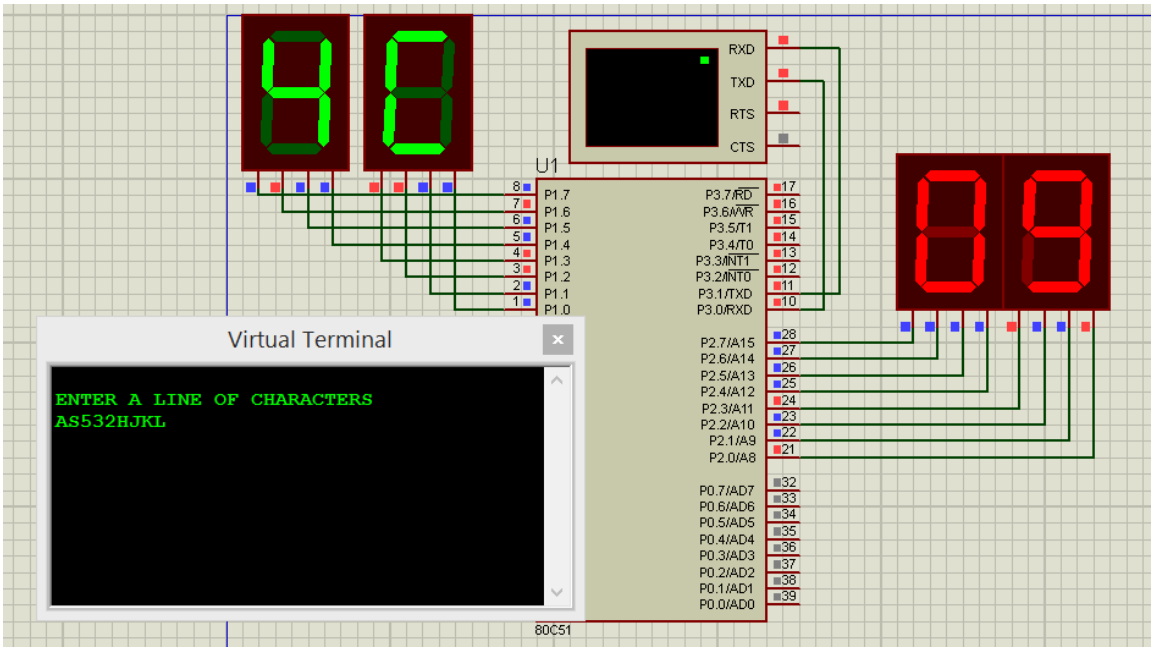
Problem 19



This project reads a line of characters from the serial port and display the character code on port P1, and the number of characters on 7-segment connected to port P2

Character codes 13 "CR" and 10 "LF" indicates line end

Variable definitions

```
4
5  COUNT EQU 30H
6  DISP_PORT EQU P2
```

We need a count variable to hold the number of characters received (5)

DISP_PORT → the port used for displaying the number of characters

Main code

```
20  Start:
21     MOV SCON,#50H         ;Asynchronous mode, 8-bit dat
22     MOV TMOD,#20H         ;Timer1 in Mode2.
23     MOV TH1,#253          ; // Load timer value for bau
24     MOV TL1,#253
25     SETB TR1      ;       //Turn ON the timer for Baud r
26
27     MOV P1,#0
28     MOV P2,#0
29  START2:
30     MOV DPTR,#WELCOME_MSG
31     CALL PRINT_MSG
```

Lines 21-25 is the same as previous problem where we setup the serial controller to work with 9600 baud rate

We use the same previous function to print a welcome message (30-31)

```
33  LOOP:
34     LCALL RECEIVE_BYTE
35     CJNE A,#0DH,PRINT_CHR
36     JMP LINE_END
37     CJNE A,#0AH,PRINT_CHR
38     JMP LINE_END
39  PRINT_CHR:
40     MOV P1,A
41     CALL INC_BCD
42     MOV P2,COUNT
43     LCALL SEND_BYTE
44     JMP LOOP
45  LINE_END:
46     MOV P2,COUNT
47     MOV COUNT,#0
48     JMP START2
```

In this part, we receive a byte (34), test if it is CR or LF to end line reception (35-38).

If not line end, we display the character code on P1 (40) and increment the count in BCD to make it ready for display on 7-segment(41). Then we display it (42). Line 43 echo received character back.

If line end received, we reset the count to be ready for another line

Functions

Same previous functions are used (SEND_BYTE, RECEIVE_BYTE, PRINT_MSG).

The new function is INC_BCD

```
77  INC_BCD:
78      PUSH ACC
79      MOV A,COUNT
80      ADD A, #1
81      DA A
82      MOV COUNT,A
83      POP ACC
84  RET
```

We save ACC into the stack where we will use it in this function (78) and after finishing the function, we will restore it back (83)

We increment the count by '1' using ADD instruction followed by DA instruction (80-81), where DA→ decimal adjust → will adjust the result in BCD after the addition by '1' (Note: DA has no effect for all instruction except "ADD")