

# Desarrollo de software para sistemas empotrados

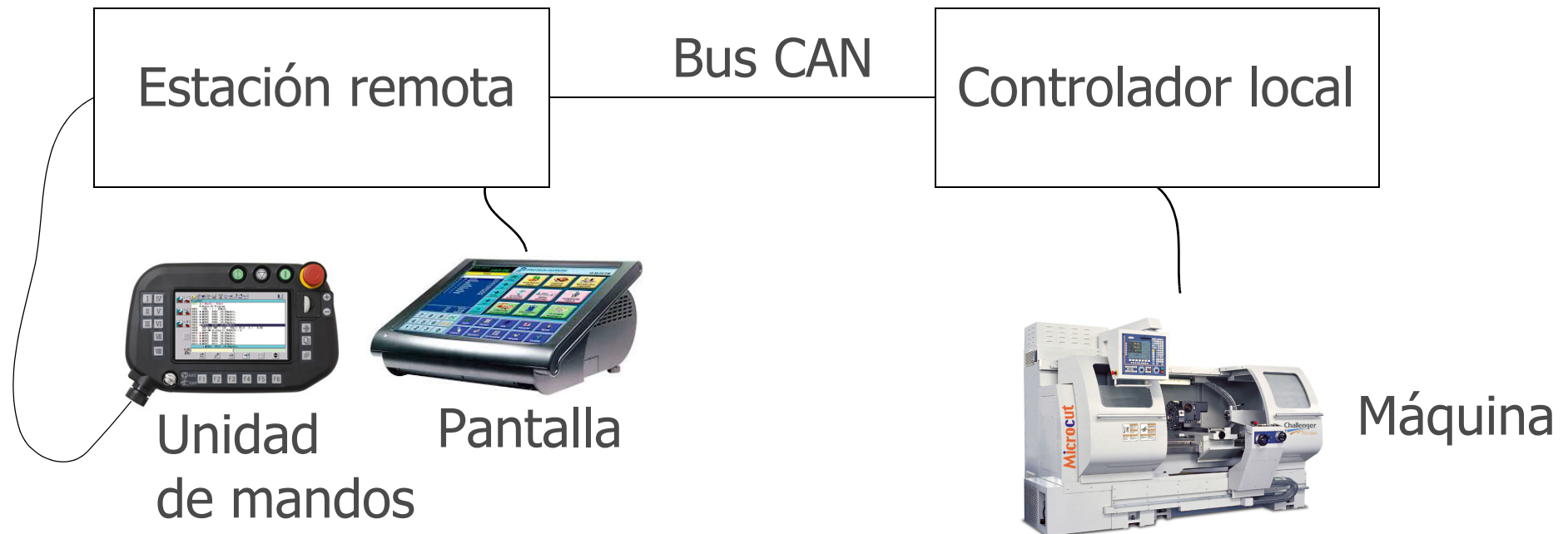
---

## Examen Junio 2016

# Introducción

El objetivo de este ejercicio es desarrollar parte de un sistema de control de una máquina compuesto por dos procesadores unidos por un bus CAN de 1Mb/s

## Plataforma



# Software funcional

---

La funcionalidad del software está ya desarrollada en forma de 4 funciones

- dos de ellas para funcionar en la estación remota
  - lectura\_unidad\_de\_mandos()
  - escritura\_en\_pantalla()
- otras dos para el controlador local
  - actuacion()
  - monitorizacion()

# Requisitos funcionales

---

1. La estación remota lee periódicamente la unidad de mandos invocando a `lectura_unidad_de_mandos()` y envía un mensaje de actuación al controlador local
  - el mensaje ocupa 32 bytes, y por tanto requiere 4 frames del bus CAN
2. Cada vez que la estación remota recibe un mensaje de monitorización muestra el estado de la máquina en la pantalla, con `escritura_en_pantalla()`
  - el mensaje de monitorización ocupa 64 bytes, y por tanto requiere 8 frames del bus CAN
3. Cada vez que el controlador local recibe un mensaje de actuación realiza la actuación sobre la máquina invocando la función `actuacion()`
4. El controlador local obtiene periódicamente el estado de la máquina invocando a `monitorizacion()` y envía un mensaje de monitorización por el bus CAN

# Requisitos temporales (no funcionales)

---

1. Los tiempos de ejecución de peor caso medidos para las funciones software ya disponibles son los siguientes:
  - `lectura_unidad_de_mandos()`: 5.5 ms
  - `escritura_en_pantalla()`: 47.0 ms
  - `actuacion()`: 5.5 ms
  - `monitorizacion()`: 14.0 ms
2. El muestreo de la unidad de mandos se hará en la estación remota cada 20ms. La actividad debe estar terminada en un plazo de 20ms
3. La monitorización de la máquina se hará en el controlador local cada 100ms. La actividad debe estar terminada en un plazo de 100ms

Estos requisitos deben validarse con un análisis de planificabilidad

# Otros requisitos no funcionales

---

Los desarrollos y el software básico estarán basados en la plataforma indicada arriba, con dos procesadores y sendos sistemas operativos gobernados por eventos, con prioridades fijas

- el rango de prioridades es 1..255 en ambos procesadores
- el tiempo de cambio de contexto es de:
  - estación remota: 2  $\mu$ s
  - controlador local: 10  $\mu$ s

# Ejercicios

---

1. Dentro del proceso de análisis de requisitos, generar diagramas UCM para los requisitos del sistema
  - exceptuando los de la plataforma
2. Modelar con AADL una arquitectura para este sistema, con
  - dos threads periódicos
  - dos flujos de mensajes
  - dos threads activados por la llegada de un mensaje
3. Modelar con MAST la arquitectura del sistema para realizar un análisis de planificabilidad inicial
  - ¿Cuál es la asignación óptima de prioridades?
  - ¿Es planificable el sistema con esa asignación?
  - ¿Cuánto podríamos aumentar los tiempos de ejecución y seguir manteniendo el sistema planificable?
    - en la estación remota
    - en el controlador local

# Entregar

---

Un informe en pdf con:

- diagrama(s) UCM
- diagrama(s) AADL de nivel de sistema
- una captura de pantalla de los resultados de MAST y las respuestas a las preguntas planteadas

Workspace de UCMNav comprimido

Workspace de OSATE comprimido

Ficheros del modelo MAST en un archivo comprimido