SILICON LABORATORIES

# C8051F30X DEVELOPMENT KIT USER'S GUIDE

## 1. Kit Contents

The C8051F30x Development Kit contains the following items:

- C8051F30x Target Board
- C8051Fxxx Development Kit Quick-Start Guide
- Silicon Laboratories IDE and Product Information CD-ROM.  CD content includes:
    - Silicon Laboratories Integrated Development Environment (IDE)
    - Keil Software 8051 Development Tools (macro assembler, linker, evaluation 'C' compiler)
    - Source code examples and register definition files
    - Documentation
    - C8051F30x Development Kit User's Guide (this document)
- AC to DC Power Adapter
- USB Debug Adapter (USB to Debug Interface)
- USB Cable

## 2. Hardware Setup using a USB Debug Adapter

The target board is connected to a PC running the Silicon Laboratories IDE via the USB Debug Adapter as shown in Figure 1.

1. Connect the USB Debug Adapter to the DEBUG connector on the target board with the 10-pin ribbon cable.
2. Connect one end of the USB cable to the USB connector on the USB Debug Adapter.
3. Connect the other end of the USB cable to a USB Port on the PC.
4. Connect the ac/dc power adapter to power jack P1 on the target board.

**Notes:**
- Use the Reset button in the IDE to reset the target when connected using a USB Debug Adapter.
- Remove power from the target board and the USB Debug Adapter before connecting or disconnecting the ribbon cable from the target board. Connecting or disconnecting the cable when the devices have power can damage the device and/or the USB Debug Adapter.
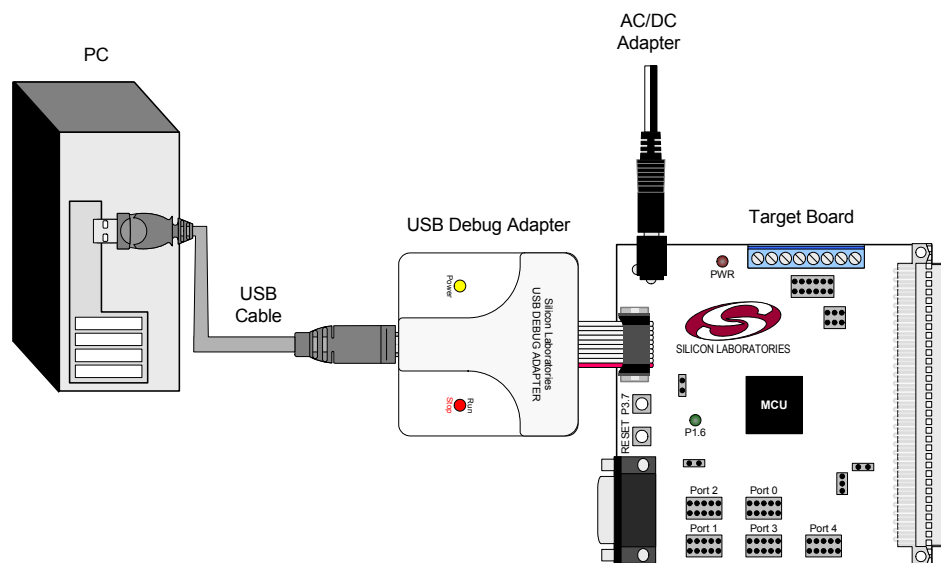


**Figure 1. Hardware Setup using a USB Debug Adapter**

**Copyright © 2006 by Silicon Laboratories**

# C8051F30x-DK

## 3. Software Setup

The included CD-ROM contains the Silicon Laboratories Integrated Development Environment (IDE), Keil software 8051 tools and additional documentation. Insert the CD-ROM into your PC's CD-ROM drive. An installer will automatically launch, allowing you to install the IDE software or read documentation by clicking buttons on the Installation Panel. If the installer does not automatically start when you insert the CD-ROM, run *autorun.exe* found in the root directory of the CD-ROM. Refer to the *readme.txt* file on the CD-ROM for the latest information regarding known IDE problems and restrictions.

## 4. Silicon Laboratories Integrated Development Environment

The Silicon Laboratories IDE integrates a source-code editor, source-level debugger and in-system Flash programmer. The use of third-party compilers and assemblers is also supported. This development kit includes the Keil Software A51 macro assembler, BL51 linker and evaluation version C51 'C' compiler. These tools can be used from within the Silicon Laboratories IDE.

### 4.1. System Requirements

The Silicon Laboratories IDE requirements:

- Pentium-class host PC running Microsoft Windows 98SE or later.
- One available COM or USB port.
- 64 MB RAM and 40 MB free HD space recommended.

### 4.2. Assembler and Linker

A full-version Keil A51 macro assembler and BL51 banking linker are included with the development kit and are installed during IDE installation. The complete assembler and linker reference manual can be found under the **Help** menu in the IDE or in the "*SiLabs\MCU\hlp*" directory (A51.pdf).

### 4.3. Evaluation C51 'C' Compiler

An evaluation version of the Keil C51 'C' compiler is included with the development kit and is installed during IDE installation. The evaluation version of the C51 compiler is the same as the full professional version except code size is limited to 4K bytes and the floating point library is not included. The C51 compiler reference manual can be found under the **Help** menu in the IDE or in the "*SiLabs\MCU\hlp*" directory (C51.pdf).

### 4.4. Using the Keil Software 8051 Tools with the Silicon Laboratories IDE

To perform source-level debugging with the IDE, you must configure the Keil 8051 tools to generate an absolute object file in the OMF-51 format with object extensions and debug records enabled. You may build the OMF-51 absolute object file by calling the Keil 8051 tools at the command line (e.g. batch file or make file) or by using the project manager built into the IDE. The default configuration when using the Silicon Laboratories IDE project manager enables object extension and debug record generation. Refer to Applications Note **AN104 - Integrating Keil 8051 Tools Into the Silicon Labs IDE** in the "*SiLabs\MCU\Documentation\Appnotes*" directory on the CD-ROM for additional information on using the Keil 8051 tools with the Silicon Laboratories IDE.

To build an absolute object file using the Silicon Laboratories IDE project manager, you must first create a project. A project consists of a set of files, IDE configuration, debug views, and a target build configuration (list of files and tool configurations used as input to the assembler, compiler, and linker when building an output object file).

The following sections illustrate the steps necessary to manually create a project with one or more source files, build a program and download the program to the target in preparation for debugging. (The IDE will automatically create a single-file project using the currently open and active source file if you select **Build/Make Project** before a project is defined.)

SILICON LABORATORIES

### 4.4.1. Creating a New Project

1.  Select **Project**→**New Project** to open a new project and reset all configuration settings to default.

2.  Select **File**→**New File** to open an editor window. Create your source file(s) and save the file(s) with a recognized extension, such as .c, .h, or .asm, to enable color syntax highlighting.

3.  Right-click on "New Project" in the **Project Window**. Select **Add files to project**. Select files in the file browser and click Open. Continue adding files until all project files have been added.

4.  For each of the files in the **Project Window** that you want assembled, compiled and linked into the target build, right-click on the file name and select **Add file to build**. Each file will be assembled or compiled as appropriate (based on file extension) and linked into the build of the absolute object file.

    **Note:** If a project contains a large number of files, the "Group" feature of the IDE can be used to organize. Right-click on "New Project" in the **Project Window**. Select **Add Groups to project**. Add pre-defined groups or add customized groups. Right-click on the group name and choose **Add file to group**. Select files to be added. Continue adding files until all project files have been added.

### 4.4.2. Building and Downloading the Program for Debugging

1.  Once all source files have been added to the target build, build the project by clicking on the **Build/Make Project** button in the toolbar or selecting **Project**→**Build/Make Project** from the menu.

    **Note:** After the project has been built the first time, the **Build/Make Project** command will only build the files that have been changed since the previous build. To rebuild all files and project dependencies, click on the **Rebuild All** button in the toolbar or select **Project**→**Rebuild All** from the menu.

2.  Before connecting to the target device, several connection options may need to be set. Open the **Connection Options** window by selecting **Options**→**Connection Options...** in the IDE menu. First, select the appropriate adapter in the "Serial Adapter" section. Next, the correct "Debug Interface" must be selected. C8051F30x family devices use the Silicon Labs 2-wire (C2) debug interface. Once all the selections are made, click the OK button to close the window.

3.  Click the **Connect** button in the toolbar or select **Debug**→**Connect** from the menu to connect to the device.

4.  Download the project to the target by clicking the **Download Code** button in the toolbar.

    **Note:** To enable automatic downloading if the program build is successful select **Enable automatic connect/download after build** in the **Project**→**Target Build Configuration** dialog. If errors occur during the build process, the IDE will not attempt the download.

5.  Save the project when finished with the debug session to preserve the current target build configuration, editor settings and the location of all open debug views. To save the project, select **Project->Save Project As...** from the menu. Create a new name for the project and click on **Save**.

SILICON LABORATORIES

# C8051F30x-DK

## 4.5. Example Source Code

Example source code and register definition files are provided in the "*SiLabs\MCU\Examples\C8051F30x*" directory during IDE installation. These files may be used as a template for code development. Example applications include a blinking LED example which configures the green LED on the target board to blink at a fixed rate.

## 4.6. Register Definition Files

Register definition files *C8051F300.inc and C8051F300.h* define all SFR registers and bit-addressable control/status bits. They are installed into the "*SiLabs\MCU\Examples\C8051F30x*" directory during IDE installation. The register and bit names are identical to those used in the C8051F30x data sheet. Both register definition files are also installed in the default search path used by the Keil Software 8051 tools. Therefore, when using the Keil 8051 tools included with the development kit (A51, C51), it is not necessary to copy a register definition file to each project's file directory.

## 4.7. Blinking LED Example

The example source files *blink.asm* and *blinky.c* show examples of several basic C8051F300 functions. These include; disabling the watchdog timer (WDT), configuring the Port I/O crossbar, configuring a timer for an interrupt routine, initializing the system clock, and configuring a GPIO port. When compiled/assembled and linked this program flashes the green LED on the C8051F300 target board about five times a second using the interrupt handler with a C8051F300 timer.

## 5. Target Board

The C8051F30x Development Kit includes a target board with a C8051F30x device pre-installed for evaluation and preliminary software development. Numerous input/output (I/O) connections are provided to facilitate prototyping using the target board. Refer to Figure 2 for the locations of the various I/O connectors.

P1          Power connector (accepts input from 7 to 15 VDC unregulated power adapter)
J1          34-pin Expansion I/O connector
J3          Port I/O Configuration Header
J4          DEBUG connector for Debug Adapter interface
J5          DB-9 connector for UART0 RS232 interface
J6          Analog I/O terminal block
J7          Low pass filter connector
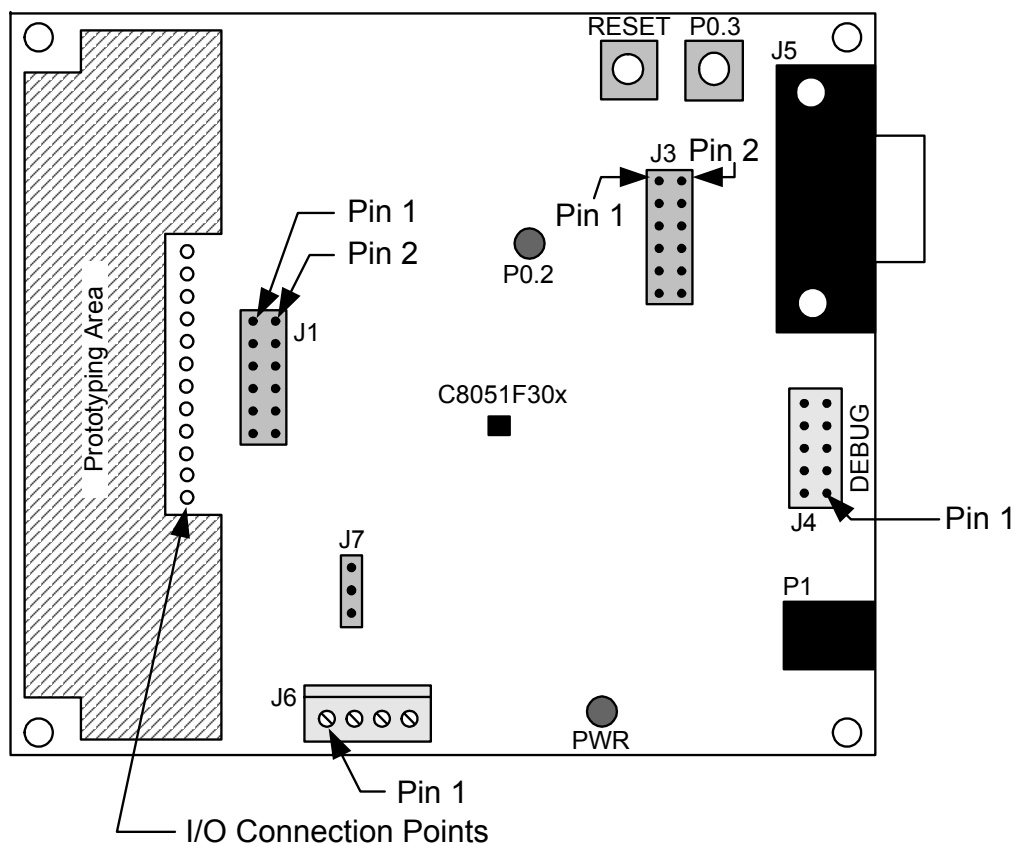J9, J10     External crystal enable connectors



**Figure 2. C8051F300 Target Board**

SILICON LABORATORIES

# C8051F30x-DK

## 5.1.  System Clock Sources

The C8051F300 device installed on the target board features a calibrated programmable internal oscillator which is enabled as the system clock source on reset. After reset, the internal oscillator operates at a frequency of 3.0625MHz (+/-2%) by default but may be configured by software to operate at other frequencies. Therefore, in many applications an external oscillator is not required. However, if you wish to operate the C8051F300 device at a frequency not available with the internal oscillator, an external crystal may be used. Refer to the C8051F30x data sheet for more information on configuring the system clock source.

The target board is designed to facilitate the installation of an external crystal. Remove resistors R7 and R8 and install the crystal at the pads marked Y1. Install a 10 MΩ resistor at R9 and install capacitors at C14 and C15 using values appropriate for the crystal you select. Refer to the C8051F30x data sheet for more information on the use of external oscillators.

## 5.2.  Switches and LEDs

Two switches are provided on the target board. Switch SW1 is connected to the RESET pin of the C8051F300. Pressing SW1 puts the device into its hardware-reset state. Switch SW2 is connected to the C8051F300's general purpose I/O (GPIO) pin through headers. Pressing SW2 generates a logic low signal on the port pin. Remove the shorting block from the header to disconnect SW2 from the port pins. The port pin signal is also routed to a pin on the J1 I/O connector. See Table 1 for the port pins and headers corresponding to each switch.

Two LEDs are also provided on the target board. The red LED labeled PWR is used to indicate a power connection to the target board. The green LED labeled with a port pin name is connected to the C8051F300's GPIO pin through headers. Remove the shorting block from the header to disconnect the LED from the port pin. The port pin signal is also routed to a pin on the J1 I/O connector. See Table 1 for the port pins and headers corresponding to each LED.

### Table 1. Target Board I/O Descriptions

| Description | I/O | Header |
|---|---|---|
| SW1 | Reset | none |
| SW2 | P0.3 | J3[3-4] |
| Green LED | P0.2 | J3[1-2] |
| Red LED | PWR | none |

SILICON LABORATORIES

## 5.3. Expansion I/O Connector (J1)

The 12-pin Expansion I/O connector J1 provides access to all signal pins of the C8051F300 device. Pins for +3V, digital ground and the output of an on-board low-pass filter are also available. A small through-hole prototyping area is also provided. All I/O signals routed to connector J1 are also routed to through-hole connection points between J1 and the prototyping area (see Figure 2 on page 5). Each connection point is labeled indicating the signal available at the connection point. See Table 2 for a list of pin descriptions for J1.

**Table 2. J1 Pin Descriptions**

| Pin # | Description |
|-------|-------------|
| 1 | +3VD (+3.3VDC) |
| 2 | PWM Output |
| 3 | P0.0 |
| 4 | P0.1 |
| 5 | P0.2 |
| 6 | P0.3 |
| 7 | P0.4 |
| 8 | P0.5 |
| 9 | P0.6 |
| 10 | P0.7 |
| 11 | GND (Ground) |
| 12 | /RST (Reset) |

## 5.4. Target Board DEBUG Interface (J4)

The DEBUG connector (J4) provides access to the DEBUG (C2) pins of the C8051F300. It is used to connect the Serial Adapter to the target board for in-circuit debugging and Flash programming. Table 3 shows the DEBUG pin definitions.

**Table 3. DEBUG Connector Pin Descriptions**

| Pin # | Description |
|-------|-------------|
| 1 | +3 VD (+3.3 VDC) |
| 2, 3, 9 | GND (Ground) |
| 4 | C2D |
| 5 | /RST (Reset) |
| 6 | P3.0 |
| 7 | C2CK |
| 8,10 | Not Connected |

SILICON LABORATORIES

## 5.5.  Serial Interface (J5)

A RS232 transceiver circuit and DB-9 (J5) connector are provided on the target board to facilitate serial connections to UART0 of the C8051F300. The TX, RX, RTS and CTS signals of UART0 may be connected to the DB-9 connector and transceiver by installing shorting blocks on headers J3.

    J3[5–6]    - Install shorting block to connect UART0 TX (P0.4) to transceiver.
    J3[7–8]    - Install shorting block to connect UART0 RX (P0.5) to transceiver.
    J3[9–10]   - Install shorting block to connect UART0 RTS (P0.2) to transceiver.
    J3[11–12] - Install shorting block to connect UART0 CTS (P0.3) to transceiver.

## 5.6.  Analog I/O (J6)

Several of the C8051F300 target device's port pins are connected to the J6 terminal block. Refer to Table 4 for the J6 terminal block connections. Install a shorting block on J7[2-3] to connect the AIN6 input to the P0.6 pin of the target device.

**Table 4. J6 Terminal Block Pin Descriptions**

| Pin # | Description |
|---|---|
| 1 | P0.1 / AIN1 (ADC input 1) |
| 2 | AIN6 (ADC input 6) |
| 3 | GND (Ground) |
| 4 | P0.0 / Vref (Voltage Reference) |

## 5.7.  Low-Pass Filter (J7)

The C8051F300 target board features a low-pass filter that may be connected to port pin P0.6. Install a shorting block on J7[1-2] to connect the P0.6 pin of the target device to the low-pass filter input. The output of the low-pass filter is routed to the PWM signal at J1[2]. The C8051F300 may be programmed to generate a PWM (Pulse-Width Modulated) waveform which is then input to the low-pass filter to implement a user-controlled PWM digital-to-analog converter. Refer to Applications Note "AN107: Implementing 16-Bit PWM Using the PCA" in the "*documentation*" directory on the CD-ROM for a discussion on generating a programmable dc voltage level with a PWM waveform and low-pass filter.
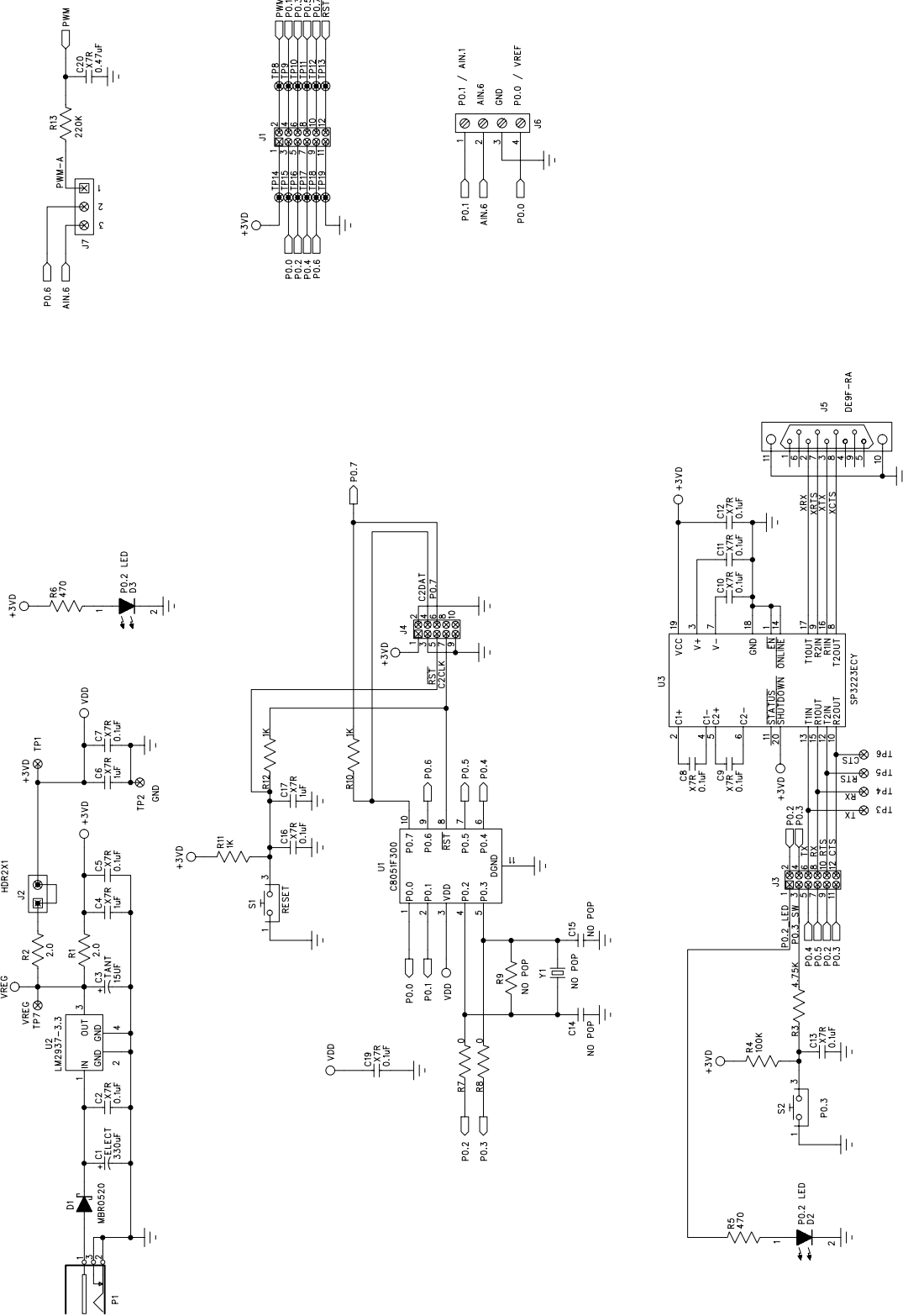
SILICON LABORATORIES

## 6. Schematic



**Figure 3. C8051F300TB Schematic**

# C8051F30x-DK

## DOCUMENT CHANGE LIST

### Revision 0.3 to Revision 0.4

- Section 1, added USB Debug Adapter and USB Cable.
- Section 2, changed name from "Hardware Setup" to "Hardware Setup using an EC2 Serial Adapter".
- Section 2, added 2 Notes bullets.
- Section 2, removed Note from bottom of page.
- Added Section 3, "Hardware Setup using a USB Debug Adapter".
- Section 5.4.2, changed step 2 to include new instructions.
- Section 7, J4, changed "Serial Adapter" to "Debug Adapter".
- Target Board DEBUG Interface Section, added USB Debug Adapter.
- DEBUG Connector Pin Descriptions Table, changed pin 4 to C2D.
- Changed "jumper" to "header".
- EC2 Serial Adapter section, added EC2 to the section title, table title and figure title.
- EC2 Serial Adapter section, changed "JTAG" to "DEBUG".
- Added "USB Debug Adapter" section.

### Revision 0.4 to Revision 0.5

- Removed EC2 Serial Adapter from Kit Contents.
- Removed Section 2. Hardware Setup using an EC2 Serial Adapter. See RS232 Serial Adapter (EC2) User's Guide.
- Removed Section 8. EC2 Serial Adapter. See RS232 Serial Adapter (EC2) User's Guide.
- Removed Section 9. USB Debug Adapter. See USB Debug Adapter User's Guide.

**NOTES:**

SILICON LABORATORIES

# C8051F30x-DK

## CONTACT INFORMATION

Silicon Laboratories Inc.
4635 Boston Lane
Austin, TX 78735
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

Email: MCUinfo@silabs.com
Internet: www.silabs.com

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Silicon Laboratories assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Silicon Laboratories assumes no responsibility for the functioning of undescribed features or parameters. Silicon Laboratories reserves the right to make changes without further notice. Silicon Laboratories makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Silicon Laboratories assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Silicon Laboratories products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Silicon Laboratories product could create a situation where personal injury or death may occur. Should Buyer purchase or use Silicon Laboratories products for any such unintended or unauthorized application, Buyer shall indemnify and hold Silicon Laboratories harmless against all claims and damages.

Silicon Laboratories and Silicon Labs are trademarks of Silicon Laboratories Inc.

Other products or brandnames mentioned herein are trademarks or registered trademarks of their respective holders.

SILICON LABORATORIES