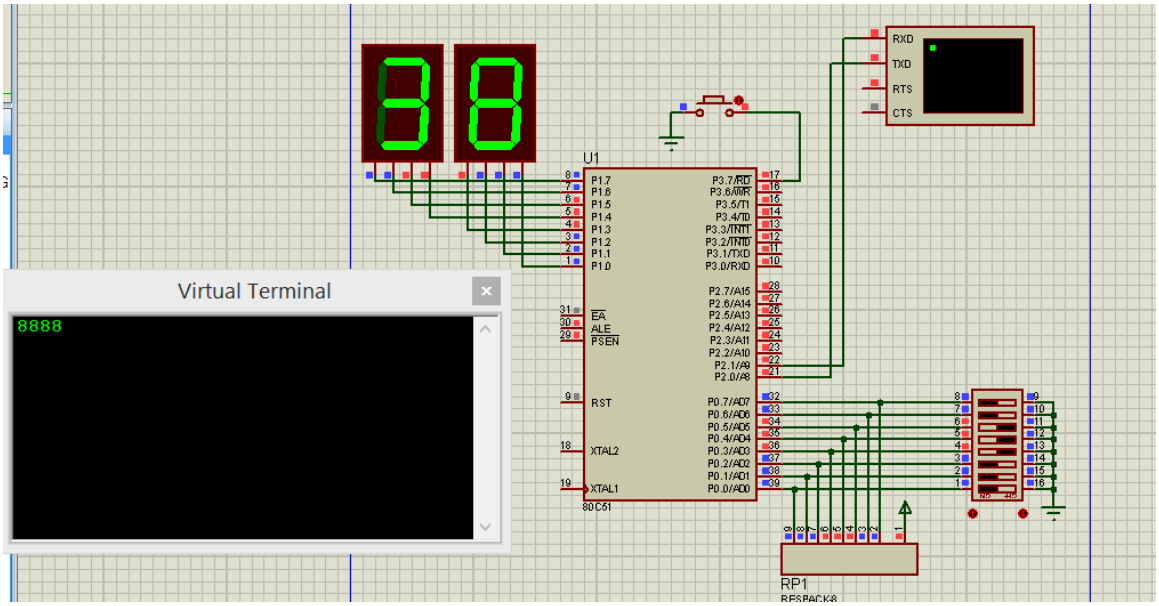
Problem 24



In this problem, we use software serial implementation to trasnmit the byte code at P0 serially through pin P2.1 with a baud rate of 9600 and 1 stop bit and even parity

We use button to send the byte

Variables

```
1  TX_PIN  EQU     P2.1            ;Transmit on this pin
2  RX_PIN  EQU     P2.0            ;Receive on this pin
3
4  SEND_BUTTON EQU P3.7
5  PARITY EQU 00H
6  ;Formula to calculate the bit time delay constant
7  ;This constant is calculated as: (((crystal/baud)/12) - 5) / 2
8  ;crystal is the frequency of crystal in Hz
9  ;baud is required baudrate
10 ;Please try to keep baudrate below 9600
11 ;to get best results :)
12
13 BIT_TIME  EQU   45;            (((11059200/9600)/12) - 5) / 2
```

1,2→ definition of the software TX and RX

4→ button used to enable transmission

5→ a bit variable used to calculate the parity bit

13→ is the number of cycles to repeat an instruction that will cause a delay = bit-time

Main code

```
18  START:
19     MOV A,P0
20     MOV P1,A
21     JB SEND_BUTTON,START
22     CALL SEND_CHAR
23     MOV R5,#2
24     CALL DELAY_100MS
25     JMP START
```

19,20→ read P0 and pass it to P1 for displaying the current byte

21→ if send byte is pressed "0", we call function 'send_char'

Functions

1-Send_Char

```
27  SEND_CHAR:
28     CALL CALC_PARITY
29     CLR TX_PIN
30     MOV R0,#BIT_TIME
31  WAIT_START_BIT:
32     DJNZ R0,WAIT_START_BIT
33
34     MOV R1,#8
35  NEXT_BIT:
36     RRC A
37     MOV TX_PIN,C
38     MOV R0,#BIT_TIME
39  WAIT_BIT:
40     DJNZ R0,WAIT_BIT
41     DJNZ R1,NEXT_BIT
42
43     MOV TX_PIN,PARITY
44     MOV R0,#BIT_TIME
45  WAIT_PARITY:
46     DJNZ R0,WAIT_PARITY
47     SETB TX_PIN
48     RRC A
49     MOV R0,#BIT_TIME
50  STOP_BIT:
51     DJNZ R0,STOP_BIT
52  RET
```

First we calculate the parity bit (28), then we make TX=0(29) → this is the start bit; then wait for a bit time (30-32)

Now we send the byte; bit by bit with a time delay of bit-time (34-41) → note that we use the RRC instruction to isolate bits of ACC into the carry (36), then sending it to the TX_PIN(37)

43-46 → send the parity bit

47-51 → send '1' → stop bit

Now, how 1-bit delay time is done?

```
    SETB TX
    MOV R0,#BIT_TIME
L1:
    DJNZ R0,L1
```

$$total\ time = \frac{5 + 2 * bit_{time}}{crystal\ freq/12} = \frac{1}{9600}$$

2→ is the number of cycles for the DJNZ

5 → is the total cycles for SETB ,and MOV R0

If we use 11.0592Mhz crystal

$$bit_{time} = \frac{\left(\frac{crystal\ freq}{12 \times 9600} - 5\right)}{2} = 45$$

2-CALC_PARITY

```
54  CALC_PARITY:
55     PUSH ACC
56     CLR PARITY
57     MOV R0,#8
58  ALL:
59     RRC A
60     JNC NO_C
61     CPL PARITY
62  NO_C:
63     DJNZ R0,ALL
64     POP ACC
65  RET
```

Start by zero parity bit (56); then we scan all bit (57,63) as follows

Get a bit into carry (59); if carry = 0 → do nothing; else complement the current parity (if it 0 → 1: if it is 1 → 0)

This will add all bits of the accumulator and give an even parity.

If we need an odd parity → just compement the final value of parity before

returning;