



MQX RTOS

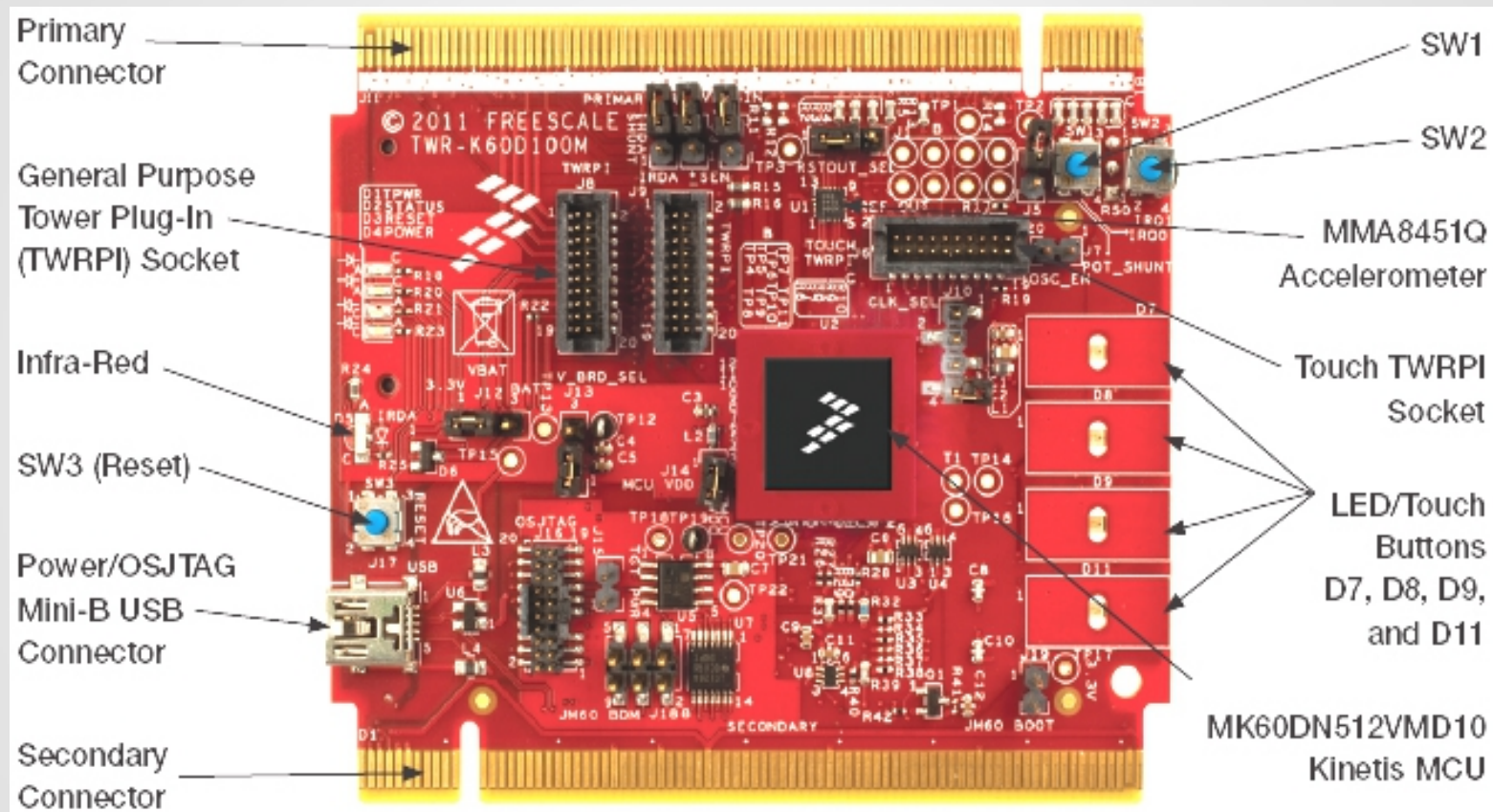
Módulo TWR-K60D100M

Freescall



Módulo TWR-K60D100M

TWR-K60D100M





Módulo TWR-K60D100M

Módulo TWR-K60D100M

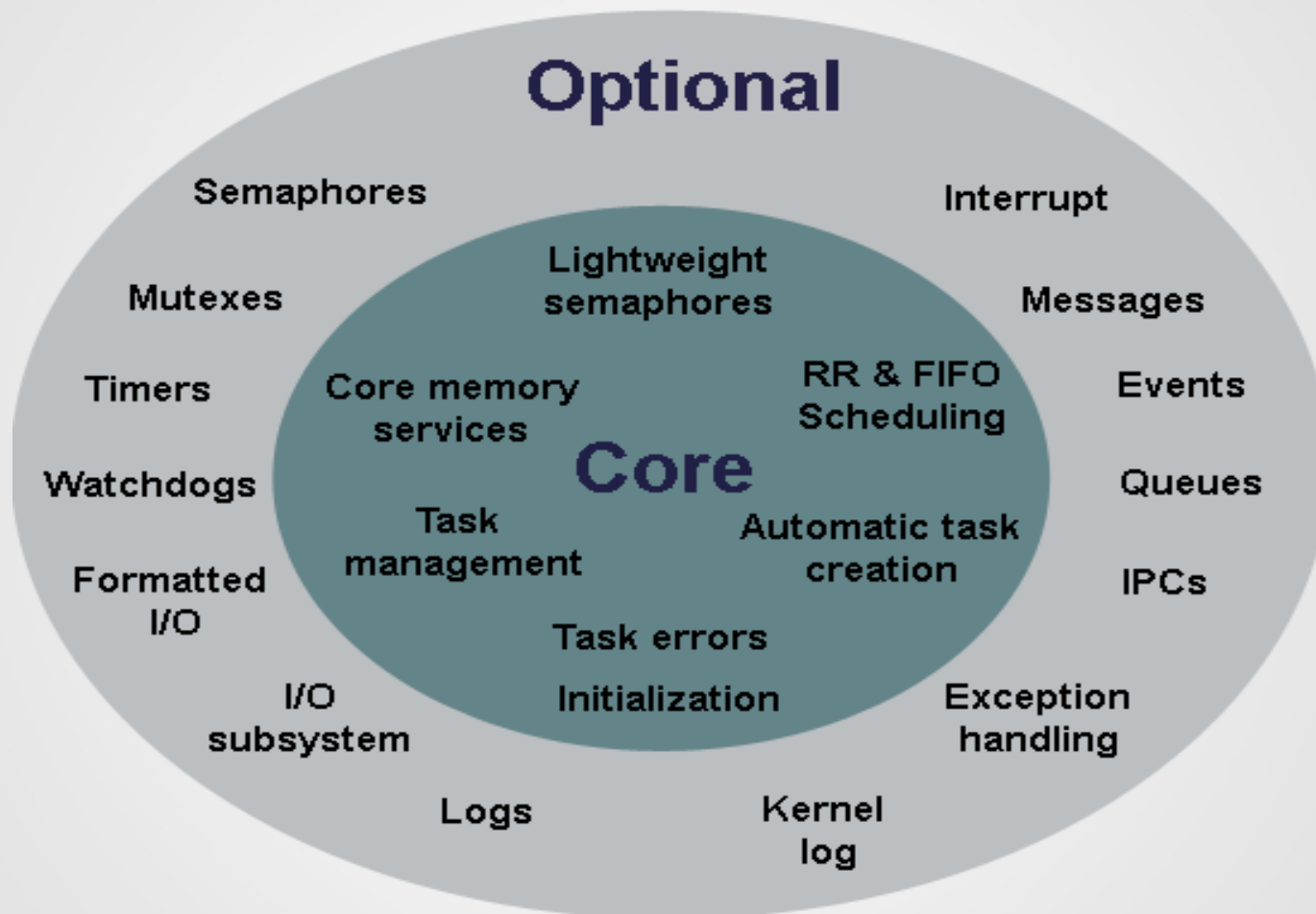
La política de planificación soportada en la placa TWR-K60D100M es FIFO, es decir la tarea de mayor es la que primero se ejecuta. La tarea activa se ejecuta hasta:

- La tarea activa abandona voluntariamente el uso del procesador, ya que llama a una función que bloquea la tarea. Por ejemplo `_lwsem_wait()`.
- Se presenta una interrupción que tiene mayor prioridad que la tarea activa.
- Una tarea que tiene mayor prioridad que la tarea activa, está lista.



Módulo TWR-K60D100M

Sincronización de tareas





Módulo TWR-K60D100M

Sincronización de tareas

Se puede sincronizar tareas mediante el uso de los siguientes mecanismos:

- Eventos
- Eventos livianos
- Mutex
- Cola de mensajes liviana
- Cola de tareas
- Semáforos
- Semáforos livianos



Módulo TWR-K60D100M

Eventos:

Son un componente opcional. Las tareas y rutinas de servicio de interrupción pueden utilizar los eventos para sincronizar y transmitir información sencilla en forma de cambios de estado de bits.

Eventos livianos:

Son un componente opcional. Son una forma bajos costos para sincronizar tareas utilizando los cambios de estado de bits. Requieren una cantidad mínima de memoria.



Módulo TWR-K60D100M

Mutex:

Son un componente opcional. Un mutex ofrece la exclusión mutua entre tareas, cuando tienen acceso a un recurso compartido.

Cola de mensajes livianas:

Son un componente opcional. Las tareas envían y reciben mensajes utilizando colas de mensajes livianas. Los mensajes tienen un tamaño fijo, un múltiplo de 32 bits. Las funciones de lectura y escritura son bloqueantes.

Colas de tareas:

Las colas de tareas proporcionan una forma simple y eficiente para sincronizar las tareas. Las tareas se pueden suspender y eliminar de la cola de tareas.



Módulo TWR-K60D100M

Semáforos:

Son un componente opcional. Se puede usar semáforos para sincronizar las tareas o para proteger el acceso a un recurso compartido, o para implementar un mecanismo de productor - consumidor.

Semáforos livianos (lwsem):

Son un componente central. Los semáforos lwsem requieren una cantidad mínima de memoria y se ejecutan rápidamente.



Módulo TWR-K60D100M

Semáforos livianos (Lightweight Semaphores)

Son semáforos sin nombre. Se crean a partir de estructuras de datos estática. Para crear un semáforo de liviano utilizamos la función `_lwsem_create()`

`_mqx_uint _lwsem_create(LWSEM_STRUCT_PTR lwsem_ptr, _mqx_int initial_count)`

`lwsem_ptr`: Puntero al semáforo a crear

`initial_count [IN]`: Valor inicial del semáforo

Devuelve `MQX_OK` en caso de éxito. En caso de error devuelve `MQX_EINVAL` (`lwsem` ya está inicializado) o `MQX_INVALID_LWSEM` se intenta acceder a un `lwsem` para el cual no tiene permisos.

```
LWSEM_STRUCT lwsem_free;  
_mqx_uint result = _lwsem_create(&lwsem_free, valor_inicial);
```



Módulo TWR-K60D100M

Esperando a un semáforo

Una tarea espera a un semáforo con la función `_lwsem_wait()`. Si el valor del semáforo es mayor que cero, MQX RTOS resta 1 al semáforo, y la tarea continúa funcionando. Si el valor del semáforo es cero, MQX RTOS bloquea la tarea, hasta que otra tarea incremente al semáforo.

`_mqx_uint _lwsem_wait(LWSEM_STRUCT_PTR lwsem_ptr)`

`lwsem_ptr`: Puntero del semáforo

Devuelve `MQX_OK` en caso de éxito, o `MQX_INVALID_LWSEM` en caso de error (`lwsem` no es un puntero válido de semáforo)

```
result=_lwsem_wait(&lwsem_sem1);  
if (result != MQX_OK) {  
    printf("\n_lwsem_wait falló");  
    _task_block(); }
```



Módulo TWR-K60D100M

Incrementando un semáforo

Se incremente el semáforo con la función `_lwsem_post()`.

`_mqx_uint _lwsem_post(LWSEM_STRUCT_PTR lwsem_ptr)`

`lwsem_ptr`: Puntero del semáforo

Devuelve `MQX_OK` en caso de éxito, o `MQX_INVALID_LWSEM` en caso de error (`lwsem` no es un puntero válido de semáforo)

```
result= _lwsem_post(&lwsem_sem1);  
if (result != MQX_OK) {  
    printf("\n_lwsem_post falló");  
    _task_block(); }
```



Módulo TWR-K60D100M

Destrucción de un semáforo

Cuando una tarea ya no necesita un semáforo, lo puede destruir con `_lwsem_destroy()`.

```
_mqx_uint _lwsem_destroy( LWSEM_STRUCT_PTR lwsem_ptr)
```

lwsem_ptr: Puntero del semáforo

Devuelve `MQX_OK` en caso de éxito, o `MQX_INVALID_LWSEM` en caso de error (lwsem no es un puntero válido de semáforo)



Módulo TWR-K60D100M

Función `_sched_yield`

Esta función pone la tarea activa al final de la cola de listos.
Si no hay otras tareas en esta cola de listos, la tarea sigue siendo la tarea activa.



Bibliografía

Freescale MQX™ RTOS Reference Manual. 2014.

Freescale MQX™ RTOS User's Guide. 2014.

TWR-K60D100M Tower Module User's Manual. 2011.