Seminario de Electrónica: Sistemas Embebidos - Trabajo Práctico N° 4

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 Vitual del curso a fin de usarlas durante la realización del TP)**:**

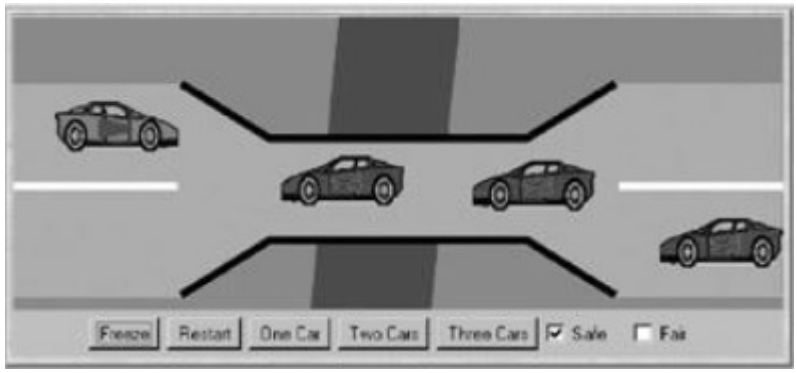
- **Real Time Operative Systems // FreeRTOS - Tasks // FreeRTOS - Interrupt // FreeRTOS**
- **Using the FreeRTOS Real Time Kernel, NXP LPC1768 Edition**, R. Barry
- **FreeRTOS Reference Manual - API Functions and Configuration Options**, R. Barry
- **LPC435X_3X_2X_1X Product Data Sheet:** http://campus.fi.uba.ar/mod/resource/view.php?id=28519
- **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
- **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

1. Uso del **IDE** (Integrated Development Environment) **GNU MCU Eclipse** (p/Linux o p/Windows)
   a. Previo a éste TP ya se Descargó, Instaló, Ejecutó, Configuró y Licenció todo lo necesario para desarrollar éste TP
      i. Instaló **CIAA-LAUCHER**, ejecutó **GNU MCU Eclipse**, agregó **firmware_v3**, configuró Debug para EDU-CIAA-NXP, agregó el plug-in de **eGIT**, agregó **Yakindu StateChart Tools** (**STC**), gestionó la licencia respectiva y al recibirla la cargó
      ii. Antes de ejecutar asegúrese tener conectada la placa **EDU-CIAA-NXP** a su PC (recuerde conectarla **siempre al mismo puerto USB**) a través de la interfaz **Debug**
         1. Seleccionar como nombre de Workspace: **C:\CIAA\CIAA_Software_1.1-Win\workspaces\eclipse-ws**
         2. En el archivo **program.mk** podrá configurar el **programa** en el que se trabajara:
            PROGRAM_PATH = **examples/c/freertos_book**
            PROGRAM_NAME = **Example001**
         3. En el archivo **board.mk** podrá configurar la **placa** a utilizar:
            BOARD = **edu_ciaa_nxp**
         4. **Documentar** mediante tablas c/texto e imágenes la estructura de **archivos**, su tipo/contenido (especialmente **readme.txt**) de c/proyecto importado
   b. **Documentar** mediante tablas c/texto e imágenes la secuencia de **funciones** invocadas durante la ejecución del ejemplo de aplicación "**Example001 - 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
   c. **Idem b** pero con **datos** (definiciones, constantes, variables, estructuras, etc.)

2. Tome **3** ejemplos del **Example001**al **Example009** (de libre elección)
   a. **Example001 - Creating tasks**
   b. **Example002 - Using the task parameter**
   c. **Example003 - Experimenting with priorities**
   d. **Example004 - Using the Blocked state to create delay**
   e. **Example005 - Converting the example tasks to use vTaskDelayUntil**
   f. **Example006 - Combining blocking and non-blocking tasks**
   g. **Example007 - Defining an idle task hook function**
   h. **Example008 - Changing task priorities**
   i. **Example009 - Deleting tasks**
      i. **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)
      ii. **Documentar** observaciones
      iii. **Documentar** el **time slice** de FreeRTOS, dónde y cómo modificarlo (**FreeRTOSConfig.h**)
      iv. **Documentar** el efecto de **time slice** sobre las tareas (probar con **1000mS**/**100mS**/**10mS**/**1mS**)
      v. **Documentar** el criterio a aplicar para la elección del valor de **time slice** para una aplicación

3. Tome **2** ejemplos del **Example010** al **Example016** (de libre elección)
    a. **Example010 - Blocking when receiving from a queue**
    b. **Example011 - Blocking when sending to a queue or sending structures on a queue**
    c. **Example012 - Using a binary semaphore to synchronize a task with an interrupt**
    d. **Example013 - Using a counting semaphore to synchronize a task with an interrupt**
    e. **Example014 - Sending and receiving on a queue from within an interrupt**
    f. **Example015 - Re-writing vPrintString() to use a semaphore**
    g. **Example016 - Re-writing vPrintString() to use a gatekeeper task**
        i. **Documentar** mediante un diagrama temporal de la distribución del tiempo de CPU entre tareas, Kernel, Interrupciones (detallar qué ocurre en cada cambio de contexto)
        ii. **Documentar** observaciones

4. Se trata de **una ruta de doble sentido** que llega a un **puente estrecho** por el que **cabe un solo vehículo**.



Implementar un **Sistema de Control de Acceso** que **monitoree & controle** la entrada y salidas de vehículos por uno y otro lado del puente estrecho.

Para la implementación se sugieren dos tareas (una para c/u de las entradas al puente estrecho), a saber:

- void **vNorthGatewayTask** (void * **pvParameters**);
- void **vSouthGatewayTask** (void * **pvParameters**);

- Observaciones:
    - Al puente estrecho **se ingresa por orden de llegada**
    - Por el puente estrecho **circula un solo vehículo a la vez**
    - Se cuenta con una tercer tarea (a falta de hardware para generar estímulos), que periódicamente **genera estímulos** para **probar** el funcionamiento de las dos **tareas** que **monitorean & controlan** los respectivos acceso al **puente estrecho**:
        - void **vTestingTask** (void * **pvParameters**);
            - ***Estímulos***:
                - ***North_Entry***
                - ***South_Enntry***
                - ***Exit***

- **examples/c/projects/TP4** (**TP4.zip => FreeRTOSConfig.h**, **main.h** => **escenario de Testing**, **main.c** => **implementación**)

- Indique **modificaciones** necesarias para gobernar dos **semáforos vehiculares** (**Rojo**, **Verde**), uno en c/u de las entradas al puente estrecho

IMPORTANTE: Se recomienda seguir los siguientes pasos:
- Comenzar por una de las tareas (sincronizar con los estímulos generados por **vTestingTask**())
- Luego agregar la otra tarea (sincronizar con los estímulos generados por **vTestingTask**())
- Luego resolver el uso del recurso compartido
- Por último resolver los semáforos vehiculares (Rojo, Amarillo, Verde) uno en c/u de las entradas al puente estrecho