

```
1  //inclusion of headers
2  #include "_threadsCore.h"
3  #include "_kernelCore.h"
4  #include "osDefs.h"
5
6  //inclusion of libraries
7  #include <LPC17xx.h>
8  #include "stdio.h"
9  #include "uart.h"
10 #include "LED.h"
11
12 //global variables
13 int thread1Call = 0;
14 int thread2Call = 0;
15 int thread3Call = 0;
16 //test constants
17 const int TEST_RANGE = 3000;
18 const int TEST_INTERVAL1 = 1000;
19 const int TEST_INTERVAL2 = 2000;
20
21 //thread functions
22 void thread1(){
23     while(1){
24         thread1Call++;
25
26         //turn on LED 1 but keep others off
27         if ((thread1Call % TEST_RANGE) <= TEST_INTERVAL1){
28             printf("This is a pause");
29
30             LED_set(1);
31             LED_clear(3);
32             LED_clear(5);
33         }
34
35         osSched(); //call yield/scheduler function
36     }
37 }
38
39 void thread2(){
40     while(1){
41         thread2Call++;
42
43         //turn on LED 3 but keep others off
44         if ((thread1Call % TEST_RANGE) > TEST_INTERVAL1 && (thread1Call % TEST_RANGE) <= TEST_INTERVAL2){
45             printf("This is a pause");
46             LED_clear(1);
47             LED_set(3);
48             LED_clear(5);
49         }
50         osSched(); //call yield/scheduler function
51     }
52 }
53
54 void thread3(){
55     while(1){
56         thread3Call++;
57
58         //turn on LED 5 but keep others off
59         if ((thread1Call % TEST_RANGE) > TEST_INTERVAL2){
60             printf("This is a pause");
61             LED_clear(1);
62             LED_clear(3);
63             LED_set(5);
64         }
65         osSched(); //call yield/scheduler function
66     }
67 }
68
69 int main( void )
70 {
71     SystemInit();
72     printf("\nRunning L-OS-S...\r\n");
```

```
73
74 //LED initialization (set directions to output, start with all LEDs turned off)
75 LED_setup();
76 for(int i = 0; i < 7; i++){
77     LED_clear(i);
78 }
79
80 //Initialize the kernel
81 kernelInit();
82
83 //Initialize each thread
84 osThreadNew(thread1);
85 osThreadNew(thread2);
86 osThreadNew(thread3);
87
88 //Start running the threads
89 osKernelStart();
90
91 printf("L-OS-S is lost (done)");
92 while(1);
93 }
94
```