Rafael ya había hecho esta parte
en el driver.h se cambia el pvportmalloc y free por :

```
typedef void (*MemoryFreeFCN)(void *);

typedef void* (*MemoryAllocFCN)(uint64_t);
```

en la estryctura tambien :

```
    MemoryAllocFCN MemoryAllocFunction;
    MemoryFreeFCN MemoryFreeFunction;
```

en la inicialización tambien.

```
MemoryAllocFCN  MemoryAllocFCN,                                    MemoryAllocFCN  MemoryAllocFCN,
MemoryFreeFCN  MemoryFreeFCN,
```

en driver .c

```
        obj->MemoryAllocFunction = MemoryAllocFCN;
        obj->MemoryFreeFunction = MemoryFreeFCN;
```

en el freeRTOS2 . c

```
if (PcStringToSend == NULL) PcStringToSend = obj->MemoryAllocFunction( obj->xMaxStringLength );
```

PEro acá es donde se debe hacer lo del wrapper.

==========el prototipo de Qmpoll es así: ==========

```
void *         QMPool_get (QMPool *const me, uint_fast16_t const margin)

void         QMPool_put (QMPool *const me, void *b)
```

==========WRAPPERS==========

wrapper masomenos sería algo así

```
void *         Memory_get (QMPool *const me, uint_fast16_t const margin, uint64_t tamañomalloc,
uint8_t  QmpoolOrMalloc){
        qmpool  pool; // la sintaxys debe estar mal
        char * poolmalloc;
        if(QmpoolOrMalloc){
                pool = QMPool_get (me,margin);

        }else {
                poolmalloc = pvportmalloc(tamañomalloc);


        }

        return QmpoolOrMalloc ? pool :poolmalloc ;  // si es 1 retorno pool si no retorno malloc
```

```
}
void         Memory_Free (QMPool * const me, void *b, uint8_t interrupcion ,void *
ptrfreemalloc,uint8_t  QmpoolOrfree){


        if(QmpoolOrfree){
                QMPool_put(me, b,interrupcion );

        }else {
                pvportfree(ptrfreemalloc);


        }

}


================================================================================================
===============


en INICIALIZAR EL DRIVER SE PASA ESTA FUNCIÓN Memory_get  O Memory_Free  y en el driver


Si es con malloc así
obj->MemoryAllocFunction ( NULL, 0,obj->xMaxStringLength , 0)


si es con qmpool así
obj->MemoryAllocFunction ( me, 0,NULL , 1); // habría que prpbar pero masomenos es así  lo mismo
para free




/* ejemplo de qmpool


void * block1 = QMPool_get (& myMemPool1, 0U); / * afirma en la piscina vacía * /
/ * Se garantiza que block1 no es NULL * /
~ ~ ~
QMPool_put (& myMemPool1, block1);
void * block2 = QMpool_get (& myMemPool2, 5U); / * versión no afirmativa * /
if (block2! = ( void *) 0) { / * la asignación tuvo éxito? * /
   ~ ~ ~
}
~ ~ ~
QMPool_put (& myMemPool2, block2);


*/
```