1- The new ready list (xReadyTasksListEDF) is declared

```
/* ============================ Code Added by Abdallah Salem ======================== */
/* Declare the new ready list */
#if ( configUSE_EDF_SCHEDULER == 1 )

    PRIVILEGED_DATA static List_t xReadyTasksListEDF;   /* Ready tasks ordered by their deadline */

#endif /* if ( configUSE_EDF_SCHEDULER == 1 ) */

/* ============================ End of Code Added by Abdallah Salem ================== */
```

2- prvInitialiseTaskLists ( ) method modification in order to add the initialization of xReadyTasksListEDF

```

    /* ==================== Code Added by Abdallah Salem ================== */
    /* Adding the initialization of xReadyTasksListEDF    */
    #if ( configUSE_EDF_SCHEDULER == 1)
    {
        vListInitialise( &xReadyTasksListEDF );
    }
    #endif
    /* ==================== End of Code Added by Abdallah Salem ============ */

```

3- prvAddTaskToReadyList ( ) method modification

```

/* ============================ Code Modified by Abdallah Salem ========================= */

/*
 * Place the task represented by pxTCB into the appropriate ready list for
 * the task.  It is inserted at the end of the list.
 */
#if configUSE_EDF_SCHEDULER == 0
    #define prvAddTaskToReadyList( pxTCB )                                                  \
        traceMOVED_TASK_TO_READY_STATE( pxTCB );                                            \
        taskRECORD_READY_PRIORITY( ( pxTCB )->uxPriority );                                 \
        listINSERT_END( &( pxReadyTasksLists[ ( pxTCB )->uxPriority ] ), &( ( pxTCB )->xStateListItem ) ); \
        tracePOST_MOVED_TASK_TO_READY_STATE( pxTCB )
#else
    #define prvAddTaskToReadyList( pxTCB ) /*xStateListIteam must contain the deadline value */ \
        traceMOVED_TASK_TO_READY_STATE(pxTCB);                                              \
        vListInsert( &(xReadyTasksListEDF), &( ( pxTCB )->xStateListItem ) );

#endif
/*-----------------------------------------------------------*/

/* ========================== End of Code Modified by Abdallah Salem ======================== */
```

4- A new variable is added to the tskTaskControlBlock structure (TCB)

```
367
368     /* ================== Code Added by Abdallah Salem ==================== */
369     /* The period of the task */
370     #if ( configUSE_EDF_SCHEDULER == 1 )
371         TickType_t xTaskPeriod;   /* stores the period in tick of the task */
372     #endif
373     /* ================== End of Code Added by Abdallah Salem ============== */
374
```

5- Create the method xTaskPeriodicCreate

```
769 /* ============================== Code Added by Abdallah Salem ===============
770
771 /* Create the method xTaskPeriodicCreate */
772 #if ( configUSE_EDF_SCHEDULER == 1 )
773
774     BaseType_t xTaskPeriodicCreate( TaskFunction_t pxTaskCode,
775                                 const char * const pcName, /*lint !e971 Unqualifi
776                                 const configSTACK_DEPTH_TYPE usStackDepth,
777                                 void * const pvParameters,
778                                 UBaseType_t uxPriority,
779                                 TaskHandle_t * const pxCreatedTask,
780                 TickType_t period )   /* the same parameters for normal task cr
781     {
782         TCB_t * pxNewTCB;   /* declaring a pointer of type TCB for the task t
783         BaseType_t xReturn;   /* declaring a variable to hold the return valu
784
```

```
856     /*E.C. : initialize the period */
857     pxNewTCB->xTaskPeriod = period;
858
859     /*E.C. : insert the period value in the generic/State list iteam before to add the task in RL: */
860     listSET_LIST_ITEM_VALUE( &( ( pxNewTCB )->xStateListItem ), ( pxNewTCB )->xTaskPeriod + xTaskGetTickCount() );
861
862     /* Add the task to the ready list */
863     prvAddNewTaskToReadyList( pxNewTCB );
```

6- Modification of IDLE Task Management throughout modifying the vTaskStartScheduler ( ) method

```
2192            #if (configUSE_EDF_SCHEDULER == 1)
2193            {
2194                TickType_t initIDLEPeriod = 300;  /* The larger number of period
2195                xReturn = xTaskPeriodicCreate( prvIdleTask,
2196                      "IDLE",
2197                      configMINIMAL_STACK_SIZE,
2198                      (void * ) NULL,
2199                      ( tskIDLE_PRIORITY | portPRIVILEGE_BIT ),
2200                      NULL,initIDLEPeriod );
2201            }
2202            #else
2203            {
2204                /* The Idle task is being created using dynamically allocated RAI
2205                xReturn = xTaskCreate( prvIdleTask,
2206                                       configIDLE_TASK_NAME,
2207                                       configMINIMAL_STACK_SIZE,
2208                                       ( void * ) NULL,
2209                                       portPRIVILEGE_BIT,  /* In effect ( tskIDLE_
2210                                       &xIdleTaskHandle ); /*lint !e961 MISRA exce
2211            }
2212            #endif
```

7- Modificatino in context switch mechanism throughout vTaskSwitchContext ( ) method to update the *pxCurrentTCB pointer to the new running task

```
3270
3271    /* ============================== Code Added by Abdallah Salem ============================== */
3272    #if (configUSE_EDF_SCHEDULER == 0)
3273    {
3274        taskSELECT_HIGHEST_PRIORITY_TASK();
3275    }
3276    #else
3277    {
3278          pxCurrentTCB = (TCB_t * ) listGET_OWNER_OF_HEAD_ENTRY( &(xReadyTasksListEDF ) );
3279      traceTASK_SWITCHED_IN();
3280    }
3281    #endif
3282    /* ============================== End of Code Added by Abdallah Salem ======================== */
```