Práctica de Laboratorio #4
Puertos Paralelos en C

Objetivo

Que el estudiante aprenda el procedimiento necesario para controlar los puertos paralelos de una tarjeta de evaluación del microcontrolador LPC1115 utilizando el IDE Keil μVision.

Teoría

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 bits configurados como salidas, y estos estados aparecerán en los pines asociados. Los bits configurados como entradas tienen el estado lógico de los pines asociados.

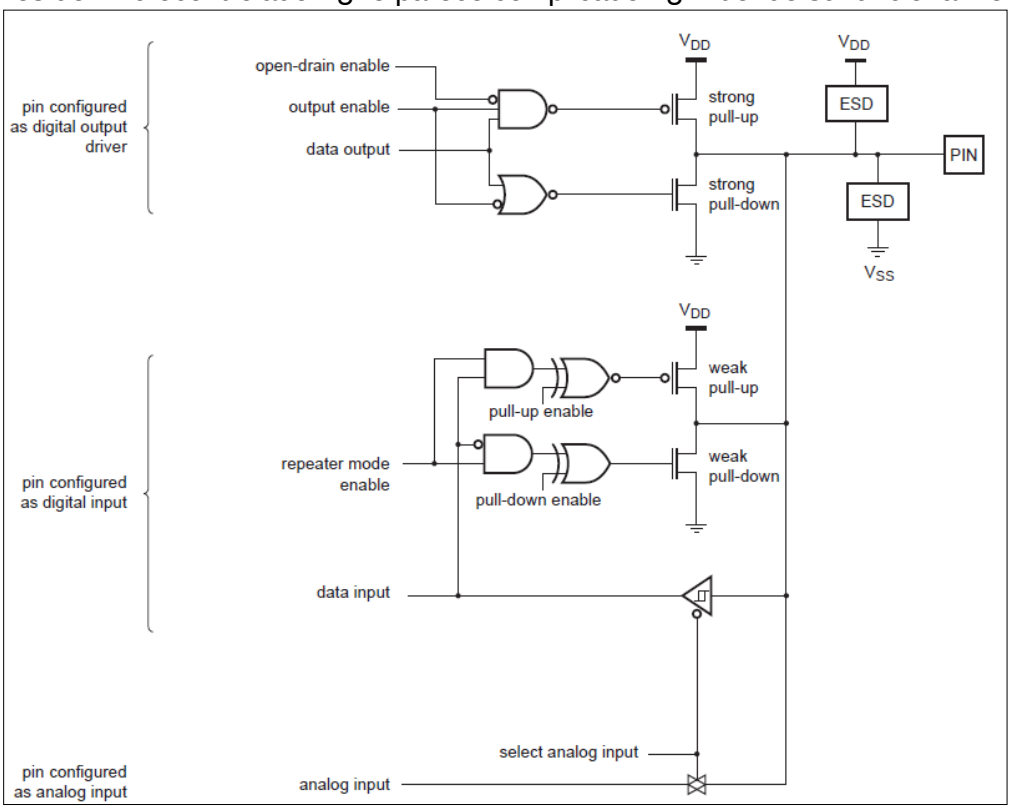
Cada puerto tiene también un registro de dirección de datos (Data Direction Register o **`GPIOnDIR`**), que se utiliza para configurar los bits del puerto como entradas o salidas individualmente (**`GPIO2DIR`** por ejemplo). Si se desea configurar un bit del puerto como salida, es necesario escribir un 1 en el bit correspondiente en el registro **`GPIOnDIR`**. Para entradas se utiliza 0.

Cada pin de cada puerto tiene también un registro de configuración llamado **`IOCON`** (por ejemplo **`IOCON_PIO2_11`**) que controla lo siguiente:

- Su función (GPIO o función de su periférico asociado).
- Su modo de entrada (puede habilitar un Pull-Up/Pull-Down, lo cual ahorra al usuario un resistor en el exterior del circuito integrado).
- Histéresis del estado lógico reconocido en la entrada.
- Modo de drenador abierto en la salida.
- Varios modos de operación para los pines con funcionalidad I^2C.
- Los pines utilizados como entradas del ADC tienen un bit en este registro para habilitar la entrada analógica.

Para más información, puede consultar el manual de usuario del microcontrolador.

Tómese unos minutos para revisar el siguiente diagrama simplificado del hardware de los pines del microcontrolador. ¿Le parece complicado? ¿Entiende su funcionamiento?

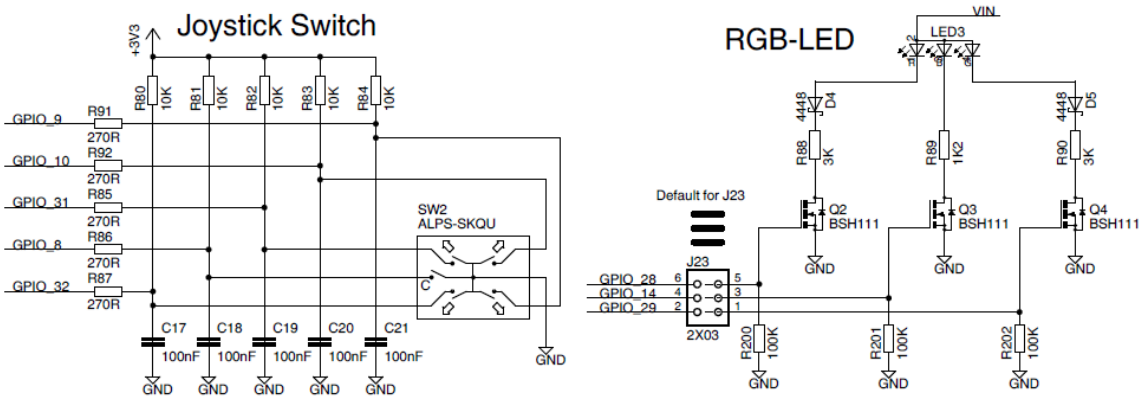


Hardware de los pines del microcontrolador LPC1114/15

En la tarjeta LPCXpresso Base Board, la mayoría de los pines del microcontrolador están conectados a un periférico. Algunos de estos periféricos son controlados digitalmente sin protocolos especiales. En esta práctica, usted utilizará los siguientes:

- Joystick discreto equivalente a 5 interruptores momentáneos.
- LED tricolor (RGB) diseñado para generar múltiples colores con PWM.

A continuación tenemos parte del diagrama esquemático de la tarjeta LPCXpresso Base Board, en donde aparecen estos periféricos. Los nodos llamados GPIO_XX corresponden a pines de puertos paralelos indicados más adelante.



Materiales y Equipo **(Proporcionados por el Laboratorio)**

- Tarjeta LPCXpresso Base Board con una tarjeta LPCXpresso LPC1115.
- PC con Windows 7 y Keil μVision 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.

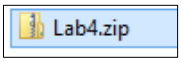
1. Asegúrese de conocer el hardware con el que trabajará. En particular verifique lo siguiente:

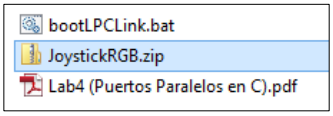
    a. Mantenga la tarjeta alejada de piezas metálicas como joyas y llaves.
    b. Mantenga la tarjeta alejada de piezas metálicas como joyas y llaves.
    c. Note que en adelante nos referiremos a dos tarjetas:
        i. La tarjeta LPCXpresso Base Board, que contiene periféricos.
        ii. La tarjeta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

Tarjeta LPCXpresso LPC1115, de Embedded Artists

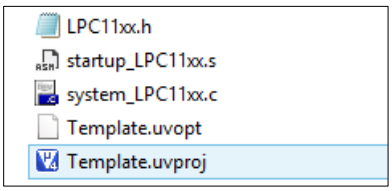
    d. La tarjeta LPCXpresso Base Board es alimentada por el conector USB Mini-B llamado X1 (Cerca del conector Ethernet RJ45), sin interruptor de encendido. No la conecte a la PC todavía.
    e. La tarjeta LPCXpresso LPC1115 es programada por su conector USB Mini-B llamado J3 (El único). No la conecte a 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 las carpetas **`Lab4.zip`** y **`JoystickRGB.zip`** donde prefiera.





3. Abra el proyecto incluido con este documento (**`Template.uvproj`**).



4. Este es el momento en que usted agregará un poco de código al proyecto. Su aplicación debe encender el LED RGB parcialmente cuando se cierren los interruptores del Joystick, de acuerdo a la siguiente tabla:

Joystick	Pin	Color del LED	Pin
Sentido 4	**PIO3_4**	Rojo	**PIO1_9**
Sentido 3	**PIO2_3**	Verde	**PIO1_10**
Sentido 2	**PIO2_2**	Azul	**PIO1_2**
Sentido 1	**PIO2_1**	Verde y Azul	Ver arriba
Presionar	**PIO2_0**	Rojo, Verde, Azul	Ver arriba

Sí, cuando el usuario mueva el Joystick en el sentido indicado, el LED debe iluminarse con los colores indicados. Si el Joystick está en reposo, el LED debe permanecer apagado.

5. Para lograr esto, es necesario configurar el módulo GPIO y los pines específicos que utilizará. Todo esto puede suceder al principio de su función principal. No se preocupe, los siguientes pasos le dirán qué hacer.

6. Suministre la señal de reloj al módulo GPIO. Esto es necesario para habilitar el módulo. Si se intenta utilizar un módulo al que no se le ha suministrado una señal de reloj, ocurrirá un problema de hardware (hardware fault), lo cual debemos evitar.

   Para suministrar la señal de reloj al módulo GPIO, conviene entender cómo funciona la arquitectura de bus del microcontrolador LPC1114/15, llamada AMBA (Advanced Microcontroller Bus Architecture). A continuación tenemos parte del diagrama de bloques:

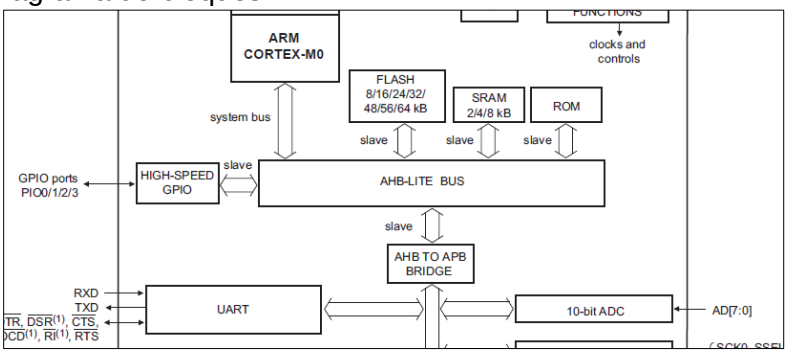


Diagrama de bloques parcial del microcontrolador LPC1115

   El módulo GPIO está conectado al microcontrolador por medio de un bus llamado AHB (Advanced High-performance Bus). De hecho, GPIO es el único módulo que utiliza este bus de alta velocidad. El resto de los módulos utilizan un bus llamado APB (Advanced Peripheral Bus), de menor velocidad.

   Para proporcionar la señal de reloj a cualquier módulo, es necesario habilitar un bit específico en el registro **SYSAHBCLKCTRL**. Cada módulo tiene un bit reservado en este registro para habilitar su señal de reloj. Puede leer más al respecto en el manual de usuario del microcontrolador, Capítulo 3 (System Configuration).

Ya que es necesario escribir únicamente en un bit específico, usted debe prestar especial atención al código que escriba para implementar esta escritura. Dos cosas pueden salir mal:

- Podría no habilitar el módulo GPIO, en cuyo caso su programa dejará de funcionar correctamente al intentar utilizar los puertos paralelos.
- Podría habilitar otros módulos por error. En este caso, podría modificar la funcionalidad esperada de su microcontrolador, y seguramente consumirá más energía que la necesaria por tener otro módulo funcionando.

Como puede ver a continuación, el bit que nos interesa es el número 6.

Bit	Symbol	Value	Description	Reset value
4	FLASHARRAY		Enables clock for flash array access.	1
		0	Disabled	
		1	Enabled	
5	I2C		Enables clock for I2C.	0
		0	Disable	
		1	Enable	
6	GPIO		Enables clock for GPIO.	1
		0	Disable	
		1	Enable	
7	CT16B0		Enables clock for 16-bit counter/timer 0.	0
		0	Disable	
		1	Enable	

Algunos bits del registro `SYSAHBCLKCTRL`

Escriba un “1” al bit 6 (únicamente al bit 6) del registro **`SYSAHBCLKCTRL`**. Puede hacerlo de la siguiente manera. Observe el código y pregunte si tiene dudas. Explore las librerías en Keil. Lea el manual. Haga un intento por entender el código.

```
//Enable AHB clock to the GPIO domain
LPC_SYSCON->SYSAHBCLKCTRL |= (1<<6);
```

7. Configure los pines del Joystick como entradas, y los pines del LED como salidas. Esto lo puede hacer escribiendo “1” a los bits que desea configurar como salidas, y “0” a los que desea configurar como entradas, en el registro llamado **`LPC_GPIOx->DIR`**, donde **`x`** es el número de puerto. Por ejemplo, para volver salida el pin 7 del puerto 0 y volver entrada el pin 2 del puerto 3, puede utilizar las siguientes líneas:

```
LPC_GPIO0->DIR |= MASK(7);
LPC_GPIO3->DIR &= ~MASK(2);
```

No use esas líneas textualmente, ya que esos pines no son utilizados por la aplicación actual. Podría dañar la tarjeta Base Board o el microcontrolador si configura puertos sin cuidado.

Revise la definición de la función **`MASK()`**. En su reporte, explique qué hace esa función, qué tipo de función es, y qué diferencia hay entre utilizar esa función y lo que aparece al lado derecho de la asignación del paso anterior. Si no incluye estas tres explicaciones en su reporte, entenderemos que usted renuncia a obtener puntos por elaborar un reporte en esta práctica.

8. Es muy común que en los microcontroladores, los pines tengan múltiples funciones. Esto se hace para que el microcontrolador sea versátil, y para ahorrar pines, ya que rara vez una aplicación hace uso de todos los módulos de un microcontrolador. Normalmente, la función por default de los pines es la de puerto paralelo (GPIO).

   Como cosa rara, el pin **`PIO1_2`** del microcontrolador LPC1115 **no** tiene como default su función de puerto paralelo. Configúrelo como puerto paralelo utilizando el registro que selecciona la funcionalidad de este pin (**`IOCON`** para ese pin).

```
LPC_IOCON->R_PIO1_2 = (LPC_IOCON->R_PIO1_2 & ~0x7) | 0x1;
```

   Esa línea de código selecciona la función GPIO en ese pin. Puede leer más sobre este registro en el manual de usuario.

IOCON_R_PIO1_2

Table 88. IOCON_R_PIO1_2 register (IOCON_R_PIO1_2, address 0x4004 4080) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function. All other values are reserved.	000
		0x0	Selects function R. This function is reserved. Select one of the alternate functions below.	
		0x1	Selects function PIO1_2.	
		0x2	Selects function AD3.	
		0x3	Selects function CT32B1_MAT1.	

Los 3 bits menos significativos del registro `LPC_IOCON->R_PIO1_2` configuran la función del pin

9. Ahora estamos listos para escribir el ciclo infinito. Para que su código sea legible, conviene utilizar símbolos significativos para el hardware utilizado. Utilice directivas de preprocesador para este fin, por *ejemplo*:

```
#define  Puerto2  LPC_GPIO2->DATA
#define  LEDcito  7
#define  Switchecito 3
...
  if(Puerto2 & MASK(Switchecito)) //Leemos PIO2_3
    Puerto2 |= (MASK(LEDcito));  // Escribimos 1 en PIO2_7
    ...
    Puerto2 &= ~(MASK(LEDcito));  // Escribimos 0 en PIO2_7
```

Es más: Puede implementar funciones para que su código sea aún más amigable. Le recomendamos producir un programa con la siguiente forma:

```
while(1){
  if(arriba())
    encender(Rojo);
  else if(abajo())
    encender(Verde);
  else if(derecha())
    encender(Azul);
  else if(izquierda())
    encender(VerdeAzul);
  else if(boton())
    encender(RGB);
  else
    apagar();
}
```

Esto reduce su tarea a implementar las constantes y funciones allí utilizadas.

10. Compile y descargue su programa. Para esto, recuerde que debe conectar ambas tarjetas a la PC, con cuidado. Luego debe ejecutar el archivo **`bootLPCLink.bat`** proporcionado, para arrancar el driver LPC-Link.

    Para su conveniencia, hemos incluido en el proyecto de Keil el comando para descargar su archivo ejecutable a la tarjeta LPCXpresso. No hace falta agregarlo.

11. Si su programa tiene algún problema, utilice sus habilidades de programador experimentado para depurarlo. Recuerde que siempre existe la posibilidad de que el problema provenga del autor del código. Entiéndase: Usted.