

```

/*
*
* \filename      : app.c
* \version       : V1.00
* \author        : Otto Parzhuber
*
* Beispiel für Blinklicht
*
* Heartbeat = 200 Hz = 5 ms (oc_cfg)
*
*/

#include <includes.h>

static OS_STK App_TaskStartStk[APP_CFG_TASK_START_STK_SIZE];
static void App_TaskCreate (void);
static void App_TaskStart (void *p_arg);

int main (void)
{
    CPU_Init(); // 1) Einstellungen für die CPU
    BSP_Init(); // 2) Initialisieren des Board Support Package

    OSInit(); // 3) Initialisieren des "uC/OS-II, The Real-Time Kernel"
    OSTaskCreateExt((void (*)(void *))App_TaskStart, // 4) Start Task erzeugen
                   (void *)0,
                   (OS_STK *)&App_TaskStartStk[APP_CFG_TASK_START_STK_SIZE - 1],
                   (INT8U )APP_CFG_TASK_START_PRIO,
                   (INT16U )APP_CFG_TASK_START_PRIO,
                   (OS_STK *)&App_TaskStartStk[0],
                   (INT32U )APP_CFG_TASK_START_STK_SIZE,
                   (void *)0,
                   (INT16U )(OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR));
    OSStart(); // 5) Starten des Betriebssystem
    return(1); // 6) wird nie erreicht
}

/**
* Start Task
*
*/
static void App_TaskStart (void *p_arg)
{
    BSP_Start(); // 7) Einstellen einiger Parameter des BSP

    #if (OS_TASK_STAT_EN > 0)
        OSStatInit(); //8) Funktion für Bestimmen der CPU Auslastung
    #endif

    App_TaskCreate(); //9) Hier werden alle NutzerTasks erzeugt

    LPC_GPIO0->FIODIR |= (1<<10) | (1<<16) | (1<<17) | (1<<18) | (1<<19)
                       | 1<<20);

    while (1) { //10) Alle tasks laufen in einer Endlosschleife
        LPC_GPIO0->FIOPIN ^= (1<<10);
    }
}

```

```

        OSTimeDlyHMSM(0, 0, 5,0);
    }
}

/*
 * Task 1 mit Stack Deklaration
 */
static OS_STK Task1Stk[128];

static void Task1(void *p_arg)
{
    LPC_GPIO0->FIODIR |= (1<<20);
    while(1){
        OSTimeDly(4);
        LPC_GPIO0->FIOPIN |= (uint32_t)(1<<20);
    }
}

/*
 * Task 2 mit Stack Deklaration
 */

static OS_STK Task2Stk[128];

static void Task2(void *p_arg)
{
    LPC_GPIO0->FIODIR |= (1<<20);
    while(1){
        LPC_GPIO0->FIOPIN &= ~(uint32_t)(1<<20);
        OSTimeDly(20);
    }
}

static void App_TaskCreate (void)
{
    OSTaskCreateExt((void (*)(void *))Task1,           //Taskname
                   (void *)0,                         //nicht verwendet
                   (OS_STK *)&Task1Stk[127],         //Zeiger auf das
                                                           // letzte Element des Stacks
                   (INT8U)15,                          //Priorität
                   (INT16U)15,                        //Id, wird nicht verwendet
                   (OS_STK *)&Task1Stk[0],           //Zeiger auf das
                                                           //erste Element des Stacks
                   (INT32U)128,                       //Größe des Stacks
                   (void *)0,                         //nicht verwendet
                   (INT16U)(OS_TASK_OPT_STK_CHK |
                           OS_TASK_OPT_STK_CLR));

    OSTaskCreateExt((void (*)(void *))Task2,
                   (void *)0,
                   (OS_STK *)&Task2Stk[127],
                   (INT8U)16,
                   (INT16U)16,

                   (OS_STK *)&Task2Stk[0],
                   (INT32U)128,
                   (void *)0,
                   (INT16U)(OS_TASK_OPT_STK_CHK |
                           OS_TASK_OPT_STK_CLR));
}

```