```
/* ************************************************************** */
/* File name:        tc_hal.c                                     */
/* File description: This file has a couple of useful functions to   */
/*                   timer and counter hardware abstraction layer    */
/*                                                                */
/* Author name:      dloubach                                     */
/* Creation date:    23out2015                                    */
/* Revision date:    25fev2016                                    */
/* ************************************************************** */

#include "lptmr.h"

/* system includes */
#include "fsl_lptmr_driver.h"

#include "fsl_clock_manager.h"
#include "fsl_port_hal.h"
#include "fsl_gpio_hal.h"


/* LPTMR configurations */
lptmr_user_config_t lptmrConfig =
{
    .timerMode              = kLptmrTimerModeTimeCounter,
    .freeRunningEnable      = false,
    .prescalerEnable        = true,
    .prescalerClockSource   = kClockLptmrSrcLpoClk,
    .prescalerValue         = kLptmrPrescalerDivide2,
    .isInterruptEnabled     = true,
};

/* LPTMR driver state information */
lptmr_state_t lptmrState;


/* LPTMR IRQ handler that would cover the same name's APIs in startup code */
/* Do not edit this part */
void LPTMR0_IRQHandler(void)
{
    LPTMR_DRV_IRQHandler(0U);
}



/* ********************************************** */
/* Method name:        tc_installLptmr            */
/* Method description: Low power timer 0          */
/*                     initialization and start   */
/* Input params:       uiTimeInUs:                */
/*                     time in micro seconds      */
/*                     tUserCallback              */
/*                     function pointer to be called*/
/*                     when counter achieves      */
/*                     uiTimeInUs                 */
/* Output params:      n/a                        */
/* ********************************************** */
void tc_installLptmr0(uint32_t uiTimeInUs, lptmr_callback_t tUserCallback)
{
    /* Initialize LPTMR */
    LPTMR_DRV_Init(LPTMR0_IDX, &lptmrState, &lptmrConfig);

    /* Set timer period for TMR_PERIOD micro seconds */
    LPTMR_DRV_SetTimerPeriodUs(LPTMR0_IDX, uiTimeInUs);

    /* Install interrupt call back function for LPTMR */
    LPTMR_DRV_InstallCallback(LPTMR0_IDX, tUserCallback);
```

```
    /* Start LPTMR */
    LPTMR_DRV_Start(LPTMR0_IDX);
}
```