

微计算机应用

APPLICATION
ON MICROCOMPUTER



中国科学院声学研究所
微计算机应用编辑部

1980

8

致 读 者

微型计算机是当代科学技术的重要新成就，应用非常广泛，目前已深入到各个领域。《微计算机应用》杂志主要介绍国内外有关微型机的新应用，新动向，各种实用参考手册，微计算机语言以及各种有关参考资料。

《微计算机应用》自创刊以来，受到了全国有关单位的热情欢迎和支持。全国各省市、自治区、包括科研单位、高等院校、工厂企业、各军兵种、农业、商业等各行各业都来函订购，畅销全国。编辑部还收到许多单位和个人的来信和稿件。

为满足全国各兄弟单位推广新技术，加速四个现代化的迫切要求，本刊经上级主管部门批准，从1980年第五期开始，改为国内外公开发行。1981年定为季刊（增刊除外）。热烈欢迎国内外读者订阅和投稿，提出宝贵意见，以便不断提高刊物的质量和服务效果。

中国科学院声学研究所

《微计算机应用》编辑部

目 录

TRS—80 RS—232—C 接 口	(1)
Z—80 指令详解 (續第七期)	
二、Z—80 指令详解 (續)	(29)
5. 通用算术运算及 CPU 控制指令组	(29)
6. 十六位算术运算指令组	(40)
7. 循环移位和移位指令组	(50)
8. 位操作指令组	(73)
9. 转移指令组	(85)
10. 调用及返回指令组	(96)
11. 输入和输出指令组	(104)
三、附 录	(117)
A. 数制基本变换码	(117)
B. ASC II 码 (7 位)	(118)
C. 采用 Z—80 的微计算机一览表	(119)
D. Z—80 指令索引 (以英文字母为序)	(120)
〈国外声学〉征订启事	(39)
简	(49)

TRS-80 RS-232-C 接口

“接口”是什么？

这是在各位拥有的 TRS-80 和任何外部设备间进行通讯时最广泛使用的设备。该接口虽然合乎和数据识别、数据传送速度、发送/接收顺序、故障检查等有关的规程，但并不包含欲利用某个特定的外部设备系统的程序设计。

例如，即使配置了这种接口，并通过接口在诸如行式打印机上输出，也不能做到从某个 TRS-80 向其地于 RS-80 自动地传送 BASIC 程序。必须有“驱动程序”，才能这样使用。这种程序必须单独设计以适合于预定使用的设备。

在本手册中，附有两种驱动程序的文本。其中之一程序给各位的 TRS-80 附加了作为向普通行式打印机提供程序给 TRS-80 提供了向普通的串行接口行式打印机 (DEC 行式打印机) 提供程序。

另外，RS-232-C 接口的设计虽然符合 EIA (电子工业协会) 规格，但是，即使是行式打印机与 RS-232-C 有互换性，的设备也未必能完全地、很好地与其连接和通讯。在这种情况下，或者在关于各位用户自己的使用设备，这里难以提供硬件和软件的支持。

不过，本接口设备规定能和装用 Tandy Radio Shack 公司的 RS-232-C 任何机器相联接。

序

“RS-232-C”是表示 EIA 的特定规格的用语。在该规格中，定义了数据终端设备和数据传送设备连接时，一般的使用方法。本 RS232-C 接口作为数据处理装置连接用的标准设备，已蒙各方面广泛使用。特别是在大多数图象终端设备，调制解调器，卡片输入机，行式打印机，小型机、微型机等方面，被用于设备间的数据交换。

当 TRS-80 扩展接口中安装上 RS-232-C 时，在互换性方面就开辟了一个全新的领域。本设备是为安装在 TRS-80 扩展接口的内部而设计的。串行行式打印机，调制解调器，图象终端设备等能用作外部设备。总之，和 RS-232-C 有互换性的设备，各位用户都能和其连接。

此外，如果使用高度利用 RS-232-C 接口功能的应用程序，就是利用电话接口，也能把各位的 TRS-80 和分时计算机系统连接起来。名为 TERM 的程序以磁带形式附于 RS-232-C 接口。

TRS-80 的许多用户希望用 LEVEL I BASIC 的 LPRINT 命令来操作自己的串行打印机，那么，只要把这种 RS-232-C 接口安装在扩展接口中，就能实现之。本手册中附有实现这种应用的程序示例和文本。

数字数据的傳送

在比较远距离的传送数字数据时，一般采用串行式的数据发送技术。此时，可用结构简单的双绞线连接发送和接收设备。通常，在所谓“非同步式”和“同步式”两种传送技术中，选择使用一种。在 *Radio Shack* 的系统中，使用非同步的位串行式技术。关于同步式技术，请允许不在此处涉及。在非同步传送时，没有必要和数据一起发送同步时钟，字符也没有必要是连续的。因此，对各字符间的间隔长度没有限制。

如图 1 所示，数据字符（通常是 5~8 位长）和表示起始码以及停止码的一个位组构成了一个字符。起始码用逻辑 0 表示，位于各字符的头部；停止码用逻辑 1 表示，位于各字符的尾部。停止码在后续字符的起始码被传送以前，一直保持着。停止码的长度没有上限，可是有下限，这取决于系统的特性。典型的数值是 1.0, 1.42, 2.0 数据位间隔（但是，在最新的系统中，采用 1.0 或 2.0 停止位）。起始码往负向变位，表示传送中的字符内的数据位的位置。接收设备的时钟源根据这种变位被复位。通常使用各数据位的中央部位。

采用非同步式数据传送系统的理由如下：由于没有必要和数据一起发送时钟讯号。所以设备非常简单。另外，由于字符也可在一段时间内不发送，所以能根据需要来传送。这种特性，在操作员通过手工操作从必要的输入设备（例如键盘等）传送数据的情况下，特别有效。非同步式传送的最大缺点在于起始码和停止码占有相当长的通信带区。使用这种起始/停止码的方式称为“起停式”。

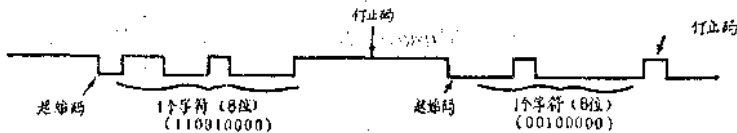


图1 非同步式数据

非同步式数据的传送速度被定义为“波特率”。所谓波特率，即传送最短讯号码所需时间的倒数。一般情况下，这表示 1 个数据位间隔。在使用一个停止位时，波特率等于位速度，在使用一个以上的停止位时，波特率不等于位速度。

在非同步式传送时，用稍高的波特率（10K 波特或 10K 以上，因导线长度、激励器种类而异）把数据发送到简单的双绞线上。在利用电话网传送时，通常被限制为约 2K 波特。另外，为了要使数据脉冲变换为能通过那种电话网的音调，就必须有调制解调器。

关于讯号的规定

在 *EIA* 的 *RS-232-C* 电气系统规格中，定义了与设备间传送的数据和控制讯息有关的电平以及通信上的逻辑规定。在数据交换时，认为在接口部位测得的电平低于 $-3V$ （相对于信号的地电平）的信号为“标记状态”，高于 $+3V$ 的信号为“空状态”。“标记状态”用逻辑 1 表示；“空状态”用逻辑 0 表示。

另一方面，在定时和控制交换电路中，认为交换电路的电平高于 $+3V$ （相对于信号的

地电平)时,它的功能处于“接通”状态;而在低于-3V时,处于“断开”状态。“接通”状态用逻辑0表示,“断开”状态用逻辑1表示。概括上述内容,有下表:

项 目	传 送 电 平	
	负	正
二进制表示	1	0
信号的状态	标 记	空
功 能	接 通	断 开

表1 接通/断开状态

扦针和信号的对应关系

在RS-232-C的机械系统规格中,必须要有一种25个扦针的连接器(称为DB-25)。表2说明了Tandy Radio Shack的RS-232-C接口的扦针和信号的对应关系。

扦 针 号	缩 写 形 式	信 号
1	PGND	保护用的接地
2	TD	发送数据
3	RD	接收数据
4	RTS	发送要求
5	CTS	发送用的清除
6	DSR	数据建立就绪
7	SGND	信号用的接地
8	CD	载波检测
20	DTR	终端设备就绪
22	RI	循环指示器

表2 扦针和信号的对应

保护用的接地:它和机架或设备的框架相连,也可和“信号用的接地”连接。

发送数据:给数据通信设备的指示。本线路上的信号由把数据传送给远程设备的数据终端设备发送。在字符间的空白部分或不传送数据时,该信号必须保持为“标记状态”。

接收数据:来自数据通信设备的指示。本线路上的信号是由传送数据给终端设备的远程设备所发送。在字符间的空白部分或没有发送来数据时,该信号必须保持为“标记状态”。

发送要求：给数据通信设备的指示。该信号由终端设备要求，用于控制数据通信设备的数据传送方向。单向或双向通道处于“接通状态”时，数据通信设备保持传送模式。在“断开”状态时，数据通信设备则为非传送模式。

半双通道为“接通”状态时，数据通信设备保持传送模式，但接收模式被禁止。“断开”状态时，数据通信设备转为接收模式。

发送用的清除：来自数据通信设备的指示。该信号由数据通信设备发送，表示“数据建立”（调制解调器）是否处于传送的就绪状态。该线路的“接通”状态指示数据终端设备：“数据建立”的传送数据线路处于可接收数据的状态，而“断开”状态则表示数据终端设备不得向“数据建立”传送数据。

数据建立就绪：来自数据通信设备的指示。该信号给数据终端设备指示“局部数据建立”的状态。该线路的“接通”状态表示：数据通信设备的当前模式不是测试、会活、拨号模式，就是完全结束了由呼号设定的必要的定时功能（回答声等）。除此以外则都成为“断开”状态，它表示数据终端设备接收的只是循环指示器信号，其他信号（交换线路上的所有其他信号）完全不予考虑。

终端设备就绪：给数据通信设备的指示。该信号用于控制数据通信设备和通信通道间的连接的切换。“接通”状态对应数据通信设备连向通信通道，处于“接通”状态期间，表示继续保持这种连接。一转为“断开”状态，在传送中的数据结束传送后，数据通信设备和通信通道切断。

循环指示器：来自通信设备的指示。该线路的“接通”状态表示循环讯号在通信通道上。这一般意味着要送出“数据建立”，其他设备要进行数据通信。在循环周期的（实际的循环表示时的）断开段期间以及不接收循环讯号时，通常保持“断开”状态。

载波检测（接收线路信号检测）：来自数据通信设备的指示。该信号为“接通”时，表示“数据建立”通过通信通道，从远程“数据建立”接收载波。另外，在不接收载波，或信号的特性不适于数据的调制时，变为“断开”状态

RS-232-C 的安装方法

在实际安装以前，请注意下列事项：

TRS-80扩展接口设备中，有时没有配置42线连接器（安装RS-232-C时所必需的），因此，万一所有的扩展接口中都没有安装这种连接器时，由于不能和RS-232-C恰当地连接，请退回本公司。此外，若自己随意安装的话，则敬请谅解今后不能兑现所作的保证。

扩展接口中配有必要的42线连接器时，请按下列步骤安装RS-232-C：

1. 如图2所示，放置好TRS-80扩展接口，卸下扩展用的盖板。
2. 从42线连接器上卸下两个螺钉和垫片。
3. 如图所示准备好RS-232-C接口使42线连接器上的螺孔和RS-232-C接口的印制板上的螺孔对准，然后，用两个螺钉和垫片将其紧固。
4. 按下图（正面图）所示，放置好RS-232-C，卸下扩展接口中央的盖板。然后，把RS-232-C所带的电缆的40线连接器牢牢在露出的直杆连接器上。扁形电缆往下方伸展，电缆的另一端接在要用的调制解调器、行式打印机等设备上。

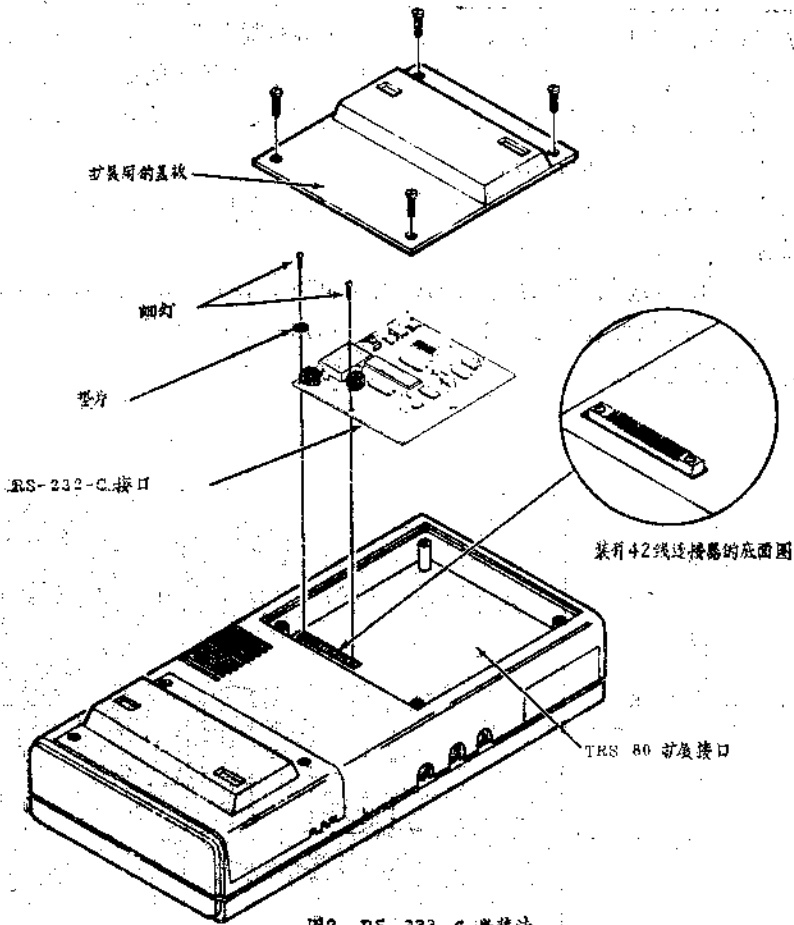
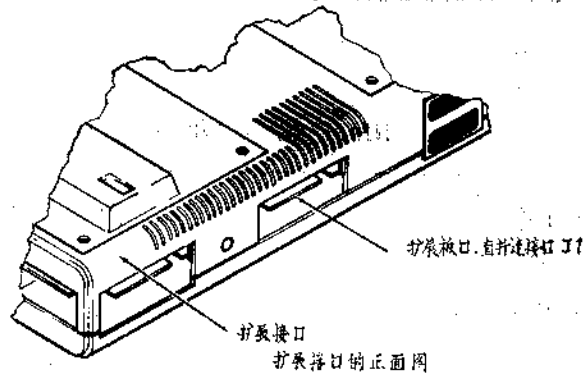


图2 RS-232-C 安装法

檢 驗：

1. 如下例所示，为了设定要用的波德率，奇偶校验等状态，按下表置放RS-232-C的转换开关（变换操作列表于表3）：

开	关	S ₈	S ₇	S ₆	S ₅	S ₄	S ₃	S ₂	S ₁
波德率=300		闭	闭	开					
停止位和奇偶各1个					闭	闭			
7位字长的数据和1个奇偶位							闭	开	
偶 校 验									开

然后，把TERM/COMM开关置于TERM（和调制解调器相连的情况）。

2. 使用LEVEL I BASIC "SYSTEM" 命令，从所带的磁带中取出"TERM"（TERM是文件名）。然后，打印出斜线（/）记号，按下"ENTER"键，启动程序[规定存貯单元地址为5000（16进制）]。

3. 根据图象显示屏幕是否清除来判断TERM带是否正常启动。在图象显示屏幕的左上方应该出现光标。

注：若将RS-232-C电缆的25线连接器的扦针2和3连接，则键盘的输出就返回呈现在图象显示屏幕上，由此可判断接口和TERM程序的正常工作。

4. 安装好扩展用的盖板。

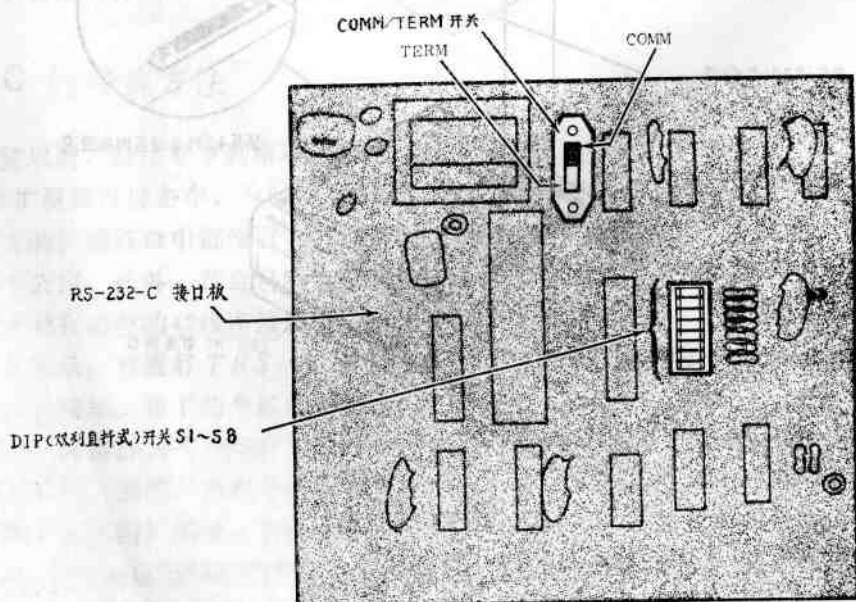


图3 RS-232-C 的各种开关

工作原理

由于 *Radio Shack* 的 RS-232-C 接口是一种灵活的、可编程序的接口，所以只要和 RS-232-C 有互换性，任何设备都能和其连接。本来，在 RS-232-C 设备中，有所谓数据终端设备和数据计算机（数据通信）设备两种不同类型，但是，*Radio Shack* 的 RS-232-C 接口的情况，则是作为和数据计算机相连的设备，构成了以数据终端设备为主体的机器。

接口的两种机器的切换控制既可由转换开关指定，也可用软件方法规定。

通用异步接收/发送器

图 4 是 RS-232-C 接口的框图。接口的中心部分是 UART（通用异步接收/发送器）串行数据的接收/发送所必要的硬件几乎都组装在这种 MOS 大规模集成电路 (TR1602A) 中。UART 是通用的，且符合工业规格。它是可编程序的设备。在把非同步式串行数据通道和具有并行数据结构的微处理器连接时使用这种 UART。UART 的发送部分把微处理器送出的并行的数据变换成由起始、奇偶、停止和数据各位组成的串行字。而在接收部分中，

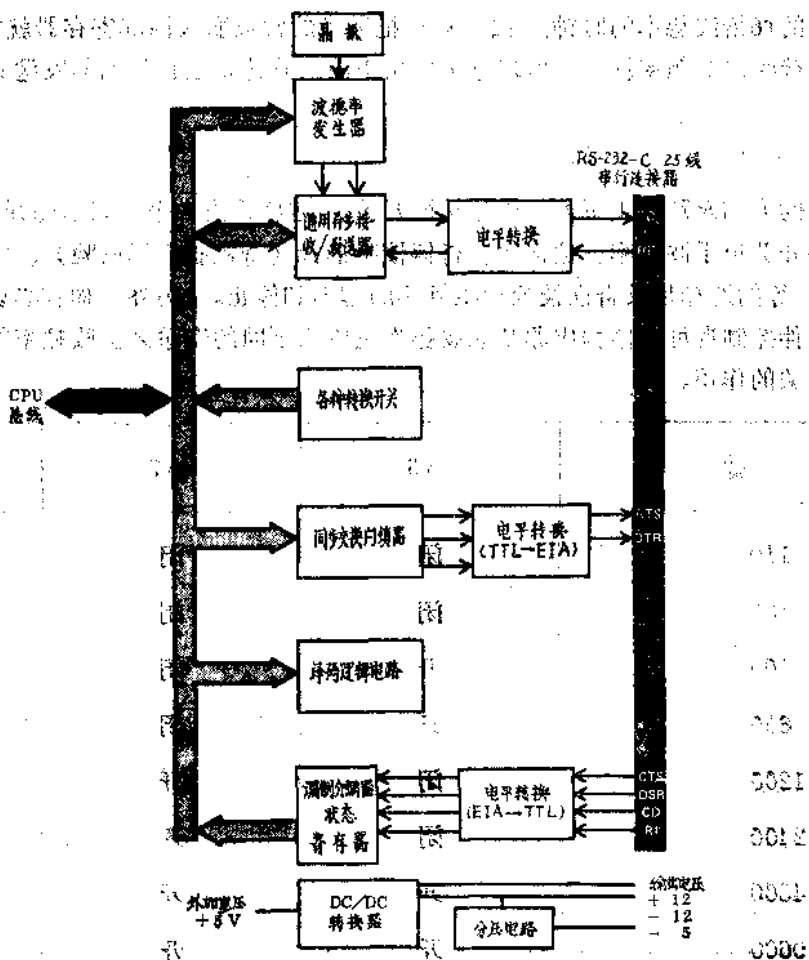


图4 RS-232-C 框图

与此相反，把串行字（由起始、数据、奇偶、停止各位组成）变换为并行数据，而且验证数据传送是否正确地进行（对接收到的奇偶和停止位不进行检验）。

UART 能根据用下述方法编制的程序来操作：字长可在 5, 6, 7, 8 位中选择使用。奇偶的形成和校验能禁止。可以规定奇偶校验为奇校验还是偶校验。停止位的个数可以使用 1 个或 2 个，但是，在传送 5 位码时，停止位则为 1.5 位。

由于本 UART 中备有若干个内部寄存器，所以可以进行串行数据通道的构成，UART 的状态的监视，传送用的数据的取入，串行数据通道上的数据的读入等。关于这些寄存器的详情，将在后面叙述。

波德率发生器

图 4 中，UART 的正上方框内是 BRG（波德率发生器）。这部分输出两个时钟信号，这些信号是 UART 正确地工作所不可缺少的条件，即由这两个信号 [TRANSMIT FREQ（发送频率）和 RECEIV FREQ（接收频率）] 来决定串行通道的发送/接收波德率。作为 BRG 的定时基准器，采用 5.0688 MC 的晶体振荡器。在这个 BRG 中，用程序可以把定时基准器的频率分频成 UART 用的各种频率。UART 在发送/接收时，必须要一个既定的 16 倍波德率的时钟。由于只要把适当的常数取入内部寄存器就能程序指定 BRG 的发送/接收用的频率输出，所以也可能用不同的波德率进行数据的发送和接收。

转换开关

图 4 的 UART 的正下方框内是转换开关。这部分由 8 个单刀单掷开关 (S1~S8) 组成。这些开关用于波德率、位/字、奇偶校验形式（奇校验或偶校验）、停止位个数（1, 1.5, 2）等的选择以及奇偶校验产生机构的起停。另外，如有必要，可不用这些开关，用软件控制也可直接构成非基准波德率或使用不同的发送/接收频率的接口。表 3 汇总了转换开关的作用。

波 德 率	S6	S7	S8
110	闭	闭	闭
150	闭	闭	开
300	开	闭	闭
600	开	闭	开
1200	闭	开	闭
2400	闭	开	开
4800	开	开	闭
9600	开	开	开

奇偶校验功能	S4		停止位个数	S5
起 动	闭		1	闭
停 止	开		2	开

字长(除奇偶位)	S2	S3	奇偶校验形式	S1
5 位 长	闭	闭	奇 校 验	闭
6 位 长	闭	开	偶 校 验	开
7 位 长	开	闭		
8 位 长	开	开		

表 3 转换开关的作用

同步交换门锁器

Tandy Radio Shack 的 RS-232-C 接口能够 (使用适当的软件) 控制内部的两类控制信号 (发送要求和终端设备就绪) 的逻辑状态。为了实行这种功能, 必须把适当的位模式取入同步交换门锁器中。如图 4 所示, CPU 读出调制解调器状态寄存器的内容, 最多能检测到四个接口信号 (发送用的清除, 数据建立就绪, 载波检测, 循环指示器)。如使用适当的软件, 同步交换门锁器和调制解调器状态寄存器能在 CPU 和 RS-232-C 与相连的设备间进行同步交换式对话。

逻辑规定

RS-232-C 的内部逻辑根据 TTL 电平工作 (3.5V 以上=逻辑值 1, 0.8V 以下=逻辑值 0)。另一方面, 在两种 RS-232-C 设备相互连接时的逻辑规定中, 使用 EIA 电平 (-3V 以下=逻辑值 1, +3V 以上=逻辑值 0)。因此, 为了连接不同的 RS-232-C 设备, 逻辑电平必须向某个方面统一。这种转换工作由图 4 中写有 EIA→TTL, TTL→EIA 的框来进行。

口地址

RS-232-C 接口和 CPU 通信时, 在 I/O 变换装置上使用四个口地址 (E8H, E9H, EAH, EBH)。在 Z80 系统的 I/O 变换装置的情况下, 则和 1N*, QUT* 信号一起, 还使用低位地址位 (A₀-A₇) 来寻址既定的口。图 5 是和这种地址指定机构有关的时序图。

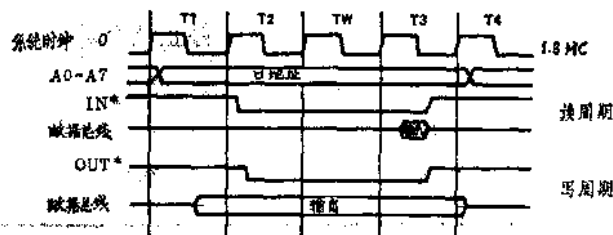


图5 I/O 时序图

译码逻辑电路

图4中写有“译码逻辑电路”的框产生8个负选通，由RS-232-C使用。根据这些选通，在I/O时，CPU可选定RS-232-C接口的任意寄存器和门锁器。表4汇总了用RS-232-C接进行的I/O口地址分配和它的功能：

地 址	输 出 (I/O 口 写)	输 入 (I/O 口 读)
E8H	主复位 (全数据)	调制解调器状态寄存器
E9H	波德率选择	结构转换开关
EAH	UART控制寄存器和同步交换门锁器	UART状态寄存器
EBH	发送数据寄存器	接收数据寄存器

表4 I/O 变换的存储器分配

上表中所列的寄存器各自执行RS-232-C接口的特定功能。当向主复位口(E8H)输出时，就把UART复位到一定的状态。这种操作在不读取UART寄存器的任何内容时，必须进行一次，这可以由下列Z-80指令的执行来实现：

OUT (0E8H), A; 输出到主复位单元

此时，A的内容并不重要，对操作结果毫无关系。当从这个口地址(E8H)进行输入时，调制解调器状态寄存器的内容就被取入CPU的寄存器。该操作可以通过执行下列Z80的指令来实现：

IN A, (0E8H); 读调制解调器状态寄存器

当执行该指令时，调制解调器状态寄存器的内容取入A中。取入A中的讯息是和接口相连的外部设备的状态有关的讯息。

当在口E9H输出时，某常数就被取入波德率发生器中。由该常数确定串行通道的接收/发送波德率。此操作可通过执行下列Z80的指令来实现：

LDA, 22H; 把常数取入A中

OUT (0E9H), H; 把常数取入BRG中

一执行上述程序，就使得串行通道以110波德的速率进行数据的发送/接收。在表5中概述了取入BRG中的常数和由此选定的波德率之间的关系。

由取入 BRG 中的常数的高位部分 ($D_7 \sim D_4$) 来决定串行通道的发送波德率, 低位部分 ($D_3 \sim D_0$) 决定接收波德率。

取入的 半字节	发送/接收 波德率	10×时钟 频率	误码率 (%)	开关的选定
0H	50	0.8KC	0	NO
1H	75	1.2KC	0	NO
2H	110	1.76KC	0	Yes
3H	134.5	2.1523KC	0.016	NO
4H	150	2.4KC	0	Yes
5H	300*	4.8KC	0	Yes
6H	600	9.6KC	0	Yes
7H	1200	19.2KC	0	Yes
8H	1800	28.8KC	0	NO
9H	2000	32.081KC	0.253	NO
AH	2400	38.4KC	0	Yes
BH	3600	57.6KC	0	NO
CH	4800	76.8KC	0	Yes
DH	7200	115.2KC	0	NO
EH	9600	113.6KC	0	Yes
FH	19,000	316.8KC	3.125	NO

表 5 BRG 用的程序编制概要

* 原文误为 130——译者注

BRG 的程序编制

表 5 开关选定栏中印有“*Yes*”的波德率由 *Radio Shack* 的软件支持。非标准波德率的程序根据取入 BRG 的适当的常数来编制。从口 E9H 的输入操作把由结构转换开关程序化了的位模式取入 CPU 的寄存器中。如前所述, 图 4 的转换开关框由 8 个单刀单掷开关组成。因此, 开关为“开”状态时, 结构为 8 进制表示的三态缓冲器的输入被提高(达 +5V)。相反地, 在开关为“闭”状态时, 输入被降低。

译码逻辑电路响应从口 E9H 的输入操作, 产生负选通。该选通和三态缓冲器的“允许”输入端相连。为向 CPU 存数, 缓冲器利用这个负选通, 把其输入逻辑状态给出在数据

总线上。此操作根据下列指令的执行来实现：

$INA, (0E9H)$ ；取“开关选定”

执行该指令后，和设定的开关相对应的位模式就被存入CPU的A寄存器中。被存入的讯息立刻由软件和CPU发出的一系列命令加以翻译，在这个基础上才能构成接口。

当在口地址EAH输出时，UART控制寄存器和同步交换门锁器就被取入。这两个寄存器的功能彼此不同，控制时使用的寄存器位也不同。该口地址的高5位($D_7 \sim D_3$)表示串行通道的字长，停止位和奇偶规则等，而低3位($D_2 \sim D_0$)则表示被门锁的控制输出。在这个输出中，一位在低电平时为处于数据不可传送的状态，另两位是控制“终端设备就绪”和“发送要求”的信号的。在这个口地址上输出时，欲不产生干扰，必须给于一定的注意。表6是RS-232-C的各寄存器和位的分配的概要。

数据位	调制解调器的状态寄存器	结构转换开关	UART控制寄存器和同步交换门锁器	UART的状态寄存器
D_7	发送用的清除 扦针5, DB-25	偶校验 1=偶数, 0=奇数	偶校验 1=偶数, 0=奇数	接收数据的状态 1=正常
D_6	数据建立就绪 扦针6, DB-25	字长选定1	字长选定1	发送保持寄存器为空 1=正常
D_5	载波检测 扦针8, DB-25	字长选定2	字长选定2	溢出故障的状态 1=正常
D_4	循环指示器 扦针22, DB-25	停止位个数选定 1=2位, 0=1位	停止位个数选定 1=2位, 0=1位	成帧误差的状态 1=正常
D_3	不用	禁止奇偶校验 1=不进行奇偶校验	禁止奇偶校验 1=不进行奇偶校验	奇偶校验故障的状态 1=正常
D_2	不用	波德率3	发送中断 0=数据不能传送	不用
D_1	接收设备输入 UART P20	波德率1	发送要求 扦针4, BD-25	不用
D_0	不用	波德率2	终端设备就绪 扦针20, DB-25	不用
	输入0E8H	输入0E9H	输出0EAH	输入0EAH

表6 各寄存器和门锁器的位分配

用 Z80 汇编语言书写的下述程序示例的内容是：读取转换开关并和既定的门锁位组合在一起，取入 *UART* 控制寄存器和同步交换门锁器，为了以后在必要时进行修改，把这些控制位的现状保存起来。

```
IN A, (0E9H) ; 读转换开关
AND 0F8H ; 使最低三位为 0
OR 05H ; 置位“发送要求”，复位“终端设备就绪”并建立断点
OUT (0EAH) ; 取入 UART 控制寄存器和同步交换门锁器
LD (IMAGE), A ; 保存位模式以备修改
```

上例最后一行中的 *IMAGE* 表示存贮单元。该存贮单元用于保存控制寄存器和同步交换门锁器的现状。下面是欲改变发送要求的逻辑状态的程序。若使用这种方法，在不改变 *UART* 控制寄存器和其他同步交换位的情况下也能改变逻辑状态。

```
LD A, (IMAGE) ; 取 UART 控制寄存器和同步交换门锁器的现行状态
RES 0, A ; 复位 A 中的位 0 (发送要求)
OUT (0EAH), A ; 存入新的位模式
LD (IMAGE), A ; 保存新的位模式以备修改
```

在口地址 *EAH* 输入时，*UART* 状态寄存器的内容就被读取。位 $D_7 \sim D_0$ 传递了有关 *UART* 现状的讯息（输入时的接收数据的准备情况，发送保持寄存器是否为空的情况，有无溢出故障，有无成帧差错，有无奇偶错）。其他的 $D_7 \sim D_0$ 位不用。这种寄存器的读取可通过执行下述汇编指令来实现：

```
IN A, (0EAH) ; 读 UART 状态寄存器
```

当执行此指令时，*UART* 状态寄存器的内容就被取入 *A* 寄存器。若使用附加指令，就能译出该状态讯息，并能验证新字符能否传送，是否接收了新字符，接收了的新字符中是否有错误。

当向口地址 *EBH* 输出时，从 *UART* 传送来的新字符被取入 *UART* 发送保持寄存器中。但是，在没有判明保持寄存器为空状态（即前面的字节已传送到准备传送的发送设备的寄存器中）以前，不能进行此操作。在实际进行操作时，要读出 *UART* 状态寄存器，验证对应位为逻辑 1。下述汇编指令组就是这种方法：

```
OUT (0EBH), A ; 把字符输出到 UART 以便发送
IN A, (0EAH) ; A 中应含有要被发送的字符
PUSH A ; 保存要被发送的字符
STATIN IN A, (0EAH) ; 取 UART 状态寄存器
BIT 6, A ; 测试发送保持寄存器是否为高电平
JR NZ, STATIN ; 如果不是，重新测试
```



```

POP AF ; 恢复要被发送的字符
OUT (0EBH), A ; 把字符取入保持寄存器

```

当从口地址 *EBH* 输入时, *UART* 的接收保持寄存器的内容被读取, *UART* 的状态寄存器的接收数据位被置位。在 *UART* 状态寄存器内的接收数据位被置位 (即一字符被接收并被传送到接收保持寄存器) 以前, 此操作不能进行。下述汇编指令组就是这种方法:

```

; 从 UART 输入一字符
; 接收到的字符将在 A 寄存器中
INCHAR IN A, (0EAH) ; 取 UART 状态
; 测试接收到的数据是否为高电平
BIT 7, A ; 如果不是, 重新测试
JR Z, INCHAR ; 取接收到的字符
IN A, (0EBH) ;

```

执行上述程序后, 接收到的数据被取入寄存器 *A* 中, *UART* 控制寄存器的接收数据位被复位。

本手册的开始部分说过: 在 *RS-232-C* 设备中, 有数据终端设备和数据通讯设备两种不同类别。它们不同点在于: 在发送数据和接收数据的功能中, 利用 *DB-25* 连接器的不同扦针。数据终端设备在扦针 2 发送数据, 在扦针 3 接收数据。因此, 和数据终端设备相连的设备, 在扦针 2 接收数据, 在扦针 3 发送数据 (请参阅表 2)。换言之, 根据接口设备的特定的要求, 有时也有必要使这两种设备的信号操作反演。考虑到这种情况, 在 *Tandy Radio Shack* 的 *RS-232-C* 接口中安装一个双掷开关 (请参阅接口板的略图), 开关的一边写有 “*TERM*” (数据终端设备), 另一边写有 “*COMM*” (通信设备)。

RS-232-C 接口中, 必须供给四种电压 (+5V, -5V, +12V, -12V)。扩展接口供给 +5V 电压, 但是, 使用 *DC-DC* 转换器, 可从这个电压产生更高的电压。在 *Tandy Radio Shack* 接口 *RS-232-C* 中使用的 *DC-DC* 转换器, 只要输入 +5V, 就能输出 +12V 或 -12V。如对 -12V 使用简单的电阻分压电路, 还能供给 -5V 电压 (请参阅图 4 的略图)。由于 *DC-DC* 转换器本来是一种会发出令人讨厌的噪声的部件, 所以必须和输入源进行适当的隔离, 而在输出端要加滤波器。这种隔离和滤波功能由 L_1 , L_2 , C_1 , C_{10} , C_8 和 C_9 来实现。

终端设备的程序

下列设备是必要的:

- * 带 16K *RAM* 的 *TRS-80 LEVEL I*
- * 配备 *RS-232* 的扩展接口

TRS-80/RS-232-C 由于 “*TERM*” *inal* 程序而能获得作为全双工终端设备的功能。这种所谓全双工终端设备可以显示从 *RS-232-C* 的口输入的文本 (变换成 *ASCII*