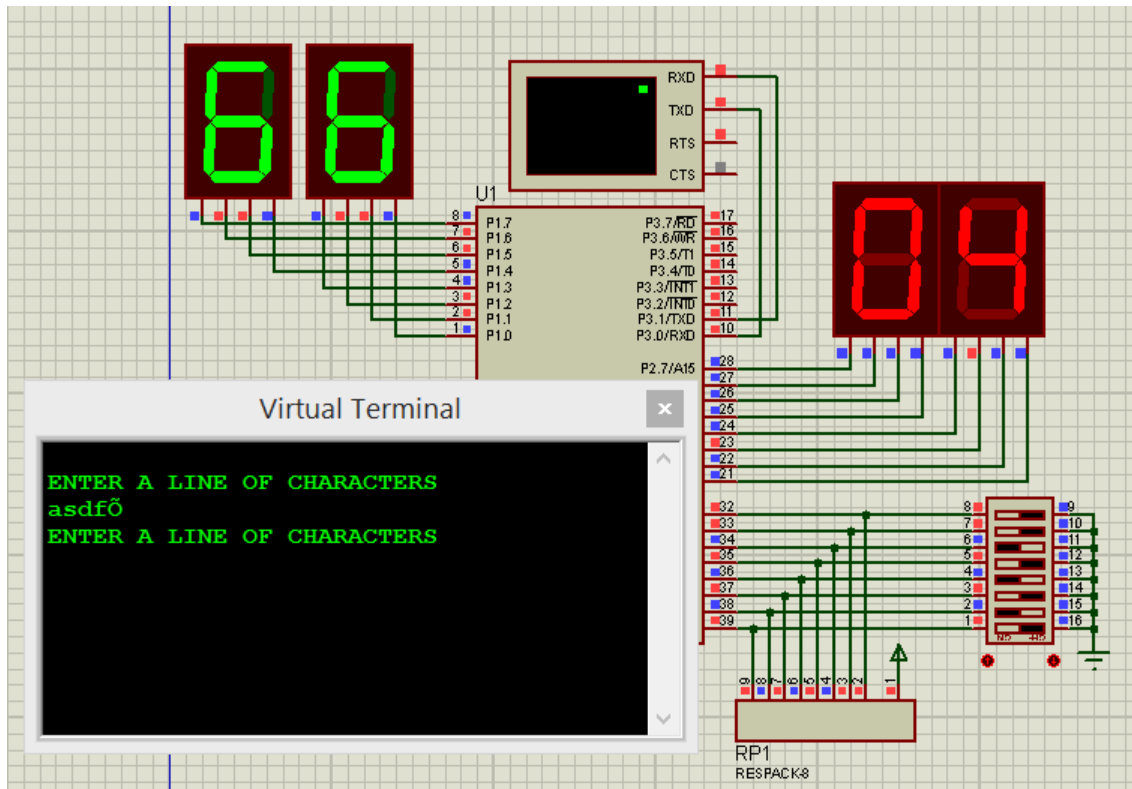


## Problem 20



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

This example will end line reception if there is no character received for a certain time "TimeOut". In our example we set Timeout to 2 seconds approximately

Also this program will send the character code read from port0 to the terminal after the timeout period

This program is typically similar to previous problem with the following differences.

No need to test for CR or LF, and this is replaced by a testing the carry flag after calling the "RECEIVE\_BYTE" function. This function will return either by a received character with carry flag cleared, or no character received after timeout period with carry flag set.

Here is the function

```

75 RECEIVE_BYTE:
76 WAIT_RX:
77 L4:
78     MOV R5,#10
79 L3:
80     MOV R6,#200
81 L2:
82     MOV R7,#250
83 L1:
84
85     JB RI,RECEIVED      ; Wait till the data is received
86     DJNZ R7,L1
87     DJNZ R6,L2
88     DJNZ R5,L3
89     DJNZ R4,L4
90     SETB C
91     RET
92 RECEIVED:
93     CLR RI              ; Clear Receive Interrupt Flag for next cycle
94     MOV A,SBUF          ; return the received char
95     CLR C
96     RET

```

It will wait for RI to be '1' indicating new byte received but at the same time it will count for a timeout period. To do this, we put the RI test instruction (85) inside the 3 nested delay loops (R5,R6,R7) (78-89). If RI is set before nested loops finished, then we will clear the carry flag and put the received character in ACC(93-95). If the nested loop finished counting, this means that no data received within the timeout period → so we set the carry and return (90-91)

The timeout period can be calculated as follows approximately

$$4\mu s * 250 * 200 * 10 = 2 \text{ seconds}$$

We have 4 $\mu$ s instead 2 $\mu$ s as in delay subroutines because another instruction is executed inside the loops (85)

The other difference will be found in the main loop

```

34 START2:
35     MOV DPTR,#WELCOME_MSG
36     CALL PRINT_MSG
37 FIRST_CHR:
38     JNB RI,FIRST_CHR
39 LOOP:
40     MOV R4,#1
41     LCALL RECEIVE_BYTE
42     JC TIMEOUT
43 PRINT_CHR:
44     MOV P1,A
45     CALL INC_BCD
46     ;MOV P2,COUNT
47     LCALL SEND_BYTE
48     JMP LOOP
49 TIMEOUT:
50     MOV A,P0
51     CALL SEND_BYTE
52     MOV P2,COUNT
53     JMP START2

```

First it will wait for the first character to be received by testing the RI directly (38) 40-41, we call the receive function with  $R4 = 1$ , so the timeout will be multiplied by 1

If carry = 1 (42) this means timeout occurs and we end the line reception by reading the character code from P0 and sending it (50-51). Also we display the character count on P2 (52).