```

AVRASM ver. 2.2.7  E:\ESE_280\$MyDocuments$\Atmel Studio\7.0\lab_11\ADC_MCP9700A\ADC_MCP9700A\main.asm Tue Nov 17 18:37:40 2020

E:\ESE_280\$MyDocuments$\Atmel Studio\7.0\lab_11\ADC_MCP9700A\ADC_MCP9700A\main.asm(9): Including file 'C:/Program Files (x86)\Atmel\Studio\7.0\Packs\atmel\ATmega_DFP\1.3.300\avrasm\inc\m4809def.inc'
E:\ESE_280\$MyDocuments$\Atmel Studio\7.0\lab_11\ADC_MCP9700A\ADC_MCP9700A\main.asm(9): Including file 'C:/Program Files (x86)\Atmel\Studio\7.0\Packs\atmel\ATmega_DFP\1.3.300\avrasm\inc\m4809def.inc'


                                 ; ADC_MCP9700A.asm
                                 ;
                                 ; Created: 11/17/2020 2:02:20 PM
                                 ; Author : Judah Ben-Eliezer
                                 ;

                                 .list

                                 .equ PERIOD_EXAMPLE_VALUE = 25

                                 .dseg
002800                           led_display: .byte 4
002804                           digit_num: .byte 1


                                 .cseg

                                 reset:
000000 940c 002e                    jmp start

                                 .org TCA0_OVF_vect
00000e 940c 0056                    jmp post_display_ISR

                                 .org ADC0_RESRDY_vect
00002c 940c 006a                    jmp read_ISR

                                 start:
                                    ;configure inputs and outputs
00002e 9883                          cbi VPORTE_DIR, 3
00002f ef0f                         ldi r16, $FF
000030 b908                         out VPORTC_DIR, r16
000031 b90c                         out VPORTD_DIR, r16
000032 9500                         com r16
000033 b909                         out VPORTC_OUT, r16
000034 b90d                         out VPORTD_OUT, r16

                                    ;configure TCA0
```

```
000035 e000                     ldi r16, TCA_SINGLE_WGMODE_NORMAL_gc ;WGMODE normal
000036 9300 0a01                sts TCA0_SINGLE_CTRLB, r16

                                ;enable overflow interrupt
000038 e001                     ldi r16, TCA_SINGLE_OVF_bm
000039 9300 0a0a                sts TCA0_SINGLE_INTCTRL, r16

                                ;load period low byte then high byte
00003b e109                     ldi r16, LOW(PERIOD_EXAMPLE_VALUE)
00003c 9300 0a26                sts TCA0_SINGLE_PER, r16
00003e e000                     ldi r16, HIGH(PERIOD_EXAMPLE_VALUE)
00003f 9300 0a27                sts TCA0_SINGLE_PER + 1, r16

                                ;set clock and start timer
000041 e00d                     ldi r16, TCA_SINGLE_CLKSEL_DIV256_gc | TCA_SINGLE_ENABLE_bm
000042 9300 0a00                sts TCA0_SINGLE_CTRLA, r16

                                ;set voltage reference
000044 e200                     ldi r16, VREF_ADC0REFSEL_2V5_gc
000045 9300 00a0                sts VREF_CTRLA, r16

                                ;select PE1/ AIN9
000047 e00b                     ldi r16, ADC_MUXPOS_AIN11_gc
000048 9300 0606                sts ADC0_MUXPOS, r16

                                ;enable internal reference and set prescaler to div 64
00004a e005                     ldi r16, ADC_PRESC_DIV64_gc | ADC_REFSEL_INTREF_gc
00004b 9300 0602                sts ADC0_CTRLC, r16

                                ;set resolution to 10 bit and enable adc
00004d e001                     ldi r16, ADC_RESSEL_10BIT_gc | ADC_ENABLE_bm;
00004e 9300 0600                sts ADC0_CTRLA, r16

                                ;start conversion
000050 e001                     ldi r16, ADC_STCONV_bm;
000051 9300 0608                sts ADC0_COMMAND, r16

                                ;enable interrupts
000053 9478                     sei
000054 940c 0065                jmp wait_for_post

                             ;*****************************************************************************
                             ;*
```

```
                            ;* "post_display" - title
                            ;*
                            ;* Description: toggles value for all PORTC pins.  Since PORTC is used to multiplex the led display, this will
                            ;* turn the LED display on and off
                            ;* Author: Judah Ben-Eliezer
                            ;* Version:    1.0
                            ;* Last updated:   11/17
                            ;* Target: ATmega4809
                            ;* Number of words:    13
                            ;* Number of cycles:   6
                            ;* Low registers modified:
                            ;* High registers modified:
                            ;* Parameters: none
                            ;* Returns:        none
                            ;*
                            ;* Notes:
                            ;*
                            ;***************************************************************************
                            post_display_ISR:
000056 930f                    push r16
000057 b70f                    in r16, CPU_SREG
000058 930f                    push r16
000059 931f                    push r17

00005a ef1f                    ldi r17, $FF
00005b b109                    in r16, VPORTC_OUT
00005c 2701                    eor r16, r17
00005d b909                    out VPORTC_OUT, r16

                               ;ldi r16, TCA_SINGLE_OVF_bm ;clear OVF flag
                               ;sts TCA0_SINGLE_INTFLAGS, r16

00005e 911f                    pop r17
00005f 910f                    pop r16
000060 bf0f                    out CPU_SREG, r16
000061 910f                    pop r16

000062 9478                    sei
000063 940c 0067               jmp main_loop

                            wait_for_post:
000065 0000                    nop
000066 cffe                    rjmp wait_for_post

                            main_loop:
000067 d02a                    rcall multiplex_display
```

```
000068 d03e                          rcall mux_digit_delay
000069 cffd                          rjmp main_loop

                                 ;***************************************************************************************
                                 ;*
                                 ;* "read_ISR" - title
                                 ;*
                                 ;* Description: loads ADC0_RES into r17:r16 and calls bin16_to_led
                                 ;*
                                 ;* Author: Judah Ben-Eliezer
                                 ;* Version:     1.0
                                 ;* Last updated:    11/17/2020
                                 ;* Target: ATmega4809
                                 ;* Number of words:
                                 ;* Number of cycles:
                                 ;* Low registers modified: none
                                 ;* High registers modified:        r17:r16
                                 ;*
                                 ;* Parameters: ADC0_RES
                                 ;* Returns:         r17:r16
                                 ;*
                                 ;* Notes:
                                 ;*
                                 ;***************************************************************************************
                                 read_ISR:
00006a 930f                          push r16
00006b b70f                          in r16, CPU_SREG
00006c 930f                          push r16
00006d 2f0b                          mov r16, XH
00006e 930f                          push r16
00006f 2f0a                          mov r16, XL
000070 930f                          push r16
000071 2f0f                          mov r16, ZH
000072 930f                          push r16
000073 2f0e                          mov r16, ZL
000074 930f                          push r16
000075 931f                          push r17
000076 932f                          push r18
000077 933f                          push r19

000078 9110 0611                     lds r17, ADC0_RESH
00007a 9100 0610                     lds r16, ADC0_RESL
00007c d031                          rcall bin16_to_led

                                     ;reset interrupt flag
00007d e001                          ldi r16, ADC_RESRDY_bm;
```

```
00007e 9300 060a                    sts ADC0_INTCTRL, r16

                                    ;restart conversion
000080 e001                         ldi r16, ADC_STCONV_bm;
000081 9300 0608                    sts ADC0_COMMAND, r16

000083 913f                         pop r19
000084 912f                         pop r18
000085 911f                         pop r17
000086 910f                         pop r16
000087 2fe0                         mov ZL, r16
000088 910f                         pop r16
000089 2ff0                         mov ZH, r16
00008a 910f                         pop r16
00008b 2fa0                         mov XL, r16
00008c 910f                         pop r16
00008d 2fb0                         mov XH, r16
00008e 910f                         pop r16
00008f bf0f                         out CPU_SREG, r16
000090 910f                         pop r16

000091 9518                         reti

                                 ;***************************************************************************
                                 ;*
                                 ;* "multiplex_display" - title
                                 ;*
                                 ;* Description:    outputs values from led_display array to 7 segment display on PORTD driven by highest two bits of PORTC
                                 ;*
                                 ;* Author: Judah Ben-Eliezer
                                 ;* Version:    1.0
                                 ;* Last updated:   11/10/2020
                                 ;* Target: ATmega4809
                                 ;* Number of words:
                                 ;* Number of cycles:
                                 ;* Low registers modified:
                                 ;* High registers modified:
                                 ;*
                                 ;* Parameters:
                                 ;* Returns:
                                 ;*
                                 ;* Notes:
                                 ;*
                                 ;***************************************************************************
                                 multiplex_display:
```

```
000092 e2d8                            ldi YH, HIGH(led_display)
000093 e0c0                            ldi YL, LOW(led_display)
000094 9110 2804                       lds r17, digit_num
000096 7013                            andi r17, $03
000097 2f41                            mov r20, r17
000098 0fc1                            add YL, r17
000099 8128                            ld r18, Y
00009a e850                            ldi r21, $80
00009b 9543                            inc r20
                                 loop:
00009c 9556                            lsr r21
00009d 954a                            dec r20
00009e f7e9                            brne loop
00009f 0f55                            lsl r21
0000a0 9550                            com r21
0000a1 b959                            out VPORTC_OUT, r21
0000a2 b92d                            out VPORTD_OUT, r18
0000a3 9513                            inc r17
0000a4 9310 2804                       sts digit_num, r17
0000a6 9508                            ret

                                 ;****************************************************************************
                                 ;*
                                 ;* "mux_digit_delay" - title
                                 ;*
                                 ;* Description: delays 0.1 * r23
                                 ;*
                                 ;* Author:  Judah Ben-Eliezer
                                 ;* Version:     1.0
                                 ;* Last updated:
                                 ;* Target:
                                 ;* Number of words:
                                 ;* Number of cycles:
                                 ;* Low registers modified:
                                 ;* High registers modified:
                                 ;*
                                 ;* Parameters:
                                 ;* Returns:
                                 ;*
                                 ;* Notes:
                                 ;*
                                 ;****************************************************************************
                                 mux_digit_delay:
0000a7 e078                            ldi r23, $08 ; 0.1 * r23 = delay
                                 outer_loop:
0000a8 e086                            ldi r24, $06
                                 inner_loop:
```

```
0000a9 958a                     dec r24
0000aa f7f1                     brne inner_loop
0000ab 957a                     dec r23
0000ac f7d9                     brne outer_loop
0000ad 9508                     ret

                            ;***************************************************************************
                            ;*
                            ;* "bin16_to_led" - title
                            ;*
                            ;* Description:    Converts bin16 input to 7seg output, from bcd_entries array to led_display array
                            ;*
                            ;* Author: Judah Ben-Eliezer
                            ;* Version:    1.0
                            ;* Last updated:   11/17/2020
                            ;* Target: ATmega4809
                            ;* Number of words:
                            ;* Number of cycles:
                            ;* Low registers modified:
                            ;* High registers modified:
                            ;*
                            ;* Parameters: r17:r16 16 bit binary number.
                            ;* Returns:    none
                            ;*
                            ;* Notes:
                            ;*
                            ;***************************************************************************

                            bin16_to_led:
0000ae e2b8                     ldi XH, HIGH(led_display)
0000af e0a0                     ldi XL, LOW(led_display)
0000b0 2f21                     mov r18, r17
0000b1 7f20                     andi r18, $F0
0000b2 9522                     swap r18
0000b3 d00f                     rcall hex_to_7seg
0000b4 932d                     st X+, r18
0000b5 2f21                     mov r18, r17
0000b6 702f                     andi r18, $0F
0000b7 d00b                     rcall hex_to_7seg
0000b8 932d                     st X+, r18
0000b9 2f20                     mov r18, r16
0000ba 7f20                     andi r18, $F0
0000bb 9522                     swap r18
0000bc d006                     rcall hex_to_7seg
0000bd 932d                     st X+, r18
0000be 2f20                     mov r18, r16
```

```
0000bf 702f                         andi r18, $0F
0000c0 d002                         rcall hex_to_7seg
0000c1 932c                         st X, r18
0000c2 9508                         ret

                                 ;***************************************************************************
                                 ;*
                                 ;* "hex_to_7seg" - Hexadecimal to Seven Segment Conversion
                                 ;*
                                 ;* Description: Converts a right justified hexadecimal digit to the seven
                                 ;* segment pattern required to display it. Pattern is right justified a
                                 ;* through g. Pattern uses 0s to turn segments on ON.
                                 ;*
                                 ;* Author:                      Ken Short
                                 ;* Version:                         1.0
                                 ;* Last updated:                101620
                                 ;* Target:                      ATmega4809
                                 ;* Number of words:                 8
                                 ;* Number of cycles:            13
                                 ;* Low registers modified:      none
                                 ;* High registers modified:         r19, r18, ZL, ZH
                                 ;*
                                 ;* Parameters: r18: right justified hex digit, high nibble 0
                                 ;* Returns: r18: segment values a through g right justified
                                 ;*
                                 ;* Notes:
                                 ;*
                                 ;***************************************************************************

                                 hex_to_7seg:
0000c3 702f                         andi r18, 0x0F              ;clear ms nibble
0000c4 e0f1                          ldi ZH, HIGH(hextable * 2)  ;set Z to point to start of table
0000c5 e9e6                          ldi ZL, LOW(hextable * 2)
0000c6 e030                          ldi r19, $00                ;add offset to Z pointer
0000c7 0fe2                          add ZL, r18
0000c8 1ff3                          adc ZH, r19
```

```
0000c9 9124                          lpm r18, Z                  ;load byte from table pointed to by Z
0000ca 9508                         ret


                                     ;Table of segment values to display digits 0 - F
                                     ;!!! seven values must be added - verify all values
0000cb 4f01
0000cc 0612
0000cd 244c
0000ce 0f20
0000cf 0400
0000d0 6008
0000d1 4231
0000d2 3830                      hextable: .db $01, $4F, $12, $06, $4C, $24, $20, $0F, $00, $04, $08, $60, $31, $42, $30, $38


RESOURCE USE INFORMATION
------------------------

Notice:
The register and instruction counts are symbol table hit counts,
and hence implicitly used resources are not counted, eg, the
'lpm' instruction without operands implicitly uses r0 and z,
none of which are counted.

x,y,z are separate entities in the symbol table and are
counted separately from r26..r31 here.

.dseg memory usage only counts static data declared with .byte

"ATmega4809" register use summary:
x  :   4 y  :   1 z  :   1 r0 :   0 r1 :   0 r2 :   0 r3 :   0 r4 :   0
r5 :   0 r6 :   0 r7 :   0 r8 :   0 r9 :   0 r10:   0 r11:   0 r12:   0
r13:   0 r14:   0 r15:   0 r16:  64 r17:  15 r18:  21 r19:   4 r20:   3
r21:   5 r22:   0 r23:   2 r24:   2 r25:   0 r26:   3 r27:   3 r28:   2
r29:   1 r30:   4 r31:   4
Registers used: 17 out of 35 (48.6%)

"ATmega4809" instruction use summary:
.lds  :   0 .sts  :   0 adc   :   1 add   :   2 adiw  :   0 and   :   0
andi  :   6 asr   :   0 bclr  :   0 bld   :   0 brbc  :   0 brbs  :   0
brcc  :   0 brcs  :   0 break :   0 breq  :   0 brge  :   0 brhc  :   0
brhs  :   0 brid  :   0 brie  :   0 brlo  :   0 brlt  :   0 brmi  :   0
brne  :   3 brpl  :   0 brsh  :   0 brtc  :   0 brts  :   0 brvc  :   0
brvs  :   0 bset  :   0 bst   :   0 call  :   0 cbi   :   1 cbr   :   0
clc   :   0 clh   :   0 cli   :   0 cln   :   0 clr   :   0 cls   :   0
```

```
clt   :   0 clv   :   0 clz   :   0 com   :   2 cp    :   0 cpc   :   0
cpi   :   0 cpse  :   0 dec   :   3 des   :   0 eor   :   1 fmul  :   0
fmuls :   0 fmulsu:   0 icall :   0 ijmp  :   0 in    :   3 inc   :   2
jmp   :   5 ld    :   1 ldd   :   0 ldi   :  24 lds   :   3 lpm   :   2
lsl   :   1 lsr   :   1 mov   :  13 movw  :   0 mul   :   0 muls  :   0
mulsu :   0 neg   :   0 nop   :   1 or    :   0 ori   :   0 out   :   9
pop   :  12 push  :  12 rcall :   7 ret   :   4 reti  :   1 rjmp  :   2
rol   :   0 ror   :   0 sbc   :   0 sbci  :   0 sbi   :   0 sbic  :   0
sbis  :   0 sbiw  :   0 sbr   :   0 sbrc  :   0 sbrs  :   0 sec   :   0
seh   :   0 sei   :   2 sen   :   0 ser   :   0 ses   :   0 set   :   0
sev   :   0 sez   :   0 sleep :   0 spm   :   0 st    :   4 std   :   0
sts   :  13 sub   :   0 subi  :   0 swap  :   2 tst   :   0 wdr   :   0

Instructions used: 30 out of 114 (26.3%)

"ATmega4809" memory use summary [bytes]:
Segment   Begin    End      Code   Data   Used    Size   Use%
---------------------------------------------------------------
[.cseg] 0x000000 0x0001a6    326     16    342   49152   0.7%
[.dseg] 0x002800 0x002805      0      5      5    6144   0.1%
[.eseg] 0x000000 0x000000      0      0      0     256   0.0%

Assembly complete, 0 errors, 0 warnings
```