```c
/******************************************************************************
 *
 *    ________  _________  ____    ____  _____              ________   ______
 *   |_   __  ||  _   _  ||_   \  /   _||_   _|        |_   __  |.' ____ \
 *     | |_ \_||_/ | | \_|  |   \/   |    | |     ______ | |_ \_|| (___ \_|
 *     |  _| _     | |      | |\  /| |    | |   _|______||  _| _  _.____`.
 *    _| |__/ |   _| |_    _| |_\/_| |_  _| |__/ |      _| |__/ || \____) |
 *   |________|  |_____|  |_____||_____||________|     |________| \______.'
 *
 ******************************************************************************
 *
 * File          : counter.c
 * Version       : 1.0
 *
 ******************************************************************************
 *
 * Description   : Managing time counters for application
 *                 based on timer (1 by default)
 *
 ******************************************************************************
 *
 * Author        : Miguel Santos
 * Date          : 14.09.2023
 *
 ******************************************************************************
 *
 * MPLAB X       : 5.45
 * XC32          : 2.50
 * Harmony       : 2.06
 *
 *****************************************************************************/

#include "counter.h"

/*****************************************************************************/

/* Global infinite system counter */
uint32_t SYS_counter = 0;

/*****************************************************************************/

void CNT_Initialize(S_Counter *counter, uint32_t target)
{
    counter->value = SYS_counter;
    counter->target = target;
}

/*****************************************************************************/

bool CNT_Check( S_Counter *counter )
{
    bool checkStatus;

    checkStatus = ((SYS_counter - counter->value) >= counter->target);

    if(checkStatus)
    {
        CNT_Reset(counter);
    }

    return checkStatus;
}

/*****************************************************************************/

void CNT_Set( S_Counter *counter, uint32_t target )
{
    counter->target = target;
}

/*****************************************************************************/

void CNT_Reset( S_Counter *counter )
{
```

```
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
```