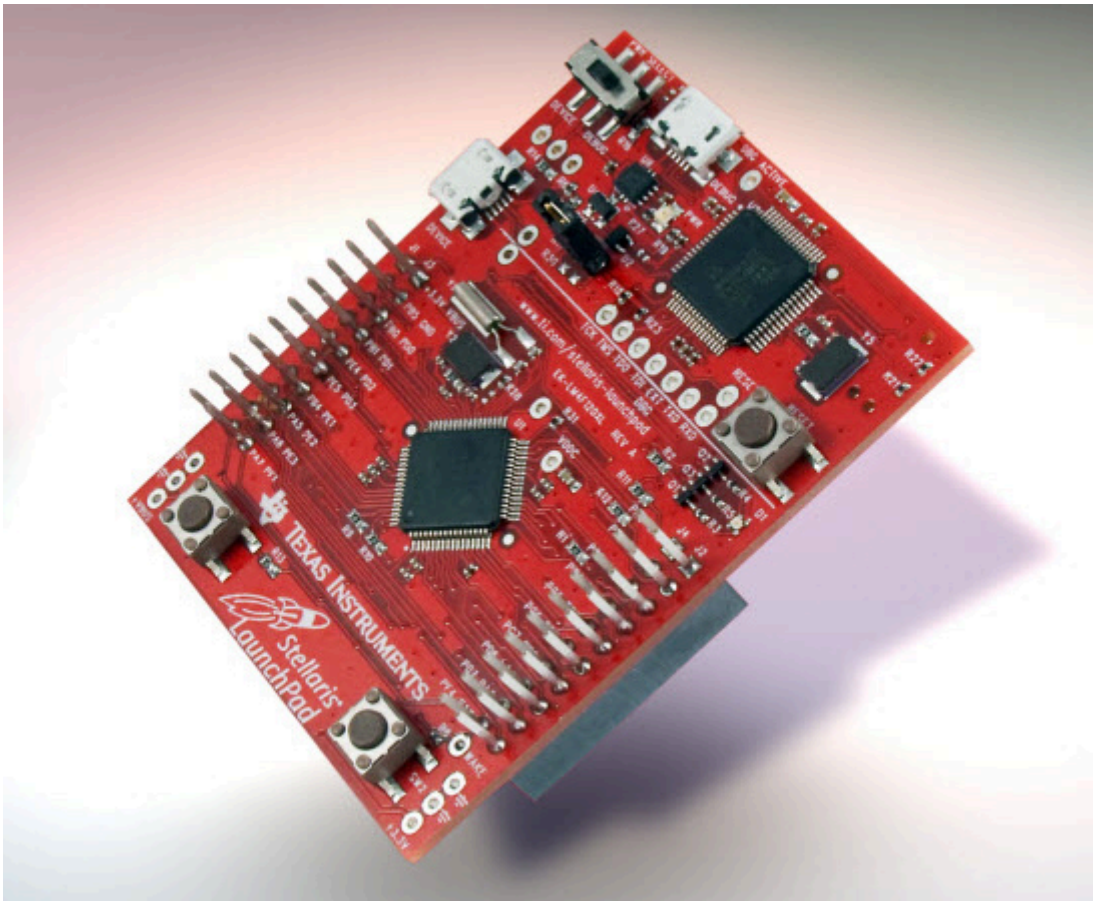
Práctica 1
SYS/BIOS



Sistemas Empotrados II

Héctor Lacueva Sacristán (869637)

12/11/2025

Apartado 1

Apartado 1.1

Instrumentar el código para que el comienzo de la actividad suponga la puesta a 1 de un pin del GPIO del procesador y el final de la actividad la puesta a cero del mismo pin. Mída el tiempo de cómputo con un osciloscopio.

No ha sido posible realizar este análisis

Apartado 1.2

Instrumentar el código con primitivas de la herramienta de análisis Duration de CCS. Mídase el tiempo de cómputo con el entorno de desarrollo.

```
C/C++
Void measure (UArg a0, UArg a1) {

    for (;;) {

        Log_write1(UIABenchmark_start, (xdc_IArg)"WCET");
        //GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_2,GPIO_PIN_2) ;

        CS (10) ;

        //GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_2,0);
        Log_write1(UIABenchmark_stop, (xdc_IArg)"WCET");

        Task_sleep(2);
    }
}
```

*Live Session | *Duration: Summary

	Source	Count	Min	Max	Average	Total	Percent
1	CORTEX_M4_0, WCET	23	10009062	10009188	10,009,114.0	230,209,623	100.0

Figura 1: Captura de pantalla del resultado obtenido con la herramienta Duration de CCS tras la ejecución del código superior.

Como se puede observar en la Figura 1, desde que comienza la medición “Log_write()”, los tiempos hasta que se ejecuta el siguiente “Log_write()” son:

- Min: 10,009062 ms
- Max: 10,009188 ms
- Media: 10,009114 ms

Se podría decir que el tiempo de cómputo es el valor máximo obtenido (WCET), por lo tanto sería **10,009188 ms**, aproximadamente **10,01ms**.

Apartado 2

El programa concurrente indep consta de tres tareas periódicas **T1, T2 y T3** activadas por retrasos que evolucionan de forma independiente (sin comunicaciones). Los períodos respectivos son: **100 ms, 150 ms y 350 ms**. El **plazo de respuesta es igual al periodo**.

Tarea	Periodo	Deadline	Tiempo de cómputo	Prioridad
T1	100 ms	100 ms	20 ms	3
T2	150 ms	150 ms	40 ms	2
T3	350 ms	350 ms	100 ms	1

Para la obtención de los tiempos de cómputo solo se han tenido en cuenta los valores de las llamadas de la función CS(), no sus mediciones ya sea con Duration u osciloscopio.

En cuanto a la asignación de prioridades se han asignado por el más frecuente (RM), que coincide que también es el más urgente (DM).

Apartado 2.1

Realizar, con los tiempos de cómputo obtenidos, el análisis temporal (demostrando que se van a cumplir, o no, los plazos correspondientes) y asignar prioridades.

Realizamos el análisis temporal:

T1

$$W_1(D_1) = C_1 = 20\,ms \Rightarrow 20\,ms \leq D_1 = 100\,ms$$

✅T1 cumple plazos.

T2

$$W_2(D_2) = \lceil \frac{D_2}{P_1} \rceil \cdot C_1 + C_2 = \lceil \frac{150\,ms}{100\,ms} \rceil \cdot 20\,ms + 40\,ms =$$
$$= 2 \cdot 20\,ms + 40\,ms = 80\,ms \Rightarrow 80\,ms \leq D_2 = 150\,ms$$

✅T2 cumple plazos.

T3

$$W_3(D_3) = \lceil \frac{D_3}{P_1} \rceil \cdot C_1 + \lceil \frac{D_3}{P_2} \rceil \cdot C_2 + C_3 =$$

$$= \lceil \frac{350\,ms}{100\,ms} \rceil \cdot 20\,ms + \lceil \frac{350\,ms}{150\,ms} \rceil \cdot 40\,ms + 100\,ms =$$

$$= 4 \cdot 20\,ms + 3 \cdot 40\,ms + 100\,ms = 300\,ms \Rightarrow 300\,ms \leq D_3 = 350\,ms$$

✅T3 cumple plazos.

Como todas las tareas cumplen plazos, la planificación es correcta.

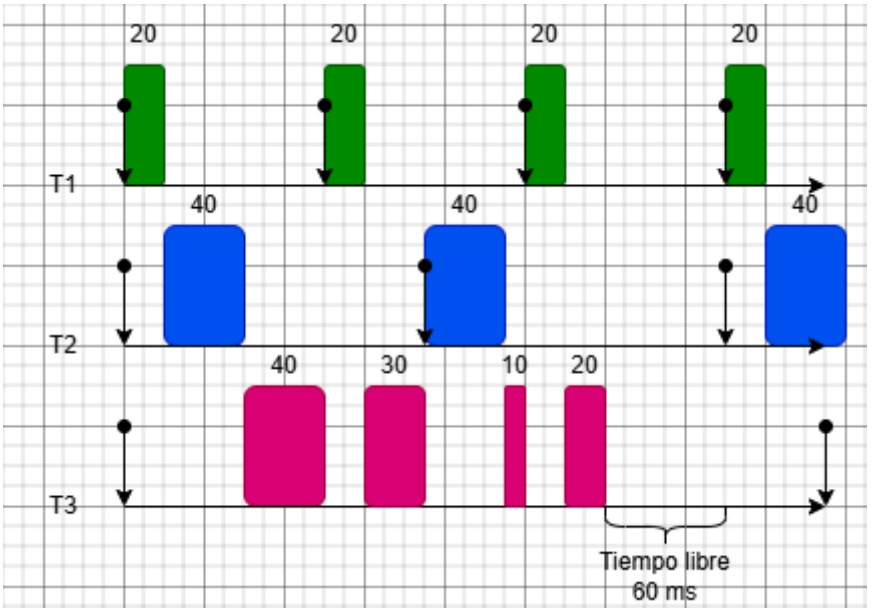


Figura 2: Análisis temporal para tres tareas, cada cuadrado representa 10 ms.

En caso de que la última tarea no se hubiese podido comprobar con el cálculo anterior, se podría calcular el **tiempo de finalización**. Como se puede ver en la Figura 2, hay **hasta 60 ms teóricos libres en el WCE**, por lo que se cumplirían los plazos.

Apartado 2.2

Verificar el cumplimiento de los plazos monitorizando la ejecución del programa mediante la herramienta Execution Graph de CCS.

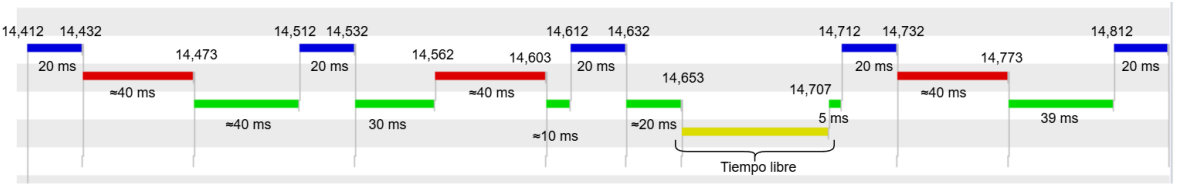


Figura 3: Captura de pantalla del Execution Graph de CCS tras ejecutar el programa con la configuración expuesta al comienzo de este apartado. Las trazas azules corresponden a la tarea 1, las trazas rojas a la tarea 2, las verdes la tarea 3 y las amarillas se corresponden a tiempo libre.

Como se puede observar en la Figura 3, todas las tareas empiezan y terminan, cuando una tarea de prioridad mayor se activa, se le cede el procesador, realiza su trabajo y se reanuda la ejecución con la tarea de mayor prioridad disponible.

Apartado 2.3

Invertir las prioridades de las tareas y comprobar sobre una traza del programa los problemas que aparecen.

Si cambiamos las prioridades, esto es, T1 tiene ahora prioridad 1 y T3 tiene prioridad 3:

Figura 4: Captura de pantalla del Execution Graph de CCS tras ejecutar el programa con las prioridades invertidas. Las trazas verdes corresponden a la tarea 1, las rojas a la tarea 2, las amarillas a la tarea 3 y las azules a tiempo libre.

Como era de esperar, al invertir las prioridades se han producido **vencimientos de plazos** en la **tarea 1**. Si la tarea 1 se activa antes de que la tarea 3 ejecute los primeros 20 ms no podría cumplir plazos.

Apartado 3

El programa que se debe analizar en este apartado es el siguiente:

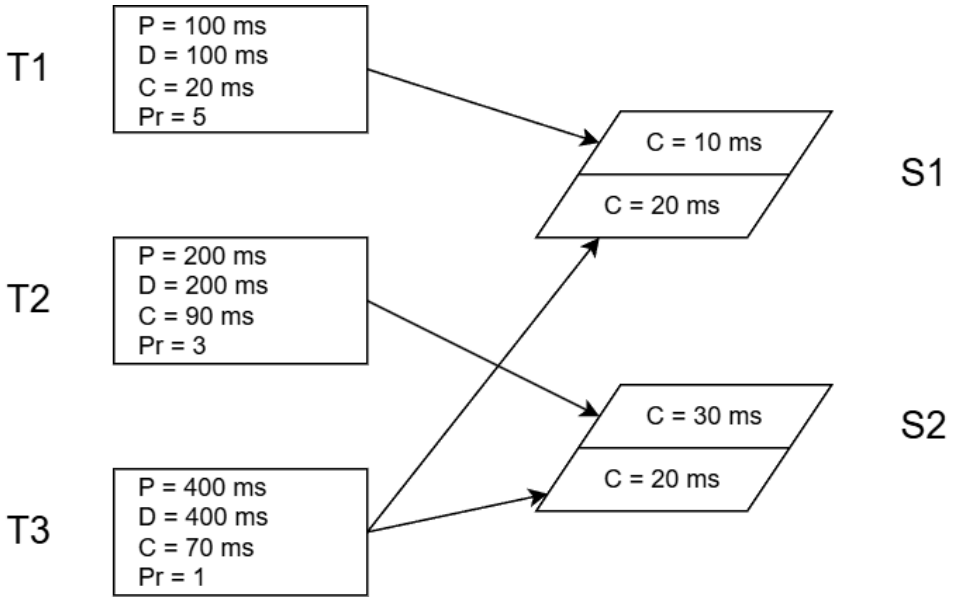


Figura 5: Diagrama del programa concurrente, a la izquierda en rectángulos están las tareas T1, T2, T3 con su información y a la derecha los "servidores" S1 y S2, con sus distintos servicios ofrecidos.

Apartado 3.1

Realizar el análisis temporal (demostrando que se van a cumplir, o no, los plazos correspondientes que son iguales a los periodos) y asignar prioridades. Calcular los tiempos de finalización de cada tarea.

Tarea	Periodo	Deadline	Tiempo de cómputo	Prioridad	S1	S2
T1	100 ms	100 ms	20 ms	5	10 ms	
T2	200 ms	200 ms	90 ms	3		30 ms
T3	400 ms	400 ms	70 ms	1	20 ms	20ms
Techo de prioridad					5	3

Primero calculamos los **tiempos de bloqueo** teniendo en cuenta el protocolo de herencia de prioridad.

Tarea	Tiempo de bloqueo (herencia de prioridad)
T1	$b_1 = min(20\,ms,\ 20\,ms) = 20\,ms$
T2	$b_2 = min(20\,ms,\ 40\,ms) = 20\,ms$
T3	$b_3 = 0\,ms$

T1

$$w_2(D_1) = C_1 + b_1 = 20\,ms + 20\,ms = 40\,ms \leq D_1 = 100\,ms$$

✅T1 cumple plazos.

$$tf_1 = C_1 + b_1 = 20\,ms + 20\,ms = 40ms$$

T2

$$w_2(D_2) = \left\lceil \frac{D_2}{P_2} \right\rceil \cdot C_1 + C_2 + b_2 =$$

$$= \left\lceil \frac{200\,ms}{100ms} \right\rceil \cdot 20\,ms + 90\,ms + 20ms = 150\,ms$$

$$150\,ms \leq D_2 = 200\,ms$$

✅T2 cumple plazos.

$$t_{f2} \Rightarrow w_2^0 = C_2 + b_2 = 90 + 20 = 110\,ms$$

$$w_2^1 = \left\lceil \frac{w_2^0}{P_1} \right\rceil C_1 + C_2 + b_2 = \left\lceil \frac{110}{100} \right\rceil 20 + 90 + 20 =$$

$$= 40 + 90 + 20 = 150\,ms$$

$$w_2^2 = \left\lceil \frac{w_2^1}{P_1} \right\rceil C_1 + C_2 + b_2 = \left\lceil \frac{150}{100} \right\rceil 20 + 90 + 20 = 150\,ms$$

T3

$$W_3(D_3) = \left\lceil \frac{D_3}{P_1} \right\rceil \cdot C_1 + \left\lceil \frac{D_3}{P_2} \right\rceil C_2 + C_3 + b_3 =$$

$$= \left\lceil \frac{400}{100} \right\rceil \cdot 20 + \left\lceil \frac{400}{200} \right\rceil \cdot 90 + 70 + 0 =$$

$$= 80 + 180 + 70 = 330 \leq D_3 = 400ms$$

✅T3 cumple plazos.

$$t_3 \Rightarrow w_3^0 = C_3 + b_3 = 70 + 0 = 70\,ms$$

$$w_3^1 = \left\lceil \frac{w_3^0}{P_1} \right\rceil C_1 + \left\lceil \frac{w_3^0}{P_2} \right\rceil C_2 + C_3 + b_3 = \left\lceil \frac{70}{100} \right\rceil 20 + \left\lceil \frac{70}{200} \right\rceil 90 + 70 + 0 = 20 + 90 + 70 = 180ms$$

$$w_3^2 = \left\lceil \frac{w_3^1}{P_1} \right\rceil C_1 + \left\lceil \frac{w_3^1}{P_2} \right\rceil C_2 + C_3 + b_3 = \left\lceil \frac{180}{100} \right\rceil 20 + \left\lceil \frac{180}{200} \right\rceil 90 + 70 + 0 = 40 + 90 + 70 = 200ms$$

$$w_3^3 = \left\lceil \frac{w_3^2}{P_1} \right\rceil C_1 + \left\lceil \frac{w_3^2}{P_2} \right\rceil C_2 + C_3 + b_3 = \left\lceil \frac{200}{100} \right\rceil 20 + \left\lceil \frac{200}{200} \right\rceil 90 + 70 + 0 = 40 + 90 + 70 = 200ms.$$

Apartado 3.2

Verificar el cumplimiento de los plazos y los tiempos de finalización sobre una traza del programa, utilizando el protocolo de herencia de prioridad (GateMutexPri). Dibujar la traza sobre el papel y señalar las inversiones de prioridad que se producen y su duración.

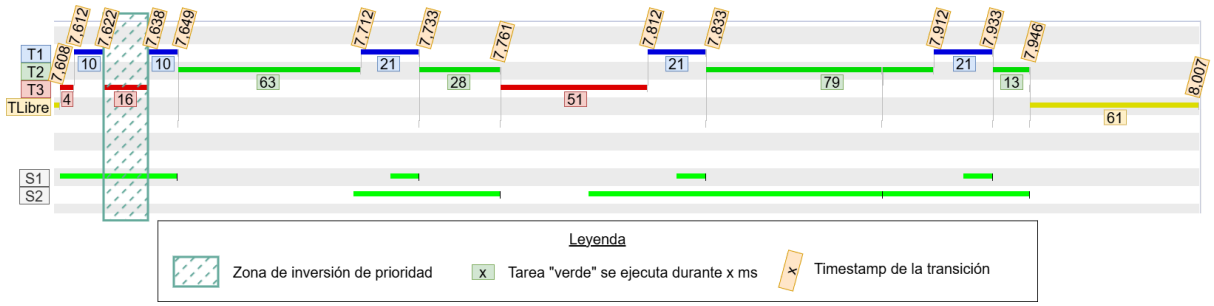


Figura 6: Captura de pantalla del Execution Graph de CCS al ejecutar el proyecto PR13 GateMutexPri.

T1

- **Cumplimiento de plazos**:
  - Desde activación, todas tardan menos de 100 ms.
- **Tiempo de finalización**:
  - El máximo tiempo hasta que una activación se completa es de **37 ms**, que está dentro del límite teórico.

T2

- **Cumplimiento de plazos**:
  - Desde activación, todas tardan menos de 200 ms.
- **Tiempo de finalización**:
  - El máximo tiempo hasta que una activación se completa es de **149 ms (suponiendo activación más temprana 7,612**, Figura 6**)**, que está dentro del límite teórico.

T3

- **Cumplimiento de plazos**:
  - Desde activación, todas tardan menos de 400 ms.
- **Tiempo de finalización**:
  - El máximo tiempo hasta que una activación se completa es de **204 ms**, es superior al tiempo de finalización teórico pero tiene explicación. **En el cálculo teórico no se han tenido en cuenta los cambios de contexto y sumado al error de la herramienta, es un valor válido.**

Inversiones de prioridad

Se ha producido una única inversión de prioridad en lo capturado en la Figura 6. La ejecución comienza con la tarea 3 accediendo al servidor S1, tras 4 ms se activa la tarea 1 y se le cede la CPU. **Cuando la tarea 1 llega a la SC se queda bloqueada debido a que la tarea 3 estaba primero (Inversión de prioridad)**. Se ejecuta la tarea 3 con la prioridad de la tarea 1 y, al acabar se reanuda la ejecución de la tarea 1.

Apartado 3.3

Cambiar a unos mutex que no tengan el protocolo de herencia de prioridad (GateMutex). Comprobar sobre la traza los tiempos de finalización y las inversiones de prioridad.

Los mutex GateMutex son mutex que no tienen en cuenta prioridades, esto implica que las tareas se ejecutan por orden de prioridad. Incluso si una tarea de prioridad 1 se encuentra en una SC y una de prioridad 5 quiere acceder a la SC deberá esperar también a activaciones de tareas de prioridad 2-4 hasta la salida de la SC de la tarea de prioridad 1. Esto implica inversiones de prioridad largas (incluso interminables) como se verá a continuación.

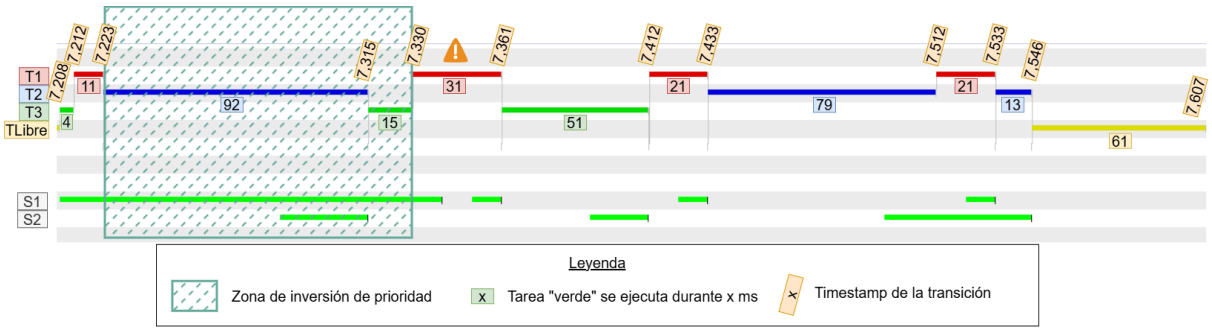


Figura 7: Captura de pantalla del Execution Graph de CCS al ejecutar el proyecto PR13 GateMutex.

T1

- **Cumplimiento de plazos**:
  - No cumple, en la primera activación, se completa la tarea en más de 100 ms.
- **Tiempo de finalización**:
  - El máximo tiempo hasta que una activación se completa es de **128 ms**. **No cumple**.

T2

- **Cumplimiento de plazos**:
  - Desde activación, todas tardan menos de 200 ms.
- **Tiempo de finalización**:
  - El máximo tiempo hasta que una activación se completa es de **134 ms**. **Cumple**.

T3

- **Cumplimiento de plazos**:
  - Desde activación, todas tardan menos de 400 ms.
- **Tiempo de finalización**:
  - El máximo tiempo hasta que una activación se completa es de **204 ms**. **Cumple**.

Inversiones de prioridad

Se ha producido una única inversión de prioridad en lo capturado en la Figura 7. La ejecución comienza con la tarea 3 accediendo al servidor S1, tras 4 ms se activa la tarea 1 y se le cede la CPU. **Cuando la tarea 1 llega a la SC se queda bloqueada debido a que la tarea 3 estaba primero (Inversión de prioridad)**. En este caso no se hereda la prioridad de la tarea 1 y, como la tarea 2 está activada y tiene mayor prioridad que la tarea 3, se ejecuta la tarea 2, al acabar se reanuda la ejecución de la tarea 3 y al acabar se reanuda la tarea 1. **Esta inversión de prioridad provoca el incumplimiento de plazos de la tarea 1**.