

1)

C:\Users\alper\Desktop\EE\EE4-1\EE 447\Lab2\Q3\q1.s

```

1          AREA          main, READONLY, CODE
2          THUMB
3          EXTERN        OutChar
4          EXPORT        DELAY150
5
6  DELAY150  PROC;
7          MOV32         R8,#6000000      ;Since each loop takes 4 cycles. # of iteration should be
8          2.400.000/4 = 600000.
9          ;Since its clock is 16MHz. Each cycle will take 0.06us (1/16M sec) to operate
10         ;To take 150ms (0.15 sec = 1.5*10^5 us),
11         ;It should take (0.15sec)/((1/16M)sec) =2.4M cycle
12         delaying      NOP              ;Taking 1 cycle
13                     SUBS             R8,#1      ;Taking 1 cycle
14                     BNE             delaying    ;Taking 2 cycle
15                     BX              LR
16         ALIGN
17         ENDP
18         END

```

2)

C:\Users\alper\Desktop\EE\EE4-1\EE 447\Lab2\Q2\q2.s

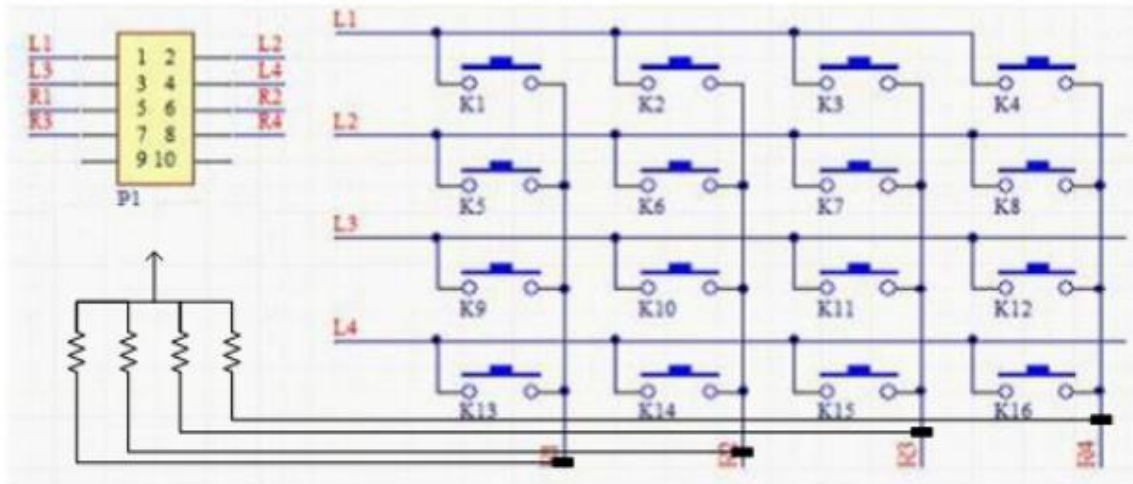
```

1          AREA          main, READONLY, CODE
2          THUMB
3          EXTERN        DELAY150
4          EXTERN        PB_INIT
5          EXPORT        __main
6
7
8  PB_INP    EQU 0x4000503C
9  PB_OUT    EQU 0x400053C0
10 GPIO_PORTB_DATA EQU 0x400053FC ; data address to all pins
11 GPIO_PORTB_DIR EQU 0x40005400
12 GPIO_PORTB_AFSEL EQU 0x40005420
13 GPIO_PORTB_DEN EQU 0x4000551C
14 GPIO_PORTB_PDR EQU 0x40005514
15 GPIO_PORTB_PUR EQU 0x40005510
16 SYSCTL_RCGCGPIO EQU 0x400FE608 ; these are written in Week-6 Lecture Notes page 58
17
18 __main     PROC;
19         BL          PB_INIT
20         nanInp      LDR          R1,=GPIO_PORTB_DATA
21                     LDR          R0,[R1]
22                     LSR          R5,R0,#4
23                     LDR          R0,[R1]
24                     CMP          R5,#0xE
25                     BEQ          LED1
26                     CMP          R5,#0xD
27                     BEQ          LED2
28                     CMP          R5,#0xB
29                     BEQ          LED3
30                     CMP          R5,#0x7
31                     BEQ          LED4
32                     BL          DELAY150
33                     B            nanInp
34         LED1        MOV          R2,R5 ;E
35                     LDR          R1,=GPIO_PORTB_DATA
36                     STR          R2,[R1]
37                     B            nanInp
38
39         LED2        MOV          R2,R5 ;D
40                     LDR          R1,=GPIO_PORTB_DATA
41                     STR          R2,[R1]
42                     B            nanInp
43
44         LED3        MOV          R2,R5
45                     LDR          R1,=GPIO_PORTB_DATA
46                     STR          R2,[R1]
47                     B            nanInp
48
49         LED4        MOV          R2,R5
50                     LDR          R1,=GPIO_PORTB_DATA
51                     STR          R2,[R1]
52                     B            nanInp
53
54         ALIGN
55         ENDP
56

```

3)

- a) When there is a push in any keys, the output data will also change. We used PB 0-3 as output and PB 4-7 as inputs to be able to do that we connected R 1-4 to PB 0-3 and L 1-4 to PB 4-7. We checked if the input pins are changed, and if they don't, it means there are no pushed buttons.



- b) If there is a button pressed, it will go to COLM1, COLM2, COLM3, and COLM4. After that input will be read again. If the input is changed that means the key is released.
- c) We shifted right the input and determined which COLM it belongs to. After that by compering the values of output we determined the ROW of the key.
- d) If we don't eliminate the bouncing effect, the results will be not acceptable. Results will be undetermined if we don't add any delay. To overcome this situation, we add a delay and after reading it will read again and checks if there is a change in data and continue to do the rest of the code.
- e)

f)

\\Users\alpen\Desktop\EE\EE4-1\EE 447\Lab2\Q3\Q3.s

```

1      AREA          main, READONLY, CODE
2      THUMB
3      EXTERN        DELAY150
4      EXTERN        PB_INIT
5      EXTERN        OutChar
6      EXPORT        _main
7      WRITE         EQU 0x20000400
8      GPIO_PORTB_DATA EQU 0x400054FC ; data address to all pins
9      GPIO_PORTB_DIR  EQU 0x40005400
10     GPIO_PORTB_AFSEL EQU 0x40005420
11     GPIO_PORTB_DEN   EQU 0x4000551C
12     GPIO_PORTB_PDR   EQU 0x40005514
13     GPIO_PORTB_PUR   EQU 0x40005510
14     SYSCTL_RCGCGPIO EQU 0x400FE608 ; these are written in Week-6
15
16     _main          PROC
17     BL              PB_INIT ;Port B init
18     start          LDR      R2,GPIO_PORTB_DATA
19                     MOV      R3,#0xF0
20                     MOV      R0,#48 ;Define R0 as 0
21                     MOV      R7,#0
22                     STR      R3,[R2]
23                     LDR      R3,[R2] ;Debouncing Effect
24                     BL        DELAY150
25                     LDR      R4,[R2]
26                     CMP      R3,R4 ;If there is no debouncing , continue
27                     BNE      start
28                     LSR      R5,R3,#4 ;in this part of the code we search row and
;column in only one loop
29                     CMP      R5,#0xD ;to determine the number of button we use column
;number
30                     ADDEQ     R0,#1 ;if pressed button is in R1 we add to R2 1
31                     CMP      R5,#0xB ;if pressed button is in R1 we add to R3 2
32                     ADDEQ     R0,#2
33                     CMP      R5,#0x7 ;if pressed button is in R1 we add to R4 3
34                     ADDEQ     R0,#3
35                     CMP      R5,#0xF ;Determine button is being pressed
36                     BNE      ROW_finder ;If there is button pressed continue
37                     B        start
38
39
40     ROW_finder      MOV      R6,#0x7 ;ROW1 Which means L4
41                     STR      R6,[R2]
42                     NOP
43                     NOP
44                     NOP
45                     LDR      R7,[R2]
46                     LSR      R7,R7,#4 ;Output is taken
47                     CMP      R7,R5 ;If output is same with R5 we can assure it is
;true
48                     ADDEQ     R0,#12 ;Since we are on L4, we should add 12 to the R0
49                     MOV      R6,#0xB; ;Same process continues
50                     STR      R6,[R2]
51                     NOP
52                     NOP
53                     NOP
54                     LDR      R7,[R2]
55                     LSR      R7,R7,#4
56                     CMP      R7,R5
57                     ADDEQ     R0,#8
58                     MOV      R6,#0xD
59                     STR      R6,[R2]
60                     NOP
61                     NOP
62                     NOP
63                     LDR      R7,[R2]
64                     LSR      R7,R7,#4
65                     CMP      R7,R5
66                     ADDEQ     R0,#4
67                     MOV      R6,#0xE
68                     STR      R6,[R2]
69                     NOP
70                     NOP
71                     NOP
72                     LDR      R7,[R2]
73                     LSR      R7,R7,#4
74                     CMP      R7,R5

```

C:\Users\alper\Desktop\EE\EE4-1\EE 447\Lab2\Q3\Q3.s

```
75      ADDEQ      R0,#0
76      LDR        R8,=0xF0
77      STR        R8,[R2]
78      NOP
79      NOP
80      NOP
81  out      LDR        R9,[R2]          ;This function determines the if button is keep
      being pressed or not
82      NOP
83      NOP
84      NOP
85      CMP        R9,R8
86      BNE        out
87      CMP        R0,#58              ;We have done lots of addition but we did not
      take care of letters. If R0 is bigger than 9+48, we should add 7 to get letters
88      BCC        noletter
89      ADD        R0,#7
90  noletter  BL        OutChar
91      B          start
92      ALIGN
93      ENDP
94
```