

FreeRTOS

Dr. Óscar Rodríguez Polo
Dr. Pablo Parra Espada
Dr. Sebastián Sánchez Prieto

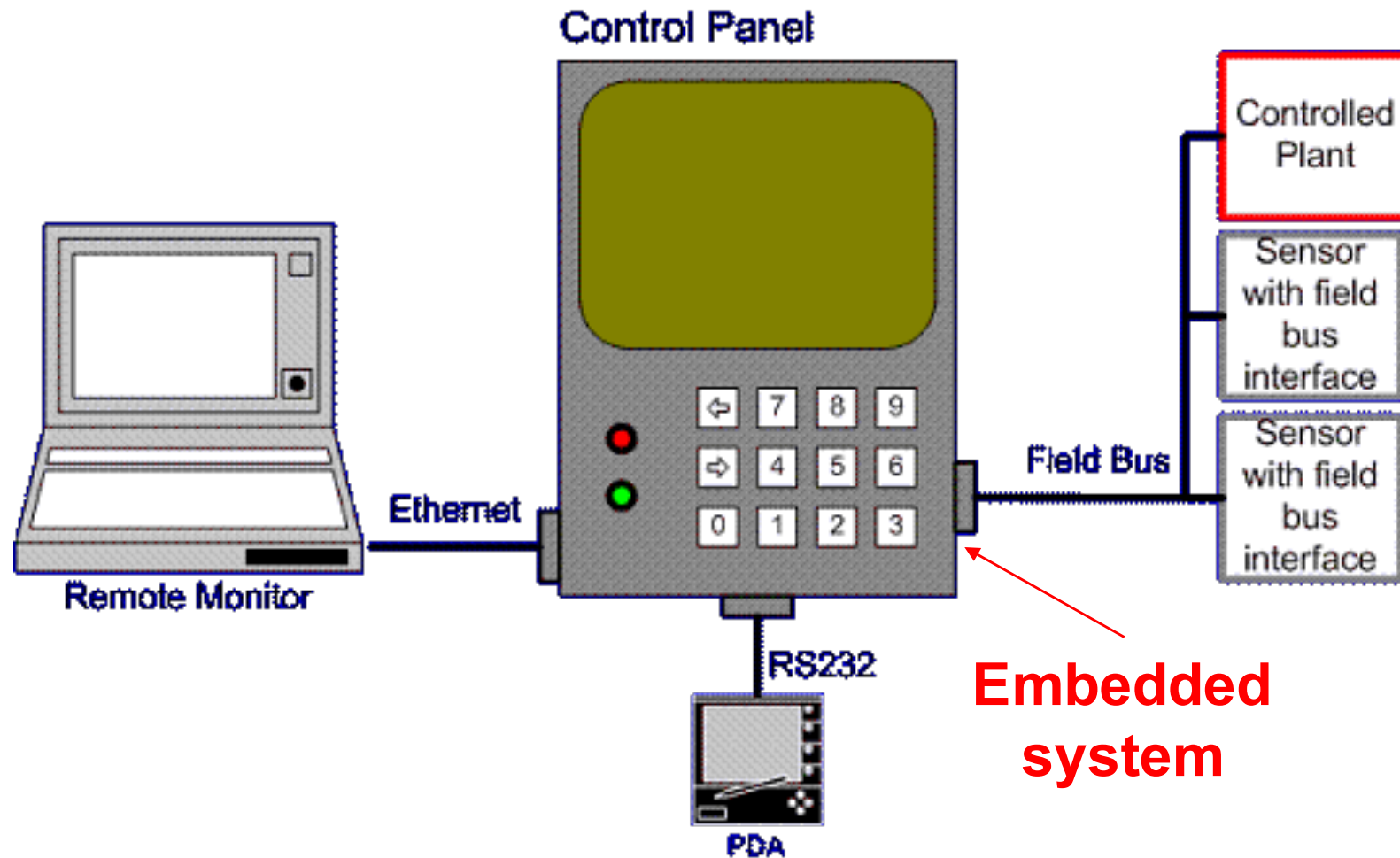
Bibliografía:

Using the FreeRTOS Real Time Kernel. A practical
Guide: Richard Barry

FreeRTOS Introducción

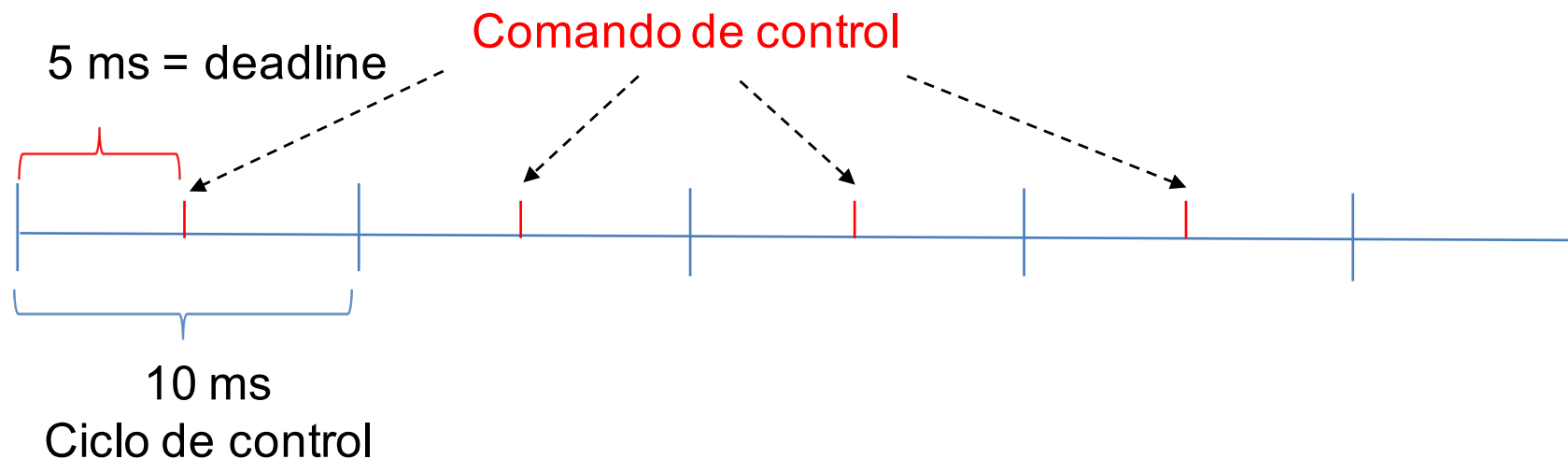
¿Por qué usar un RTOS?

Ejemplo de Sistema de Control



Requisitos del Sistema Empotrado

- Ciclo de control:
 - **Solicita datos de los sensores** vía FieldBus.
 - Espera la recepción de esos datos.
 - Ejecuta el algoritmo de control.
 - Transmite el comando de control a la planta
 - **Restricciones temporales duras**



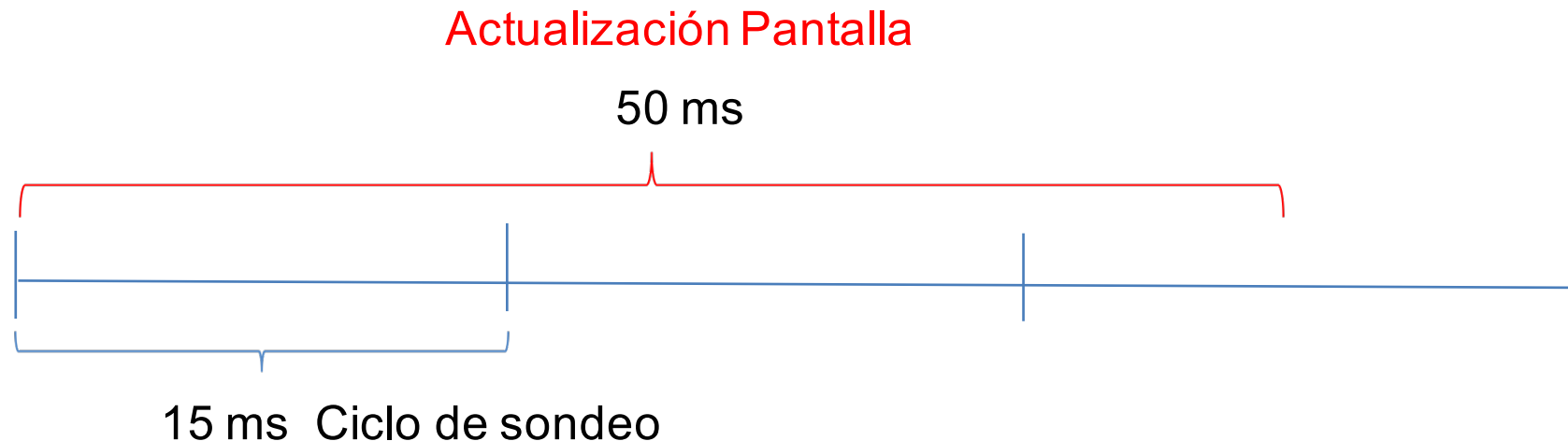
Requisitos del Sistema Empotrado

- Interface del operador local (Keyb):
 - **Sondeo** periódico del teclado (15 ms).
 - Actualización de la pantalla (retardo máximo=50ms).
 - **Jitter acotado**



Requisitos del Sistema Empotrado

- Control vía RS-232:
 - **Sondeo** periódico del comando recibido (15 ms).
 - Actualización de la pantalla = pantalla del operador.
 - **Jitter acotado**



Requisitos del Sistema Empotrado

- Atención a las peticiones de monitorización vía Ethernet:
 - Cada segundo se atienden a las peticiones.
 - **Jitter no acotado.**



1 Segundo

Requisitos del Sistema Empotrado

- Control del Led:
 - Cada segundo se emite un pulso de encendido/apagado del LED.
 - Pulso verde =OK, Pulso rojo =FAIL
 - **Jitter no acotado.**



1 Segundo

Solución con ejecutivo cíclico

```
void FPlantControl( void ) {  
    TransmitRequest();  
    WaitForFirstSensorResponse();  
  
    if( Got data from first sensor ) {  
        WaitForSecondSensorResponse();  
        if( Got data from second sensor ) {  
            PerformControlAlgorithm();  
            TransmitControl();  
        }  
    }  
}
```

Solución con ejecutivo cíclico

```
Int status=0, counter=0;
time_spec_t abs_time;
getCurrentTime(abs_time);
while (1){
    counter++;
    if (2==(counter%3) status|=EventStatus(); //Sondeo cada 15 ms
    if (1==(counter%2) ) { FPlantControl ();
    }else{ //Atención a los eventos
        if(status& EVENT_KEY) {FKeyScan();    status&= ~EVENT_KEY;}
        if(status& EVENT_232) {FRS232();  status&= ~EVENT_232;}
        if(200>=counter){
            FLED(); counter=0;
            if (Status& EVENT_TCP) {FWebServer();
                status&= ~EVENT_TCP;}
        }
    }
    Wait5msFromAbsTime(abs_time);
}
```

Solución con ejecutivo cíclico

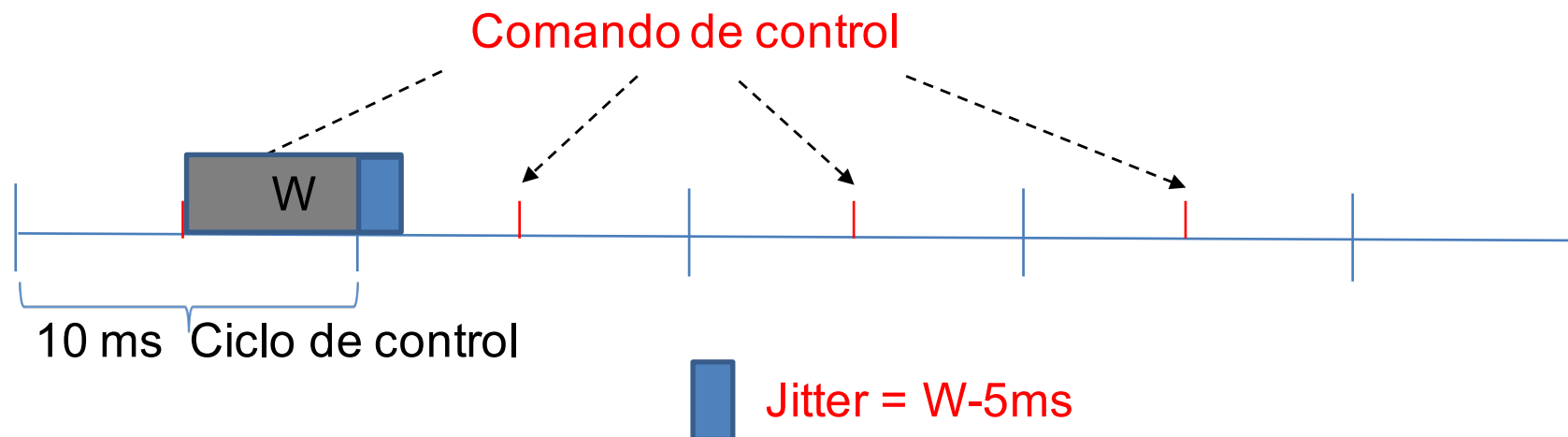
$W = (\text{WCET}(\text{EventStatus}) + \text{WCET}(\text{FKeyScan}) +$

$\text{WCET}(\text{FRS232}) + \text{WCET}(\text{FLED}) + \text{WCET}(\text{FWebServer}))$

Si $W > 5\text{ms}$

=>

!!! JITTER en FPlantControl !!!



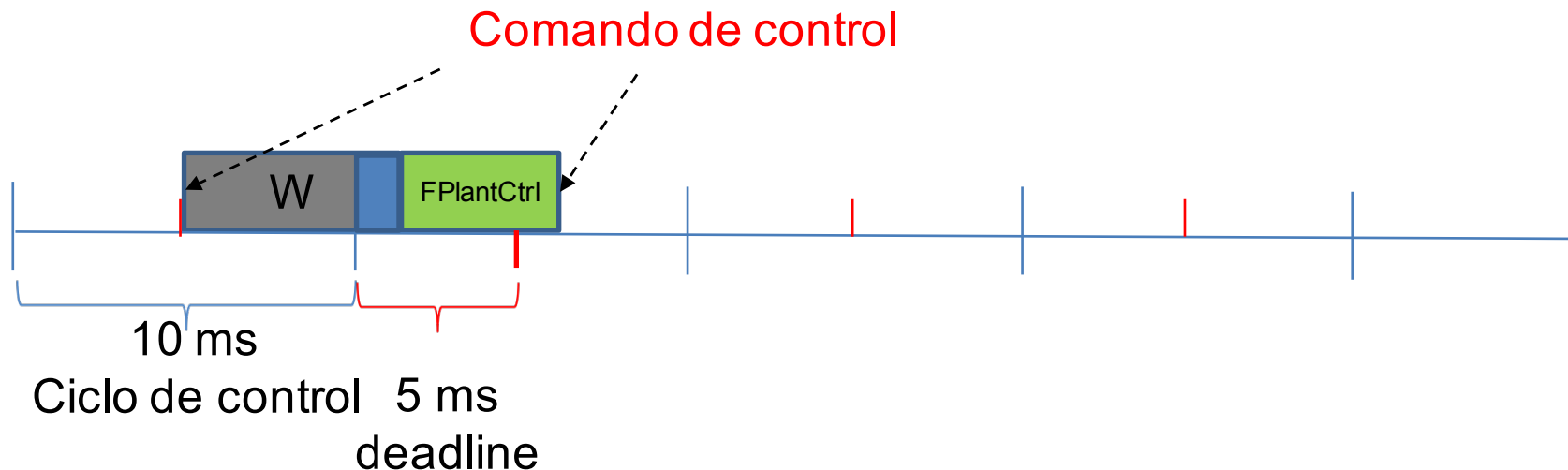
Solución con ejecutivo cíclico

SI JITTER (FPlantControl) + WCET (FPlantControl)

> 5ms

=>

!!! FPlantControl no cumple Deadline!!!



Solución con ejecutivo cíclico (2)

```
Int status=0, counter=0;
time_spec_t abs_time;
getCurrentTime(abs_time);
while (1){
```

```
    counter++;
```

```
    if (2==(counter%3) status|=EventStatus(); //Sondeo cada 15 ms
```

```
    if (1==(counter%2) ) { FPlantControl();
```

```
    }else{//Atención a los eventos
```

```
        if(status& EVENT_KEY) {FKeyScan();    status&= ~EVENT_KEY;}
```

```
        else if(status&= EVENT_232) {FRS232();  status&= ~EVENT_232;}
```

```
        else if(200>=counter){
```

```
            if (Status& EVENT_TCP) {FWebServer();
```

```
                status&= ~EVENT_TCP;}
```

```
            else {FLED(); counter=0;}
```

```
        }
```

```
    }
```

```
    Wait5msFromAbsTime(abs_time);
```

```
}
```

**W= MAX (WCET (EventStatus)
WCET (FKeyScan)
WCET (FRS232)
WCET (FLED)
WCET (FWebServer))**

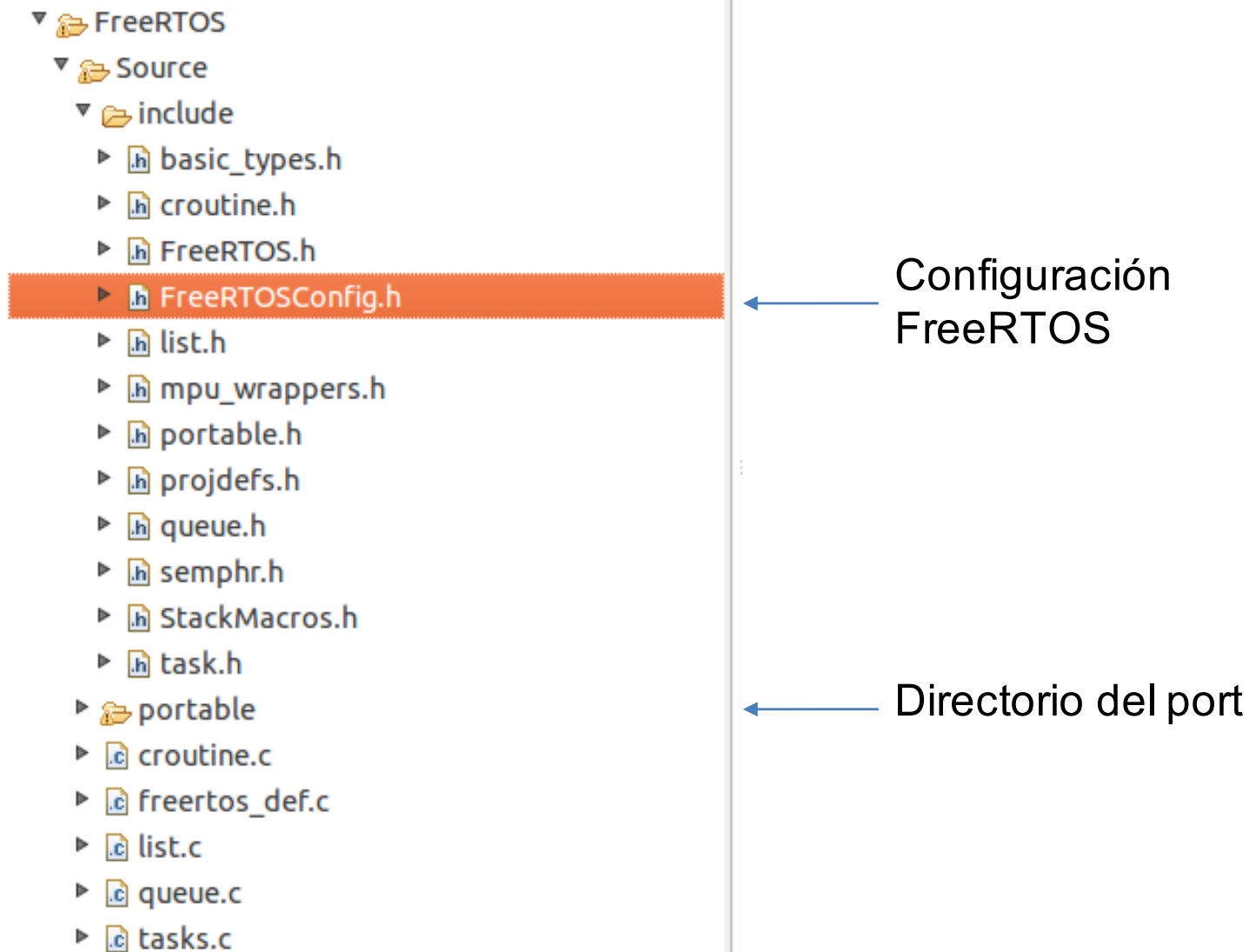
Solución Multitarea

- Tareas distintas para cada función:
 - Tarea de control de la planta, prioridad máxima=2.
 - Tareas para el control Keyb y RS232, prioridad intermedia =1
 - Tareas Background, Prioridad baja =0

Priority	Tasks
2	PlantControlTask
1	RS232Task KeyScanTask
0	IdleTask LEDTask WebServerTask

FreeRTOS Organización de Archivos

Organización archivos



Organización archivos

