/\*\*

\* CC2511 Lab 10

\*/

.text /\* The "text" section of the executable is where code is placed. \*/

.align 2 /\* Make sure the starting address is a multiple of 2. \*/

.global main /\* Declare "main" as a global symbol (callable from other files). \*/

.syntax unified /\* Enable Thumb2 instruction set support \*/

/\* Main function \*/

main:

push {lr} /\* The return address is in the "link register" (lr). Preserve it by pushing it to the stack. \*/

/\* Write your code here \*/

/\* SIM\_SCGC5 |= 0x3e00;

address is 4004\_7000h base + 1038h offset = 4004\_8038

PORTA\_PCR2 = 0x100;

PORTC\_PCR3 = 0x100;

address is 4004\_B00C

PORTD\_PCR4 = 0x100;

GPIOC\_PDDR = (1<<3);

address 400F\_F094

GPIOD\_PDDR = (1<<4);

GPIOA\_PDDR = (1<<2);

GPIOD\_PDOR = (1<<4);

GPIOA\_PDOR = (1<<2);

GPIOC\_PDOR = (1<<3);

Address 400F\_F080

create red

GPIOC\_PDOR = 0;

\*/

//Set SIM\_SCGC5

ldr r2,= 0x40048038

ldr r5,= 0x3e00

str r5, [r2]

//Set PORTC\_PCR3

ldr r3,= 0x4004B00C

ldr r6,= 0x100

str r6, [r3]

//Set GPIOC\_PDDR

ldr r4,= 0x400FF094

ldr r6,= 0b1000

str r6, [r4]

//Set GPIOC\_PDOR address to r1, this will remain as a dedicated register for turning the LED on and off

ldr r1,= 0x400FF080

ldr r7,= 0b0

str r7, [r1]

allloop:

//r0 is a dedicated to number of loop iterations

mov r0, #1000

//Turn off

ldr r7,= 0b1000

str r7, [r1]

//Wait

offloop:

mov r2, #1000

waitoffloop:

sub r2, #1

cmp r2, #0

bne waitoffloop

sub r0, #1

cmp r0, #0

bne offloop

//Turn on

mov r0, #1000

ldr r7,= 0b0

str r7, [r1]

//Wait

onloop:

mov r2, #1000

waitonloop:

sub r2, #1

cmp r2, #0

bne waitonloop

sub r0, #1

cmp r0, #0

bne onloop

cmp r0, #250

bne allloop

pop {pc} /\* Return from this function, by popping the return address into the program counter (pc) \*/