

```

1  /*****
2  *
3  *   |-----|-----|-----|-----|   |-----|-----|
4  *   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
5  *   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
6  *   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
7  *   |-----|-----|-----|-----|   |-----|-----|
8  *
9  *****/
10 *
11 * File           : counter.c
12 * Version        : 1.0
13 *
14 *****/
15 *
16 * Description    : Managing time counters for application
17 *                  based on timer (1 by default)
18 *
19 *****/
20 *
21 * Author         : Miguel Santos
22 * Date           : 14.09.2023
23 *
24 *****/
25 *
26 * MPLAB X        : 5.45
27 * XC32           : 2.50
28 * Harmony        : 2.06
29 *
30 *****/
31
32 #include "counter.h"
33
34 /*****
35
36  /* Global infinite system counter */
37  uint32_t SYS_counter = 0;
38
39  *****/
40
41 void CNT_Initialize(S_Counter *counter, uint32_t target)
42 {
43     counter->value = SYS_counter;
44     counter->target = target;
45 }
46
47 /*****
48
49 bool CNT_Check( S_Counter *counter )
50 {
51     bool checkStatus;
52
53     checkStatus = ((SYS_counter - counter->value) >= counter->target);
54
55     if(checkStatus)
56     {
57         CNT_Reset(counter);
58     }
59
60     return checkStatus;
61 }
62
63 *****/
64
65 void CNT_Set( S_Counter *counter, uint32_t target )
66 {
67     counter->target = target;
68 }
69
70 /*****
71
72 void CNT_Reset( S_Counter *counter )
73 {

```

```
74     counter->value = SYS_counter;
75 }
76
77 /*****
78
79 void CNT_CallBack( void )
80 {
81     SYS_counter++;
82 }
83
84 *****/
85
86 /* End of File *****/
87
```