18/12/2018

# EE447 LABORATORY
# EXPERIMENT 5
# PRELIMINARY REPORT

Muttalip Caner TOL

2031466

Wednesday Afternoon

```
;*****************************************************************
; Program ADC.s
; Takes analog signal between 000-FFF gives between 0-330
;*****************************************************************
; EQU Directives

;*****************************************************************
;LABEL          DIRECTIVE   VALUE           COMMENTs

; ADC Registers
RCGCADC         EQU         0x400FE638      ; ADC clock register
; ADC0 base address EQU 0x40038000
ADC0_ACTSS      EQU         0x40038000      ; Sample sequencer (ADC0 base address)
ADC0_RIS        EQU         0x40038004      ; Interrupt status
ADC0_IM         EQU         0x40038008      ; Interrupt select
ADC0_EMUX       EQU         0x40038014      ; Trigger select
ADC0_PSSI       EQU         0x40038028      ; Initiate sample
ADC0_SSMUX3     EQU         0x400380A0      ; Input channel select
ADC0_SSCTL3     EQU         0x400380A4      ; Sample sequence control
ADC0_SSFIFO3    EQU         0x400380A8      ; Channel 3 results
ADC0_PC         EQU         0x40038FC4      ; Sample rate
ADC0_ISC        EQU         0x4003800C      ; Interrupt Status and Clear
; GPIO Registers
RCGCGPIO        EQU         0x400FE608      ; GPIO clock register
;PORT E base address EQU 0x40024000
PORTE_DEN       EQU         0x4002451C      ; Digital Enable
PORTE_PCTL      EQU         0x4002452C      ; Alternate function select
PORTE_AFSEL     EQU         0x40024420      ; Enable Alt functions
PORTE_AMSEL     EQU         0x40024528      ; Enable analog


;*****************************************************************
;LABEL      DIRECTIVE   VALUE               COMMENT
            AREA        main, READONLY, CODE
            THUMB
            EXTERN      CONVRT              ; Reference external subroutines
            EXPORT      __ADC               ; Make available

__ADC       PUSH        {R0-R6,R8-R12}  ; Start clocks for features to be used
            LDR         R1, =RCGCADC        ; Turn on ADC clock
            LDR         R0, [R1]
            ORR         R0, R0, #0x01       ; set bit 0 to enable ADC0 clock
            STR         R0, [R1]
            NOP
            NOP
            NOP                             ; Let clock stabilize
            LDR         R1, =RCGCGPIO       ; Turn on GPIO clock
            LDR         R0, [R1]
            ORR         R0, R0, #0x10       ; set bit 4 to enable port E clock
            STR         R0, [R1]
            NOP
            NOP
            NOP                             ; Let clock stabilize
            ; Setup GPIO to make PE3 input for ADC0
            ; Enable alternate functions
            LDR         R1, =PORTE_AFSEL
            LDR         R0, [R1]
            ORR         R0, R0, #0x08       ; set bit 3 to enable alt functions on PE3
            STR         R0, [R1]
            ; PCTL does not have to be configured
            ; since ADC0 is automatically selected when
            ; port pin is set to analog.
            ; Disable digital on PE3
            LDR         R1, =PORTE_DEN
            LDR         R0, [R1]
            BIC         R0, R0, #0x08       ; clear bit 3 to disable digital on PE3
            STR         R0, [R1]
            ; Enable analog on PE3
            LDR         R1, =PORTE_AMSEL
            LDR         R0, [R1]
            ORR         R0, R0, #0x08       ; set bit 3 to enable analog on PE3
            STR         R0, [R1]
            ; Disable sequencer while ADC setup
            LDR         R1, =ADC0_ACTSS
            LDR         R0, [R1]
            BIC         R0, R0, #0x08       ; clear bit 3 to disable SS3
            STR         R0, [R1]
```

```
78              ; Select trigger source
79              LDR         R1, =ADC0_EMUX
80              LDR         R0, [R1]
81              BIC         R0, R0, #0xF000     ; clear bits 15:12 to select SOFTWARE
82              STR         R0, [R1]            ; trigger
83              ; Select input channel
84              LDR         R1, =ADC0_SSMUX3
85              LDR         R0, [R1]
86              BIC         R0, R0, #0x000F     ; clear bits 3:0 to select AIN0
87              STR         R0, [R1]
88              ; Config sample sequence
89              LDR         R1, =ADC0_SSCTL3
90              LDR         R0, [R1]
91              ORR         R0, R0, #0x06       ; set bits 2:1 (IE0, END0)
92              STR         R0, [R1]
93              ; Set sample rate
94              LDR         R1, =ADC0_PC
95              LDR         R0, [R1]
96              ORR         R0, R0, #0x01       ; set bits 3:0 to 0x1 for 125k sps
97              STR         R0, [R1]
98              ; Done with setup, enable sequencer
99              LDR         R1, =ADC0_ACTSS
100             LDR         R0, [R1]
101             ORR         R0, R0, #0x08       ; set bit 3 to enable seq 3
102             STR         R0, [R1]            ; sampling enabled but not initiated yet
103             ; start sampling routine
104             LDR         R3, =ADC0_RIS       ; interrupt address
105             LDR         R4, =ADC0_SSFIFO3   ; result address
106             LDR         R2, =ADC0_PSSI      ; sample sequence initiate address
107             LDR         R6, =ADC0_ISC
108             ; initiate sampling by enabling sequencer 3 in ADC0_PSSI
109             LDR         R0, [R2]
110             ORR         R0, R0, #0x08       ; set bit 3 for SS3
111             STR         R0, [R2]
112             ; check for sample complete (bit 3 of ADC0_RIS set)
113 Cont        LDR         R0, [R3]
114             ANDS        R0, R0, #8
115             BEQ         Cont
116             ;branch fails if the flag is set so data can be read and flag is cleared
117             LDR         R7,[R4]
118             ;map between 0-330
119             MOV         R0, #806
120             MOV         R1, #10000
121             MUL         R7, R0
122             UDIV        R7, R1
123
124             ;MOV         R0, R5              ;store the data
125             MOV         R0, #8
126             STR         R0, [R6]            ; clear flag
127             POP         {R0-R6,R8-R12}
128             BX          LR
129
130             END
```

```
;*****************************************************************
; Program BCD.s
; This program converts the hex value to a BCD number with three
; digits (X.YZ) between 3.30 and 0.00 .
;*****************************************************************
; EQU Directives
;*****************************************************************
;LABEL      DIRECTIVE   VALUE               COMMENT
            AREA        main, READONLY, CODE
            THUMB
            EXTERN      __ADC
            EXTERN      CONVRT              ; Reference external subroutines
            EXTERN      OutChar
            EXPORT      __main              ; Make available

__main      PUSH        {LR}
            BL          __ADC
            POP         {LR}
            PUSH        {LR}
            BL          CONVRT
            POP         {LR}
            MOV         R0, R7
loop        BL          __ADC
            CMP         R7, R0
            ITE HS                          ; R1 <- |R7-R0|
            SUBHS       R1, R7, R0          ;
            SUBLO       R1, R0, R7          ;
            CMP         R1, #20             ; Treshold = 0.20 Volt
            BLO         loop
            CMP         R7, #10             ; Single digit or not
            BHS         high
            MOV         R5,#0xA             ; New line
            BL          OutChar
            MOV         R5,#0x30            ; print "0"
            BL          OutChar
            MOV         R5,#0x2E            ; print "."
            BL          OutChar
            MOV         R5,#0x30            ; print "0"
            BL          OutChar
            PUSH        {LR}
            BL          CONVRT              ; print single digit
            POP         {LR}
            MOV         R0, R7
            B           loop

high        MOV         R2, #100
            MOV         R8, R7
            UDIV        R7, R2
            MOV         R5,#0xA             ; add new line
            BL          OutChar
            PUSH        {LR}
            BL          CONVRT              ; print the quotient
            POP         {LR}
            MOV         R5,#0x2E            ; print "."
            BL          OutChar
            CMP         R7, #0
            BNE         THREE
            MOV         R7, R8
            PUSH        {LR}
            BL          CONVRT              ; print the quotient
            POP         {LR}
            MOV         R0, R8
            B           loop

THREE       MUL         R7, R2
            SUBS        R7, R8, R7
            CMP         R7, #10
            BHS         high2
            MOV         R5,#0x30            ; print "0"
            BL          OutChar
high2       PUSH        {LR}
            BL          CONVRT              ; print the remainder
            POP         {LR}
            MOV         R0, R8
            B           loop
            END
```