

EJECUTIVO CÍCLICO

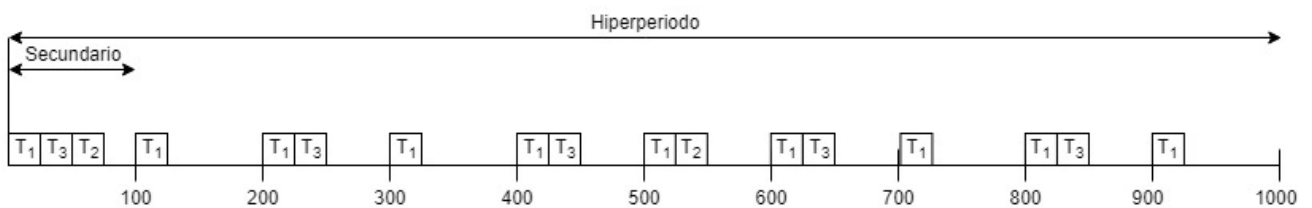
	C	T	
T_1	1	100	* Considerando las tareas dado su rápido tiempo de procesado, medido usando <code>getTickCount()</code> , se asigna un coste de 1 Tick del sistema para cada una.
T_2	1	500	
T_3	1	200	* Para asignar los periodos se han tenido en cuenta los tiempos máximos razonables que aseguran un funcionamiento adecuado del sistema.

* $T_1 \equiv$ Tarea correspondiente a la máquina de estados para los dígitos.

* $T_2 \equiv$ Tarea correspondiente a la máquina de estados de la alarma.

* $T_3 \equiv$ Tarea correspondiente a la máquina de estados del led.

1. Cálculo del hiperperiodo:
M.C.M. (100, 500, 200) \rightarrow 1000
2. Cálculo del periodo de los ciclos secundarios:
M.C.D. (100, 500, 200) \rightarrow 100
3. Colocación de las tareas en el cronograma, comenzando por los de periodo más bajo:



4. Implementación software en C:

```

5.     xLastWakeTime = xTaskGetTickCount ();
6.     frame = 0;
7.
8.     while(true) {
9.         switch (frame){
10.            case 0:
11.                fsm_fire (fsm_digit);
12.                fsm_fire (fsm_led);
13.                fsm_fire (fsm_alarm);
14.                break;
15.            case 100:
16.            case 300:
17.            case 700:
18.            case 900:
19.                fsm_fire (fsm_digit);
20.                break;
21.            case 200:
22.            case 400:

```

```
23.         case 600:
24.             fsm_fire (fsm_digit);
25.             fsm_fire (fsm_led);
26.             break;
27.         case 500:
28.             fsm_fire (fsm_digit);
29.             fsm_fire (fsm_alarm);
30.             break;
31.         case 800:
32.             fsm_fire (fsm_alarm);
33.             fsm_fire (fsm_led);
34.             break;
35.         default:
36.             break;
37.     }
38.     if (DEBUG) printf("Frame: %d\r", frame);
39.     frame = (1 + frame) % (HYPERPERIOD);
40.     vTaskDelayUntil(&xLastWakeTime, PERIOD_TICK/SECONDARY);
41. }
42.
```