

```

1 /*****
2 * SysTick_Handler
3 * Interrupt Service Routine for system tick counter
4 * The name of this function cannot be changed - it establishes the linkage
5 *****/
6 .equ TIMER3_CMD,      0x40010C04
7 .equ GPIO_PD_DOUTCLR, 0x40006080
8
9 .syntax unified
10 .text
11 .thumb
12 .thumb_func
13 .align 4
14 .globl SysTick_Handler
15 .type SysTick_Handler, %function
16 SysTick_Handler:
17     push    {r4-r11, lr}
18
19     ldr r8,=TIMER3_CMD // timer 3 command register
20     ldr r0,=2           // stop timer command
21     str r0,[r8]         // timer 3 now stopped
22
23     ldr r6,=SystemTick // r6 is address of SystemTick
24     ldr r7,[r6]         // r7 is current tick
25     add r7,r7,#1        // increment tick
26     str r7,[r6]         // save new tick
27     //keep this below
28     ldr r4,=CurrentTask // r4 is address of current task
29     ldr r5,[r4]         // r5 is current task
30     str sp,[r5,#0]      // stack pointer is first thing in TCB
31     bl scheduler
32     str r0,[r4]         // save new CurrentTask
33     ldr sp,[r0,#0]      // get sp from new current task
34
35     ldr r1,[r0,#4]      // get timer3_on value
36     str r1,[r8]         // start timer 3 if appropriate
37
38     pop     {r4-r11, pc}
39
40     .thumb_func
41     .align 4
42     .globl SVC_Handler
43     .type SVC_Handler, %function
44 SVC_Handler:
45     push    {r4-r11, lr}
46
47     ldr r8,=TIMER3_CMD // timer 3 command register
48     ldr r0,=2           // stop timer command
49     str r0,[r8]         // timer 3 now stopped
50
51     ldr r4,=CurrentTask // r4 is address of current task
52     ldr r5,[r4]         // r5 is current task
53     str sp,[r5,#0]      // stack pointer is first thing in TCB
54     bl scheduler
55     str r0,[r4]         // save new CurrentTask
56     ldr sp,[r0,#0]      // get sp from new current task
57

```

```

58     ldr r9,=GPIO_PD_DOUTCLR // Port D "clear" register
59     ldr r10,=0x1f           // clear bits 0-4
60     str r10,[r9]            // and clear
61     add r9,#-4              // point to Port D "set" register
62     ldr r10,=0x20           // set bit 5 (context switch bit)
63     str r10,[r9]            // and it is set
64
65     ldr r1,[r0,#8]          // get task_mask from current TCB
66     str r1,[r9]             // set task_mask bits on port D
67     add r9,#4               // point to port D "clear" register
68     str r10,[r9]            // clear bit 5 (context switch bit)
69     // possibly start timer 3
70
71     ldr r1,[r0,#4]          // get timer3_on value
72     str r1,[r8]             // start timer 3 if appropriate
73
74     pop     {r4-r11, pc}
75
76     .thumb_func
77     .align 4
78     .globl Yield
79     .type Yield, %function
80 Yield:
81     svc #0 // raise SVC interrupt
82     bx lr // return from subroutine
83
84     .end
85

```