串口通信仿真报告

1. 数字按键串口通信实例

两个 mcs51 之间的串口通信，在 proteus 和 keil 联合仿真， 推荐使用串口收发中断来处理收发事件，并扩展收发控制按键，和收发显示验证模块。(比如，通过按键控制什么时候开始收发，接收端的数据可以显示出来以便和发送端比较，是否一致等)。

具体演示例程，可以参考如下：在 51 单片机 1 上扩展数字按键，按数字键之后，通过串口传输到单片机 2，单片机 2 扩展数字显示 LED，显示通过串口传输过来的数字。

如：在单片机 1 按下数字 5，在单片机 2 可以显示数字 5。

设计思路：51 单片机 1 连接数字键盘进行发送任务，由按键连接的 p0.0 口控制发送；51 单片机 2 进行接收和 led 显示。

使用的程序与硬件连接如下图所示：

发送端代码：

homework7_1_trans.a51*

```
;TRANSMIT
ORG 0000H
    LJMP MAIN
ORG 1000H
MAIN:
    CLR EA
    MOV TCON, #20H        ;SET T1
    MOV TL1, #0FAH
    MOV TH1, #0FAH
    SETB TR1
    SETB ES
    MOV  PCON,#00H        ;SET UART
    MOV  SCON,#50H
    MOV P1,#00H
    READ:

    MOV P2, #0FFH             ;READ IN KEYBOARD
    CLR P2.3
    JNB P2.2, NUM1
    JNB P2.1, NUM2
    JNB P2.0, NUM3
    SETB P2.3
    CLR P2.4
    JNB P2.2, NUM4
    JNB P2.1, NUM5
    JNB P2.0, NUM6
    SETB P2.4
    CLR P2.5
    JNB P2.2, NUM7
    JNB P2.1, NUM8
```

homework7_1_trans.a51*

```
31      JNB P2.0, NUM9
32      SETB P2.5
33      CLR P2.6
34      JNB P2.1, NUM0
35      SETB P2.6
36      SJMP READ
37      NUM0:
38      MOV A,#3FH
39      SJMP SEND
40      NUM1:
41      MOV A,#06H
42      SJMP SEND
43      NUM2:
44      MOV A,#5BH
45      SJMP SEND
46      NUM3:
47      MOV A,#4FH
48      SJMP SEND
49      NUM4:
50      MOV A,#66H
51      SJMP SEND
52      NUM5:
53      MOV A,#6DH
54      SJMP SEND
55      NUM6:
56      MOV A,#7DH
57      SJMP SEND
58      NUM7:
59      MOV A,#07H
60      SJMP SEND
```

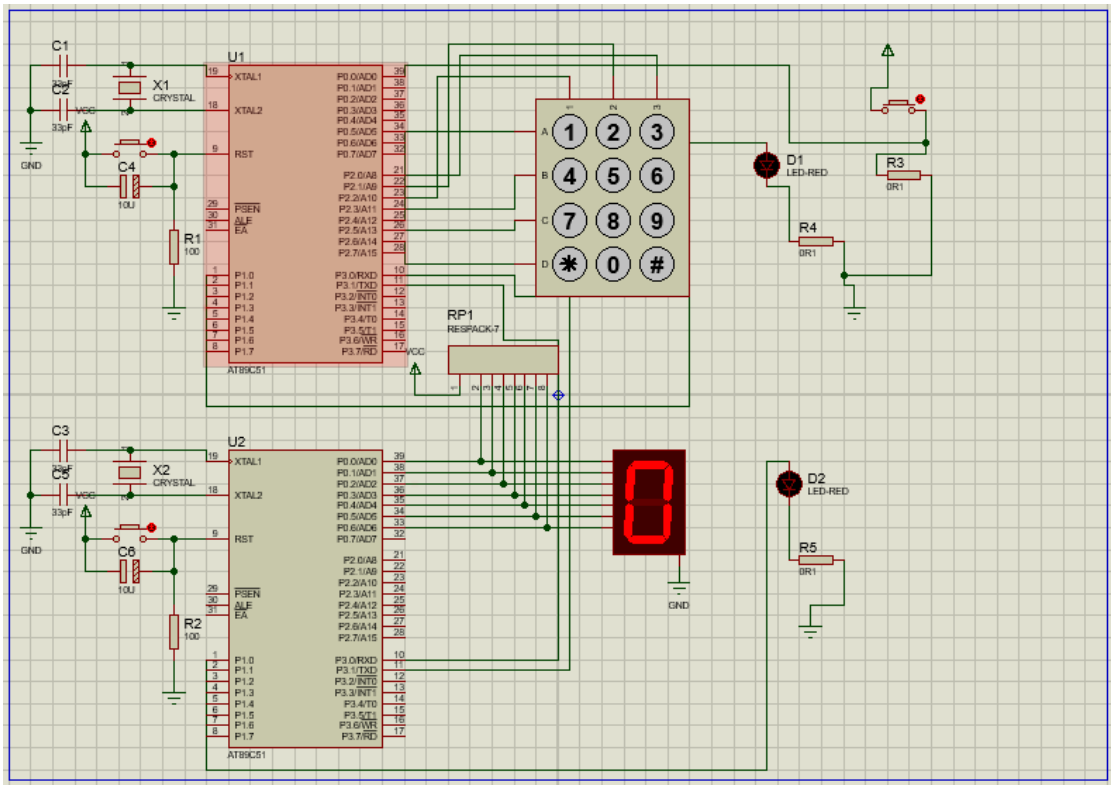
homework7_1_trans.a51*

```
48      SJMP SEND
49      NUM4:
50      MOV A,#66H
51      SJMP SEND
52      NUM5:
53      MOV A,#6DH
54      SJMP SEND
55      NUM6:
56      MOV A,#7DH
57      SJMP SEND
58      NUM7:
59      MOV A,#07H
60      SJMP SEND
61      NUM8:
62      MOV A,#7FH
63      SJMP SEND
64      NUM9:
65      MOV A,#6FH
66      SJMP SEND
67      SEND:     ;TRANSMIT
68
69      JNB P0.0,READ
70      MOV SBUF,A          ;UART TRANS
71
72      JNB  TI,$           ;WAIT FOR TRANS
73      CLR  TI             ;OPEN RE-TRANS
74      MOV P1,#0FH
75      LJMP READ
76  END
77
```

接收端代码:

homework7_1_recept.a51*

```
1  ;RECEPT
2  ORG 0000H
3      LJMP MAIN
4  ORG 0023H
5      LJMP RINT
6  ORG 1000H
7  MAIN:
8      SETB EA
9      MOV TCON, #20H           ;SET T1
10     MOV TL1, #0FAH
11     MOV TH1, #0FAH
12     SETB TR1              ;OPEN T1
13     SETB ES
14     MOV  PCON,#00H         ;SET UART
15     MOV  SCON,#50H
16     MOV P1,#00H
17     SJMP $
18 RINT:
19     MOV P1,#0FH
20     JB TI,INT_OVER        ;RI INTERUPT
21     CLR RI
22     MOV A,SBUF
23     MOV  P0,A             ;LED
24     INT_OVER:
25     RETI
26 END
27
```



发送端程序逻辑：

在主程序内完成定时器 T1 与 UART 的初始化，使用方式 1 进行串口通信，循环读取数字键盘的信号并转换成对应的七段数码管显示控制字符存入累加器 A 中，若 P0.0 上为高电位则表示发送控制已开启，将 A 存入 SBUF 进行发送，发送完成后清除 TI 准备下一次发送。

接收端程序逻辑：

在主程序内完成定时器 T1 与 UART 的初始化，使用方式 1 进行串口通信，之后在原地等待。在中断程序内先判断中断标志位 TI 与 RI，若为接收中断 RI，则读取 SBUF 中的信息并在 LED 上显示，复位后准备下一次接收。

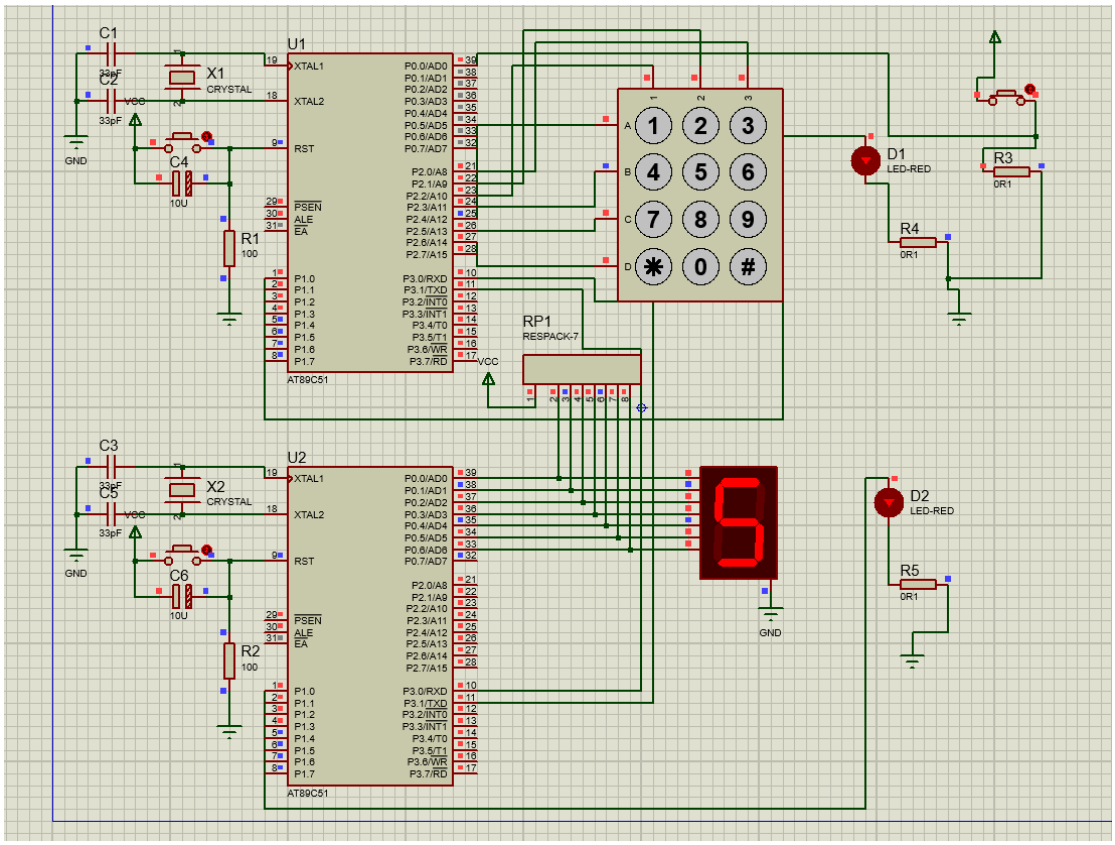
其中两单片机的 P1.0 口连接 LED-RED 分别显示 TI 与 RI 的状态，以便于调试。

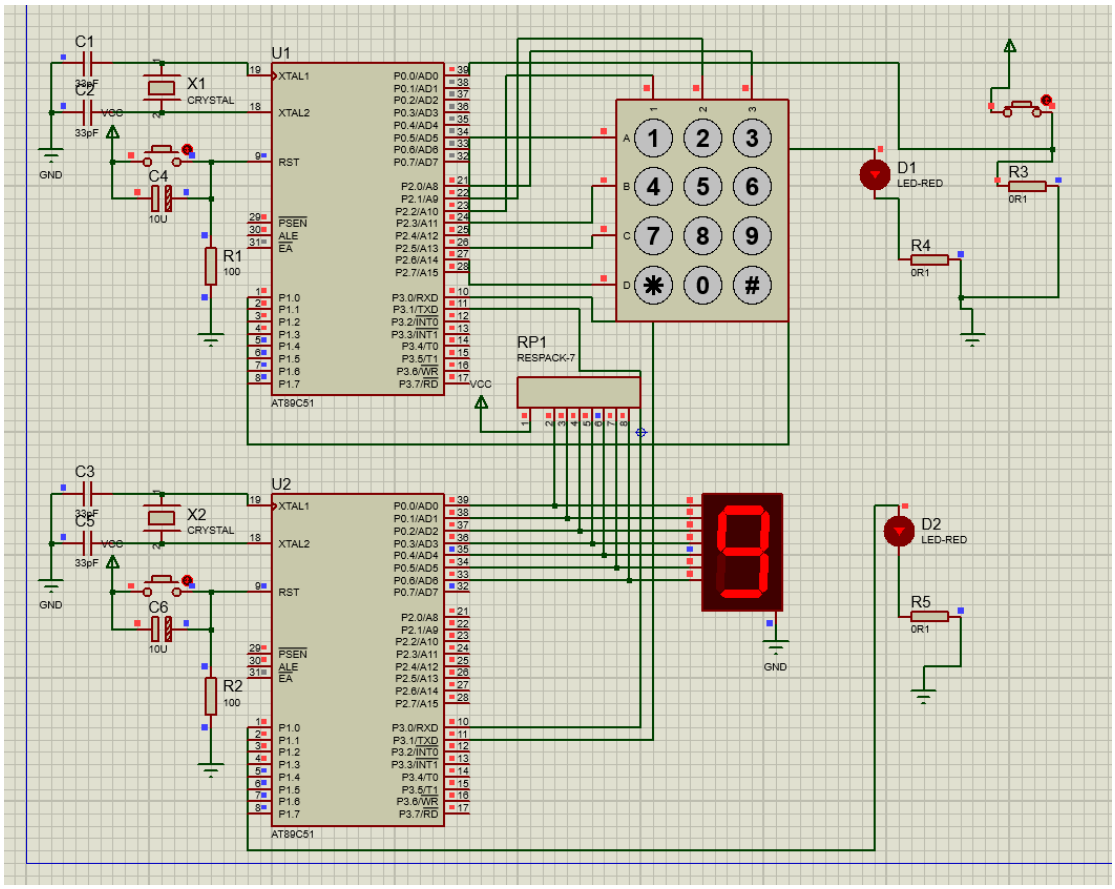
硬件连接：

单片机 1 P2 口用于从数字键盘输入需要传输的信息，P0.0 口用于控制发送，RXD 与 TXD 分别与单品机 2 的 TXD 和 RXD 进行连接；单片机 2 的 P0 后进行数码管的输出。

结果：

单片机 1 P0.0 口的按键按下后，可完成在单片机 1 上按数字键之后，通过串口传输到单片机 2，单片机 2 扩展数字显示 LED，显示通过串口传输过来的数字。


C1
33pF
C2
33pF
X1
CRYSTAL
GND
VCC
U1
XTAL1
XTAL2
RST
C4
10U
R1
100
PSEN
ALE
EA
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
AT89C51
RP1
RESPACK-7
D1
LED-RED
R3
0R1
R4
0R1
C3
C5
X2
C6
R2
D2
R5


C1
33pF
C2
33pF
X1
CRYSTAL
GND
VCC
U1
XTAL1
XTAL2
RST
C4
10U
R1
100
PSEN
ALE
EA
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
AT89C51
RP1
RESPACK-7
D1
LED-RED
R3
0R1
R4
0R1
C3
C5
X2
C6
R2
D2
R5

具体可见附件视频

2. 两单片机之间串口通信实例验证

其程序逻辑为：

发送端：发出联络信号 E1H->接收端：接收并发出联络信号 E2H->发送端：接收验证后发出数据块（循环）->接收端：接收数据块->发送端：发送校验位->接收端：接收校验位并验证，若正确则结束，若错误则重复此过程。

为了展示通信正确的结果，当验证正确时，对发送端程序添加指令点亮 P1.0 口连接的 LED，并使两程序均在原地等待。

使用的程序和硬件连接图为：

发送端：

homework7_2_trans.a51*

```
1  ;TRANSMIT
2  ORG 0000H
3      LJMP START
4  ORG 0100H
5  START:CLR  EA        ;INIT
6             MOV  TMOD,#20H
7             MOV  TH1,#0F4H
8             MOV  TL1,#0F4H
9             MOV  PCON,#00H
10            SETB  TR1
11            MOV P1,#00H
12            MOV  SCON,#50H
13   ALOOP1:MOV  SBUF,#0E1H  ;SEND HANDSHAKE SIGNAL
14            JNB  TI,$
15            CLR  TI
16            JNB  RI,$           ;RECEIVE
17            CLR  RI
18            MOV  A,SBUF
19            XRL  A,#0E2H        ;VERIFY HANDSHAKE
20            JNZ  ALOOP1
21
22   ALOOP2:MOV  R0,#40H
```

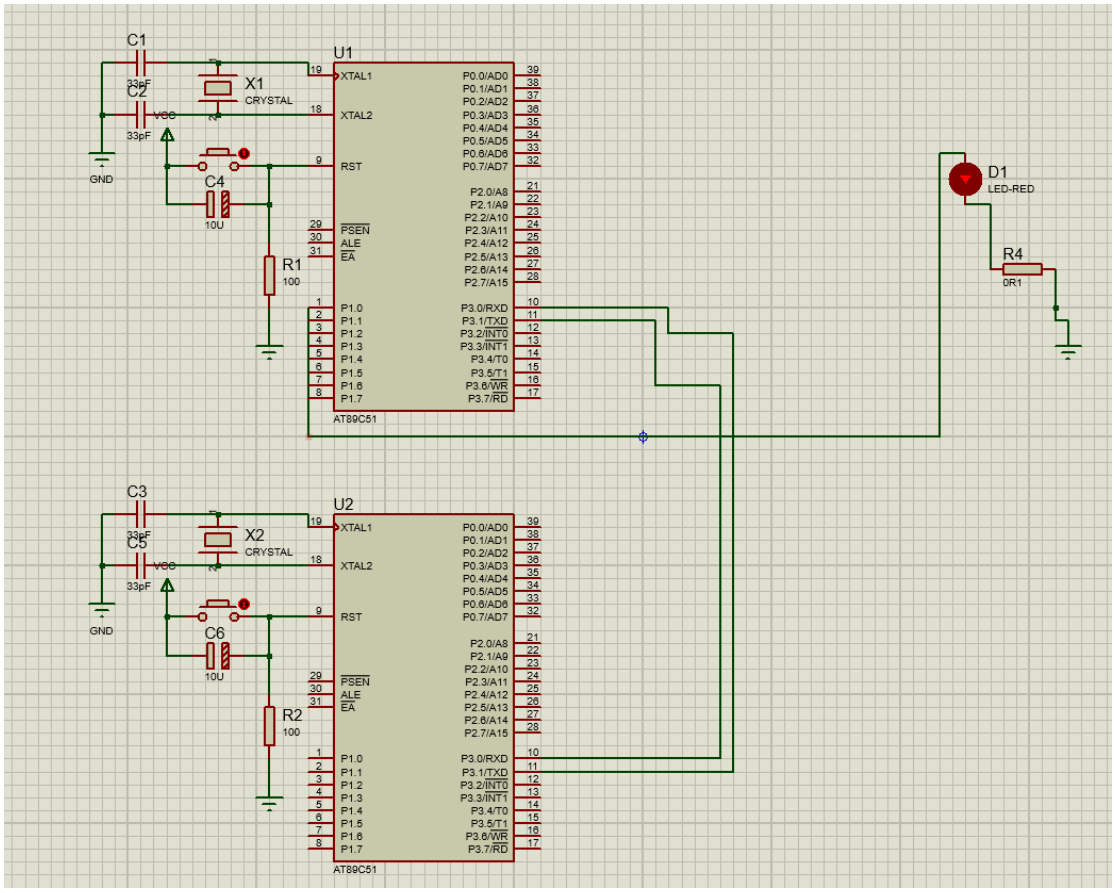
homework7_2_trans.a51*

```
17              CLR  RI
18              MOV  A,SBUF
19              XRL  A,#0E2H          ;VERIFY HANDSHAKE
20              JNZ  ALOOP1
21
22   ALOOP2:MOV  R0,#40H
23              MOV  R7,#10H          ;DATA LENGTH
24              MOV  R6,#00H
25  ALOOP3:MOV  SBUF,@R0     ;SEND DATA
26              MOV  A,R6
27              ADD  A,@R0
28              MOV  R6,A
29              INC R0
30              JNB  TI,$
31              CLR  TI
32              DJNZ  R7,ALOOP3
33
34              MOV  SBUF,R6          ;SEND CHECKSUM
35             JNB  TI,$
36              CLR  TI
37
38              JNB  RI,$             ;RECEIVE VERIFICATION RESULT
39              CLR  RI
40
41              MOV  A,SBUF
42              JNZ   ALOOP2      ;WRONG
43              MOV P1,#0FH
44              SJMP $
45  END
```

接收端:

homework7_2_recept.a51*

```
;RECEPT
ORG 0000H
    LJMP START
ORG 1000H
START:CLR   EA          ;INIT
            MOV  TMOD,#20H
            MOV  TH1,#0F4H
            MOV  TL1,#0F4H
            MOV  PCON,#00H
            SETB  TR1
            MOV  SCON,#50H
  BLOOP1:JNB  RI,$           ;RECEIVE HANDSHAKE
            CLR  RI
            MOV  A,SBUF
            XRL  A,#0E1H        ;VERIFY HANDSHAKE
            JNZ   BLOOP1
            MOV  SBUF,#0E2H     ;SEND HANDSHAKE SIGNAL
            JNB   TI,$
            CLR   TI
            MOV  R0,#40H
            MOV  R7,#10H
            MOV  R6,#00H
BLOOP2:JNB  RI,$            ;RECEIVE DATA
            CLR   RI
            MOV  A,SBUF
            MOV  @R0,A
            INC   R0
            ADD  A,R6          ;COUNT CHECKSUM
            MOV  R6,A
            DJNZ  R7,BLOOP2
            DJNZ  R7,BLOOP2
            JNB  RI,$             ;RECEIVE CHECKSUM
            CLR  RI
            MOV  A,SBUF
            XRL  A,R6            ;VERIFY
            JZ   END1
            MOV  SBUF,#0FFH      ;WRONG
            JNB  TI,$
            CLR  TI
END1:MOV  SBUF,#00H              ;CORRECT
            JNB  TI,$
            CLR  TI
SJMP $
END
```



运行结果：

传送完毕后 D1 点亮，说明信号传输无误。