```
 2  * FlashLEDCodeCPP.cpp
 7 #include <uTTCOS2013/uTTCOS.h>
 8 #include "FlashLEDCodeCPP.h"
 9
10 // State Machine Variables (global)
11 bool LED1_is_on = false;
12 enum LED3_State {JUST_STARTED, IS_OFF, IS_ON};
13 LED3_State currentState = JUST_STARTED;
14 enum LED5_State {LED5_JUST_STARTED, LED5_IS_OFF, LED5_IS_ON, LED5_STAYS_ON};
15 LED5_State currentState_LED5 = LED5_JUST_STARTED;
16
17 // Reset to Base States
18 void ResetStates_CPP(void) {
19     LED1_is_on = false;
20     currentState = JUST_STARTED;
21     currentState_LED5 = LED5_JUST_STARTED;
22 }
23 #define LED1_BITMASK 0x01
24 void FlashLED1_CPP(void) {
25     unsigned char LEDCurrentState = uTTCOS_ReadLED();
26     unsigned char nxt_State = NULL;
27     if (!LED1_is_on) {
28         nxt_State = LED1_BITMASK | LEDCurrentState;
29         uTTCOS_WriteLED(nxt_State);  // Turn on LED 1
30         LED1_is_on = true;
31     }
32     else {
33         nxt_State = ~LED1_BITMASK & LEDCurrentState;
34         uTTCOS_WriteLED(nxt_State);  // Turn off LED 1
35         LED1_is_on = false;
36     }
37 }
38
39 #define LED3_BIT_MASK  0x04
40 void FlashLED3_CPP(void) {
41     LED3_State nextStateToDo = currentState;
42     unsigned char LEDCurrentState = uTTCOS_ReadLED();
43     unsigned char nxt_State = NULL;
44
45     switch (currentState) {
46     case JUST_STARTED:
47         nxt_State = ~LED3_BIT_MASK & LEDCurrentState;
48         uTTCOS_WriteLED(nxt_State); // Turn off LED 3
49         nextStateToDo = IS_OFF;
50         break;
51     case IS_OFF:
52         nxt_State = LED3_BIT_MASK | LEDCurrentState;
53         uTTCOS_WriteLED(nxt_State);  // Was off turn it on
54         nextStateToDo = IS_ON;
55         break;
56     case IS_ON:
57         nxt_State = ~LED3_BIT_MASK & LEDCurrentState;
58         uTTCOS_WriteLED(nxt_State); // Turn off LED 3
59         nextStateToDo = IS_OFF;
60     break;
61     }
62
```

```
63      currentState = nextStateToDo;
64  }
65
66  // Making some C++ extensions to handle "LED state machine"
67
68  #define BEGIN_STATE case
69  #define END_STATE break;
70
71  #define LED5_BITMASK 0x10
72
73  void FlashLED5_CPP(void) {
74      unsigned char LEDCurrentState = uTTCOS_ReadLED();
75      unsigned char nxt_State = NULL;
76      LED5_State nextStateToDo = currentState_LED5;
77
78      switch (currentState_LED5) {
79      BEGIN_STATE LED5_JUST_STARTED:
80          nxt_State = ~LED5_BITMASK & LEDCurrentState;
81          uTTCOS_WriteLED(nxt_State);              // Turn off LED 5
82          nextStateToDo = LED5_IS_OFF;
83      END_STATE
84
85      BEGIN_STATE LED5_IS_OFF:
86          nxt_State = LED5_BITMASK | LEDCurrentState;
87          uTTCOS_WriteLED(nxt_State);  // Was off turn it on
88          nextStateToDo = LED5_IS_ON;
89      END_STATE
90
91      BEGIN_STATE LED5_IS_ON:
92          nextStateToDo = LED5_STAYS_ON;        // Just on -- keep on
93      END_STATE
94
95      BEGIN_STATE LED5_STAYS_ON:
96          nxt_State = ~LED5_BITMASK & LEDCurrentState;
97          uTTCOS_WriteLED(nxt_State);                 // Was off turn it on
98          nextStateToDo = LED5_IS_OFF;
99      END_STATE
100     }
101
102     currentState_LED5 = nextStateToDo;
103 }
104
```