**Casey Sobecks**

**ECE 362**

**Pre-Lab #7**

**Introduction:**

The purpose of this lab is to learn to use interrupts. The interrupts will be used to perform several tasks at the same time. Along with this, the linker file will be altered in order to handle multiple entries in a vector table.

**Lab 7.1:**

Objective/Purpose:

The purpose of this lab is to create a code that uses the Pulse width modulation to control the DC motor speed.

Expected Results:

When this program is complete, the DC motor should turn faster when the PWM button is pressed for longer. The shorter it is pressed, the slower it will run.

Code:

XDEF Entry

XREF \_\_SEG\_END\_SSTACK

Variables: Section

Portt equ $240

Tddr equ $242

Ton ds.b 1

Toff ds.b 1

sequence dc.b $70, $B0, $D0, $E0

port\_u equ $268

lookup dc.b $07, $0B, $0D, $0E

padval dc.b $0, $1, $2, $3, $4, $5, $6, $7, $8, $9, $a, $b, $C, $D, $E, $F

keypadval dc.b $eb, $77, $7b, $7d, $b7, $bb, $bd, $d7, $db, $dd, $e7, $ed, $7e, $be, $de, $ee

Code: Section

Entry:

LDS #\_\_SEG\_END\_SSTACK ;You should comment your code

Movb #%00001000 tddr

JSR Delay

Delay: ldy #4000

Cpy #0

Bne

RTS

movb #$F0, $26A;initialize the output pins of the hex pad

movb #$F0, $26D;set the polarity register as pull up devices

movb #$0F, $26c;activate the pull up devices on pins 0-3

lds #\_\_SEG\_END\_SSTACK;load the stack

reset: ldx #$0;load accumulator x with the zero value

again: ldaa sequence,x;load the sequence value

inx;increment x

cpx #$5;check if sequence is done

beq reset;reset if sequence is done

staa port\_u;load value in port u to a

ldaa port\_u;load the value of port u to a

cmpa #$0F;compare the value with the no output value

beq again;if no output, run again

ldaa port\_u;if output, load hex pad value again

Woot: JSR Keypad ;jump to keypad

Ohbb: Cmpa #0

Beq checkoff

Bne running

Checkoff: cmpb#0

Beq woot

Bne continue

Running:

Movb #%00001000 portt

JSR Delay

Ldaa ton

Suba 1

Staa ton

Bra ohbb

Continue:

Movb #%00000000 portt

JSR Delay

Ldab toff

Subb 1

Stab toff

Bra checkoff

bra again;run again

Keypad:

ldy #0;load 0 to y

more: ldab keypadval,y;load the keypad value sequence to b

cpy #$10;check if sequence complete

beq back;if complete, jump to end of subroutine

cba ;compare value of port u with value of sequence

bne no;if not equal, jump to no

beq yes;if equal jump to yes

no:

iny;increment y

bra more;run again by branching to more

yes:

ldaa padval,y;load the value of the pad to a

Staa ton

Ldab #$F

Subb padval,y

Stab toff

bra back;jump to end of program

back:

RTS;return to main program

nop

**Lab 7.2:**

Objective/Purpose:

The purpose of this lab is to create a code to do the same thing as 7.1, but using interrupts instead this time.

Expected Results:

The code should run the same as lab 7.1, but no delay loops will be used to control the PWM signal.

Code:

XDEF Entry

XREF \_\_SEG\_END\_SSTACK

**Lab 7.3:**

Objective/Purpose:

The purpose of this lab is to create a wall clock code in assembly.

Expected Results:

When the code is complete, the time should be displayed in binary on the LEDs, or in bcd on the LCD

Code:

XDEF Entry

XREF \_\_SEG\_END\_SSTACK