

## Seminario de Electrónica: Sistemas Embebidos - Trabajo Práctico N° 3

### LPC43xx Entradas y Salidas (Digitales) de Propósito General (GPIO) – FreeRTOS

#### Objetivo:

- **Uso del IDE** (edición, compilación y depuración de programas)
- **Uso de GPIO & FreeRTOS** (manejo de Salidas y de Entradas Digitales en Aplicaciones)
- **Documentar lo que se solicita en c/ítems**

#### Referencias (descargar del Campus Virtual del curso a fin de usarlas durante la realización del TP):

- Diagrama de Estado: <http://campus.fi.uba.ar/mod/resource/view.php?id=51884>
- [Real Time Operative Systems](#) // [FreeRTOS - Tasks](#) // [FreeRTOS - Interrupt](#) // [FreeRTOS](#)
- Using the FreeRTOS Real Time Kernel, NXP LPC1768 Edition, R. Barry
- FreeRTOS Reference Manual - API Functions and Configuration Options, R. Barry
- LPC435X\_3X\_2X\_1X Product Data Sheet: <http://campus.fi.uba.ar/mod/resource/view.php?id=28519>
- LPC43XX User Manual (Chapter 1, 18 & 19): <http://campus.fi.uba.ar/mod/resource/view.php?id=77765>
- EDU-CIAA-NXP (web site): <http://proyecto-ciaa.com.ar/devwiki/doku.php?id=desarrollo:edu-ciaa:edu-ciaa-nxp>
- EDU-CIAA-NXP (esquemático): [http://www.proyecto-ciaa.com.ar/devwiki/lib/exe/fetch.php?media=desarrollo:edu-ciaa:edu-ciaa-nxp:edu-ciaa-nxp\\_color.pdf](http://www.proyecto-ciaa.com.ar/devwiki/lib/exe/fetch.php?media=desarrollo:edu-ciaa:edu-ciaa-nxp:edu-ciaa-nxp_color.pdf)
- EDU-CIAA-NXP (pinout): [http://proyecto-ciaa.com.ar/devwiki/lib/exe/fetch.php?media=desarrollo:edu-ciaa:edu-ciaa-nxp\\_pinout\\_a4\\_v4r2\\_es.pdf](http://proyecto-ciaa.com.ar/devwiki/lib/exe/fetch.php?media=desarrollo:edu-ciaa:edu-ciaa-nxp_pinout_a4_v4r2_es.pdf)

#### 1. Uso del IDE (Integrated Development Environment) LPCXpresso

- En TP1.1.a ya se Registró, Descargó, Instaló, Ejecutó y Licenció **LPCXpresso IDE v8.2.0** (o posterior)
  - Antes de ejecutar conectar la placa **EDU-CIAA-NXP** a su PC
    - Seleccionar como nombre de Workspace: **workspace-EDU\_CIAA\_NXP-TP3**
    - Mediante **Import project(s)** de los archivo: **LPCXpresso-FreeRTOS-Examples.zip** importar los proyectos:
      - lpc\_chip\_43xx** (librería p/chips LPC43xx de NXP)
      - lpc\_board\_nxp\_lpcxpresso\_4337** (librería p/placas NXP)
      - freertos\_examples\_1\_to\_9** (ejemplos de aplicación)
        - Verifique tener en la carpeta: **freertos\_examples\_1\_to\_9/example/** el archivo:
          - lpc4337.cfg** (opciones de configuración de OpenOCD)
        - Copiar la configuración de **Debug** de **periph\_systick** (ejemplo de aplicación) y adecuarla a **freertos\_examples\_1\_to\_9**
      - freertos\_examples\_10\_to\_16** (ejemplos de aplicación)
        - Verifique tener en la carpeta: **freertos\_examples\_10\_to\_16/example/** el archivo:
          - lpc4337.cfg** (opciones de configuración de OpenOCD)
      - Copiar la configuración de **Debug** de **periph\_systick** (ejemplo de aplicación) y adecuarla a **freertos\_examples\_10\_to\_16**
    - Documentar** mediante tablas c/texto e imágenes la estructura de **archivos**, su tipo/contenido (especialmente **readme.txt**) de c/proyecto importado
  - Documentar** mediante tablas c/texto e imágenes la secuencia de **funciones** invocadas durante la ejecución del ejemplo de aplicación **"Example 1 - Creating tasks"**, en qué archivo se encuentran, su descripción detallada, qué efecto tiene la aplicación sobre el hardware (identificar circuitos, puertos, pines, niveles, etc.) así como la interacción entre las mismas
  - Idem b** pero con **datos** (definiciones, constantes, variables, estructuras, etc.)

#### 2. A partir del proyecto **freertos\_examples\_1\_to\_9** => **Example 1 - Creating tasks**

- Documentar** mediante un diagrama temporal de la distribución del tiempo de CPU entre tareas, Kernel, Interrupciones (**buscar** imagen en: **Using the FreeRTOS Real Time Kernel, NXP LPC1768 Edition, R. Barry**)
- Documentar** observaciones
- Documentar** el **time slice** de FreeRTOS, dónde y cómo modificarlo ([FreeRTOSConfig.h](#))

- d. **Documentar** el efecto de **time slice** sobre las tareas (probar con **1000mS/100mS/10mS/1mS**)
  - e. **Documentar** el criterio a aplicar para la elección del valor de **time slice** para una aplicación
  - f. **Idem a y b p/Example 2 - Using the task parameter**
  - g. **Idem a y b p/Example 3 - Experimenting with priorities**
  - h. **Idem a y b p/Example 4 - Using the Blocked state to create delay**
  - i. **Idem a y b p/Example 5 - Converting the example tasks to use vTaskDelayUntil**
  - j. **Idem a y b p/Example 6 - Combining blocking and non-blocking tasks**
  - k. **Idem a y b p/Example 7 - Defining an idle task hook function**
  - l. **Idem a y b p/Example 8 - Changing task priorities**
  - m. **Idem a y b p/Example 9 - Deleting tasks**
3. A partir del proyecto **freertos\_examples\_10\_to\_16** => **Example 10 - Blocking when receiving from a queue**
- a. **Documentar** mediante un diagrama temporal de la distribución del tiempo de CPU entre tareas, Kernel, Interrupciones (detallar qué ocurre en cada cambio de contexto)
  - a. **Documentar** observaciones
  - b. **Idem a y b p/Example 11 - Blocking when sending to a queue or sending structures on a queue**
  - c. **Idem a y b p/Example 12 - Using a binary semaphore to synchronize a task with an interrupt**
  - d. **Idem a y b p/Example 13 - Using a counting semaphore to synchronize a task with an interrupt**
  - e. **Idem a y b p/Example 14 - Sending and receiving on a queue from within an interrupt**
  - f. **Idem a y b p/Example 15 - Re-writing vPrintString() to use a semaphore**
  - g. **Idem a y b p/Example 16 - Re-writing vPrintString() to use a gatekeeper task**
4. Implementar una aplicación que conste de: una **Tarea 1 periódica** que se ejecute cada **500mS**, una **interrupción (productor)** que se **sincronice** con la **Tarea 2 (consumidor)** mediante un **semáforo** y que la **Tarea 2 (productor)** que se **sincronice** con la **Tarea 3 (consumidor)** mediante una **cola** (ver **diagrama 1**).
- a. **Documentar** la asignación de prioridades
  - b. **Documentar** mediante un diagrama temporal de la distribución del tiempo de CPU entre tareas, Kernel, Interrupciones (detallar qué ocurre en cada cambio de contexto)
  - c. **Documentar** observaciones
5. **Idem 4** pero intercambiando las palabras **semáforo** ⇔ **cola** (ver **diagrama 2**).
6. **Idem 4** pero con 3 (tres) tareas que comparten el uso del **LED**, cada tarea enviará una secuencia fija de 1s y 0s (mayor que **time slice**) y no deben mezclarse las secuencias (ver **diagrama 3**).

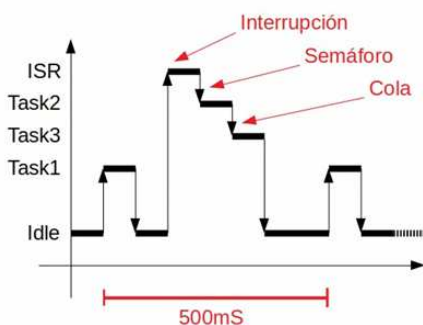


diagrama 1

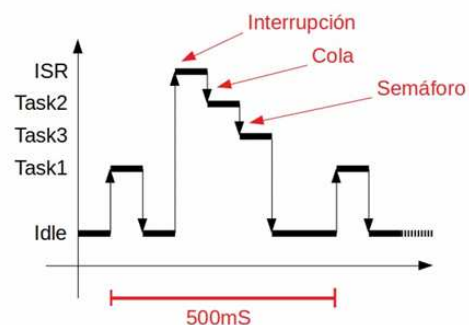


diagrama 2

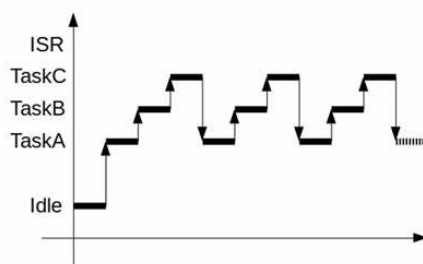


diagrama 3