```
; Pulse.s
; Routine for creating a pulse train using interrupts
; This uses Channel 0, and a 1MHz Timer Clock (_TAPR = 15 )
; Uses Timer0A to create pulse train on PF2

;Nested Vector Interrupt Controller registers
NVIC_EN0_INT19      EQU 0x00080000 ; Interrupt 19 enable
NVIC_EN0            EQU 0xE000E100 ; IRQ 0 to 31 Set Enable Register
NVIC_PRI4           EQU 0xE000E410 ; IRQ 16 to 19 Priority Register

; 16/32 Timer Registers
TIMER0_CFG          EQU 0x40030000
TIMER0_TAMR         EQU 0x40030004
TIMER0_CTL          EQU 0x4003000C
TIMER0_IMR          EQU 0x40030018
TIMER0_RIS          EQU 0x4003001C ; Timer Interrupt Status
TIMER0_ICR          EQU 0x40030024 ; Timer Interrupt Clear
TIMER0_TAILR        EQU 0x40030028 ; Timer interval
TIMER0_TAPR         EQU 0x40030038
TIMER0_TAR          EQU 0x40030048 ; Timer register

;GPIO Registers
GPIO_PORTF_DATA     EQU 0x40025010 ; Access BIT2
GPIO_PORTF_DIR      EQU 0x40025400 ; Port Direction
GPIO_PORTF_AFSEL    EQU 0x40025420 ; Alt Function enable
GPIO_PORTF_DEN      EQU 0x4002551C ; Digital Enable
GPIO_PORTF_AMSEL    EQU 0x40025528 ; Analog enable
GPIO_PORTF_PCTL     EQU 0x4002552C ; Alternate Functions

;System Registers
SYSCTL_RCGCGPIO     EQU 0x400FE608 ; GPIO Gate Control
SYSCTL_RCGCTIMER    EQU 0x400FE604 ; GPTM Gate Control

;---------------------------------------------------
LOW                 EQU 20
HIGH                EQU 30
;---------------------------------------------------

            AREA    routines, CODE, READONLY
            THUMB
            EXPORT  My_Timer0A_Handler
            EXPORT  PULSE_INIT

;---------------------------------------------------
My_Timer0A_Handler\
            PROC
            PUSH {LR,R0-R12}
            LDR R0, =GPIO_PORTF_DATA
            LDR R1, [R0]
            EOR R1, #0x04
            STR R1, [R0]
            CMP R1, #0x04
            BNE jump
            LDR R2, =HIGH
            B   finish

jump        LDR R2, =LOW

finish      LDR R1, =TIMER0_TAILR ; initialize match clocks
            STR R2, [R1]
            LDR R0, =TIMER0_ICR ; clear interrupt flag
            MOV R1, #0x05
            STR R1, [R0]
            POP {LR,R0-R12}
            BX  LR
            ENDP
;---------------------------------------------------

PULSE_INIT  PROC
            LDR R1, =SYSCTL_RCGCGPIO ; start GPIO clock
            LDR R0, [R1]
            ORR R0, R0, #0x20 ; set bit 5 for port F
            STR R0, [R1]
            NOP ; allow clock to settle
            NOP
            NOP
            LDR R1, =GPIO_PORTF_DIR ; set direction of PF2
            LDR R0, [R1]
            ORR R0, R0, #0x04 ; set bit2 for output
            STR R0, [R1]
            LDR R1, =GPIO_PORTF_AFSEL ; regular port function
            LDR R0, [R1]
            BIC R0, R0, #0x04
            STR R0, [R1]
            LDR R1, =GPIO_PORTF_PCTL ; no alternate function
            LDR R0, [R1]
            BIC R0, R0, #0x00000F00
            STR R0, [R1]
            LDR R1, =GPIO_PORTF_AMSEL ; disable analog
            MOV R0, #0
            STR R0, [R1]
            LDR R1, =GPIO_PORTF_DEN ; enable port digital
            LDR R0, [R1]
            ORR R0, R0, #0x04
            STR R0, [R1]

            LDR R1, =SYSCTL_RCGCTIMER ; Start Timer0
            LDR R2, [R1]
            ORR R2, R2, #0x01
            STR R2, [R1]
            NOP ; allow clock to settle
            NOP
            NOP
            LDR R1, =TIMER0_CTL ; disable timer during setup LDR R2, [R1]
            BIC R2, R2, #0x01
            STR R2, [R1]
            LDR R1, =TIMER0_CFG ; set 16 bit mode
            MOV R2, #0x04
            STR R2, [R1]
            LDR R1, =TIMER0_TAMR
            MOV R2, #0x02 ; set to periodic, count down
            STR R2, [R1]
            LDR R1, =TIMER0_TAILR ; initialize match clocks
            LDR R2, =LOW
            STR R2, [R1]
            LDR R1, =TIMER0_TAPR
            MOV R2, #15 ; divide clock by 16 to
            STR R2, [R1] ; get 1us clocks
            LDR R1, =TIMER0_IMR ; enable timeout interrupt
            MOV R2, #0x01
            STR R2, [R1]
; Configure interrupt priorities
; Timer0A is interrupt #19.
; Interrupts 16-19 are handled by NVIC register PRI4.
; Interrupt 19 is controlled by bits 31:29 of PRI4.
; set NVIC interrupt 19 to priority 2
            LDR R1, =NVIC_PRI4
            LDR R2, [R1]
            AND R2, R2, #0x00FFFFFF ; clear interrupt 19 priority
            ORR R2, R2, #0x40000000 ; set interrupt 19 priority to 2
            STR R2, [R1]
; NVIC has to be enabled
; Interrupts 0-31 are handled by NVIC register EN0
; Interrupt 19 is controlled by bit 19
; enable interrupt 19 in NVIC
            LDR R1, =NVIC_EN0
            MOVT R2, #0x08 ; set bit 19 to enable interrupt 19
            STR R2, [R1]
; Enable timer
            LDR R1, =TIMER0_CTL
            LDR R2, [R1]
            ORR R2, R2, #0x03 ; set bit0 to enable
            STR R2, [R1] ; and bit 1 to stall on debug
            BX LR ; return
            ENDP
            END
```