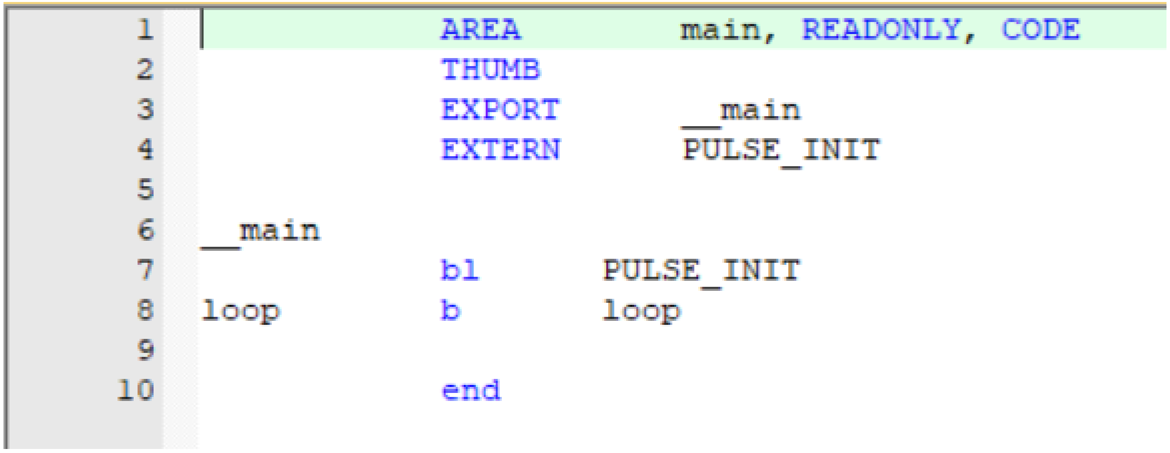
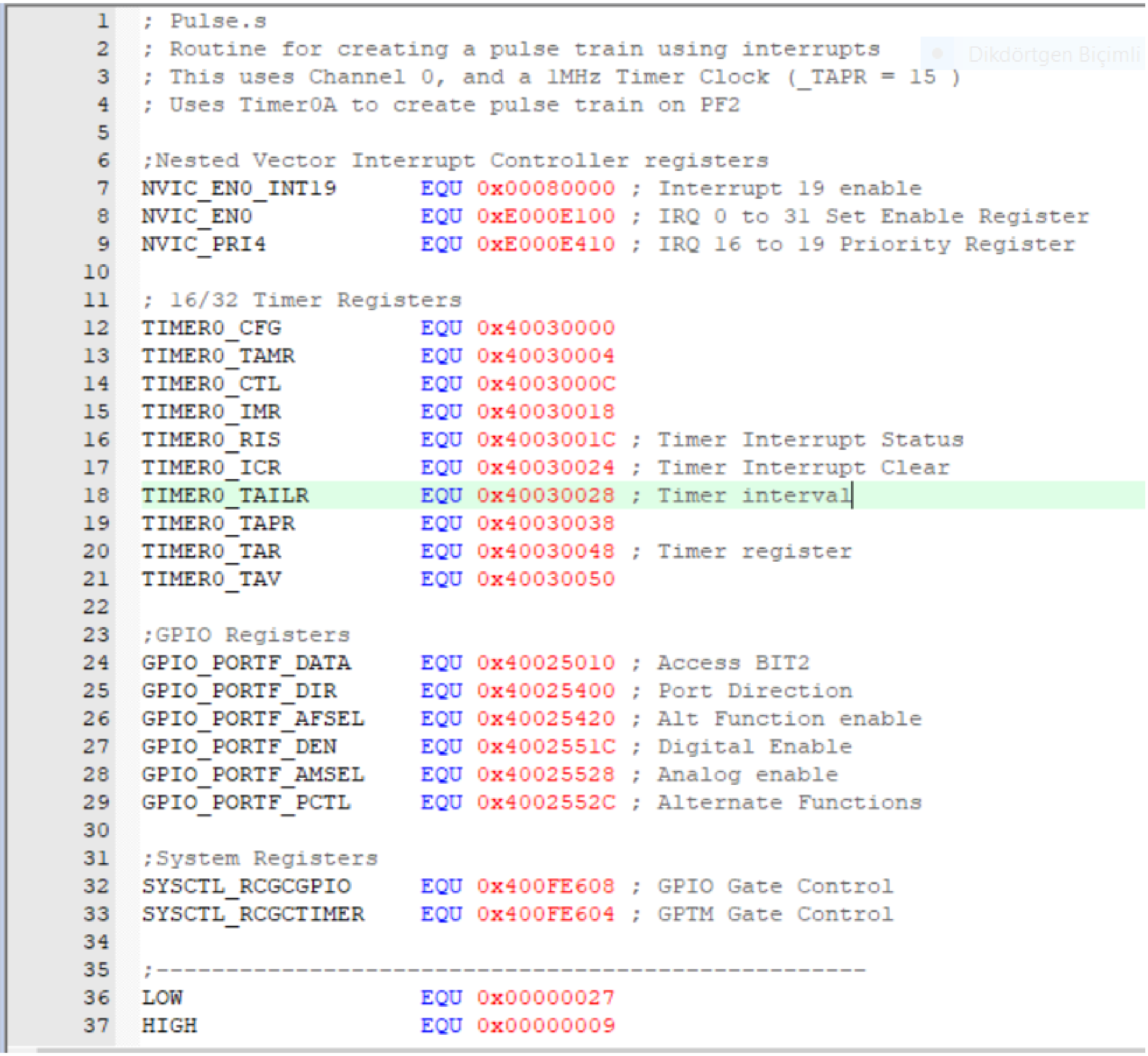
Mehmet Ataş 2304020 Preliminary Work for Exp. #4

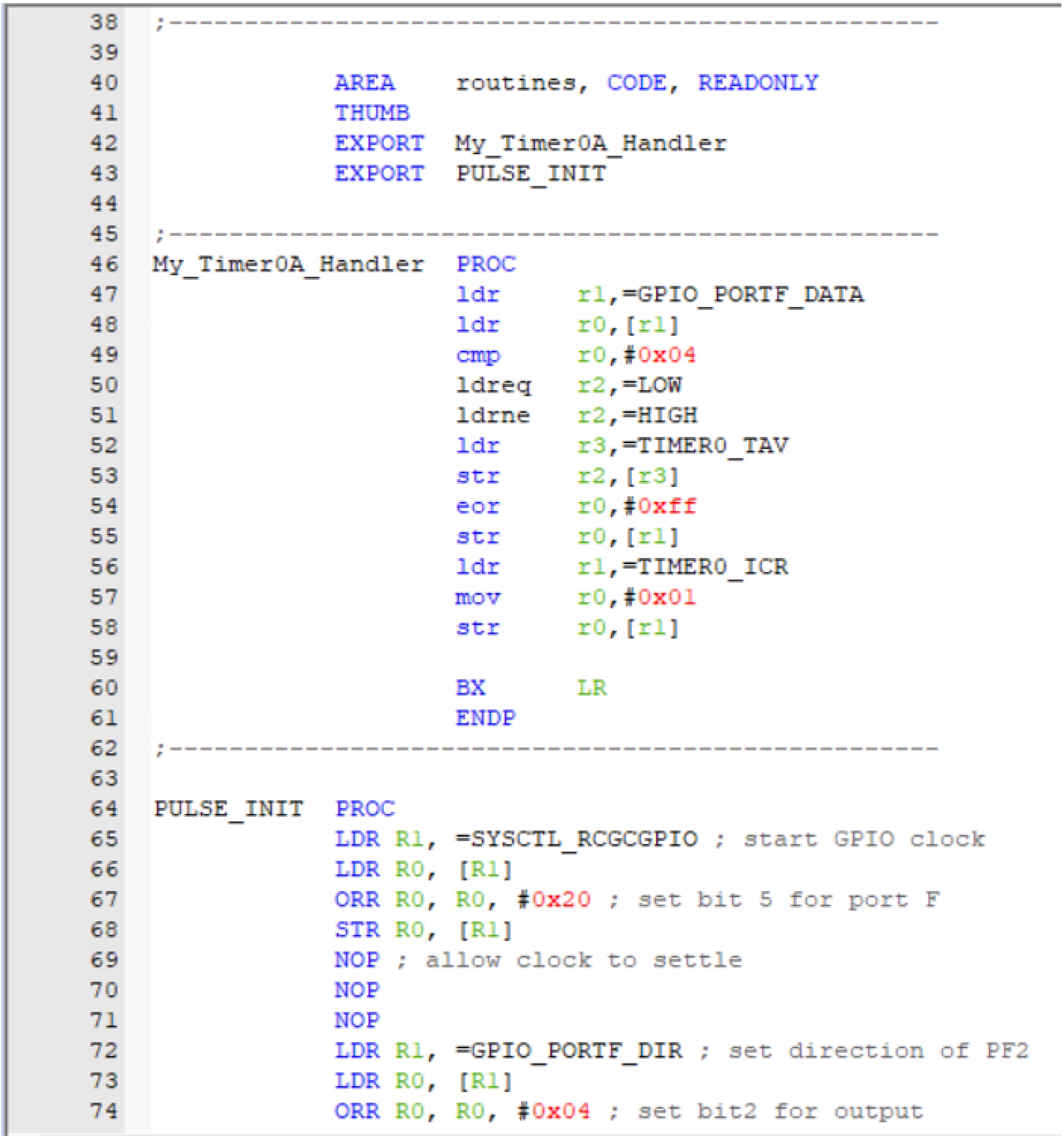
1)



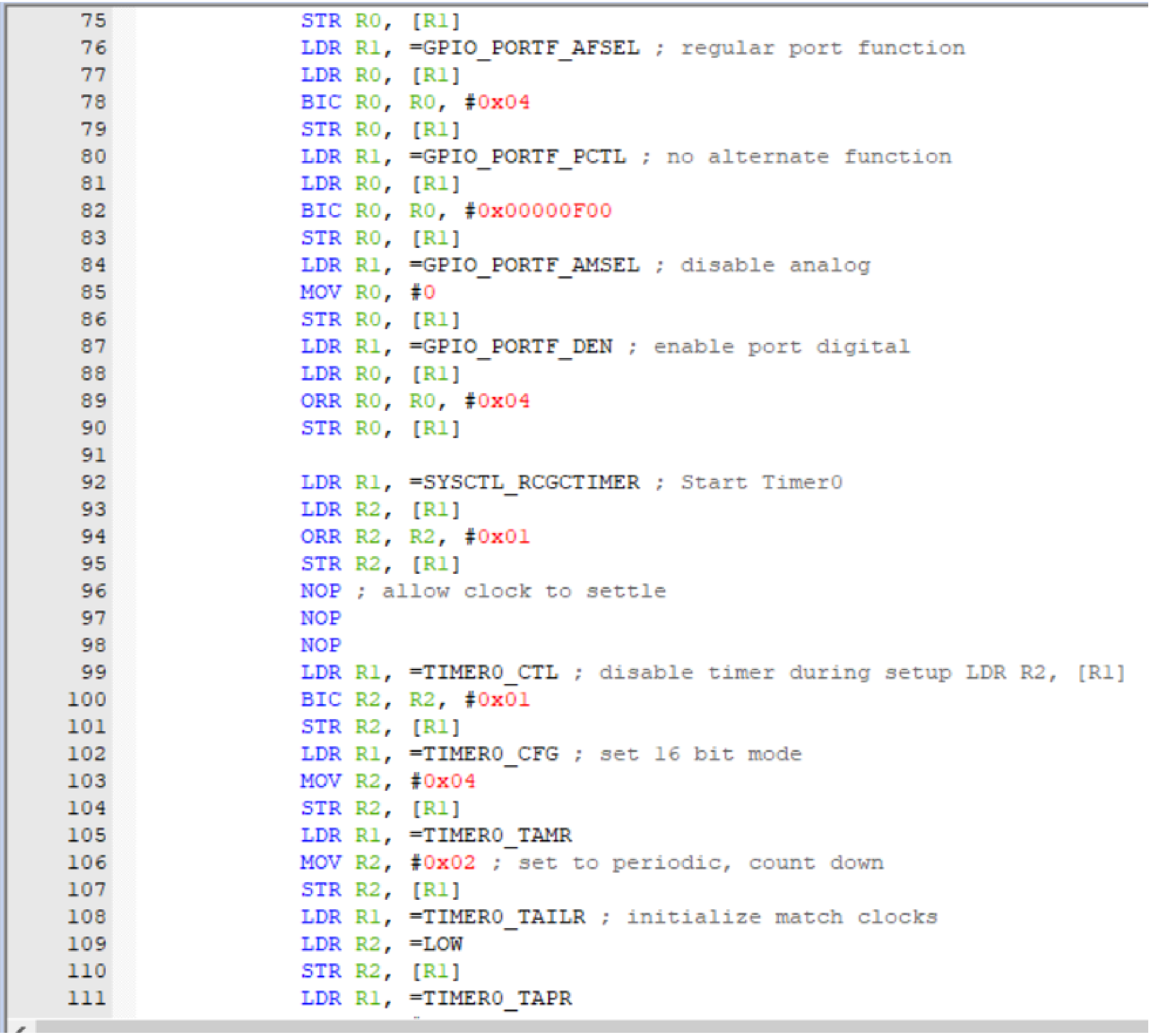
```
            AREA        main, READONLY, CODE
            THUMB
            EXPORT      __main
            EXTERN      PULSE_INIT

__main
            bl      PULSE_INIT
loop        b       loop

            end
```



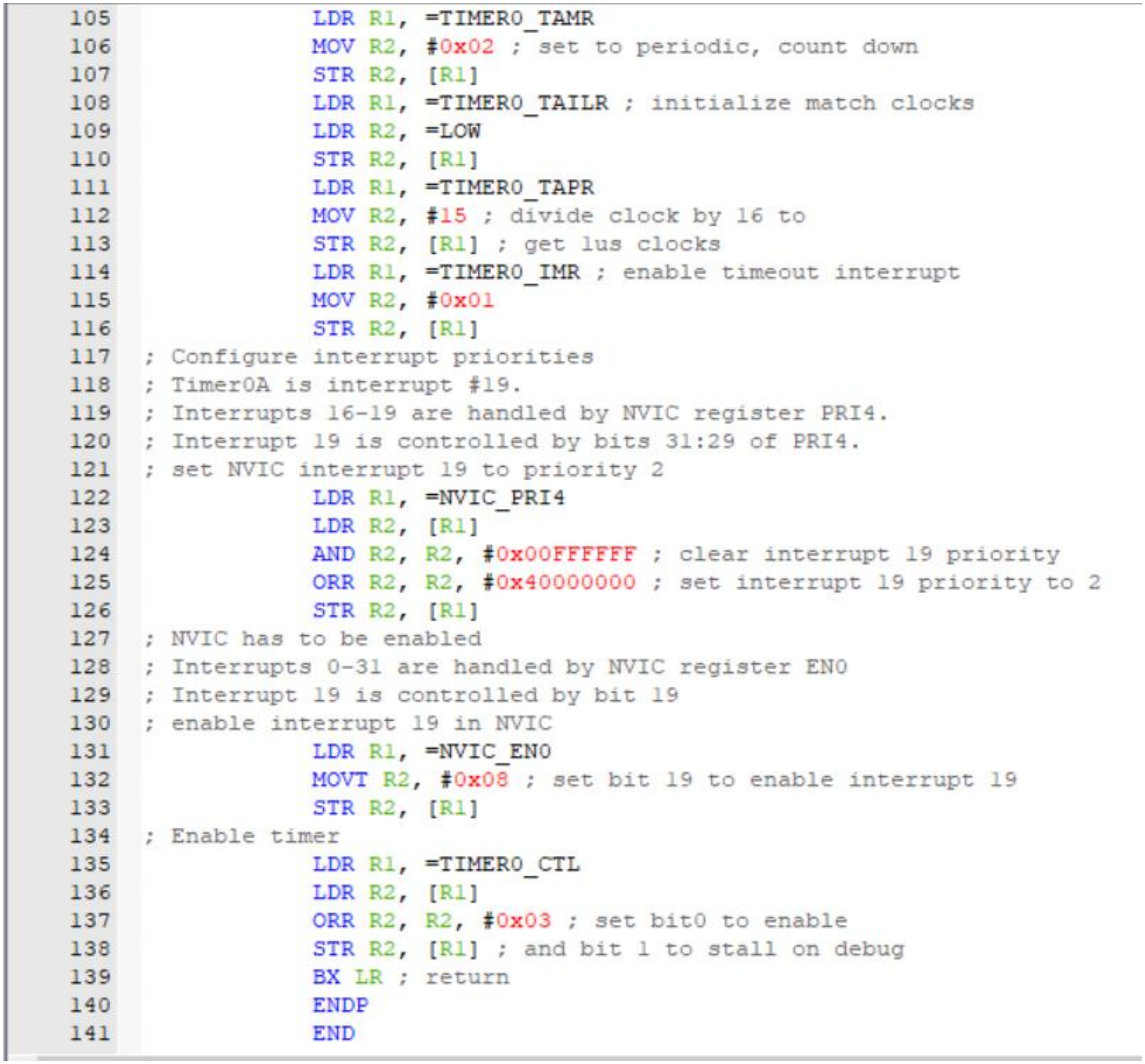
```
; Pulse.s
; Routine for creating a pulse train using interrupts
; This uses Channel 0, and a 1MHz Timer Clock (_TAPR = 15 )
; Uses Timer0A to create pulse train on PF2

;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
TIMER0_TAV          EQU 0x40030050

;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 0x00000027
HIGH                EQU 0x00000009
```



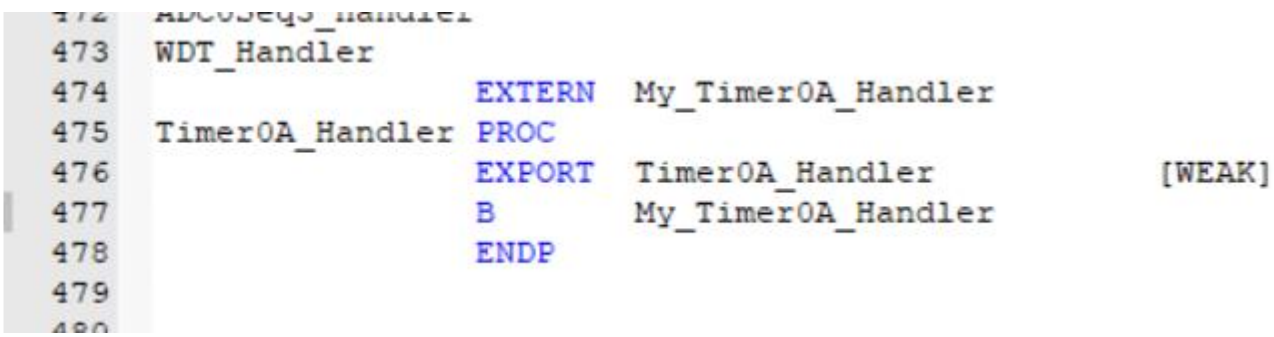
```
;---------------------------------------------------

            AREA    routines, CODE, READONLY
            THUMB
            EXPORT  My_Timer0A_Handler
            EXPORT  PULSE_INIT

;---------------------------------------------------
My_Timer0A_Handler  PROC
                    ldr     r1,=GPIO_PORTF_DATA
                    ldr     r0,[r1]
                    cmp     r0,#0x04
                    ldreq   r2,=LOW
                    ldrne   r2,=HIGH
                    ldr     r3,=TIMER0_TAV
                    str     r2,[r3]
                    eor     r0,#0xff
                    str     r0,[r1]
                    ldr     r1,=TIMER0_ICR
                    mov     r0,#0x01
                    str     r0,[r1]

                    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
```

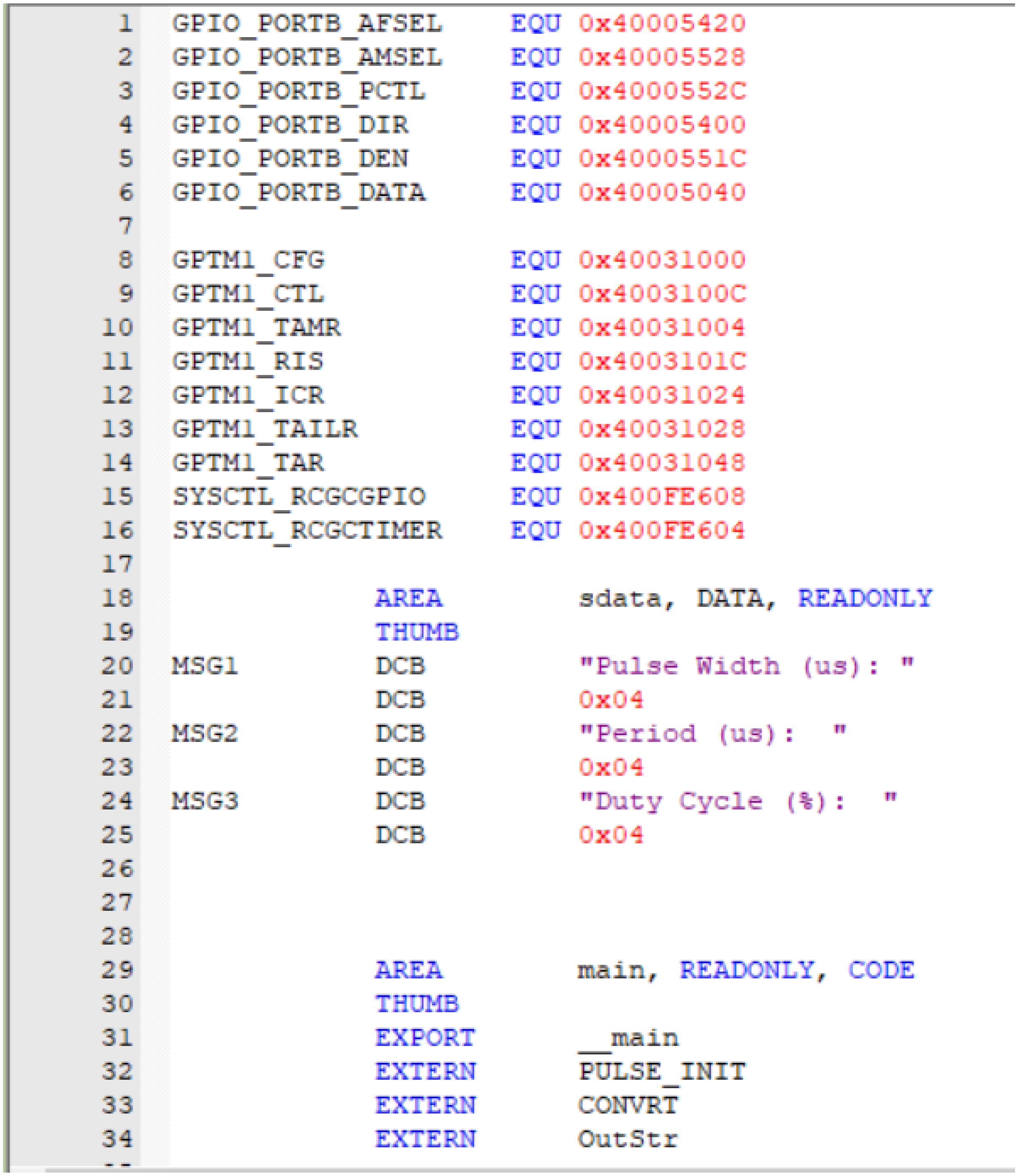


```
            LDR R1, =TIMER0_TAMR
            MOV R2, #0x02 ; set to periodic, count down
            STR R2, [R1]
            LDR R1, =TIMER0_TAILR ; initialize match clocks
            LDR R2, =LOW
            STR R2, [R1]
            LDR R1, =TIMER0_TAPR
            MOV R2, #15 ; divide clock by 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
```

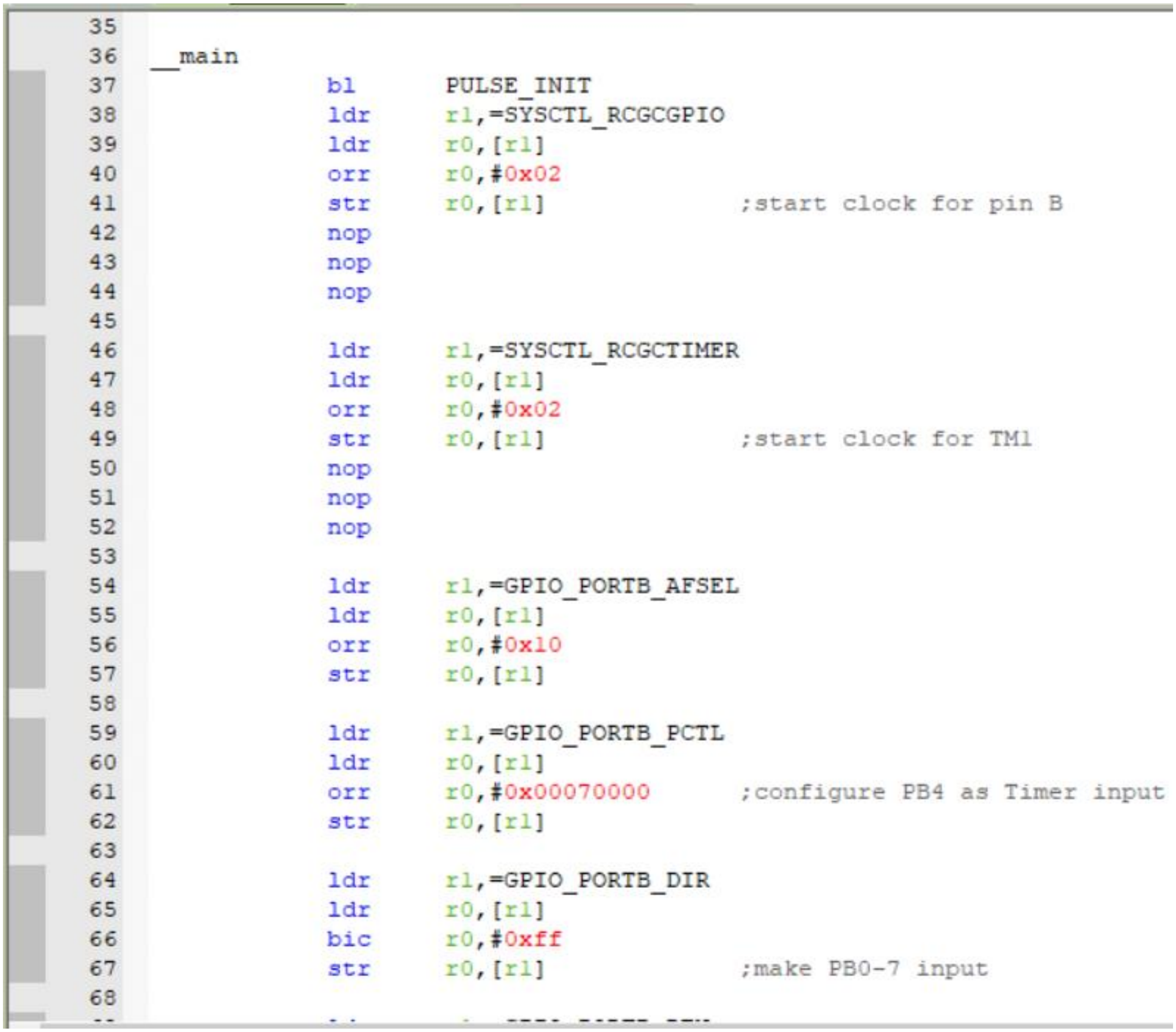


```
ADC0Seq3_Handler
WDT_Handler
                  EXTERN  My_Timer0A_Handler
Timer0A_Handler PROC
                  EXPORT  Timer0A_Handler             [WEAK]
                  B       My_Timer0A_Handler
                  ENDP

```

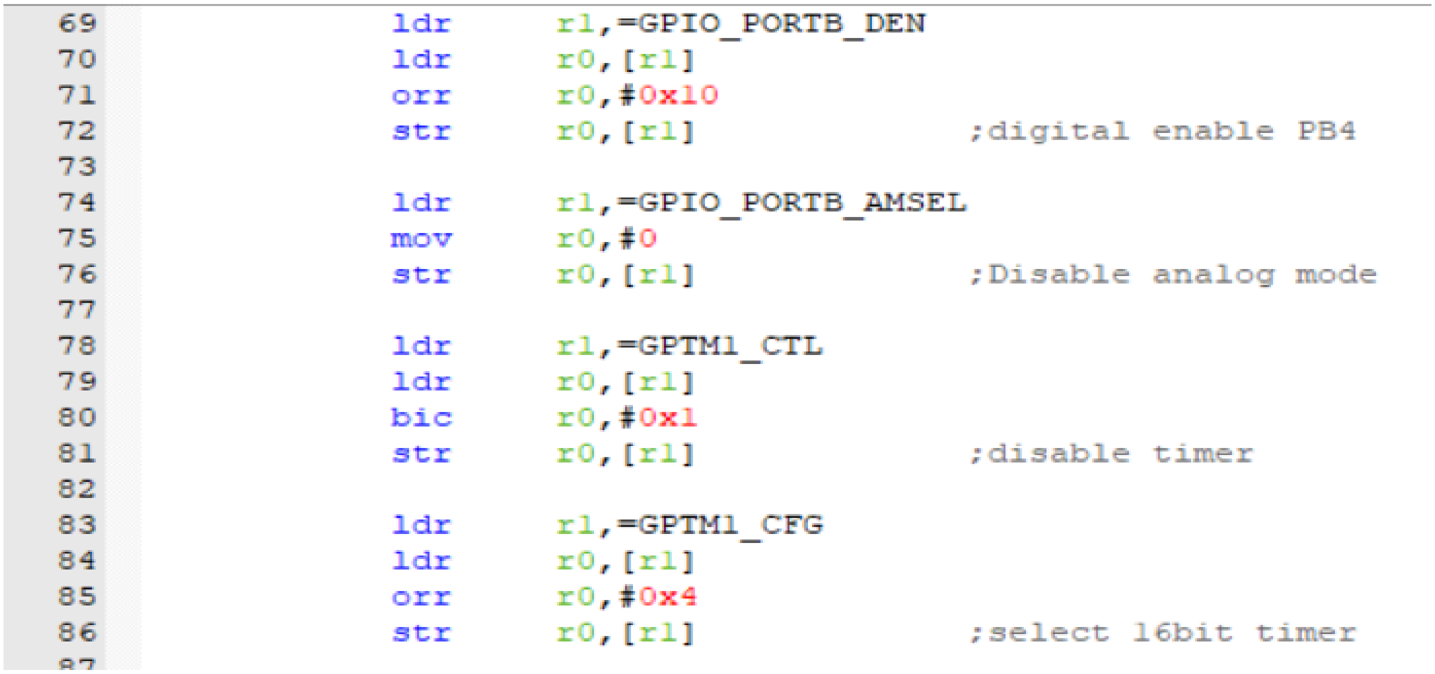
2) Timer1 and PB4 are chosen to implement this part.



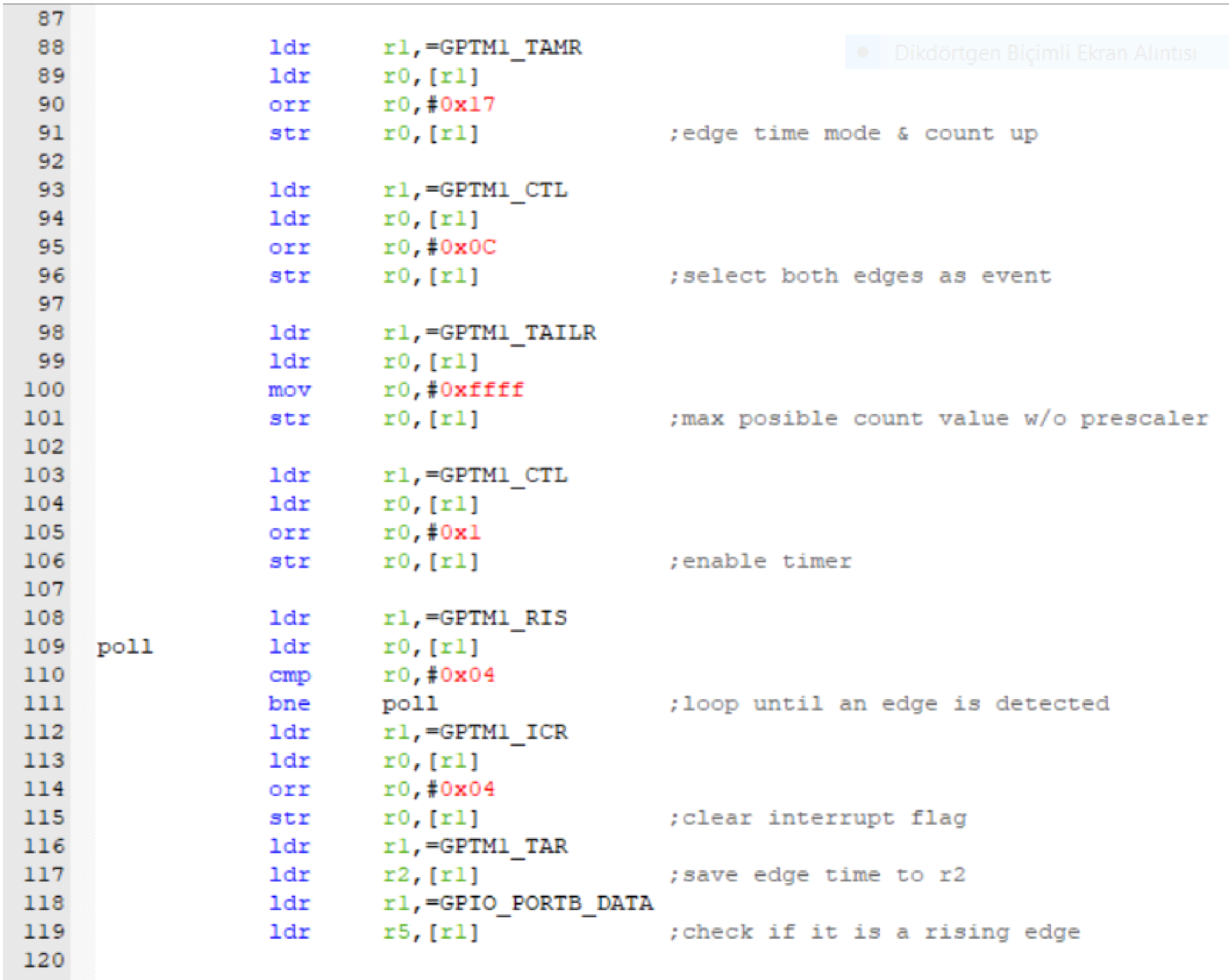
```
GPIO_PORTB_AFSEL    EQU 0x40005420
GPIO_PORTB_AMSEL    EQU 0x40005528
GPIO_PORTB_PCTL     EQU 0x4000552C
GPIO_PORTB_DIR      EQU 0x40005400
GPIO_PORTB_DEN      EQU 0x4000551C
GPIO_PORTB_DATA     EQU 0x40005040

GPTM1_CFG           EQU 0x40031000
GPTM1_CTL           EQU 0x4003100C
GPTM1_TAMR          EQU 0x40031004
GPTM1_RIS           EQU 0x4003101C
GPTM1_ICR           EQU 0x40031024
GPTM1_TAILR         EQU 0x40031028
GPTM1_TAR           EQU 0x40031048
SYSCTL_RCGCGPIO     EQU 0x400FE608
SYSCTL_RCGCTIMER    EQU 0x400FE604

            AREA        sdata, DATA, READONLY
            THUMB
MSG1        DCB         "Pulse Width (us): "
            DCB         0x04
MSG2        DCB         "Period (us):  "
            DCB         0x04
MSG3        DCB         "Duty Cycle (%):  "
            DCB         0x04



            AREA        main, READONLY, CODE
            THUMB
            EXPORT      __main
            EXTERN      PULSE_INIT
            EXTERN      CONVRT
            EXTERN      OutStr
```



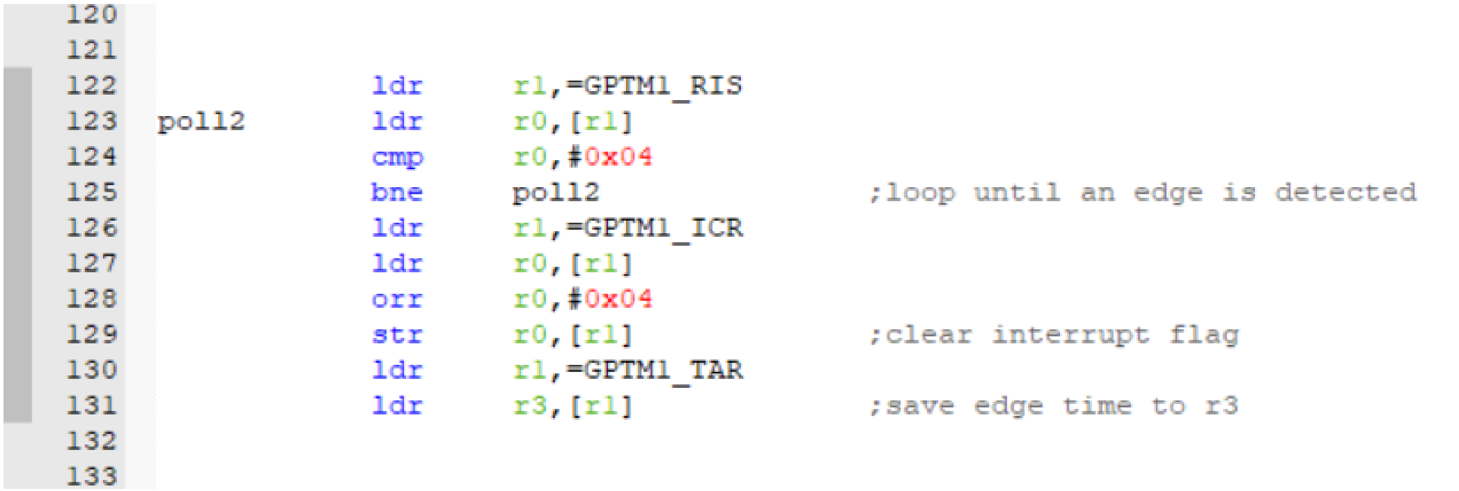
```
__main
                bl      PULSE_INIT
                ldr     r1,=SYSCTL_RCGCGPIO
                ldr     r0,[r1]
                orr     r0,#0x02
                str     r0,[r1]             ;start clock for pin B
                nop
                nop
                nop

                ldr     r1,=SYSCTL_RCGCTIMER
                ldr     r0,[r1]
                orr     r0,#0x02
                str     r0,[r1]             ;start clock for TM1
                nop
                nop
                nop

                ldr     r1,=GPIO_PORTB_AFSEL
                ldr     r0,[r1]
                orr     r0,#0x10
                str     r0,[r1]

                ldr     r1,=GPIO_PORTB_PCTL
                ldr     r0,[r1]
                orr     r0,#0x00070000      ;configure PB4 as Timer input
                str     r0,[r1]

                ldr     r1,=GPIO_PORTB_DIR
                ldr     r0,[r1]
                bic     r0,#0xff
                str     r0,[r1]             ;make PB0-7 input

```



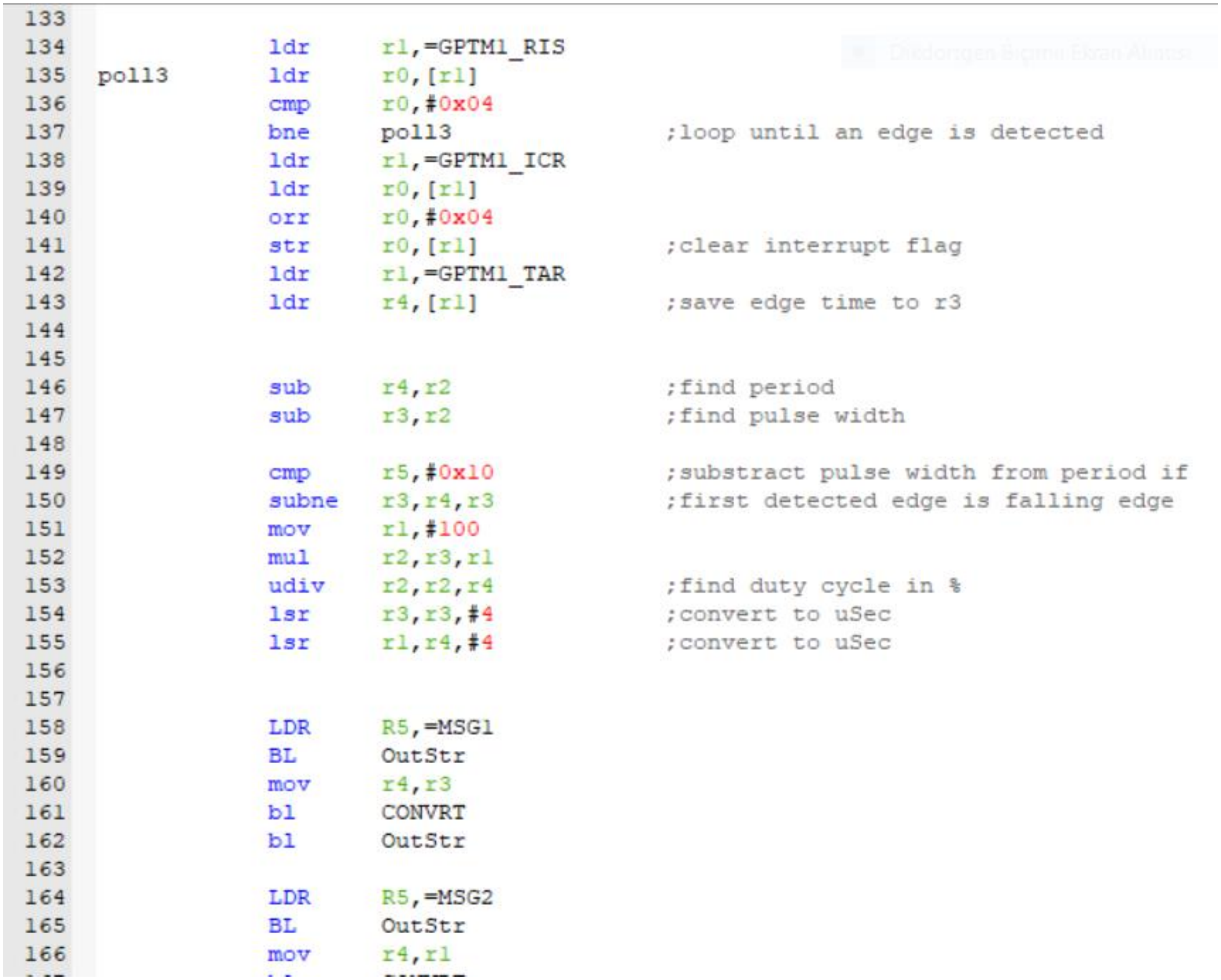
```
                ldr     r1,=GPIO_PORTB_DEN
                ldr     r0,[r1]
                orr     r0,#0x10
                str     r0,[r1]             ;digital enable PB4

                ldr     r1,=GPIO_PORTB_AMSEL
                mov     r0,#0
                str     r0,[r1]             ;Disable analog mode

                ldr     r1,=GPTM1_CTL
                ldr     r0,[r1]
                bic     r0,#0x1
                str     r0,[r1]             ;disable timer

                ldr     r1,=GPTM1_CFG
                ldr     r0,[r1]
                orr     r0,#0x4
                str     r0,[r1]             ;select 16bit timer
```



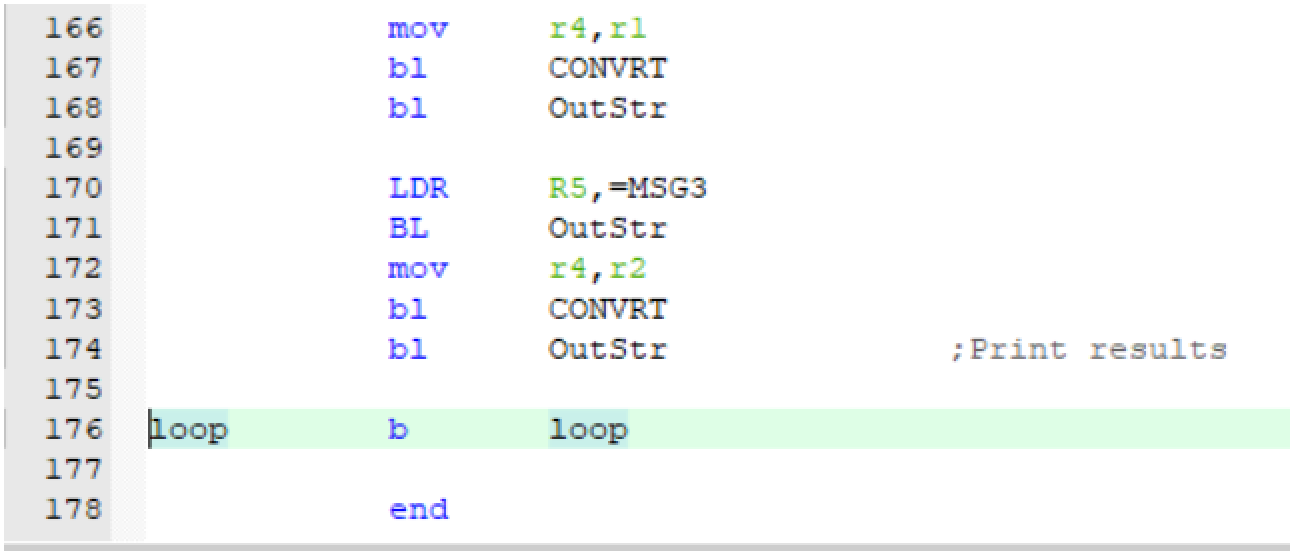
```

                ldr     r1,=GPTM1_TAMR
                ldr     r0,[r1]
                orr     r0,#0x17
                str     r0,[r1]             ;edge time mode & count up

                ldr     r1,=GPTM1_CTL
                ldr     r0,[r1]
                orr     r0,#0x0C
                str     r0,[r1]             ;select both edges as event

                ldr     r1,=GPTM1_TAILR
                ldr     r0,[r1]
                mov     r0,#0xffff
                str     r0,[r1]             ;max posible count value w/o prescaler

                ldr     r1,=GPTM1_CTL
                ldr     r0,[r1]
                orr     r0,#0x1
                str     r0,[r1]             ;enable timer

                ldr     r1,=GPTM1_RIS
poll            ldr     r0,[r1]
                cmp     r0,#0x04
                bne     poll                ;loop until an edge is detected
                ldr     r1,=GPTM1_ICR
                ldr     r0,[r1]
                orr     r0,#0x04
                str     r0,[r1]             ;clear interrupt flag
                ldr     r1,=GPTM1_TAR
                ldr     r2,[r1]             ;save edge time to r2
                ldr     r1,=GPIO_PORTB_DATA
                ldr     r5,[r1]             ;check if it is a rising edge

```



```


                ldr     r1,=GPTM1_RIS
poll2           ldr     r0,[r1]
                cmp     r0,#0x04
                bne     poll2               ;loop until an edge is detected
                ldr     r1,=GPTM1_ICR
                ldr     r0,[r1]
                orr     r0,#0x04
                str     r0,[r1]             ;clear interrupt flag
                ldr     r1,=GPTM1_TAR
                ldr     r3,[r1]             ;save edge time to r3


```

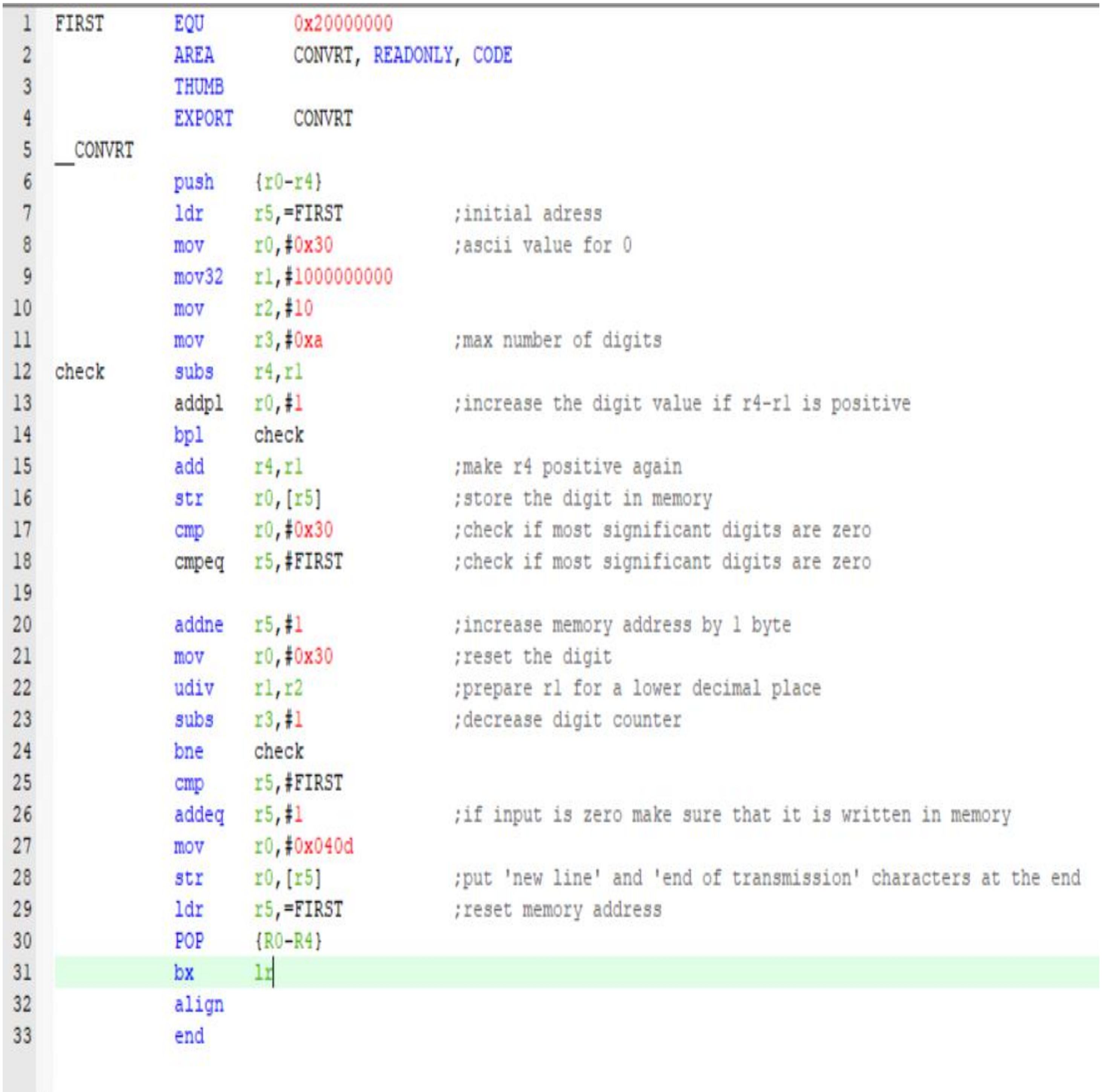


```

            ldr     r1,=GPTM1_RIS
poll3       ldr     r0,[r1]
            cmp     r0,#0x04
            bne     poll3               ;loop until an edge is detected
            ldr     r1,=GPTM1_ICR
            ldr     r0,[r1]
            orr     r0,#0x04
            str     r0,[r1]             ;clear interrupt flag
            ldr     r1,=GPTM1_TAR
            ldr     r4,[r1]             ;save edge time to r3


            sub     r4,r2               ;find period
            sub     r3,r2               ;find pulse width

            cmp     r5,#0x10            ;substract pulse width from period if
            subne   r3,r4,r3            ;first detected edge is falling edge
            mov     r1,#100
            mul     r2,r3,r1
            udiv    r2,r2,r4            ;find duty cycle in %
            lsr     r3,r3,#4            ;convert to uSec
            lsr     r1,r4,#4            ;convert to uSec


            LDR     R5,=MSG1
            BL      OutStr
            mov     r4,r3
            bl      CONVRT
            bl      OutStr

            LDR     R5,=MSG2
            BL      OutStr
            mov     r4,r1
```



```
            mov     r4,r1
            bl      CONVRT
            bl      OutStr

            LDR     R5,=MSG3
            BL      OutStr
            mov     r4,r2
            bl      CONVRT
            bl      OutStr              ;Print results

loop        b       loop

            end
```

CONVRT:

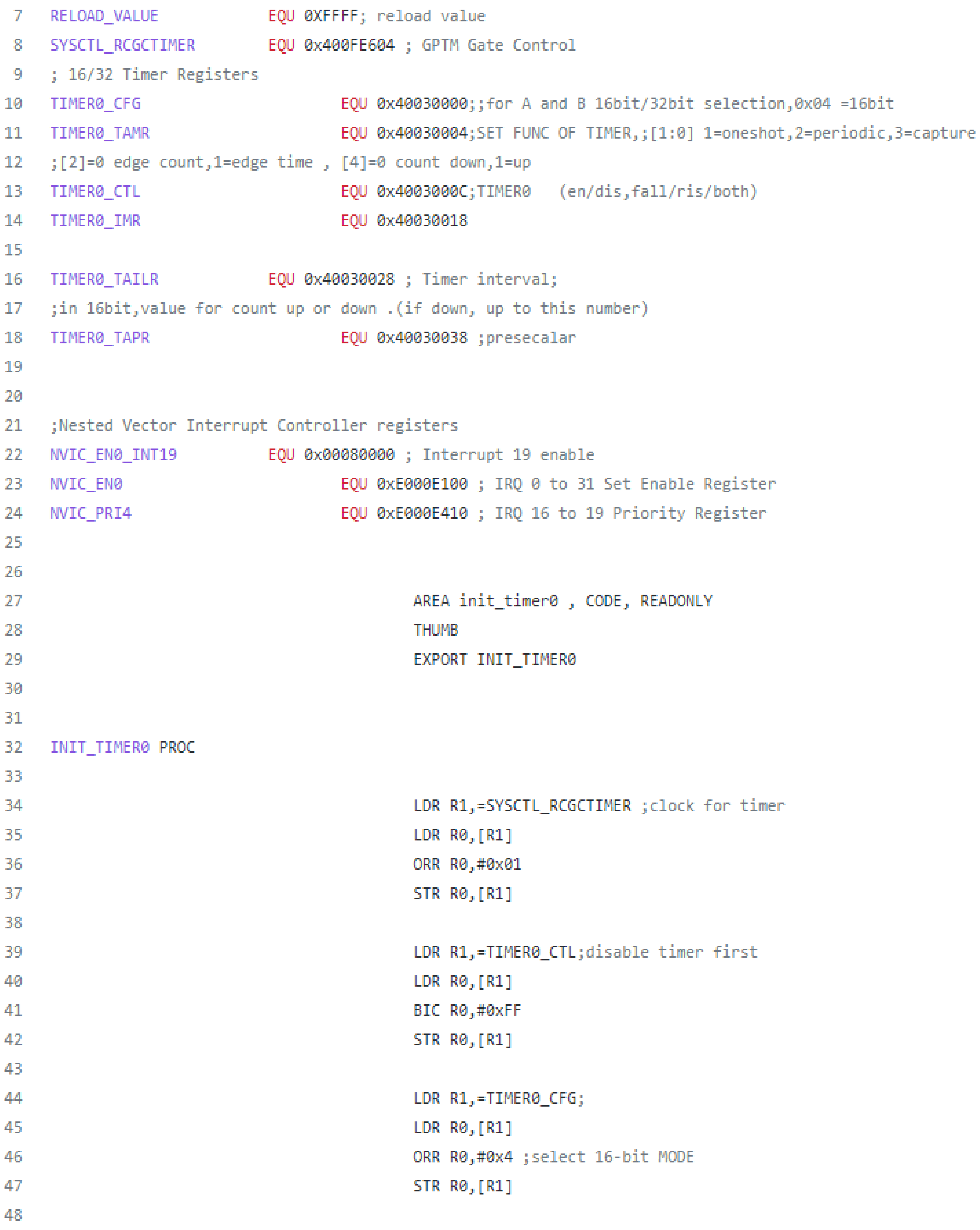


```
FIRST       EQU         0x20000000
            AREA        CONVRT, READONLY, CODE
            THUMB
            EXPORT      CONVRT
__CONVRT
            push    {r0-r4}
            ldr     r5,=FIRST           ;initial adress
            mov     r0,#0x30            ;ascii value for 0
            mov32   r1,#1000000000
            mov     r2,#10
            mov     r3,#0xa             ;max number of digits
check       subs    r4,r1
            addpl   r0,#1               ;increase the digit value if r4-r1 is positive
            bpl     check
            add     r4,r1               ;make r4 positive again
            str     r0,[r5]             ;store the digit in memory
            cmp     r0,#0x30            ;check if most significant digits are zero
            cmpeq   r5,#FIRST           ;check if most significant digits are zero

            addne   r5,#1               ;increase memory address by 1 byte
            mov     r0,#0x30            ;reset the digit
            udiv    r1,r2               ;prepare r1 for a lower decimal place
            subs    r3,#1               ;decrease digit counter
            bne     check
            cmp     r5,#FIRST
            addeq   r5,#1               ;if input is zero make sure that it is written in memory
            mov     r0,#0x040d
            str     r0,[r5]             ;put 'new line' and 'end of transmission' characters at the end
            ldr     r5,=FIRST           ;reset memory address
            POP     {R0-R4}
            bx      lr
            align
            end
```

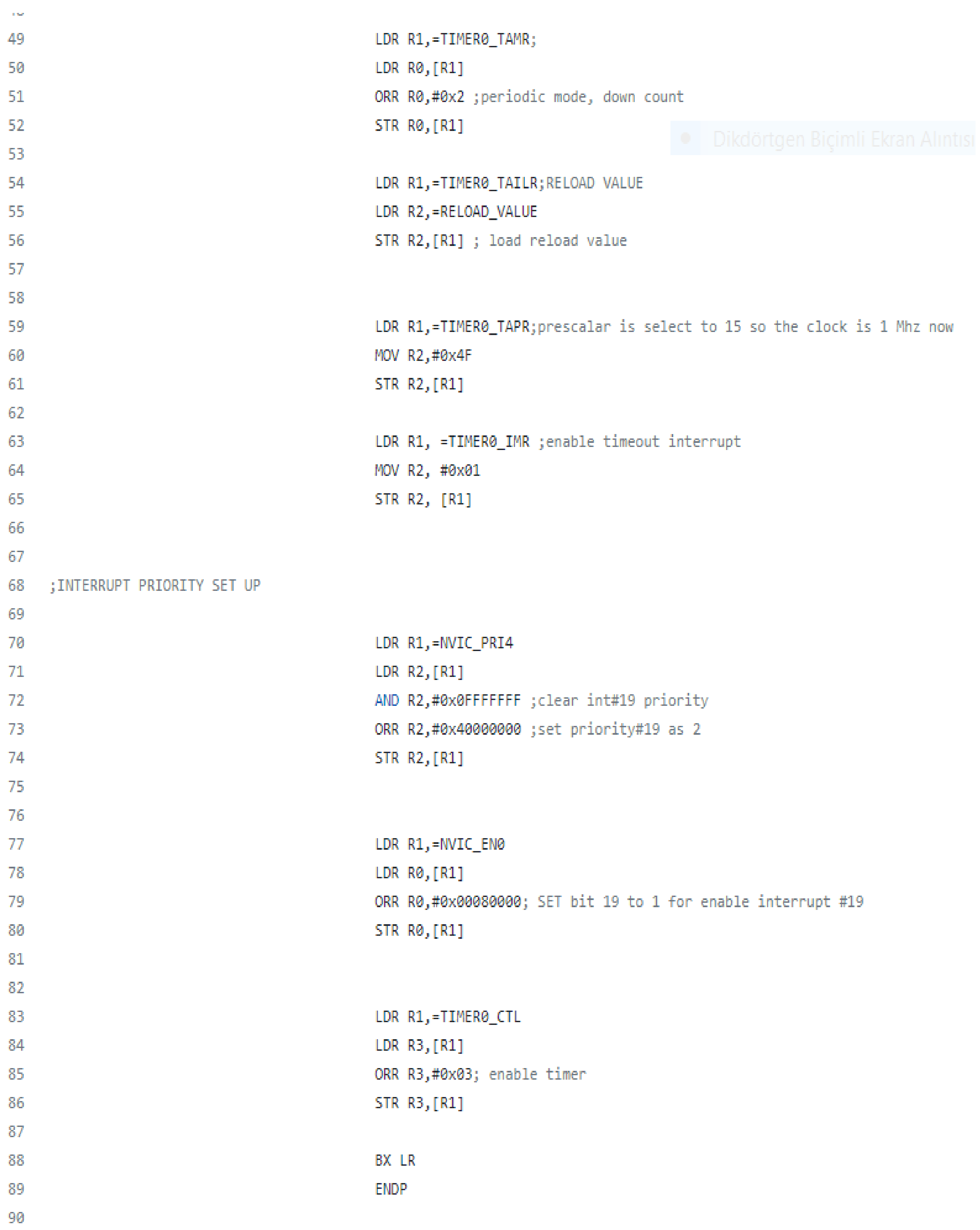
INIT_GPIO.s

```
GPIO_PORTB_DIR          EQU 0x40005400;GPIO direction register
GPIO_PORTB_AFSEL        EQU 0x40005420;GPIO alternate function select
GPIO_PORTB_DEN          EQU 0x4000551C;GPIO Digital enable regisgter
GPIO_PORTB_AMSEL        EQU 0x40005528;
GPIO_PORTB_PCTL         EQU 0x4000552C;pctl register for timer function selection
IOB                             EQU 0x00
SYSCTL_RCGCGPIO         EQU 0x400FE608;run clock gate for gpio


                                        AREA init_gpio , CODE, READONLY
                                        THUMB
                                        EXPORT INIT_GPIO
                                        EXPORT GPIO_PORTF_INIT

INIT_GPIO  PROC
;--------------------------------GPIO INIT BEGIN------------------------------
;done in also previous video
                                        LDR R1,=SYSCTL_RCGCGPIO
                                        LDR R0,[R1]
                                        ORR R0,#0x2;enable port B
                                        STR R0,[R1]
                                        NOP
                                        NOP
                                        NOP

                                        LDR R1,=GPIO_PORTB_DIR
                                        LDR R0,[R1]
                                        BIC R0,#0xFF
                                        ORR R0,#0x20;set PB6 as OUTPUT
                                        STR R0,[R1]

                                        LDR R1,=GPIO_PORTB_DEN
                                        LDR R0,[R1]
                                        BIC R0,#0xFF
                                        ORR R0,#0x01; digital enabled
                                        STR R0,[R1]

                                        LDR R1, =GPIO_PORTB_AFSEL
                                        LDR R0,[R1]
                                        BIC R0,#0xFF
                                        ORR R0,#0x40 ;set the pin6 as alternative function
                                        STR R0,[R1]

                                        LDR R1,=GPIO_PORTB_PCTL
                                        LDR R0,[R1]
                                        BIC R0,#0xFF
                                        ORR R0,#0x07000000;for PB6, 7 means it is TIMER on the tabel!
                                        STR R0,[R1]

                                        BX LR
                                        ENDP
;---------------------------------GPIO INIT END-------------------------------
;done in also previous video


;Port F base address is : 0x40025000
GPIO_PORTF_DIR                  EQU     0x40025400
PORT_F_Pin_OUTPUTS              EQU     0x0E ; 0000_1110 PF1,PF2,PF3 are outputs
GPIO_PORTF_AFSEL                EQU 0x40025420 ;PORTF AFSEL
GPIO_PORTF_DEN                  EQU 0x4002551C ;portf den
GPIO_PORTF_DATA                 EQU     0x40025038; DATA REGISTER

GPIO_PORTF_INIT PROC

                                        LDR R1,=SYSCTL_RCGCGPIO
                                        LDR R0,[R1]
                                        ORR R0,R0,#0x20 ;enable port F ,disable rest of the ports
                                        STR R0,[R1]
                                        NOP
                                        NOP
                                        NOP ; let the GPIO clock stabilize

                                        LDR R1,=GPIO_PORTF_DIR
                                        LDR R0,[R1]
                                        BIC R0,#0xFF ;clear pins 0-7
                                        ORR R0,#PORT_F_Pin_OUTPUTS;set PF1 PF2 PF3 as output
                                        STR R0,[R1]

                                        LDR R1,=GPIO_PORTF_AFSEL
                                        LDR R0,[R1]
                                        BIC R0,#0xFF ;clear all pins as no alternate function
                                        STR R0,[R1]

                                        LDR R1,=GPIO_PORTF_DEN
                                        LDR R0,[R1]
                                        ORR R0,#0xFF ;SET AS ALL DIGITAN ENABLED
                                    STR R0,[R1]

                                        BX LR
                                        ENDP
```

INIT_TIMER0.s

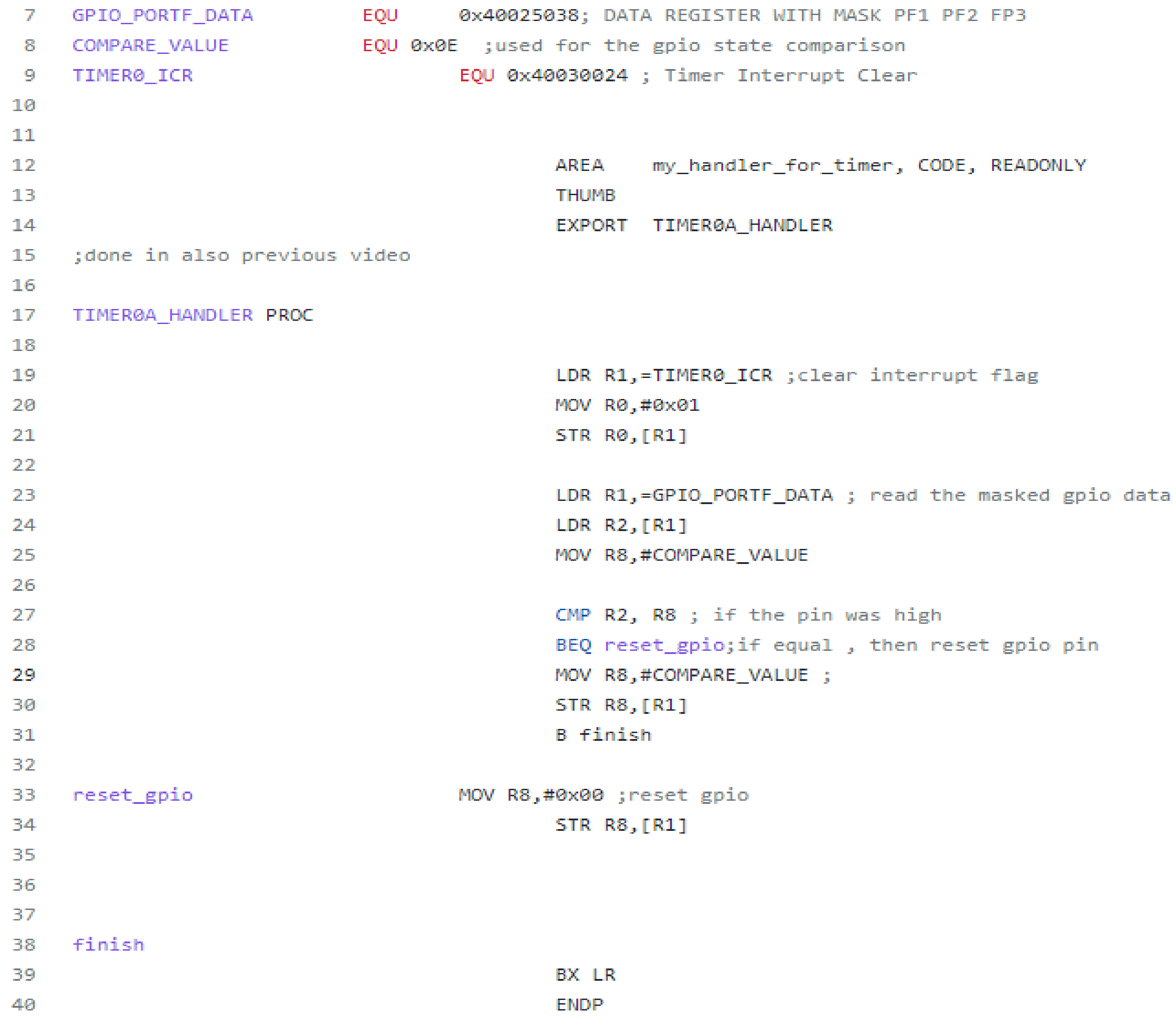


```
RELOAD_VALUE            EQU 0XFFFF; reload value
SYSCTL_RCGCTIMER        EQU 0x400FE604 ; GPTM Gate Control
; 16/32 Timer Registers
TIMER0_CFG                      EQU 0x40030000;;for A and B 16bit/32bit selection,0x04 =16bit
TIMER0_TAMR                     EQU 0x40030004;SET FUNC OF TIMER,;[1:0] 1=oneshot,2=periodic,3=capture
;[2]=0 edge count,1=edge time , [4]=0 count down,1=up
TIMER0_CTL                      EQU 0x4003000C;TIMER0   (en/dis,fall/ris/both)
TIMER0_IMR                      EQU 0x40030018

TIMER0_TAILR            EQU 0x40030028 ; Timer interval;
;in 16bit,value for count up or down .(if down, up to this number)
TIMER0_TAPR                     EQU 0x40030038 ;presecalar


;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


                                        AREA init_timer0 , CODE, READONLY
                                        THUMB
                                        EXPORT INIT_TIMER0


INIT_TIMER0 PROC

                                        LDR R1,=SYSCTL_RCGCTIMER ;clock for timer
                                        LDR R0,[R1]
                                        ORR R0,#0x01
                                        STR R0,[R1]

                                        LDR R1,=TIMER0_CTL;disable timer first
                                        LDR R0,[R1]
                                        BIC R0,#0xFF
                                        STR R0,[R1]

                                        LDR R1,=TIMER0_CFG;
                                        LDR R0,[R1]
                                        ORR R0,#0x4 ;select 16-bit MODE
                                        STR R0,[R1]

```



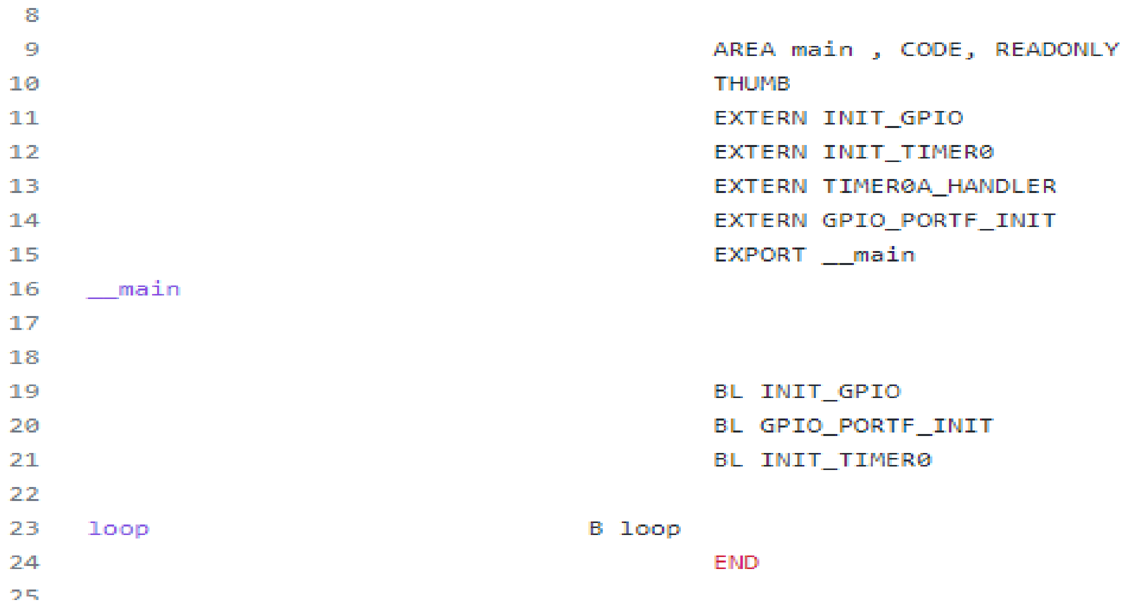
```
                LDR R1,=TIMER0_TAMR;
                LDR R0,[R1]
                ORR R0,#0x2 ;periodic mode, down count
                STR R0,[R1]

                LDR R1,=TIMER0_TAILR;RELOAD VALUE
                LDR R2,=RELOAD_VALUE
                STR R2,[R1] ; load reload value


                LDR R1,=TIMER0_TAPR;prescalar is select to 15 so the clock is 1 Mhz now
                MOV R2,#0x4F
                STR R2,[R1]

                LDR R1, =TIMER0_IMR ;enable timeout interrupt
                MOV R2, #0x01
                STR R2, [R1]


;INTERRUPT PRIORITY SET UP

                LDR R1,=NVIC_PRI4
                LDR R2,[R1]
                AND R2,#0x0FFFFFFF ;clear int#19 priority
                ORR R2,#0x40000000 ;set priority#19 as 2
                STR R2,[R1]


                LDR R1,=NVIC_EN0
                LDR R0,[R1]
                ORR R0,#0x00080000; SET bit 19 to 1 for enable interrupt #19
                STR R0,[R1]


                LDR R1,=TIMER0_CTL
                LDR R3,[R1]
                ORR R3,#0x03; enable timer
                STR R3,[R1]

                BX LR
                ENDP

```

TIMER0A_HANDLER.s



```
GPIO_PORTF_DATA         EQU     0x40025038; DATA REGISTER WITH MASK PF1 PF2 FP3
COMPARE_VALUE           EQU 0x0E  ;used for the gpio state comparison
TIMER0_ICR                      EQU 0x40030024 ; Timer Interrupt Clear


                                        AREA    my_handler_for_timer, CODE, READONLY
                                        THUMB
                                        EXPORT  TIMER0A_HANDLER
;done in also previous video

TIMER0A_HANDLER PROC

                                        LDR R1,=TIMER0_ICR ;clear interrupt flag
                                        MOV R0,#0x01
                                        STR R0,[R1]

                                        LDR R1,=GPIO_PORTF_DATA ; read the masked gpio data
                                        LDR R2,[R1]
                                        MOV R8,#COMPARE_VALUE

                                        CMP R2, R8 ; if the pin was high
                                        BEQ reset_gpio;if equal , then reset gpio pin
                                        MOV R8,#COMPARE_VALUE ;
                                        STR R8,[R1]
                                        B finish

reset_gpio                      MOV R8,#0x00 ;reset gpio
                                        STR R8,[R1]



finish
                                        BX LR
                                        ENDP
```

Main



```

                                        AREA main , CODE, READONLY
                                        THUMB
                                        EXTERN INIT_GPIO
                                        EXTERN INIT_TIMER0
                                        EXTERN TIMER0A_HANDLER
                                        EXTERN GPIO_PORTF_INIT
                                        EXPORT __main
__main


                                        BL INIT_GPIO
                                        BL GPIO_PORTF_INIT
                                        BL INIT_TIMER0

loop                            B loop
                                        END
```