

EE447 EXPERIMENT 2

PRELIMINARY REPORT

Q1-) DELAY.s

```

4          AREA      subroutine , CODE, READONLY
5          THUMB
6          EXPORT    DELAY
7  DELAY    PROC
8  GoBack   SUBS      R0, R0, #1      ;subtract 1 to R0
9          BEQ      End_Delay      ;if r0 = 0, go end
10         B        GoBack          ;
11 End_Delay BX       LR            ;end
12         ALIGN
13         ENDP
14         ENM

```

Q2-) __main.s

```

4  OFFSET      EQU 0x10
5  FIRST       EQU 0x20000480
6  STORE       EQU 0x20000410
7  GPIO_PORTB_DATA EQU 0x400053FC
8  GPIO_PORTB_DIR EQU 0x40005400
9  GPIO_PORTB_AFSEL EQU 0x40005420
10 GPIO_PORTB_DEN EQU 0x4000551C
11 GPIO_PORTB_PUR EQU 0x40005510
12 GPIO_PORTB_PDR EQU 0x40005514
13 IOB          EQU 0xF0
14 SYSCCTL_RCGCGPIO EQU 0x400FE608
15          AREA      main, READONLY, CODE
16          THUMB
17          EXTERN     DELAY;
18          EXPORT     __main
19  __main
20 Start      LDR      R1, =SYSCCTL_RCGCGPIO
21          LDR      R0, [R1]
22          ORR      R0, R0, #0x2      ;Port B clock is enabled R0 = 00000010
23          STR      R0, [R1]
24          NOP
25          NOP
26          NOP
27
28          LDR      R1, =GPIO_PORTB_DIR ;direction of port B pins
29          LDR      R0, [R1]
30          BIC      R0, #0xFF
31          ORR      R0, #IOB          ;r0 = 11110000 ;B7-B4 is output ;B3-B0 is input
32          STR      R0, [R1]
33          LDR      R1, =GPIO_PORTB_AFSEL ;for afsel disable
34          LDR      R0, [R1]
35          BIC      R0, #0xFF
36          STR      R0, [R1]
37          LDR      R1, =GPIO_PORTB_DEN ;all pins are digital
38          LDR      R0, [R1]
39          ORR      R0, #0xFF
40          STR      R0, [R1]
41          LDR      R1, =GPIO_PORTB_PUR ;pull up resistor settle
42          LDR      R0, [R1]
43          ORR      R0, #0x00
44          STR      R0, [R1]
45
46 Begin      LDR      R1, =GPIO_PORTB_DATA ;Data address in R1
47          MOV      R0, #0xFF
48          STR      R0, [R1]          ;All outputs OFF

```

```

50 InputCheck
51     LDR    R1,=GPIO_PORTB_DATA
52     LDR    R0, [R1]
53     AND    R2, R0, #0x0F           ;and r0 with F, to find out which button is pressed
54     CMP    R2,#0x0F               ;if buttons are not pressed, go to beginning of loop
55     BEQ    InputCheck
56     MOV32  R0,#160000             ;wait 100msec delay
57     BL     DELAY                  ;prevent from bouncing, wait some sec
58     LDR    R1,=GPIO_PORTB_DATA
59     LDR    R0, [R1]
60     AND    R4, R0, #0x0F           ;and r0 with F, to find out which button is pressed
61     CMP    R2, R4                 ; IF they are equal set the output
62     BNE    InputCheck
63                                     ; If an input is read
64     LDR    R1,=GPIO_PORTB_DATA     ; Data address in R1
65     LSL    R4, #4                  ;shift 4 times for output
66     STR    R4,[R1]                 ;Corresponding Outputs set high
67     MOV32  R0,#16000000           ;wait 5Sec
68     BL     DELAY
69     B      Begin
70
71     ALIGN
72     END

```

Q3-) a-) we have 4 bits input. If we do not press any button, we give all input value is 1 since we enable the pull up resistor. If we press any button, it gives us 0.

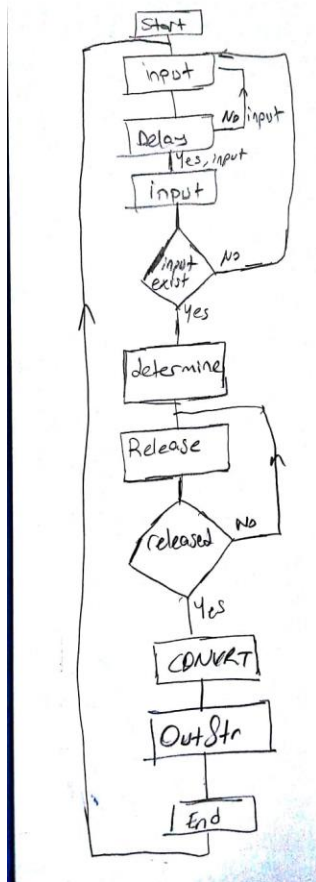
b-) we can compare F after button is pressed. Then, if compare result is 0, we understand button is released.

c-) We give 0 to first row and others is 1. Also, columns are 1 since we enable pull up resistor. Then, if we observe 0 any column, we understand that switch is pressed. For example, we give E for output. That means first column is zero. Then, if we take E from input, first row first column is pressed (K1).

If we take D, K2 is pressed.

d-) We cannot understand exactly which button is pressed. If we add delay, we can avoid bouncing effects.

e-)



f-)_main.s

```

3  OFFSET      EQU      0x10
4  FIRST       EQU      0x20000480
5  STORE       EQU      0x20000400
6  GPIO_PORTB_DATA EQU 0x400053FC
7  GPIO_PORTB_DIR EQU 0x40005400
8  GPIO_PORTB_AFSEL EQU 0x40005420
9  GPIO_PORTB_DEN EQU 0x4000551C
10 GPIO_PORTB_PUR EQU 0x40005510
11 GPIO_PORTB_PDR EQU 0x40005514
12 IOB         EQU      0xF0
13 SYSCCTL_RCGCGPIO EQU 0x400FE608
14
15              AREA      main, READONLY, CODE
16              THUMB
17              EXTERN    OutStr ; Reference external subroutine
18              EXTERN    DELAY;
19              EXTERN    CONVRT
20              EXPORT    __main
21
22 __main
23 Start        LDR R1, =SYSCCTL_RCGCGPIO
24              LDR R0, [ R1 ]
25              ORR R0, R0, #0x2;Port B clock enabled
26              STR R0, [ R1 ]
27              NOP      ;Wait for clock to stabilize
28              NOP
29              LDR R1, =GPIO_PORTB_DIR ;
30              LDR R0, [ R1 ]
31              BIC R0, #0xFF
32              ORR R0, #IOB
33              STR R0, [ R1 ]

```

```

34      LDR R1 , =GPIO_PORTB_AFSEL
35      LDR R0 , [ R1 ]
36      BIC R0 , #0xFF
37      STR R0 , [ R1 ]
38      LDR R1 , =GPIO_PORTB_DEN
39      LDR R0 , [ R1 ]
40      ORR R0 , #0xFF
41      STR R0 , [ R1 ]
42      LDR R1 , =GPIO_PORTB_PUR
43      LDR R0 , [ R1 ]
44      ORR R0 , #0x0F
45      STR R0 , [ R1 ]
46
47  Begin      LDR R1,=GPIO_PORTB_DATA ;Data address in R1
48            MOV R0,#0x00
49            STR R0,[R1]           ;All outputs GND
50            MOV R2,#0;           R2 is the switch ID
51  LOOP
52            LDR R1,=GPIO_PORTB_DATA; Data address in R1
53            MOV32 R5,#0xE0
54            STR R5,[R1];         All outputs GND
55            LDR R1,=GPIO_PORTB_DATA
56            LDR R2, [R1]
57            AND R2, #0xF         ;AND r2 with f, to determine pressed button
58            CMP R2, #0xF
59            BEQ LOOP2           ;if button is not pressed, go to loop2
60            MOV32 R0, #160000
61            BL DELAY            ;wait 100msec
62            LDR R1,=GPIO_PORTB_DATA
63            LDR R3, [R1]
64            AND R3, #0xF         ;AND r3 with f, to determine pressed button
65            CMP R2, R3          ;if they are equal, button is pressed
66
67            BNE LOOP
68            ADD R5, R5, R2
69            CMP R5, #0xEE        ;for k1
70            MOVEQ R4, #0
71            CMP R5, #0xED        ;for k2
72            MOVEQ R4, #1
73            CMP R5, #0xEB        ;for k3
74            MOVEQ R4, #2
75            CMP R5, #0xE7        ;for k4
76            MOVEQ R4, #3
77            B RELEASE            ;go release func.
78  LOOP2
79            LDR R1,=GPIO_PORTB_DATA; Data address in R1
80            MOV32 R5,#0xD0
81            STR R5,[R1];         All outputs GND
82            LDR R1,=GPIO_PORTB_DATA
83            LDR R2, [R1]
84            AND R2, #0xF
85            CMP R2, #0xF
86            BEQ LOOP3
87            MOV32 R0, #160000
88            BL DELAY
89            LDR R1,=GPIO_PORTB_DATA
90            LDR R3, [R1]
91            AND R3, #0xF
92            CMP R2, R3
93            BNE LOOP2
94            ADD R5, R5, R2
95            CMP R5, #0xDE        ;for k5
96            MOVEQ R4, #4
97            CMP R5, #0xDD        ;for k6
98            MOVEQ R4, #5

```

```

98      CMP R5, #0xDB    ;for k7
99      MOVEQ R4, #6
100     CMP R5, #0xD7    ;for k8
101     MOVEQ R4, #7
102     B    RELEASE
103 LOOP3
104     LDR R1,=GPIO_PORTB_DATA; Data address in R1
105     MOV32 R5,#0xB0
106     STR R5,[R1];    All outputs GND
107     LDR R1,=GPIO_PORTB_DATA
108     LDR R2, [R1]
109     AND R2, #0xF
110     CMP R2, #0xF
111     BEQ LOOP4
112     MOV32 R0, #160000
113     BL  DELAY
114     LDR R1,=GPIO_PORTB_DATA
115     LDR R3, [R1]
116     AND R3, #0xF
117     CMP R2, R3
118     BNE LOOP3
119     ADD R5, R5, R2
120     CMP R5, #0xBE    ;for k9
121     MOVEQ R4, #8
122     CMP R5, #0xBD    ;for k10
123     MOVEQ R4, #9
124     CMP R5, #0xBB    ;for k11
125     MOVEQ R4, #10
126     CMP R5, #0xB7    ;for k12
127     MOVEQ R4, #11
128     B    RELEASE
129
130 LOOP4
131     LDR R1,=GPIO_PORTB_DATA; Data address in R1
132     MOV32 R5,#0x70
133     STR R5,[R1];    All outputs GND
134     LDR R1,=GPIO_PORTB_DATA
135     LDR R2, [R1]
136     AND R2, #0xF
137     CMP R2, #0xF
138     BEQ LOOP
139     MOV32 R0, #160000
140     BL  DELAY
141     LDR R1,=GPIO_PORTB_DATA
142     LDR R3, [R1]
143     AND R3, #0xF
144     CMP R2, R3
145     BNE LOOP4
146     ADD R5, R5, R2
147     CMP R5, #0x7E    ;for k13
148     MOVEQ R4, #12
149     CMP R5, #0x7D    ;for k14
150     MOVEQ R4, #13
151     CMP R5, #0x7B    ;for k15
152     MOVEQ R4, #14
153     CMP R5, #0x77    ;for k16
154     MOVEQ R4, #15
155     B    RELEASE

```

```
157 RELEASE
158     LDR R1,=GPIO_PORTB_DATA
159     LDR R2, [R1]      ;again take input
160     AND R2, #0xF      ;if button is released
161     CMP R2, #0xF      ;go to output
162     BEQ OUTPUT        ;if not go release and wait
163     B   RELEASE
164 OUTPUT
165
166     LDR R5, =STORE    ;r4 is button number
167     BL  CONVERT       ;convert number to ascii
168     LDR R5, =STORE
169     BL  OutStr        ;write number to termite
170     ALIGN
171     END
```