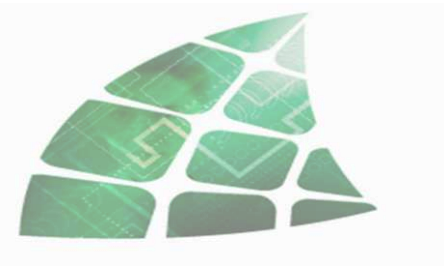


Tema 10. Programación en Tiempo Real

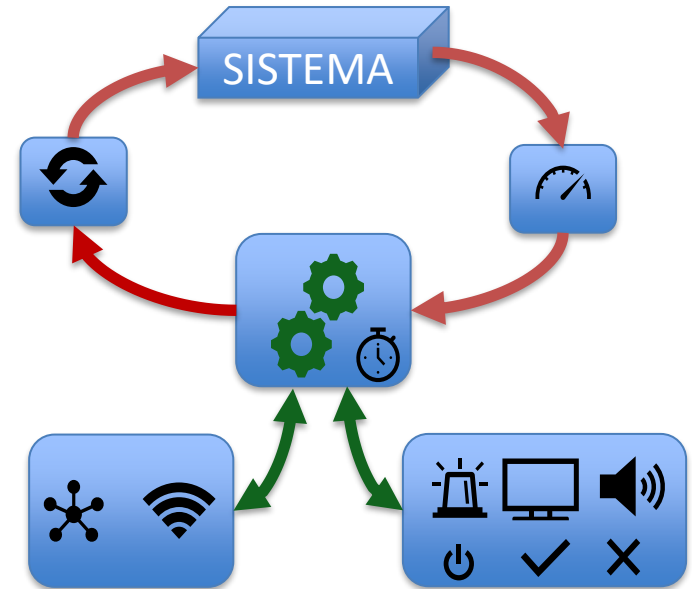
4º Grado de Ingeniería en Electrónica, Robótica y Mecatrónica
Andalucía Tech

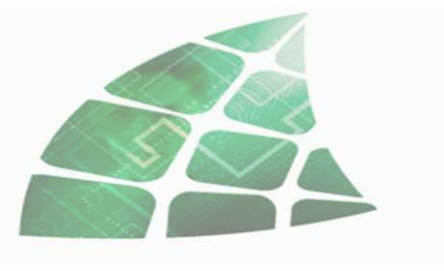
- Introducción
- Sistemas en Tiempo Real
- TI-RTOS



Introducción

- Esquema general:
- Bucle de control
 - Medir-calcular-actuar
 - Ni muy lento ni muy rápido
- RTOS
- Programación en Tiempo Real





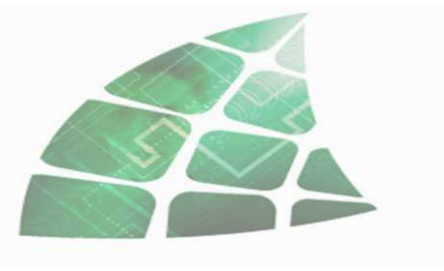
Sistemas en Tiempo Real

- Adecuar la actuación y la medida al sistema
- Si muy lento, pierdo información
- Si muy rápido, mayor consumo y sobreactuó
- Dependerá de los tiempos característicos
 - Requiere conocer bien el sistema
 - A veces, estimar o suponer la dinámica



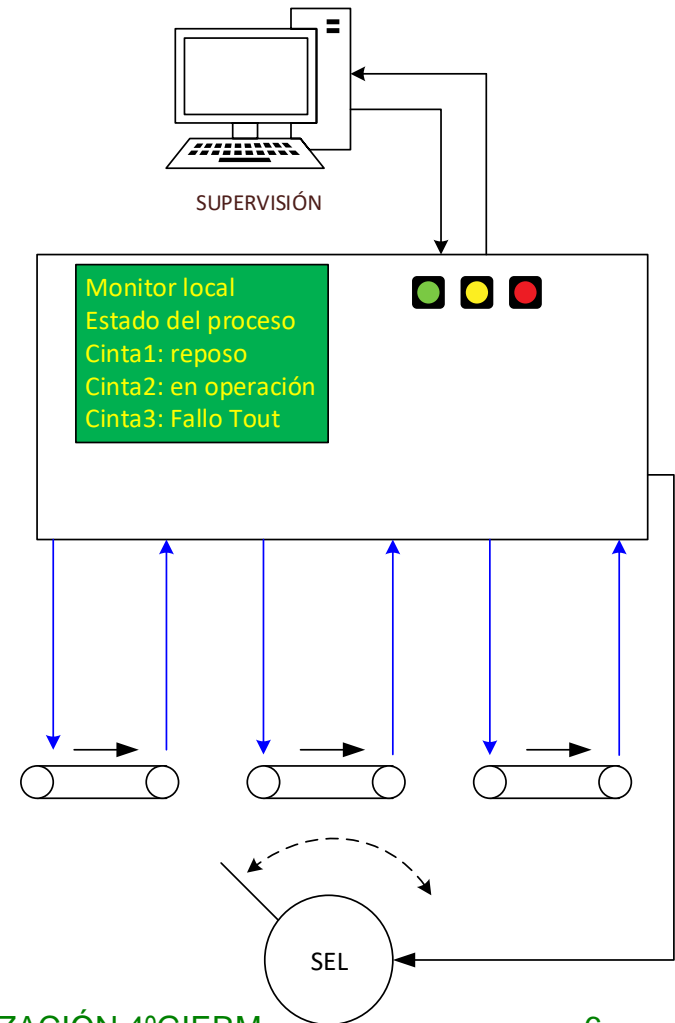
Sistemas operativos en RT

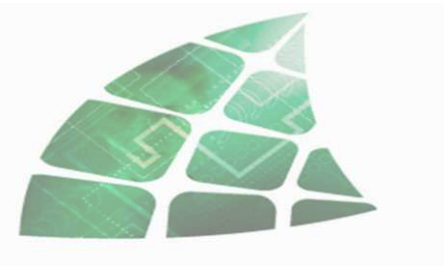
- Necesidad de coordinar múltiples tareas
 - Paralelismo o multitarea
- Temporizar eventos
- Gestionar recursos comunes
 - Evitar colisiones
- Capa extra de software que gestiona los periféricos y el uso de la CPU



Ejemplo

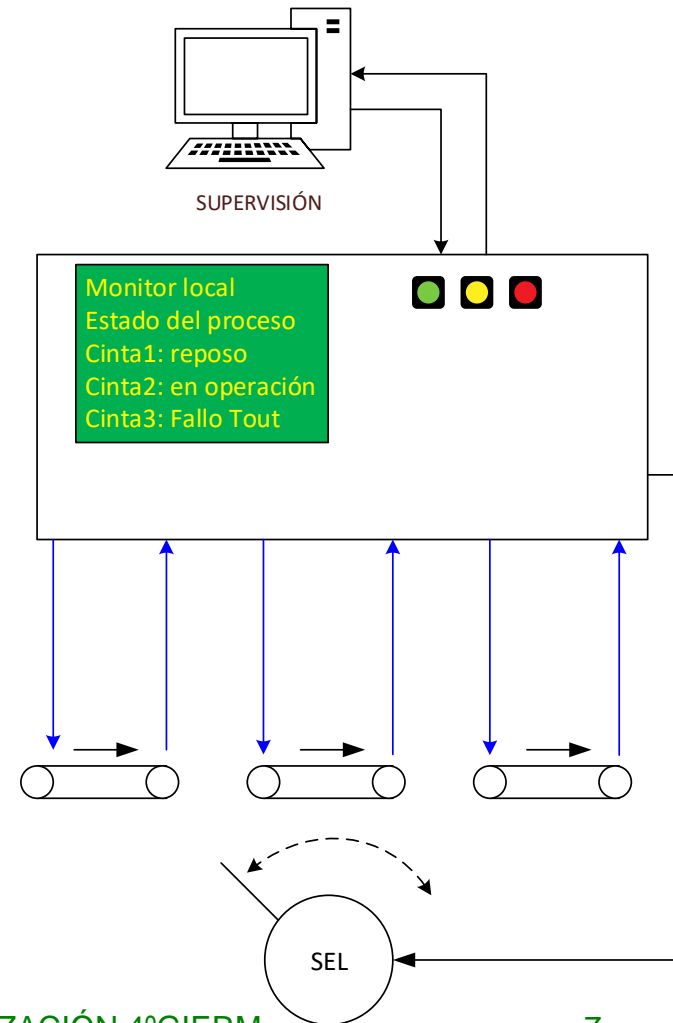
- Control de una instalación
 - 3 cintas transportadoras
 - Mando único
 - Funcionamiento en paralelo
 - Supervisión remota
 - 3 tareas simultáneas y otras en *background*
 - Gestión coordinada

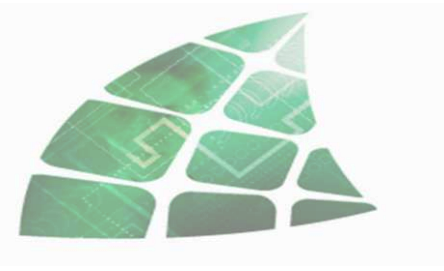




Ejemplo

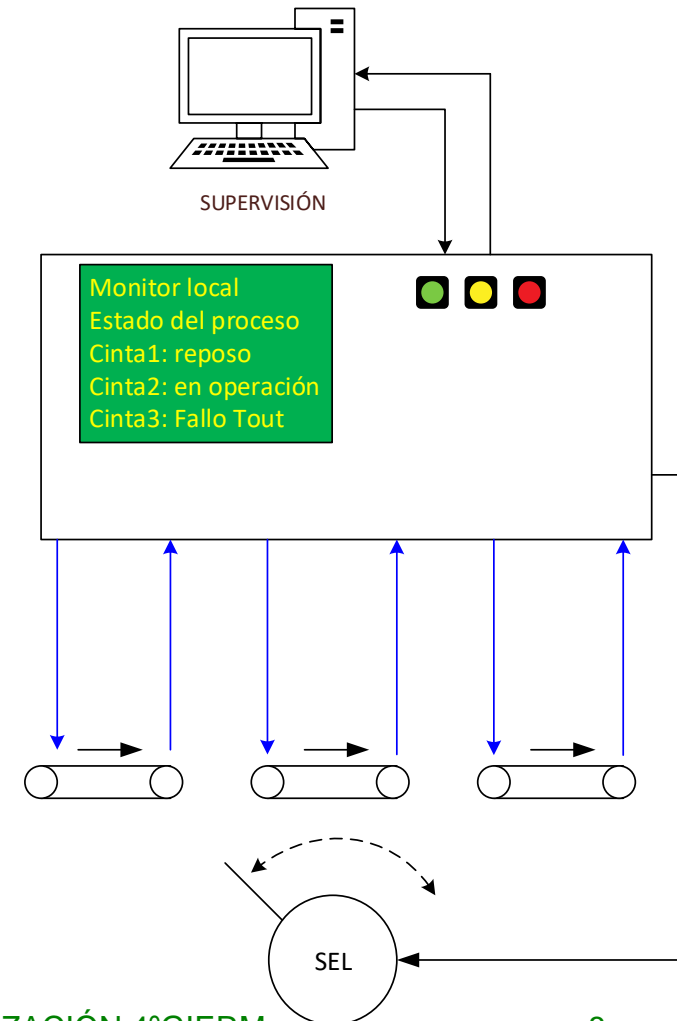
- Coordinar las 3 cintas
 - Mover el selector para recoger piezas
 - Notificar fallos en la cinta
 - Señalar el estado
 - Detectar entrada y salida de piezas
 - Informar a supervisión. Recibir comandos

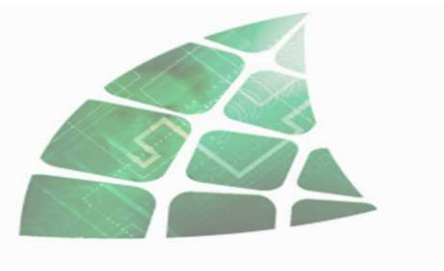




Ejemplo

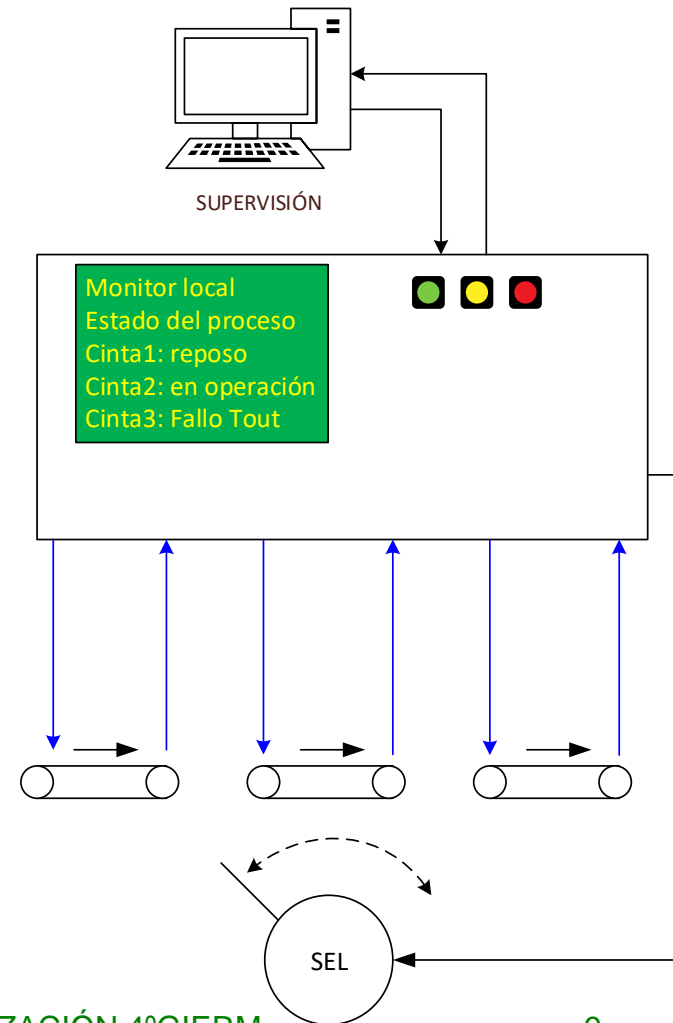
- Tarea 'Cinta i'
 - Leer sensores
 - Computar estado
 - Parado, moviendo, final
 - Comprobar estado selector
 - Variar velocidad cinta
 - Informar a tarea central
 - Si error, parar y esperar
 - Mover selector

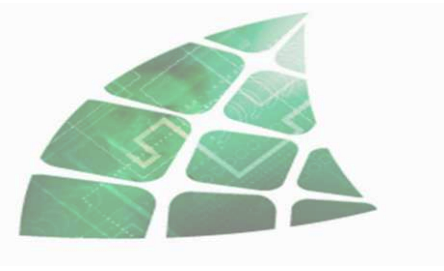




Ejemplo

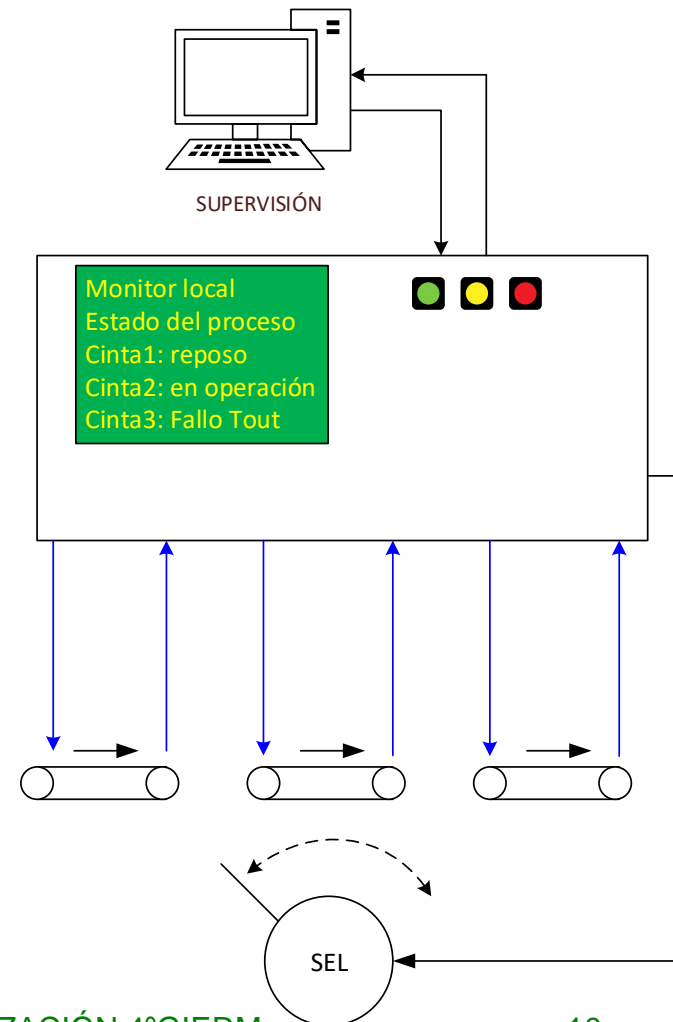
- Tarea central
 - Comprobar estado cintas
 - Señalizar localmente
 - Mensajes a monitor
 - Leds
 - Informar supervisión
 - Gestionar alarmas

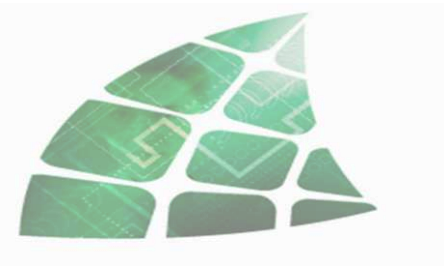




Ejemplo

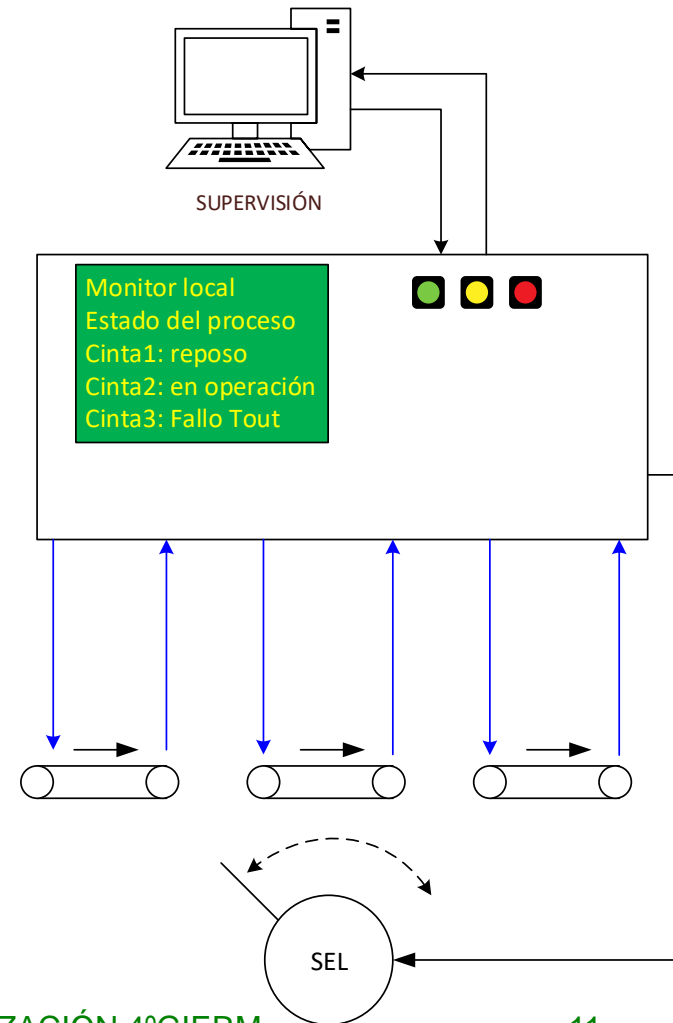
- Tiempos característicos:
 - Selector en uso: 5s
 - Cintas moviendo: 10s
 - Manejo monitor local: 40ms
 - Mensajes a supervisión: 20ms
 - Muestreo cada 50ms

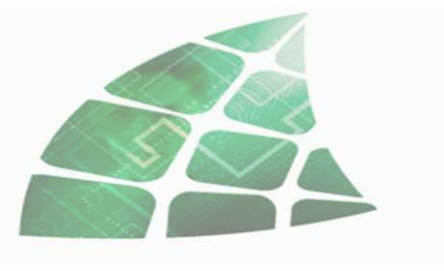




Ejemplo

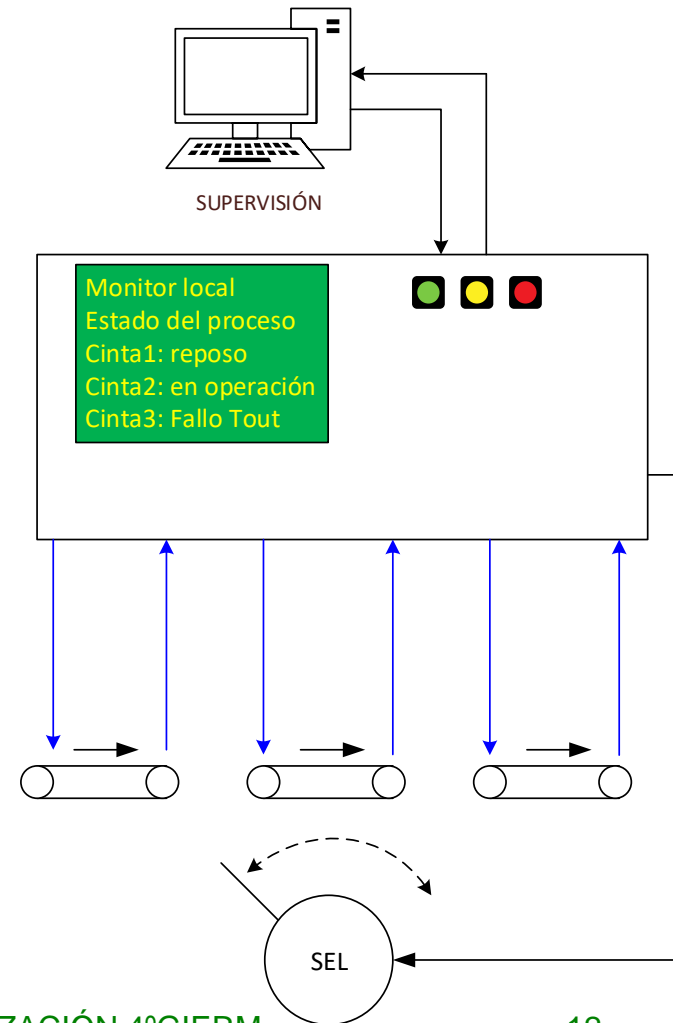
- Sin RTOS:
 - Programar 3 timers para las 3 cintas
 - Variables globales para comprobar
 - Rutina de background para la tarea central
 - Cuidado con los recursos compartidos (selector)
 - Prioridad de las interrupciones





Ejemplo

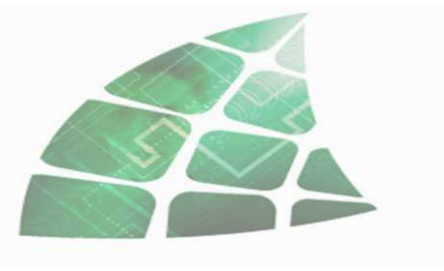
- Con RTOS:
 - Definir 3 tareas en *paralelo*
 - Establecer sincronización entre ellas
 - Comunicación con la tarea de background
 - Reparto del tiempo de proceso y recursos





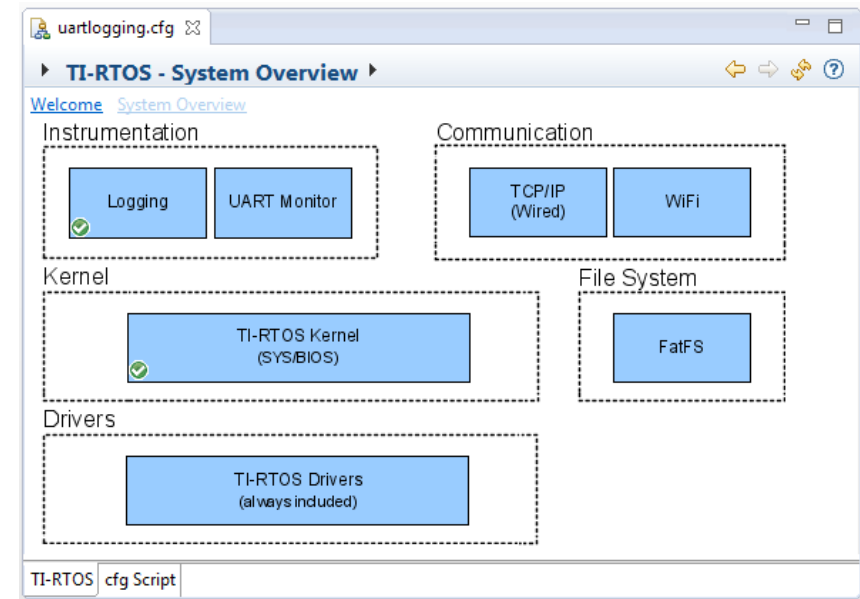
Ventajas e inconvenientes

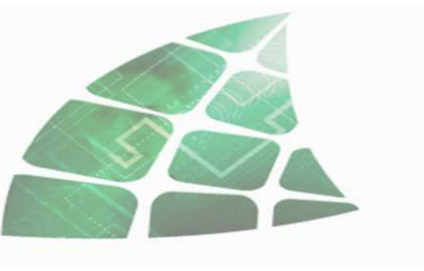
- Ventajas RTOS:
 - Mejor gestión de tareas y recursos
 - Escalabilidad y soporte de la complejidad
- Inconvenientes de los RTOS:
 - Consumo de memoria y recursos (timers...)
 - Capa extra de software



TI-RTOS

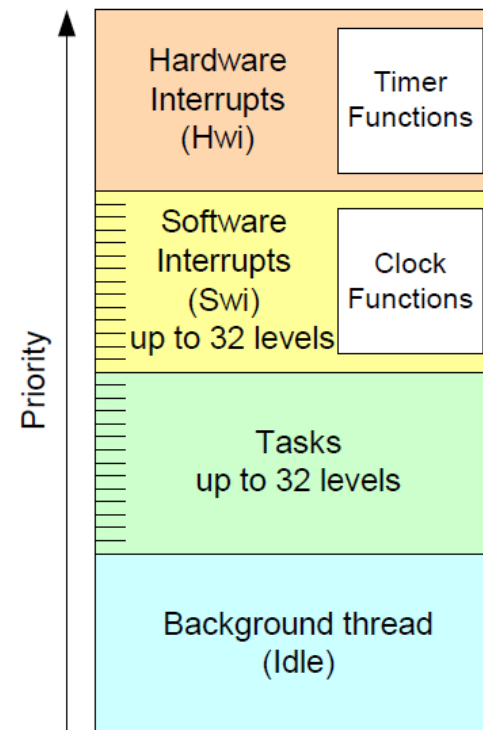
- Sistema completo y escalable para TIVA
 - Núcleo multitarea (SYS/BIOS)
 - Drivers para la placa
 - Analizadores del sistema
 - Middleware:
 - Network
 - USB
 - FatFS

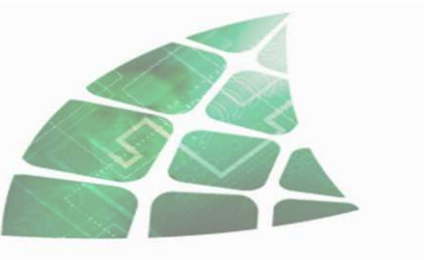




SYS/BIOS

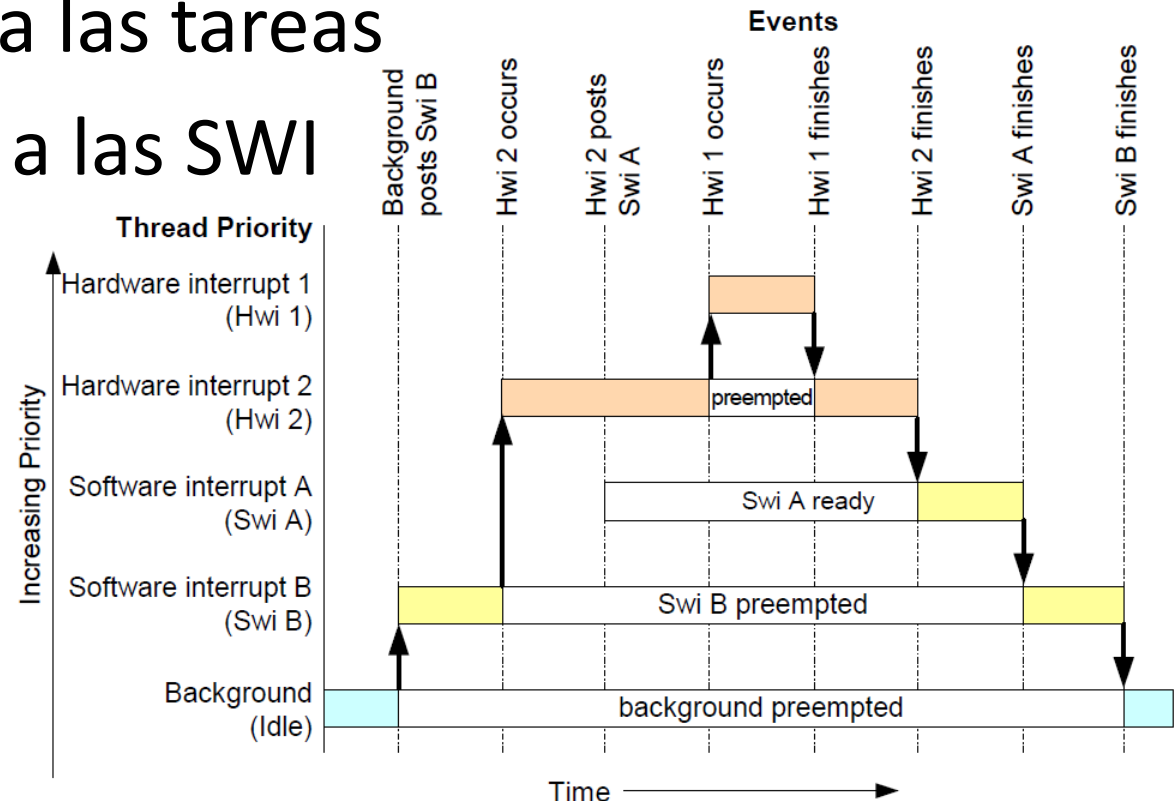
- *CORE* del RTOS. Gestión de tareas y sincronización
- Tipos de tareas:
 - Tarea IDLE
 - Tareas simultáneas
 - Interrupciones SW
 - Interrupciones HW
- Gestión de prioridades

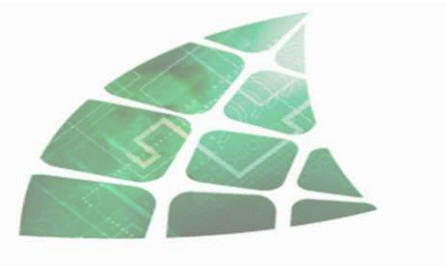




SYS/BIOS

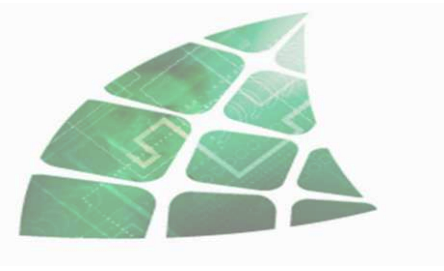
- Prioridades por tipo y dentro del tipo
- Las SWI paran a las tareas
- Las HWI paran a las SWI





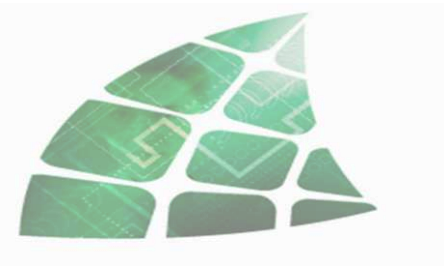
Métodos de sincronización

- Varios métodos para sincronizar tareas:
 - Semáforos
 - Eventos
 - Mailbox
 - Colas
 - Puertas
- ~ ✨ Synchronization
 - ✨ Semaphore
 - ✨ Event
 - 📧 Mailbox
 - 🔄 Queue
 - > 📦 Gates
 - > ✨ Syncs



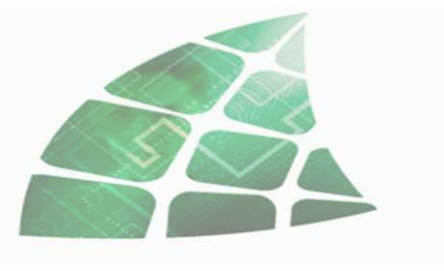
Métodos de sincronización

- Semáforos
 - Para 2 tareas (binarios) o más.
 - Una tarea “señala” y la otra “espera” a que la primera señale.
- Eventos
 - Como semáforos, pero con condiciones múltiples
 - And y Or de varias condiciones



Métodos de sincronización

- Mailbox
 - Mandar *mensajes* entre tareas. Se suele usar en conjunción con eventos o semáforos
- Colas
 - Colas FIFO.
 - Se depositan elementos o se leen elementos de la cola.



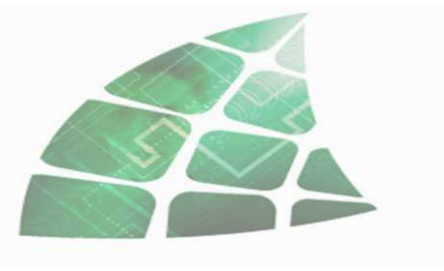
Métodos de sincronización

- Puertas (gates)
 - Previene acceso concurrente en regiones críticas
 - Diversos tipos, según qué deshabilitan:
 - GateHwi: deshabilita incluso interrupciones hw
 - GateSwi: deshabilita hasta interrupciones sw
 - GateTask: solo deshabilita el resto de tareas
 - Se debe minimizar el tiempo de uso



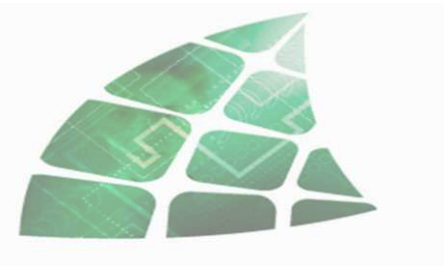
Tareas

- Se definen tareas con propiedades específicas
 - Paso de parámetros
 - Tamaño de la pila
 - Función que se ejecuta
- Tarea 'Idle' en caso de no ejecutar ninguna
- Se accede a todas las tareas de manera concurrente
 - Cada cierto tiempo se cambia de una a otra



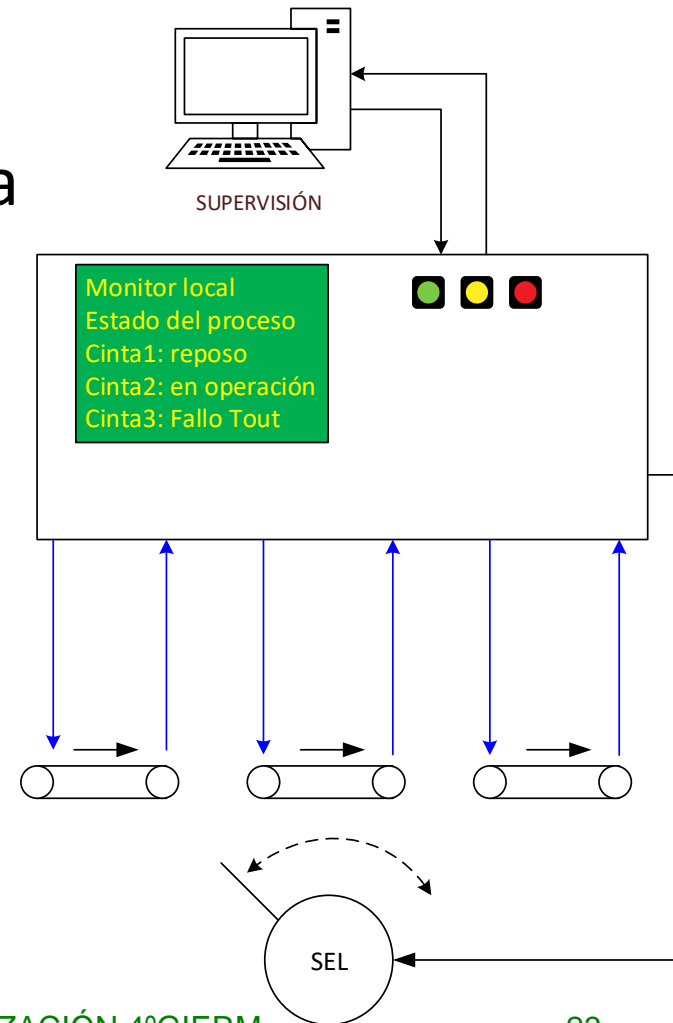
Procesos temporizados

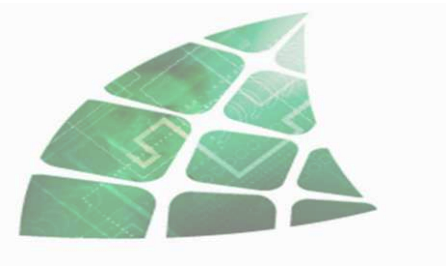
- Usar Tareas continuas con un sleep
 - Las tareas se ejecutan *en paralelo*, cada una con un bucle infinito y un sleep de tiempo fijo
- Usar tareas con Clock (swi)
 - Definir funciones que se ejecutan cada **N** ticks de sistema.
- Usar interrupciones con Timers (hwi)
 - Usar timers del micro para generar interrupciones hw.
- Ordenadas de menor a mayor prioridad y exactitud



Ejemplo

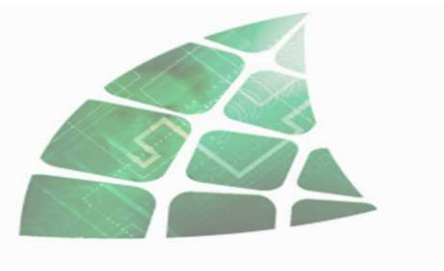
- Con TI-RTOS:
 - Tareas de manejo de cada cinta
 - Gestión de SEL:
 - GATE y variable global
 - Cola FIFO
 - Mailbox
 - Tarea central:
 - Manejo de terminal local y remoto





Programación en TI-RTOS

- Instalar el módulo TI-RTOS
 - Se instala a la vez el XDCTOOLS 3.32.0.06
- Crear un proyecto con TIRRTOS
- Editar el fichero *.cfg con XGCONF
- Añadir los drivers/librerías
- Ver los ejemplos:
 - C:\ti\tirtos_tivac_2_16_01_14\tirtos_tivac_2_16_01_14_examples\TI\EK_TM4C1294XL



Por desarrollar

- Adaptar drivers para poder usar pantalla y boosterpack
- Rehacer ejemplos para manejo de periféricos y características de Tiempo Real