们所要作的只是令一个字节数据输出到 UART 即可，这个字符会自动加上所有正确的特征位(characteristics)而连续发送出去，当传输工作正在进行时，计算机可以做其他工作，并周期性地去检查以确定是否发送保持寄存器已经就绪，可以接受另外的输出字节。

UART 将它所接收到的每一个连续输入放到“接收器位移寄存器”(receiver shift register)上，在收到正确数目的终止位并执行完校验位之后，此位便放入“接收器数据寄存器”(receiver data register)中，假如使用低于 8 位的数据，则只有接收器寄存器的低位部份(7 位)是正确的数据，位 8 则以逻辑 0 代替，然后 UART 便更新其状态(status)，以指示接收数据已经就绪，当“接收器数据就绪”(receiver data-ready)状态正确无误之后，计算机便可以从 UART 中读取另一字节的数据，当计算机读完数据后，UART 要一直等到将另一个字符放进接收数据寄存器之后，才会再发出接收器数据就绪信号。

UART 使串行数据的接收变得非常简单，我们所要做的只是监视一位的状态，等到此状态指示时间正确了，就输入一个字节的数据，UART 同时也告诉我们输入的字符是否发生错误，当我们检查 UART 的状态是否是数据就绪时，如果数据就绪，这个状态也可以反应出任何可能的输入错误。

假如 UART 必须在计算机读取第一个字符之前开放第二个字符到接收器数据寄存器，结果便产生一个“超越错误”(overrun error)，这种错误可能在计算机没有频繁检查接收器就绪状态时发生，假如所使用的协议要求 7 位数据、校验位和一个终止位，则将传输速度(波特率)除 10，我们就可以计算出 UART 每秒放入接收器数据寄存器字符数的最大值，如果程序没有以每秒高于那么多次地监视 UART 状态，就可能发生超越错误。

有时因为传输线问题，可能使发送中的数据其中某个位在接收器收到它们之前发生改变，例如一条有杂音的电话线就会发生这种问题。如果数据位有任何改变，则 UART 重新组合出来的字符便不

正确，由于校验位应该表示数据位中逻辑 1 的位数目，因此 UART 可利用他们来检查这种问题，假如检查到某个字符有校验位错误(Parity error)时，则至少必需重新发送所接收数据的一部份才行。

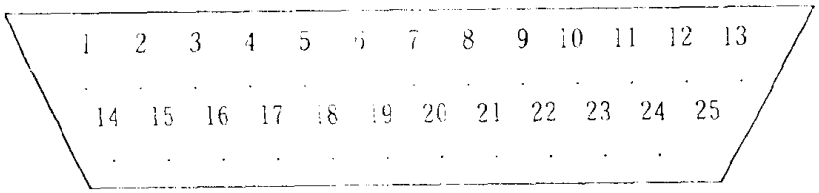
如果 UART 期待收到一个终止位而却未收到时，便产生“结构错误”(frame error)。这种错误可能有很多个原因，例如接收的 UART 其原始设置方式可能和发送的 UART 不适合，或接收器和发送器的时钟脉冲在整个字符组合之前，刚好发生足够的不同相位而引起的同步失调或丧失(miss of Synchronization)，或传输线上的噪音(noise)可能改变终止位的接收等，在任何情况下，所采取的补救方法都和补救奇偶检验错误或结构错误一样。

某些异步串行通信协议允许传输某种特殊状态，称为“终止状态”(break condition)。要送出一个终止状态，传输线必须先保持在间隔状态(spacing condition)，即逻辑 0 的状态一段时间才行，这段时间的长度最少要能让它发送出一个字符及其相伴随的控制位，UART 将在它的状态输出中反映出接收到终止状态的信息。

假如我们必须时时检查 UART 的状态以判断是否已经有一个字符就绪准备输入，或 UART 已经就绪准备接收另一个字符以输出，则须编写很多的通信程序。为了处理这个问题，UART 提供了一个“程序控制中断”(programmed interrupt)，我们可以控制 UART 使它在有一个字符就绪准备接收，或有一输入错误，或传输保持寄存器空时，提供一个中断信号。因此，程序便没必要时刻监视 UART 的状态，而可以做其他工作，并且当时间不正确时，使程序强制去和 UART 打交道。

现在我们对 UART 的工作方法和它可能的应用方式已经有了一个完整的认识，下面我们来详细研究 8250 异步通讯元件，同时提供一个简易的执行串行通信的程序。

个人计算机上的通信接收器提供了一个标准 RS-232 接口以和外界连接，从接收器出来的插头为一个标准的凸形 25 引脚“D”形插头(male 25-pin connector)，下图显示了这个插头及其引脚的编号。



RS-232 接口将 UART 的电信号变成标准的 EIA 电压而出现在插头上，原来代表一个间隔状态的逻辑 0 以＋3 到＋15 伏特的电压来代表，表示记号状态的逻辑 1 则以－3 到－15 伏特的电压来代表，同时，这个接口也能够将输入的电压转换成正确的二进制电信号给 UART 使用。

下表列出了 RS-232 插头的引脚配置，当通电并启动通信接收器之后，就应该可以在第 2 和第 7 脚之间测到一个负电压，当通信线不传送数据时此引脚应保持在记号状态，这个接口可以不必更改引脚的相关位置而直接插到调制解调器上。

标准的 RS-232 引脚配置表

引脚号码	方　向	功　　能
2	输出	Transmit Data
3	输入	Received Data
4	输出	Request To Send
5	输入	Clear To Send
6	输入	Data Set Ready
7		Signal Ground
8	输入	Data Carrier Detect
20	输出	Data Terminal Ready
22	输入	Ring Indicator

但是，假如要将两台个人计算机相连，就不能仅仅将其中第 2 脚接第 2 脚.....等而已，如果我们这样做，则两个发送数据的引脚