

Preliminary work

EE 447: Lab #5

Analog to Digital converter

Berkay İPEK

2304814 – Sec.2

Question 1)

In all questions I connected 3.3V to left pin of POT, GND to right pin of POT, PE3 to middle pin of POT.

- *Main code is:*

```
1 ;LABEL      DIRECTIVE  VALUE      COMMENT
2             AREA      main, READONLY, CODE
3             THUMB
4             EXTERN    Init
5             EXTERN    calc
6             EXPORT    __main      ; Make available
7
8 __main      PROC
9             BL        Init        ; Init gpio and adc0
10            BL        calc        ; Calculate and store
11            B         .           ; Dummy code since program can not exit calc
12
13            ENDP
14 ;*****
15 ; End of the program section
16 ;*****
17 ;LABEL      DIRECTIVE  VALUE      COMMENT
18             ALIGN
19             END
20
```

- *Init subroutine is: (taken from lecture notes)*

```

1  ;-----ADC-----;
2  RCGCADC EQU 0x400FE638 ; ADC clock register
3  ; ADC0 base address EQU 0x40038000
4  ADC0_ACTSS EQU 0x40038000 ; Sample sequencer (ADC0 base address)
5  ADC0_RIS EQU 0x40038004 ; Interrupt status
6  ADC0_IM EQU 0x40038008 ; Interrupt select
7  ADC0_EMUX EQU 0x40038014 ; Trigger select
8  ADC0_PSSI EQU 0x40038028 ; Initiate sample
9  ADC0_SSMUX3 EQU 0x400380A0 ; Input channel select
10 ADC0_SSCTL3 EQU 0x400380A4 ; Sample sequence control
11 ADC0_SSIFO3 EQU 0x400380A8 ; Channel 3 results
12 ADC0_PC EQU 0x40038FC4 ; Sample rate
13
14 ;-----PORT-----;
15 RCGCGPIO EQU 0x400FE608 ; GPIO clock register
16 ;PORT E base address EQU 0x40024000
17 PORTE_DEN EQU 0x4002451C ; Digital Enable
18 PORTE_PCTL EQU 0x4002452C ; Alternate function select
19 PORTE_AFSEL EQU 0x40024420 ; Enable Alt functions
20 PORTE_AMSEL EQU 0x40024528 ; Enable analog
21
22
23 ;PORTS ARE CONNECTED AS
24 ;PE<=Middle port of POT
25 ;Vbus <= Top port of POT
26 ;GND <= Bottom port of POT
27
28 AREA                INIT, READONLY, CODE
29 THUMB
30 EXPORT              Init
31
32 Init
33     PROC
34     PUSH             {R0,R1}
35     LDR               R1,=RCGCADC ; Turn on ADC clock
36     LDR               R0,[R1]
37     ORR               R0,R0,#0x01 ; set bit 0 to enable ADC0 clock
38     STR               R0,[R1]
39
40     NOP
41     NOP
42     NOP ; Let clock stabilize
43
44     LDR               R1,=RCGCGPIO
45     LDR               R0,[R1]
46     ORR               R0,R0,#0x10 ;To init port E
47     STR               R0,[R1]
48
49     NOP
50     NOP
51     NOP ; to stabilize clocks
52
53     ;These part is taken from lecture notes
54     LDR R1, =PORTE_AFSEL
55     LDR R0, [R1]
56     ORR R0, R0, #0x08 ; set bit 3 to enable alt functions on PE3
57     STR R0, [R1]
58
59     ; PCTL does not have to be configured
60     ; since ADC0 is automatically selected when
61     ; port pin is set to analog.
62
63     ; Disable digital on PE3
64     LDR R1, =PORTE_DEN
65     LDR R0, [R1]
66     BIC R0, R0, #0x08 ; clear bit 3 to disable digital on PE3
67     STR R0, [R1]
68
69     ; Enable analog on PE3
70     LDR R1, =PORTE_AMSEL
71     LDR R0, [R1]
72     ORR R0, R0, #0x08 ; set bit 3 to enable analog on PE3
73     STR R0, [R1]
74
75     ; Disable sequencer while ADC setup
76     LDR R1, =ADC0_ACTSS
77     LDR R0, [R1]
78     BIC R0, R0, #0x08 ; clear bit 3 to disable seq 3

```

```

78          STR        R0,[R1]
79
80          ; Select trigger source
81          LDR        R1,=ADC0_EMUX
82          LDR        R0,[R1]
83          BIC        R0,R0,#0xF000 ; clear bits 15:12 to select SOFTWARE
84          STR        R0,[R1] ; trigger
85
86          ; Select input channel
87          LDR        R1,=ADC0_SSMUX3
88          LDR        R0,[R1]
89          BIC        R0,R0,#0x000F ; clear bits 3:0 to select AIN0
90          STR        R0,[R1]
91
92          ; Config sample sequence
93          LDR        R1, =ADC0_SSCTL3
94          LDR        R0, [R1]
95          ORR        R0, R0, #0x06 ; set bits 2:1 (IE0, END0) IE0 is set since we want
RIS to be set
96          STR        R0, [R1]
97
98          ; Set sample rate
99          LDR        R1, =ADC0_PC
100         LDR        R0, [R1]
101         ORR        R0, R0, #0x01 ; set bits 3:0 to 1 for 125k sps
102         STR        R0, [R1]
103
104         ; Done with setup, enable sequencer
105         LDR        R1, =ADC0_ACTSS
106         LDR        R0, [R1]
107         ORR        R0, R0, #0x08 ; set bit 3 to enable seq 3
108         STR        R0, [R1] ; sampling enabled but not initiated yet;Disable
109
110         POP        {R0,R1}
111         BX         LR

```

- *Calc subroutine is:*

```

1  ADC0_RIS      EQU      0x40038004      ; Interrupt status
2  ADC0_PSSI     EQU      0x40038028      ; Initiate sample
3  ADC0_ISC      EQU      0x4003800C      ; Interrupt Status and Clear Register
4  ADC0_SSIFIFO3 EQU      0x400380A8      ; Channel 3 results
5
6              AREA      calculate, READONLY, CODE
7              THUMB
8              EXPORT    calc
9
10 calc         PROC
11             LDR        R0,=ADC0_PSSI
12             LDR        R1,=ADC0_RIS
13             LDR        R2,=ADC0_ISC
14             LDR        R3,=ADC0_SSIFIFO3
15 ;initiate sampling by enabling sequencer 3 in ADC0_PSSI
16 takeSample   LDR        R4,[R0]
17             MOV        R4,#0x08          ; which will enable sequencer3
18             STR        R4,[R0]
19 ;check for sample complete, wait otherwise
20 wait         LDR        R4,[R1]
21             ANDS       R4,#0x08
22             BEQ        wait
23 ;Sample is finished
24             LDR        R5,[R3]          ;Taking data and store it in R5
25
26             ;Clear interrput register
27             LDR        R4,[R2]
28             MOV        R4,#8
29             STR        R4,[R2]
30             ;Return the cycle
31             B          takeSample
32             ENDP

```

As you can understand I store data in R5. Some sample results are below:

Registers		Registers	
Register	Value	Register	Value
Core		Core	
R0	0x40038028	R0	0x40038028
R1	0x40038004	R1	0x40038004
R2	0x4003800c	R2	0x4003800c
R3	0x400380a8	R3	0x400380a8
R4	0x00000000	R4	0x00000000
R5	0x00000000	R5	0x00000fff
R6	0x00000000	R6	0x00000000
R7	0x00000000	R7	0x00000000
R8	0x00000000	R8	0x00000000
R9	0x00000000	R9	0x00000000
R10	0x00000000	R10	0x00000000
R11	0x00000000	R11	0x00000000
R12	0x00000000	R12	0x00000000
R13 (SP)	0x20000400	R13 (SP)	0x20000400
R14 (LR)	0x00000371	R14 (LR)	0x00000371
R15 (PC)	0x00000346	R15 (PC)	0x00000346
xPSR	0x41000000	xPSR	0x41000000

Registers		Registers	
Register	Value	Register	Value
Core		Core	
R0	0x40038028	R0	0x40038028
R1	0x40038004	R1	0x40038004
R2	0x4003800c	R2	0x4003800c
R3	0x400380a8	R3	0x400380a8
R4	0x00000008	R4	0x00000000
R5	0x000008b2	R5	0x000002dc
R6	0x00000000	R6	0x00000000
R7	0x00000000	R7	0x00000000
R8	0x00000000	R8	0x00000000
R9	0x00000000	R9	0x00000000
R10	0x00000000	R10	0x00000000
R11	0x00000000	R11	0x00000000
R12	0x00000000	R12	0x00000000
R13 (SP)	0x20000400	R13 (SP)	0x20000400
R14 (LR)	0x00000371	R14 (LR)	0x00000371
R15 (PC)	0x0000034a	R15 (PC)	0x0000033e
xPSR	0x01000000	xPSR	0x01000000

Question 2)

In this question, I just changed calc subroutine. Init and main remains same.

- Calc subroutine

D:\OKUL\ee_4un_1\Lab-447\LabSix\QuestionTwo\RESULTS.s

```
1  ADC0_RIS      EQU      0x40038004      ; Interrupt status
2  ADC0_PSSI     EQU      0x40038028      ; Initiate sample
3  ADC0_ISC      EQU      0x4003800C      ;Interrupt Status and Clear Register
4  ADC0_SSIFO3   EQU      0x400380A8      ; Channel 3 results
5
6              AREA      calculate, READONLY, CODE
7              THUMB
8              EXPORT    calc
9
10 calc          PROC
11              LDR        R0,=ADC0_PSSI
12              LDR        R1,=ADC0_RIS
13              LDR        R2,=ADC0_ISC
14              LDR        R3,=ADC0_SSIFO3
15              ;1.65/3.3 = 0.5
16              ;0xFFF * 0.5 ~= 0x800
17              MOV        R10,#0x800
18              ;initiate sampling by enabling sequencer 3 in ADC0_PSSI
19 takeSample     LDR        R4,[R0]
20              MOV        R4,#0x08          ; which will enable sequencer3
21              STR        R4,[R0]
22              ;check for sample complete, wait otherwise
23 wait          LDR        R4,[R1]
24              ANDS       R4,#0x08
25              BEQ        wait
26              ;Sample is finished
27              LDR        R5,[R3]          ;Taking data and store it in R5
28              SUB        R5,R10          ;Offset is taken account.
29              ;Clear interrupt register
30              LDR        R4,[R2]
31              MOV        R4,#8
32              STR        R4,[R2]
33              ;Return the cycle
34              B          takeSample
35              ENDP
```

Results: (R5)

Register	Value
Core	
R0	0x40038028
R1	0x40038004
R2	0x4003800c
R3	0x400380a8
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000800
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000400
R14 (LR)	0x00000379
R15 (PC)	0x00000348
xPSR	0x41000000

Register	Value
Core	
R0	0x40038028
R1	0x40038004
R2	0x4003800c
R3	0x400380a8
R4	0x00000008
R5	0x0000007f
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000800
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000400
R14 (LR)	0x00000379
R15 (PC)	0x0000034a
xPSR	0x41000000

Register	Value
Core	
R0	0x40038028
R1	0x40038004
R2	0x4003800c
R3	0x400380a8
R4	0x00000008
R5	0xffff800
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000800
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000400
R14 (LR)	0x00000379
R15 (PC)	0x00000340
xPSR	0x01000000

Register	Value
Core	
R0	0x40038028
R1	0x40038004
R2	0x4003800c
R3	0x400380a8
R4	0x00000000
R5	0xffffb88
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000800
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000400
R14 (LR)	0x00000379
R15 (PC)	0x00000348
xPSR	0x41000000

Question 3)

In this question, I couldn't get the difference with last question. Therefore, in this question I stored value in a register in this question. There is no update on main and init files. However, I changed results part as follows:

D:\OKUL\ee_4un_1\Lab-447\LabSix\QuestionThreeandFour\RESULTS.s

```
1  ADC0_RIS      EQU      0x40038004      ; Interrupt status
2  ADC0_PSSI     EQU      0x40038028      ; Initiate sample
3  ADC0_ISC      EQU      0x4003800C      ;Interrupt Status and Clear Register
4  ADC0_SSIFIFO3 EQU      0x400380A8      ; Channel 3 results
5
6              AREA      calculate, READONLY, CODE
7              THUMB
8              EXPORT   calc
9
10
11  calc          PROC
12              LDR      R0,=ADC0_PSSI
13              LDR      R1,=ADC0_RIS
14              LDR      R2,=ADC0_ISC
15              LDR      R3,=ADC0_SSIFIFO3
16              ;1.65/3.3 = 0.5
17              ;0xFFF * 0.5 ~= 0x800
18              MOV      R10,#0x800
19              ;initiate sampling by enabling sequencer 3 in ADC0_PSSI
20  takeSample    LDR      R4,[R0]
21              MOV      R4,#0x08          ; which will enable sequencer3
22              STR      R4,[R0]
23              ;check for sample complete, wait otherwise
24  wait          LDR      R4,[R1]
25              ANDS     R4,#0x08
26              BEQ      wait
27              ;Sample is finished
28              LDR      R5,[R3]            ;Taking data and store it in R5
29              SUB      R5,R10            ;Offset is taken account.
30              CMP      R5,#0x01          ;Is it negative ?
31              BMI      negativ            ;yes go to negativ subroutine
32              B         decimals         ;no go to decimals subroutine
33
34
35
36
37  negativ        MOV      R4,#0xFFFFFFFF ;Since result is negative,
38              SUB      R5,R4,R5          ;Take absolute value of it
39
40              ;Result is in form of X.YZ
41              ;#2048 ~= #0x800
42  decimals        MOV      R11,#1241      ;2048 ~ 1.65 ; 1241 ~1.00
43              SDIV     R6,R5,R11          ; R6 is holding X digit R6=X
44              MUL      R9,R6,R11          ; Update the reminder
45              SUB      R5,R5,R9          ; Update the reminder; R5= 0.YZ
46              LSL      R6,#8              ; R6 = 0xX00
47
48              MOV      R11,#124          ;2048 ~ 1.65 ; 124 ~0.10
49              SDIV     R7,R5,R11          ;R7 is holding Y digit R7=Y
50              MUL      R9,R7,R11          ;Update the reminder
51              SUB      R5,R5,R9          ;Update the reminder ; R5= 0.0Z
52              LSL      R7,#4              ;R7=0xXY0
53              ADD      R6,R7              ;R6=0xXY0
54
55              MOV      R11,#12           ;2048 ~ 1.65 ; 12 ~0.01
56              SDIV     R8,R5,R11          ;R8 is holding Z digit R8=Z
57              ADD      R5,R6,R8          ;R5=0xXYZ
58              B         exit
59
60              ;Clear interrupt register
61  exit          LDR      R4,[R2]
62              MOV      R4,#8
63              STR      R4,[R2]
64              ;Return the cycle
65              B         takeSample
66              ENDP
```


Results again in R5: (Note that left top => -1.65, right bottom => +1.65)

Registers		Registers	
Register	Value	Register	Value
Core		Core	
R0	0x40038028	R0	0x40038028
R1	0x40038004	R1	0x40038004
R2	0x4003800c	R2	0x4003800c
R3	0x400380a8	R3	0x400380a8
R4	0xffffffff	R4	0xffffffff
R5	0x00000165	R5	0x00000045
R6	0x00000160	R6	0x00000040
R7	0x00000060	R7	0x00000040
R8	0x00000005	R8	0x00000005
R9	0x000002e8	R9	0x000001f0
R10	0x00000800	R10	0x00000800
R11	0x0000000c	R11	0x0000000c
R12	0x00000000	R12	0x00000000
R13 (SP)	0x20000400	R13 (SP)	0x20000400
R14 (LR)	0x000003c1	R14 (LR)	0x000003c1
R15 (PC)	0x0000039a	R15 (PC)	0x0000039a
xPSR	0xa1000000	xPSR	0xa1000000
+ Banked		+ Banked	
		+ System	

Question 4)

```

1  ADC0_RIS      EQU      0x40038004      ; Interrupt status
2  ADC0_PSSI     EQU      0x40038028      ; Initiate sample
3  ADC0_ISC      EQU      0x4003800C      ; Interrupt Status and Clear Register
4  ADC0_SSIF03   EQU      0x400380A8      ; Channel 3 results
5
6              AREA      calculate, READONLY, CODE
7              THUMB
8              IMPORT    OutChar
9              IMPORT    DELAY100
10             IMPORT    CONVRT
11             EXPORT    calc
12
13
14 calc          PROC
15             LDR        R0,=ADC0_PSSI
16             LDR        R1,=ADC0_RIS
17             LDR        R2,=ADC0_ISC
18             LDR        R3,=ADC0_SSIF03
19
20 ;1.65/3.3 = 0.5
21 ;0xFFFF * 0.5 ~= 0x800
22 ;initiate sampling by enabling sequencer 3 in ADC0_PSSI
23 takeSample    LDR        R4,[R0]
24             MOV        R4,#0x08          ; which will enable sequencer3
25             STR        R4,[R0]
26 ;check for sample complete, wait otherwise
27 wait         LDR        R4,[R1]
28             ANDS       R4,#0x08
29             BEQ        wait
30 ;Sample is finished
31             LDR        R5,[R3]          ;Taking data and store it in R5
32             SUB        R5,R10          ;Offset is taken account.
33             CMP        R5,#0x01        ;Is it negative ?
34             BMI        negativ          ;yes go to negativ subroutine
35             B          decimals        ;no go to deicmals subroutine
36
37
38
39
40 negativ       MOV        R4,#0xFFFFFFFF ;Since result is negative,
41             SUB        R5,R4,R5        ;Take absolte value of it
42
43 ;Result is in form of X.YZ
44 ;#2048 ~= #0x800
45 decimals      MOV        R11,#1241      ;2048 ~ 1.65 ; 1241 ~1.00
46             SDIV       R6,R5,R11        ; R6 is holding X digit R6=X
47             MUL        R9,R6,R11        ; Update the reminder
48             SUB        R5,R5,R9        ; Update the reminder; R5= 0.YZ
49             LSL        R6,#8            ; R6 = 0xX00
50
51             MOV        R11,#124         ;2048 ~ 1.65 ; 124 ~0.10
52             SDIV       R7,R5,R11        ;R7 is holding Y digit R7=Y
53             MUL        R9,R7,R11        ;Update the reminder
54             SUB        R5,R5,R9        ;Update the reminder ; R5= 0.0Z
55             LSL        R7,#4            ;R7=0x0Y0
56             ADD        R6,R7           ;R6=0xXY0
57
58             MOV        R11,#12          ;2048 ~ 1.65 ; 12 ~0.01
59             SDIV       R8,R5,R11        ;R8 is holding Z digit R8=Z
60             ADD        R5,R6,R8        ;R5=0xXYZ
61             B          exit
62
63
64 exit          CMP        R4,#0xFFFFFFFF ;NEGATIVE CHECKER
65             BLEQ       minus           ;If it is negative then print minus
66 ;R5 = X.YZ
67             AND        R4,R5,0xF00      ; Take digit X
68             LSR        R4,#8
69             BL         CONVRT           ;Print X
70             BL         dot             ;Print dot
71             AND        R4,R5,0x0F0     ;Take digit Y
72             LSR        R4,#4
73             BL         CONVRT           ;Print Y
74             AND        R4,R5,0x00F     ;Take digit Z
75             BL         CONVRT           ;Print digit Z
76 ;Clear interrput register
77             LDR        R4,[R2]

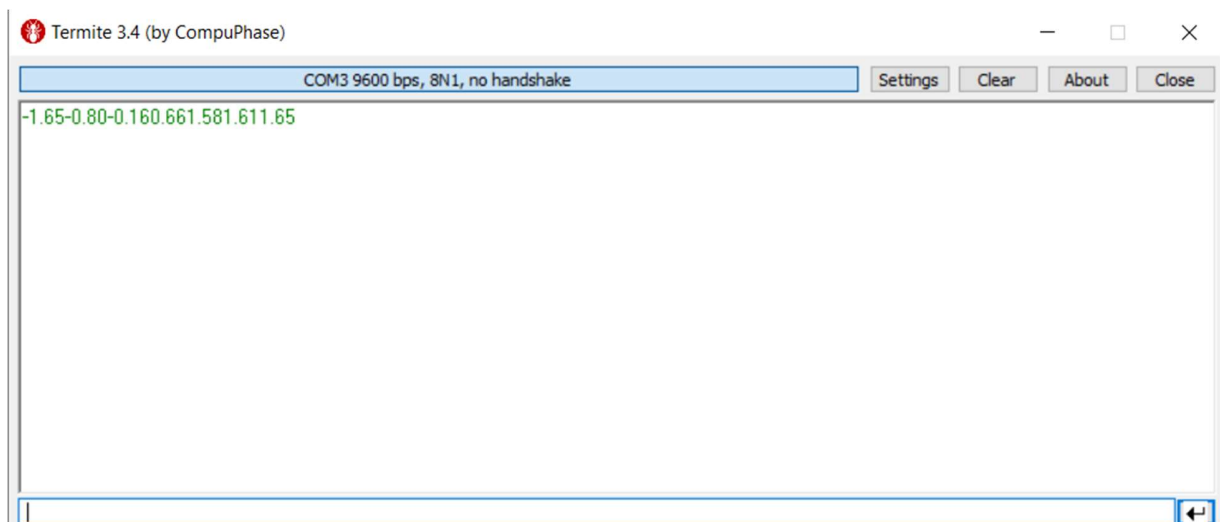
```

D:\OKUL\ee_4un_1\Lab-447\LabSix\QuestionFour\RESULTS.s

```
78      MOV      R4, #8
79      STR      R4, [R2]
80      ;Return the cycle
81      BL       DELAY100      ;Delaying
82      B        takeSample   ;Looping
83
84
85      dot      PUSH    {R5, LR}
86              MOV     R5, 0x2E
87              BL      OutChar
88              POP     {R5, LR}
89              BX      LR
90
91      minus    PUSH    {R5, LR}
92              MOV     R5, 0x2D
93              BL      OutChar
94              POP     {R5, LR}
95              BX      LR
96
97      ENDP
```

I just modified exit subroutine in the code as you can see. DELAY100 subroutine is to delay system.

Results is:



Note that I do not put a space character in this question.