

27/11/2018

EE447 LABORATORY  
EXPERIMENT 3  
PRELIMINARY REPORT

Muttalip Caner TOL

2031466

Wednesday Afternoon

```
1 ;*****
2 ; Program rotationSignal.s
3 ; This program initializes the GPIO port B
4 ;*****
5 ; EQU Directives
6 ;*****
7 ;LABEL      DIRECTIVE  VALUE      COMMENT
8 GPIO_PORTB_DATA    EQU      0x400053FC      ; data address to all
pins
9 ;*****
10 ;LABEL      DIRECTIVE  VALUE      COMMENT
11             AREA      main, READONLY, CODE
12             THUMB
13             EXPORT    __main      ; make available
14             EXTERN    InitSysTick
15             EXTERN    INIT_GPIO
16
17 __main      MOV      R4, #0x80
18             MOV      R5, #1
19             BL        INIT_GPIO      ; initialize GPIO port B
20             BL        InitSysTick
21
22 loop        B         loop
23             END
24
25
```

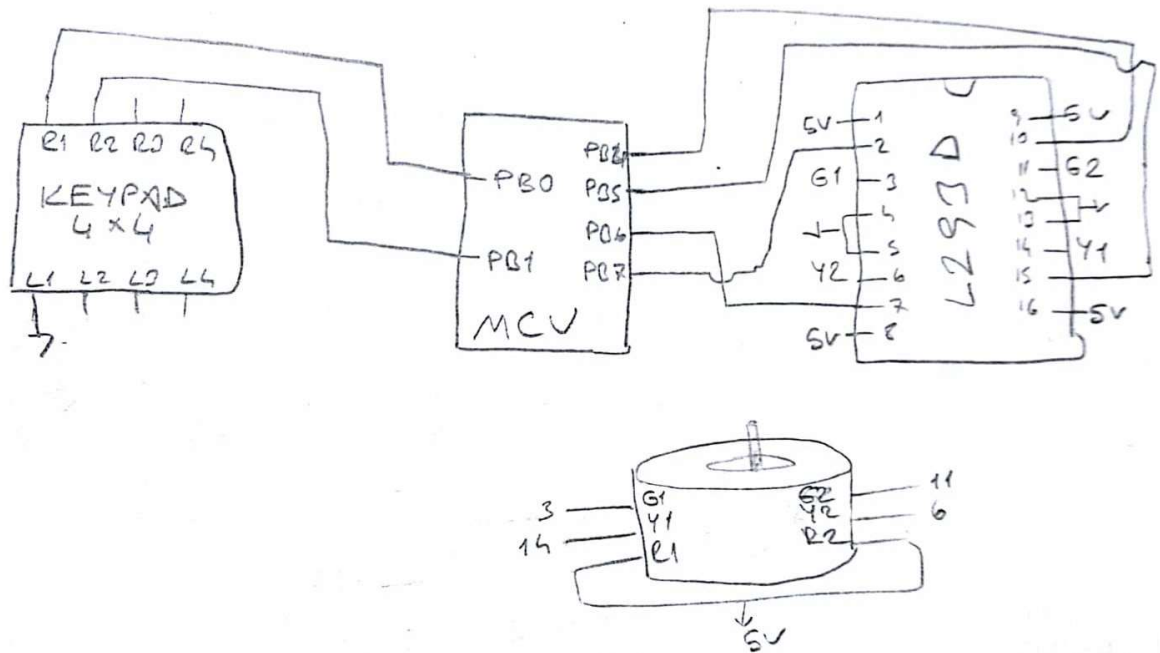
```

1  ;*****
2  ; Program rotationSignal.s
3  ; This subroutine sends  GPIO port B the necessary signals to
4  ; demonstrate the Full Step Mode in both directions
5  ; (cw or ccw depending on the register R5 ).
6  ;*****
7  ; EQU Directives
8  ;*****
9  ;LABEL      DIRECTIVE  VALUE      COMMENT
10 GPIO_PORTB_DATA EQU      0x400053FC      ; data address to all pins
11 PB_INP      EQU      0x4000503C
12 PB_OUT      EQU      0x400053C0
13 ;*****
14 ;LABEL      DIRECTIVE  VALUE      COMMENT
15              AREA      main, READONLY, CODE
16              THUMB
17              EXPORT    rotationSignal      ; make available
18              EXTERN    INIT_GPIO           ;
19              EXTERN    InitSysTick
20
21 rotationSignal PUSH      {LR}
22              BL        InitSysTick         ; initialize Sys Tick
23              BL        INIT_GPIO          ; initialize GPIO port B
24              CMP       R5, #0              ; R5=0 => clock wise
25              BNE       counterclockwise    ; otherwise counter clockwise
26
27 clockwise   LDR        R0, =PB_OUT        ; output pins PB[7:4]
28              LDR        R1, [R0]          ;
29              BIC        R1, #0xFF
30              ORR        R1, R4            ; make output high
31              STR        R1, [R0]
32              LSR        R4, #1            ; shift right the output
33              CMP        R4, #0x08         ;
34              MOVEQ      R4, #0x80         ;
35              B          finish
36
37 counterclockwise LDR      R0, =PB_OUT      ; output pins PB[7:4]
38              LDR        R1, [R0]          ;
39              BIC        R1, #0xFF
40              ORR        R1, R4            ; make output high
41              STR        R1, [R0]
42              LSL        R4, #1            ; shift right the output
43              CMP        R4, #0x100        ;
44              MOVEQ      R4, #0x10         ;
45              B          finish
46
47 finish      POP        {LR}              ; exit from interrupt handler
48              BX        LR
49
50              END

```

```
1 ;*****
2 ; Your SystemTimer.s source file to implement
3 ; initialization and ISR
4 ;*****
5 ; Definitions of the labels standing for
6 ; the address of the registers
7 NVIC_ST_CTRL EQU 0xE00E010
8 NVIC_ST_RELOAD EQU 0xE00E014
9 NVIC_ST_CURRENT EQU 0xE00E018
10 SHP_SYS_PRI3 EQU 0xE00ED20
11 ; end of the register label definitions
12 ; 0x30D40 = 200000 -> 200000*250 ns = 50ms
13 RELOAD_VALUE EQU 0xC350
14 ;RELOAD_VALUE EQU 0xFF350
15 ;*****
16 ; Initialization area
17 ;*****
18 ;LABEL DIRECTIVE VALUE COMMENT
19 AREA init_isr, CODE, READONLY, ALIGN=2
20 THUMB
21 EXPORT InitSysTick
22
23 InitSysTick PROC
24 ; first disable system timer and the related interrupt
25 ; then configure it to use internal oscillator PIOSC/4
26 LDR R1, =NVIC_ST_CTRL
27 MOV R0, #0
28 STR R0, [R1]
29 ; now set the time-out period
30 LDR R1, =NVIC_ST_RELOAD
31 LDR R0, =RELOAD_VALUE
32 STR R0, [R1]
33 ; time out period is set
34 ; now set the current timer value to the time out value
35 LDR R1, =NVIC_ST_CURRENT
36 STR R0, [R1]
37 ; current timer = time out period
38 ; now set the priority level
39 LDR R1, =SHP_SYS_PRI3
40 MOV R0, #0x40000000
41 STR R0, [R1]
42 ; priority is set to 2
43 ; now enable system timer and the related interrupt
44 LDR R1, =NVIC_ST_CTRL
45 MOV R0, #0x03
46 STR R0, [R1]
47 ; set up for system time is now complete
48 BX LR
49 ENDP
50
51 END
```

Q2.



```
1 ;*****
2 ; Program rotationSignal.s
3 ; This program initializes the GPIO port B
4 ;*****
5 ; EQU Directives
6 ;*****
7 ;LABEL      DIRECTIVE  VALUE      COMMENT
8 GPIO_PORTB_DATA    EQU      0x400053FC      ; data address to all pins
9 GPIO_PORTB_MIS      EQU      0x40005418
10 PB_INP             EQU      0x4000503C
11 ;*****
12 ;LABEL      DIRECTIVE  VALUE      COMMENT
13             AREA      main, READONLY, CODE
14             THUMB
15             EXPORT    __main      ; make available
16             EXTERN    INIT_GPIO
17             EXTERN    stepSignal
18
19 __main
20             MOV        R4, #0x80
21             MOV        R5, #0
22             MOV        R6, #2
23             MOV        R7, #2
24             BL         INIT_GPIO      ; initialize GPIO port B
25             CPSIE      I;
26             MOV        R8, #0
27             MOV        R9, #0
28 loop        LDR        R0,=PB_INP
29             LDR        R0,[R0]
30             CMP        R8, #1
31             BEQ        loop
32 loop2       CMP        R9, #1
33             BNE        skip
34             BEQ        loop2
35 skip        LDR        R0,=PB_INP
36             LDR        R0,[R0]
37             CMP        R0,#0xF
38             BEQ        loop
39             BL         stepSignal
40             B          loop
41             ENDP
42             END
43
44
```

```

1  ;*****
2  ; Program rotationSignal.s
3  ; This subroutine sends  GPIO port B the necessary signals to
4  ; demonstrate the Full Step Mode in both directions
5  ; (cw or ccw depending on the register R5 ).
6  ;*****
7  ; EQU Directives
8  ;*****
9  ;LABEL      DIRECTIVE    VALUE      COMMENT
10 GPIO_PORTB_DATA EQU      0x400053FC      ; data address to all pins
11 PB_INP      EQU      0x4000503C
12 PB_OUT      EQU      0x400053C0
13 GPIO_PORTB_ICR EQU      0x4000541C
14 NVIC_ST_CTRL EQU 0xE000E010
15 ;*****
16 ;LABEL      DIRECTIVE    VALUE      COMMENT
17             AREA        main, READONLY, CODE
18             THUMB
19             EXPORT      stepSignal      ; make available
20             EXTERN      INIT_GPIO      ;
21             EXTERN      InitSysTick
22
23 stepSignal   PUSH        {LR}
24             CMP          R8, #1
25             BEQ          CHECK
26             MOV          R6, R0
27             BL           InitSysTick
28
29             MOV          R8, #1
30             B            break
31
32 CHECK        CMP          R0, #0xE
33             MOVEQ        R5, #0
34             CMP          R0, #0xD
35             MOVEQ        R5, #1
36
37
38 release      LDR          R0, =PB_INP
39             LDR          R0, [R0]
40             CMP          R0, #0xF
41             BEQ          check2
42             B            check3
43
44 check2       CMP          R9, #1
45             BEQ          turn
46             MOV          R9, #1
47             B            release
48
49 check3       CMP          R9, #1
50             BEQ          finish
51             MOV          R9, #1
52             MOV          R8, #0
53             BL           InitSysTick
54             B            break
55
56 turn        CMP          R5, #0      ; R5=0 => clockwise
57             BEQ          clockwise
58             B            counterclockwise
59
60
61 clockwise    LDR          R0, =PB_OUT      ; output pins PB[7:4]
62             LDR          R1, [R0]          ;
63             BIC          R1, #0xFF
64             ORR          R1, R4            ; make output high
65             STR          R1, [R0]
66             LSR          R4, #1            ; shift right the output
67             CMP          R4, #0x08        ;
68             MOVEQ        R4, #0x80        ;
69             B            finish
70
71 counterclockwise LDR          R0, =PB_OUT      ; output pins PB[7:4]
72             LDR          R1, [R0]          ;
73             BIC          R1, #0xFF
74             ORR          R1, R4            ; make output high
75             STR          R1, [R0]
76             LSL          R4, #1            ; shift right the output
77             CMP          R4, #0x100        ;

```

```
78      MOVEQ    R4, #0x10      ;
79      B        finish
80
81  finish      MOV     R8, #0
82             MOV     R9, #0
83             LDR     R1, =NVIC_ST_CTRL
84             MOV     R0, #0
85             STR     R0, [R1]
86  break      POP     {LR}      ; exit from interrupt handler
87             BX      LR
88
89             END
```

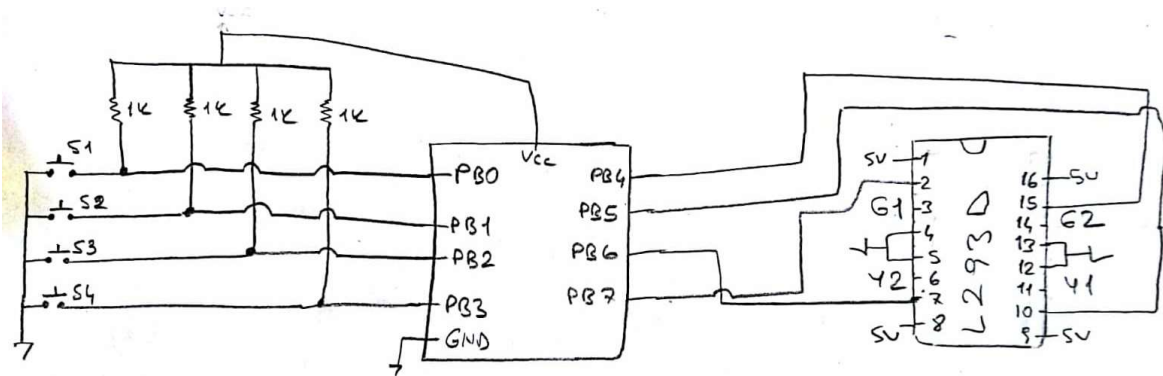


```

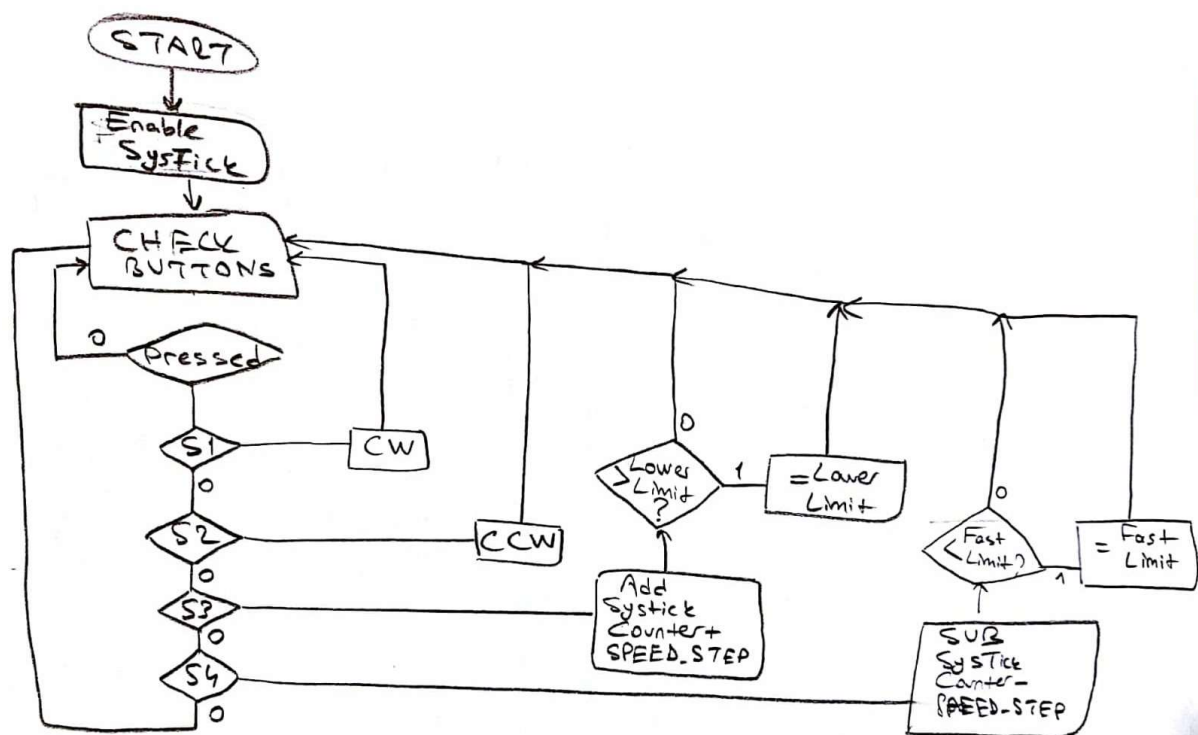
1  ;*****
2  ; Program INIT_GPIO.s
3  ; This program initializes the GPIO port B
4  ;*****
5  ; EQU Directives
6  ;*****
7  ; LABEL      DIRECTIVE  VALUE      COMMENT
8  GPIO_PORTB_DATA      EQU          0x400053FC      ; data address to all pins
9                                     ; PUR Offset 0x510
10 GPIO_PORTB_PUR        EQU          0x40005510      ; PUR actual address
11 PUB                  EQU          0x0F            ; or #2_00001111 B0-3 PULL-UP, B4-7 PULL-DOWN
12 GPIO_PORTB_DIR        EQU          0x40005400
13 GPIO_PORTB_AFSEL      EQU          0x40005420
14 GPIO_PORTB_DEN        EQU          0x4000551C
15 GPIO_PORTB_IS         EQU          0x40005404
16 GPIO_PORTB_IBE        EQU          0x40005408
17 GPIO_PORTB_IEV        EQU          0x4000540C
18 GPIO_PORTB_IM         EQU          0x40005410
19 GPIO_PORTB_ICR        EQU          0x4000541C
20 IOB                   EQU          0xF0            ; B0-3 INPUT, B4-7 OUTPUT
21 NVIC_ENABLE           EQU          0xE000E100
22 SYSCCTL_RCGCGPIO      EQU          0x400FE608
23 ;*****
24 ; LABEL      DIRECTIVE  VALUE      COMMENT
25                                     AREA      main, READONLY, CODE
26                                     THUMB
27                                     EXPORT      INIT_GPIO      ; make available
28
29 INIT_GPIO          PUSH      {LR,R0,R1}
30                   LDR        R1, =SYSCCTL_RCGCGPIO
31                   LDR        R0, [R1]
32                   ORR        R0, R0, #0x1F
33                   STR        R0, [R1]
34                   NOP
35                   NOP
36                   NOP      ; let GPIO clock stabilize
37
38                   LDR        R1, =GPIO_PORTB_DIR      ; config. of port B starts
39                   LDR        R0, [R1]
40                   BIC        R0, #0xFF
41                   ORR        R0, #IOB
42                   STR        R0, [R1]
43                   LDR        R1, =GPIO_PORTB_AFSEL
44                   LDR        R0, [R1]
45                   BIC        R0, #0xFF
46                   STR        R0, [R1]
47                   LDR        R1, =GPIO_PORTB_DEN
48                   LDR        R0, [R1]
49                   ORR        R0, #0xFF
50                   STR        R0, [R1]
51                   LDR        R0, =GPIO_PORTB_PUR
52                   MOV        R1, #PUB
53                   STR        R1, [R0]      ; config. of port B ends
54                   MOV        R0, #0
55                   LDR        R1, =GPIO_PORTB_IS
56                   STR        R0, [R1]
57                   LDR        R1, =GPIO_PORTB_IBE
58                   MOV        R0, #0x0
59                   STR        R0, [R1]
60                   LDR        R1, =GPIO_PORTB_IEV
61                   MOV        R0, #0xFF
62                   ;STR        R0, [R1]
63                   LDR        R1, =GPIO_PORTB_IM
64                   MOV        R0, #0x0F
65                   STR        R0, [R1]
66                   LDR        R1, =NVIC_ENABLE
67                   MOV        R0, #0xF0
68                   STR        R0, [R1]
69                   LDR        R1, =GPIO_PORTB_ICR
70                   MOV        R0, #0xFF
71                   STR        R0, [R1]
72
73
74                   POP        {LR,R0,R1}
75                   BX         LR
76                   END

```

Q4.



Q5. This problem includes very similar works with previous steps of this experiment. We should use the SysTick interrupt to drive the step motor and in the main function we can use polling to check for buttons.



```

1  ;*****
2  ; Program rotationSignal.s
3  ; This program initializes the GPIO port B
4  ;*****
5  ; EQU Directives
6  ;*****
7  ; LABEL      DIRECTIVE  VALUE      COMMENT
8  GPIO_PORTB_DATA EQU      0x400053FC      ; data address to all pins
9  PB_INP        EQU      0x4000503C
10 RELOAD_VALUE  EQU      0x30D40
11 SPEED_STEP    EQU      0x9350
12 FAST_LIMIT    EQU      0x9350
13 SLOW_LIMIT     EQU      0xF4240
14 ;*****
15 ; LABEL      DIRECTIVE  VALUE      COMMENT
16              AREA      main, READONLY, CODE
17              THUMB
18              EXPORT    __main          ; make available
19              EXTERN    InitSysTick
20              EXTERN    INIT_GPIO
21              EXTERN    DELAY100
22
23 __main
24      MOV      R4, #0x80
25      MOV      R5, #1
26      LDR      R7, =RELOAD_VALUE
27
28      BL       INIT_GPIO          ; initialize GPIO port B
29      BL       InitSysTick
30
31 loop
32      LDR      R0, =PB_INP
33      LDR      R0, [R0]
34      CMP      R0, #0xF
35      BEQ      loop
36
37 S_1
38      MOV      R6, #0
39      LDR      R0, =PB_INP
40      LDR      R0, [R0]
41      CMP      R0, #0xE
42      BNE      S_2
43      BL       DELAY100          ; delay 100ms
44      CMP      R6, #1          ; if it is the second time
45      BEQ      S_neg_1          ; turn on led1
46      MOV      R6, #1
47      B        S__1
48
49 S_neg_1
50      MOV      R6, #0
51      LDR      R0, =PB_INP
52      LDR      R0, [R0]
53      CMP      R0, #0xF
54      BNE      S__neg_1
55      BL       DELAY100          ; delay 100ms
56      CMP      R6, #1          ; if it is the second time
57      BEQ      out1
58      MOV      R6, #1
59      B        S__neg_1
60
61 out1
62      MOV      R5, #0
63      B        loop
64
65 S_2
66      MOV      R6, #0
67      LDR      R0, =PB_INP
68      LDR      R0, [R0]
69      CMP      R0, #0xD
70      BNE      S_3
71      BL       DELAY100          ; delay 100ms
72      CMP      R6, #1          ; if it is the second time
73      BEQ      S_neg_2          ; turn on led1
74      MOV      R6, #1
75      B        S__2
76
77 S_neg_2
78      MOV      R6, #0
79      LDR      R0, =PB_INP
80      LDR      R0, [R0]
81      CMP      R0, #0xF
82      BNE      S__neg_2
83      BL       DELAY100          ; delay 100ms
84      CMP      R6, #1          ; if it is the second time
85      BEQ      out2

```

```

78      MOV      R6, #1
79      B        S__neg_2
80
81      out2      MOV      R5, #1
82      B        loop
83
84      S_3      MOV      R6, #0
85      S__3      LDR      R0, =PB_INP
86      LDR      R0, [R0]
87      CMP      R0, #0xB
88      BNE      S_4
89      BL        DELAY100          ; delay 100ms
90      CMP      R6, #1          ; if it is the second time
91      BEQ      S_neg_3        ; turn on led1
92      MOV      R6, #1
93      B        S__3
94
95      S_neg_3    MOV      R6, #0
96      S__neg_3   LDR      R0, =PB_INP
97      LDR      R0, [R0]
98      CMP      R0, #0xF
99      BNE      S__neg_3
100     BL        DELAY100          ; delay 100ms
101     CMP      R6, #1          ; if it is the second time
102     BEQ      out3
103     MOV      R6, #1
104     B        S__neg_3
105
106     out3      LDR      R8, =SPEED_STEP
107     ADD      R7, R8
108     LDR      R9, =SLOW_LIMIT
109     CMP      R7, R9
110     BLO      loop
111     LDR      R7, =SLOW_LIMIT
112     B        loop
113
114     S_4      MOV      R6, #0
115     S__4      LDR      R0, =PB_INP
116     LDR      R0, [R0]
117     CMP      R0, #0x7
118     BNE      loop
119     BL        DELAY100          ; delay 100ms
120     CMP      R6, #1          ; if it is the second time
121     BEQ      S_neg_4        ; turn on led1
122     MOV      R6, #1
123     B        S__4
124
125     S_neg_4    MOV      R6, #0
126     S__neg_4   LDR      R0, =PB_INP
127     LDR      R0, [R0]
128     CMP      R0, #0xF
129     BNE      S__neg_4
130     BL        DELAY100          ; delay 100ms
131     CMP      R6, #1          ; if it is the second time
132     BEQ      out4
133     MOV      R6, #1
134     B        S__neg_4
135
136     out4      LDR      R8, =SPEED_STEP
137     SUB      R7, R8
138     LDR      R9, =FAST_LIMIT
139     CMP      R7, R9
140     BHS      loop
141     LDR      R7, =FAST_LIMIT
142     B        loop
143
144
145
146     END
147
148

```

```

1  ;*****
2  ; Program rotationSignal.s
3  ; This subroutine sends  GPIO port B the necessary signals to
4  ; demonstrate the Full Step Mode in both directions
5  ; (cw or ccw depending on the register R5 ).
6  ;*****
7  ; EQU Directives
8  ;*****
9  ;LABEL      DIRECTIVE  VALUE      COMMENT
10 GPIO_PORTB_DATA    EQU      0x400053FC      ; data address to all pins
11 PB_INP             EQU      0x4000503C
12 PB_OUT             EQU      0x400053C0
13 ;*****
14 ;LABEL      DIRECTIVE  VALUE      COMMENT
15             AREA      main, READONLY, CODE
16             THUMB
17             EXPORT    rotationSignal      ; make available
18             EXTERN    INIT_GPIO           ;
19             EXTERN    InitSysTick
20
21 rotationSignal      PUSH      {LR}
22                     BL        InitSysTick      ; initialize Sys Tick
23                     BL        INIT_GPIO        ; initialize GPIO port B
24                     CMP       R5, #0          ; R5=0 => clock wise
25                     BNE      counterclockwise ; otherwise counter clockwise
26
27 clockwise          LDR       R0, =PB_OUT      ; output pins PB[7:4]
28                     LDR       R1, [R0]        ;
29                     BIC       R1, #0xFF
30                     ORR       R1, R4          ; make output high
31                     STR       R1, [R0]
32                     LSR       R4, #1          ; shift right the output
33                     CMP       R4, #0x08       ;
34                     MOVEQ     R4, #0x80       ;
35                     B         finish
36
37 counterclockwise   LDR       R0, =PB_OUT      ; output pins PB[7:4]
38                     LDR       R1, [R0]        ;
39                     BIC       R1, #0xFF
40                     ORR       R1, R4          ; make output high
41                     STR       R1, [R0]
42                     LSL       R4, #1          ; shift right the output
43                     CMP       R4, #0x100      ;
44                     MOVEQ     R4, #0x10       ;
45                     B         finish
46
47 finish            POP       {LR}              ; exit from interrupt handler
48                     BX        LR
49
50                     END

```

```
1 ;*****
2 ; Your SystemTimer.s source file to implement
3 ; initialization and ISR
4 ;*****
5 ; Definitions of the labels standing for
6 ; the address of the registers
7 NVIC_ST_CTRL EQU 0xE00E010
8 NVIC_ST_RELOAD EQU 0xE00E014
9 NVIC_ST_CURRENT EQU 0xE00E018
10 SHP_SYS_PRI3 EQU 0xE00ED20
11 ; end of the register label definitions
12 ; 0x30D40 = 200000 -> 200000*250 ns = 50ms
13 RELOAD_VALUE EQU 0x30D40
14 ;RELOAD_VALUE EQU 0xFF350
15 ;*****
16 ; Initialization area
17 ;*****
18 ;LABEL DIRECTIVE VALUE COMMENT
19 AREA init_isr, CODE, READONLY, ALIGN=2
20 THUMB
21 EXPORT InitSysTick
22
23 InitSysTick PROC
24 ; first disable system timer and the related interrupt
25 ; then configure it to use internal oscillator PIOSC/4
26 LDR R1, =NVIC_ST_CTRL
27 MOV R0, #0
28 STR R0, [R1]
29 ; now set the time-out period
30 LDR R1, =NVIC_ST_RELOAD
31 MOV R0, R7 ;*****MODIFIED****
32 STR R0, [R1]
33 ; time out period is set
34 ; now set the current timer value to the time out value
35 LDR R1, =NVIC_ST_CURRENT
36 STR R0, [R1]
37 ; current timer = time out period
38 ; now set the priority level
39 LDR R1, =SHP_SYS_PRI3
40 MOV R0, #0x40000000
41 STR R0, [R1]
42 ; priority is set to 2
43 ; now enable system timer and the related interrupt
44 LDR R1, =NVIC_ST_CTRL
45 MOV R0, #0x03
46 STR R0, [R1]
47 ; set up for system time is now complete
48 BX LR
49 ENDP
50
51 END
```