

```

1 /*****
2  * @file main.c
3  * @brief main() function.
4  *****/
5  * # License
6  * <b>Copyright 2020 Silicon Laboratories Inc. www.silabs.com</b>
7  *****/
8  *
9  * The licensor of this software is Silicon Laboratories Inc. Your use of this
10 * software is governed by the terms of Silicon Labs Master Software License
11 * Agreement (MSLA) available at
12 * www.silabs.com/about-us/legal/master-software-license-agreement. This
13 * software is distributed to you in Source Code format and is governed by the
14 * sections of the MSLA applicable to Source Code.
15 *
16 *****/
17 #include "sl_component_catalog.h"
18 #include "sl_system_init.h"
19 #include "app.h"
20 #include "EM_device.h"
21 #include "EM_chip.h"
22 #if defined(SL_CATALOG_POWER_MANAGER_PRESENT)
23 #include "sl_power_manager.h"
24 #endif
25 #if defined(SL_CATALOG_KERNEL_PRESENT)
26 #include "sl_system_kernel.h"
27 #else // SL_CATALOG_KERNEL_PRESENT
28 #include "sl_system_process_action.h"
29 #endif // SL_CATALOG_KERNEL_PRESENT
30
31 /*****
32  * Extern Includes for Lab03
33  *****/
34 extern void task_A(), task_B(), task_C(), task_D(), task_E(), task_F();
35
36 int n = 0;
37
38 void SysTick_Handler(void)
39 {
40     switch(n)
41     {
42     case 0:
43         task_A();
44         task_B();
45         task_D();
46         break;
47     case 1:
48         task_F();
49         task_A();
50         task_C();
51         break;
52     case 2:
53         task_A();
54         task_B();
55         task_C();
56         break;
57     case 3:

```

```
58     task_A();
59     task_E();
60     break;
61     case 4:
62         task_B();
63         task_A();
64         task_D();
65         break;
66     case 5:
67         task_A();
68         task_B();
69         task_C();
70         break;
71     case 6:
72         task_F();
73         task_A();
74         task_C();
75         break;
76     case 7:
77         task_A();
78         task_B();
79         task_D();
80         break;
81     case 8:
82         task_A();
83         task_E();
84         break;
85     case 9:
86         task_A();
87         task_B();
88         task_C();
89         //n = 0;
90         break;
91 }
92 n = (n+1)%10;
93 }
94
95
96 int main(void)
97 {
98     // Initialize Silicon Labs device, system, service(s) and protocol stack(s).
99     // Note that if the kernel is present, processing task(s) will be created by
100    // this call.
101    sl_system_init();
102    SystemCoreClock = 14000000; // 14 MHz for this device
103    SysTick_Config(3 * SystemCoreClock / 1000); //3ms slice
104
105
106    // Initialize the application. For example, create periodic timer(s) or
107    // task(s) if the kernel is present.
108    app_init();
109
110    #if defined(SL_CATALOG_KERNEL_PRESENT)
111        // Start the kernel. Task(s) created in app_init() will start running.
112        sl_system_kernel_start();
113    #else // SL_CATALOG_KERNEL_PRESENT
114
```

```
115
116 while (1) {
117     // Do not remove this call: Silicon Labs components process action routine
118     // must be called from the super loop.
119     sl_system_process_action();
120
121     // Application process.
122     app_process_action();
123
124 #if defined(SL_CATALOG_POWER_MANAGER_PRESENT)
125     // Let the CPU go to sleep if the system allows it.
126     sl_power_manager_sleep();
127 #endif
128 }
129 #endif // SL_CATALOG_KERNEL_PRESENT
130 }
131
```