



C8051F35X DEVELOPMENT KIT USER'S GUIDE

1. Kit Contents

The C8051F35x Development Kit contains the following items:

- C8051F350 Target Board
- C8051Fxxx Development Kit Quick-Start Guide
- C8051F35x Development Kit User's Guide (this document)
- Silicon Laboratories IDE and Product Information CD-ROM. CD content includes:
 - Silicon Laboratories Integrated Development Environment (IDE)
 - Keil 8051 Development Tools (macro assembler, linker, evaluation 'C' compiler)
 - Source code examples and register definition files
 - Documentation
- AC to DC Power Adapter
- Debug Adapter. Items include:
 - EC2 Serial Adapter (RS232 to Debug Interface)
 - RS232 Serial Cable
 - 7" Ribbon Cable
- OR
- USB Debug Adapter (USB to Debug Interface)
- USB Cable

2. Hardware Setup using an EC2 Serial Adapter

Connect the target board to a PC running the Silicon Laboratories IDE via the Serial Adapter as shown in Figure 1.

1. Connect one end of the RS232 serial cable to a serial (COM) port on the PC.
2. Connect the other end of the RS232 serial cable to the DB-9 connector on the Serial Adapter.
3. Connect the Serial Adapter to the DEBUG connector on the target board using the 10-pin ribbon cable.
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 an EC2 Serial Adapter.
- Remove power from the target board and the EC2 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 EC2.

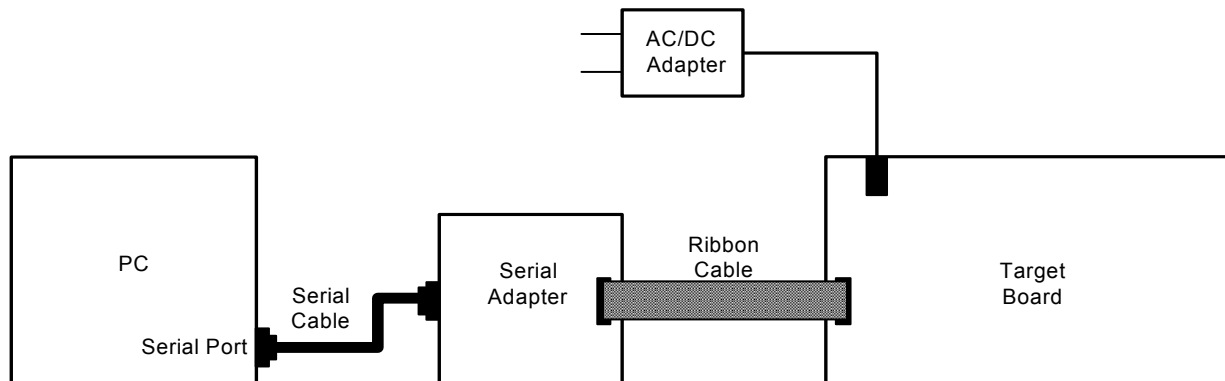


Figure 1. Hardware Setup using an EC2 Serial Adapter

3. 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 2.

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.

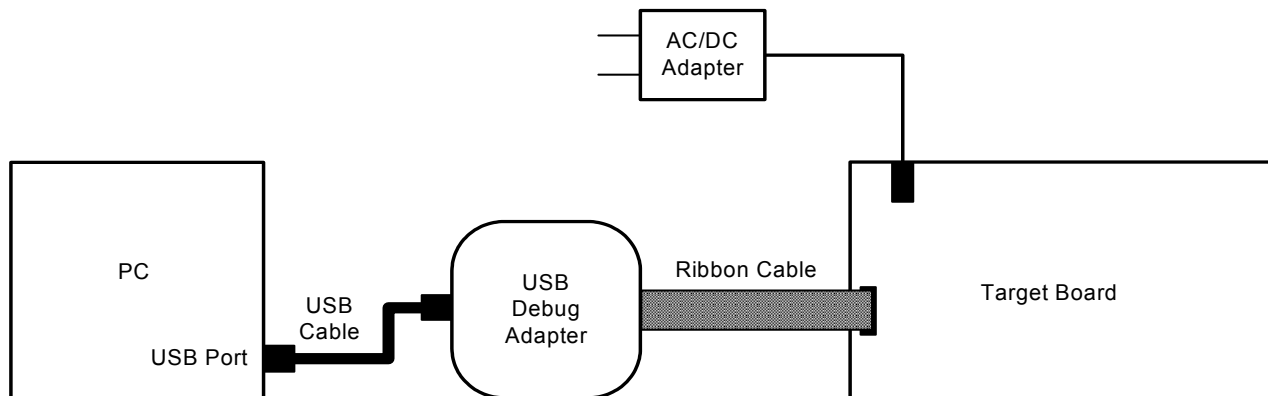


Figure 2. Hardware Setup using a USB Debug Adapter

4. 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.

5. 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.

5.1. 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.

5.2. 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).

5.3. 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).

5.4. 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.)

5.4.1. 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.

5.4.2. 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 adapter that was included with the kit in the “Serial Adapter” section. Next, the correct “Debug Interface” must be selected. C8051F35x 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**.

6. Example Source Code

Example source code and register definition files are provided in the “*SiLabs\MCU\Examples\C8051F35x*” 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.

6.1. Register Definition Files

Register definition files *C8051F350.inc* and *C8051F350.h* define all SFR registers and bit-addressable control/status bits. They are installed into the “*SiLabs\MCU\Examples\C8051F35x*” directory during IDE installation. The register and bit names are identical to those used in the C8051F35x 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.

6.2. Blinking LED Example

The example source files *blink.asm* and *blink.c* show examples of several basic C8051F35x 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 C8051F350 target board about five times a second using the interrupt handler with a C8051F350 timer.

7. Target Board

The C8051F35x Development Kit includes a target board with a C8051F350 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 3 for the locations of the various I/O connectors.

- P1 Power connector (accepts input from 7 to 15 VDC unregulated power adapter)
- J1 22-pin Expansion I/O connector
- J3 Port I/O Configuration Jumper Block
- J4 DEBUG connector for Debug Adapter interface
- J5 DB-9 connector for UART0 RS232 interface
- J6 Analog I/O terminal block
- J7 Connector for IDAC0 voltage circuit
- J8 USB Debug Adapter target board power connector
- J9, J10 External crystal enable connectors
- J11 Connector for IDAC1 voltage circuit
- J12 Connector block for Thermistor circuitry
- J13, J14 ADC external voltage reference connectors

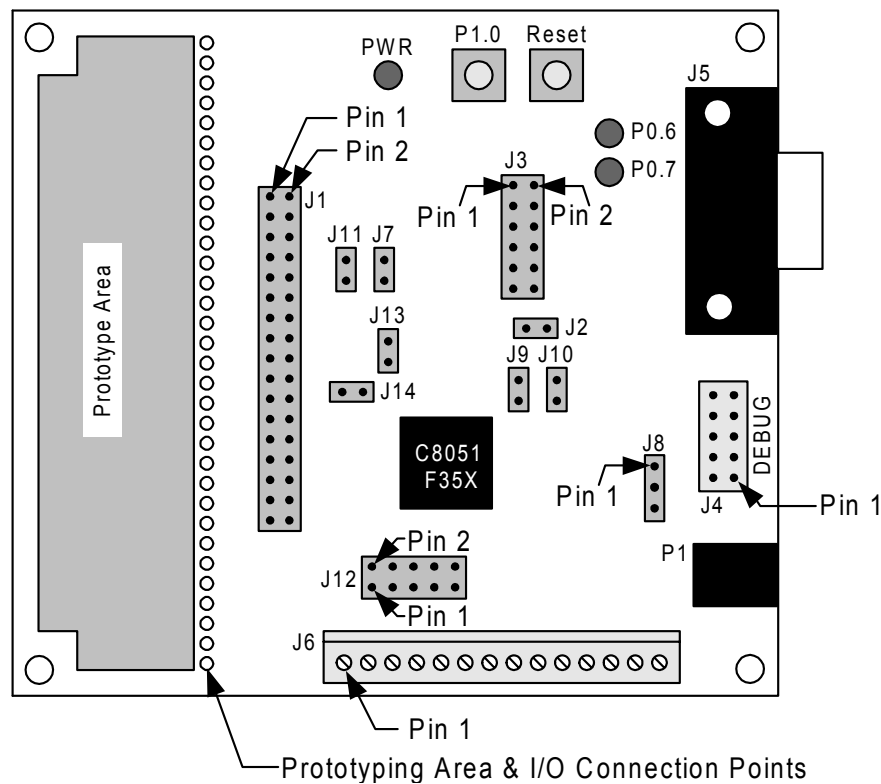


Figure 3. C8051F350 Target Board

7.1. System Clock Sources

The C8051F350 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 3.0625 MHz ($\pm 2\%$) 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 C8051F350 device at a frequency not available with the internal oscillator, an external crystal may be used. Refer to the C8051F35x 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 shorting blocks at headers J9 and J10 and install the crystal at the pads marked Y1. Install a 10 M Ω resistor at R9 and install capacitors at C14 and C15 using values appropriate for the crystal you select. Refer to the C8051F35x data sheet for more information on the use of external oscillators.

7.2. Switches and LEDs

Two switches are provided on the target board. Switch SW1 is connected to the RESET pin of the C8051F350. Pressing SW1 puts the device into its hardware-reset state. Switch SW2 is connected to the C8051F350'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 jumper 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.

Three 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 LEDs labeled with port pin names are connected to the C8051F350's GPIO pins through headers. Remove the shorting blocks from the headers to disconnect the LEDs from the port pins. The port pin signals are also routed to pins on the J1 I/O connector. See Table 1 for the port pins and headers corresponding to each LED.

Description	I/O	Jumper
SW1	Reset	none
SW2	P1.0	J3[5-6]
Green LED D2	P0.6	J3[1-2]
Green LED D1	P0.7	J3[3-4]
Red LED	PWR	none

Table 1. Target Board I/O Descriptions

7.3. Expansion I/O Connector (J1)

The 34-pin Expansion I/O connector J1 provides access to all signal pins of the C8051F350 device. Pins for V_{DD} and GND as well as pins for V_{DDA} and AGND 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 3 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.

Pin #	Description	Pin #	Description	Pin #	Description
1	V_{DD}	13	P1.2	25	AIN3
2	GND	14	P1.3	26	AIN4
3	P0.0	15	P1.4	27	AIN5
4	P0.1	16	P1.5	28	AIN6
5	P0.2	17	P1.6/IDAC0	29	AIN7
6	P0.3	18	P1.7/IDAC1	30	VREF+
7	P0.4	19	P2.0	31	/RST
8	P0.5	20	AGND	32	VREF-
9	P0.6	21	AGND	33	V_{DDA}
10	P0.7	22	AIN0	34	AGND
11	P1.0	23	AIN1	-	-
	P1.1	24	AIN2	-	-

Table 2. J1 Pin Descriptions

7.4. Target Board DEBUG Interface (J4)

The DEBUG connector (J4) provides access to the DEBUG (C2) pins of the C8051F350. 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 3 shows the DEBUG pin definitions.

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

Table 3. DEBUG Connector Pin Descriptions

7.5. 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 C8051F350. The TX, RX, RTS and CTS signals of UART0 may be connected to the DB-9 connector and transceiver by installing shorting blocks on header J3.

- J3[7-8] - Install shorting block to connect UART0 TX (P0.4) to transceiver.
- J3[9-10] - Install shorting block to connect UART0 RX (P0.5) to transceiver.
- J3[11-12] - Install shorting block to connect UART0 RTS (P1.4) to transceiver.
- J3[13-14] - Install shorting block to connect UART0 CTS (P1.5) to transceiver.

7.6. Analog I/O (J6)

Many of the C8051F350 target device's port pins are connected to the J6 terminal block. Connections for VDDA, AGND, ADC external voltage references, IDAC outputs and ADC inputs are available. Refer to Table 4 for the J6 terminal block connections.

Pin #	Description	Pin #	Description
1	IDAC0	8	AIN3
2	IDAC1	9	AIN4
3	VREF+	10	AIN5
4	VREF-	11	AIN6
5	AIN0	12	AIN7
6	AIN1	13	AGND
7	AIN2	14	VDDA

Table 4. J6 Terminal Block Pin Descriptions

7.7. IDAC Connectors (J7, J11)

The C8051F350 target board also features two Current-to-Voltage 1 K Ω load resistors that may be connected to the 10-bit current-mode Digital-to-Analog Converters (IDACs) on port pins P1.6 and P1.7. Install a shorting block on J7 to connect the IDAC0/P1.6 pin of the target device to a load resistor. Install a second shorting block on J11 to connect the IDAC1/P1.7 pin of the target device to a load resistor. The IDAC signals are then routed to the J1 and J6 connectors.

7.8. USB Debug Adapter Target Board Power Connector (J8)

The USB Debug Adapter includes a connection to provide power to the target board. This connection is routed from J4[10] to J8[1]. Place a shorting block at header J8[2-3] to power the board directly from an AC/DC power adapter. Place a shorting block at header J8[1-2] to power the board from the USB Debug Adapter. Please note that the second option is not supported with either the EC1 or EC2 Serial Adapters.

7.9. Thermistor Connector Block (J12)

A thermistor circuit is included on the board which allows for a ratiometric measurement of the included thermistor. To use the circuit, remove the shorting blocks on the IDAC headers (J7, J11) and the External Voltage Reference headers (J13, J14). Next, connect all of the signals on the thermistor connector block (J12). Software on the chip can be written to use IDAC0 as a reference current source for the circuit. The current creates a voltage across R27 which is used as an external Voltage Reference input, while the voltage across the thermistor (R26) can be measured by connecting AIN0.0 to the AIN+ ADC input, and AIN0.1 to the AIN- ADC input.

7.10. External Voltage Reference (J13, J14)

A precision 2.5V voltage reference source is also included on the C8051F350 target board. To use this circuit as a reference source for the ADC, place shorting block on headers J13 and J14. Note that software on the device should be configured to use an external reference when using this configuration. When using the internal (on-chip) reference source, headers J13 and J14 should be left open.

8. EC2 Serial Adapter

The Serial Adapter provides the interface between the PC's RS232 serial port and the C8051F35x's in-system debug/programming circuitry. The Serial Adapter connects to the C8051F350 C2 debug interface on the target board using the 10-pin connector on the Serial Adapter labeled "DEBUG", see Figure 4. (The Serial Adapter supports both Silicon Laboratories JTAG and C2 debug interfaces.) All Serial Adapters may be powered from the target board, but the EC1 and EC2 Serial Adapter units cannot provide power to the target board. Table 5 shows the pin definitions for the Serial Adapter's DEBUG connector.

Notes:

- When powering the Serial Adapter via the DEBUG connector, the input voltage to the DEBUG connector's power pin must be 3.0 to 3.6 VDC. Otherwise, the Serial Adapter must be powered directly by connecting the AC/DC adapter to the Serial Adapter's DC power jack.
- The Serial Adapter requires a target system clock of 32 KHz or greater.

Pin #	Description
1	3.0 to 3.6 VDC Input
2,3,9	GND (Ground)
4	TCK (C2D)
5	TMS
6	TDO
7	TDI (C2CK)
8,10	Not Connected

Table 5. EC2 Serial Adapter DEBUG Connector Pin Descriptions

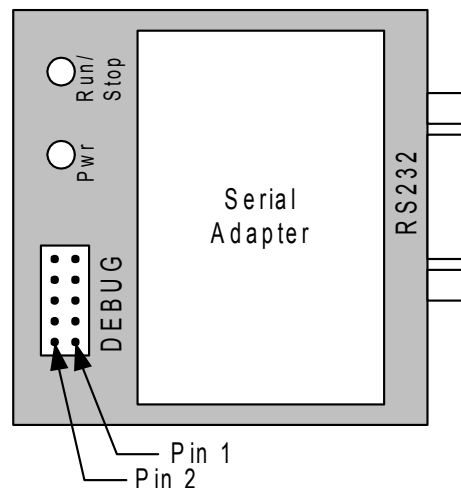


Figure 4. EC2 Serial Adapter DEBUG Connector

9. USB Debug Adapter

The USB Debug Adapter provides the interface between the PC's USB port and the C8051F35x's in-system debug/programming circuitry. The attached 10-pin DEBUG ribbon cable connects the adapter to the target board and the target device's debug interface signals. (The USB Debug Adapter supports both Silicon Laboratories JTAG and C2 debug interfaces.) Power is provided to the adapter from the USB connection to the PC. The USB Debug Adapter is capable of providing power to a circuit board via pin 10 of the DEBUG connector. See Section 7.8. for instructions on powering the C8051F350 target board from this source. Table 6 shows the pin definitions for the DEBUG ribbon cable connector.

Notes:

- The USB Debug Adapter requires a target system clock of 32 KHz or greater.
- With the default settings, the USB Debug Adapter can supply up to 100 mA to a target system.

Pin #	Description
1,8	Not Connected
2,3,9	GND (Ground)
4	TCK (C2D)
5	TMS
6	TDO
7	TDI (C2CK)
10	USB Power

Table 6. USB Debug Adapter DEBUG Connector Pin Descriptions

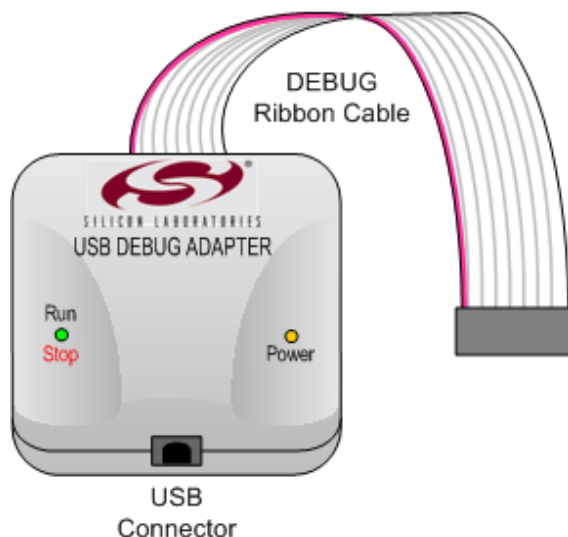


Figure 5. USB Debug Adapter

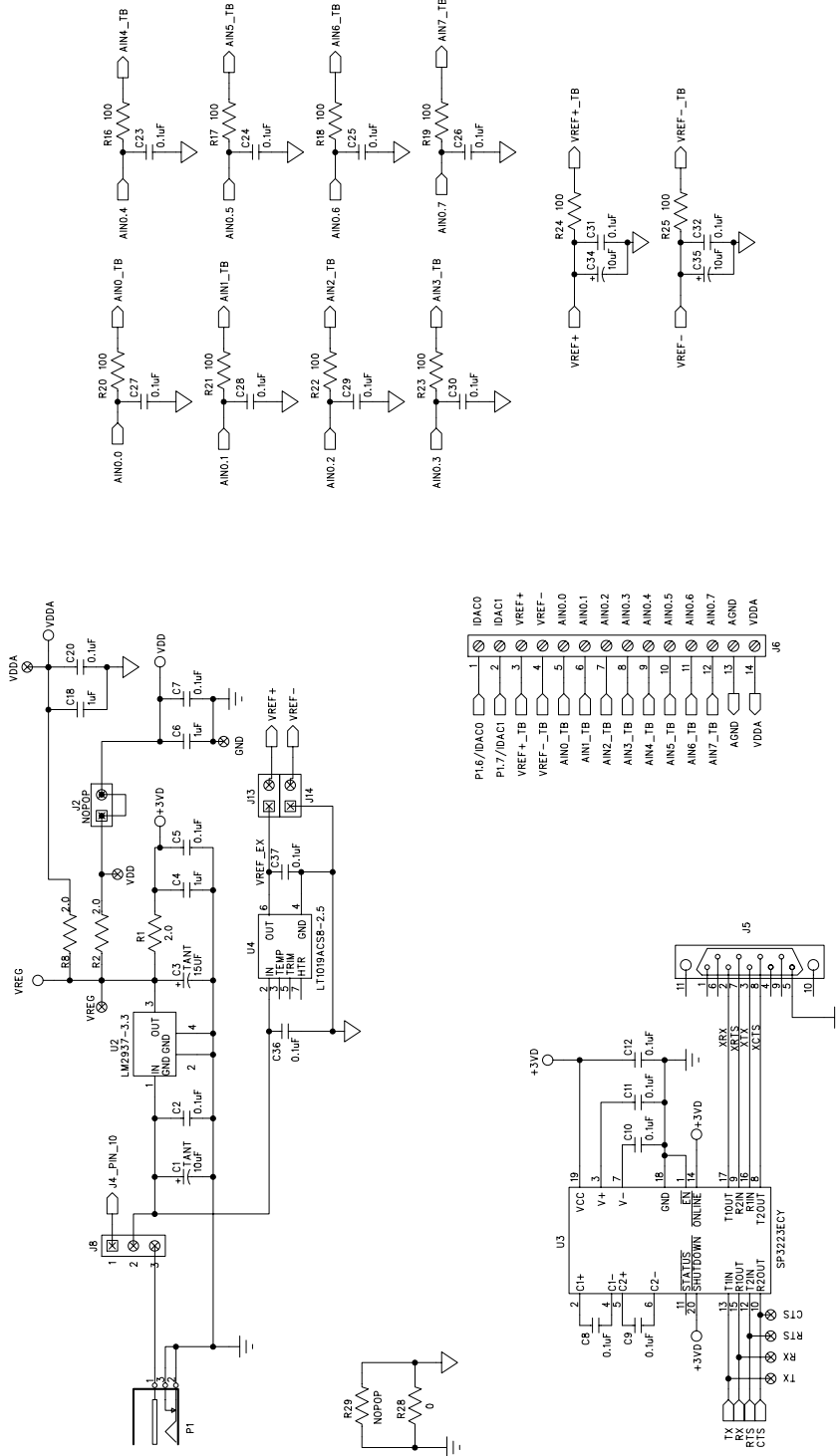


Figure 7. C8051F350 Target Board Schematic - Page 2

DOCUMENT CHANGE LIST

Revision 0.1 to Revision 0.2

- 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.
- Section 7, J8, changed "Serial Adapter" to "USB Debug Adapter"
- DEBUG Connector Pin Descriptions Table, changed pin 10 to USB Power
- USB Debug Adapter Target Board Power Connector (J8) Section, changed "Serial" to "USB Debug"



C8051F35x-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: productinfo@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.