C8051F2xx Development Kit User's Guide

1. Kit Contents

The C8051F2xx Development Kits contain the following items:

- C8051F206 or C8051F226 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
    - C8051F2xx 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 JTAG 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.

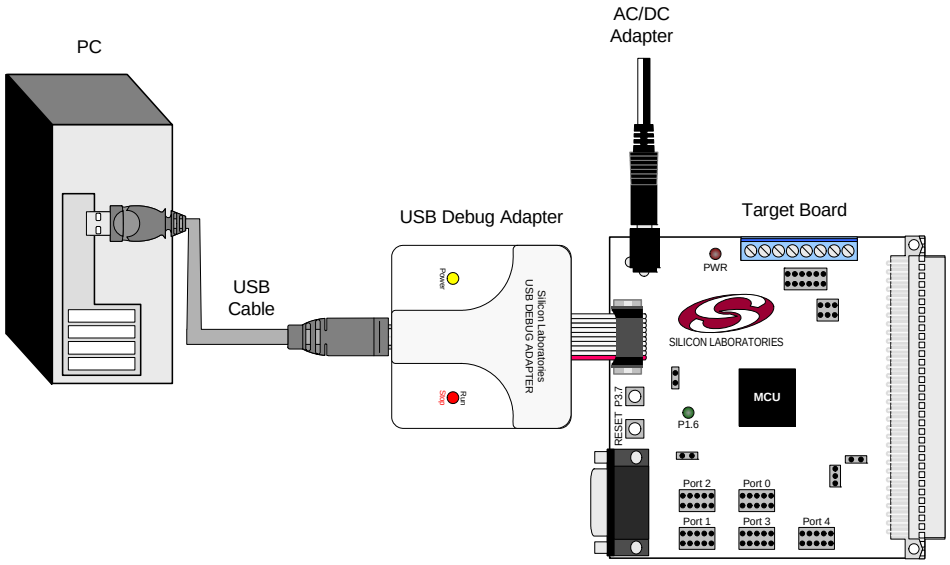


Figure 1. Hardware Setup using a USB Debug Adapter

 Copyright © 2006 by Silicon Laboratories

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 4 kB 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.)

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. C8051F2xx family devices use the JTAG 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**.

5. Example Source Code

Example source code and register definition files are provided in the "*SiLabs\MCU\Examples\C8051F2xx*" 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.

5.1. Register Definition Files

Register definition files *C8051F200.inc and C8051F200.h* define all SFR registers and bit-addressable control/status bits. They are installed into the "*SiLabs\MCU\Examples\C8051F2xx*" directory during IDE installation. The register and bit names are identical to those used in the C8051F2xx 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.

5.2. Blinking LED Example

The example source files *blink.asm* and *blinky.c* show examples of several basic C8051F2xx 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 target board about five times a second using the interrupt handler with a timer.

6. Target Board

The C8051F2xx Development Kit includes a target board with a C8051F206/26 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	VDDMonEn, Ties MONEN to +3VD2 or GND to enable/disable the VDD monitor
J2	64-pin I/O connector providing access to all I/O signals
P2.5	Connects SW2 to port pin P2.5
LED	Connects LED D2 to port pin P2.4
PWMIN	Connects PWM low-pass filter to port pin P2.7
J4	JTAG connector for Debug Adapter interface
J5	Connects P3.0 to analog signal from J6
J6	Analog I/O configuration connector
J7	Connects P3.1 to analog signal from J6
X1	Analog I/O terminal block

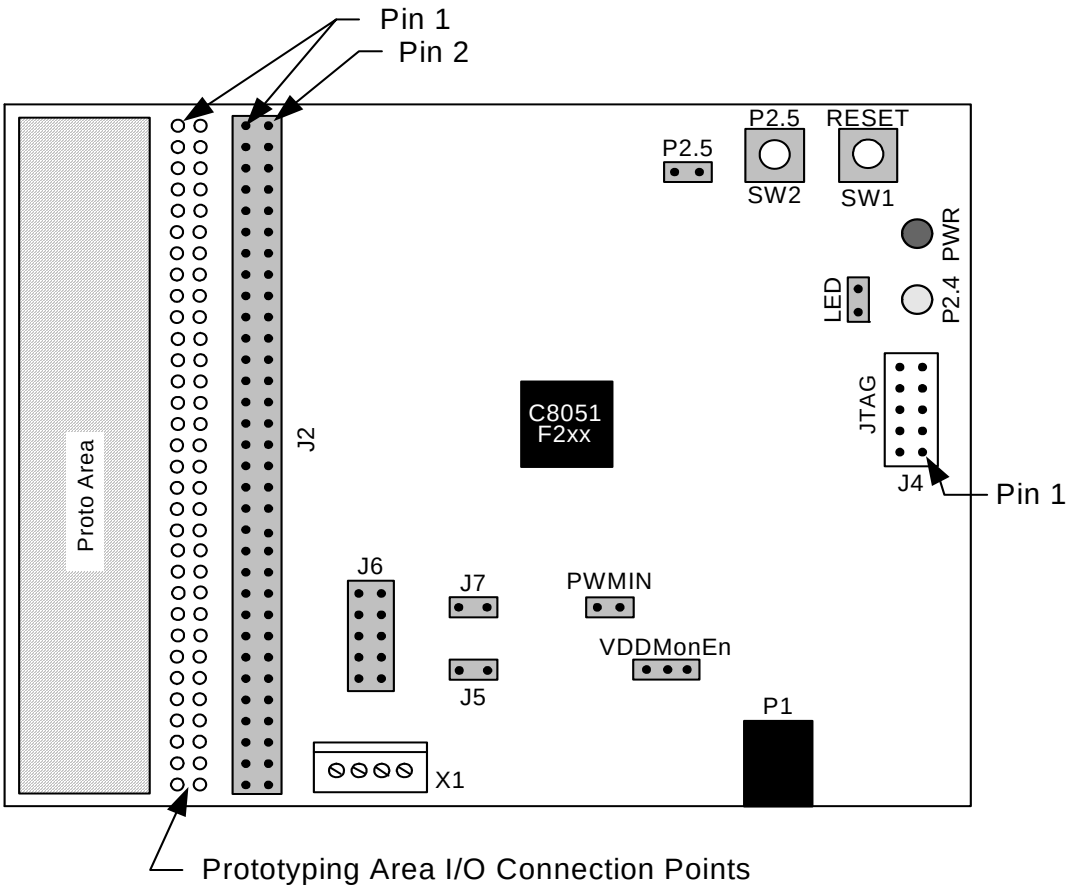


Figure 2. C8051F206 and C8051F226 Target Boards

6.1. System Clock Sources

The C8051F2xx device installed on the target board features a internal oscillator which is enabled as the system clock source on reset. After reset, the internal oscillator operates at a frequency of 2MHz (+/-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, an external crystal may be installed on the target board for additional applications. The target board is designed to facilitate the installation of an external crystal at the pads marked Q1. Refer to the C8051F2xx datasheet for more information on configuring the system clock source. Following are a few part numbers of suitable crystals:

Freq(MHz)	Digikey P/N	ECS P/N	
22.1184	X063-ND	ECS-221-20-1	(20 pF loading capacitance)
18.432	X146-ND	ECS-184-20-1	(20 pF loading capacitance)
11.0592	X089-ND	ECS-110.5-20-1	(20 pF loading capacitance)

6.2. Switches and LEDs

Two switches are provided on the target board. Switch SW1 is connected to the RESET pin of the C8051F2xx device on the target board. Pressing SW1 puts the device into its hardware-reset state. The device will leave the reset state after SW1 is released. Switch SW2 is connected to the device'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 J2 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 device'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 J2 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	P2.5	P2.5
Green LED	P2.4	LED
Red LED	PWR	none

6.3. Target Board JTAG Interface (J4)

The JTAG connector (J4) provides access to the JTAG pins of the C8051F2xx. It is used to connect the Serial Adapter or the USB Debug Adapter to the target board for in-circuit debugging and Flash programming. Table 2 shows the JTAG pin definitions.

Table 2. JTAG Connector Pin Descriptions

Pin #	Description
1	+3VD (+3.3VDC)
2, 3, 9	GND (Ground)
4	TCK
5	TMS
6	TDO
7	TDI
8, 10	Not Connected

6.4. Analog I/O (J5, J6, J7, Terminal Block)

An Analog I/O Configuration connector (J6) provides the ability to route analog I/O signals from the C8051F2xx to a terminal block in addition to connector J2 by installing shorting blocks on J6. Additionally, if shorting blocks are installed on J5 and J7, the analog signals routed through J6 can be inputs to the C8051F2xx at port pins P3.0 and/or P3.1. The port pins can then be configured as inputs to the on-chip ADC for evaluation. J6 also allows the user to route analog signals from the terminal block to port pins P1.3 and P1.4. These port pin can then be configured as inputs to Comparator 1. The PWM signal from the low-pass filter is also routed to J6 to provide a user controlled analog voltage level. This signal can then be used to evaluate the on-chip ADC by placing a shorting block on J6 (provided the J5 or J7 header is shorted). Refer to Figure 3 to determine the shorting block installation positions required to create the desired analog signal paths. Refer to Table 3 for terminal block connections and Table 4 for J6 pin definitions.

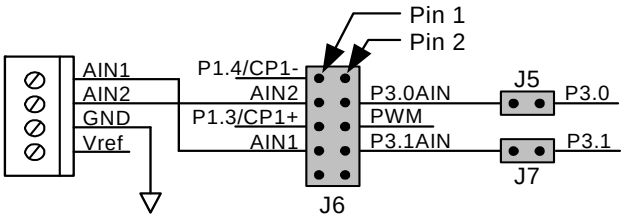


Figure 3. J6 Analog I/O Configuration Connector

Table 3. Terminal Block Pin Descriptions

Pin #	Description
1	VREF
2	GND
7	AIN2
8	AIN1

Table 4. J6 Connector Pin Descriptions

Pin #	Description
1	P1.4/CP1-
2, 9, 10	NC
3	AIN2
4	P3.0AIN
5	P1.3/CP1+
6	PWM
7	AIN1
8	P3.1AIN

6.5. VDD Monitor Disable (J1)

The VDD Monitor of the C8051F2xx may be disabled by moving the shorting block on J1 from pins 1-2 to pins 2-3, as shown in Figure 4.

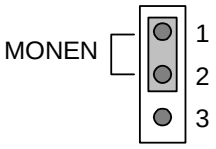


Figure 4. VDD Monitor Hardware Setup

6.6. Low-pass Filter (PWMIN)

The C8051F2xx target board features a low-pass filter that may be connected to port pin P2.7. Install a shorting block on connector “PWMIN” to connect the P2.7 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 J6[6]. To route the PWM signal to the ADC on the target device, a shorting block should be placed on “PWMIN”, and a shorting block placed on connector “P3.0AIN” (J5) or “P3.1AIN” (J7). Connector J6 is used to connect the PWM voltage to the P3.0AIN or P3.1AIN signal routed to the corresponding port pins (See Figure 3). These port pins are then configured for use by the ADC.

The C8051F2xx 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 **AN110 - Implementing 16-Bit PWM Using an On-Chip Timer** 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.

6.7. Expansion I/O Connector (J2)

The target board provides access to all C8051F2xx signals (except the four JTAG signals: TCK, TMS, TDO and TDI used to connect the Emulation Cartridge – these are accessed using test points in place near the J4 header) via the 64-pin connector J2. A small through-hole prototyping area is also provided. All I/O signals routed to connector J2 are also routed to through-hole connection points between J2 and the prototyping area (see Figure 2). The signal layout pattern of these connection points is identical to the adjacent J2 connector pins. Table 5 shows the pin-out of the J2 connector.

Table 5. J2 Pin Descriptions

Pin	Description
1,46,64	+3 VD2 (voltage supply)
2	XTAL1
3	P1.6
4	P1.7
5	P1.4
6	P1.5
7	P1.2
8	P1.3
9	P1.0
10	P1.1
11	P0.6
12	P0.7
13	P0.4
14	P0.5
15	P0.2
16	P0.3
17	P0.0
18	P0.1
19	P2.6
20	P2.7

Pin	Description
21	P2.4
22	P2.5
23	P2.2
24	P2.3
25	P2.0
26	P2.1
27	P3.6
28	P3.7
29	P3.4
30	P3.5
31	P3.2
32	P3.3
33	P3.0
34	P3.1
36	/RST
39,41,42,45,47,63	GND
48,50	PWM (pulse-width modulator)
53	VREF
62	VDDMONEN

7. Schematic

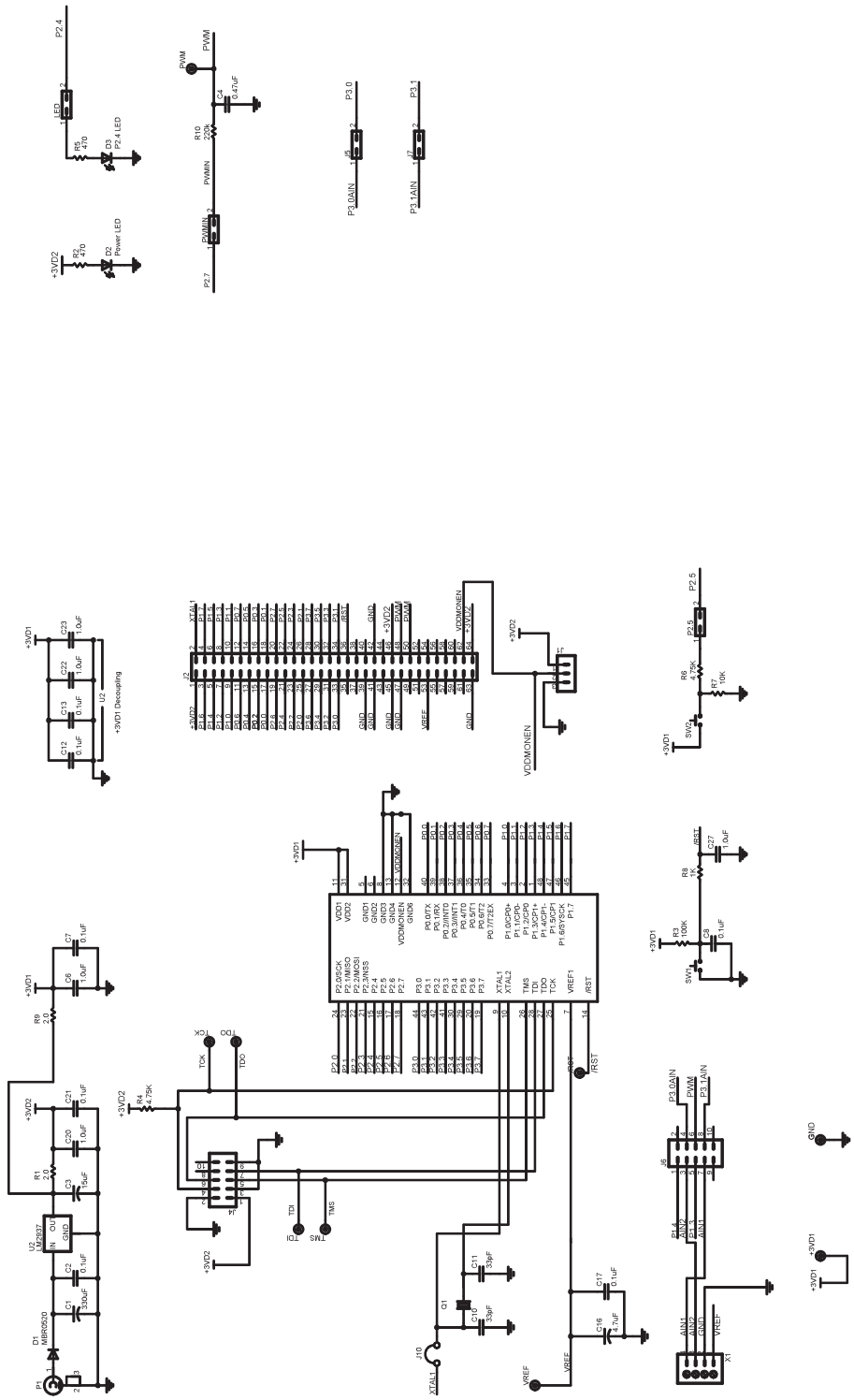


Figure 5. C8051F206/226 Target Board Schematic

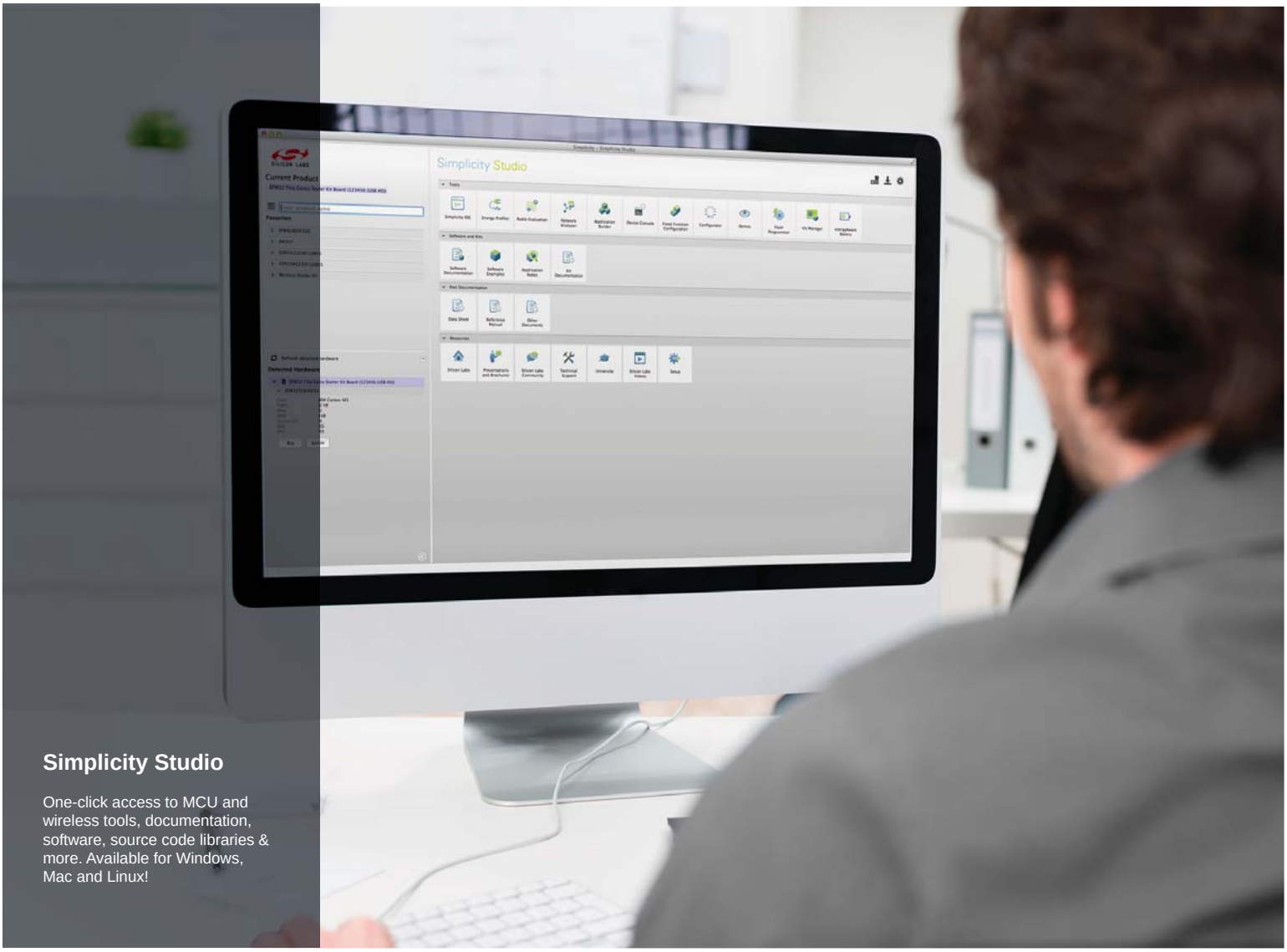
DOCUMENT CHANGE LIST

Revision 0.4 to Revision 0.5

- 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 0.5 to Revision 0.6

- 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.



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

Disclaimer
Silicon Laboratories intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Laboratories products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Laboratories reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Laboratories shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products must not be used within any Life Support System without the specific written consent of Silicon Laboratories. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Laboratories products are generally not intended for military applications. Silicon Laboratories products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

Trademark Information
Silicon Laboratories Inc., Silicon Laboratories, Silicon Labs, SiLabs and the Silicon Labs logo, CMEMS®, EFM, EFM32, EFR, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZMac®, EZRadio®, EZRadioPRO®, DSPLL®, ISOmodem ®, Precision32®, ProSLIC®, SiPHY®, USBXpress® and others are trademarks or registered trademarks of Silicon Laboratories Inc. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com