SISTEMAS EMPOTRADOS
3º Grado en Ingeniería Informática

PRÁCTICA 3

MANEJO DE GPIO Y TIMERS II

3.1 Introducción

En esta práctica vamos a poner en práctica los conocimientos adquiridos en la práctica anterior, utilizando dos pines del puerto y dos temporizadores.

Queremos generar dos señales digitales por un puerto GPIO con distinta anchura a baja que a alta. La primera tendrá una duración a baja de 500 µs y a alta 700 µs; la segunda tendrá una duración a baja de 200 µs y a alta de 300 µs. Se comprobará con el osciloscopio que estas señales se generan correctamente.

Los pines escogidos para visualizar las dos señales son P4.24 y P4.25 que se corresponden con los pines 127 y 124 respectivamente , como se puede observar en el documento donde se incluye el esquemático de la placa MCB2300 (disponible en la plataforma *moodle*).

Para completar el estudio de los puertos GPIO debemos mirar como referencia los capítulos 8, 9 y 10 del manual de usuario LPC23xx de NXP *Semiconductors*. Para estudiar la configuración de los temporizadores el capítulo 23 de dicho manual.

Para el estudio teórico utilizamos el apartado 3.8.1 del libro de referencia de la asignatura "The Insider's guide to the NXP LPC2300/2400 Based Microcontrollers. An engineer's introduction to the LPC2300 & LPC2400 series". **http://www.hitex.co.uk.** Además de los apartados 4.2 y 4.3 para el estudio de los GPIO y timers.

3.2 Objetivos

Los objetivos marcados en esta práctica son los siguientes:

- Que el alumnado estudie y aprenda a configurar los puertos básicos de entrada/salida (GPIO) del LPC2378.

- Estudio y programación de los temporizadores (*timers*) de que dispone dicho microcontrolador.

- Diseño de un programa de aplicación, su simulación y su comprobación real con el osciloscopio.

3.3 Material utilizado

El material necesario para la realización de esta práctica es el siguiente:

- Ordenador personal bajo *Windows* con el *software Keil µVision* 5 instalado y el *pack* correspondiente a nuestra placa.

- Placa de desarrollo *Keil* MCB2300.

- Adaptador USB–JTAG de la familia ULINK para depurar programas.

- Dos cables USB A–B conectados a dos puertos USB disponibles del ordenador.

- Osciloscopio.

3.4 Desarrollo de la práctica

Crear un nuevo proyecto (practica3) en una carpeta personalizada para cada práctica. Copiar en esa carpeta los ficheros:

- LPC2300.s
- retarget.c (para configurar el microcontrolador, entradas/salidas, estándar de C, stdio.h, etc.)
- serial.c

Vamos a crear los ficheros:

- Practica3.c.
- Practica3.h
- HAL.c
- misTipos.h

El programa principal (Practica3.c) de la aplicación quedaría de la forma siguiente:

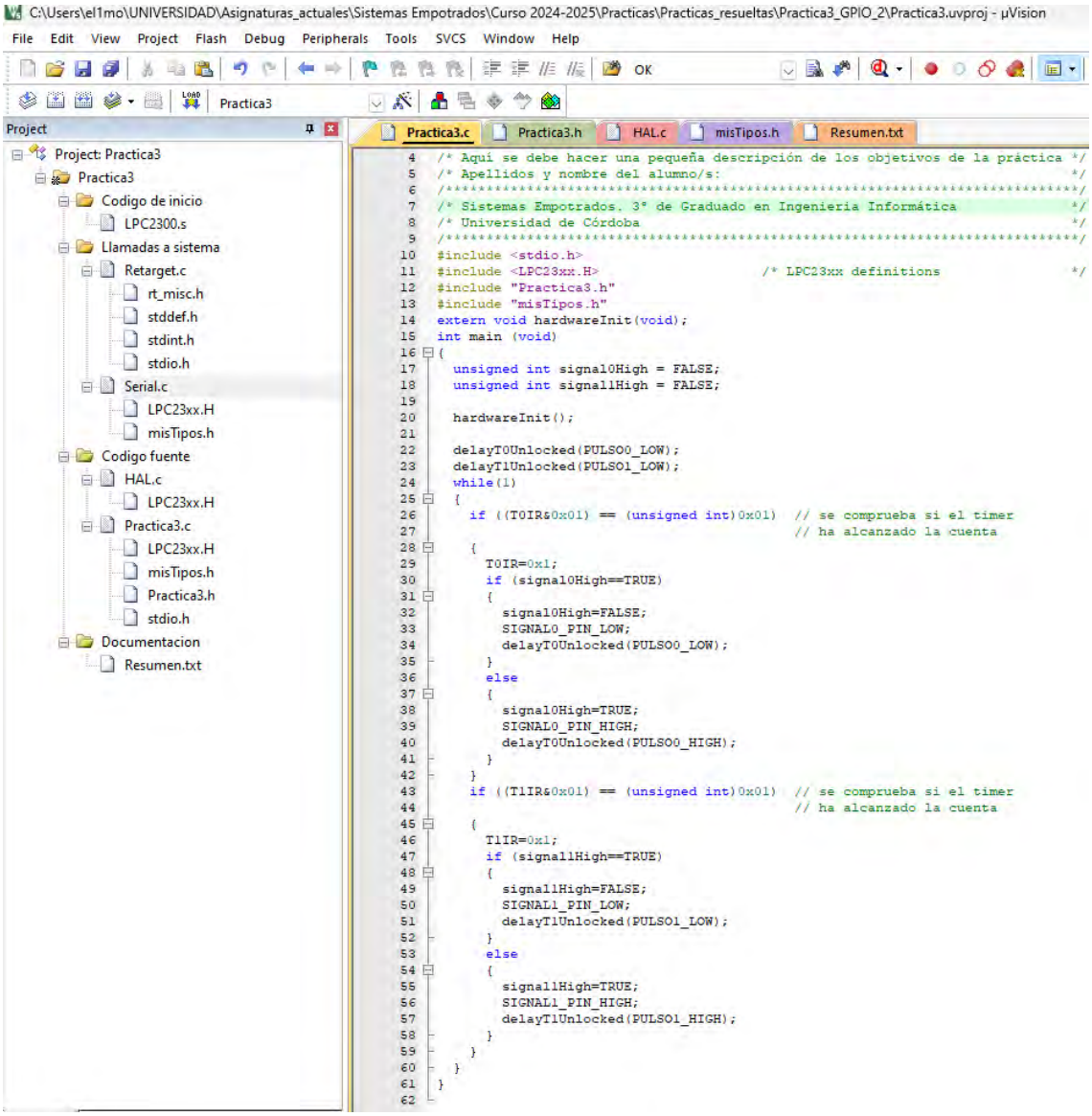

Figura 3-1 Programa principal de la aplicación.

El fichero Practica3.h para configurar los dos temporizadores tendría que ser algo como el de la figura 3-2. Hay que observar la configuración de los registros FIO4SET3 y FIO4CLR3 en el manual del fabricante. Así mismo también debemos estudiar la configuración de ambos temporizadores T0 y T1.

```c
/*****************************************************************************/
/* Practica3.h: definiciones para la práctica 3 de timers y algunas funciones */
/* Sistemas Empotrados. Universidad de Córdoba                               */
/*****************************************************************************/
#define PULSO0_LOW  2                       /* duración del pulso 0 a nivel bajo */
#define PULSO0_HIGH 3                       /* duración del pulso 0 a nivel alto */
#define PULSO1_LOW  5                       /* duración del pulso 1 a nivel bajo */
#define PULSO1_HIGH 7                       /* duración del pulso 1 a nivel alto */
#define SIGNAL0_PIN_HIGH  FIO4SET3 = 0x01;  /* Pin señal 0 a alto P4.24          */
#define SIGNAL0_PIN_LOW   FIO4CLR3 = 0x01;  /* Pin señal 0 a bajo P4.24          */
#define SIGNAL1_PIN_HIGH  FIO4SET3 = 0x02;  /* Pin señal 1 a alto P4.25          */
#define SIGNAL1_PIN_LOW   FIO4CLR3 = 0x02;  /* Pin señal 1 a bajo P4.25          */

#include <LPC23xx.H>                    /* LPC23xx definitions                   */
/*****************************************************************************/
/* delayT0Unlocked: Esta función arranca el timer 0 y programa el registro match0 */
/*****************************************************************************/
void delayT0Unlocked(unsigned int delayInDecimaMiliseg)
{
  T0TCR = 0x02;                                             /* reset timer 0    */
  T0MR0 = delayInDecimaMiliseg * 12000000 / 10000;
  T0MCR = 0x07;                                             /* timer 0 on match */
  T0TCR = 0x01;      /* inicia timer 0 y para cuando se llegue al final de cuenta */
}
/*****************************************************************************/
/* delayT1Unlocked: Esta función arranca el timer 1 y programa el registro match1 */

void delayT1Unlocked(unsigned int delayInDecimaMiliseg)
{
  T1TCR = 0x02;                                             /* reset timer 1    */
  T1MR0 = delayInDecimaMiliseg * 12000000 / 10000;
  T1MCR = 0x07;                                             /* timer 1 on match */
  T1TCR = 0x01;      /* inicia timer 1 y para cuando se llegue al final de cuenta */
}
```

Figura 3-2 Fichero de configuración de los temporizadores.

En este caso, el fichero HAL.c de capa de abstracción del *hardware* para inicializar el *hardware* quedaría de la forma siguiente:

```c
/*****************************************************************************/
/* HAL.C: Funciones generales que acceden al hardware                        */
/* Capa de abstracción del hardware (Hardware Abstract Layer)                */
/* Sistemas Empotrados. Universidad de Córdoba                               */
/*****************************************************************************/
 #include <LPC23xx.H>                    /* LPC23xx definitions              */
/*****************************************************************************/
/* pinesSignalInit: Esta función configura los pines                         */
/*                  P4.24 y P4.25 como salida                                */
/*****************************************************************************/
void pinesSignalInit(void)
{
  PINSEL9 = 0x00000000;
  PINMODE9 = 0x000000000;
  FIO4DIR3  = 0x03;
}
/*****************************************************************************/
/* timer0Init: Esta función configura el timer 0 con los parámetros          */
/*             que no cambian durante la aplicación                          */
/*****************************************************************************/
void timer0Init(void)
{
  T0PR  = 0x00;      /* activa el preescalador a cero                        */
}
/*****************************************************************************/
/* timer1Init: Esta función configura el timer 1 con los parámetros          */
/*             que no cambian durante la aplicación                          */
/*****************************************************************************/
void timer1Init(void)
{
  PCONP |= (0 << 2); /* Habilito Timer 1, power y reloj                      */
  T1PR = 0x00;       /* activa el preescalador a cero                        */
}
/*****************************************************************************/
/* hardwareInit: Esta función se llama al comienzo del programa              */
/*               para inicializar el Hardware                                */
/*****************************************************************************/
void hardwareInit(void)
{
  pinesSignalInit();  /* Configura los pines del circuito                    */
  timer0Init();       /* Inicializa el timer 0                               */
  timer1Init();       /* Inicializa el timer 1                               */
 }
```

Figura 3-3 Fichero HALc para acceder al *hardware*. (*Hardware Abstract Layer*)

El fichero misTipos.h para definir los tipos de variables, se corresponde con el de la práctica anterior:

Practica3.c | Practica3.h | HAL.c | misTipos.h | Resumen.txt

```
/*****************************************************************************/
/* misTipos.h: definiciones de tipos de variables para prácticas con LPC2378  */
/* Sistemas Empotrados. Universidad de Córdoba                                */
/*****************************************************************************/

#ifndef __misTipos_H
#define __misTipos_H

/* Byte */
#define UINT8 unsigned char
#define INT8 char

/*16 bits */
#define UINT16 unsigned short int
#define INT16  short int

/*32 bits WORD para el LPC2378 */
#define UINT32 unsigned int
#define INT32  int

/* Tipos para control */
#define STATUS UINT32

/* Boolenas */
#define BOOL   UINT32
#define FALSE  (unsigned int)0x000000000
#define TRUE   (unsigned int)0x000000001

/* flags */
#define FLAG  BOOL
#define ESC 0x1B
#define OK 0x01
#endif

```

Figura 3-4 Fichero para la definición de tipos de variables.

Los periféricos que deberemos observar en esta práctica son el timer 0, el timer 1 y el GPIO4.

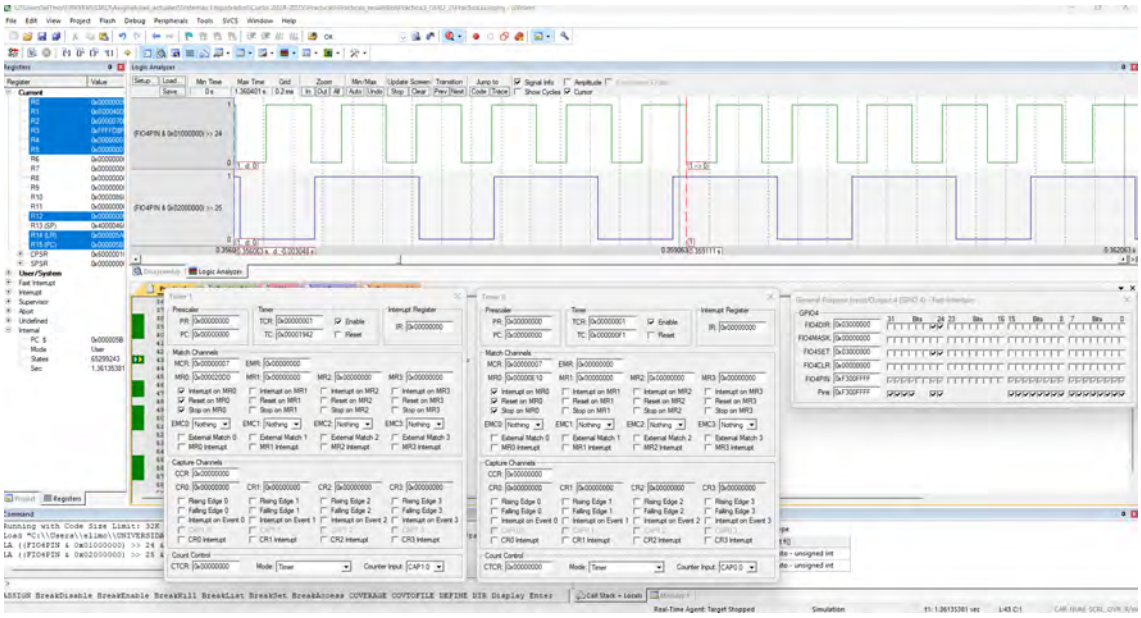

Figura 3-5 Vista del entorno de depuración.

Al igual que en la práctica anterior, después de haber compilado, enlazado, simulado y depurado el programa, comprobaremos con el analizador lógico y el osciloscopio, que ambas señales se generan correctamente, observando y midiendo sus características. Los pines son P4.24 (127) y P4.25 (124) que tendrán que localizarlos en la placa.

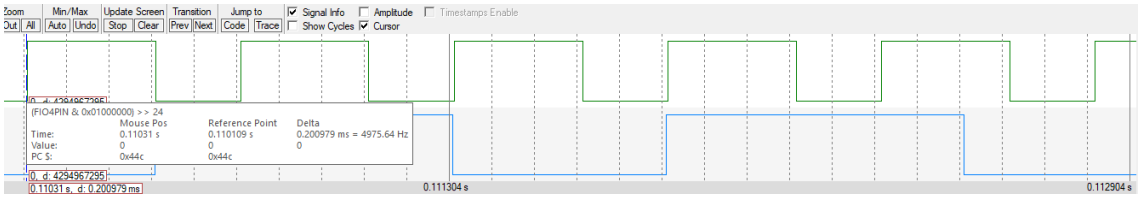

Figura 3-6 Resultado de la simulación con el analizador lógico virtual.