Diseño Basado en Microprocesadores
Práctica 13
Comunicación I2C en el LPC40xx

Índice

---

1. Objetivos

En esta práctica usaremos unos de los interfaces I2C del microcontrolador LPC40xx para comunicarnos con un sensor de temperatura con salida digital.

2. Biblioteca de funciones para manejar los interfaces I2C

Para manejar los interfaces I2C del LPC40xx usaremos una biblioteca de funciones que nos permitirá realizar las operaciones básicas: inicialización, creación de condiciones de START y STOP y el envío y recepción de bytes. La biblioteca está compuesta por el fichero i2c_lpc40xx.c y su correspondiente cabecera i2c_lpc40xx.h. Para que sea más fácil comprender las funciones, se ha sacrificado la eficiencia en favor de la sencillez, así que no usarán interrupciones ni DMA sino que operarán mediante el sondeo de los bits de estado de los interfaces. A continuación, se describen las funciones disponibles.

-
```
void  i2c_inicializar(LPC_I2C_TypeDef  *i2c_regs,
                      uint32_t  frecuencia_scl,
                      LPC_GPIO_TypeDef  *puerto_sda,
                      uint32_t  mascara_pin_sda,
                      LPC_GPIO_TypeDef  *puerto_scl,
                      uint32_t  mascara_pin_scl)
```

La función i2c_inicializar servirá para inicializar un interfaz I2C del microcontrolador. El microcontrolador quedará configurado como maestro. El argumento i2c_regs es un puntero

al bloque de registros del interfaz I2C que queremos inicializar. El parámetro frecuencia_scl es la frecuencia de reloj SCL deseada. Los parámetros puerto_sda, mascara_pin_sda, puerto_scl y mascara_pin_cls indican qué pines del microcontrolador deseamos usar como señales SDA y SCL.

La razón de indicar puertos y pines mediante punteros a los registros GPIO y máscaras de pin, respectivamente, en lugar de números de puerto y números de pin, es conseguir que la forma de especificarlos sea igual a la usada en los módulos gpio_lpc40xx e iocon_lpc40xx. Esto permite emplear los mismos símbolos de tipo PUERTOx y PINx definidos en gpio_lpc40xx.h que ya usamos en funciones como gpio_ajustar_dir o gpio_leer_pin.

- `void i2c_start(LPC_I2C_TypeDef *i2c_regs)`

  La función i2c_start servirá para crear la condición de START en el bus I2C conectado al interfaz indicado por el argumento i2c_regs.

- `void i2c_stop(LPC_I2C_TypeDef *i2c_regs)`

  La función i2c_stop servirá para crear la condición de STOP en el bus I2C conectado al interfaz indicado por el argumento i2c_regs.

- `bool_t i2c_transmitir_byte(LPC_I2C_TypeDef *i2c_regs, uint8_t byte)`

  La función i2c_transmitir_byte enviará a través del interfaz indicado por el argumento i2c_regs el dato de ocho bits indicado por byte. La función retorna TRUE si el esclavo reconoció el byte y FALSE si no lo reconoció.

- `uint8_t i2c0_recibir_byte(LPC_I2C_TypeDef *i2c_regs, bool_t ack)`

La función i2c_recibir_byte servirá para recibir un byte desde un esclavo. Si el parámetro ack es TRUE el microcontrolador realizará el reconocimiento del byte recibido, mientras que si es FALSE no reconocerá el byte recibido. En cualquier caso, la función retorna el byte recibido.

3. Ejercicios

3.1. Ejercicio 1

En la tarjeta Embedded Artists LPC4088 Developer's Kit hay un sensor de temperatura integrado con interfaz digital I2C modelo LM75B que está conectado al interfaz I2C 0 del LPC4088. La figura 1 indica la ubicación del LM75B. Estudia el datasheet del LM75B para comprender cómo funciona.

Los pines del microcontrolador que están conectados a las señales SDA y SCL del sensor son el P0[27]/I2C0_SDA/USB_SDA y el P0[28]/I2C0_SCL/USB_SCL. Las funciones que tenemos que seleccionar en dichos pines son I2C0_SDA y I2C0_SCL, respectivamente.

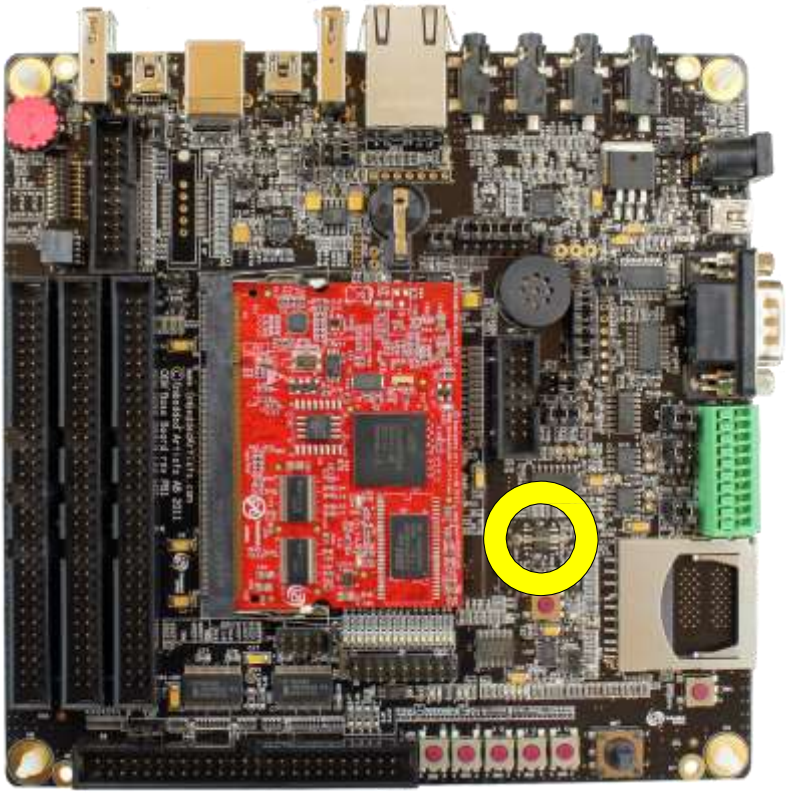

Figura 1: Ubicación del LM75B en la tarjeta LPC4088 Developer's Kit.

Como has visto en su datasheet, el sensor LM75B tiene tres pines que permiten configurar tres de sus bits de dirección I2C. En nuestra tarjeta de desarrollo, estos tres se encuentran conectados a masa y el LM75B queda así en la dirección I2C 0x48.

Para facilitar el acceso al sensor crearemos un conjunto específico de funciones. Estas funciones se basarán a su vez en las funciones genéricas de comunicación I2C que hemos comentado previamente. Las funciones que desarrollaremos serán las siguientes:

-
```
void lm75b_inicializar (LPC_I2C_TypeDef  *i2c_regs,
                        uint32_t  frecuencia_scl,
                        LPC_GPIO_TypeDef  *puerto_sda,
                        uint32_t  mascara_pin_sda,
                        LPC_GPIO_TypeDef  *puerto_scl,
                        uint32_t  mascara_pin_scl,
                        uint8_t  direccion_i2c_lm75b)
```

La función lm75b_inicializar servirá para inicializar el interfaz que queremos usar para comunicarnos con el sensor de temperatura LM75. El significado de los seis primeros parámetros es el mismo que en el caso de la función lm75b_inicializar e indican el interfaz I2C, la frecuencia de reloj SCL y los pines SDA y SCL del microcontrolador que queremos usar para comunicarnos con el LM75B. El parámetro direccion_i2c_lm75b es la dirección I2C del LM75B.

Es cierto que las conexiones y la dirección de esclavo I2C del LM75B están fijadas por el diseño de la tarjeta Embedded Artists LPC4088 Developer's Kit, pero tener la libertad especificar estos parámetros nos permitirá usar las funciones en otro sistema distinto donde las conexiones y/o dirección del sensor sean diferentes.

- `void lm75b_escribir_registro (uint8_t registro_a_escribir, uint16_t dato_a_escribir)`

La función lm75b_escribir_registro servirá para escribir en un registro interno del LM75B. El parámetro registro_a_escribir indica el registro del LM75B en el que queremos escribir. El parámetro dato_a_escribir es el dato que queremos escribir en el registro.

- `uint16_t lm75b_leer_registro (uint8_t registro_a_leer);`

La función lm75b_leer_registro servirá para leer un registro interno del LM75B. El parámetro registro_a_leer indica el registro del LM75B que queremos leer. La función retorna el valor leído del LM75B.

- `float lm75b_leer_temperatura(void)`

La función lm75b_leer_temperatura servirá para leer el registro TEMP del LM75B y retornar la temperatura en grados centígrados correspondiente.

Usa las funciones en un programa que muestre en la pantalla LCD la temperatura captada por el sensor una vez por segundo.

3.2. Ejercicio 2

Amplía el ejercicio 1 para que la temperatura también se envíe a través de la UART 0 del microcontrolador y visualízala en un programa de terminal en el PC.