

Práctica de Laboratorio #5 Puertos Paralelos en Lenguaje Ensamblador

Objetivo

Que el estudiante aprenda el procedimiento para desarrollar aplicaciones en lenguaje ensamblador para el microcontrolador LPC1115, utilizando el IDE Keil μ Vision.

Teoría

El microcontrolador LPC1115/303 utiliza el CPU *Cortex-M0* de ARM, el cual soporta el set de instrucciones *Thumb*. En esta práctica usted experimentará con una aplicación para el LPC1115 desarrollada en lenguaje ensamblador, que es equivalente a la que utilizó en la práctica de laboratorio #3 (Tarjetas LPCXpresso Base Board). Para su comodidad, hemos incluido la información pertinente a continuación.

Lenguaje Ensamblador

El CPU Cortex-M0 implementa la arquitectura ARMv6-M, la cual utiliza el set de instrucciones Thumb de 16 bits, con tecnología Thumb-2.

El código que usted verá en esta práctica contiene una cantidad muy reducida de instrucciones, cuyo uso usted debe consultar en la guía que prefiera. Nuestra recomendación siempre es la especificación de ARM (infocenter.arm.com) y el manual de su microcontrolador.

GPIO

El microcontrolador LPC1115/303 cuenta con 4 puertos paralelos bidireccionales de propósito general: Los puertos 0, 1, y 2 (12 pines cada uno), y el puerto 3 (6 pines).

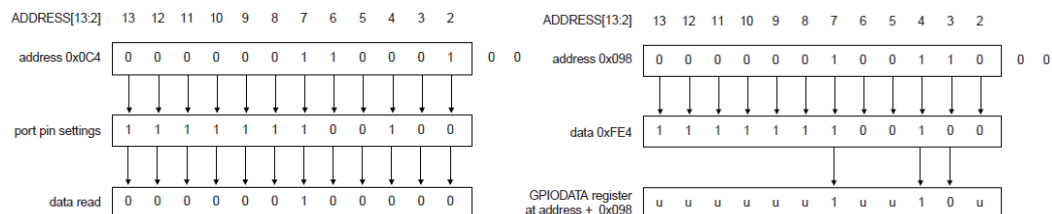
Cada puerto tiene un registro de datos asociado (GPIO2DATA por ejemplo). Si se escribe a este registro, se escribe a los pines configurados como salidas, y los bits configurados como entradas tienen el estado lógico de los pines asociados.

Cada puerto tiene también un registro de dirección de datos (Data Direction Register o **GPIO_nDIR**), que se utiliza para configurar los bits del puerto como entradas o salidas individualmente (**GPIO2DIR** por ejemplo).

Cada pin de cada puerto tiene un registro de configuración llamado **IOCON** (por ejemplo **IOCON_PIO2_11**) que controla, entre otras cosas, la función del pin, habilitación de *Pull Resistors*, y otros modos de operación.

Los cuatro registros de datos soportan escritura y lectura enmascarada por la dirección de memoria. Es decir: La dirección de memoria que utilice

¡Sí, La dirección de memoria se está utilizando para hacer contrabando de información!
A continuación tenemos la lógica de operación de esta característica:



Para conocer más, puede buscar "Masked access" en el *ARM Infocenter*.

Materiales y Equipo (Proporcionados por el Laboratorio)

- Tarjeta LPCXpresso Base Board con una tarjeta LPCXpresso LPC1115.
- PC con Windows 7 y Keil μ Vision 4.
- 2 Cables USB 2.0 a USB Mini-B (no *Micro-B*, sino *Mini-B*).

Procedimiento

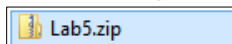
Siga los siguientes pasos prestando atención a las indicaciones de su instructor. Si tiene alguna duda o no está seguro de cómo proceder, pregunte a su instructor.

1. Asegúrese de conocer el hardware con el que trabajará. En particular verifique lo siguiente:
 - a. Mantenga la tarjeta alejada de piezas metálicas como joyas y llaves.
 - b. Mantenga la tarjeta alejada de piezas metálicas como joyas y llaves.
 - c. Note que en adelante nos referiremos a dos tarjetas:
 - i. La tarjeta LPCXpresso Base Board, que contiene periféricos.
 - ii. La tarjeta LPCXpresso LPC1115, que contiene:
 1. El hardware LPC-Link, para descargar programas.
 2. El microcontrolador LPC1115 de NXP.



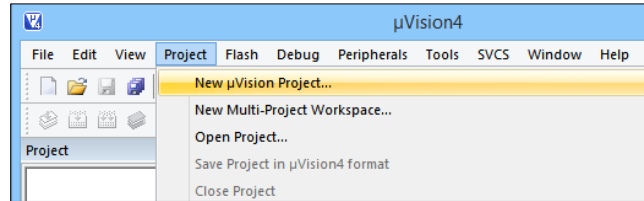
Tarjeta LPCXpresso LPC1115, de Embedded Artists

- d. La tarjeta LPCXpresso Base Board es alimentada por el conector USB Mini-B llamado X1 (Cerca del conector Ethernet RJ45), sin interruptor de encendido. No la conecte a la PC todavía.
 - e. La tarjeta LPCXpresso LPC1115 es programada por su conector USB Mini-B llamado J3 (El único). No la conecte a la PC todavía.
 - f. Para conectar los extremos USB Mini-B de los cables, asegúrese de sujetar *el conector hembra* y **NO** la tarjeta. Si no entendió esto, pregunte a su instructor.
 - g. Si en algún momento necesita apagar las tarjetas temporalmente, desconecte los cables **de sus extremos de la PC** (USB standard). No manipule los extremos de los cables que se conectan a las tarjetas más que para iniciar y finalizar su práctica. Cuando lo haga, recuerde hacerlo sujetando el conector y **NO** la tarjeta.
2. Descomprima la carpeta **Lab5.zip** donde prefiera.

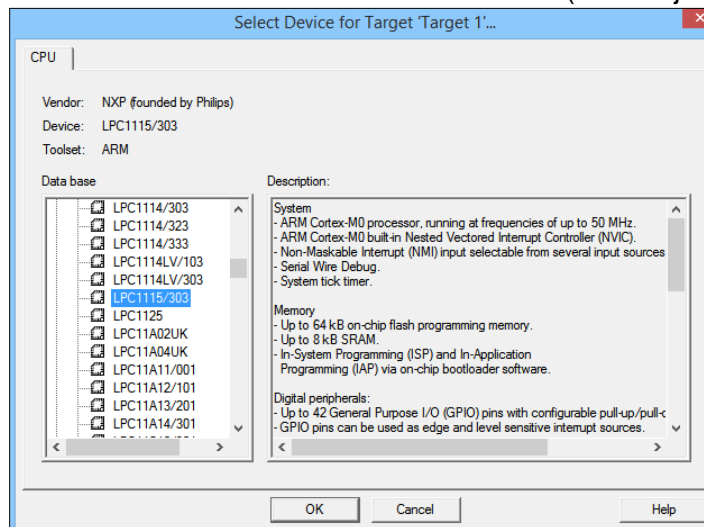


3. En esta práctica usted creará un proyecto en Keil “desde cero” para desarrollar en lenguaje ensamblador.

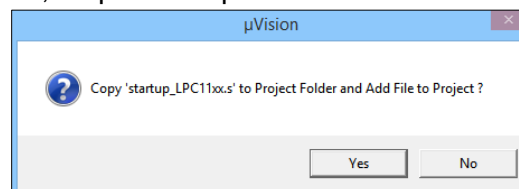
Cree un proyecto nuevo en Keil, en un directorio vacío creado para ese proyecto. Póngale un nombre sencillo, sin espacios ni caracteres especiales, ni longitudes grandes. No olvide utilizar la extensión **.uvproj**



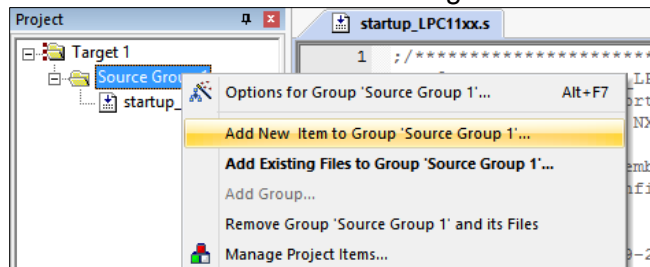
4. Seleccione el microcontrolador LPC1115 como destino (Está bajo NXP).



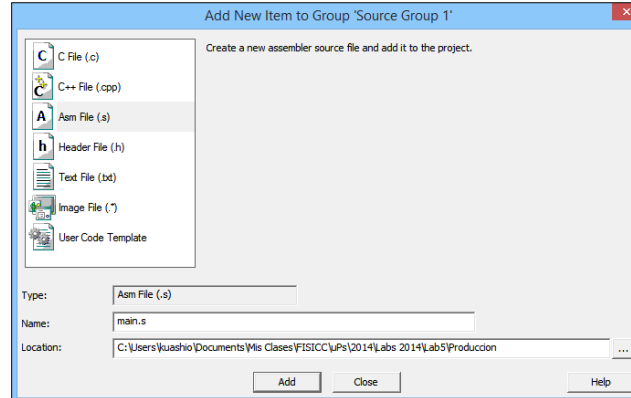
5. Cuando Keil le pregunte si desea copiar a su proyecto el archivo **startup_LPC11xx.s**, respóndale que sí.



6. Agregue un archivo nuevo a su directorio de código fuente.



7. Informe a la caja de diálogo que desea agregar un archivo de lenguaje ensamblador, y nómbrelo **main.s**

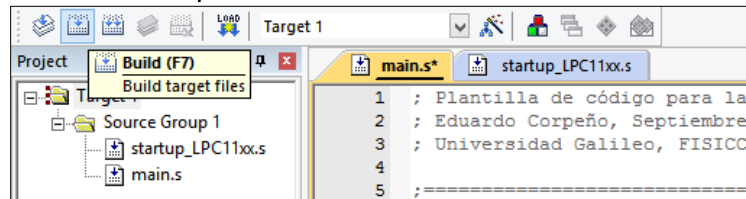


8. En este nuevo archivo, copie y pegue el contenido completo del archivo **main.s** proporcionado. Guarde su trabajo (**File->Save All**).
9. Este es el paso más importante de esta práctica. Se le recomienda llevarlo a cabo antes de presentarse al laboratorio.

Lea el código proporcionado. Trate de entender todo su contenido a excepción de la primera sección. A partir de la rutina principal, usted debe entender todo el código. No le recomendamos dejar esto para el momento de su práctica, ya que si tiene dudas, su instructor probablemente estará atendiendo a sus compañeros y no le podrá ayudar tan pronto como usted lo necesite.

Revise la especificación de las instrucciones utilizadas en las fuentes recomendadas.

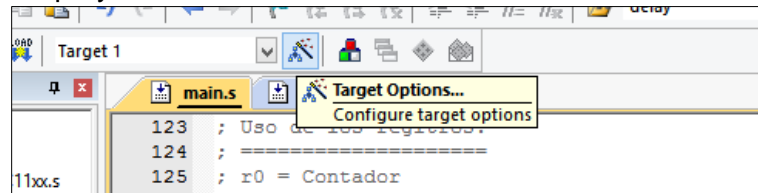
10. Ejecute el comando **Build** para ensamblar.



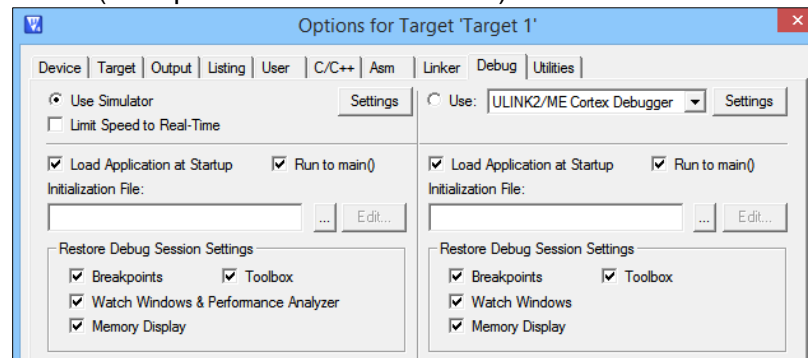
11. Revise la sección inferior de Keil, llamada "**Build Output**".
- ¿Obtuvo errores? ¿Cuáles?
 - ¿Obtuvo advertencias? ¿Cuáles?
12. Pruebe "descomentar" la penúltima línea y ensamble de nuevo.
- ¿Cambiaron los errores o advertencias?
13. Ahora pruebe "descomentar" también la última línea y ensamble de nuevo.
- ¿Cambiaron los errores o advertencias?
14. ¿A qué atribuye lo sucedido? Si en su reporte no responde a las preguntas de este documento, usted renuncia a la mitad de su nota para esta práctica.

15. ¡Ahora vamos a simular su código!

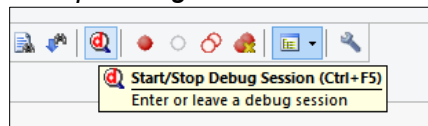
Para esto, es necesario configurar algunas cosas. Abra la caja de diálogo de opciones de proyecto.



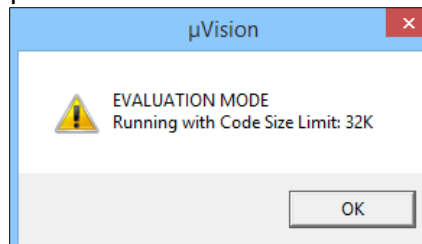
16. Seleccione la pestaña “Debug” y seleccione el *Radio Button* de la izquierda, “Use Simulator” (Este podría o no ser el default). Presione OK.



17. Presione el botón “Start/Stop Debug Session” en la barra de herramientas.

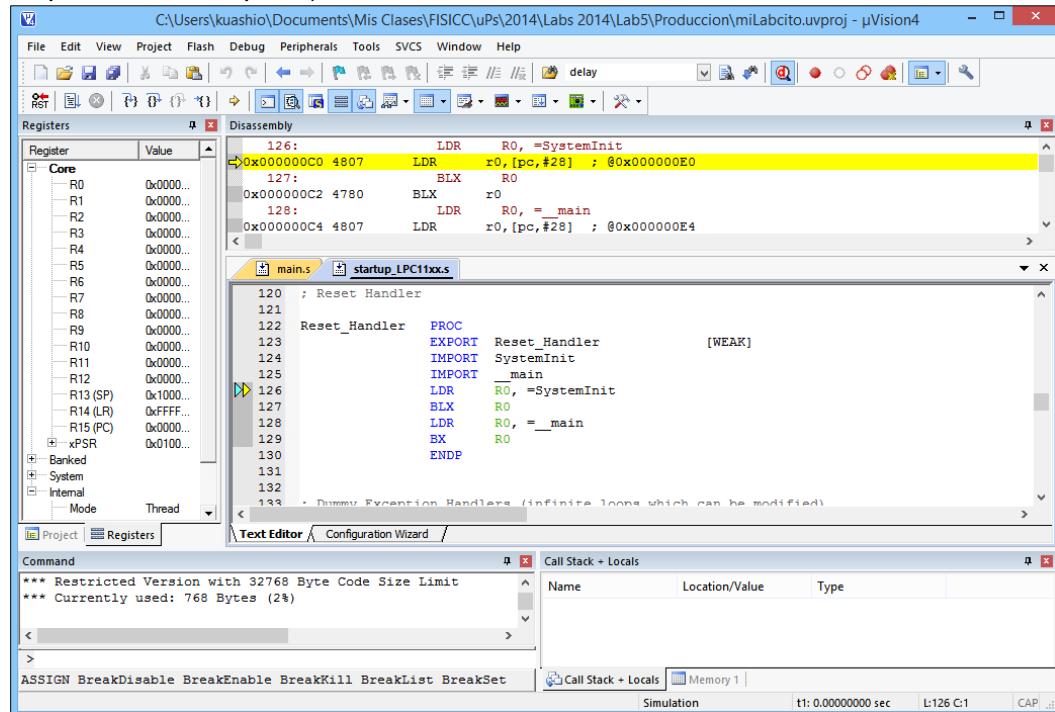


18. Nuevamente, este es el precio de contar con una licencia limitada:



Recuerde que si sus archivos ejecutables exceden los 32kB, siempre puede utilizar el ambiente de desarrollo LPCxpresso de NXP, diseñado específicamente para estas tarjetas.

19. Ahhh. ¡Mire el debugger! ¿No es esta es una de las razones por las cuales usted regresa día a día a su casa de estudios? ¿Acaso no es por esto que no ha renunciado a la programación de computadoras a estas alturas? ¿No siente que nació para hacer esto luego de ver ese debugger? ¿No le dan ganas de insertar algunos *breakpoints* y *watches*, ver la memoria y hacer ejecución paso a paso? (Estas son preguntas retóricas. En consecuencia, no esperamos que las responda en su reporte).



20. Antes de ejecutar el programa, ponga atención a las ventanas desplegadas. Especialmente, tres de ellas le servirán: *Registers*, *Disassembly*, y el *Código Fuente*.

- *Registers* contiene todos los registros del CPU para ver y modificar su valor en todo momento.
- *Disassembly* contiene lo que se conoce como el *Listado* del programa. Esto es un archivo de texto que muestra para cada línea de lenguaje ensamblador:
 - La dirección de memoria donde está almacenada.
 - El código hexadecimal almacenado.
 - La línea de código desensamblada (sin símbolos).
 - La línea de código original en su lenguaje original, tal cual.
- *Código Fuente* contiene los archivos que escribió y librerías que utilizó. Según la próxima línea a ejecutar, muestra el archivo que la contiene.

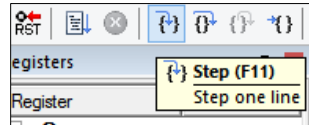
Siéntase en la libertad de modificar el tamaño de estas ventanas a su gusto.

21. Explore el debugger hasta que se sienta en la comodidad de utilizarlo para depurar su código como lo ha hecho en otros IDEs. Revise las herramientas y descubra cómo insertar breakpoints, agregar watches, explorar el contenido de la memoria, etc. No debe hacer todas estas cosas, sólo revise las herramientas.

22. Haga ejecución paso a paso de este programa prestando atención a las instrucciones a partir de la rutina **main**, con el siguiente procedimiento:

Antes de ejecutar cada instrucción, haga una hipótesis de lo que debe pasar en los registros y compruebe si tenía la razón. Cada instrucción debe tomarle mucho menos de un minuto.

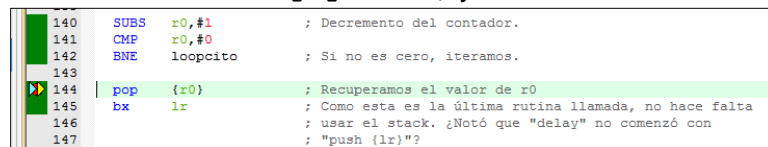
Preste especial atención a las instrucciones de llamado (BL) y retorno (BX) de subrutinas. Note qué para con el Program Counter y el Link Register.



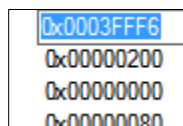
23. Cuando llegó al ciclo **loopcito** de la rutina **delay**, debió darse cuenta de que cuesta salir. Podemos hacer “fast forward” de este ciclo, pero nos interesa detener la ejecución cuando finalice el ciclo (para no perdernos), ya que no sabemos en dónde encontraremos la ejecución si simplemente ponemos al simulador a hacer ejecución continua y lo detenemos cuando queramos.

Para lograr este objetivo, usted tiene tres opciones sensatas (o tal vez más) mientras se encuentra dentro del ciclo (son mutuamente excluyentes):

- Poner un *breakpoint* en la línea **pop {r0}**, y hacer ejecución continua.
- Click derecho en la línea **pop {r0}**, y seleccionar “Run to Cursor line”.

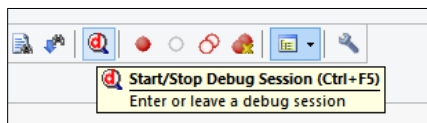


- Modificar el valor del registro que sirve como contador para que tenga un valor cercano al que le permite finalizar el ciclo, y ejecutar paso a paso hasta que eso suceda.




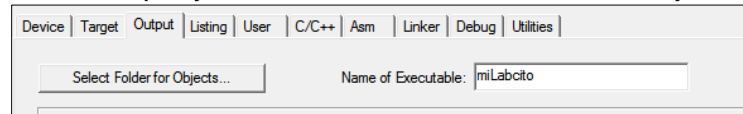
24. Continúe ejecutando paso a paso hasta llegar a la segunda iteración del ciclo principal llamado **loop**.

25. ¡Muy bien hecho! Presione el botón “Start/Stop Debug Session” nuevamente para detener la simulación.

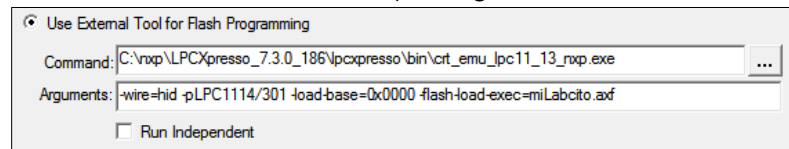


26. ¡Ahora vamos a descargar el programa a la tarjeta LPCXpresso!

27. Presione el botón  para abrir la caja de diálogo de *Opciones* de su proyecto. Vaya a la pestaña *Output* y tome nota del nombre del archivo ejecutable.



28. A continuación, vaya a la pestaña *Utilities* y copie el comando y argumento del archivo **config.txt** proporcionado. Como habrá notado, el nombre del ejecutable debe aparecer al final del argumento en lugar del texto **NOMBRE** (Recuerde conservar la extensión **.axf**). Asegúrese de hacer esto bien.



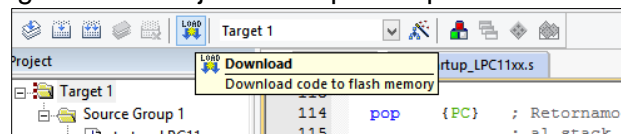
29. Ahora estamos listos para conectar la tarjeta LPCXpresso. Alimente ambas tarjetas como aprendió anteriormente. Recuerde hacerlo sujetando los conectores hembra en la placa para no esforzarlos.

30. Probablemente la tarjeta LPCXpresso LPC1115 tendrá su LED titilando, lo cual no es su programa, sino uno anteriormente cargado. No le haga caso.

31. Ahora arranque el driver LPC-Link en la PC ejecutando el script proporcionado **bootLPCLink.bat**. Recuerde que para revisar si el driver está funcionando, puede utilizar la aplicación *Devices and Printers*.



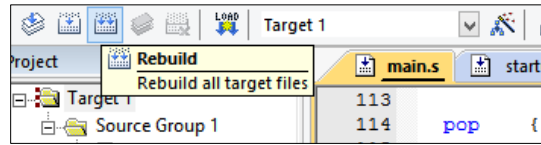
32. Descargue el programa a la tarjeta LPCXpresso presionando el botón *Download*.



33. Después de la descarga, la tarjeta LPCXpresso debe encender y apagar su LED a una frecuencia notable de unos Hertz.

34. Para verificar la efectividad de su descarga, vaya al código fuente y ubique la asignación de la constante que determina la duración de la rutina **delay**. Apunte o copie el valor original en esta asignación.

35. Primero reemplace el valor asignado por $\frac{1}{4}$ de su valor original, luego presione *Rebuild* para volver a ensamblar su código.



36. Descargue el nuevo programa a la tarjeta y note el cambio en la frecuencia de intermitencia del LED.
37. Ahora reemplace el valor asignado por 4 veces su valor original, ensamble, y descargue el programa a la tarjeta. Note el cambio en la frecuencia de intermitencia del LED.
38. Desconecte las tarjetas de la PC. Recuerde desconectarlas sujetando los conectores y no las tarjetas.