

TRABAJO FINAL – RTOS 2 – CESE
GRUPO 1 v22-06-2019

TP3 (prelim.)

INTEGRANTES:

Julian Bustamante Narvaez
Jacobó Salvador
Gustavo Paredes D.
Rafael Oliva

Estructura TP3 / Grupo 1 RTOS2

1.0) Estructura y diagrama conceptual

Paquetes de datos: (en negrita los campos diferentes de la práctica 1 y 2)



Todos los caracteres son ASCII legibles.

Delimitación de paquete:

SOF: Carácter '{'
EOF: Carácter '}'.

Campos:

OP (Operación):

"0": Convertir los caracteres recibidos a mayúsculas. (CMD/RTA)

"1": Convertir los caracteres recibidos a minúsculas. (CMD/RTA)

"2": Reportar stack disponible (RTA)

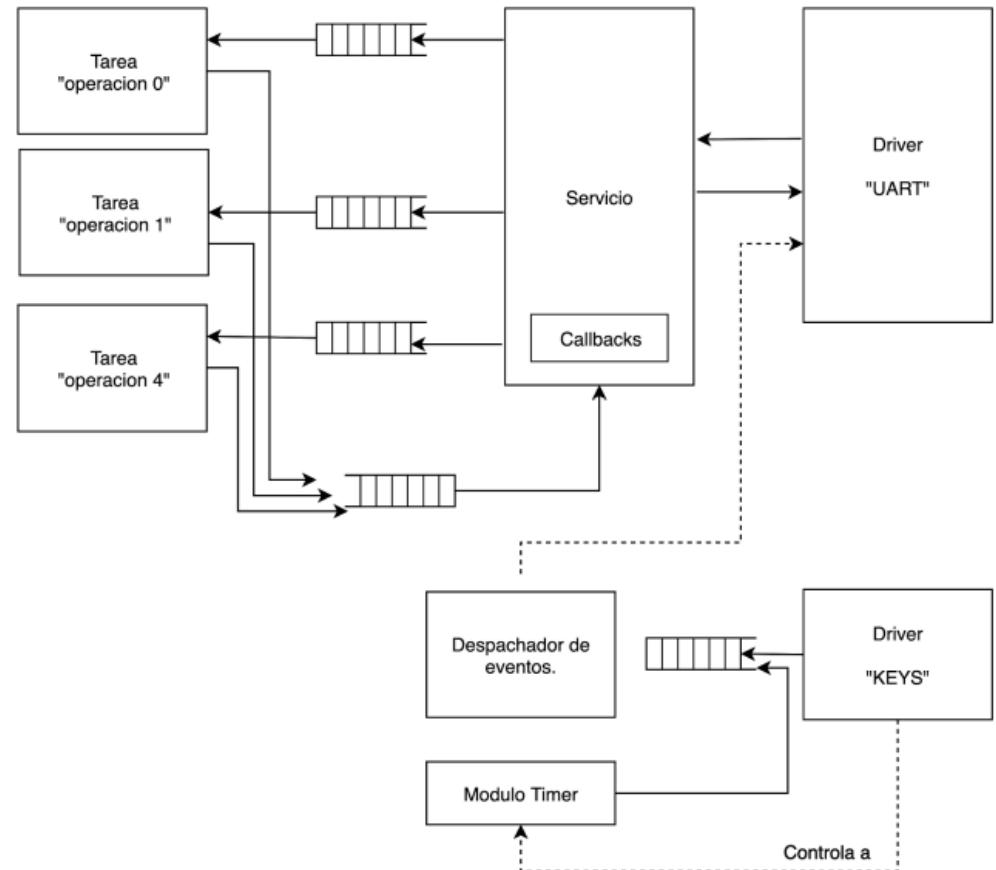
"3": Reportar heap disponible. (RTA)

"4": Medir performance del sistema (CMD/RTA)

"5": Resultado de la medición performance del sistema (RTA)

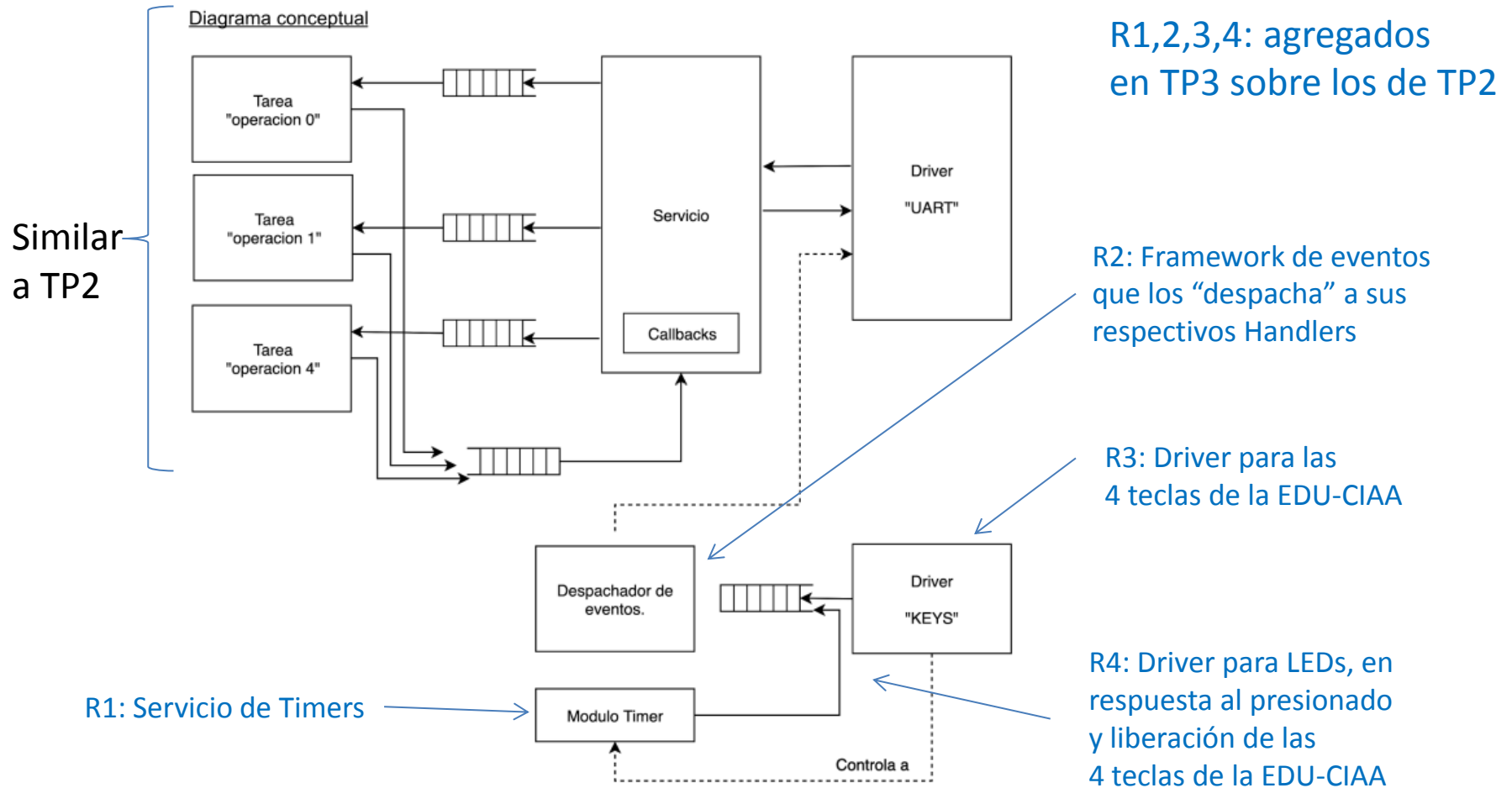
"6": Texto descriptivo de evento (RTA)

Diagrama conceptual



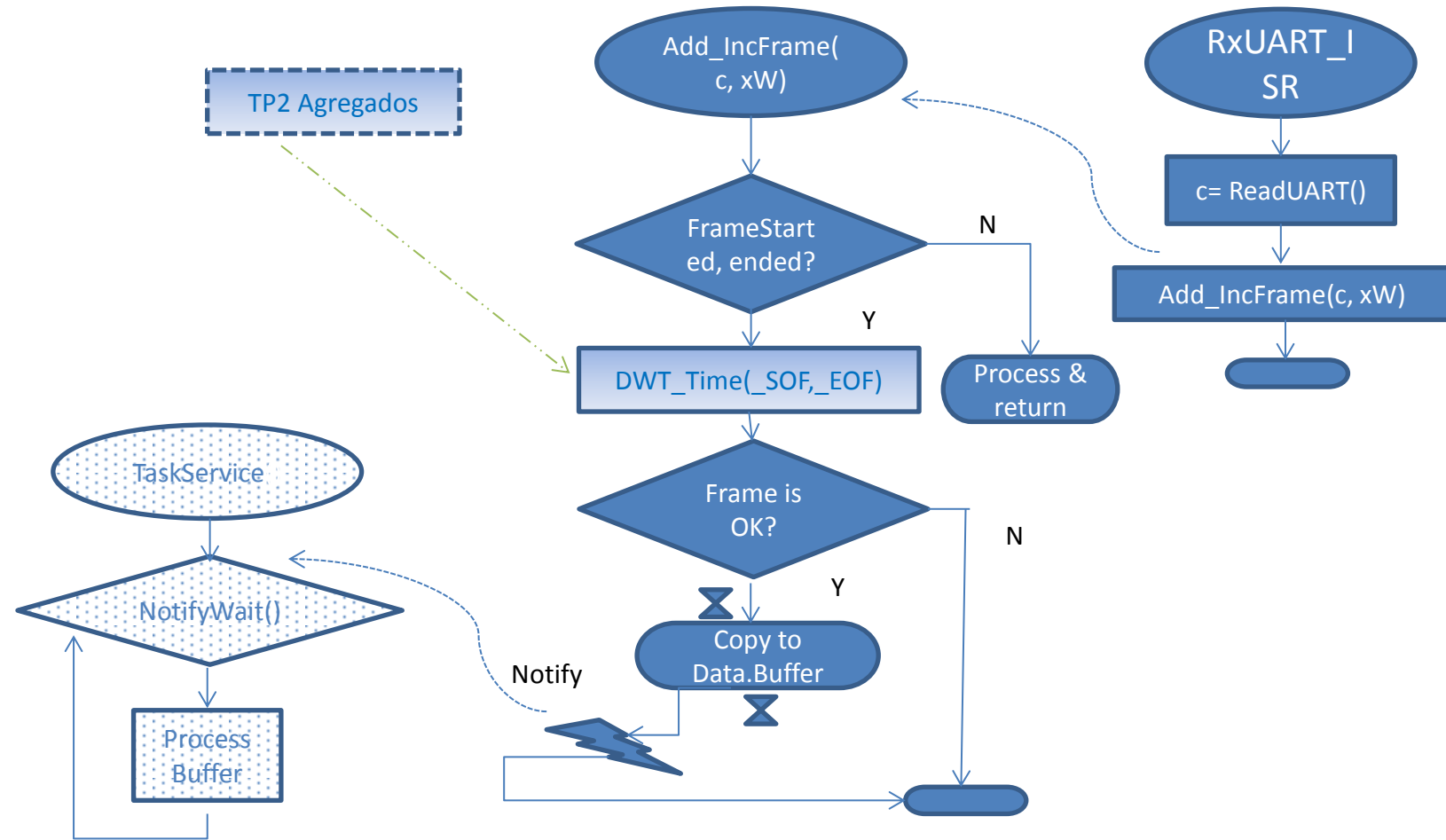
Estructura TP3 / Grupo 1 RTOS2

1.1) Estructura y diagrama conceptual



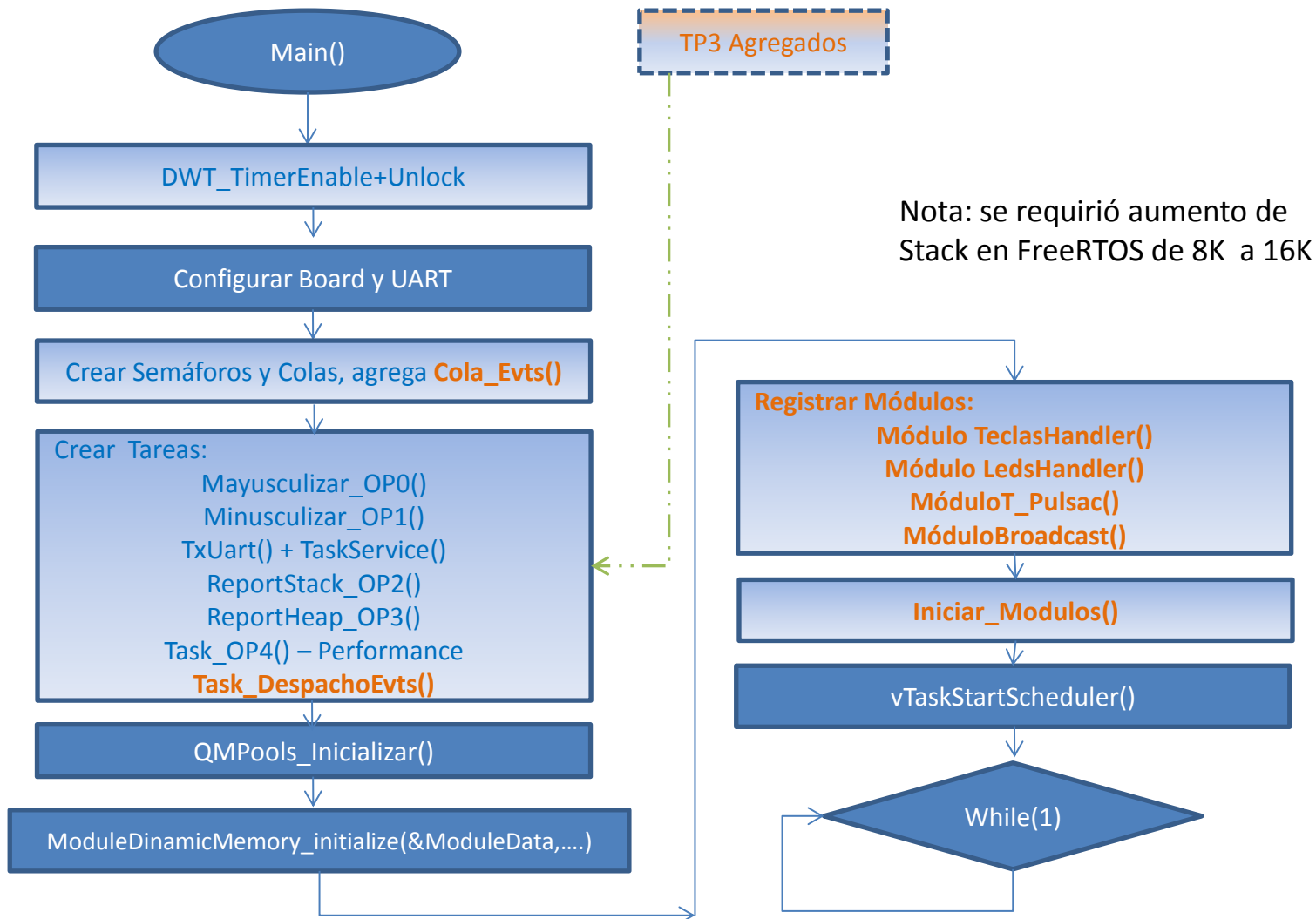
Estructura TP3 / Grupo 1 RTOS2

2.a) Ingreso caracteres dde UART, Time _SOF,_EOF idem TP2 v15.6.19



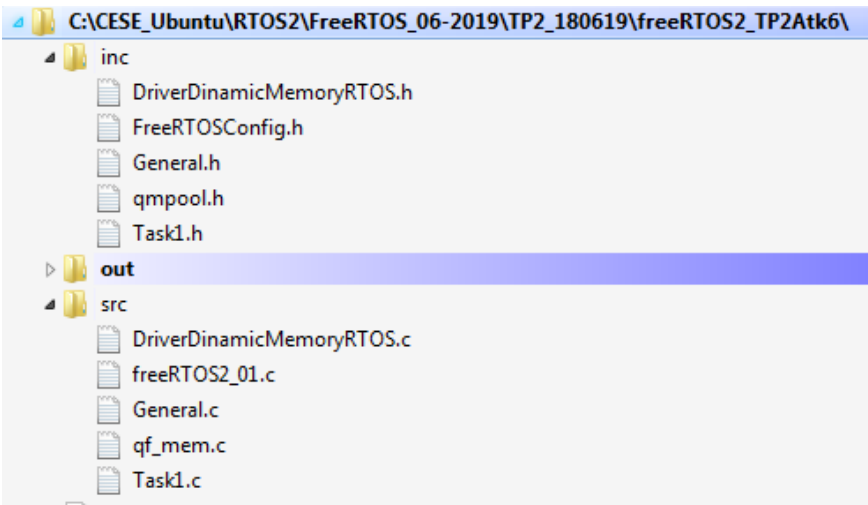
Estructura actual TP3 Main() / Grupo 1 RTOS2

2.b) TP3 Función main() Se agregan Task_DespEvt, ColaEvts, Registro 4 Módulos e Inicio



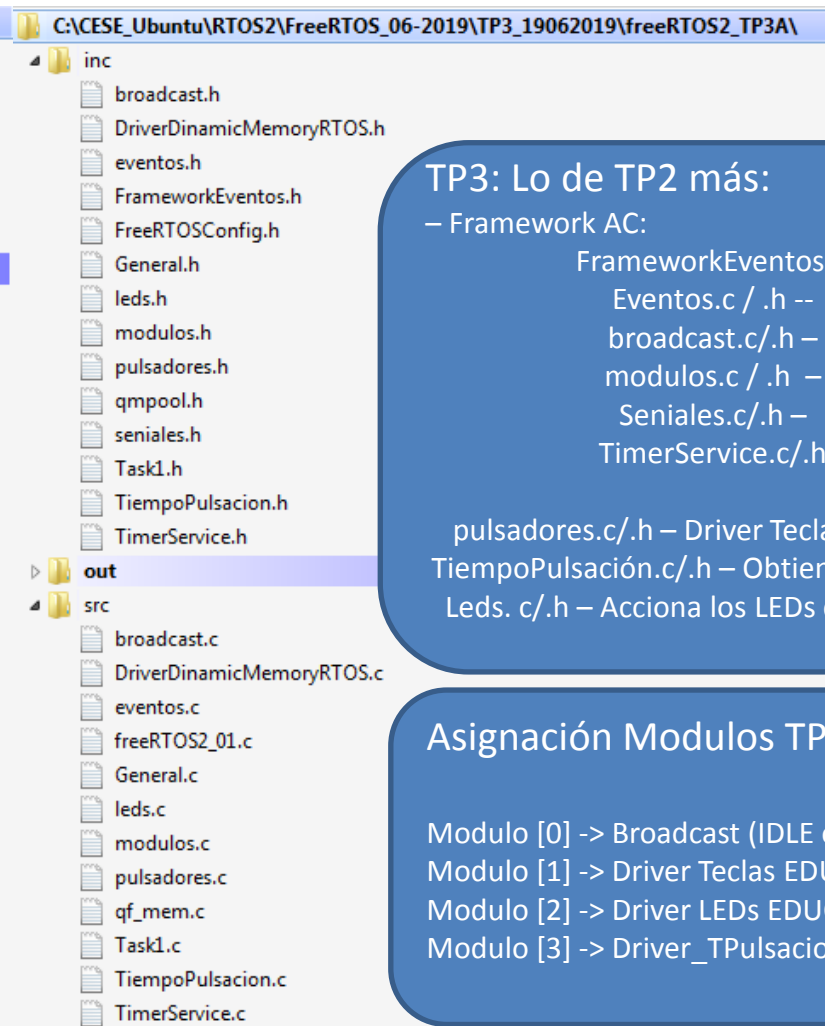
Estructura de archivos TP3 vs TP2 / Grupo 1 RTOS2

2.c) TP3 v19.6.19 comparado con TP2



TP2:

freeRTOS2_01.c – main()
Task1.c/.h – Tareas
General.c / .h – Funciones aux.
DDMRTOS.c/.h – Asignacion Mem.



TP3: Lo de TP2 más:

– Framework AC:

FrameworkEventos.h

Eventos.c / .h --

broadcast.c/.h –

modulos.c / .h –

Seniales.c/.h –

TimerService.c/.h

pulsadores.c/.h – Driver Teclas EDUCIAA
TiempoPulsación.c/.h – Obtiene TPulsacTec
Leds. c/.h – Acciona los LEDs de EDUCIAA

Asignación Modulos TP3

Modulo [0] -> Broadcast (IDLE o NORMAL)
Modulo [1] -> Driver Teclas EDUCIAA
Modulo [2] -> Driver LEDs EDUCIAA
Modulo [3] -> Driver_TPulsacionTEC1 a 4

Estructura de datos TP3 / Grupo 1 RTOS2

2.d) Versión Idem TP2 v15.6.19

TP2 – Datos Performance

```
typedef struct {  
    State_perf State-Token;  
    uint32_t Id_de_paquete;  
    char * Payload;  
    uint32_t t_sof;  
    uint32_t t_eof;  
    uint32_t t_InitConvert;  
    uint32_t t_EndConvert;  
    uint32_t t_InitTx;  
    uint32_t t_EndTx;  
    uint16_t Length_Frame;  
    uint16_t Memory_Allocated;  
    void (*Completion_HandlerFCN)( void *T, BaseType_t * xHig );  
} Token_t;
```

```
/* Medir Performance */  
typedef enum State_med {  
    Time_LL = 0,  
    Time_R,  
    Time_I,  
    Time_F,  
    Time_S,  
    Time_T  
} State_perf;  
  
static uint32_t Id_Frame = 0;
```

```
/* ===== Datos para llenar buffer local ISR===== */  
typedef struct {  
    char Buffer[106];  
    uint32_t t_sof;  
    uint32_t t_eof;  
    uint32_t Id_Frame;  
    uint8_t Ready;  
    uint8_t Index;  
    uint8_t StartFrame;  
} DataFrame_t;  
extern volatile DataFrame_t Data;
```

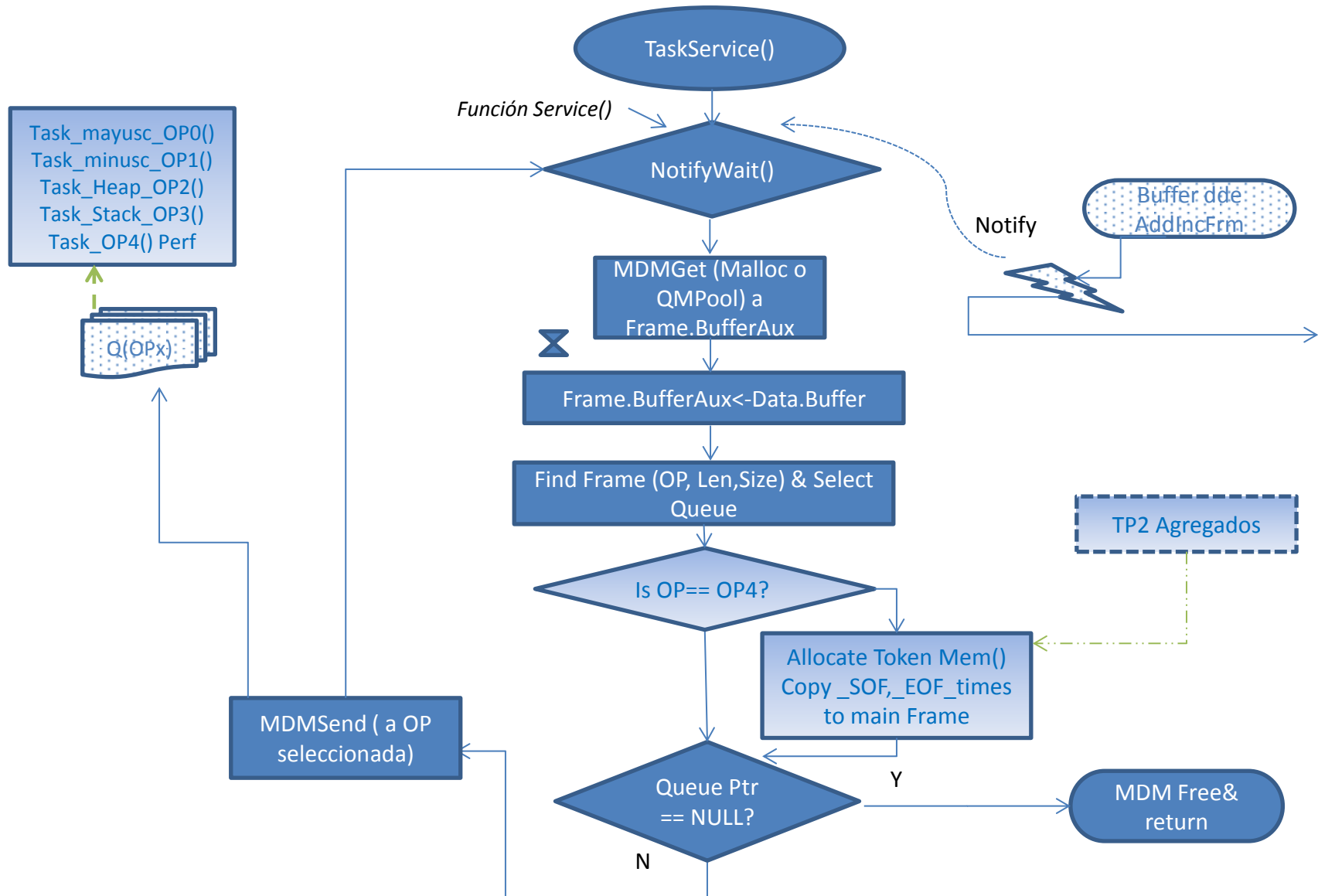
```
/* =====Parametros de la trama de llegada ===== */  
typedef struct {  
    Enum_Op_t Operation;  
    uint8_t T;  
    char* BufferAux;  
    Token_t *Token;  
} Frame_parameters_t;
```

TP2 – trama usada por ISR

TP2 – Frame c/Token procesamiento

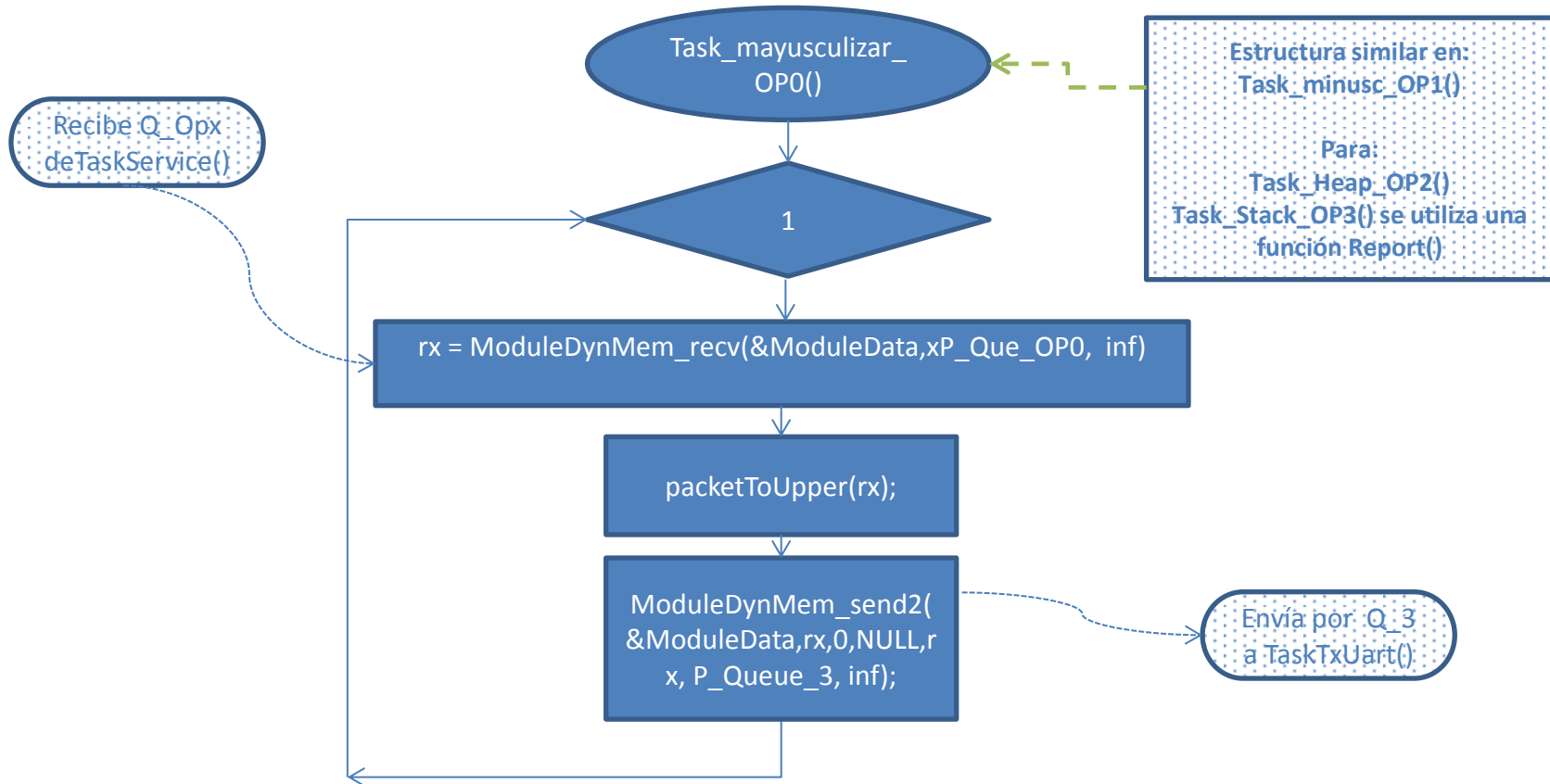
Estructura actual TP3 / Grupo 1 RTOS2

2.e) Proceso en TaskService idem TP2 v15.6.19



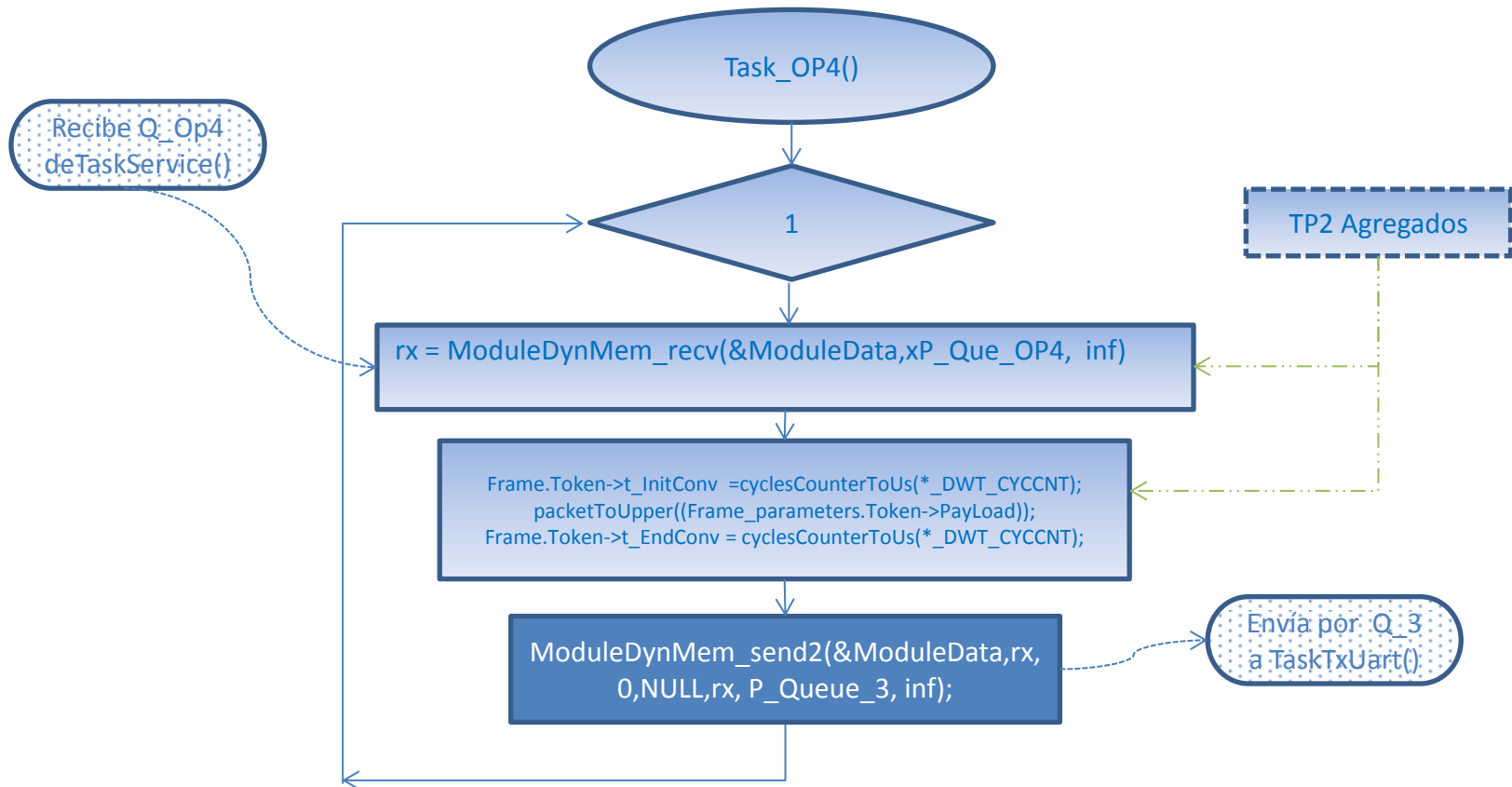
Estructura actual TP3 / Grupo 1 RTOS2

3) Formato de Tasks Mayusculizar, Minusc, Stack y Heap – Idem TP2



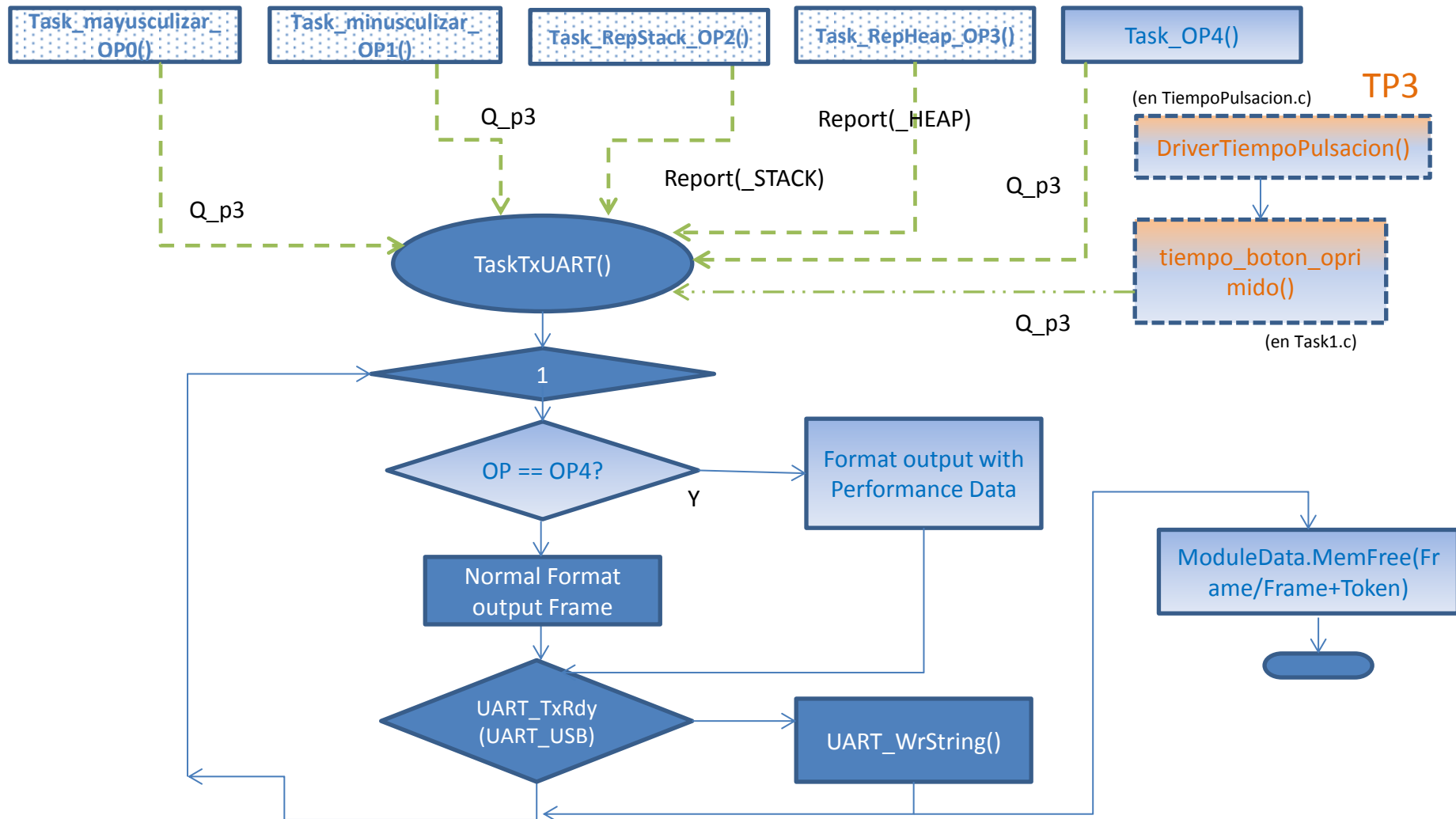
Estructura actual TP3/ Grupo 1 RTOS2

4) Formato de Task OP4 _ Mayusculizar con Medición performance (idem TP2)



Estructura actual TP3 / Grupo 1 RTOS2

5) Salida de Tasks Mayusculizar, Minusc, Stack y Heap, OP4 – agrega OP6 en TP3



Funcion envio Tpuls - TP3 / Grupo 1 RTOS2

6) Versión TP3 19.6.19

```
void DriverTiempoPulsacion ( Evento_t *evn )
```

```
{  
    //int i;
```

```
    static TickType_t contadorTick = 0;          (en TiempoPulsacion.c)
```

```
    switch( evn->signal )
```

```
{
```

```
    case SIG_MODULO_INICIAR:
```

```
        mod = (Modulo_t *) evn->receptor;
```

```
        TpoPulsadoresInit(mod);
```

```
        timerArmarUnico(mod, mod->periodo);
```

```
        break;
```

```
    case SIG_BOTON_APRETADO:
```

```
        contadorTick = xTaskGetTickCount();
```

```
        break;
```

```
    case SIG_BOTON_LIBERADO:
```

```
        contadorTick = xTaskGetTickCount() - contadorTick;
```

```
        tiempo_boton_oprimido( contadorTick, evn->valor );
```

```
        break;
```

```
    default:    // Ignoro todas las otras seniales
```

```
        break;
```

(en Task1.c)

```
}
```

```
}
```

```
void tiempo_boton_oprimido( TickType_t contadorTick, int TECid )
```

```
{
```

```
    Frame_parameters_t Frame_parameters;
```

```
    char LocalBuff[100];
```

```
    memset( LocalBuff, 0, sizeof( LocalBuff ) );
```

```
    sprintf( LocalBuff, "{OP6: TEC%d durante %d ms }", TECid, contadorTick / portTICK_RATE_MS );
```

```
    /*Proteger datos para hacer copia local*/
```

```
    taskENTER_CRITICAL();
```

```
        Frame_parameters.BufferAux = ModuleData.MemoryAllocFunction(sizeof(LocalBuff));
```

```
        strcpy((char*)Frame_parameters.BufferAux, (const char*)LocalBuff);
```

```
    taskEXIT_CRITICAL();
```

```
    ModuleDinamicMemory_send2(&ModuleData,&Frame_parameters,0,NULL,NULL, xPointerQueue_3,portMAX_DELAY);
```

```
}
```

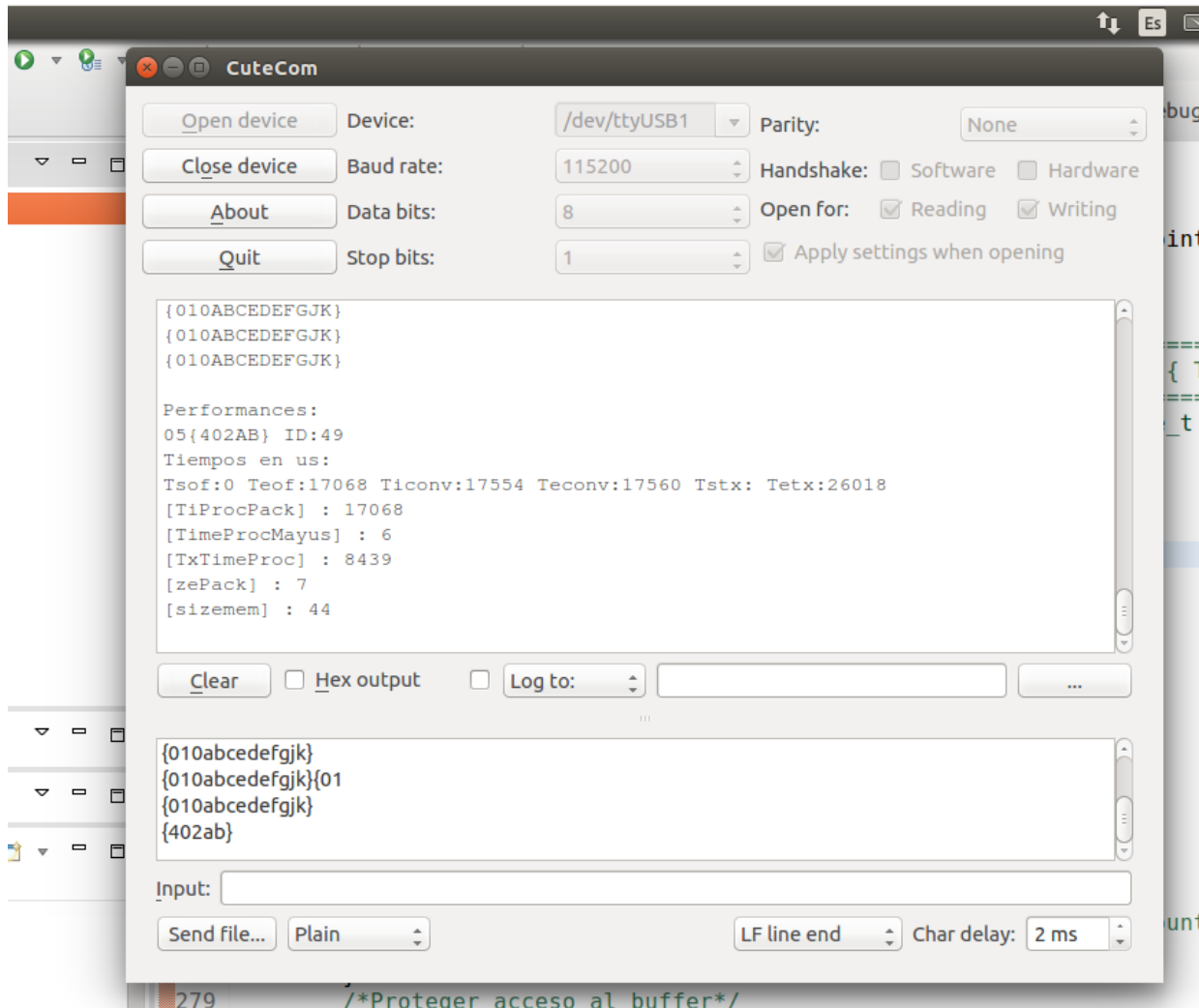
Timeout- TP3 / Grupo 1 RTOS2

7a) Agregado de Timeout para Tramas cortadas, en función
AddIncomingFrame() en general.c - TP3 22.6.19

```
1 // 20/6/19 - 22:55 Los tiempos entre Tsof y Teof pueden llegar a > 900000 us
2 // para strings de 99 - ponemos un timeout de 1500000 us
3 // Luego reducido a 1400000
4
5 AddIncomingFrame()
6 ...
7     static uint32_t Timeout_In; // Timeout frame cortados 1500ms
8     #define MAX_US_WAIT_EOF 1400000
9
10 ... al final de todo..
11 // Timeout para secuencias cortadas..
12 taskENTER_CRITICAL_FROM_ISR();
13 Timeout_In = cyclesCounterToUs(*_DWT_CYCCNT); //cyclesCounterToUs
14 Timeout_In = Timeout_In - Data.t_sof;
15 taskEXIT_CRITICAL_FROM_ISR(uxSavedInterruptStatus);
16 if(Timeout_In > MAX_US_WAIT_EOF){
17     Data.StartFrame = 0; // Reset frame reading if EOF takes too long..
18     memset(Data.Buffer,0,sizeof(Data.Buffer));
19     Data.Index = 0;
20     InitTimeFlag = 1;
21 }
22
```

Resultados - TP3 / Grupo 1 RTOS2

8a) Resultados en Cutecom / Ubuntu_ TP3 22.6.19



Resultados - TP3 / Grupo 1 RTOS2

8b) Resultados en Hercules / Windows _ TP3 22.6.19

