```c
#include "DriverDinamicMemoryRTOS.h"

#include "Task1.h"

void ModuleDinamicMemory_initialize( Module_Data_t *obj , uint32_t MaxLength, xQueueSendFCN
xQueueSendFCN,xQueueSendFromISRFCN xQueueSendFromISRFCN, xQueueReceiveFCN xQueueReceiveFCN,
xQueueCreateFCN xQueueCreateFCN, pvPortMallocFCN pvPortMallocFCN,vPortFreeFCN vPortFreeFCN,
Add_IncommingFrameFCN Add_IncommingFrameFCN){
    obj->xMaxStringLength = MaxLength;
    obj->xQueueCreateFunction =xQueueCreateFCN;
    obj->xQueueSendFunction = xQueueSendFCN;
    obj->xQueueSendFromISRFunction = xQueueSendFromISRFCN;
    obj->xQueueReceiveFunction = xQueueReceiveFCN;
    obj->pvPortMallocFunction = pvPortMallocFCN;
    obj->vPortFreeFunction = vPortFreeFCN;
    obj->Add_IncommingFrameFunction = Add_IncommingFrameFCN;
}

void ModuleDinamicMemory_send( Module_Data_t *obj ,uint8_t Isr, long * const xHigherPriorityTaskWoken
, char* pbuf ,char * XpointerQueue, uint32_t portMaxDelay){
    static char* PcStringToSend = NULL;
    if (PcStringToSend == NULL) PcStringToSend = obj->pvPortMallocFunction( obj->xMaxStringLength );
    if(PcStringToSend != NULL) strcpy(PcStringToSend ,pbuf);

    /*Si uso el enviar en una isr*/
    if(Isr) obj->xQueueSendFromISRFunction(XpointerQueue ,&PcStringToSend,xHigherPriorityTaskWoken, 0
    );
    else  obj->xQueueSendFunction(XpointerQueue ,&PcStringToSend,portMaxDelay, 0);
}

void ModuleDinamicMemory_send2( Module_Data_t *obj ,char *PcStringToSend, uint8_t Isr, long * const
xHigherPriorityTaskWoken, char* pbuf ,char * XpointerQueue, uint32_t portMaxDelay){
    if(PcStringToSend != NULL) strcpy(PcStringToSend ,pbuf);

    /*Si uso el enviar en una isr*/
    if(Isr) obj->xQueueSendFromISRFunction(XpointerQueue ,&PcStringToSend,xHigherPriorityTaskWoken, 0
    );
    else  obj->xQueueSendFunction(XpointerQueue ,&PcStringToSend,portMaxDelay, 0);
}

char* ModuleDinamicMemory_receive(Module_Data_t *obj ,char * XpointerQueue, uint32_t portMaxDelay){

    char* pbuffer; /*Dato recibido*/
    obj->xQueueReceiveFunction(XpointerQueue , &pbuffer, portMaxDelay );
    return pbuffer;
}

void ModuleDinamicMemory_Free(Module_Data_t *obj , char *ultimo_mensaje){
    obj->vPortFreeFunction(ultimo_mensaje);
}
```