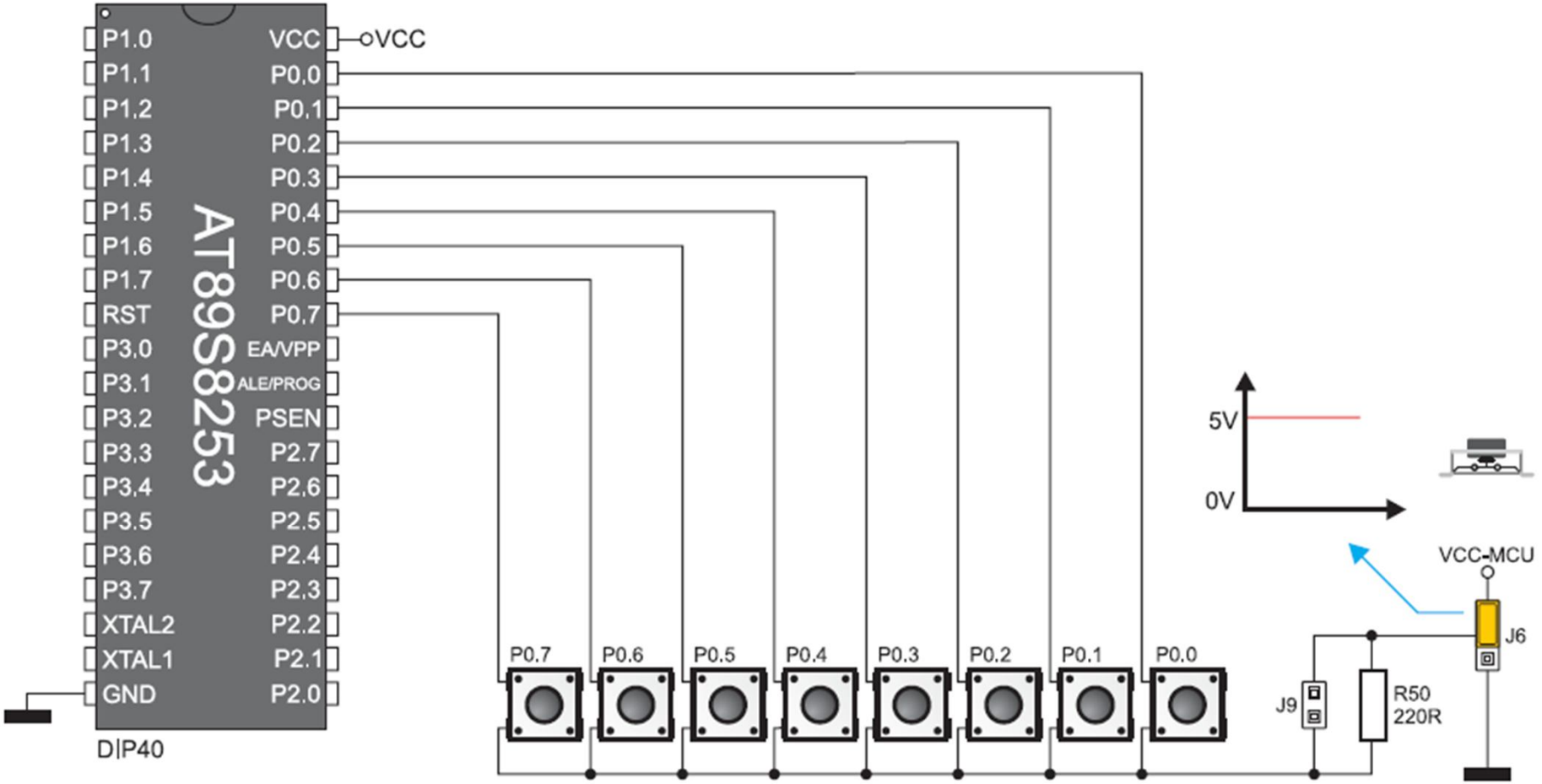
Microprocessors and Microcontrollers (EE-231)

Lab-5

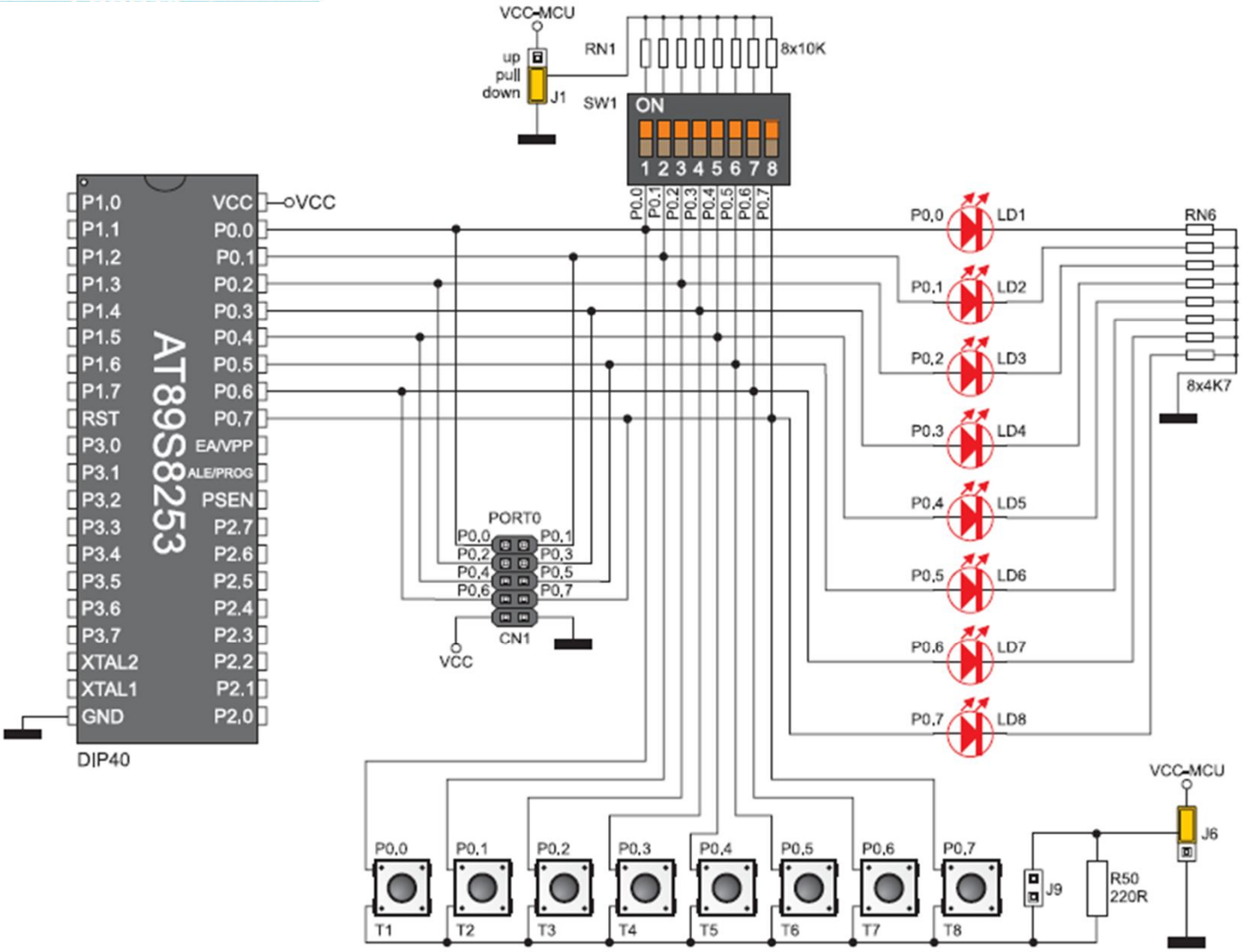
Main Objectives

- Learning to use DIP switch on 8051 board as I/Os
- Viewing memory contents in KEIL
- Implementation of a simple 4-bit (Hex) Calculator on Proteus
- Implementation of a simple 4-bit (Hex) Calculator on Easy8051v6 Development Board

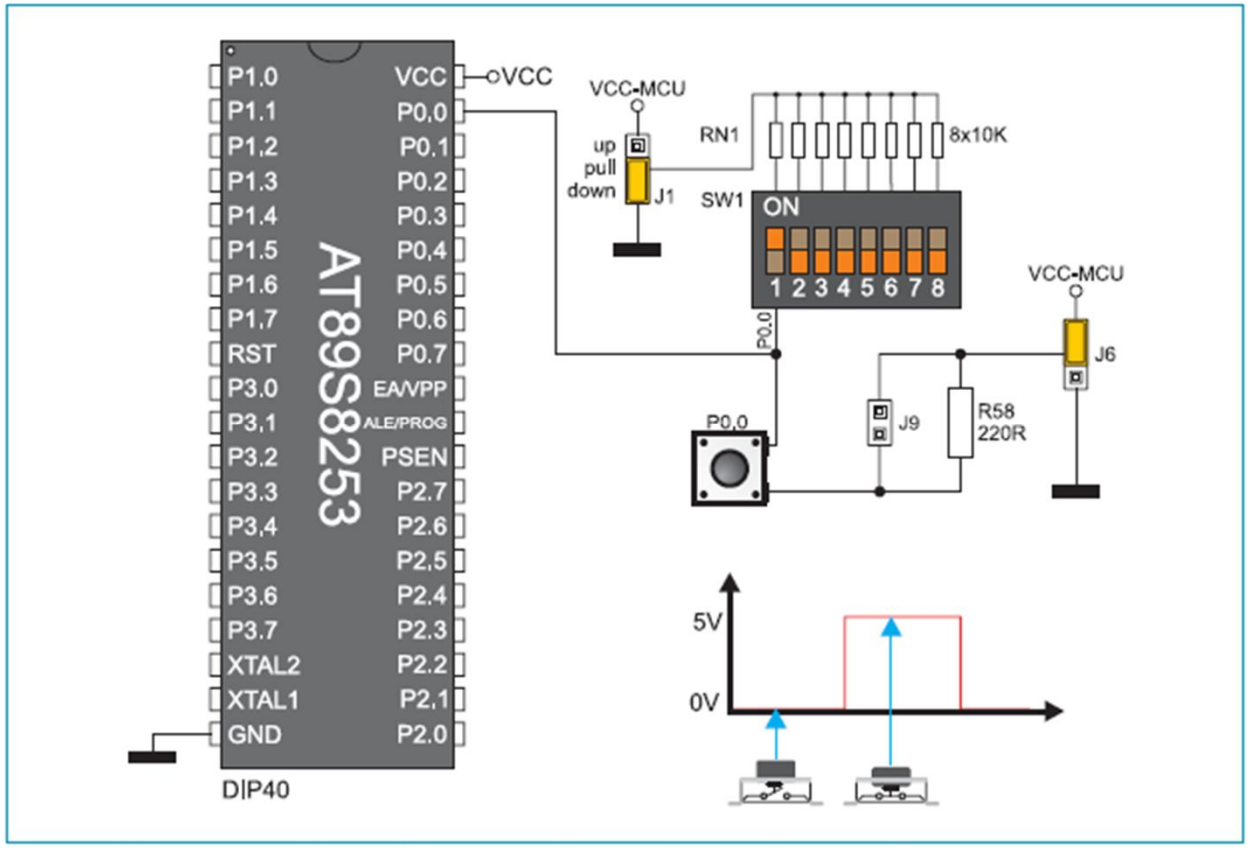
Connection of Push Buttons in Easy 8051 Kit



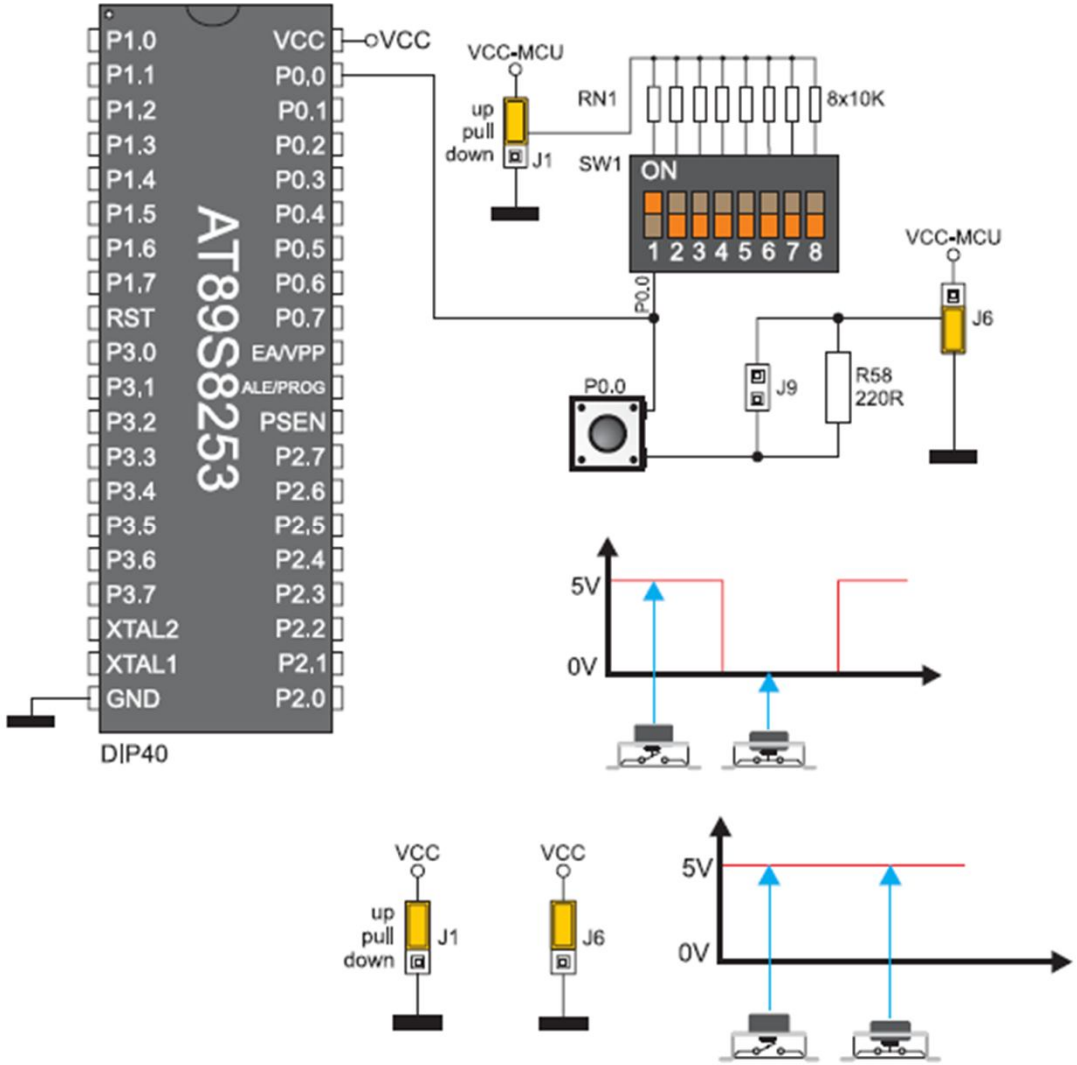
Connection of DIP Switches in Easy 8051 Kit



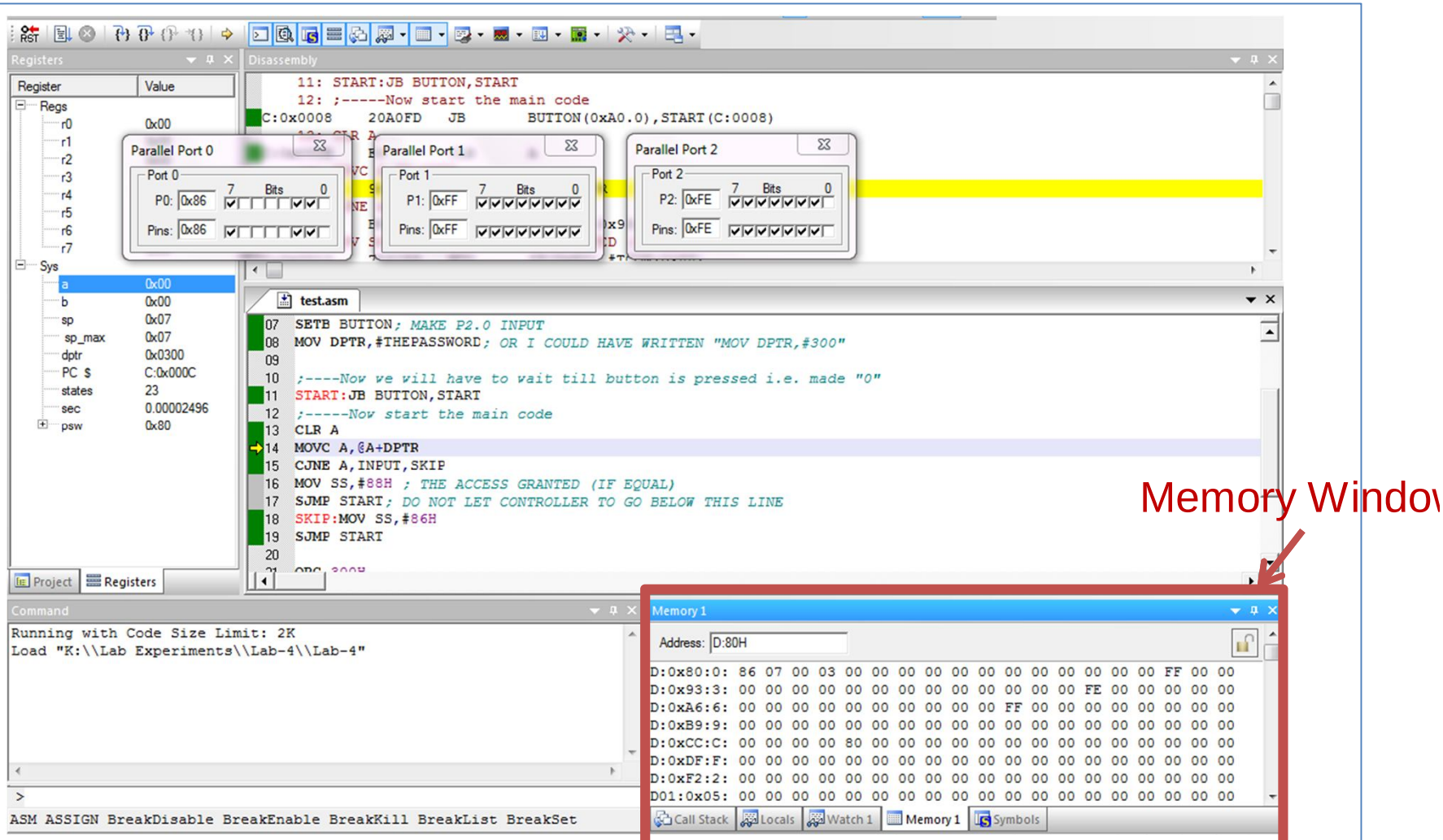
Connection of DIP Switches in Easy 8051 Kit



Connection of DIP Switches in Easy 8051 Kit



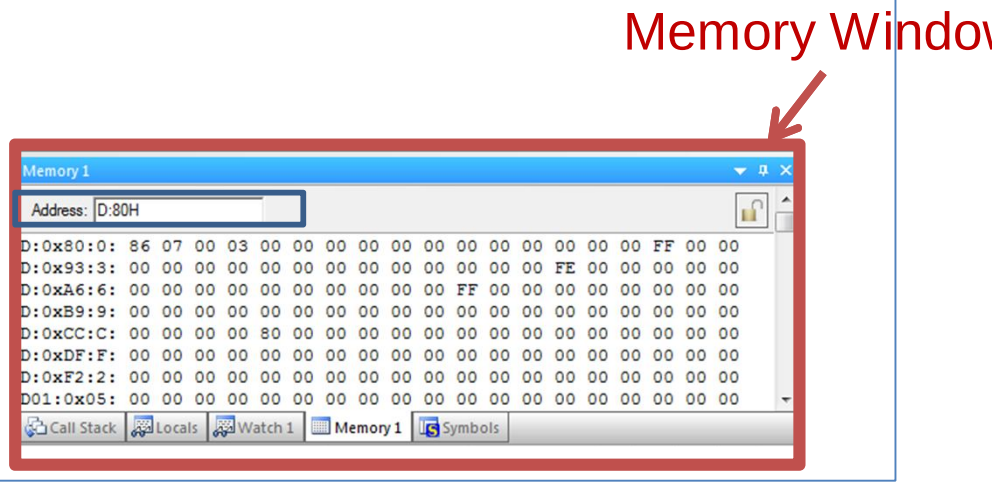
To view Memory in KEIL



To view Memory in KEIL

- We can view memory contents while **debugging**.
- The window in bottom-right corner shows the contents of each location
- By writing in **Address** space, we can access different memories and their different addresses. Following format is used
- [**Memory Type**]:[**starting location you want to view**]
- E.g. D:10H means the address number 10 of data memory
- We can view at a time 4 windows by opening more by going to **view** tab.

Key	Memory Type
C	Code
D	Data
X	External



Todays Task 1

- You will have to implement todays task 1 in Proteus.
- Make a calculator, keeping in view the guidelines mentioned below.
- Use P1 as inputs of two operands. 4-bit for each input i.e. 4 bit of high nibble and 4-bit of lower nibble.
- Use P3.1 and P3.2 as a selection for operation i.e. Add(1) or Subtract(0), Multiply (2) and Divide (3). Display the result on P2 LEDs. Debug as well as implement the design on proteus.

Operation	P3.1-P3.0
Add	0-0
Subtract	0-1
Multiply	1-0
Divide	1-1

Task 1 Code

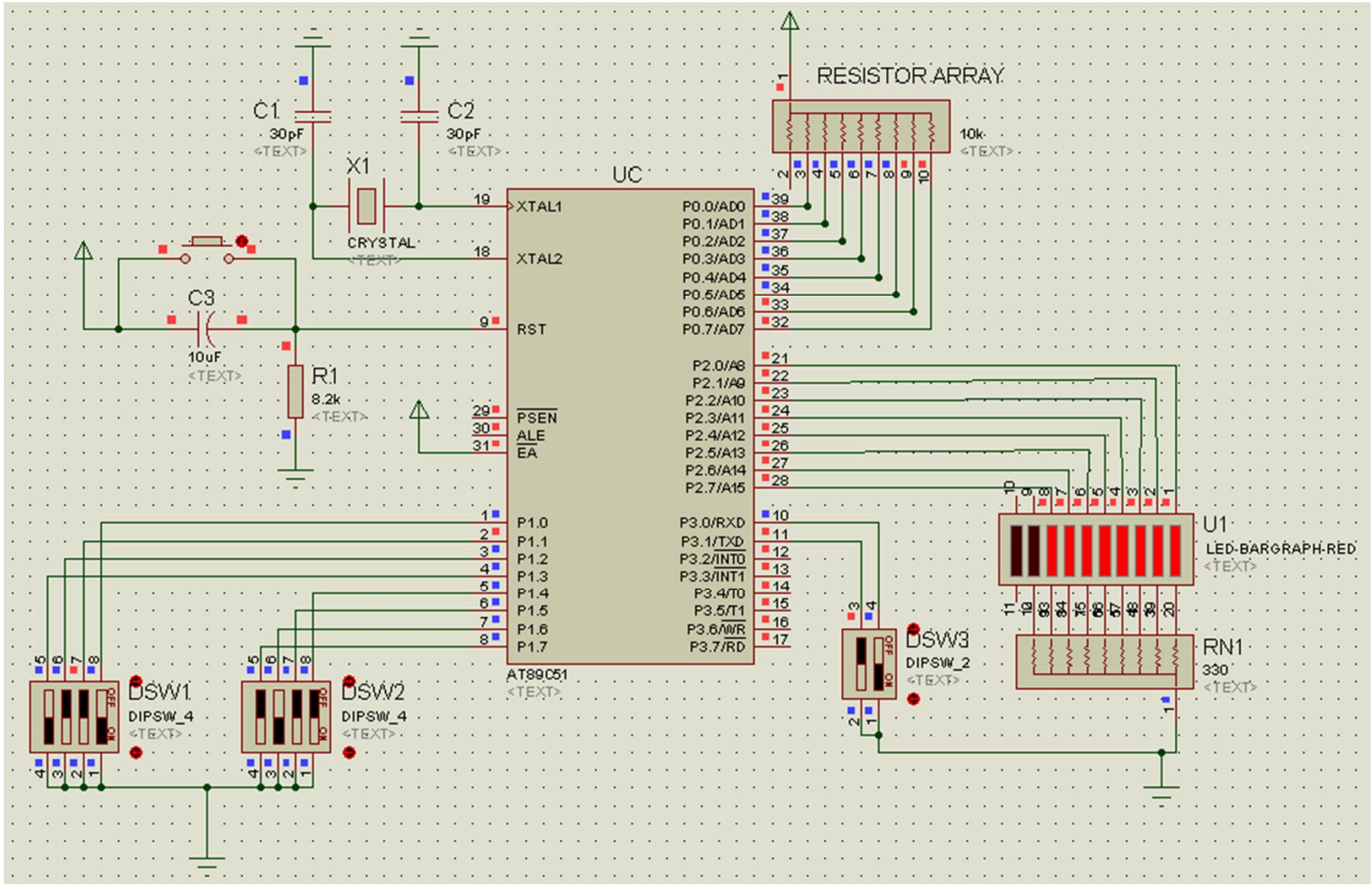
```
ORG 0
MOV P1,#0FFH; MAKE P1 an input port
CLR A
MOV P2,00H; clear P2 first, it is result
SETB P3.1
SETB P3.0

;_____The operation will start from here_____
START:
;---Some necessary manipulations
MOV A,P1
ANL A,#0FH
MOV B,A
MOV A,P1
RR A
RR A
RR A
RR A
ANL A,#0FH
CLR C
;--Start of if else
```

```
;--Start of if else
JB P3.1,OP1X
JB P3.0,OP01
ADD A,B     ;____OP=00
MOV P2,A
SJMP START

OP01:
SUBB A,B    ;____OP=01
MOV P2,A
SJMP START
;*****************************************
;_____________If the BIT1 is set, come here
;*****************************************
OP1X:
JB P3.0,OP11
MUL AB      ;____OP=10
MOV P2,A
SJMP START
OP11:
DIV AB      ;____OP=11
MOV P2,A
SJMP START
END
```

Todays Task 1



Todays Task 2

- Implement the design on 8051 Kit.
- This time display the result on Seven Segment.
- You will need to change the output port from Port2 to Port0.
- Modify the code in such a way that once the code is inside the ‘Seven segment display’ Loop, in order to recalculate the result, the user will press push button P3.7 and the Microcontroller will recalculate the result.
- Hint:
- Use Look-up Table for Seven Segment Calculation.
- You will need 2 Seven Segments to display the complete 8-bits of the result in hex.

Task 2 Code

```
1  ORG 0
2  MOV P2,#0FFH; MAKE P2 an input port
3  CLR A
4  MOV P0,00H; clear P0 first, it is 7 Segment Display
5  SETB P3.1
6  SETB P3.0
7  MOV DPTR,#300H
8  ;_____The operation will start from here_____
9
10 START:
11 ;---Some necessary manipulations
12 MOV A,P2
13 ANL A,#0FH
14 MOV B,A
15 MOV A,P2
16 RR A
17 RR A
18 RR A
19 RR A
20 ANL A,#0FH
21 CLR C
22 ;--Start of if else
23 JB P3.1,OP1X
24 JB P3.0,OP01
25
26 ADD A,B     ;____OP=00
27 MOV R7,A
28 SJMP DISPLAY
29
30 OP01:
31 SUBB A,B    ;____OP=01
32 MOV R7,A
```

```
32 MOV R7,A
33 SJMP DISPLAY
34 ;*****************************************
35 ;_____________If the BIT1 is set, come here
36 ;*****************************************
37 OP1X:
38 JB P3.0,OP11
39 MUL AB      ;____OP=10
40 MOV R7,A
41 SJMP DISPLAY
42
43 OP11:
44 DIV AB      ;____OP=11
45 MOV R7,A
46 SJMP DISPLAY
47
48 ;------------Seven Segment Display Code--------------;
49 DISPLAY:
50 MOV A,R7
51 MOV P1,#01H
52 ANL A,#0FH
53 MOVC A,@A+DPTR
54 MOV P0,A
55
56 ACALL Delay
57
58 MOV P1,#02H
59 MOV A,R7
60 ANL A,#0F0H
61 SWAP A
62 MOVC A,@A+DPTR
63 MOV P0,A
```

Task 2 Code

```
63 MOV P0,A
64
65 ACALL Delay
66
67 JNB P3.7, START
68 SJMP DISPLAY
69
70
71
72 ;------------------Delay SubRoutine------------------;
73 Delay:
74 MOV R1,#255
75 LOOP:DJNZ R1,LOOP
76 RET
77
78
79
80 ;----------ROM LOOK-UP Table for Seven Segment-------;
81 ORG 300H
82 Mydata:
83 DB 0C0H , 0F9H , 0A4H , 0B0H , 99H , 92H , 82H , 0F8H , 80H , 90H , 0A0H , 83H , 0A7H , 0A1H , 84H , 8EH
84 END
```

Task 2 Alternative Code

```
1  ORG 0
2  MOV P2,#0FFH; MAKE P2 an input port
3  CLR A
4  MOV P0,#00H; clear P0 first, it is 7 Segment Display
5  MOV P3,#0FFH; Make P3 an input port
6  OP EQU 30H
7
8  MOV DPTR,#300H
9  ;_____The operation will start from here______
10
11 START:
12 ;---Some necessary manipulations
13 MOV OP,P3
14 ANL OP,#03
15 MOV R4,OP
16 MOV A,P2
17 ANL A,#0FH
18 MOV B,A
19 MOV A,P2
20 RR A
21 RR A
22 RR A
23 RR A
24 ANL A,#0FH
25 CLR C
26 ;--Start of if else
27 CJNE R4,#00H,ELSE1
28 ADD A,B     ;____OP=00
29 MOV R7,A
30 SJMP DISPLAY
31
32 ELSE1:
```

```
32 ELSE1:
33 CJNE R4,#01H,ELSE2
34 SUBB A,B     ;____OP=01
35 MOV R7,A
36 SJMP DISPLAY
37
38 ELSE2:
39 CJNE R4,#02H,ELSE3
40 MUL AB       ;____OP=10
41 MOV R7,A
42 SJMP DISPLAY
43
44 ELSE3:
45 CJNE R4,#03H,START
46 DIV AB      ;____OP=11
47 MOV R7,A
48 SJMP DISPLAY
```

The DISPLAY code is the same as was in the last case

Assignment

- Connect the circuit according to the given scheme on breadboard and run a toggle program on BAR LED.