; main.s

; Runs on any Cortex M processor

; A very simple first project implementing a random number generator

; Daniel Valvano

; May 4, 2012

; This example accompanies the book

; "Embedded Systems: Introduction to Arm Cortex M Microcontrollers",

; ISBN: 978-1469998749, Jonathan Valvano, copyright (c) 2012

; Section 3.3.10, Program 3.12

;

;Copyright 2012 by Jonathan W. Valvano, valvano@mail.utexas.edu

; You may use, edit, run or distribute this file

; as long as the above copyright notice remains

;THIS SOFTWARE IS PROVIDED "AS IS". NO WARRANTIES, WHETHER EXPRESS, IMPLIED

;OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF

;MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE.

;VALVANO SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL,

;OR CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.

;For more information about my classes, my research, and my books, see

;http://users.ece.utexas.edu/~valvano/

THUMB

;AREA DATA, ALIGN=2

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

EXPORT Start

ALIGN

;unlock 0x4C4F434B

;PF4 is SW1

;PF0 is SW2

;PF1 is RGB Red

;Enable Clock RCGCGPIO p338

;Set direction 1 is out 0 is in. GPIODIR

;DEN

; 0x3FC

;GPIO Port B (APB): 0x4000.5000

;GPIO Port E (APB): 0x4002.4000

Start

mov32 R0, #0x400FE108 ; Enable GPIO Clock

mov R1, #0x32

str R1, [R0]

mov32 R0, #0x40005000 ;Port B

;mov R1, #0x4C4F434B

mov R1, #0x0

str R1, [R0, #0x420] ;turn off alt function, may need unlock

mov R1,#0xFF

str R1,[R0,#0x400] ;Port B output 0-7

mov R1,#0xFF

str R1,[R0,#0x51C] ;Port B pin 0-7 on

;mov R1, #0x09 turn on led 0 and 3

;str R1, [R0, #0x3FC]

mov32 R2, #0x40024000 ;Port E

mov R1, #0x0

str R1, [R2, #0x420] ;turn off alt function, may need unlock

mov R1,#0x03

str R1,[R2,#0x400] ;Port E output 0-1, input 2

mov R1, #0x04

str R1, [R0,#0x510] ;pull up

mov R1,#0x07

str R1,[R2,#0x51C] ;Port E pin 0-2 on

mov32 R7, #0x40025000 ;Port F

mov32 R1, #0x4C4F434B ;Unlock code

str R1, [R7,#0x520] ;Unlocks F

mov R1, #0x11

str R1, [R7,#0x524];GPIOCR

mov R1, #0x0

str R1, [R7, #0x420] ;turn off alt function

mov R1, #0x11

str R1, [R7,#0x510] ;pull up

mov R1, #0x00

str R1, [R7, #0x400] ;Port F input 1 & 4

mov R1, #0x11

str R1, [R7,#0x51C] ;Turn on pin 0,4

mov32 R3, #0x000 ;counter

Loop

mov32 R5, #0x6cd06 ;delay counter

mvn R3, R3

str R3, [R0, #0x3FC] ;set current value of R3 to led 0-7

lsr R4, R3, #8 ;

str R4, [R2, #0x3FC]

mvn R3, R3

delay

ldr R1, [R7, #0x3FC]

mvn R1, R1

AND R1, #0x10

cmp R1, #0x10

beq stop

start

ldr R1, [R2, #0x3FC]

and R1, #0x04

cmp R1, #0x04

beq reset

sub R5, #0x1

cmp R5, #0x0

bgt delay

add R3, #0x1

mov32 R6, #0x3FF

cmp R3, R6

beq reset

b Loop

stop

ldr R1, [R2, #0x3FC]

and R1, #0x04

cmp R1, #0x04

beq reset

ldr R1, [R7, #0x3FC]

mvn R1, R1

AND R1, #0x01

cmp R1, #0x01

beq start

b stop

reset

mov R3, #0x000

b Loop

ALIGN

END