

Manejo de Puertos en el microcontrolador LPC43XX con la biblioteca LPCOpen

Para la realización de todos los trabajos prácticos de este curso, se utilizará la librería LPCOpen para el acceso a los periféricos específicos del LPC4337.

Esta biblioteca, se incluye en el Firmware de la CIAA, en:
`...CIAA\Firmware\externals\drivers\cortexM4\lpc43xx`

Es destacable, que para emplear un procesador diferente, se debe cambiar esta biblioteca, por ejemplo, para trabajar con la CIAA.FSL (CIAA Freescale), es necesario cargar los drivers que se encuentran en:
`...CIAA\Firmware\externals\drivers\cortexM4\k60_120`.

1. Puertos de Salida: LEDS

Para el manejo de puertos de entrada salida de uso general (GPIO), el archivo `gpio_18xx_43xx.c`, contiene todos los drivers para manejo del mismo. Ver documentación de LPCOpen.

Para configurar un GPIO, lo primero es llamar a la función `Chip_GPIO_Init(LPC_GPIO_T * pGPIO)`, a la hay que pasarle como parámetro la dirección base del periférico GPIO definida ya en `chip_lpc_43xx.h` como `LPC_GPIO_PORT`.

Luego hay que configurar la System Control Unit (SCU), para indicarle las características eléctricas de cada pin empleado y remapearlos como puertos GPIO. Hay que recordar que en este procesador, se puede elegir entre varias funciones disponibles para cada pin (ver Tabla 189 en la página 397 del User Manual):

```
Chip_SCU_PinMux(2,0,MD_PUP,FUNC4); /* mapea P2_0 en GPIO5[0], LED0R y habilita el
pull up */
Chip_SCU_PinMux(2,1,MD_PUP,FUNC4); /* mapea P2_1 en GPIO5[1], LED0G y habilita el
pull up */
Chip_SCU_PinMux(2,2,MD_PUP,FUNC4); /* mapea P2_2 en GPIO5[2], LED0B y habilita el
pull up */
Chip_SCU_PinMux(2,10,MD_PUP,FUNC0); /* remapea P2_10 en GPIO0[14], LED1 y habilita
el pull up */
Chip_SCU_PinMux(2,11,MD_PUP,FUNC0); /* remapea P2_11 en GPIO1[11], LED2 y habilita
el pull up */
Chip_SCU_PinMux(2,12,MD_PUP,FUNC0); /* remapea P2_12 en GPIO1[12], LED3 y habilita
el pull up */
```

A continuación, se debe seleccionar el modo (entrada o salida) de cada pin con la función:

```
Chip_GPIO_SetDir(LPC_GPIO_T * pGPIO, uint8_t portNum, uint32_t portValue, uint8_t out);
```

Para setear y resetear los pines, existen numerosas funciones, entre ellas:

```
Chip_GPIO_ClearValue();
Chip_GPIO_SetValue();
```

```

Chip_GPIO_SetPinOutLow();
Chip_GPIO_SetPinOutHigh();
Chip_GPIO_SetPortOutHigh();
Chip_GPIO_SetPinToggle();
Chip_GPIO_SetPortToggle();

```

A las que siempre hay que pasarles como parámetro la dirección base del periférico GPIO (LPC_GPIO_PORT), el número de puerto y el bit a modificar. En caso de las funciones que hacen referencia a un solo GPIO (contienen la palabra “Pin”) se le indica el número de bit del puerto que se desea modificar.

En caso de funciones que acceden a todo el puerto (Port), se le pasa una máscara del tipo uint32_T con los bits a modificar en 1.

2. Puertos de Entrada: Pulsadores

Para trabajar con los puertos de salida disponibles en la placa se deben configurar las entradas digitales correspondientes a las teclas.

```

Chip_SCU_PinMux(1,0,MD_PUP|MD_EZI|MD_ZI,FUNC0);/* mapea P1_0 en GPIO 0[4], SW1 */
Chip_SCU_PinMux(1,0,MD_PUP|MD_EZI|MD_ZI,FUNC0);/* mapea P1_0 en GPIO 0[4], SW1 */
Chip_SCU_PinMux(1,1,MD_PUP|MD_EZI|MD_ZI,FUNC0);/* mapea P1_1 en GPIO 0[8], SW2 */
Chip_SCU_PinMux(1,2,MD_PUP|MD_EZI|MD_ZI,FUNC0);/* mapea P1_2 en GPIO 0[9], SW3 */
Chip_SCU_PinMux(1,6,MD_PUP|MD_EZI|MD_ZI,FUNC0);/* mapea P1_6 en GPIO 1[9], SW4 */

```

Además, habilita para cada pin el buffer de entrada y deshabilita el filtro de glitch (ver figura 41 del User Manual). Es interesante como se realiza esta acción en el driver de GPIO del Firmware de la CIAA: “ciaaDriverDio.c”, que se encuentra en

Luego setear estos pines como entrada (Chip_GPIO_SetDir), y para leer, se pueden utilizar las funciones:

```

Chip_GPIO_ReadValue()
Chip_GPIO_ReadPortBit()

```