

1)

1	AREA	main, READONLY, CODE
2	THUMB	
3	EXPORT	__main
4	EXTERN	PULSE_INIT
5		
6	__main	
7	bl	PULSE_INIT
8	loop	loop
9		
10	end	

1	; Pulse.s	
2	; Routine for creating a pulse train using interrupts	
3	; This uses Channel 0, and a 1MHz Timer Clock (_TAPR = 15)	
4	; Uses Timer0A to create pulse train on PF2	
5		
6	;Nested Vector Interrupt Controller registers	
7	NVIC_EN0_INT19	EQU 0x00080000 ; Interrupt 19 enable
8	NVIC_EN0	EQU 0xE000E100 ; IRQ 0 to 31 Set Enable Register
9	NVIC_PRI4	EQU 0xE000E410 ; IRQ 16 to 19 Priority Register
10		
11	; 16/32 Timer Registers	
12	TIMER0_CFG	EQU 0x40030000
13	TIMER0_TAMR	EQU 0x40030004
14	TIMER0_CTL	EQU 0x4003000C
15	TIMER0_IMR	EQU 0x40030018
16	TIMER0_RIS	EQU 0x4003001C ; Timer Interrupt Status
17	TIMER0_ICR	EQU 0x40030024 ; Timer Interrupt Clear
18	TIMER0_TAILR	EQU 0x40030028 ; Timer interval
19	TIMER0_TAPR	EQU 0x40030038
20	TIMER0_TAR	EQU 0x40030048 ; Timer register
21	TIMER0_TAV	EQU 0x40030050
22		
23	;GPIO Registers	
24	GPIO_PORTF_DATA	EQU 0x40025010 ; Access BIT2
25	GPIO_PORTF_DIR	EQU 0x40025400 ; Port Direction
26	GPIO_PORTF_AFSEL	EQU 0x40025420 ; Alt Function enable
27	GPIO_PORTF_DEN	EQU 0x4002551C ; Digital Enable
28	GPIO_PORTF_AMSEL	EQU 0x40025528 ; Analog enable
29	GPIO_PORTF_PCTL	EQU 0x4002552C ; Alternate Functions
30		
31	;System Registers	
32	SYSCTL_RCGCGPIO	EQU 0x400FE608 ; GPIO Gate Control
33	SYSCTL_RCGCTIMER	EQU 0x400FE604 ; GPTM Gate Control
34		
35	;-----	
36	LOW	EQU 0x00000027
37	HIGH	EQU 0x00000009

```

38 ;-----
39
40         AREA      routines, CODE, READONLY
41         THUMB
42         EXPORT    My_Timer0A_Handler
43         EXPORT    PULSE_INIT
44
45 ;-----
46 My_Timer0A_Handler  PROC
47                     ldr      r1,=GPIO_PORTF_DATA
48                     ldr      r0,[r1]
49                     cmp      r0,#0x04
50                     ldreq    r2,=LOW
51                     ldrne    r2,=HIGH
52                     ldr      r3,=TIMER0_TAV
53                     str      r2,[r3]
54                     eor      r0,#0xff
55                     str      r0,[r1]
56                     ldr      r1,=TIMER0_ICR
57                     mov      r0,#0x01
58                     str      r0,[r1]
59
60                     BX       LR
61                     ENDP
62 ;-----
63
64 PULSE_INIT  PROC
65             LDR R1, =SYSCTL_RCGCGPIO ; start GPIO clock
66             LDR R0, [R1]
67             ORR R0, R0, #0x20 ; set bit 5 for port F
68             STR R0, [R1]
69             NOP ; allow clock to settle
70             NOP
71             NOP
72             LDR R1, =GPIO_PORTF_DIR ; set direction of PF2
73             LDR R0, [R1]
74             ORR R0, R0, #0x04 ; set bit2 for output

```

```

75     STR R0, [R1]
76     LDR R1, =GPIO_PORTF_AFSEL ; regular port function
77     LDR R0, [R1]
78     BIC R0, R0, #0x04
79     STR R0, [R1]
80     LDR R1, =GPIO_PORTF_PCTL ; no alternate function
81     LDR R0, [R1]
82     BIC R0, R0, #0x00000F00
83     STR R0, [R1]
84     LDR R1, =GPIO_PORTF_AMSEL ; disable analog
85     MOV R0, #0
86     STR R0, [R1]
87     LDR R1, =GPIO_PORTF_DEN ; enable port digital
88     LDR R0, [R1]
89     ORR R0, R0, #0x04
90     STR R0, [R1]
91
92     LDR R1, =SYSCTL_RCGCTIMER ; Start Timer0
93     LDR R2, [R1]
94     ORR R2, R2, #0x01
95     STR R2, [R1]
96     NOP ; allow clock to settle
97     NOP
98     NOP
99     LDR R1, =TIMER0_CTL ; disable timer during setup LDR R2, [R1]
100    BIC R2, R2, #0x01
101    STR R2, [R1]
102    LDR R1, =TIMER0_CFG ; set 16 bit mode
103    MOV R2, #0x04
104    STR R2, [R1]
105    LDR R1, =TIMER0_TAMR
106    MOV R2, #0x02 ; set to periodic, count down
107    STR R2, [R1]
108    LDR R1, =TIMER0_TAILR ; initialize match clocks
109    LDR R2, =LOW
110    STR R2, [R1]
111    LDR R1, =TIMER0_TAPR

```

```

105         LDR R1, =TIMER0_TAMR
106         MOV R2, #0x02 ; set to periodic, count down
107         STR R2, [R1]
108         LDR R1, =TIMER0_TAILR ; initialize match clocks
109         LDR R2, =LOW
110         STR R2, [R1]
111         LDR R1, =TIMER0_TAPR
112         MOV R2, #15 ; divide clock by 16 to
113         STR R2, [R1] ; get 1us clocks
114         LDR R1, =TIMER0_IMR ; enable timeout interrupt
115         MOV R2, #0x01
116         STR R2, [R1]
117 ; Configure interrupt priorities
118 ; Timer0A is interrupt #19.
119 ; Interrupts 16-19 are handled by NVIC register PRI4.
120 ; Interrupt 19 is controlled by bits 31:29 of PRI4.
121 ; set NVIC interrupt 19 to priority 2
122         LDR R1, =NVIC_PRI4
123         LDR R2, [R1]
124         AND R2, R2, #0x00FFFFFF ; clear interrupt 19 priority
125         ORR R2, R2, #0x40000000 ; set interrupt 19 priority to 2
126         STR R2, [R1]
127 ; NVIC has to be enabled
128 ; Interrupts 0-31 are handled by NVIC register EN0
129 ; Interrupt 19 is controlled by bit 19
130 ; enable interrupt 19 in NVIC
131         LDR R1, =NVIC_EN0
132         MOVT R2, #0x08 ; set bit 19 to enable interrupt 19
133         STR R2, [R1]
134 ; Enable timer
135         LDR R1, =TIMER0_CTL
136         LDR R2, [R1]
137         ORR R2, R2, #0x03 ; set bit0 to enable
138         STR R2, [R1] ; and bit 1 to stall on debug
139         BX LR ; return
140     ENDP
141     END

```

```

472 ADC0Seqs_Handler
473 WDT_Handler
474
475 Timer0A_Handler EXTERN My_Timer0A_Handler
476                 PROC
477                 EXPORT Timer0A_Handler [WEAK]
478                 B My_Timer0A_Handler
479                 ENDP
480

```

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

```
1  GPIO_PORTB_AFSEL    EQU 0x40005420
2  GPIO_PORTB_AMSEL    EQU 0x40005528
3  GPIO_PORTB_PCTL     EQU 0x4000552C
4  GPIO_PORTB_DIR      EQU 0x40005400
5  GPIO_PORTB_DEN      EQU 0x4000551C
6  GPIO_PORTB_DATA     EQU 0x40005040
7
8  GPTM1_CFG           EQU 0x40031000
9  GPTM1_CTL           EQU 0x4003100C
10 GPTM1_TAMR          EQU 0x40031004
11 GPTM1_RIS           EQU 0x4003101C
12 GPTM1_ICR           EQU 0x40031024
13 GPTM1_TAILR         EQU 0x40031028
14 GPTM1_TAR           EQU 0x40031048
15 SYSCTL_RCGCGPIO     EQU 0x400FE608
16 SYSCTL_RCGCTIMER    EQU 0x400FE604
17
18                     AREA      sdata, DATA, READONLY
19                     THUMB
20 MSG1                DCB       "Pulse Width (us): "
21                     DCB       0x04
22 MSG2                DCB       "Period (us):  "
23                     DCB       0x04
24 MSG3                DCB       "Duty Cycle (%):  "
25                     DCB       0x04
26
27
28
29                     AREA      main, READONLY, CODE
30                     THUMB
31                     EXPORT    __main
32                     EXTERN    PULSE_INIT
33                     EXTERN    CONVRT
34                     EXTERN    OutStr
--
```



```

35
36  _main
37      bl      PULSE_INIT
38      ldr     r1,=SYSCTL_RCGCGPIO
39      ldr     r0,[r1]
40      orr     r0,#0x02
41      str     r0,[r1]                ;start clock for pin B
42      nop
43      nop
44      nop
45
46      ldr     r1,=SYSCTL_RCGCTIMER
47      ldr     r0,[r1]
48      orr     r0,#0x02
49      str     r0,[r1]                ;start clock for TM1
50      nop
51      nop
52      nop
53
54      ldr     r1,=GPIO_PORTB_AFSEL
55      ldr     r0,[r1]
56      orr     r0,#0x10
57      str     r0,[r1]
58
59      ldr     r1,=GPIO_PORTB_PCTL
60      ldr     r0,[r1]
61      orr     r0,#0x00070000        ;configure PB4 as Timer input
62      str     r0,[r1]
63
64      ldr     r1,=GPIO_PORTB_DIR
65      ldr     r0,[r1]
66      bic     r0,#0xff
67      str     r0,[r1]                ;make PB0-7 input
68
69      .
70      .
71      .
72      .
73      .
74      .
75      .
76      .
77      .
78      .
79      .
80      .
81      .
82      .
83      .
84      .
85      .
86      .
87      .

```

```

69      ldr     r1,=GPIO_PORTB_DEN
70      ldr     r0,[r1]
71      orr     r0,#0x10
72      str     r0,[r1]                ;digital enable PB4
73
74      ldr     r1,=GPIO_PORTB_AMSEL
75      mov     r0,#0
76      str     r0,[r1]                ;Disable analog mode
77
78      ldr     r1,=GPTM1_CTL
79      ldr     r0,[r1]
80      bic     r0,#0x1
81      str     r0,[r1]                ;disable timer
82
83      ldr     r1,=GPTM1_CFG
84      ldr     r0,[r1]
85      orr     r0,#0x4
86      str     r0,[r1]                ;select 16bit timer
87

```

```

87
88     ldr    r1,=GPTM1_TAMR
89     ldr    r0,[r1]
90     orr    r0,#0x17
91     str    r0,[r1]                ;edge time mode & count up
92
93     ldr    r1,=GPTM1_CTL
94     ldr    r0,[r1]
95     orr    r0,#0x0C
96     str    r0,[r1]                ;select both edges as event
97
98     ldr    r1,=GPTM1_TAILR
99     ldr    r0,[r1]
100    mov    r0,#0xffff
101    str    r0,[r1]                ;max posible count value w/o prescaler
102
103    ldr    r1,=GPTM1_CTL
104    ldr    r0,[r1]
105    orr    r0,#0x1
106    str    r0,[r1]                ;enable timer
107
108    ldr    r1,=GPTM1_RIS
109 poll1 ldr    r0,[r1]
110    cmp    r0,#0x04
111    bne    poll1                  ;loop until an edge is detected
112    ldr    r1,=GPTM1_ICR
113    ldr    r0,[r1]
114    orr    r0,#0x04
115    str    r0,[r1]                ;clear interrupt flag
116    ldr    r1,=GPTM1_TAR
117    ldr    r2,[r1]                ;save edge time to r2
118    ldr    r1,=GPIO_PORTB_DATA
119    ldr    r5,[r1]                ;check if it is a rising edge
120

```

```

120
121
122    ldr    r1,=GPTM1_RIS
123 poll12 ldr    r0,[r1]
124    cmp    r0,#0x04
125    bne    poll12                ;loop until an edge is detected
126    ldr    r1,=GPTM1_ICR
127    ldr    r0,[r1]
128    orr    r0,#0x04
129    str    r0,[r1]                ;clear interrupt flag
130    ldr    r1,=GPTM1_TAR
131    ldr    r3,[r1]                ;save edge time to r3
132
133

```

```

133
134      ldr      r1,=GPTM1_RIS
135 poll13  ldr      r0,[r1]
136      cmp     r0,#0x04
137      bne     poll13          ;loop until an edge is detected
138      ldr     r1,=GPTM1_ICR
139      ldr     r0,[r1]
140      orr     r0,#0x04
141      str     r0,[r1]        ;clear interrupt flag
142      ldr     r1,=GPTM1_TAR
143      ldr     r4,[r1]        ;save edge time to r3
144
145
146      sub     r4,r2          ;find period
147      sub     r3,r2          ;find pulse width
148
149      cmp     r5,#0x10      ;subtract pulse width from period if
150      subne   r3,r4,r3      ;first detected edge is falling edge
151      mov     r1,#100
152      mul     r2,r3,r1
153      udiv    r2,r2,r4      ;find duty cycle in %
154      lsr     r3,r3,#4      ;convert to uSec
155      lsr     r1,r4,#4      ;convert to uSec
156
157
158      LDR     R5,=MSG1
159      BL      OutStr
160      mov     r4,r3
161      bl      CONVRT
162      bl      OutStr
163
164      LDR     R5,=MSG2
165      BL      OutStr
166      mov     r4,r1
167
168
166      mov     r4,r1
167      bl      CONVRT
168      bl      OutStr
169
170      LDR     R5,=MSG3
171      BL      OutStr
172      mov     r4,r2
173      bl      CONVRT
174      bl      OutStr          ;Print results
175
176 loop    b      loop
177
178      end

```


CONVRT:

```

1 FIRST      EQU      0x20000000
2             AREA     CONVRT, READONLY, CODE
3             THUMB
4             EXPORT   CONVRT
5
6 _CONVRT
7             push     {r0-r4}
8             ldr      r5,=FIRST      ;initial adress
9             mov      r0,#0x30       ;ascii value for 0
10            mov32    r1,#1000000000
11            mov      r2,#10
12            mov      r3,#0xa        ;max number of digits
13 check      subs     r4,r1
14            addpl    r0,#1           ;increase the digit value if r4-r1 is positive
15            bpl      check
16            add      r4,r1           ;make r4 positive again
17            str      r0,[r5]        ;store the digit in memory
18            cmp      r0,#0x30       ;check if most significant digits are zero
19            cmpeq     r5,#FIRST      ;check if most significant digits are zero
20            addne    r5,#1           ;increase memory address by 1 byte
21            mov      r0,#0x30       ;reset the digit
22            udiv     r1,r2           ;prepare r1 for a lower decimal place
23            subs     r3,#1           ;decrease digit counter
24            bne      check
25            cmp      r5,#FIRST
26            addeq     r5,#1           ;if input is zero make sure that it is written in memory
27            mov      r0,#0x040d
28            str      r0,[r5]        ;put 'new line' and 'end of transmission' characters at the end
29            ldr      r5,=FIRST      ;reset memory address
30            POP      {R0-R4}
31            bx       lr
32            align
33            end

```

INIT_GPIO.s

```

7  GPIO_PORTB_DIR          EQU 0x40005400;GPIO direction register
8  GPIO_PORTB_AFSEL       EQU 0x40005420;GPIO alternate function select
9  GPIO_PORTB_DEN         EQU 0x4000551C;GPIO Digital enable regisgter
10 GPIO_PORTB_AMSEL       EQU 0x40005528;
11 GPIO_PORTB_PCTL        EQU 0x4000552C;pctl register for timer function selection
12 IOB                    EQU 0x00
13 SYSCTL_RCGCGPIO        EQU 0x400FE608;run clock gate for gpio
14
15
16                         AREA init_gpio , CODE, READONLY
17                         THUMB
18                         EXPORT INIT_GPIO
19                         EXPORT GPIO_PORTF_INIT
20
21 INIT_GPIO PROC
22 ;-----GPIO INIT BEGIN-----
23 ;done in also previous video
24
25                         LDR R1,=SYSCTL_RCGCGPIO
26                         LDR R0,[R1]
27                         ORR R0,#0x2;enable port B
28                         STR R0,[R1]
29                         NOP
30                         NOP
31                         NOP
32
33                         LDR R1,=GPIO_PORTB_DIR
34                         LDR R0,[R1]
35                         BIC R0,#0xFF
36                         ORR R0,#0x20;set PB6 as OUTPUT
37                         STR R0,[R1]
38
39                         LDR R1,=GPIO_PORTB_DEN
40                         LDR R0,[R1]
41                         BIC R0,#0xFF
42                         ORR R0,#0x01; digital enabled
43                         STR R0,[R1]
44
45                         LDR R1, =GPIO_PORTB_AFSEL
46                         LDR R0,[R1]
47                         BIC R0,#0xFF
48                         ORR R0,#0x40 ;set the pin6 as alternative function
49                         STR R0,[R1]
50
51

```

```

49
50         LDR R1,=GPIO_PORTB_PCTL
51         LDR R0,[R1]
52         BIC R0,#0xFF
53         ORR R0,#0x07000000;for PB6, 7 means it is TIMER on the tabel!
54         STR R0,[R1]
55
56         BX LR
57     ENDP
58 ;-----GPIO INIT END-----
59 ;done in also previous video
60
61
62 ;Port F base address is : 0x40025000
63 GPIO_PORTF_DIR            EQU    0x40025400
64 PORT_F_Pin_OUTPUTS       EQU    0x0E ; 0000_1110 PF1,PF2,PF3 are outputs
65 GPIO_PORTF_AFSEL         EQU    0x40025420 ;PORTF AFSEL
66 GPIO_PORTF_DEN           EQU    0x4002551C ;portf den
67 GPIO_PORTF_DATA          EQU    0x40025038; DATA REGISTER
68
69 GPIO_PORTF_INIT PROC
70
71         LDR R1,=SYSCTL_RCGCGPIO
72         LDR R0,[R1]
73         ORR R0,R0,#0x20 ;enable port F ,disable rest of the ports
74         STR R0,[R1]
75         NOP
76         NOP
77         NOP ; let the GPIO clock stabilize
78
79         LDR R1,=GPIO_PORTF_DIR
80         LDR R0,[R1]
81         BIC R0,#0xFF ;clear pins 0-7
82         ORR R0,#PORT_F_Pin_OUTPUTS;set PF1 PF2 PF3 as output
83         STR R0,[R1]
84
85         LDR R1,=GPIO_PORTF_AFSEL
86         LDR R0,[R1]
87         BIC R0,#0xFF ;clear all pins as no alternate function
88         STR R0,[R1]
89
90         LDR R1,=GPIO_PORTF_DEN
91         LDR R0,[R1]
92         ORR R0,#0xFF ;SET AS ALL DIGITAN ENABLED
93         STR R0,[R1]
94
95         BX LR
96     ENDP

```

INIT_TIMER0.s

```
7  RELOAD_VALUE          EQU 0xFFFF; reload value
8  SYSCTL_RCGCTIMER      EQU 0x400FE604 ; GPTM Gate Control
9  ; 16/32 Timer Registers
10 TIMER0_CFG             EQU 0x40030000;;for A and B 16bit/32bit selection,0x04 =16bit
11 TIMER0_TAMR            EQU 0x40030004;SET FUNC OF TIMER;;[1:0] 1=oneshot,2=periodic,3=capture
12 ;[2]=0 edge count,1=edge time , [4]=0 count down,1=up
13 TIMER0_CTL             EQU 0x4003000C;TIMER0 (en/dis,fall/ris/both)
14 TIMER0_IMR             EQU 0x40030018
15
16 TIMER0_TAILR           EQU 0x40030028 ; Timer interval;
17 ;in 16bit,value for count up or down .(if down, up to this number)
18 TIMER0_TAPR            EQU 0x40030038 ;presecalar
19
20
21 ;Nested Vector Interrupt Controller registers
22 NVIC_EN0_INT19          EQU 0x00080000 ; Interrupt 19 enable
23 NVIC_EN0                EQU 0xE000E100 ; IRQ 0 to 31 Set Enable Register
24 NVIC_PRI4              EQU 0xE000E410 ; IRQ 16 to 19 Priority Register
25
26
27                         AREA init_timer0 , CODE, READONLY
28                         THUMB
29                         EXPORT INIT_TIMER0
30
31
32 INIT_TIMER0 PROC
33
34                         LDR R1,=SYSCTL_RCGCTIMER ;clock for timer
35                         LDR R0,[R1]
36                         ORR R0,#0x01
37                         STR R0,[R1]
38
39                         LDR R1,=TIMER0_CTL;disable timer first
40                         LDR R0,[R1]
41                         BIC R0,#0xFF
42                         STR R0,[R1]
43
44                         LDR R1,=TIMER0_CFG;
45                         LDR R0,[R1]
46                         ORR R0,#0x4 ;select 16-bit MODE
47                         STR R0,[R1]
48
```

```

49      LDR R1,=TIMER0_TAMR;
50      LDR R0,[R1]
51      ORR R0,#0x2 ;periodic mode, down count
52      STR R0,[R1]
53
54      LDR R1,=TIMER0_TAILR;RELOAD VALUE
55      LDR R2,=RELOAD_VALUE
56      STR R2,[R1] ; load reload value
57
58
59      LDR R1,=TIMER0_TAPR;prescaler is select to 15 so the clock is 1 Mhz now
60      MOV R2,#0x4F
61      STR R2,[R1]
62
63      LDR R1, =TIMER0_IMR ;enable timeout interrupt
64      MOV R2, #0x01
65      STR R2, [R1]
66
67
68 ;INTERRUPT PRIORITY SET UP
69
70      LDR R1,=NVIC_PRI4
71      LDR R2,[R1]
72      AND R2,#0xFFFFFFFF ;clear int#19 priority
73      ORR R2,#0x40000000 ;set priority#19 as 2
74      STR R2,[R1]
75
76
77      LDR R1,=NVIC_EN0
78      LDR R0,[R1]
79      ORR R0,#0x00080000; SET bit 19 to 1 for enable interrupt #19
80      STR R0,[R1]
81
82
83      LDR R1,=TIMER0_CTL
84      LDR R3,[R1]
85      ORR R3,#0x03; enable timer
86      STR R3,[R1]
87
88      BX LR
89      ENDP
90

```


TIMER0A_HANDLER.s

```
7  GPIO_PORTF_DATA      EQU      0x40025038; DATA REGISTER WITH MASK PF1 PF2 FP3
8  COMPARE_VALUE        EQU 0x0E   ;used for the gpio state comparison
9  TIMER0_ICR           EQU 0x40030024 ; Timer Interrupt Clear
10
11
12                          AREA    my_handler_for_timer, CODE, READONLY
13                          THUMB
14                          EXPORT  TIMER0A_HANDLER
15 ;done in also previous video
16
17 TIMER0A_HANDLER PROC
18
19                          LDR R1,=TIMER0_ICR ;clear interrupt flag
20                          MOV R0,#0x01
21                          STR R0,[R1]
22
23                          LDR R1,=GPIO_PORTF_DATA ; read the masked gpio data
24                          LDR R2,[R1]
25                          MOV R8,#COMPARE_VALUE
26
27                          CMP R2, R8 ; if the pin was high
28                          BEQ reset_gpio;if equal , then reset gpio pin
29                          MOV R8,#COMPARE_VALUE ;
30                          STR R8,[R1]
31                          B finish
32
33 reset_gpio               MOV R8,#0x00 ;reset gpio
34                          STR R8,[R1]
35
36
37
38 finish
39
40                          BX LR
41                          ENDP
```

Main

```
8
9                          AREA main , CODE, READONLY
10                         THUMB
11                         EXTERN INIT_GPIO
12                         EXTERN INIT_TIMER0
13                         EXTERN TIMER0A_HANDLER
14                         EXTERN GPIO_PORTF_INIT
15                         EXPORT __main
16 __main
17
18
19                         BL INIT_GPIO
20                         BL GPIO_PORTF_INIT
21                         BL INIT_TIMER0
22
23 loop                    B loop
24
25                         END
```