Introducción

- RTOS
- FreeRTOS

FreeRTOS Tasks

- Creación de tareas/Tasks
- Comportamiento de las tareas
- Practica

¿Qué es un Sistema Operative de propósito general?

Programa que da soporte a otros programas/aplicaciones que corren en una PC

Administra Prioridades

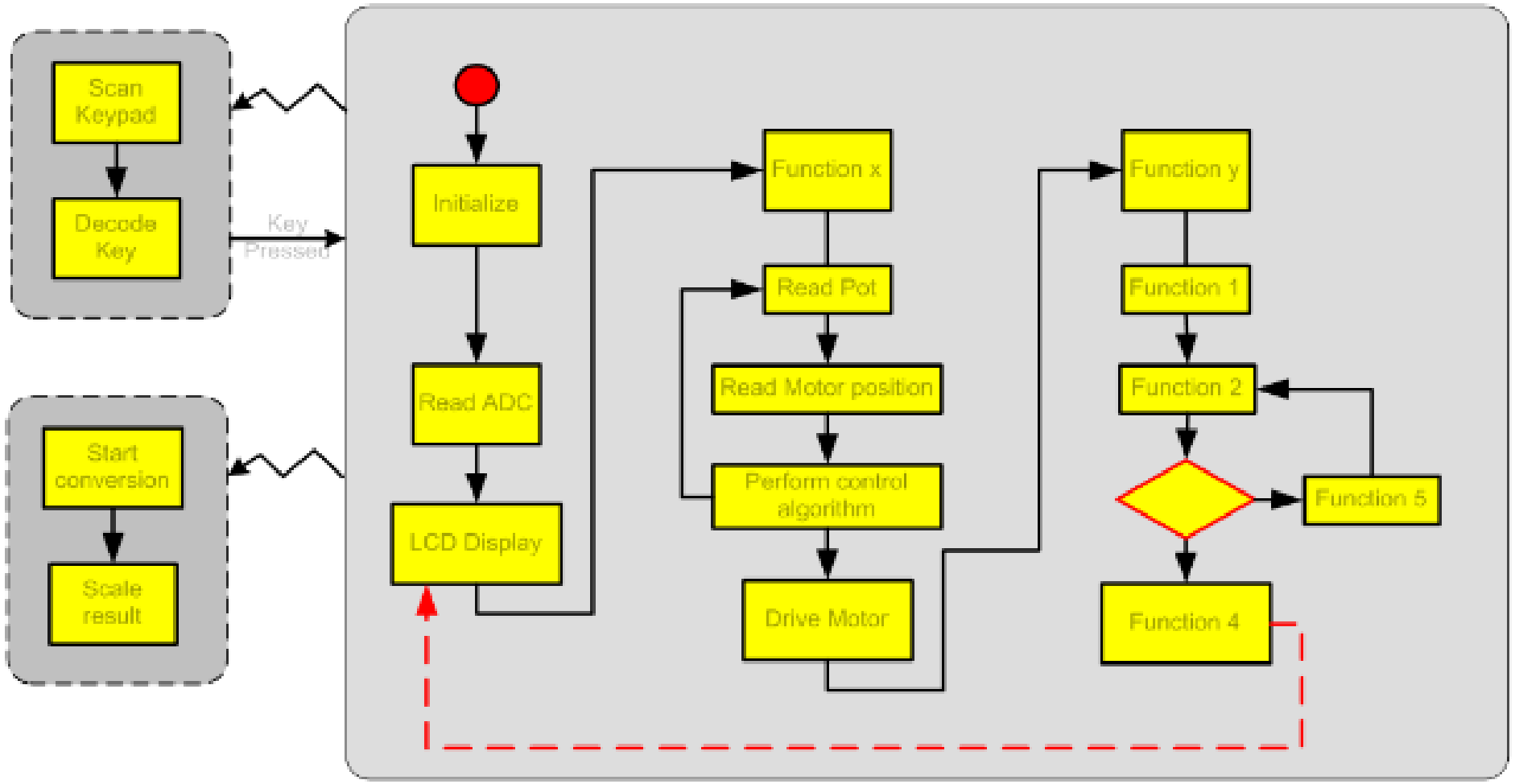
¿Que es RTOS?

El planificador es diseñado para producir un patrón de ejecución predecible.(Determinista)

El Sistema Embebido debe responder a un determinado evento dentro de un tiempo estrictamente definido.

Los Scheduler de RTOS, como el usado en FreeRTOS, logran determinismo permitiendo al usuario asignar prioridades a cada hilo de ejecución

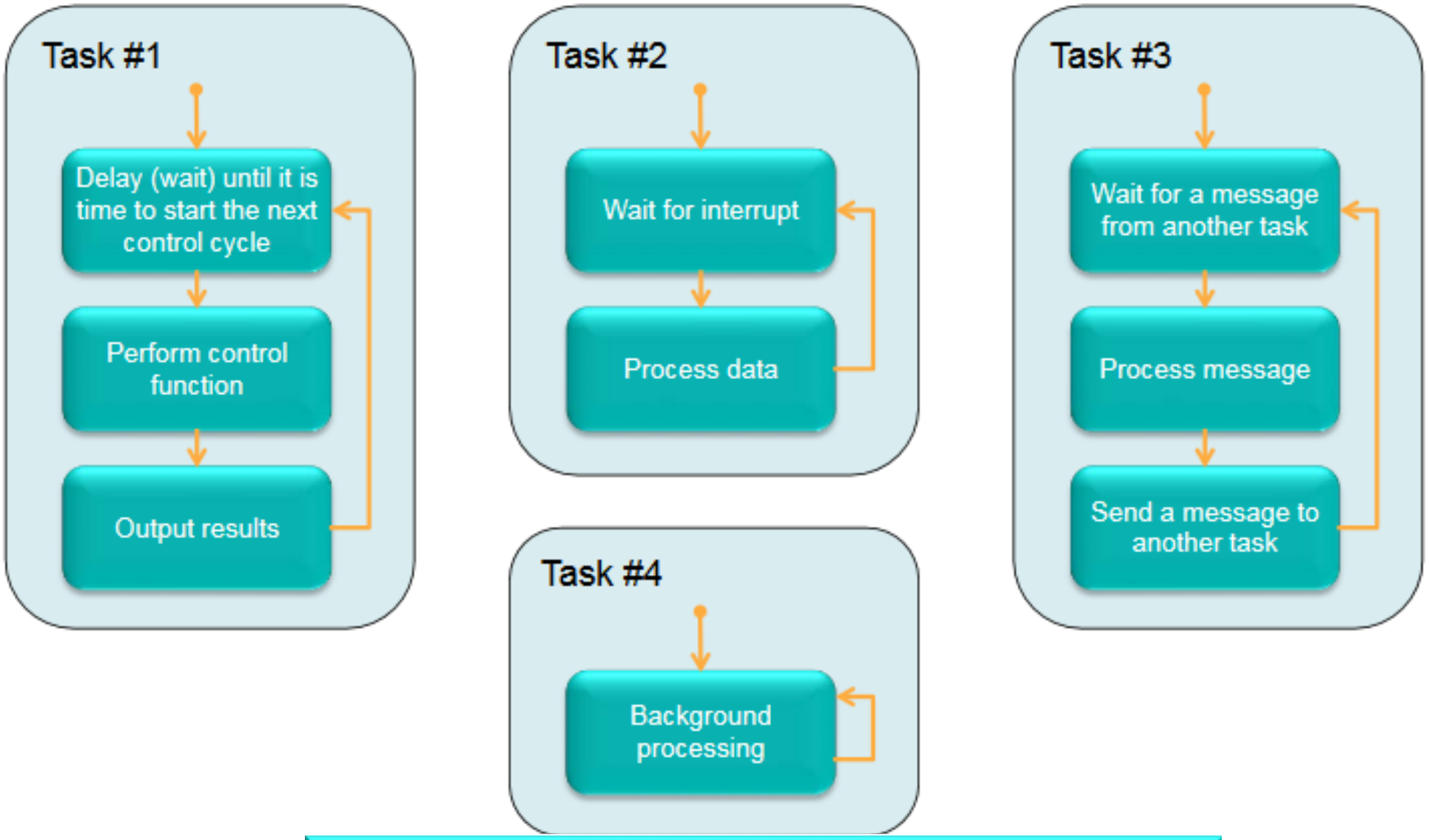
Diseño de Super Loops



Desarrollo Concurrente	☹
Reutilización de código	😐
Testing	😐
Escalabilidad	☹
Tolerancia a cambios de HW	😐

Interdependencia entre el Tiempo y la Funcionalidad

Alternativa RTOS

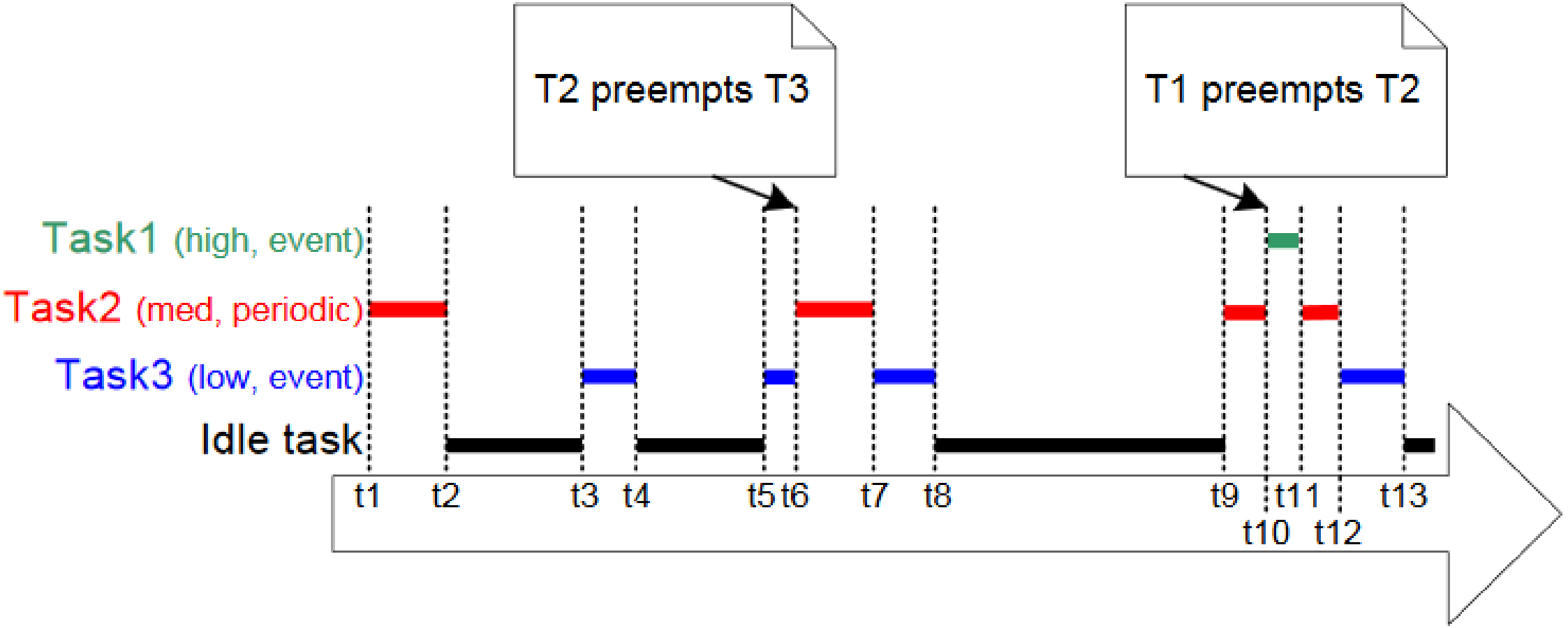


Tareas Autonomas, RTOS se encarga de administrar el tiempo, las señales y el paso de mensajes

Desarrollo Concurrente	☺
Reutilización de código	☺
Testing	😐
Escalabilidad	☺
Tolerancia a cambios de HW	☺

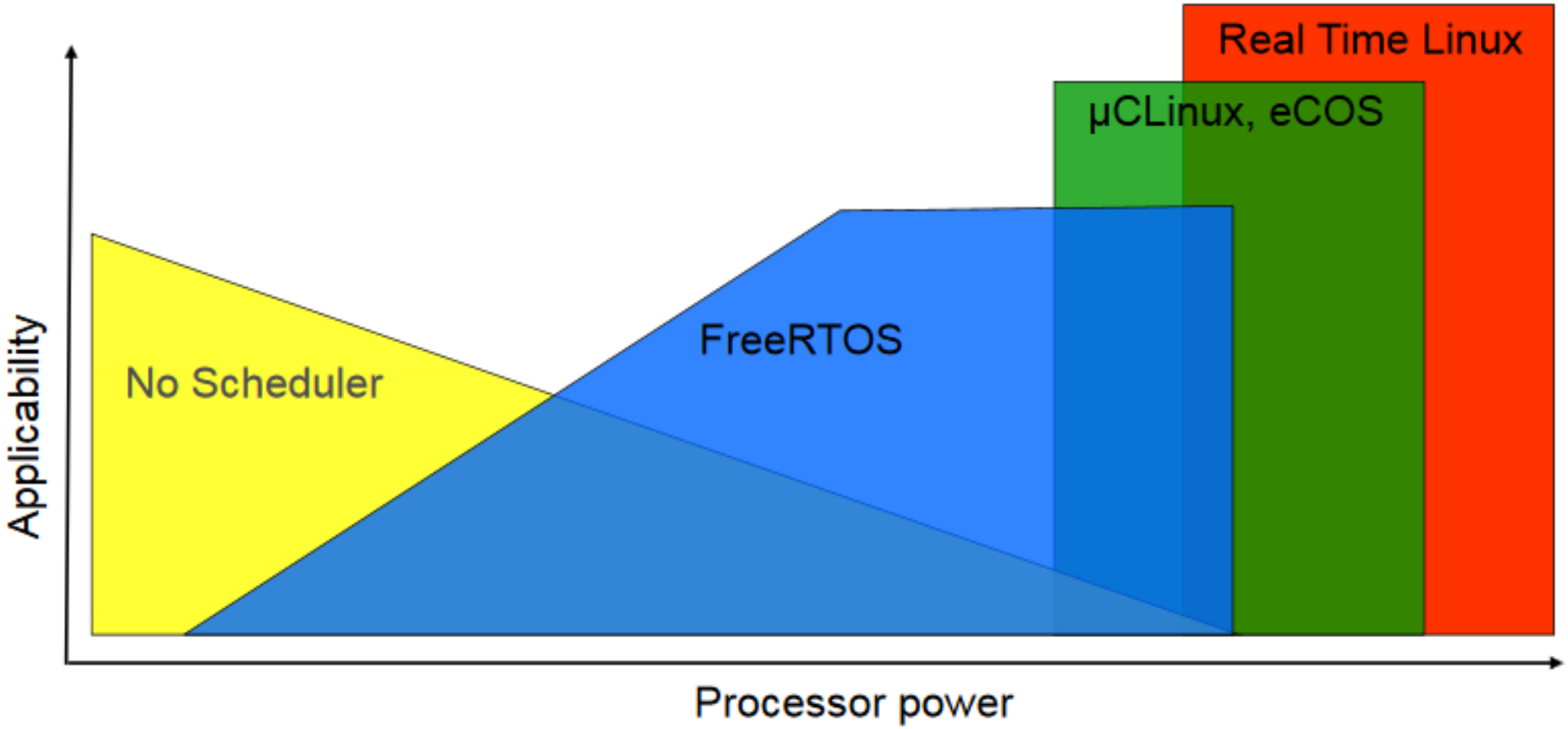
Desacoplado, código funcionalmente coherente.

- Deterministic



For Microcontrollers

- 33 architectures and 18 tool chains



Como arrancamos? www.freertos.org

Quality RTOS & Embedded Software

About Contact Support FAQ Download

FreeRTOS Tutorial Books and Reference Manuals
Become an expert, while supporting the FreeRTOS project

Quick Start | Supported MCUs | Books & Kits | Trace Tools | Ecosystem | TCP & FAT | Training | Email List

- Home
- FreeRTOS Books and Manuals
- FreeRTOS
  - About FreeRTOS
  - Features / Getting Started...
  - More Advanced...
  - Demo Projects
  - Supported Devices & Demos
  - API Reference
    - PDF Reference Manual
    - Task Creation
    - Task Control
    - Task Utilities
    - RTOS Kernel Control
    - Direct To Task Notifications
    - FreeRTOS-MPU Specific
    - Queues
    - Queue Sets
    - Semaphore / Mutexes
    - Software Timers
    - Event Groups (or 'flags')
    - Co-routines
  - Contact, Support, Advertising
- FreeRTOS Interactive!
  - Contributed Vs Official Code

FreeRTOS™

The Market Leading, De-facto Standard and Cross Platform Real Time Operating System (RTOS). Don't Let Your RTOS Lock You In.

Developed in partnership with the world's leading chip companies over a 12 year period, FreeRTOS is the market leading real time operating system (or RTOS), and the de-facto standard solution for microcontrollers and small microprocessors.

With millions of deployments in all market sectors, blue chip companies trust FreeRTOS because it is professionally developed, strictly quality controlled, robust, supported, free to use in commercial products without a requirement to expose proprietary source code, and has no IP infringement risk.

The FreeRTOS value proposition (why use anything else?) . . .
Things you might not know about FreeRTOS . . .
Why choose FreeRTOS? . . .

- World's Most Innovative IoT Solution
- Best in Class RTOS Trace Tools
- Commercial Licensing & Middleware
- Safety Critical Applications
- FreeRTOS Masterclass Training

Latest News:

FreeRTOS V9.0.0 is now available for download.

Ejemplo Tarea

```
/* Tasks always have the same prototype. */
void vProcessMessages( void *pvParameters )
{
  for( ;; )
  {
    xQueueReceive( xQueue, &xMessage, portMAX_DELAY );
    ProcessMessage( &xMessage );
  }

  /* A task cannot exit without first deleting itself. */
  vTaskDelete( NULL );
}
```

Ejemplo tarea periódica

```
/* Tasks always have the same prototype. */
void vProcessMessages( void *pvParameters )
{
  portTickType xLastWakeTime;
  const portTickType xFrequency = 10;

  /* Initialise the xLastWakeTime variable with the
  current time.  */
  xLastWakeTime = xTaskGetTickCount();

  for( ;; )
  {
    /* Wait for the next cycle.  */
    vTaskDelayUntil( &xLastWakeTime, xFrequency );
    vPeriodicProcessingDoneHere();
  }
}
```

Creando una tarea y llamando al Scheduler

```
xTaskCreate(  /* A pointer to the task function. */
              aTask,

              /* Textual name. */
              "LED",

              /* Dimensions of the task stack. */
              configMINIMAL_STACK_DEPTH,

              /* Parameters passed into the task. */
              (void *) 0,

              /* The priority of the task. */
              2,

              /* A handle for the task. */
              NULL
            );

vTaskStartScheduler();
```

Blinking led con FreeRTOS

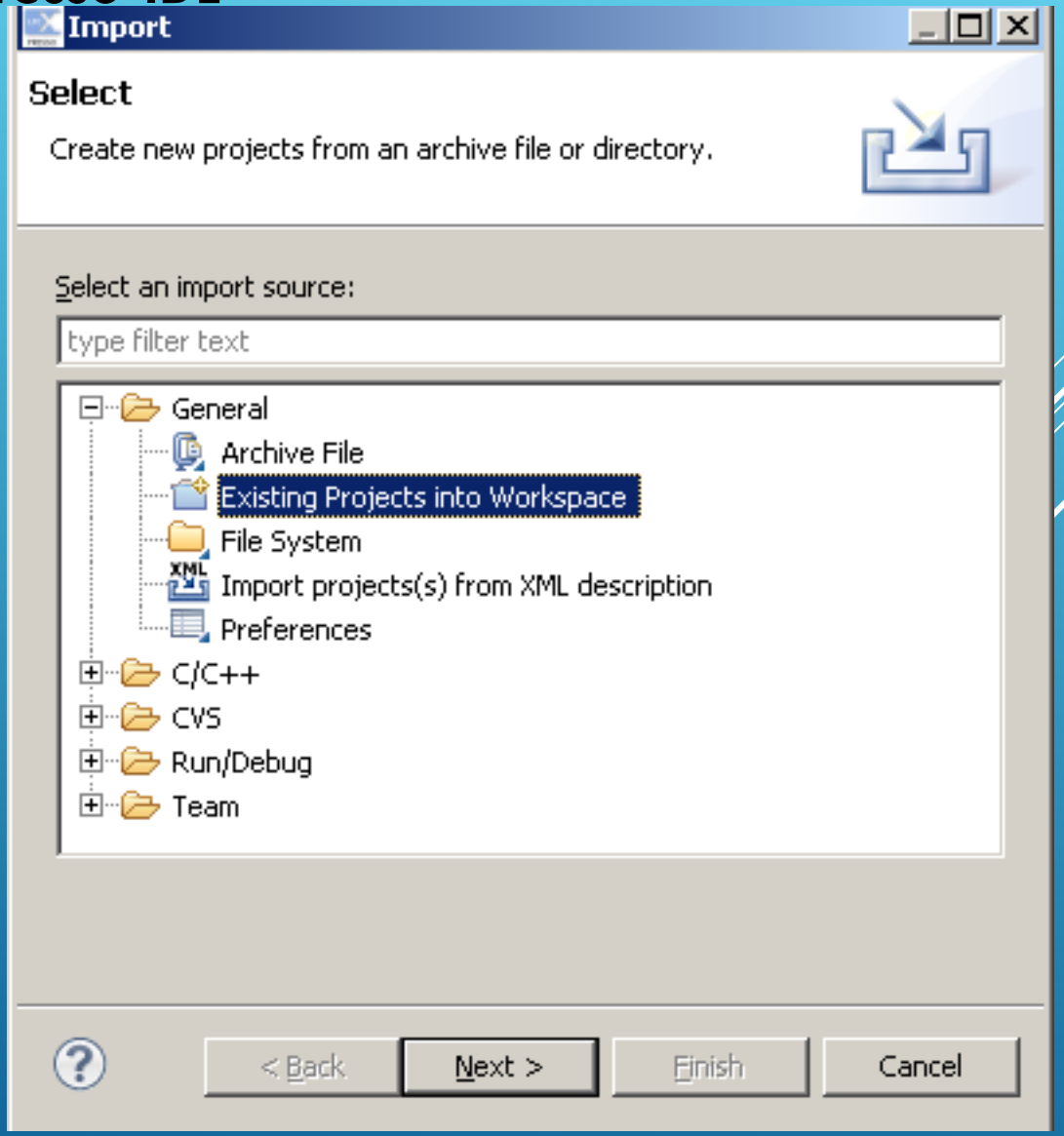
- https://interactive.freertos.org/hc/en-us/community/posts/210027006-Very-simple-LPCXpresso-LPC1768-LPC1343-demo-using-LPCXpresso-IDE

Bajamos el archivo

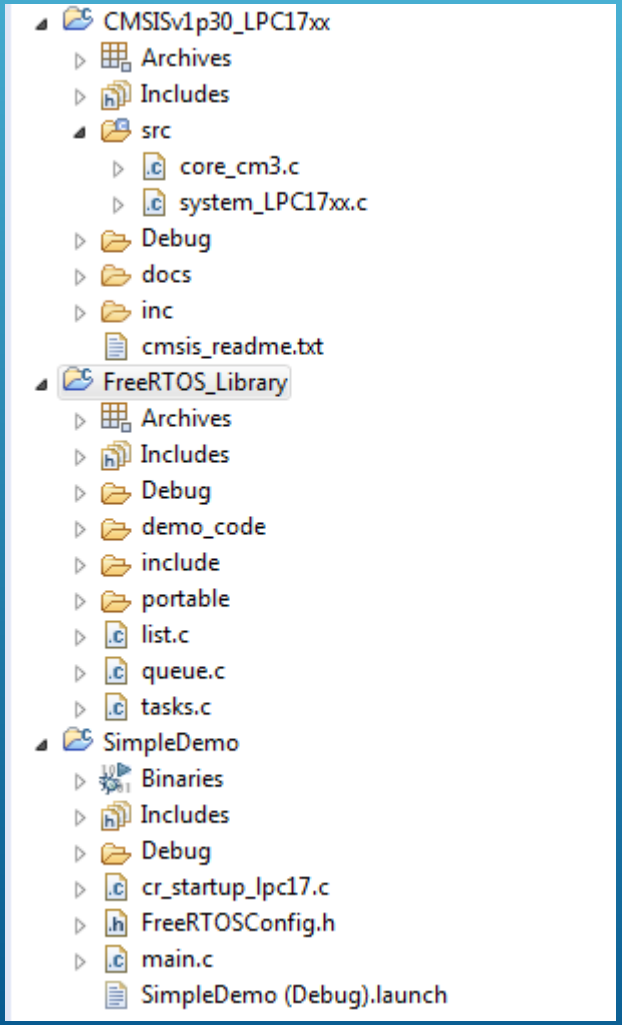
FreeRTOS-simple-demo-for-LPCXpresso-LPC1768.zip

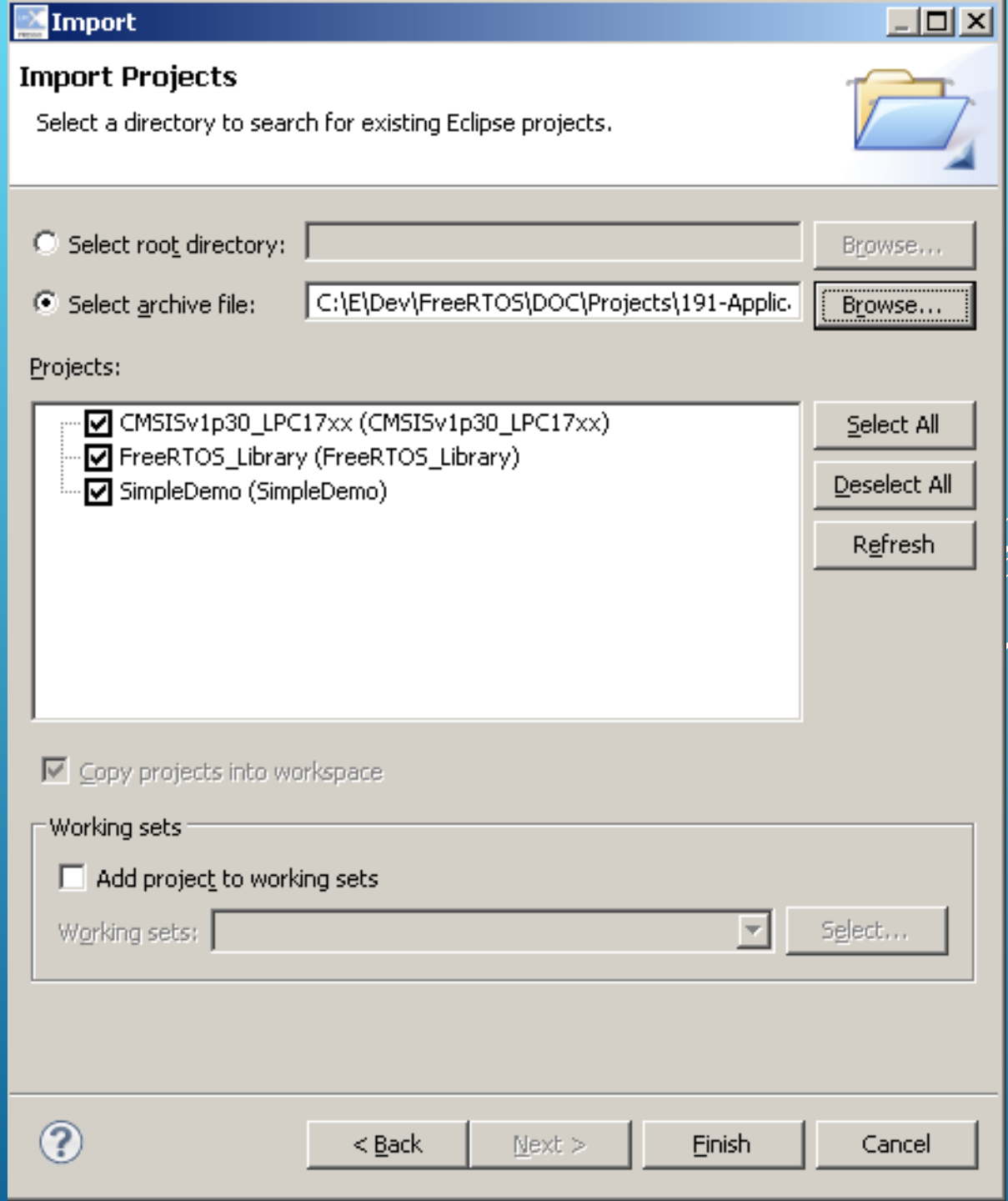
- Abrimos el LPCXpresso y seleccionamos un workspace o hacemos uno si no lo tenemos.

- File -> Import -> Existing Projects into Workspace



- Seleccionamos el archivo .zip que bajamos y nos Aparece la librería CMSIS, FreerTos y el proyecto demo
- Seleccionamos todo, click en finish y ya tenemos Todo listo





Lo primero que hacemos es compilar el programa y … ERROR.

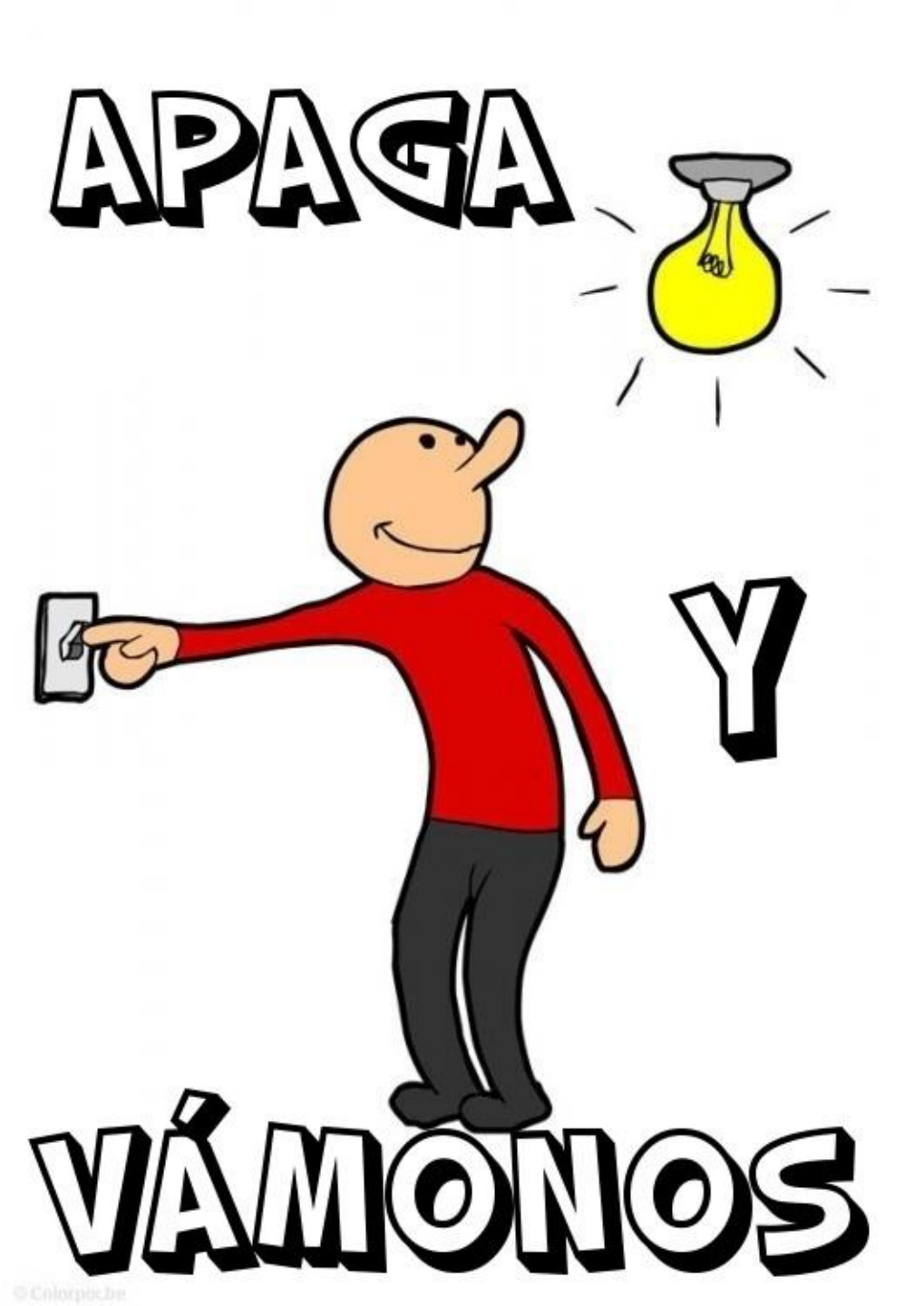
Error: registers may not be the same -- `strexb r3,r2,[r3]'

Error: registers may not be the same -- `strexh r3,r2,[r3]'

Solución

https://www.lpcware.com/content/forum/cmsisv1p30lpc13xx-cannot-be-build-lpcxpresso-ide-ver7

Vamos a la IDE


APAGA
Y
VÁMONOS