Zeynepnur ŞAHİNEL
2305399-Wed, 12:40 26.12.21

EE447 EXPERIMENT 2 PRELIMINARY WORK

Part 1

```
;LABEL          DIRECTIVE     VALUECOMMENT

                                        AREA routines, CODE, READONLY
                                        THUMB
                                        EXPORT DELAY100

HUNDREDMSEC     EQU 240963 ; for 100ms delay at  16 MHz clock,

;in main part->LDR R0, =HUNDREDMSEC

DELAY100

AGAIN                   NOP
                        NOP
                        SUBS R0,R0,#1
                        BNE  AGAIN
                        BX LR

                        ALIGN
                        END
```

Figure 1. Delay100 Code

Here, lecture note is taken as reference. (CH5-delay example) Instructions takes 0.083 usec, whereas branch instractions takes 3*0.083 sec. Thus to find R0 below calculation can be hold:

$$R0=(100\ msec)/[(3+1+1+1)*0.083usec)=240963$$

Part2

```
PB_IN                           EQU             0X4000503C     ; data adress to all input pins
PB_OUT                                  EQU             0X400053C0     ; data adress to all input
pins

GPIO_PORTB_DIR_R  EQU 0x40005400
GPIO_PORTB_AFSEL_R      EQU 0x40005420
GPIO_PORTB_PUR_R        EQU 0x40005510

GPIO_PORTB_DEN_R        EQU 0x4000551C
GPIO_PORTB_LOCK_R       EQU 0x40005520
GPIO_PORTB_CR_R   EQU 0x40005524

SYSCTL_RCGCGPIO_R       EQU 0x400FE608

                                        AREA |.text|, CODE, READONLY, ALIGN=2
                                        THUMB
```

```
                                EXTERN OutChar
                                EXPORT __main

__main

        BL PortB_Init
loop

        LDR R0, =HUNDREDMSEC
        BL DELAY


        LDR R1,=PB_IN           ;r1=0x4000503c
        LDR R2, [R1]        ;read pb0-pb3
        LDR R1,=PB_OUT
        LDR R3, [R1]        ;read pb4-pb7

        CMP R2, #0x0F ;all switches are off
        BEQ loop

        LSL R2, #4              ;to compare pb7-4 with related switch
        EOR R3, R3, R2          ;toggling acc to switch values
        EOR R3, R3,#0xF0    ;Since switches are 0 when they are closed, and 1 when
open
        ;however toggling happens when they are 1(open).
    ;therefore we need to EOR here.

        LDR R0, =HUNDREDMSEC
    BL DELAY

        LDR R1, =PB_OUT
        STR R3, [R1] ; write to PB4-7

        B loop

;**************DELAY***************
HUNDREDMSEC     EQU 240963 ; 100ms delay at ~16 MHz clock,

DELAY
    NOP
    NOP
    NOP
    SUBS R0, R0, #1 ; R0 = R0 - 1
    BNE DELAY
    BX LR

PortB_Init
    LDR R1, =SYSCTL_RCGCGPIO_R ; 1) activate clock for Port B
    LDR R0, [R1]
    ORR R0, R0, #0x02 ; set bit 5 to turn on clock
    STR R0, [R1]
    NOP
    NOP
    NOP ; allow time for clock to finish
```

```
LDR R1, =GPIO_PORTB_LOCK_R ; 2) unlock the lock register
LDR R0, =0x4C4F434B
STR R0, [R1]
LDR R1, =GPIO_PORTB_CR_R ; enable commit for Port B
MOV R0, #0xFF
STR R0, [R1]

; set direction register
LDR R1, =GPIO_PORTB_DIR_R
MOV R0,#0xF0 ; PB 0-3 input, PB 4-7 output
STR R0, [R1]

LDR R1, =GPIO_PORTB_AFSEL_R
MOV R0, #0 ; 0: disable alternate function
STR R0, [R1]

LDR R1, =GPIO_PORTB_PUR_R ; pull-up resistors for PB0-3
MOV R0, #0x0F ; enable weak pull-up
STR R0, [R1]

LDR R1, =GPIO_PORTB_DEN_R ; 7) enable Port B digital port
MOV R0, #0xFF ; 1: enable digital I/O
STR R0, [R1]

BX LR

ALIGN
END
```

Figure 2. Toggling Led Code In Terms Of Switch Modes

The below link shows the demonstration.
part2 demonstartaion

Part 3

a)How can you detect whether any key is pressed?

When a key is pressed, input becomes zero; since connection between rows and columns are provided, leading short circuit.

b)How can you detect whether a pressed key is released?

Similar to part a, when a key is released; open circuit occurs. Due to open circuit case, input becomes 1 (high). Also it should be noted that input ports are pulled up.

c)Assuming that you have detected that a key is pressed. Explain your algorithm to determine which one is pressed.

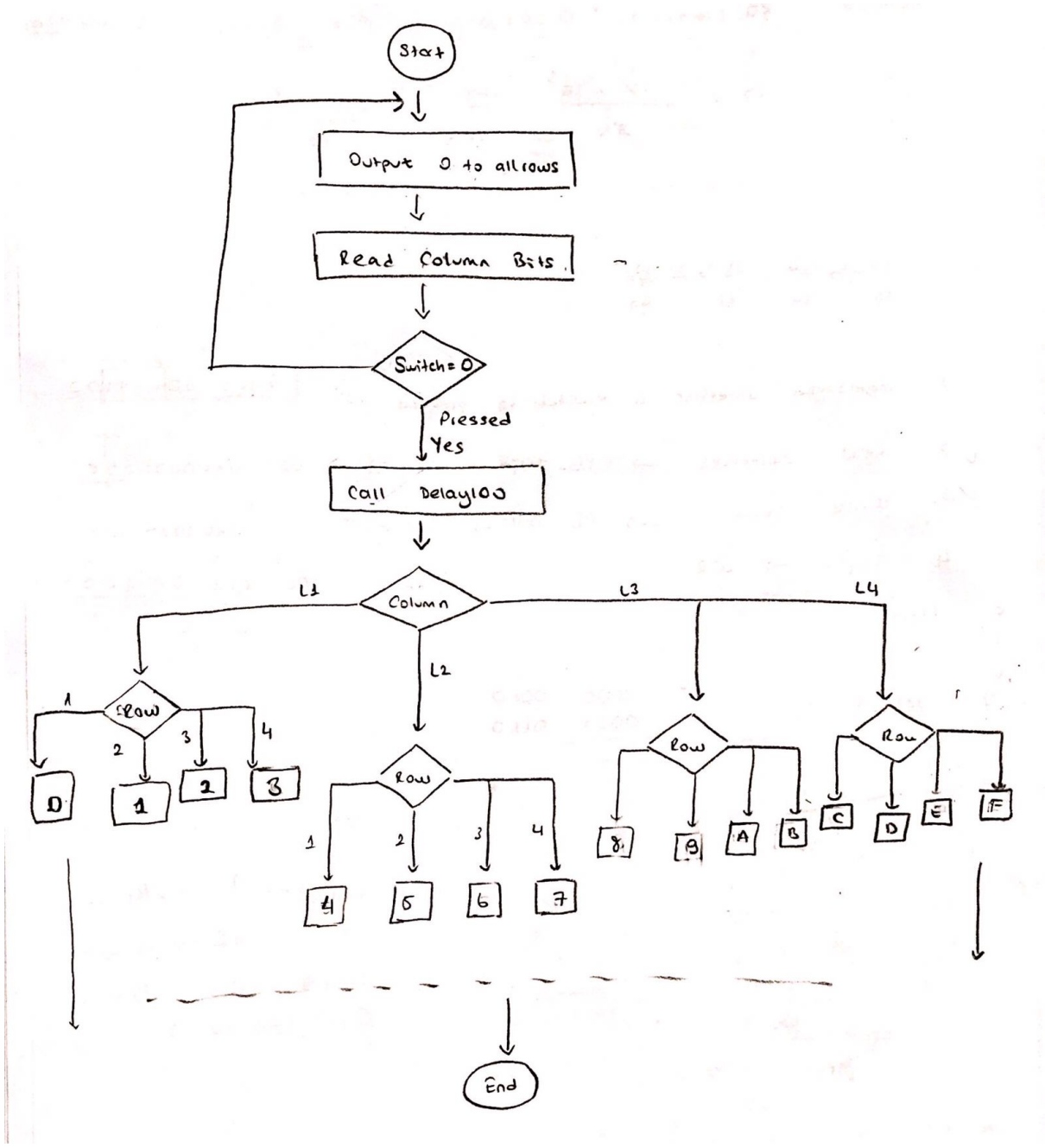
In a keypad, one can determine the input and output ports. For this case, Lines are assumed as Input Ports whereas Rows are determined as Output Ports. To understant the which key is pressed, all rows should be scanned.

Row1: 1110 (E) If Input: 0x0E, L1 pressed

	If Input: 0x0D, L2 pressed
	If Input: 0x0B, L3 pressed
	If Input: 0x07, L4 pressed
Row2: 1101 (D)	If Input: 0x0E, L1 pressed
	If Input: 0x0D, L2 pressed
	If Input: 0x0B, L3 pressed
	If Input: 0x07, L4 pressed
Row3: 1011 (B)	If Input: 0x0E, L1 pressed
	If Input: 0x0D, L2 pressed
	If Input: 0x0B, L3 pressed
	If Input: 0x07, L4 pressed
Row4: 0111 (7)	If Input: 0x0E, L1 pressed
	If Input: 0x0D, L2 pressed
	If Input: 0x0B, L3 pressed
	If Input: 0x07, L4 pressed

d)Discuss what can happen due to bouncing. How can you avoid bouncing effects?
Due to bouncing one can print more than one numbers on Termite. To avoid bouncing Delay subroutine should be used.

e)Now, develop your overall end-to-end algorithm that outputs ID of the pressed key to the terminal window and draw its flow chart.



f)

```
;LABEL DIRECTIVE VALUE COMMENT
PB_IN                   EQU    0X4000503C    ; data adress to all input pins
PB_OUT                  EQU    0X400053C0    ; data adress to all input pins

GPIO_PORTB_DIR_R        EQU 0x40005400
GPIO_PORTB_AFSEL_R      EQU 0x40005420
GPIO_PORTB_PUR_R        EQU 0x40005510

GPIO_PORTB_DEN_R        EQU 0x4000551C
GPIO_PORTB_LOCK_R       EQU 0x40005520
GPIO_PORTB_CR_R   EQU 0x40005524

SYSCTL_RCGCGPIO_R       EQU 0x400FE608
```

```
                AREA |.text|, CODE, READONLY, ALIGN=2
                THUMB
                EXTERN OutChar

                EXPORT __main
__main

        BL      PortB_Init

loop
        MOV R2, #0
        BL Write_output
        BL Read_Input
        CMP R2, #0x0F ; switches are off (1111)
        BEQ loop

        LDR R0, =MSEC100
        BL DELAY100

;_____________row 1_____________
        LDR R2, =0xE0 ;Row 1 1110 0000
        BL Write_output
        BL Read_Input

        CMP R2, #0x0E  ;Line1 is on
        MOVEQ R5,#0x30
        BEQ final

        CMP R2, #0x0D  ;Line2 is on
        MOVEQ R5,#0x31
        BEQ final

        CMP R2, #0x0B   ;Line3 is on
        MOVEQ R5,#0x32
        BEQ final

        CMP R2, #0x07   ;Line4 is on
        MOVEQ R5,#0x33
        BEQ final

;_____________row 2_____________

        MOV R2, #0xD0
        BL Write_output
        BL Read_Input

        CMP R2, #0x0E
        MOVEQ R5,#0x34
        BEQ final

        CMP R2, #0x0D
        MOVEQ R5,#0x35
        BEQ final
```

```
                CMP R2, #0x0B
                MOVEQ R5,#0x36
                BEQ final

                CMP R2, #0x07
                MOVEQ R5,#0x37
                BEQ final

;_____________________row 3_____________
                MOV R2, #0xB0
                BL Write_output
                BL Read_Input

                CMP R2, #0x0E
                MOVEQ R5,#0x38
                BEQ final

                CMP R2, #0x0D
                MOVEQ R5,#0x39
                BEQ final

                CMP R2, #0x0B
                MOVEQ R5,#0x41
                BEQ final

                CMP R2, #0x07
                MOVEQ R5,#0x42
                BEQ final

;;_____________________row 4_____________

                MOV R2, #0x70
                BL Write_output
                BL Read_Input

                CMP R2, #0x0E
                MOVEQ R5,#0x43
                BEQ final

                CMP R2, #0x0D
                MOVEQ R5,#0x44
                BEQ final

                CMP R2, #0x0B
                MOVEQ R5,#0x45
                BEQ final

                CMP R2, #0x07
                MOVEQ R5,#0x46
                BEQ final

                B loop
```

```
final
                BL Read_Input
                CMP R2, #0x0F   ;for stabilization of the keys
                BNE final

                BL OutChar

                B loop

;********DELAY100*********************
MSEC100 EQU 240963

DELAY100
        NOP
        NOP
        SUBS R0, R0, #1
        BNE DELAY100
        BX LR


PortB_Init
        LDR R1, =SYSCTL_RCGCGPIO_R ; 1) activate clock for Port B
        LDR R0, [R1]
        ORR R0, R0, #0x02 ; set bit 5 to turn on clock
        STR R0, [R1]
        NOP
        NOP
        NOP ; allow time for clock to finish

        LDR R1, =GPIO_PORTB_LOCK_R ; 2) unlock the lock register
        LDR R0, =0x4C4F434B
        STR R0, [R1]
        LDR R1, =GPIO_PORTB_CR_R ; enable commit for Port B
        MOV R0, #0xFF
        STR R0, [R1]

        ; set direction register
        LDR R1, =GPIO_PORTB_DIR_R
        MOV R0,#0xF0 ; PB 0-3 input, PB 4-7 final
        STR R0, [R1]

        LDR R1, =GPIO_PORTB_AFSEL_R
        MOV R0, #0 ; 0: disable alternate function
        STR R0, [R1]

        LDR R1, =GPIO_PORTB_PUR_R ; pull-up resistors for PB0-3
        MOV R0, #0x0F ; enable weak pull-up
        STR R0, [R1]

        LDR R1, =GPIO_PORTB_DEN_R ; 7) enable Port B digital port
        MOV R0, #0xFF ; 1: enable digital I/O
        STR R0, [R1]

        BX LR
```

```
Read_Input
        LDR R1, =PB_IN
        LDR R2, [R1] ; read pb0-pb3
 BX LR

Write_output
        LDR R1,=PB_OUT
        STR R2, [R1]            ;write pb4-pb7
 BX LR

 ALIGN
 END
```

Below link shows the demonstaration:
Part3 demonstration