```c
/*
 * Task.c
 *
 *  Created on: May 19, 2019
 *      Author: julian
 */




#include "Task1.h"

#include "General.h"

/*Datos de Trama Recibida*/
volatile DataFrame_t Data;

/*Datos de trama para decodificar */
volatile Frame_parameters_t Frame_parameters = { '{',0 , {0,0}, NULL,NULL, '}' };

/*Semaforo Sincronizar UartTx*/
SemaphoreHandle_t SemTxUart;

/*Semaforo Sincronizar UartRx*/
SemaphoreHandle_t SemRxUart;

/*Semaforo Mutex proteger Uart*/
SemaphoreHandle_t SemMutexUart;

/*instanciar Driver memoria dinamica*/
Module_Data_t ModuleData;

/*Notificación para llegada de trama*/
TaskHandle_t xTaskHandle_RxNotify = NULL;

/*Puntero para crear la cola*/
QueueHandle_t xPointerQueue_OP0;

/*Puntero para crear la cola*/
QueueHandle_t xPointerQueue_OP1;

/*Puntero para crear la cola*/
QueueHandle_t xPointerQueue_OP2;

/*Puntero para crear la cola*/
QueueHandle_t xPointerQueue_OP3;

/*Puntero para crear la cola*/
QueueHandle_t xPointerQueue_3;

/**/
TaskHandle_t xTaskHandle_MayOP0;

/**/
TaskHandle_t xTaskHandle_MinOP1;


/*============================================================================
                                | Tarea  |
 ============================================================================*/

void TaskService( void* taskParmPtr ){
    while(TRUE) {
        /*Notifica que llego trama Buena*/
        xTaskNotifyWait(0,0,NULL,portMAX_DELAY);
```

```
        Service(&ModuleData);
        gpioToggle( LEDB );
    }
}

/*==================================================================================
                                | Tarea Mayusculizar |
 ==================================================================================*/
void Task_ToMayusculas_OP0( void* taskParmPtr ){
    char * rx;
    while(1){
        rx = ModuleDinamicMemory_receive(&ModuleData,xPointerQueue_OP0,  portMAX_DELAY);
        packetToUpper(rx);
        // Enviar a cola de TaskTxUARt
        ModuleDinamicMemory_send2(&ModuleData,rx,0,NULL,rx, xPointerQueue_3,portMAX_DELAY);
    }
}

/*==================================================================================
                                | Tarea Minusculizar |
 ==================================================================================*/
void Task_ToMinusculas_OP1( void* taskParmPtr ){
    char * rx;
    while(1){
        rx = ModuleDinamicMemory_receive(&ModuleData,xPointerQueue_OP1,  portMAX_DELAY);
        packetToLower(rx);
        // Enviar a cola de TaskTxUARt
        ModuleDinamicMemory_send2(&ModuleData,rx,0,NULL,rx, xPointerQueue_3,portMAX_DELAY);
    }
}

/*==================================================================================
                                | Tarea Reportar stack disponible |
 ==================================================================================*/
void Task_ReportStack_OP2( void* taskParmPtr ){
    while(1){
        Report(&ModuleData,xPointerQueue_OP2,STACK_);
    }
}
/*==================================================================================
                                | Tarea Reportar heap disponible |
 ==================================================================================*/
void Task_ReportHeap_OP3( void* taskParmPtr ){
    while(1){
        Report(&ModuleData,xPointerQueue_OP3,HEAP_);
    }
}

/*==================================================================================
                                | Tarea tx |
 ==================================================================================*/
void TaskTxUart( void* taskParmPtr ){
    char * BSend;
    char Txbuffer[100];
    while(true){

        /*Recibe por la cola*/
        BSend = ModuleDinamicMemory_receive(&ModuleData, xPointerQueue_3, portMAX_DELAY);
        gpioToggle( LED3 );
        if( uartTxReady( UART_USB ) ){
            sprintf( Txbuffer, "%s",BSend);
            //Transmit_UART( 0 );   // La primera vez – con esto arranca
            uartWriteString(UART_USB,Txbuffer);
        }
        ModuleDinamicMemory_Free(&ModuleData, BSend);
```

```
    }
}

/*==================================================================================
                                  | Callback IT RX |
 ==================================================================================*/
void CallbackRx( void *noUsado ){

    UBaseType_t uxSavedInterruptStatus;
    BaseType_t xHigherPriorityTaskWoken = pdFALSE;

    volatile char c = uartRxRead( UART_USB );  /*Char received*/

    /*Funcion pertenece al driver*/
    ModuleData.Add_IncommingFrameFunction(uxSavedInterruptStatus ,xHigherPriorityTaskWoken,c);

    if(xHigherPriorityTaskWoken) portYIELD_FROM_ISR( xHigherPriorityTaskWoken );
}
/*==================================================================================
                        | Callback IT TX | - 24.5.2019
                      readBuffer(char *buffer, char *ByteToTx);
 ==================================================================================*/
void Transmit_UART ( void* noUsado ){
    static int start_detected = 0;
    char Txbyte;
    //if( readBuffer( &Txbuffer, &Txbyte ) )
    uartTxWrite( UART_USB, Txbyte );
}
```