实验五　串行通信

实验报告

姓名：赵文亮

学号：2016011452

班级：自 64

日期：2018 年 6 月 4 日

目录

1 实验目的 **1**

2 实验准备 **1**

3 实验内容 **1**

3.1 必做任务 1

3.1.1 自检模式收发字符 1

3.1.2 两台计算机发送并接收字符 1

3.2 选做任务 1

3.2.1 两台计算机的字符串传送 1

4 实验程序及说明 **1**

4.1 自检模式收发字符 1

4.2 两台计算机发送并接收字符 4

4.3 两台计算机的字符串传送 7

5 完成情况和心得体会 **11**

1 实验目的

1. 复习 8250 工作原理及其在串行通信中的应用。

2. 练习使用 DOS 及 BIOS 功能调用来编写 I/O 程序。

2 实验准备

1. 复习 8250 及硬中断的有关知识，复习 DOS 及 BIOS 调用有关知识。

2. 按实验要求编写程序流程图及汇编语言源程序。

要求整个程序分成几个功能块，例如主功能块、串行接收功能块、键盘管理功能块及串行发送功能块。主功能块首先检查串口，发现有接收字符则转接收，否则检查键盘，发现有键入字符则调键盘管理，键入的有效字符通过发送功能块送出。每个功能块中要对字符有相应的判断处理 (例如出错信息、控制字符等)。原始的源程序清单上要有足够的注释，以备实验中调整修改。

3 实验内容

3.1 必做任务

3.1.1 自检模式收发字符

直接对计算机自带的 8250 芯片编程，将其设置成 1200 波特率、8 个数据位、1 个停止位、奇校验规则、自检方式。用 DOS 功能调用 (INT21H) 接收键入字符（需要回显），通过 8250 以查询方式发送又自己接收并在 CRT 上显示。直至键入空格 (ASC 码为 20H) 时退回 DOS。

3.1.2 两台计算机发送并接收字符

将上程序修改成两台计算机之间以查询方式通信，即一方键入的字符在另一个 CRT 上显示，反之亦然，任何一方键入空格，双方都退出。

3.2 选做任务

3.2.1 两台计算机的字符串传送

监视键盘，若键入字母键“S”则将事先存在数据区中的一个字符串串行传送给对方显示；若键入字母“R”则将对方机器数据区中的一个字符串传送过来在 CRT 上显示。两个字符串都以 '$' 为结束符。

4 实验程序及说明

4.1 自检模式收发字符

实验程序如下:

```
1  data    segment
2  divid   dw  60h
3  data    ends
4  stack   segment para stack
5          db 100 dup(?)
6  stack   ends
7  code    segment
8          assume  cs: code, ds: data, es:data, ss: stack
9  start:  mov ax, data
10         mov ds, ax
11
12 ;8250 initialize
13 ;baud rate=1200 data digits=8 stop digit=1 odd check self-check
14 ;com 1
15 ;line control
16         mov al, 80h     ;dlab=1
17         mov dx, 3fbh
18         out dx, al
19         mov ax, divid   ;get divide number
20         mov dx, 3f8h
21         out dx, al      ;lower byte
22         mov al, ah
23         mov dx, 3f9h
24         out dx, al      ;higher byte
25         mov al, 0bh
26         mov dx, 3fbh    ;write line control register
27         out dx, al
28         mov al, 13h
29         mov dx, 3fch    ;MODEM control register
30         out dx, al
31         mov al, 0
32         mov dx, 3f9h    ;interupt register
33         out dx, al
34 ;main loop
35 mloop:  mov dx, 3fdh    ;read line status register
36         in al, dx
37         test al, 1EH    ;test error
38         jnz error
39         test al, 1h     ;receive ready?
40         jnz recei
41         test al, 20h    ;sender empty?
42         jz mloop
43         mov ah, 1
44         int 16h         ;read keyboard
45         jz mloop
46         mov ah, 0
47         int 16h
```

```
48          mov ah, 0eh      ;show original char
49          int 10h
50          mov dx, 3f8h     ;send
51          out dx, al
52          cmp al, 20h      ;space exit
53          jz exit
54          jmp mloop
55
56  recei:  mov dx, 3f8h     ;read data
57          in al, dx
58          cmp al, 20h
59          jnz char
60          jmp exit
61
62  char:   push ax
63          mov ah, 0eh      ;show the char
64          int 10h
65          pop ax
66          cmp al, 0dh      ;check the cr
67          jnz mloop
68          mov al, 0ah      ;new line
69          mov ah, 0eh
70          int 10h
71          jmp mloop
72
73  error:  mov dx, 3f8h     ;read error char
74          in al, dx
75          mov al, '?'      ;show '?'
76          mov ah, 14
77          int 10h
78          jmp mloop
79
80  exit:   mov ah, 4ch      ;return to dos
81          int 21h
82  code    ends
83  end     start
```

流程图如图 1 所示。下面对主要代码进行分析：

- 16-24：写除数寄存器，先写低字节后写高字节。1200 对应的波特率在数据段由 divid 定义。
- 25-27：写线路控制字。8 个数据位、1 个停止位、奇校验规则。
- 28-30：写 MODEM 控制寄存器，自检模式。
- 31-33：写中断寄存器，屏蔽所有中断。
- 35-54：主循环。分以下几个部分

  - 35-36：读取线路状态寄存器。
  - 37-38：如果出错，转错误处理。
  - 39-40：如果接收就绪，转接收。
  - 41-42：如果发送寄存器不为空，则重新开始主循环。
  - 43-47：如果有按键则读取字符，否则重新开始主循环。
  - 48-49：显示原始字符。
  - 50-51：发送字符。
  - 52-53：如果为空格则退出

- 56-60：接收数据的程序块。读取数据，如果不为空格则显示；否则退出程序。
- 62-71：显示数据的程序块。显示字符并检测字符是否为回车符，若为回车符则追加一个换行。
- 73-78：错误处理程序块。对于错误的数据用'?'来代替显示。
- 80-81：程序退出，返回 DOS。

4.2 两台计算机发送并接收字符

代码如下所示，流程图仍为图 1 。事实上，本任务与自检模式下收发字符的程序只相差一处，即第 28 行写 MODEM 控制寄存器时应将自检模式改为正常模式，故不再赘述。

```
1  data    segment
2  divid   dw  60h
3  data    ends
4  stack   segment para stack
5          db 100 dup(?)
6  stack   ends
7  code    segment
8          assume  cs: code, ds: data, es:data, ss: stack
9  start:  mov ax, data
10         mov ds, ax
11
12 ;8250 initialize
13 ;baud rate=1200 data digits=8 stop digit=1 odd check normal
14 ;com 1
15 ;line control
16         mov al, 80h      ;dlab=1
17         mov dx, 3fbh
18         out dx, al
19         mov ax, divid    ;get divide number
20         mov dx, 3f8h
21         out dx, al       ;lower byte
22         mov al, ah
```

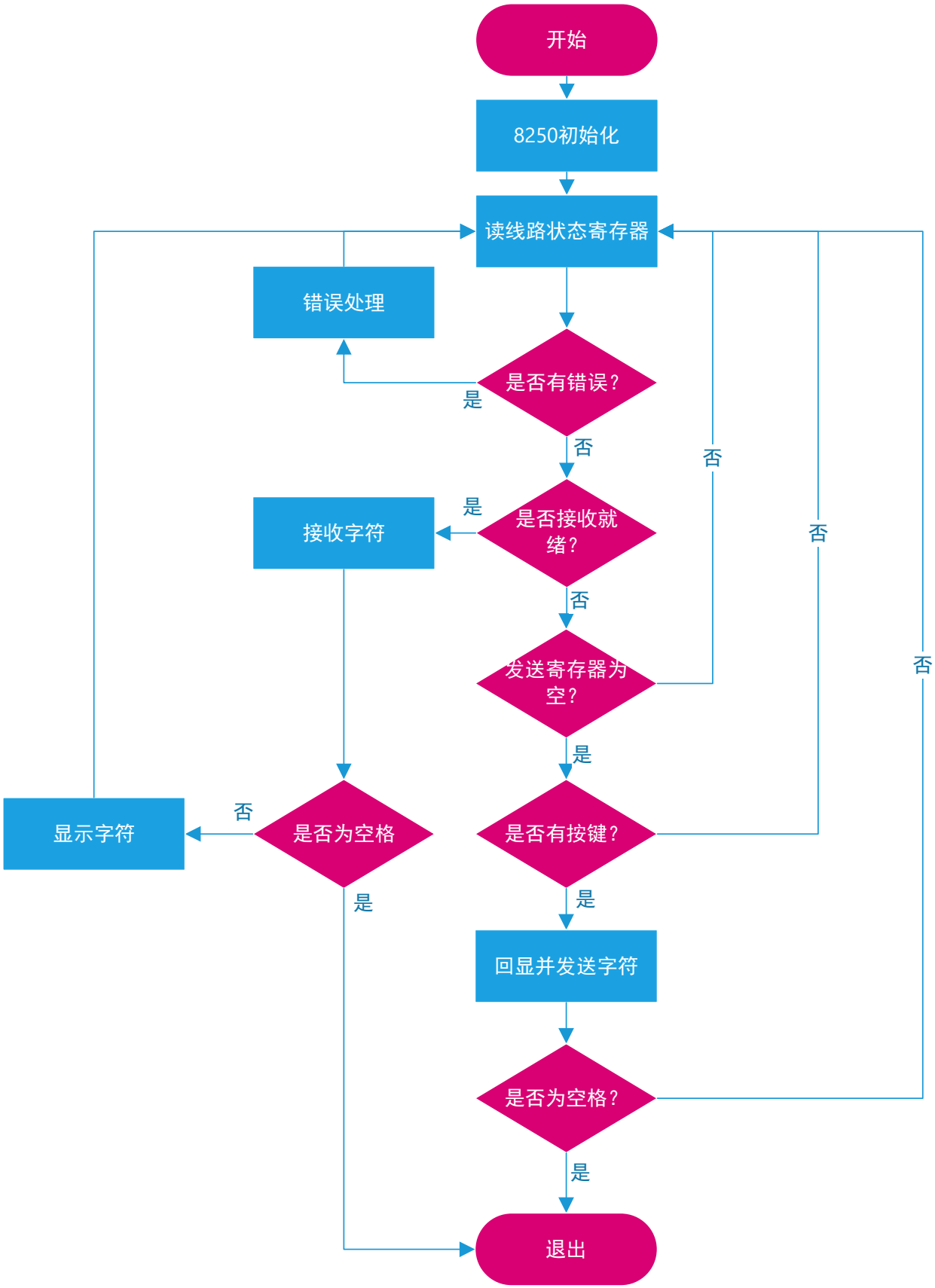
开始
8250初始化
读线路状态寄存器
错误处理
是否有错误？
是
否
否
否
否
接收字符
是
是否接收就绪？
否
发送寄存器为空？
是
显示字符
否
是否为空格
是否有按键？
是
是
回显并发送字符
是否为空格？
是
退出

图 1: 必做实验流程图

```
23          mov dx, 3f9h
24          out dx, al      ;higher byte
25          mov al, 0bh
26          mov dx, 3fbh    ;write line control register
27          out dx, al
28          mov al, 3h
29          mov dx, 3fch    ;MODEM control register
30          out dx, al
31          mov al, 0
32          mov dx, 3f9h    ;interupt register
33          out dx, al
34  ;main loop
35  mloop:  mov dx, 3fdh    ;read line status register
36          in al, dx
37          test al, 1EH    ;test error
38          jnz error
39          test al, 1h     ;receive ready?
40          jnz recei
41          test al, 20h    ;sender empty?
42          jz mloop
43          mov ah, 1
44          int 16h         ;read keyboard
45          jz mloop
46          mov ah, 0
47          int 16h
48          mov ah, 0eh     ;show original char
49          int 10h
50          mov dx, 3f8h    ;send
51          out dx, al
52          cmp al, 20h     ;space exit
53          jz exit
54          jmp mloop
55
56  recei:  mov dx, 3f8h    ;read data
57          in al, dx
58          cmp al, 20h
59          jnz char
60          jmp exit
61
62  char:   push ax
63          mov ah, 0eh     ;show the char
64          int 10h
65          pop ax
66          cmp al, 0dh     ;check the cr
67          jnz mloop
68          mov al, 0ah     ;new line
69          mov ah, 0eh
```

```
70          int 10h
71          jmp mloop
72
73  error:  mov dx, 3f8h     ;read error char
74          in al, dx
75          mov al, '?'      ;show '?'
76          mov ah, 14
77          int 10h
78          jmp mloop
79
80  exit:   mov ah, 4ch      ;return to dos
81          int 21h
82  code    ends
83  end     start
```

4.3 两台计算机的字符串传送

代码如下所示，流程图如图 2 所示。

```
1   data    segment
2   divid   dw  60h
3   string  db 'zwl', 0DH, 0AH, '$'
4   data    ends
5   stack   segment para stack
6           db 100 dup(?)
7   stack   ends
8   code    segment
9           assume  cs: code, ds: data, es:data, ss: stack
10  start:  mov ax, data
11          mov ds, ax
12
13  ;8250 initialize
14  ;baud rate=1200 data digits=8 stop digit=1 odd check normal
15  ;com 1
16  ;line control
17          mov al, 80h      ;dlab=1
18          mov dx, 3fbh
19          out dx, al
20          mov ax, divid    ;get divide number
21          mov dx, 3f8h
22          out dx, al       ;lower byte
23          mov al, ah
24          mov dx, 3f9h
25          out dx, al       ;higher byte
26          mov al, 0bh
27          mov dx, 3fbh     ;write line control register
28          out dx, al
```

```
29          mov al, 3h
30          mov dx, 3fch     ;MODEM control register
31          out dx, al
32          mov al, 0
33          mov dx, 3f9h     ;interupt register
34          out dx, al
35  ;main loop
36  mloop:  mov dx, 3fdh     ;read line status register
37          in al, dx
38          test al, 1EH     ;test error
39          jnz error
40          test al, 1h      ;receive ready?
41          jnz recei
42          test al, 20h     ;sender empty?
43          jz mloop
44          mov ah, 1
45          int 16h          ;read keyboard
46          jz mloop
47          mov ah, 0
48          int 16h
49          mov dx, 3f8h     ;send
50          out dx, al
51          cmp al, 20h      ;space exit
52          jz exit
53          cmp al, 'S'      ;send the string
54          jz s_str
55          cmp al, 'R'
56          jz r_str
57          jmp mloop
58
59  recei:  mov dx, 3f8h     ;read data
60          in al, dx
61          cmp al, 20h
62          jz exit
63          cmp al, 'S'
64          jz r_str
65          cmp al, 'R'
66          jz s_str
67          jmp mloop
68
69  error:  mov dx, 3f8h     ;read error char
70          in al, dx
71          mov al, '?'      ;show '?'
72          mov ah, 14
73          int 10h
74          jmp mloop
75
```

```
76  s_str:  lea si, string
77  s_loop: mov dx, 3f8h     ;send each
78          mov al, [si]
79          out dx, al
80          cmp al, '$'
81          jz mloop
82          inc si
83          jmp s_loop
84
85  r_str:  mov dx, 3fdh
86          in al, dx
87          test al, 1eH     ;check error
88          jnz error
89          test al, 1h
90          jz r_str
91          mov dx, 3f8h
92          in al, dx
93          cmp al, '$'
94          jz mloop
95          mov ah, 0eh
96          int 10h
97          jmp r_str
98
99  exit:   mov ah, 4ch      ;return to dos
100         int 21h
101 code    ends
102 end     start
```

相比于单个字符的传送，两台计算机的字符串传送的程序中多了以下几个部分：

- 53-56：检测到按键后，要判断是否为控制字符（发送或接受命令"R","S"），如果是则应该跳转到相应代码块处理。
- 76-83：发送字符串的代码块。首先让 SI 指向字符串，再通过一个循环将每一个字符发出，直到遇到字符串结尾的'$'
- 85-97：接受字符串的代码块。首先检测线路状态寄存器，如果有错误则跳转到错误处理模块；再检查是否接收就绪，如果未就绪则重新检查；确认就绪后读取字符，若是结束符'$' 则跳转回主循环，否则将其打印出来，并准备下一次的读取。

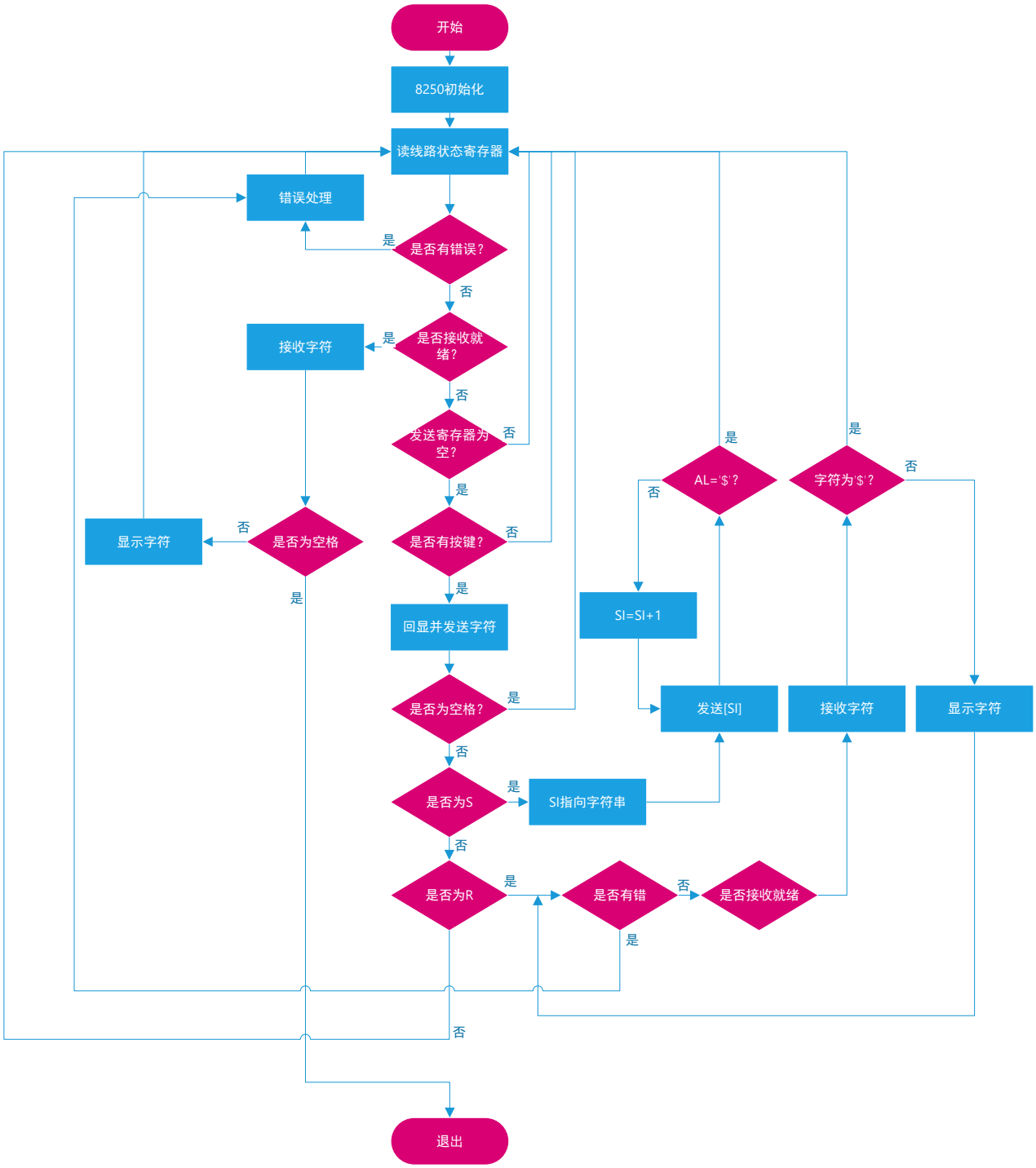
开始
8250初始化
读线路状态寄存器
错误处理
是否有错误?
是
否
接收字符
是否接收就绪?
是
否
发送寄存器为空?
否
是
显示字符
是否为空格
否
是否有按键?
否
是
是
回显并发送字符
是否为空格?
是
否
是否为S
是
否
是否为R
是
否
是否有错
否
是否接收就绪
是
是
否
AL='$'?
是
否
SI=SI+1
字符为'$'?
是
否
发送[SI]
接收字符
显示字符
SI指向字符串
退出

图 2: 选做实验流程图

5 完成情况和心得体会

本次实验的必做任务完成的较为顺利。其中第一个实验任务是在自检模式下收发字符，我参考了教材第 296 页的代码的思路完成了代码。由于是自检模式，在实验前，我已经在自己的电脑上顺利地测试通过。第二个实验任务与第一个任务几乎一样，唯一不同的地方是在写 MODEM 控制寄存器时应该将自检模式改为正常模式，所以在实验中我也很快完成了第二个任务。

而选做任务却没有那么顺利。我在测试时发现，当一台计算机发送字符串后，另一台计算机毫无反应。我首先检查了代码，发现我将第 88 行的 jnz 误写成了 jz，然而修改后仍然没有得到正确的结果。

后来我突然意识到，可能是由于我发送的字符串本身的问题。我原来的字符串设置的是我的英文名“JOHN WILLIAMS”，而这其中恰好有一个”S”。这可能会被程序误认为是发送指令，从而导致死循环。我将数据段中的字符串修改为“ZWL”后，代码就能够正常工作了。

之前在学习 8250 时，就曾经被它复杂的初始化编程困扰。为了完成这次实验，我认真地学习了教材中的例程，更加熟悉的 8250 的初始化、收发数据的编程方式。之前在做嵌入式开发时，我也经常会用到串口来进行调试。这个实验让我从原理上对串行通信有了更加深刻的认识，我也更加期待能够在小学期中的实验中获得更多的知识和锻炼。

参考文献

[1] 微型计算机系统及应用/杨素行等编著.—3 版.—北京：清华大学出版社,2009.4.ISBN 978-7-302-19352-4