;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; semi-colon = comments

; LED BLINK program in Assembly

; experimenter board RAM at 0x1100 - 0x30ff, FLASH at 0x3100 - ;

; Port 2 is used for the LED's Port 2 bit 2 is the green LED,

; Port 2 bit 1 is the yellow LED

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;----------------------------------------------------------------------

; must include the C header to get the predefined variable names

; .cdecls assembler directive tell the assembler to include C file

; headers

;----------------------------------------------------------------------

.cdecls C,LIST,"msp430fg4618.h" ; cdecls tells assembler ;

; to allow

; the c header file

;----------------------------------------------------------------------

; Main Code

;----------------------------------------------------------------------

; Need to tell the assembler where to put the assembly code Since

; there is FLASH, and RAM on the MSP430 need to tell the assembler

; where constants, variable and data goes. The .text directive tell

; the assembler what follows is assembly instructions and to be placed

; in program flash. The .sect directive can also be used to put

; variables in system memory or RAM (.sect “.sysmem”). The

; definitions of the various sections of memory is given in \*.CMD

; file included with the project. RAM in the MSP430FG4618 begins at

; 0x1100 and program flash begins at 0x3100. The system RAM is also

; mirrored starting 0x200. For example, 0x300 is the same as 0x1200.

.text ; program start

; Tell the assembler that the label name START is a global label.

; Please note the underscore in front of the laber START. This label

; name must match the label name of the first line of code.

.global \_START ; define entry point

;----------------------------------------------------------------------

; Must initialize the stack pointer to RAM

START mov.w #0x300,SP

; Initialize '0x1200 or

; 0x300 stackpointer

; Turn off the watchdog time so the program can run indefinitely

StopWDT mov.w #WDTPW+WDTHOLD,&WDTCTL

; Stop WDT

; Set port 2 bit 2 direction so thatP2.2 is an output 1 = output

; 0 = input

SetupP1 bis.b #011b,&P2DIR

; P2.2 direction = 1

; Set port P2.2 to a one to turn on the LED

Mainloop xor.b #0x06,&P2OUT

; Toggle P2.2

; Move the value of 0xA000 into register 7 so to create a delay

mov.w #0xA000,R7

; Delay with a loop is not

; the best way; interrupts

; are better Given in Lab 6

; Decrement register 7 until it’s zero. Stay in this loop until

; register 7 is zero

;On bic.b #0x04, &P2OUT

L1 dec.w R7

; Decrement R7

jnz L1 ; Delay over?

; Let’s run the program forever

jmp Mainloop ; Again

;

;----------------------------------------------------------------------

; Interrupt Vectors

;----------------------------------------------------------------------

; Need to load the MSP430fg4618 reset vector with the address of the

; location of the first line of assembly instructions to be executed.

; The .sect assembler directive does this

.sect ".reset" ; MSP430 RESET Vector

.short START ;

; The end assembler directive tell the assembler end of source code to

; be assembled

.end