

Seminario de Electrónica: Sistemas Embebidos - Trabajo Práctico N° 1

LPC43xx Entradas y Salidas (Digitales) de Propósito General (GPIO)

Objetivo:

- **Uso del IDE** (edición, compilación y depuración de programas)
- **Uso de GPIO** (manejo de Salidas y de Entradas Digitales)
- **Comprender la arquitectura de software de un proyecto LPCXpresso** nativo con librerías LPCOpen
- **Comprender las librerías LPCOpen de chip (lpc_chip_43xx) & de placa NXP (lpc_board_nxp_lpcxpresso_4337)**
- **Adaptar la librería LPCOpen de placa NXP a la placa EDU-CIAA-CIAA-NXP**
- **Documentar lo que se solicita en c/ítems**

Referencias (descargar a fin de usarlas durante la realización del TP):

- **LPCXpresso-Intro:** http://campus.fi.uba.ar/pluginfile.php/155949/mod_resource/content/4/Sistemas_Embebidos-2016_2doC-LPCXpresso-Intro-Cruz.pdf
- **LPCXpresso-Salidas:** http://campus.fi.uba.ar/pluginfile.php/156011/mod_resource/content/4/Sistemas_Embebidos-2016_2doC-LPCXpresso-Salidas-Cruz.pdf
- **LPCXpresso-Entradas:** http://campus.fi.uba.ar/pluginfile.php/156013/mod_resource/content/4/Sistemas_Embebidos-2016_2doC-LPCXpresso-Entradas-Cruz.pdf
- **LPCXpresso-Systick:** http://campus.fi.uba.ar/pluginfile.php/156031/mod_resource/content/5/Sistemas_Embebidos-2016_2doC-LPCXpresso-Systick-Cruz.pdf
- **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) LPCXpresso

- a. Instale **OpenOCD 0.10.0** desde el siguiente sitio (seleccione el instalador según su Sistema Operativo): <https://github.com/gnuarmclipse/openocd/releases/tag/gae-0.10.0-20160110>
- b. Registrarse, Descargar, Instalar, Ejecutar y Licenciar **LPCXpresso IDE v8.2.0** (o posterior) www.nxp.com/products/software-and-tools/software-development-tools/software-tools/lpc-microcontroller-utilities/lpcxpresso-ide-v8.2.0:LPCXPRESSO
 - i. Es necesario tener la **EDU-CIAA-NXP** conectada a su PC (recuerde conectarla **siempre al mismo puerto USB**) a través de la interfaz **Debug** para **instalar VCP driver** del chip **U6 FT2232-H** (<http://www.ftdichip.com/Drivers/VCP.htm>) (convertor USB-Serie) **DEBUG**, y configurarlo (vincularlo con Windows mediante **Zadig** <http://zadig.akeo.ie/>) como se explica en:
http://proyecto-ciaa.com.ar/devwiki/lib/exe/fetch.php?media=docu:fw:bm:ide:installciaa_ide_windows_v1.0.pdf o http://proyecto-ciaa.com.ar/devwiki/lib/exe/fetch.php?media=docu:fw:bm:ide:installciaa_ide_linux_v1.0.pdf
 - ii. Dentro de LPCXpresso, agregue el plug-in **OpenOCD Debugging**
Menú **Help** → **Install New Software ...** Work with: <http://gnuarmclipse.sourceforge.net/updates>
Seleccione el plug-in y luego siga las instrucciones del asistente (**GNU ARM C/C++ OpenOCD Debugging**).
 - iii. 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: **workspace-EDU_CIAA_NXP-TP1**
 2. Mediante **Import project(s)** del archivo: **lpcopen_2_16_lpcxpresso_nxp_lpcxpresso_4337.zip** importar los proyectos:
 - a. **lpc_chip_43xx** (librería p/chips LPC43xx de NXP)
 - b. **lpc_board_nxp_lpcxpresso_4337** (librería p/placas NXP)
 - c. **periph_systick** (ejemplo de aplicación)
 3. Mediante **Project** → **Clean** → **OK** (Build the entire workspace) compilar los proyectos importados (en Debug se genera el archivo: **periph_systick.axf**)
 4. Copiar en la carpeta: **periph_systick/example/** el archivo:
 - a. **lpc4337.cfg** (opciones de configuración de OpenOCD)
<http://campus.fi.uba.ar/mod/resource/view.php?id=77758>
 5. Configuración de **Debug**:
 - a. Clic derecho en workspace → **Debug As** → **Debug Configurations...**
 - b. Doble clic en **GDB OpenOCD Debugging**
 - c. Clic en **Search Project...** seleccione el archivo **periph_systick.axf** (si no aparece automáticamente)

- d. Pestaña **Debugger**, sección OpenOCD Setup, haga clic en **Browse...** para navegar a la carpeta de instalación de OpenOCD, luego a la carpeta bin y finalmente al ejecutable **openocd.exe**
 - e. **Config options** ingrese el siguiente texto: **-f example /lpc4337.cfg**
 - f. Sección **GDB Client Setup**, en el campo **Ejecutable** escriba: **arm-none-eabi-gdb**
 - g. Clic en **Apply**, luego en **Debug**, debería comenzar la sesión de Debug con un **breakpoint** en la primer línea de la función **main()**
 6. **Documentar** mediante tablas c/texto e imágenes la estructura de **archivos**, su tipo/contenido (especialmente **readme.txt**) de c/proyecto importado
 7. **Documentar** mediante tablas c/texto e imágenes la **secuencia de comandos**: Clean **periph_systick** -> Build **periph_systick** -> Debug **periph_systick** -> Ejecutar **periph_systick** (ejemplo de aplicación)
 - a. Completo (**Resume**), detener (**Suspend**) y resetear (**Restart**)
 - b. Por etapas colocando **breakpoints** (**Resume**)
 - c. Por línea de código (**Step Into**, **Step Over**, **Step Return**)
 - d. Recuerde siempre abandonar Debug (**Terminate**) antes de Editar o Compilar algún archivo, o Abandonar el IDE (**Exit**)
2. A partir del proyecto **periph_systick** (parpadeo del **LEDs** de **LPCOpen**)
- a. Identificar funciones de librería **lpc_board_nxp_lpcxpresso_4337** útiles para el **parpadeo de un led**
 - b. **Documentar** mediante tablas c/texto e imágenes la secuencia de **funciones** invocadas durante la ejecución del ejemplo de aplicación, 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 (tanto en **ResetISR()** como en **main()**)
 - c. **Idem b** pero con **datos** (definiciones, constantes, variables, estructuras, etc.) (tanto en **ResetISR()** como en **main()**)
 - a. **Documentar** cuál es el **período de parpadeo**, modifique el ejemplo de aplicación para soportar además del período de parpadeo original **100mS**, **10mS** y **1mS** (sin números mágicos y sin perder portabilidad)
 - b. **Modificar** el **ejemplo de aplicación** para que el período de parpadeo sea diferente del período de **SysTick_Handler** (sin números mágicos y sin perder portabilidad), migrar la **función de acción** sobre los **LEDs & RGB-LEDs** a **main()**
 - c. **Documentar** la reducción de la frecuencia de reloj a ¼ del máximo y su impacto en el ejemplo de aplicación
3. A partir del proyecto **periph_systick** (parpadeo del **LEDs & RGB-LEDs** de **EDU-CIAA-NXP**)
- a. En caso que no funcione el parpadeo de **LEDs** **modificar** las librerías para hacer parpadear todos los **LEDs** (cuidado con el uso de recursos asignados para otros fines en la librería **lpc_board_nxp_lpcxpresso_4337**)
 - b. **Modificar** el **ejemplo de aplicación** para soportar los **LEDs + RGB-LEDs** (encender/apagar LED -**250mS/250mS**- uno a la vez en secuencia)
4. A partir del proyecto **periph_systick** (sensado de **Push Buttons** de **LPCOpen**)
- a. Identificar funciones de librería **lpc_board_nxp_lpcxpresso_4337** útiles para el **sensado de un pulsador**
 - b. **Documentar** mediante tablas c/texto e imágenes la secuencia de **funciones** invocadas durante la ejecución del sensado de un pulsador, 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 (tanto en **ResetISR()** como en **main()**)
 - c. **Idem b** pero con **datos** (definiciones, constantes, variables, estructuras, etc.) (tanto en **ResetISR()** como en **main()**)
5. A partir del proyecto **periph_systick** (sensado de **Push Buttons** de **EDU-CIAA-NXP**)
- a. En caso que no funcione el sensado de **Push Buttons** **documentar** mediante tablas c/texto e imágenes la modificación de las librerías para sensar todos los **pulsadores** (cuidado con el uso de recursos asignados para otros fines en la librería **lpc_board_nxp_lpcxpresso_4337**)
 - b. **Modificar** el **ejemplo de aplicación** para soportar los **pulsadores**, (encender/apagar LED + RGB-LED -**250mS/250mS**- uno a la vez en secuencia e invertirla mediante la opresión de pulsadores)
 - c. En caso que no funcione correctamente el ejemplo de aplicación **documentar** la forma de la señal digital “pulsador” a sensar: no oprimido // transición a oprimido // mantener oprimido // transición a no oprimido // ...
6. A partir del proyecto **periph_systick** (máquinas de estado para dispositivos de **EDU-CIAA-NXP**)
- a. Para usar **UART2** como interfaz de **DEBUG**, instalar y ejecutar en su PC un programa de **terminal** comunicación serie, p/e Putty y el driver del chip **U6 FT232R-H** (convertor USB-Serie) **DEBUG**
 - b. En caso que no funcione **DEBUG** **documentar** mediante tablas c/texto e imágenes la **modificación de las librerías** para hacerlo funcionar sobre **UART2** (cuidado con el uso de recursos asignados para otros fines en la librería **lpc_board_nxp_lpcxpresso_4337**)
 - c. **Modificar** el **ejemplo de aplicación** para implementar la máquina de estado: driver de **LED** o **RGB-LED** (use la interfaz de **DEBUG** para indicar “evento/estado/transición/...”)
 - d. **Modificar** el **ejemplo de aplicación** para implementar la máquina de estado: driver de **pulsador** (use la interfaz de **DEBUG** para indicar “evento/estado/transición/...”)