

IBM PC/XT 对TP801-A的控制与管理

徐正良

摘要：本文介绍IBM PC/XT个人计算机和TP801-A单板机的一种联机方法及其控制与管理技巧。

一、前言

IBM PC/XT个人计算机和TP801-A单板机获得了广泛的应用。关于PC机和TP801-A的联机方法、控制与管理是很多用户关心的一个问题。我们在一项工程任务中使用IBM PC/XT做主控制机，TP801-A单板机做前置机，取得满意的结果。下面从使用的角度做一些介绍。

二、联机方法

1. 接口电路

为了在PC机与单板机之间进行双向信息传送，需要增加一些硬件支持。在工程控制

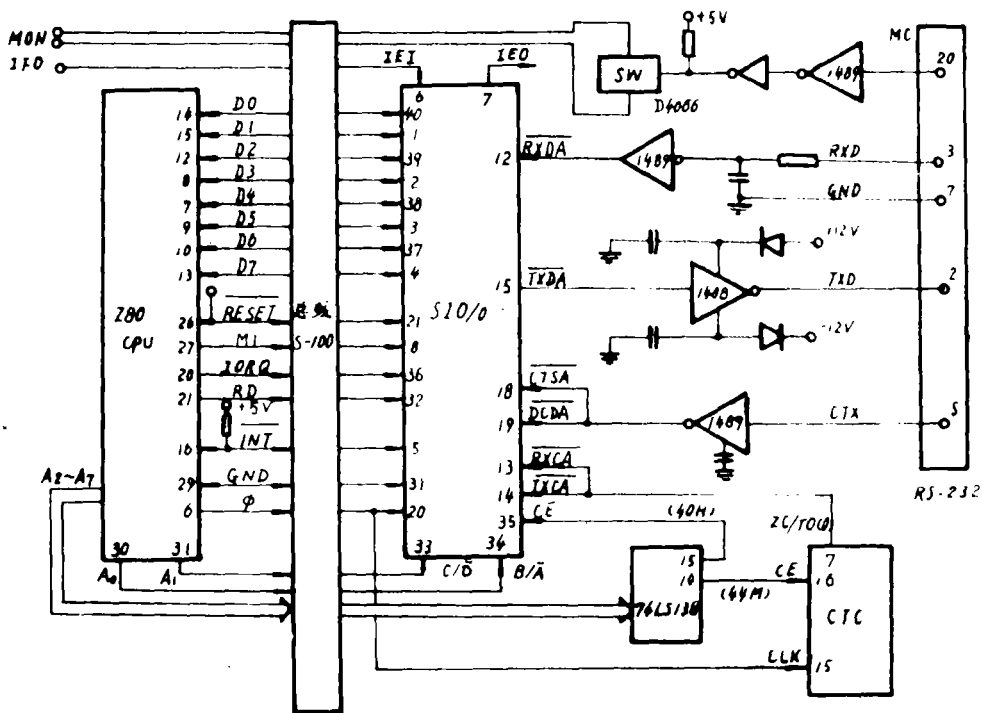


图1 单板机的串行接口电路

中，采用串行通讯较适宜。通常 PC 机，向外界提供一个标准的 RS-232C 接口。我们为 TP801-A 配接了一个串行口，实现了 PC 机和单板机的联接。接口电路见图 1。

Z80-SIO 是一个可编程的串行 I/O 接口芯片，为 TTL 电平，使用传输线驱动器 1488 (用于发送) 和 1489 (用于接收) 可转换为标准的 EIA 电平。采用异步电压工作方式，引出下列数据和控制信号，组成单板机的串行接口。

- 2 脚 发送数据 (TXD)
- 3 脚 接收数据 (RXD)
- 5 脚 清除发送 (CTS)
- 7 脚 信号地 (GND)
- 20脚 MON 键控制信号 (MC)

MC 是自定义的控制信号，控制 MON 键的闭合与释放。接口板通过 S-100 总线与 Z28-CPU 相联。

PC 机提供标准的 RS-232C 接口，其电压工作方式支持下列数据和控制信号。

- 2 脚 发送数据 (TXD)
- 3 脚 接收数据 (RXD)
- 4 脚 请求发送 (RTS)
- 5 脚 清除发送 (CTS)
- 6 脚 数据设备准备好 (DSR)
- 7 脚 信号地 (GND)
- 8 脚 接收线路信号检测 (DCD)
- 20脚 数据终端准备好 (DTR)
- 22脚 振铃指示 (RI)

2. PC 机和单板机的连接

PC 机和单板机的连接是两个终端之间的连接，称之为零调制解调器方式。将一方的输出连到另一方的输入，就能进行相互通讯。这里采用了图 2 所示的连接方法。RTS 与 CTS 相连，使 PC 机对单板机的发送起控制作用。DTR 接到自定义信号 MC，通过编程使 PC 机的 DTR 输出一个短暂的选通脉冲，作为模拟开关 SW 的输入控制信号，使单板机 MON 键闭合，用来中止现程序的执行，返回到监控程序 TPBUG。通过这两个控制信号以及相应的软件支持，就可以实现 PC 机对单板机的控制与管理。

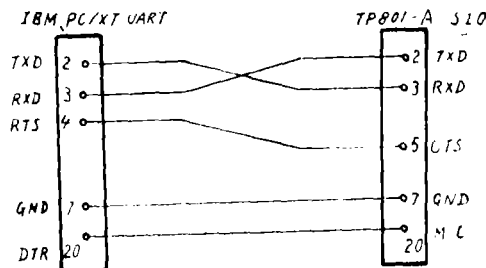


图 2 两个串行接口的连接

三、程序设计

1. 接口初始化

为了使用灵活, 不受系统提供软件支持程度的限制, *PC*机程序用8088汇编语言编写, 直接面向异步通讯适配器和其他I/O接口。异步通讯适配器的核心是INS8250芯片。它有十个可访问的寄存器, 端口地址为3F8H~3FEH, 如表1所示。我们可用输出指令向相应的寄存器写入控制字或数据, 设定波特率、奇偶校验, 停止位、数据字节位数、开关中断, 以及MODEM输出控制信号, 完成控制芯片初始化(详见实例)。

表1 INS8250寄存器的I/O编址

I/O编址	寄存器选择	DLAB状态
3F8	发送保持寄存器	0 (写)
3F8	数据接收缓冲器	0 (读)
3F8	除数锁存器低8位	1
3F9	除数锁存器高8位	1
3F9	中断允许寄存器	0
3FA	中断识别寄存器	
3FB	线路控制寄存器	
3FC	MODEM控制寄存器	
3FD	线路状态寄存器	
3FE	MODEM状态寄存器	

SIO的四个口地址设定为40^H、41^H、42^H、43^H。通道的控制部分有8个写寄存器和3个读寄存器, 它们的编号分别为WR₀~WR₇和RR₀~RR₂。主要功能见表2。对于异步工作方式, WR₆和WR₇不用。SIO初始化子程序如下:

```

SIOINI, LD HL, INITB, 置初始化表首址
        LD A, 17H, CTC初始化
        OUT (44H), A
    
```

表2 SIO各寄存器的主要功能

控制口地址		写寄存器	主要作用	读寄存器	主要作用
A通道	B通道				
42	43	WR ₀	设置命令和寄存器编号	RR ₀	反映工作状态标志
42	43	WR ₁	开放各种中断	RR ₁	反映错误标志
	43	WR ₂	中断矢量低8位	RR ₂	保存中断矢量
42	43	WR ₃	选择接收器工作状态		
42	43	WR ₄	选择工作方式, 设定通讯规程		
42	43	WR ₅	选择发送器工作状态		

* 开放“状态影响矢量”仅对B通道有效, A通道的该位不起作用。
 * WR₂和RR₂只有B通道有, 但A, B通道共用。

```

LD    A, 0DH, f=9600Hz
OUT  (44H), A
LD    BC, 0A42H,
OTLR                ; 重复输出传送
RET
    
```

INITB: 10 18 14 07 13 E1 15 EA 11 00

表中10, 14, 13, 15, 11分别指向寄存器 WR_0 , WR_4 , WR_3 , WR_5 和 WR_1 , 并且含有复位外部/状态中断命令。其它控制字的意义如下:

- 18H 通道复位。
- 07H $\times 1$ 时钟率, 停止位1, 偶校验。
- E1H 开放接收器, 开放外部控制, 数据位8。
- EAH 开放发送器, 数据位8。
- 00H 禁止中断。

2. 管理程序

为了便于PC机和单板机之间的通讯, 将监控程序 $TPBUG-A$ 稍加改动, 扩充串行通讯及其有关功能。图3为修改后的程序流程图。在显示程序 $DISUP$ 的出口处, 进入SIO接收

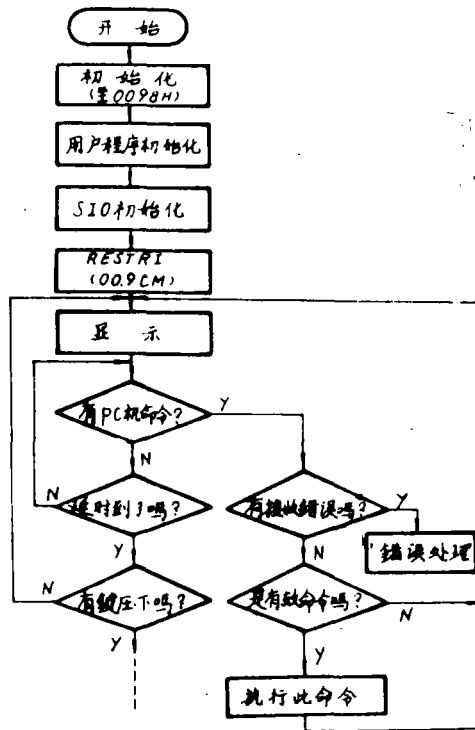


图3 修改后TPBUG-A开始部分流程图

查询状态。这时如果 PC 机发来命令, 单板机经检验是否有错及是否为有效命令后, 进行处理。否则约等待 $200\mu s$, 进入键盘分析程序。修改后监控程序原有的一切功能不变。接收查询子程序和错误处理程序如下:

```

POLL, IN  A,    (42H); 接收字符可用否?
      BIT  0,    A
      JR   Z,    POLL, 否, 再查询
      LD   A,    0IH;  可用, 读入RR1内容
      OUT (42H), A
      IN  A,    (42H)
      AND 70H; 取错误标志
      RET      , 带错误标志返回
ERR,  IN  A,    (40H); 清除数据
      LD   A,    30H;  错误复位
      OUT (42H), A
ER1,  IN  A,    (42H); 检测发送缓冲器空吗?
      BIT  2,    A
      JR   Z,    ER1
      LD   A,    07H; 空, 向PC机发送错误标志
      OUT (40H)  A
ER2,  LD   A,    06H; 单板机显示错误标志“E”
      OUT (88H), A
      LD   A,    20H
      OUT (8CH), A
      HALT
    
```

3. 联络与控制

RS-232接口没有严格的询答联络功能, 用硬件是不能确认每个字节的发送和接收的。各个终端一旦处于工作状态, 就不管对方如何即进行数据的发送或接收。解决这个问题的方法有两个。一个是正确使用RTS控制信号。PC机和单板机给电后, 主机的RTS信号被复位, 它封锁SIO的输入端 $\overline{CTS_A}$ 和 \overline{DCDA} , 使SIO既不能接收又不能发送。这时根据需要在主机程序中置位RTS, 就可开放SIO的接收和发送, 进入接收查询状态, 等待主机发送命令或数据。当主机程序执行终止时, 使RTS复位, 禁止对方发送或接收。由于PC机处于控制地位, PC机向单板机发送信息不受后者控制。因此总是先启动单板机程序, 后启动PC机程序。另一个方法是在发送方的程序中, 每发送一个字符之后安排合适的延时。最短的延时受到所使用的波特率的限制。否则就会发生数据重迭或溢出错误。

四、程序举例

PC机执行的程序可当作DOS的一个外部命令来使用。通过主机键盘可对单板机进行控制。这里仅举两个例子。

1. EXECUTE address

address是命令单板机执行程序的起始地址, 作为命令行的一个参数, 进入程序段前缀PSP的数据区。在程序的标准序之后, 用字符串命令把参数address由DOS的数据区传送到用户程序的数据区, 然后发送给单板机。单板机则把接收到的ASC II字符, 转换成16进制数字, 组成实际的启动地址, 运行此程序。

为了能够在单板机没有按EXEC键的情况下, 可以用MON命令来终止程序, 在Z80的

8088 程序

```

START  PROC  FAR
        ASSUME CS:  CODE
        PUSH  DS
        MOV   AX,  0
        PUSH  AX
        MOV   AX,  DATA
        MOV   ES,  AX
        ASSUME ES:  DATA
        MOV   SI,  5CH
        MOV   DI,  OFFSET FCB
        MOV   CX,  12
        CLD
        REP   MOVSB
        MOV   SI,  80H
        MOV   DI,  OFFSET PARM
        MOV   CX,  128
        REP   MOVSB
        MOV   DS,  AX
        ASSUME DS:  DATA
        MOV   DX,  3FBH
        MOV   AL,  80H
        OUT   DX,  AL
        MOV   DX,  3F8H
        MOV   AL,  0CH
        OUT   DX,  AL
        MOV   DX,  3F9H
        MOV   AL,  0
        OUT   DX,  AL
        MOV   DX,  3FBH
        MOV   AL,  1BH
        OUT   DX,  AL
        MOV   DX,  3FCH
        MOV   AL,  02H
        OUT   DX,  AL
        MOV   DX,  3F9H
        MOV   AL,  0
        OUT   DX,  AL
        MOV   DL,  PARM
        XOR   DH,  DH
        MOV   CX,  DX
        MOV   DI,  OFFSET PARM+2
        MOV   AL,  ","
        REPNE SCASB
        JNE  PARMIS
        MOV  BX,  DI
        CALL GETADDR
        MOV  ADDRESS, AX
        CALL GETADDR
        SUB  AX,  ADDRESS
        INC  AX
        MOV  COUNT1, AX
        MOV  DI,  OFFSET ADDRESS
AFOREV: MOV  DX,  3FDH
        IN  AL,  DX
        TEST AL,  01H
    
```

; PSP首址入栈,供返回DOS用
 ; 建立附加段可寻址性
 ; 由PSP取文件名
 ; 由PSP取输入参数
 ; 建立数据段可寻址性
 ; 8250 UART初始化
 ; 9600波特
 ; 8位数据,偶校验,1停止位
 ; 置DTR=0,RTS=1,开放SIO
 ; 禁止中断
 ; 确定输入参数偏移
 ; 检测 ","
 ; 无 "," 转示输入错误
 ; 取输入参数偏移
 ; 获得addr1,在AX中
 ; 存addr1
 ; 获得addr2,在AX中
 ; 计算转储字节数
 ; 字节数送存
 ; 取addr1偏移
 ; 检测UART状态
 ; 有对方返回信息吗?

```

JNZ     JPRECER           ; 有,转显示传送错误
TEST    AL, 20H           ; 发送缓冲器空吗?
JZ      AFOREV           ; 否,再查询
MOV     DX, 3F8H         ; 发送转储首址和字节数
MOV     AL, [DI]
OUT     DX, AL
INC     DI
CALL    DELAY            ; 发送延时
DEC     COUNT0           ; 发送次数减1(初值为4)
JNZ     AFOREV           ; 未完继续
MOV     BX, OFFSET DBUF1 ; 取数据缓冲区偏移
DFOREV: MOV    DX, 3FDH   ; UART状态查询
        IN     AL, DX
        TEST   AL, 01H   ; 接收字符可用否?
        JZ     DFOREV    ; 否,再查询
        MOV    DX, 3F8H  ; 接收转储数据
        IN     AL, DX
        MOV    [BX], AL  ; 存入数据区
        INC    RX        ; 修改数据区指针
        DEC    COUNT1    ; 转储字节数减1
        JNZ    DFOREV    ; 未完继续

```

往下缓冲区数据存入顺序文件

```

        ⋮
RE_RTS: MOV    DX, 3FCH
        MOV    AL, 0      ; 复位RTS,封锁SIO
        OUT   DX, AL
        RET
START  ENDP

```

Z80 程序

```

        CALL   SIOINI    ; SIO初始化
        LD     HL, ADDPTR ; 置指针
        LD     B, 04H    ; 置接收字节数
D1:     CALL   POLL      ; SIO接收查询
        JP     NZ, ERR   ; 有错则转
        IN    A, (40H)   ; 读入字符
        CP    07H       ; 是错误标识符吗?
        JP     Z, ER2    ; 是,转显示错误标志"E"
        LD    (HL), A    ; 否,存输入数据
        INC   HL         ; 修改指针
        DJNZ  D1         ; 未完继续
        LD   DE, (BYTPTR); 取转储字节数
        LD   HL, (ADDPTR); 取转储首址
D2:     IN    A, (42H)    ; 读SIO状态
        BIT   2, A       ; 发送缓冲器空吗?
        JR   Z, D2       ; 否,再查询
        LD   A, (HL)     ; 取转储数据
        OUT  (40H), A    ; 发送数据
        INC  HL          ; 修改指针
        CALL DLY         ; 延时等待
        DEC  DE          ; 转储字节数减1
        LD   A, D
        OR   E
        JR   NZ, D2     ; 未完继续
        HALT            ; 结束

```

程序中应写入如下程序段:

```
LD  A, 04H
OUT (8CH), A ; 允许NMI中断
LD  A, 45H
OUT (86H), A ; 置CTC2为计数方式
LD  A, 01H
OUT (86H), A ; 时间常数为1
```

这样, 当MON键闭合时, 可产生非屏蔽中断, CPU执行中断服务程序然后返回扫描键盘、显示和SIO接收查询的循环程序。

2. DUMP filename.ext, addr1, addr2

该命令是把单板机存贮器的内容送入PC机存盘, 存入顺序文件filename.ext中。addr1和addr2分别为转储的起始和终止地址。以上输入参数从PSP传送到程序的变量区。8088程序和Z80程序的主要部分见前程序单。

该程序的有关变量定义、提示信息、若干子程序和分支处理程序略。字变量COUNT1和BYTPTR分别定义在ADDRESS和ADDPTR之后, 这有利于简化程序设计。传送速率为9600波特。在Z80程序中设有错误状态检测, 以保证传送正确无误。当波特率较低时, 可以不进行此检测。在传送数据时, 不能用任何控制字符作为文件结束或传送数据结束的标志, 故以字节数作为结束的控制条件。该程序具体说明了PC机和单板机之间双向数据通讯、控制方法和技巧。

五、结 束 语

以上程序在IBM PC/XT和TP801-A联机情况下运行通过。单板机的许多功能, 现在都可以在IBM PC/XT键盘上输入命令或数据来实现。这样, 可以充分发挥单板机价格低廉、适于做实时控制和系统机数据存贮及处理功能强的优点, 极大地增强了单板机的显示功能和存贮容量, 提高了使用性能。这对于单板机的开发和使用是很有意义的。

参 考 文 献

- [1] IBM PC 汇编语言程序设计, 辽宁省电子计算机学会
- [2] IBM PC/XT 硬件技术手册, 辽宁省电子计算机学会
- [3] 微型计算机—Z80, 科学技术文献出版社重庆分社, 1979, 12
- [4] TP801 Z80 单板计算机使用手册, 北京工业大学电子厂
- [5] 何心虔; 微型计算机, 1984, 6, 47

Control and Managemet of Tp801—A by IBM PC/XT

Xu Zhengliang

Abstract

This paper introduces an on-line method of connecting IBM PC/XT with single board computer TP801-A and its control and management art.