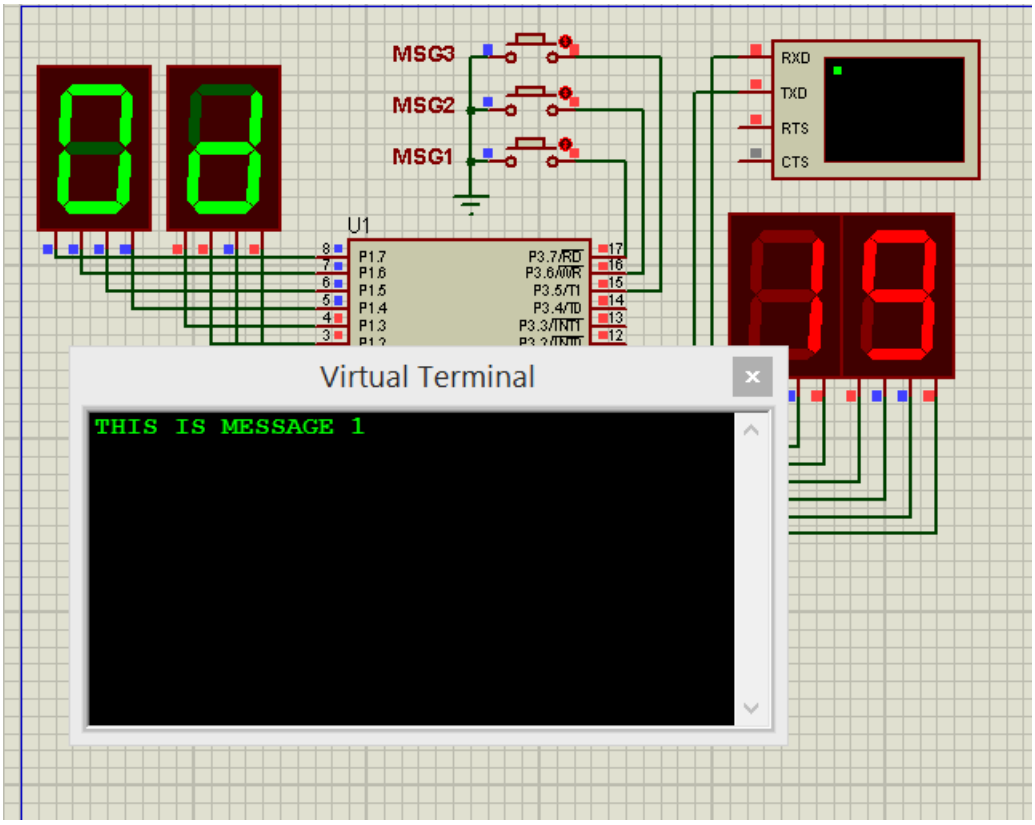
Problem 22



In this problem, we send 3 message according to which switch is being pressed

During the send process, we display the code of the character on P1, and the count of characters on P2

Message are stored with a null character that defines its end

Variables

```
1  NULL EQU 00H
2  COUNT EQU 30H
3  MSG1_BUTTON EQU P3.7
4  MSG2_BUTTON EQU P3.6
5  MSG3_BUTTON EQU P3.5
```

Null is the code for the character that defines the message end

Count is the variable for storing the number of characters being sent

Lines 3-5 define the port pin connected to the switch button

Main code

```
11  START:
12     CALL INIT_SERIAL
13     MOV COUNT,#0
14
15  START2:
16     MOV COUNT,#0
17     JNB MSG1_BUTTON,LOAD_MSG1
18     JNB MSG2_BUTTON,LOAD_MSG2
19     JNB MSG3_BUTTON,LOAD_MSG3
20     JMP START2
21  LOAD_MSG1:
22     MOV DPTR,#MSG1
23     JMP DO_SEND
24  LOAD_MSG2:
25     MOV DPTR,#MSG2
26     JMP DO_SEND
27  LOAD_MSG3:
28     MOV DPTR,#MSG3
29  DO_SEND:
30     CALL SEND_MSG
31     MOV R5,#3
32     CALL DELAY_100MS
33     JMP START2
```

First, we initialize serial as before (11)

The main loop start at (15) by resetting the count, and testing which button being pressed(17-19). For each button pressed, we load DPTR with the address of the corresponding message, then we proceed to (29) do_send: where we call the function send_msg, then we pause for 300ms to enable switch bounce.

Functions

1-Send_MSG

```
36  SEND_MSG:
37     CLR A
38     MOVC A,@A + DPTR
39     INC DPTR
40     CJNE A, #NULL, NOT_MSG_END
41     RET
42  NOT_MSG_END:
43     MOV P1,A
44     CALL INC_BCD
45     MOV P2,COUNT
46     CALL SEND_CHAR
47     MOV R5,#2
48     CALL DELAY_100MS
49     JMP SEND_MSG
50
51  SEND_CHAR:
52     MOV SBUF,A           ; Load the data to be transmitted
53  WAIT_TX:
54     JNB TI, WAIT_TX      ; Wait till the data is trasmitted
55     CLR TI               ; Clear the Tx flag for next cycle.
56     RET
```

We read message character using DPTR as before(37-39) till we find the null character(40) to end the process.

For each character we show it on P1 (43), increment the counter (44) and display it on P2, then we send it to serial TX using function SEND_CHAR.
In this routine we make a delay between characters during transmission to enable the user read the character count and code. We call delay_100ms which will pause for a time = the value of R5*100ms = 200ms in our case

Message format

```
89  MSG1: DB "THIS IS MESSAGE 1",10,13,0
90  MSG2: DB "THIS IS MESSAGE 2",10,13,0
91  MSG3: DB "THIS IS MESSAGE 3",10,13,0
```

Here is the definition of the messages with each message terminated by '0'