;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* main.s \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Program written by: Akaash Chikarmane

; Date Created: 2/26/2016

; Last Modified: 2/26/2016

; Section Tuesday 2-3

; Instructor: Ramesh Yerraballi

; Lab number: 4

; Brief description of the program

; If the switch is pressed, the LED toggles at 8 Hz

; Hardware connections

; PE1 is switch input (1 means pressed, 0 means not pressed)

; PE0 is LED output (1 activates external LED on protoboard)

;Overall functionality of this system is similar to Lab 3, with three changes:

;1- initialize SysTick with RELOAD 0x00FFFFFF

;2- add a heartbeat to PF2 that toggles every time through loop

;3- add debugging dump of input, output, and time

; Operation

; 1) Make PE0 an output and make PE1 an input.

; 2) The system starts with the LED on (make PE0 =1).

; 3) Wait about 62 ms

; 4) Shift the Port E data bit 1 into bit 4 position, and leave bit 0 into bit 0 position

; 5) Steps 3 and 4 are repeated over and over

SWITCH EQU 0x40024008 ;PE1

LED EQU 0x40024004 ;PE0

SYSCTL\_RCGCGPIO\_R EQU 0x400FE608

SYSCTL\_RCGC2\_GPIOE EQU 0x00000010 ; port E Clock Gating Control

SYSCTL\_RCGC2\_GPIOF EQU 0x00000020 ; port F Clock Gating Control

GPIO\_PORTE\_DATA\_R EQU 0x400243FC

GPIO\_PORTE\_DIR\_R EQU 0x40024400

GPIO\_PORTE\_AFSEL\_R EQU 0x40024420

GPIO\_PORTE\_PUR\_R EQU 0x40024510

GPIO\_PORTE\_DEN\_R EQU 0x4002451C

GPIO\_PORTF\_DATA\_R EQU 0x400253FC

GPIO\_PORTF\_DIR\_R EQU 0x40025400

GPIO\_PORTF\_AFSEL\_R EQU 0x40025420

GPIO\_PORTF\_DEN\_R EQU 0x4002551C

NVIC\_ST\_CTRL\_R EQU 0xE000E010

NVIC\_ST\_RELOAD\_R EQU 0xE000E014

NVIC\_ST\_CURRENT\_R EQU 0xE000E018

THUMB

AREA DATA, ALIGN=4

SIZE EQU 50

;You MUST use these two buffers and two variables

;You MUST not change their names

;These names MUST be exported

EXPORT DataBuffer

EXPORT TimeBuffer

EXPORT DataPt [DATA,SIZE=4]

EXPORT TimePt [DATA,SIZE=4]

DataBuffer SPACE SIZE\*4

TimeBuffer SPACE SIZE\*4

DataPt SPACE 4

TimePt SPACE 4

ALIGN

PRESERVE8

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

THUMB

EXPORT Start

IMPORT TExaS\_Init

IMPORT SysTick\_Init

Start BL TExaS\_Init ;running at 80 MHz, scope voltmeter on PD3

;Initialization for Port E

LDR R0,=SYSCTL\_RCGCGPIO\_R

LDR R1, [R0]

ORR R1, #0x30 ;enable clock for Port E and F

STR R1, [R0]

NOP ;stabilize clock

NOP

LDR R0,=GPIO\_PORTE\_DEN\_R

LDR R1, [R0]

ORR R1, #0x03 ;enable digital I/O for PF3, PF4

STR R1, [R0]

LDR R0,=GPIO\_PORTE\_DIR\_R

LDR R1, [R0]

ORR R1, #0x01 ;PE0=output (1)

BIC R1, #0x02 ;PE1=input (0)

STR R1, [R0]

LDR R0,=GPIO\_PORTE\_AFSEL\_R

LDR R1, [R0]

BIC R1, #0x03 ;no alternate function for PF3,4

STR R1, [R0]

LDR R0,=GPIO\_PORTE\_DATA\_R

LDR R1, [R0]

ORR R1, #0x01 ;LED initially on

STR R1, [R0]

;Initialization for Port F

LDR R0,=GPIO\_PORTF\_DEN\_R

LDR R1, [R0]

ORR R1, #0x04

STR R1, [R0]

LDR R0,=GPIO\_PORTF\_DIR\_R

LDR R1, [R0]

ORR R1, #0x04

STR R1, [R0]

LDR R0,=GPIO\_PORTF\_AFSEL\_R

LDR R1, [R0]

BIC R1, #0x04

STR R1, [R0]

LDR R0,=GPIO\_PORTF\_DATA\_R

LDR R1, [R0]

BIC R1, #0x04

STR R1, [R0]

;Initialize debugging dump, including SysTick

BL Debug\_Init

CPSIE I ; TExaS voltmeter, scope runs on interrupts

loop BL Debug\_Capture

BL DELAY

;heartbeat

LDR R0,=GPIO\_PORTF\_DATA\_R

LDR R1, [R0]

MOV R2, R1 ;R2 = original data

AND R1, #0x04 ;clear all but PF2

EOR R1, #0x04 ;toggle PF2

BIC R2, #0x04

ORR R2, R1, R2

STR R2, [R0]

;input PE1 test output PE0

LDR R0,=GPIO\_PORTE\_DATA\_R

LDR R1, [R0] ;read the switch (R1 = adjusting PE0)

MOV R2, R1 ;R2 = checking PE1

MOV R3, R1 ;R3 = original data

AND R2, #0x02 ;clear all but PE1

SUBS R2, #0x02 ;branch taken if PE1 = 0

BNE SETON ;if PE1 = 0, turn on LED

AND R1, #0x01 ;clear all but PE0

EOR R1, #0x01 ;toggle PE0

BIC R3, #0x01 ;clear original PE0

ORR R3, R1, R3 ;recombine

STR R3, [R0]

B loop

SETON

ORR R3, #0x01

STR R3, [R0]

B loop

;Delay subroutine

DELAY

LDR R0,=COUNT ;start of delay subroutine (62 ms)

LDR R1, [R0]

REPEAT

SUBS R1, R1, #1

BNE REPEAT

BX LR

COUNT DCD 0x130000

;------------Debug\_Init------------

; Initializes the debugging instrument

; Input: none

; Output: none

; Modifies: none

; Note: push/pop an even number of registers so C compiler is happy

Debug\_Init

PUSH {R14}

;Initialize two pointers

LDR R1,=DataBuffer ;R0 = address of DataBuffer

LDR R0,=DataPt ;R1 = address of DataPt

STR R1, [R0] ;Store the address of the DataBuff into the memory at DataPt

LDR R1,=TimeBuffer

LDR R0,=TimePt

STR R1, [R0]

LDR R0,=countone

LDR R2,=INITFILL

LDR R1,=DataPt

LDR R1, [R1]

againinit

STR R2, [R1] ;fill DataBuffer array with 0xFFFFFFFF

ADD R1, #0x04

SUB R0, #0x01

CMP R0, #0

BNE againinit

LDR R0,=counttwo

LDR R2,=INITFILL

LDR R1,=TimePt

LDR R1, [R1]

againinittwo

STR R2, [R1]

ADD R1, #0x04

SUB R0, #0x01

CMP R0, #0

BNE againinittwo

;initialize SysTick

BL SysTick\_Init

POP {R14}

BX LR

countone EQU 0x32

counttwo EQU 0xC8

INITFILL EQU 0xFFFFFFFF

datacurrent EQU 50

timecurrent EQU 0x00

;------------Debug\_Capture------------

; Dump Port E and time into buffers

; Input: none

; Output: none

; Modifies: none

; Note: push/pop an even number of registers so C compiler is happy

Debug\_Capture

PUSH{R14}

PUSH{R0,R12}

LDR R0, =datacurrent

LDR R1, [R0] ;If we have already filled the array we return to main

SUB R1, #0x01

CMP R1, #0

BLS stop

STR R1, [R0]

LDR R1, =GPIO\_PORTE\_DATA\_R

LDR R0, [R1]

AND R2, R0, #0x02

AND R0, #0x01

LSL R2, #3

ORR R0, R0, R2

LDR R1, =DataPt

LDR R2, [R1]

STR R0, [R2]

ADD R2, #0x04

STR R2, [R1]

LDR R1, =NVIC\_ST\_CURRENT\_R

LDR R0, [R1]

LDR R2, =TimePt

LDR R3, [R2]

STR R0, [R3]

ADD R3, #0x04

STR R3, [R2]

stop

POP{R0,R12}

POP{R14}

BX LR

ALIGN ; make sure the end of this section is aligned

END ; end of file