

SISTEMAS EMPOTRADOS

3º Grado en Ingeniería Informática

PRÁCTICA 1

INTRODUCCIÓN A LA HERRAMIENTA KEIL µVision 5

1.1. *Objetivos*

Los objetivos que se persiguen con la realización de esta práctica pueden resumirse en los siguientes puntos:

- Que el alumnado aprenda a instalar el *software Keil µVision 5* con el *pack* que se va a utilizar en la placa **Keil MCB2300**, y los ejemplos que proporciona el fabricante.
- Familiarizarse con el entorno de programación cargando en la placa un ejemplo de los proporcionados por el propio *software*.
- Que conozcan las características básicas de la placa de desarrollo MCB2300 y la forma de conectar la placa y el adaptador *ULINK* a un ordenador.
- Aprender los pasos básicos para crear un proyecto en *Keil µVision5* y simularlo en la herramienta *software*.

1.2. *Material utilizado*

El material, tanto *hardware* como *software*, que se va a utilizar en esta práctica y, como mínimo, en las restantes prácticas de la asignatura se enumera a continuación:

- Ordenador personal con sistema operativo *Windows*.
- Ficheros de instalación de la herramienta *software Keil µVision5*: mdk525.exe y el *pack* mdk79525.exe para nuestra placa, ambos disponibles en la plataforma *moodle*.
- Placa de desarrollo **Keil MCB2300** (Figura 1-1) que da soporte a la familia ARM LPC23xx de NXP (anteriormente Philips) y permite crear programas de prueba



para trabajar con arquitecturas más avanzadas. El *Keil MCB2300* se puede conectar al PC usando el puerto serie o la interfaz JTAG. Esta tarjeta consta de dos interfaces CAN y dos interfaces serie que hacen de ésta el punto de partida para futuros proyectos ARM.

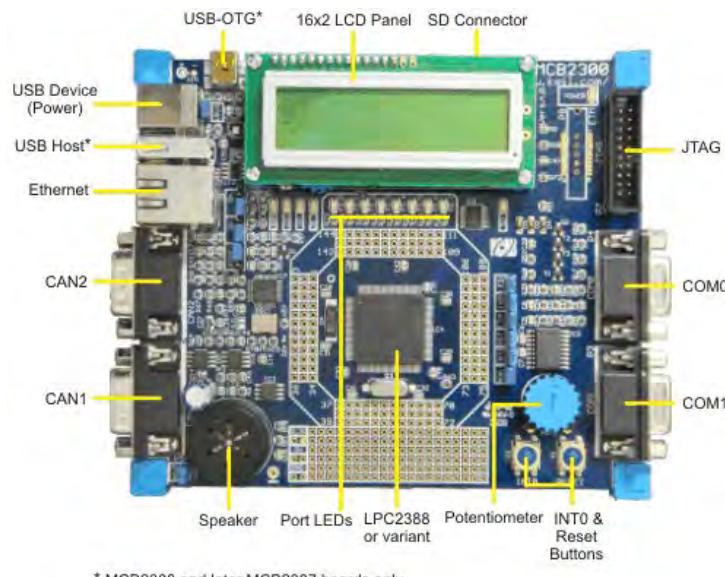


Figura 1-1: Placa de desarrollo MCB2300.

- Adaptador USB–JTAG de la familia ULINK™ de *Keil* (Figura 1-2). Permite conectar el puerto USB a la tarjeta de *hardware* de la computadora (por medio de JTAG o ACDS) para realizar la depuración de los programas mientras estos se están ejecutando. Este puede ser usado para: corrección de errores *on-chip* y para programar la memoria *flash*. Usando el *Keil uVision IDE/Debugger* con el adaptador *UNLINK*, es mucho más fácil crear, descargar, y probar aplicaciones integradas en *hardware* de tarjetas actuales.

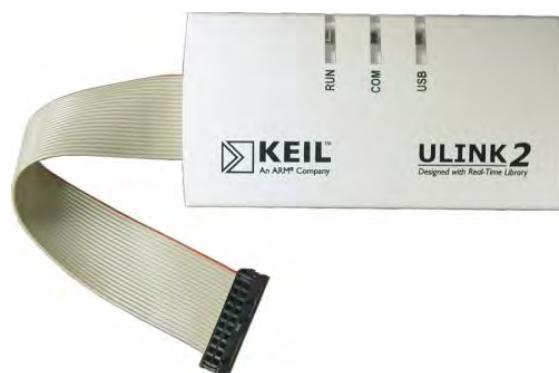


Figura 1-2: Adaptador ULINK2 de *Keil*.

- Dos cables USB A–B conectados a dos puertos USB disponibles del ordenador: uno para suministrar energía a la placa y el otro para cargar y depurar el programa conectado al adaptador ULINK2.



Figura 1-3: Cable USB tipo A a tipo B.

1.3. Desarrollo de la práctica

1.3.1. Instalación del software

Ejecutar la instalación del programa *Keil µVision 5* como administrador con el fichero mdk525.exe. Seguidamente instalar el *pack* autoejecutable para nuestra placa mdk79525.exe.

Con estas dos instalaciones tendremos configurado el *software* para el desarrollo de todas las prácticas a realizar en nuestra asignatura.

1.3.2. Ejercicio 1: Creación de un proyecto de ejemplo

Crear una nueva carpeta *practica1* en una carpeta personalizada para cada práctica.

Copiar en esa carpeta del ejemplo *Blinky* que está situado en la siguiente ruta:

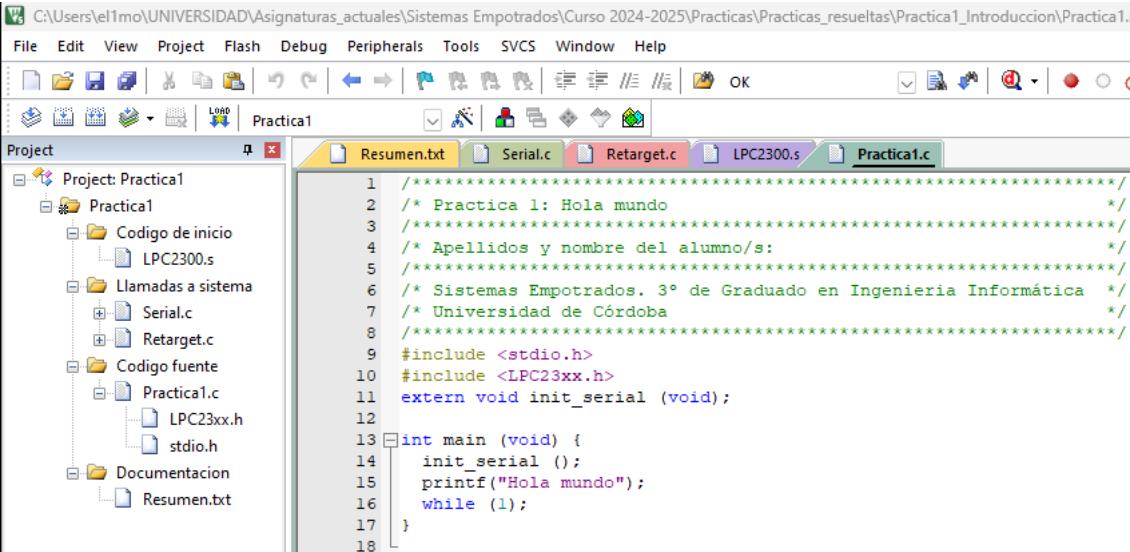
C:\Keil_v5\ARM\Boards\Keil\MCB2300\Blinky

los ficheros siguientes:

- *LPC2300.s* : donde se encuentran todas las características del microcontrolador de nuestra placa.
- *retarget.c*: para configurar el microcontrolador, entradas/salidas, estándar de C, stdio.h, etc.
- *serial.c* : para la configuración del puerto serie.



Crear un nuevo fichero fuente `practica1.c`. La imagen del entorno debe quedar como en la siguiente figura:



```

C:\Users\el1mo\UNIVERSIDAD\Asignaturas_actuales\Sistemas Empotrados\Curso 2024-2025\Practicas\Practicas_resueltas\Practica1_Introduccion\Practica1.l
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
OK
Project Practica1
  Project: Practica1
    Código de inicio
      LPC2300.s
    Llamadas a sistema
      Serial.c
      Retarget.c
    Código fuente
      Practica1.c
        LPC23xx.h
        stdio.h
    Documentación
      Resumen.txt
      Resumen.txt

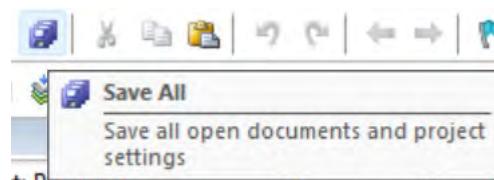
Resumen.txt Serial.c Retarget.c LPC2300.s Practica1.c

1  /*****
2  * Practica 1: Hola mundo
3  *****/
4  /* Apellidos y nombre del alumno/s:
5  *****/
6  /* Sistemas Empotrados. 3º de Graduado en Ingeniería Informática */
7  /* Universidad de Córdoba
8  *****/
9  #include <stdio.h>
10 #include <LPC23xx.h>
11 extern void init_serial (void);
12
13 int main (void) {
14     init_serial ();
15     printf("Hola mundo");
16     while (1);
17 }
18

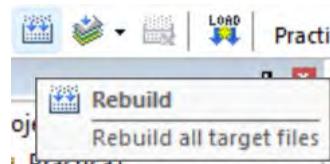
```

Figura 1-4: Programa principal.

Después de guardado,



construido el programa,



pasamos a modo depuración



Aquí podremos observar el entorno del depurador que es distinto del entorno del editor:

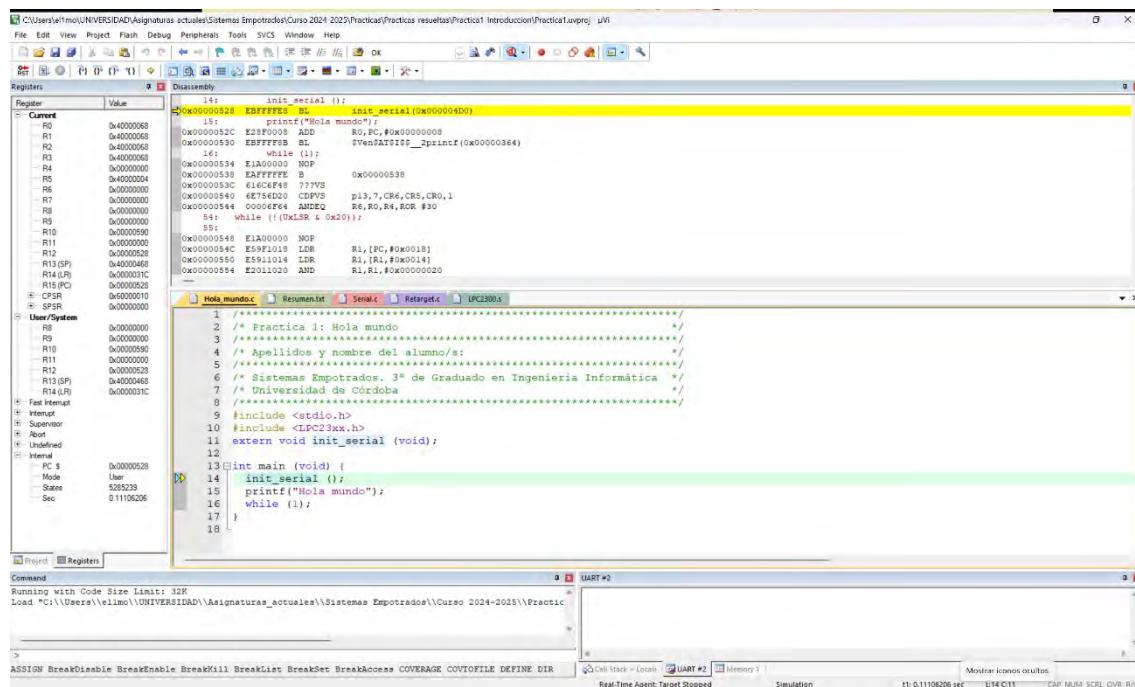


Figura 1-5: Entorno de depuración.

Para ver por el puerto serie (UART2) la frase “Hola mundo” debemos visualizarla a través del menú **View → Serial Windows → UART #2**

Podemos ejecutar nuestro pequeña aplicación con la tecla F5, con la herramienta rápida

o a través del menú **Debug → Run**

En la parte inferior a la izquierda nos saldrá lo siguiente:

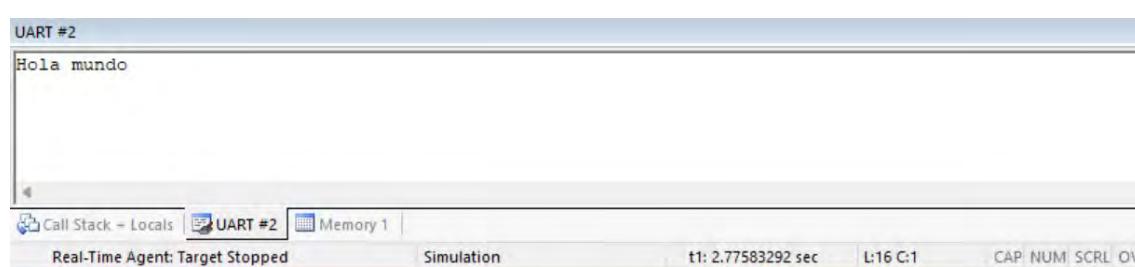


Figura 1-6: Ventana del resultado de la simulación en Keil μVision 5.

1.3.3. Ejercicio 2: Cargar un programa creado para utilizar el analizador lógico

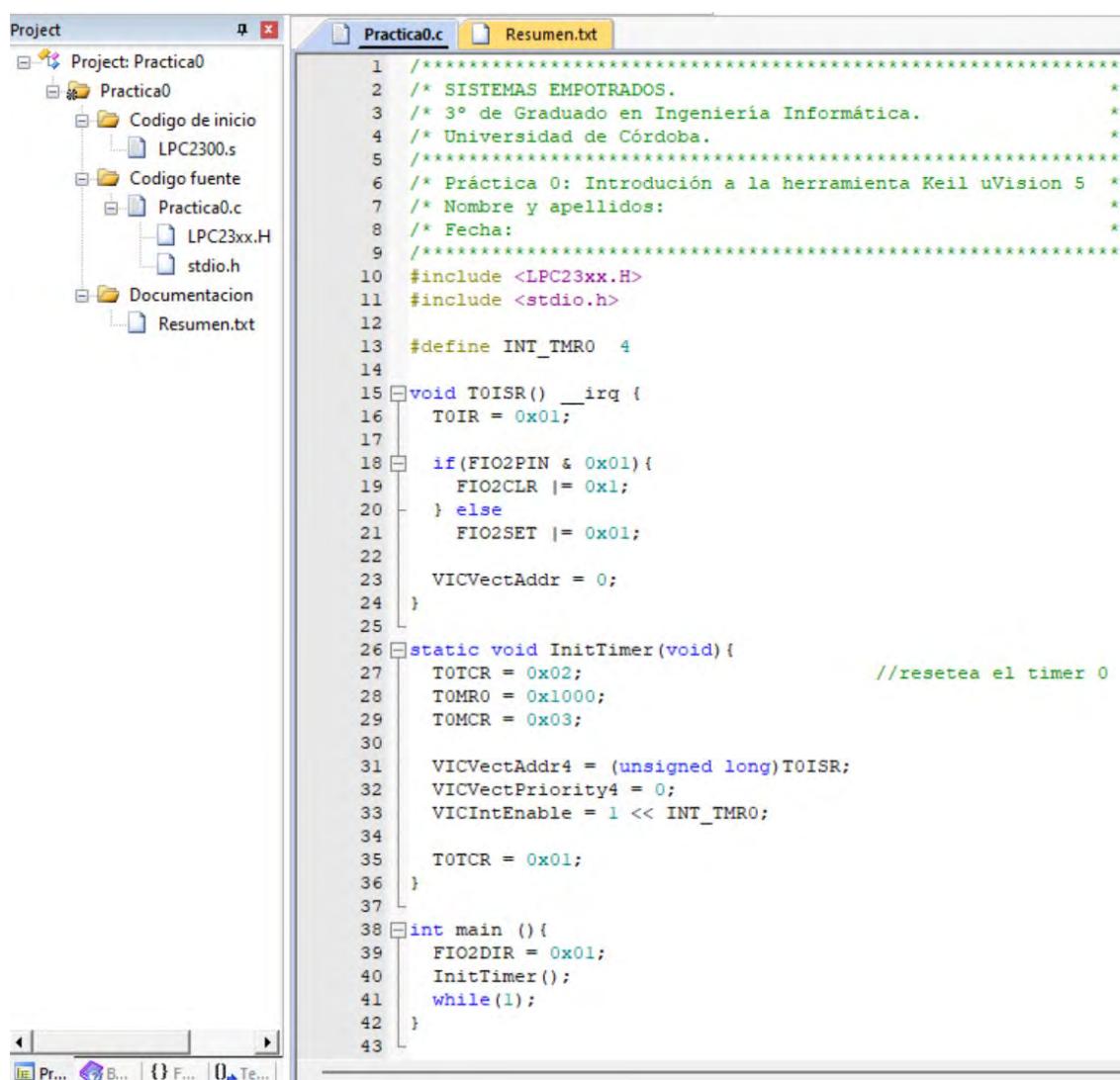


En este ejercicio vamos a crear un proyecto donde añadiremos un pequeño programa que se proporciona en la plataforma *moodle* que simplemente genera una señal cuadrada por el puerto P0.2 del microcontrolador LPC2378 de nuestra placa MCB2300.

Para ello crearemos una carpeta donde copiaremos los ficheros siguientes:

- **LPC2300.s** : donde se encuentran todas las características del microcontrolador de nuestra placa.
- **Practica0c**: fichero de ejemplo para la realización del ejercicio.

El entorno de este ejercicio debe quedar de la forma siguiente:



The screenshot shows the Keil uVision 5 IDE interface. On the left, the 'Project' window displays a project named 'Practica0' containing a folder 'Practica0' with subfolders 'Codigo de inicio' (containing 'LPC2300.s'), 'Codigo fuente' (containing 'Practica0.c' which is open in the editor, and 'LPC23xx.H' and 'stdio.h'), and 'Documentacion' (containing 'Resumen.txt'). The main window shows the source code for 'Practica0.c'. The code is a C program that generates a square wave on port P0.2. It includes comments in Spanish, header files, and defines a timer interrupt function 'TOISR' and a timer initialization function 'InitTimer'. The 'main' function initializes the port and calls 'InitTimer' before entering a loop.

```
1  /*************************************************************************/
2  /* SISTEMAS EMPOTRADOS. */
3  /* 3º de Graduado en Ingeniería Informática. */
4  /* Universidad de Córdoba. */
5  /*************************************************************************/
6  /* Práctica 0: Introducción a la herramienta Keil uVision 5 */
7  /* Nombre y apellidos: */
8  /* Fecha: */
9  /*************************************************************************/
10 #include <LPC23xx.H>
11 #include <stdio.h>
12
13 #define INT_TMRO 4
14
15 void TOISR() interrupt 1 {
16     T0IR = 0x01;
17
18     if(FIO2PIN & 0x01){
19         FIO2CLR |= 0x1;
20     } else
21         FIO2SET |= 0x01;
22
23     VICVectAddr = 0;
24 }
25
26 static void InitTimer(void) {
27     TOTCR = 0x02; //resetea el timer 0
28     TOMRO = 0x1000;
29     TOMCR = 0x03;
30
31     VICVectAddr4 = (unsigned long)TOISR;
32     VICVectPriority4 = 0;
33     VICIntEnable = 1 << INT_TMRO;
34
35     TOTCR = 0x01;
36 }
37
38 int main () {
39     FIO2DIR = 0x01;
40     InitTimer();
41     while(1);
42 }
43
```

Figura 1-7: Programa que genera una señal cuadrada por el puerto P0.2.

A continuación guardaremos, construiremos el programa y pasamos a modo depuración (*debugger*) con lo que el entorno nos queda de la forma siguiente:



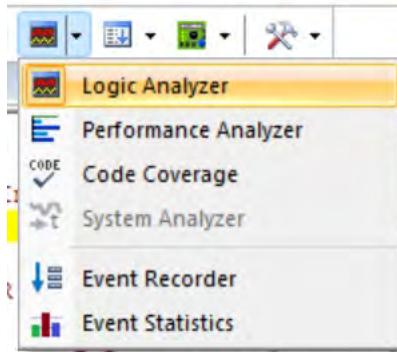
The screenshot shows the uVision IDE interface for a microcontroller project. The top menu bar includes File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, and Help. The toolbar contains various icons for file operations, simulation, and debugging. The left sidebar shows the project structure with files like 'Practica0.c' and 'Resumen.txt', and sections for User/System, Internal, and External memory. The Registers window on the left lists the current values of various processor registers (R0-R15, CPSR, SPSR, etc.). The Disassembly window in the center shows assembly code with some lines highlighted in yellow. The Logic Analyzer window at the bottom shows memory dump and call stack information. The bottom status bar displays the command line and memory addresses.

Figura 1-8: Entorno de depuración de la aplicación.

A continuación debemos de visualizar los periféricos que nos interesen en cada ejercicio a través del menú *Peripherals*. En este caso son el *timer 0*, el *GPIO2* y el *VIC* (*Vectored Interrupt Controller*). Es decir, tendremos que activar los siguientes menús:

- Peripherals → Vectored Interrupt Controller
 - Peripherals → GPIO Fast Interface → Port 2
 - Peripherals → Timer → Timer 0

Además, para la visualización de las señales, el *Keil µVision* dispone de un analizador lógico virtual que podemos visualizarlo a partir de la herramienta rápida



o bien a través del menú View → Analysis Windows → Logic Analyzer

El entorno nos debe de quedar algo parecido a la figura siguiente:

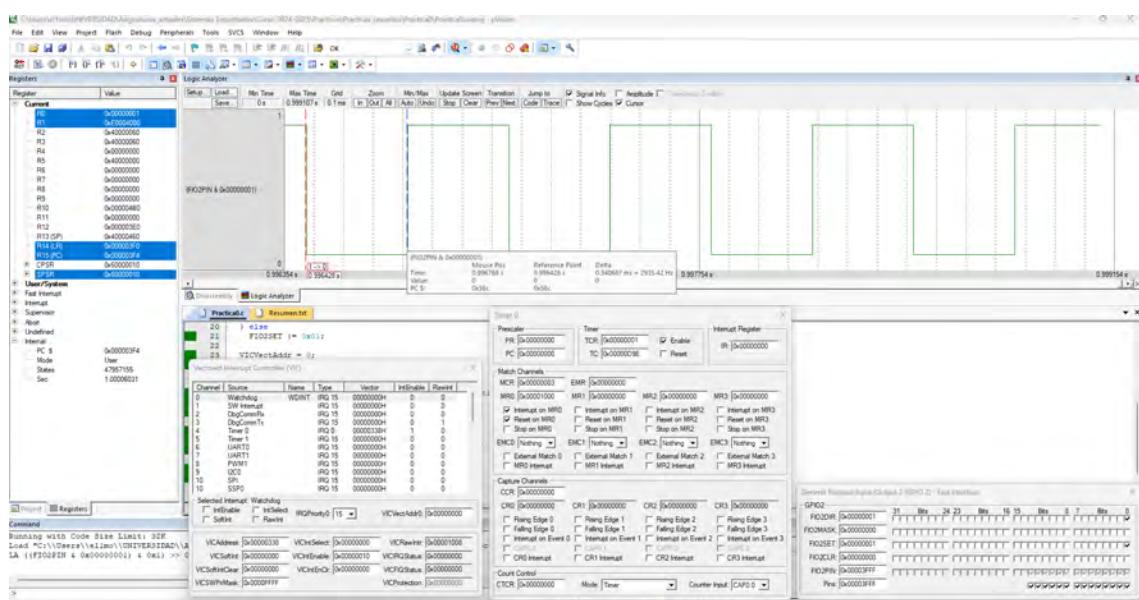


Figura 1-9: Entorno de depuración de la aplicación con los periféricos y el analizador lógico.

Para que en el analizador lógico nos salgan las señales que queremos visualizar debemos proceder de la forma siguiente:

- *Setup → New (Insert)*: FIO2PIN.0 y cambiar *Display type* a *Bit*.

- Marcar los ítems *Signal Info* y *Cursor*

- Ejecutar el programa (*Run*) y pararlo (*Stop*).

- Establecer la zona de visualización deseada con el *zoom* del analizador lógico



En la plataforma Moodle se suministran dos manuales más completos sobre el uso del analizador lógico de la herramienta *software Keil µVision*.

1.3.4. Ejercicio 3: Cargar el programa Blinky de Keil en la placa MCB2300

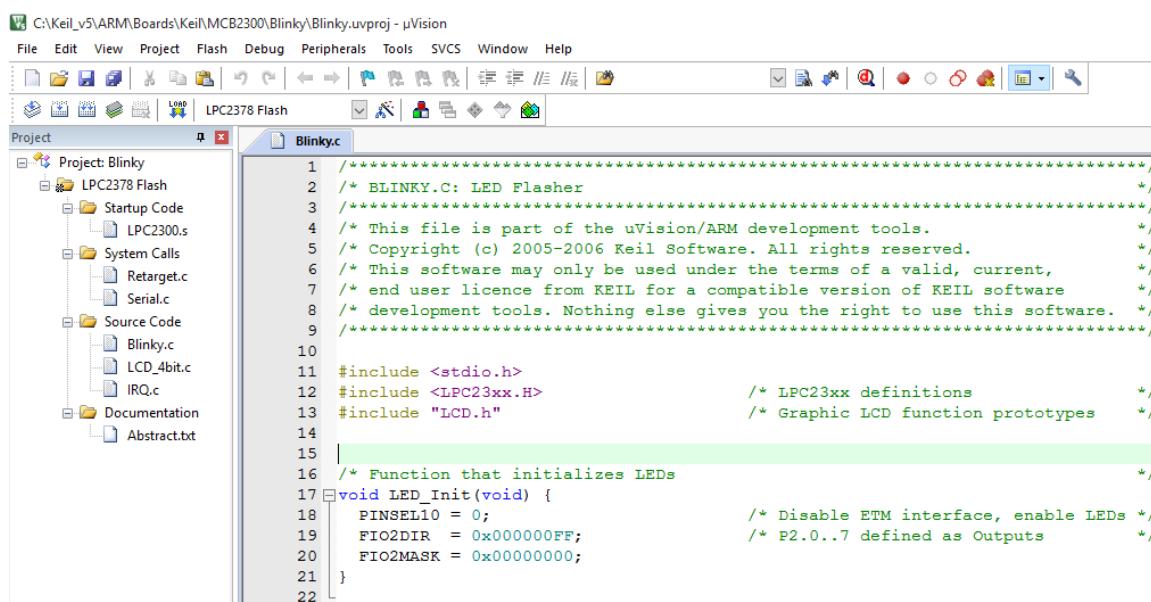
Este último ejercicio consiste en cargar en la memoria *flash* EPROM del microcontrolador LPC2378 de la placa MCB2378 un programa de ejemplo de los suministrados por el software de Keil µVision. Abriremos un proyecto sencillo llamado *Blinky* (guiños) que está situado por defecto en la siguiente ruta:

C:\Keil_v5\ARM\Boards\Keil\MCB2300\Blinky

Lo haremos a partir de la barra de menú: Project → Open Project.

Este proyecto utiliza los LEDs y el display LCD para mostrar el valor actual del convertidor A/D de la entrada AD0.0 controlada por el potenciómetro de la placa.

A continuación conoceremos los rasgos generales del entorno y aprenderemos a compilar, cargar, depurar y ejecutar un programa en dicha placa. Cerrar el proyecto de forma que en el programa quede todo configurado para las características de nuestra placa.



```

C:\Keil_v5\ARM\Boards\Keil\MCB2300\Blinky\Blinky.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project: Blinky
  LPC2378 Flash
    Startup Code
      LPC2300.s
    System Calls
      Retarget.c
        Serial.c
    Source Code
      Blinky.c
      LCD_4bit.c
      IRQ.c
    Documentation
      Abstract.txt
Blinky.c
1  /*************************************************************************/
2  /* BLINKY.C: LED Flasher */
3  /*************************************************************************/
4  /* This file is part of the uVision/ARM development tools. */
5  /* Copyright (c) 2005-2006 Keil Software. All rights reserved. */
6  /* This software may only be used under the terms of a valid, current, */
7  /* end user licence from KEIL for a compatible version of KEIL software */
8  /* development tools. Nothing else gives you the right to use this software. */
9  /*************************************************************************/
10
11 #include <stdio.h>
12 #include <LPC23xx.H>           /* LPC23xx definitions */
13 #include "LCD.h"               /* Graphic LCD function prototypes */
14
15
16 /* Function that initializes LEDs */
17 void LED_Init(void) {
18     PINSEL10 = 0;                /* Disable ETM interface, enable LEDs */
19     FIO2DIR = 0x000000FF;        /* P2.0..7 defined as Outputs */
20     FIO2MASK = 0x00000000;
21 }
22

```

Figura 1-10: Vista general del entorno Keil µVision 5.