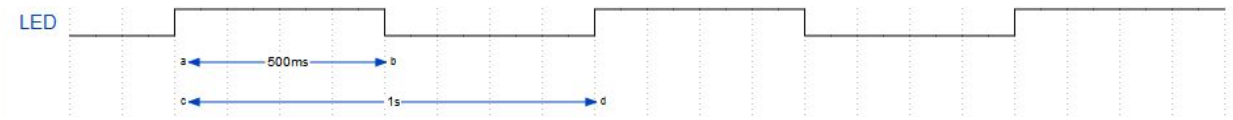
1. Temporización

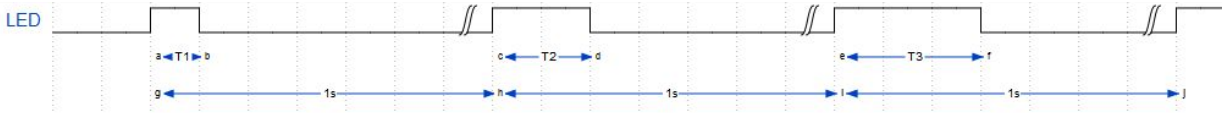
1.1 Demoras fijas

Implementar una tarea que encienda un LED durante 500 ms cada T = 1 seg.



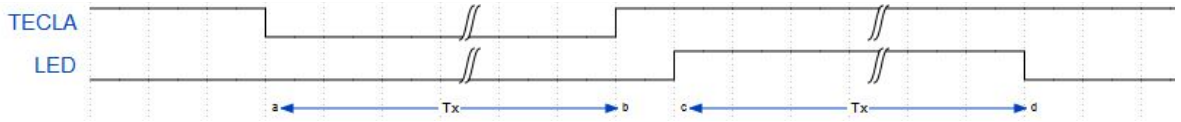
1.2 Periodos fijos

Implementar una tarea que genere una onda cuadrada (y que encienda un LED) con periodo de 1 seg y ciclos de actividad incrementándose T1 = 100 ms, T2 = 200 ms, T3 = 300 ms, .. T9 = 900ms



1.3 Medir tiempo transcurrido (ENTREGA OBLIGATORIA)

Medir el tiempo de pulsación de un botón utilizando un algoritmos anti-rebote. Luego destellar un led durante el tiempo medido. Ayuda: Se puede consultar el contador de ticks del RTOS para obtener el tiempo del sistema (en ticks) al inicio y al fin del mismo. En este caso hay que prever que esta variable puede desbordar.

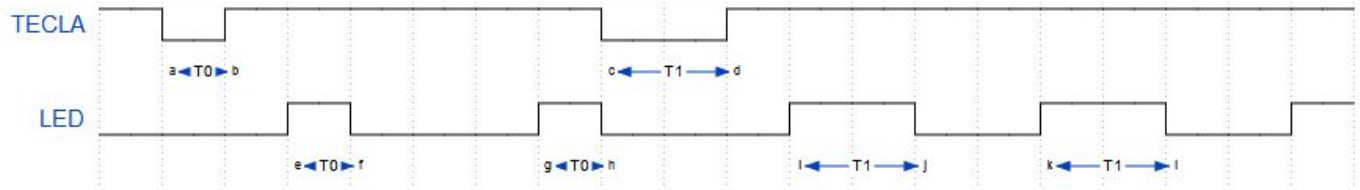


1.4 Ejercicio integrador

Escribir un programa con dos tareas:

- Tarea 1: Medirá el tiempo de pulsación de un botón, aplicando anti-rebote.
- Tarea 2: Destellará un led con un período fijo de 1 seg, y tomando como tiempo de activación el último tiempo medido.

El tiempo medido se puede comunicar entre tareas a través de una variable global, protegiendo sus operaciones dentro de una sección crítica.

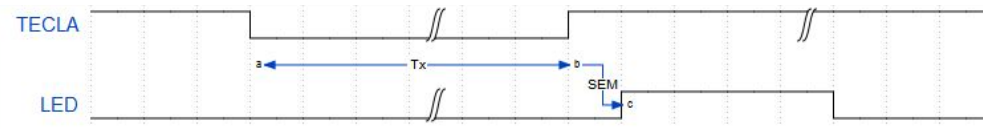


2 Sincronización de tareas mediante semáforos

2.1 Sincronizar dos tareas mediante un semáforo binario

Implementar dos tareas.

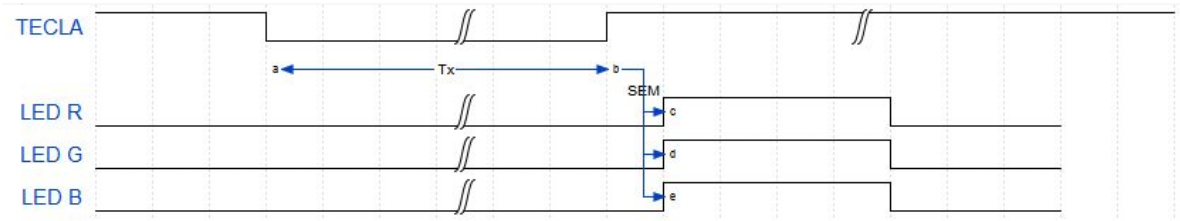
- Tarea 1: Medirá el tiempo de pulsación de un botón, aplicando anti-rebote. Liberará un semáforo al obtener la medición.
- Tarea 2: Esperará por el semáforo y destellará un LED al recibirlo.



2.2 Sincronizar varias tareas

Implementar tres tareas:

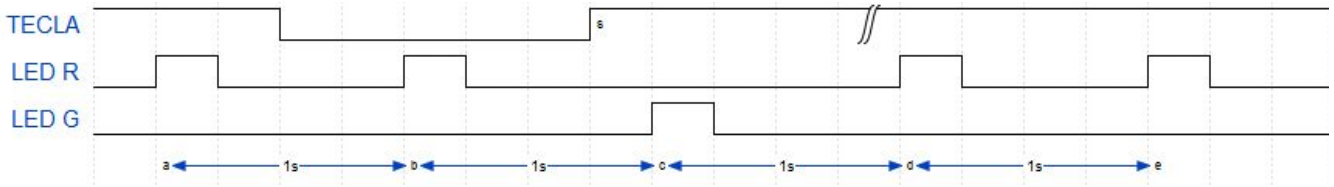
- Tarea 1: Medirá el tiempo de pulsación de un botón, aplicando anti-rebote. Liberará un semáforo al obtener la medición.
- Tarea 2,3,4: Tareas idénticas que destellarán un led (diferente) al recibir el semáforo.



2.3 Tiempo de bloqueo (ENTREGA OBLIGATORIA)

Implementar dos tareas:

- Tarea 1: Medirá el tiempo de pulsación de un botón, aplicando anti-rebote. Liberará un semáforo al obtener la medición.
- Tarea 2: Esperará el semáforo cada un segundo. Si recibe el semáforo se destellará el LED verde y si no recibe el semáforo destellará el LED rojo.

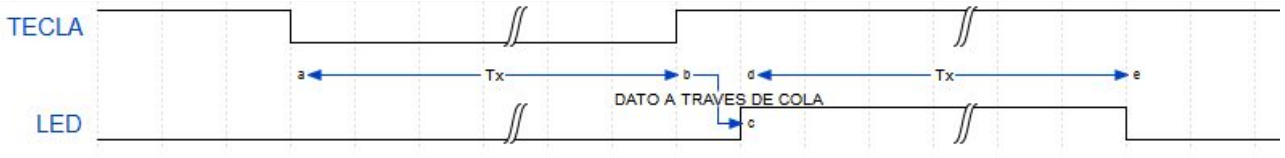


3 Comunicación de datos entre tareas

3.1 Pasaje de datos por copia

Implementar dos tareas.

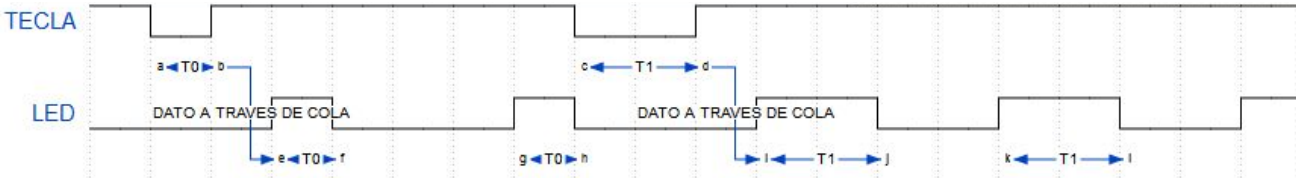
- Tarea 1: Medirá el tiempo de pulsación de un botón, aplicando anti-rebote. Envía el tiempo por la cola.
- Tarea 2: Esperará un dato de la cola y detellará un LED al recibirlo, durante el tiempo indicado.



3.2 Lectura no bloqueante de una cola (ENTREGA OBLIGATORIA)

Escribir un programa con dos tareas:

- Tarea 1: Medirá el tiempo de pulsación de un botón, aplicando anti-rebote y enviará el tiempo a una cola.
- Tarea 2: Destellará un led con un período fijo de 1 seg. El tiempo de activación lo recibirá a través de una cola. La tarea no debe bloquearse, ya que mientras no reciba mensajes debe mantener el led titilando.



3.3 Pasaje de datos por referencia (ENTREGA OBLIGATORIA)

Repetir el ejercicio anterior, pero en vez de enviar el tiempo medido enviar una referencia a una estructura cuyos campos sea el índice de LED y el tiempo medido.

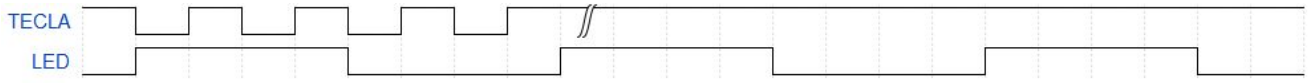
4 Manejo de recursos

Se practicará el uso de mutex para resguardar recursos compartidos entre tareas. Se practicará también el uso de semáforos contadores de eventos (ascendentes).

4.1 Cuenta de eventos con sincronización

Implementar dos tareas.

- Tarea 1: Medirá la pulsación de un botón, aplicando anti-rebote. Liberará un semáforo al liberar cada botón.
- Tarea 2: Destellará un LED 0,5 seg y lo mantendrá apagado 0,5 seg, cada vez que se pulse la tecla. Si durante el periodo se pulsan teclas, no se deberán perder esos eventos (con un límite de 3)

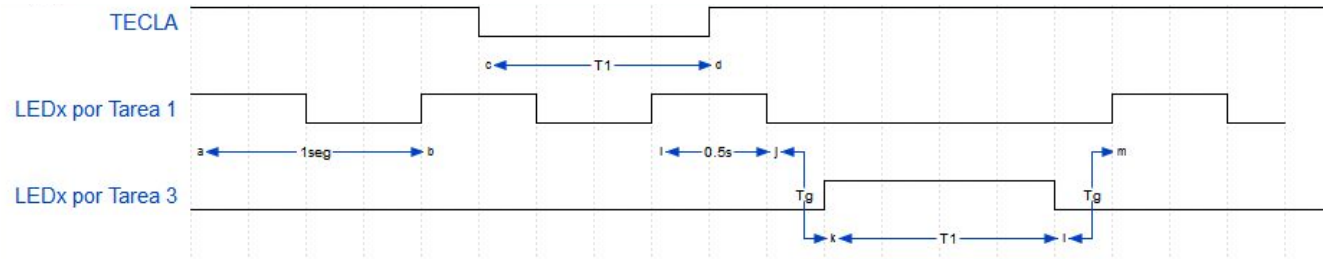


4.2 Exclusión mutua de un recurso compartido (ENTREGA OBLIGATORIA)

Escribir un programa con tres tareas:

- Tarea 1: Medirá el tiempo de pulsación de un botón, aplicando anti-rebote. Enviará el tiempo medido por cola.
- Tarea 2: Periódicamente encenderá un LEDx a 1Hz (50% duty cycle)
- Tarea 3: Obtendrá de la cola el tiempo de pulsación de la tecla medido, y destellará el LEDx durante ese tiempo.

Use un mutex para turnar el acceso al LED y que no se perturbe ninguna de las formas de onda. Lo importante es no interrumpir el tiempo de alto en exhibición, y no "pegarse" al tiempo de alto de la otra tarea (deje un tiempo de off como guardabanda).



5 Sincronización con interrupciones

5.1 Generar/Descubrir un evento en una interrupción.

Implementar el algoritmo de lectura de las teclas mediante IRQ. Utilice recursos del sistema operativo (semáforos) dentro del handler de interrupción.

Repita el ejercicio 1.3.

5.2 Procesar datos en una interrupción (ENTREGA OBLIGATORIA)

Repita el ejercicio anterior, pero utilizando colas dentro del handler de interrupción.