C8051F336DK

C8051F336 Development Kit User's Guide

1. Relevant Devices

The C8051F336 Development Kit is intended as a development platform for the microcontrollers in the C8051F336/7/8/9 MCU family. The members of this MCU family are: C8051F336, C8051F337, C8051F338, C8051F339.

Notes:

- The target board included in this kit is provided with a pre-soldered **C8051F338** MCU (QFN24 package).
- Code developed on the C8051F338 can be easily ported to the other members of this MCU family.
- Refer to the C8051F336/7/8/9 data sheet for the differences between the members of this MCU family.
- The C8051F336/7/8/9 family of devices are code-compatible with the C8051F330/1/2/3/4/5 family of devices.

2. Kit Contents

The C8051F336 Development Kit contains the following items:

- C8051F338 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 8051 Development Tools (macro assembler, linker, evaluation 'C' compiler)
    - Source code examples and register definition files
    - Documentation
    - C8051F336 Development Kit User's Guide (this document)
- AC to DC Power Adapter
- USB Debug Adapter (USB to Debug Interface)
- USB Cable

3. Getting Started

The necessary software to download, debug, and communicate with the target microcontroller is included in the CD-ROM. The following software is necessary to build a project, download code to, and communicate with the target microcontroller:

- Silicon Laboratories Integrated Development Environment (IDE)
- Keil 8051 Development Tools (macro assembler, linker, evaluation 'C' compiler)

Other useful software that is provided in the CD-ROM includes:

- Configuration Wizard 2
- Keil uVision Drivers
- CP210x USB to UART Virtual COM Port (VCP) Drivers

3.1. Software Installation

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 *ReleaseNotes.txt* file on the CD-ROM for the latest information regarding known problems and restrictions. After installing the software, see the following sections for information regarding the software and running one of the demo applications.

3.2. CP210x USB to UART VCP Driver Installation

The C8051F338 Target Board includes a Silicon Laboratories CP2102 USB-to-UART Bridge Controller. Device drivers for the CP2102 need to be installed before PC software such as HyperTerminal can communicate with the target board over the USB connection. If the "Install CP210x Drivers" option was selected during installation, this will launch a driver "unpacker" utility.

1. Follow the steps to copy the driver files to the desired location. The default directory is *C:\SiLabs\MCU\CP210x*.
2. The final window will give an option to install the driver on the target system. Select the "Launch the CP210x VCP Driver Installer" option if you are ready to install the driver.
3. If selected, the driver installer will now launch, providing an option to specify the driver installation location. After pressing the "Install" button, the installer will search your system for copies of previously installed CP210x Virtual COM Port drivers. It will let you know when your system is up to date. The driver files included in this installation have been certified by Microsoft.
4. If the "Launch the CP210x VCP Driver Installer" option was not selected in step 3, the installer can be found in the location specified in step 2, by default *C:\SiLabs\MCU\CP210x\Windows_2K_XP_S2K3_Vista*. At this location run *CP210xVCPInstaller.exe*.
5. To complete the installation process, connect the included USB cable between the host computer and the USB connector (P2) on the C8051F338 Target Board. Windows will automatically finish the driver installation. Information windows will pop up from the taskbar to show the installation progress.
6. If needed, the driver files can be uninstalled by selecting "Silicon Laboratories CP210x USB to UART Bridge (Driver Removal" option in the "Add or Remove Programs" window.

4. Software Overview

4.1. Silicon Laboratories IDE

The Silicon Laboratories IDE integrates a source-code editor, a source-level debugger, and an in-system Flash programmer. See Section 6. "Using the Keil Software 8051 Tools with the Silicon Laboratories IDE," on page 5 for detailed information on how to use the IDE. The Keil Evaluation Toolset includes a compiler, linker, and assembler and easily integrates into the IDE. The use of third-party compilers and assemblers is also supported.

4.1.1. IDE System Requirements

The Silicon Laboratories IDE requirements:

- Pentium-class host PC running Microsoft Windows 2000 or newer.
- One available USB port.
- 64 MB RAM and 40 MB free HD space recommended.

4.1.2. 3rd Party Toolsets

The Silicon Laboratories IDE has native support for many 8051 compilers. The full list of natively supported tools is:

- Keil
- IAR
- Raisonance
- Tasking
- Hi-Tech
- SDCC

Please note that the demo applications for the C8051F338 target board are written to work with the Keil and SDCC toolsets.

4.2. Keil Evaluation Toolset

4.2.1. Keil Assembler and Linker

The assembler and linker that are part of the Keil Demonstration Toolset are the same versions that are found in the full Keil Toolset. The complete assembler and linker reference manual can be found on-line under the **Help** menu in the IDE or in the "*SiLabs\MCU\hlp*" directory (A51.pdf).

4.2.2. Keil Evaluation C51 C Compiler

The evaluation version of the C51 compiler is the same as the full version with these limitations: (1) Maximum 4 kB code generation, and (2) Floating point library not included. When installed from the CD-ROM, the C51 compiler is initially limited to a code size of 2 kB, and programs start at code address 0x0800. Please refer to the Application Note "AN104: Integrating Keil Tools into the Silicon Labs IDE" for instructions to change the limitation to 4 kB, and have the programs start at code address 0x0000.

4.3. Configuration Wizard 2

The Configuration Wizard 2 is a code generation tool for all of the Silicon Laboratories devices. Code is generated through the use of dialog boxes for each of the device's peripherals.

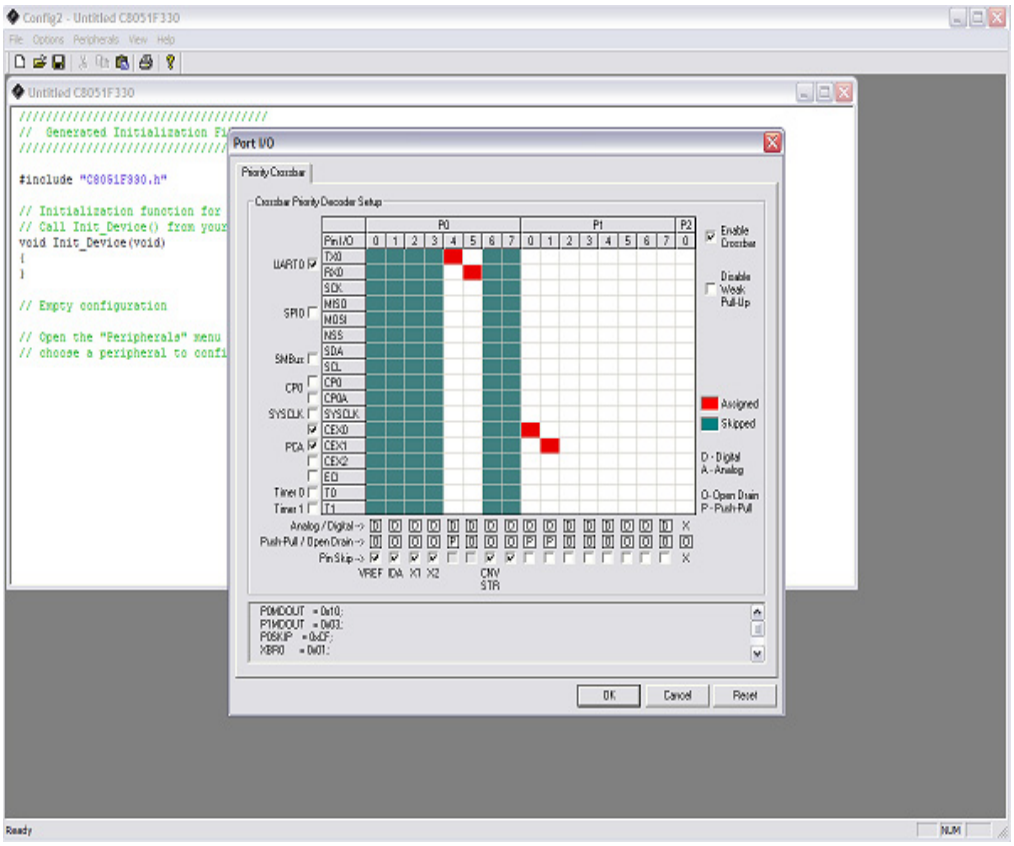

Figure 1. Configuration Wizard 2 Utility

The Configuration Wizard 2 utility helps accelerate development by automatically generating initialization source code to configure and enable the on-chip resources needed by most design projects. In just a few steps, the wizard creates complete startup code for a specific Silicon Laboratories MCU. The program is configurable to provide the output in C or assembly. For more information, please refer to the Configuration Wizard 2 help available under the **Help** menu in Config Wizard 2.

4.4. Keil uVision2 and uVision3 Silicon Laboratories Drivers

As an alternative to the Silicon Laboratories IDE, the uVision debug driver allows the Keil uVision IDE to communicate with Silicon Laboratories on-chip debug logic. In-system Flash memory programming integrated into the driver allows for rapidly updating target code. The uVision IDE can be used to start and stop program execution, set breakpoints, check variables, inspect and modify memory contents, and single-step through programs running on the actual target hardware.

For more information, please refer to the uVision driver documentation. The documentation and software are available from the Downloads webpage (www.silabs.com/mcudownloads).

5. 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.

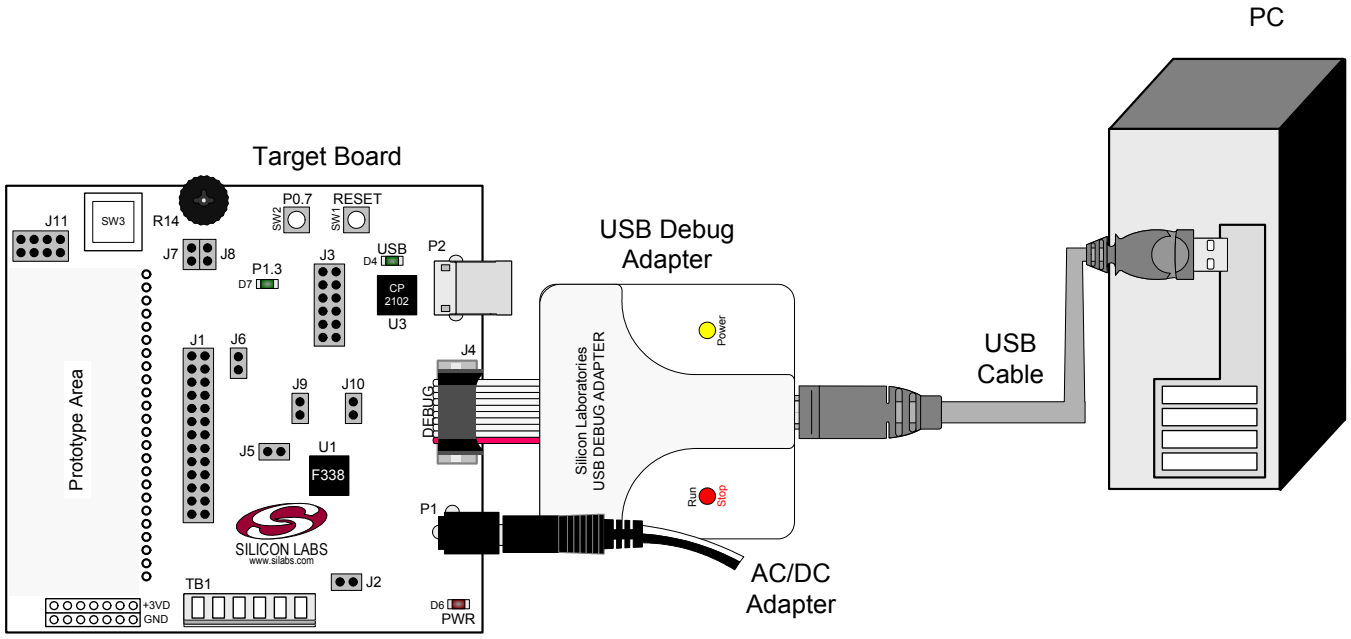

Figure 2. Hardware Setup using a USB Debug Adapter

6. 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 Application Note "AN104: Integrating Keil 8051 Tools into the Silicon Labs IDE" in the "*SiLabs\MCU\Documentation\ApplicationNotes*" directory 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.)

6.1. 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.

6.2. 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. C8051F336/7/8/9 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**.

7. Example Source Code

Example source code and register definition files are provided in the "*SiLabs\MCU\Examples\C8051F336_9*" 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.

7.1. Register Definition Files

Register definition files *C8051F336.inc and C8051F336_defs.h* define all SFR registers and bit-addressable control/status bits. A macro definition header file *compiler_defs.h* is also included, and is required to be able to use the *C8051F336_defs.h* header file with various tool chains. These files are installed into the "*SiLabs\MCU\Examples\C8051F336_9\Header_Files*" directory during IDE installation by default. The register and bit names are identical to those used in the C8051F336/7/8/9 data sheet. These 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.

7.2. Blinking LED Example

The example source files *F336_Blinky.asm* and *F336_Blinky.c* installed in the default directory "*SiLabs\MCU\Examples\C8051F336_9\Blinky*" show examples of several basic C8051F338 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 pin. When compiled/assembled and linked this program flashes the green LED on the C8051F338 target board about five times a second using the interrupt handler with a C8051F338 timer.

7.3. Touch Sensitive Switch Example

The example source file *F338_TouchSense_Switch.c* demonstrates the configuration and usage of the touch sensitive (contactless) switch SW3. Refer to the source file for step-by-step instructions to build and test this example. This is installed in the "*SiLabs\MCU\Examples\C8051F336_9\TouchSense_Switch*" directory by default.

8. Target Board

The C8051F336 Development Kit includes a target board with a **C8051F338** 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. Figure 4 on page 8 shows the factory default shorting block positions. A summary of the signal names and headers is provided in Table 6 on page 13.

P1	Power connector (accepts input from 7 to 15 VDC unregulated power adapter)
P2	USB connector (connects to PC for serial communication)
J1	26-pin Expansion I/O connector
J2	MCU power header (VDD)
J3	Port I/O configuration header
J4	DEBUG connector for Debug Adapter interface
J5	USB connector for UART0 interface
J6	Connects pin P0.1 to node IDAC and resistor R2
J7, J8	Connects the potentiometer (R14) to pin P1.6 and +3VD
J9, J10	External crystal enable connectors
J11	Touch Sense Switch connection header
TB1	Analog I/O terminal block

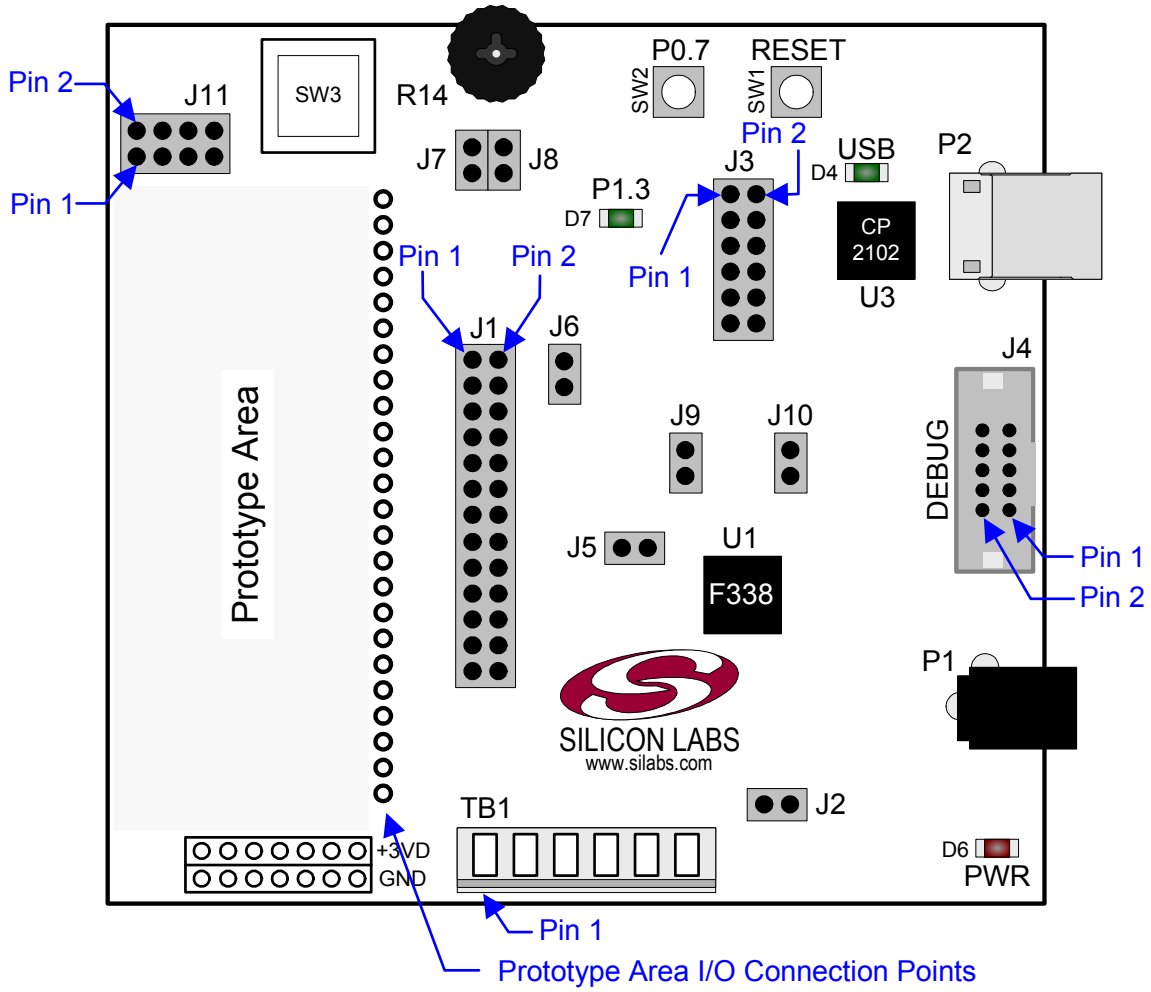

Figure 3. C8051F338 Target Board

8.1. Target Board Shorting Blocks: Factory Defaults

The C8051F338 target board comes from the factory with pre-installed shorting blocks on many headers. Figure 4 shows the positions of the factory default shorting blocks.

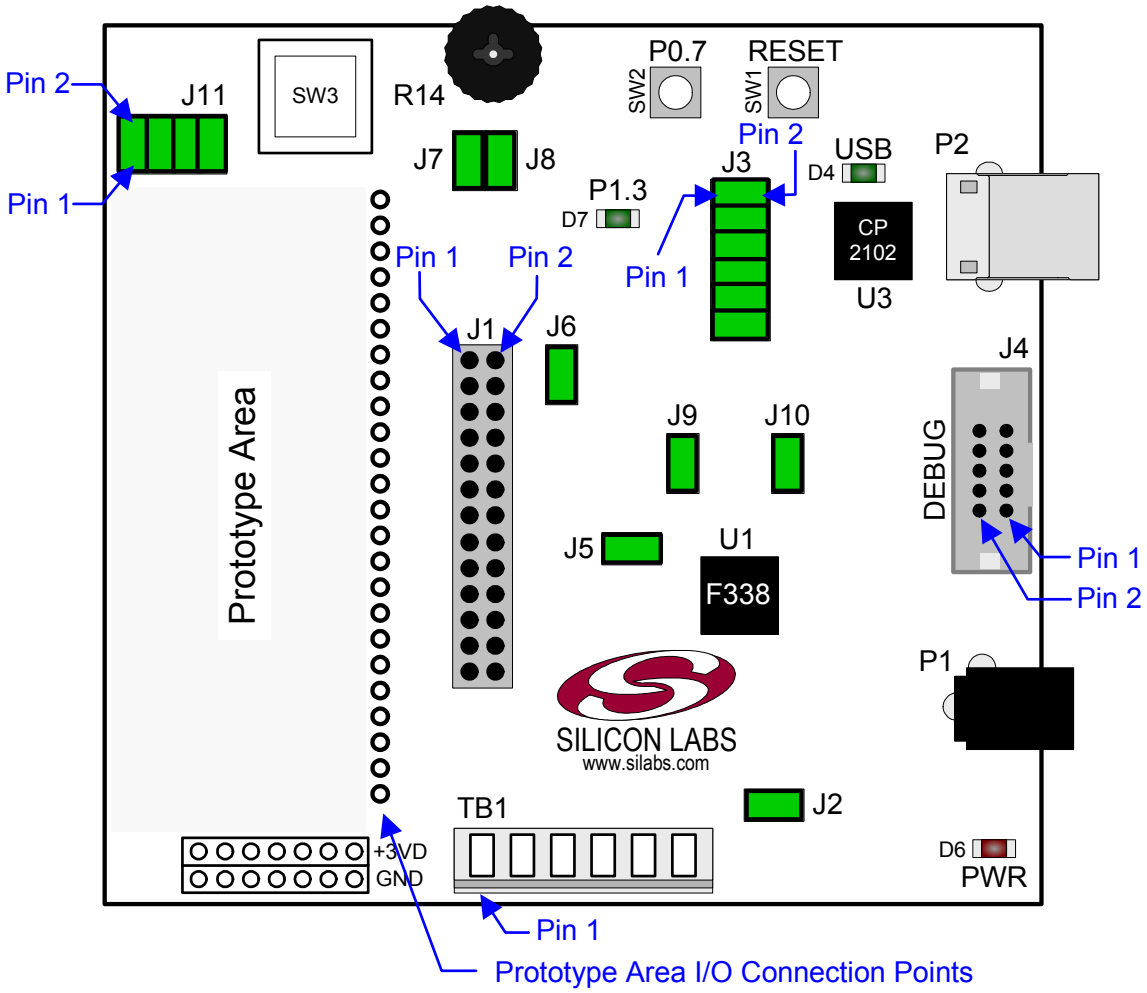

Figure 4. C8051F338 Target Board Shorting Blocks: Factory Defaults

8.2. Target Board Power Options and Current Measurement

The C8051F338 target board supports three power options:

1. 12 VDC power using the AC to DC power adapter (P1)
2. 5 VDC USB VBUS power from PC via the USB Debug Adapter (J4)
3. 5 VDC USB VBUS power from PC via the CP2102 USB connector (P2)

All the three power sources are ORed together using reverse-biased diodes (D1, D2, D3), eliminating the need for headers to choose between the sources. The target board will operate as long as any one of the power sources is present. The ORed power is regulated to a 3.3V DC voltage using a LDO regulator (U2). The output of the regulator powers the +3VD net on the target board, and is also connected to one end of the header J2. A shorting block should be installed on J2 to power the VDD net, which powers the C8051F338 MCU. With the shorting block removed, a multimeter can be used across J2 to measure the current consumption of the MCU.

8.3. System Clock Sources

8.3.1. Internal Oscillators

The C8051F338 device installed on the target board features a factory calibrated programmable high-frequency internal oscillator (24.5 MHz base frequency, ±2%), which is enabled as the system clock source on reset. After reset, the internal oscillator operates at a frequency of 3.0625 MHz (±2%) by default but may be configured by software to operate at other frequencies. For low-frequency operation, the C8051F338 features a programmable low-frequency internal oscillator (80 kHz base frequency, ±10%). Therefore, in many applications an external oscillator is not required. However, if you wish to operate the C8051F338 device at a frequency not available with the internal oscillator, an external crystal may be used. Refer to the C8051F336/7/8/9 data sheet for more information on configuring the system clock source.

8.3.2. External Oscillator Options

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 MΩ resistor at R9 and install capacitors at C13 and C14 using values appropriate for the crystal you select. If you wish to operate the external oscillator in capacitor or RC mode, options to install a capacitor or an RC network are also available on the target board. Populate C14 for capacitor mode, and populate R7 and C14 for RC mode. Refer to the C8051F336/7/8/9 data sheet for more information on the use of external oscillators.

8.4. Switches and LEDs

Two push-button switches are provided on the target board. Switch SW1 is connected to the RESET pin of the C8051F338. Pressing SW1 puts the device into its hardware-reset state. Switch SW2 is connected to the C8051F338'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.

One touch sensitive (contactless) switch is provided on the target board. The operation of this switch needs appropriate code running on the C8051F338 MCU that can sense the state of the switch. All the four shorting blocks should be present on header J11 to properly connect this switch to the MCU. See Section 7.3. "Touch Sensitive Switch Example," on page 6 for details about example source code.

Three LEDs are provided on the target board to serve as indicators. The red LED labeled PWR is used to indicate the presence of power to the target board. The green LED labeled with port pin name P1.3 is connected to the C8051F338's GPIO pin P1.3 through the header J3. 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. Another red LED labeled USB is used to indicate a valid USB connection via the connector P2. See Table 1 for the port pins and headers corresponding to each LED.

Table 1. Target Board I/O Descriptions

Description	I/O	Header(s)
SW1	Reset	none
SW2	P0.7	J3[3–4]
SW3 (Touch Sense Switch)	P1.0, P2.0, P2.1, P2.3	J11
Green LED (P1.3)	P1.3	J3[1–2]
Red LED (PWR)	Power	none
Red LED (USB ACTIVE)	USB Active	none

8.5. Expansion I/O Connector (J1)

The 26-pin Expansion I/O connector J1 provides access to all signal pins of the C8051F338 device. Pins for +3 V, 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 3 on page 7). 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	Pin #	Description
1	+3 VD (+3.3 VDC)	14	P1.3
2	IDAC	15	P1.4
3	P0.0	16	P1.5
4	P0.1	17	P1.6
5	P0.2	18	P1.7
6	P0.3	19	P2.0
7	P0.4	20	P2.1
8	P0.5	21	P2.2
9	P0.6	22	P2.3
10	P0.7	23	P2.4
11	P1.0	24	/RST
12	P1.1	25	GND
13	P1.2	26	GND

8.6. Target Board DEBUG Interface (J4)

The DEBUG connector J4 provides access to the DEBUG (C2) pins of the C8051F338. 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.

Table 3. DEBUG Connector Pin Descriptions

Pin #	Description
1	+3 VD (+3.3 VDC)
2, 3, 9	GND (Ground)
4	P2.4/C2D
5	/RST (Reset)
6	P2.4
7	/RST/C2CK
8	Not Connected
10	USB Power (+5VDC from J4)

8.7. Serial Interface (J3)

A USB-to-UART bridge circuit (U3) and USB connector (P2) are provided on the target board to facilitate serial connections to UART0 of the C8051F338. The Silicon Labs CP2102 (U3) USB-to-UART bridge provides data connectivity between the C8051F338 and the PC via a USB port. The TX, RX, RTS and CTS signals of UART0 may be connected to the CP2102 by installing shorting blocks on header J3. The shorting block positions for connecting each of these signals to the CP2102 are listed in Table 4. To use this interface, the USB-to-UART device drivers should be installed as described in Section 3.2. "CP210x USB to UART VCP Driver Installation," on page 2.

Table 4. Serial Interface Header (J3) Description

Header Pins	UART0 Pin Description
J3[5–6]	TX_MCU (P0.4)
J3[7–8]	RX_MCU (P0.5)
J3[9–10]	RTS (P1.1)
J3[11–12]	CTS (P1.2)

8.8. Analog I/O (TB1)

Several of the C8051F338 target device's port pins are connected to the TB1 terminal block. Refer to Table 5 for the TB1 terminal block connections.

Table 5. TB1 Terminal Block Pin Descriptions

Pin #	Description
1	P0.6 / CNVSTR
2	IDAC
3	P1.5
4	P1.4
5	GND (Ground)
6	P0.0 / Vref (Voltage Reference)

8.9. IDAC Connector (J6)

The C8051F338 target board also features a Current-to-Voltage 1 KΩ load resistor that may be connected to the 10-bit current-mode Digital-to-Analog Converter (IDAC) output that can be enabled on port pin (P0.1). Install a shorting block on J6 to connect port pin P0.1 of the target device to the load resistor. If enabled by software, the IDAC signal is then routed to the J1[2] and TB1[2] connectors.

8.10. C2 Pin Sharing

On the C8051F338, the debug pins C2CK and C2D are shared with the pins /RST and P2.4 respectively. The target board includes the resistors necessary to enable pin sharing which allow the /RST and P2.4 pins to be used normally while simultaneously debugging the device. See Application Note "AN124: Pin Sharing Techniques for the C2 Interface" at www.silabs.com for more information regarding pin sharing.

8.11. Target Board Pin Assignment Summary

Some GPIO pins of the C8051F338 MCU can have an alternate fixed function. For example, pin 22 on the 'F338 MCU is designated P0.4, and can be used as a GPIO pin. Also, if the UART0 peripheral on the MCU is enabled using the crossbar registers, the TX signal is routed to this pin. This is shown in the "Alternate Fixed Function" column. The "Target Board Function" column shows that this pin is used as TX on the 'F338 Target Board. The "Relevant Headers" column shows that this signal is routed to pin 7 of the J1 header and pin 5 of the J3 header. More details can be found in the C8051F336/7/8/9 data sheet. Some of the GPIO pins of the C8051F338 have been used for various functions on the target board. Table 6 summarizes the MCU pin assignments on the target board, and also shows the various headers associated with each signal.

Table 6. C8051F338 Target Board Pin Assignments and Headers

MCU Pin Name	Pin#	Primary Function	Alternate Fixed Function	Target Board Function	Relevant Headers
P0.0	2	P0.0	VREF	VREF	J1[3], TB1[6]*, J5[1]
P0.1	1	P0.1	IDAC	IDAC	J1[4], J1[2]*, TB1[2], J6[1]
P0.2	24	P0.2	XTAL1	XTAL1	J1[5]*, J9[2]
P0.3	23	P0.3	XTAL2	XTAL2	J1[6]*, J10[2]
P0.4	22	P0.4	TX_MCU	TX_MCU	J1[7], J3[5]
P0.5	21	P0.5	RX_MCU	RX_MCU	J1[8], J3[7]
P0.6	20	P0.6	CNVSTR	CNVSTR	J1[9], TB1[1]
P0.7	19	P0.7		SW2 (switch)	J1[10], J3[3]
P1.0	18	P1.0		CP0A	J1[11], J11[1]
P1.1	17	P1.1		RTS	J1[12], J3[9]
P1.2	16	P1.2		CTS	J1[13], J3[11]
P1.3	15	P1.3		LED	J1[14], J3[1]
P1.4	14	P1.4		AIN	J1[15], TB1[4]
P1.5	13	P1.5		AIN	J1[16], TB1[3]
P1.6	12	P1.6		POT	J1[17], J8[1]
P1.7	11	P1.7		GPIO	J1[18]
P2.0	10	P2.0		CP0+	J1[19], J11[3]
P2.1	9	P2.1		CP0-	J1[20], J11[5]
P2.2	8	P2.2		GPIO	J1[21]
P2.3	7	P2.3		T0	J1[22], J11[7]
P2.4/C2D	6	P2.4	C2D	P2.4/C2D	J1[23]*, J4[6]*, J4[4]
/RST/C2CK	5	/RST	C2CK	/RST/C2CK	J1[24]*, J4[5]*, J4[7]
VDD	4	VDD		VDD	J2[2]
GND	3	GND		GND	J1[25], J1[26], TB1[5]

***Note:** Headers denoted by this symbol are not directly connected to the MCU pin; the connection might be via one or more headers and/or pin-sharing resistor(s). See board schematic for details.

9. Schematics

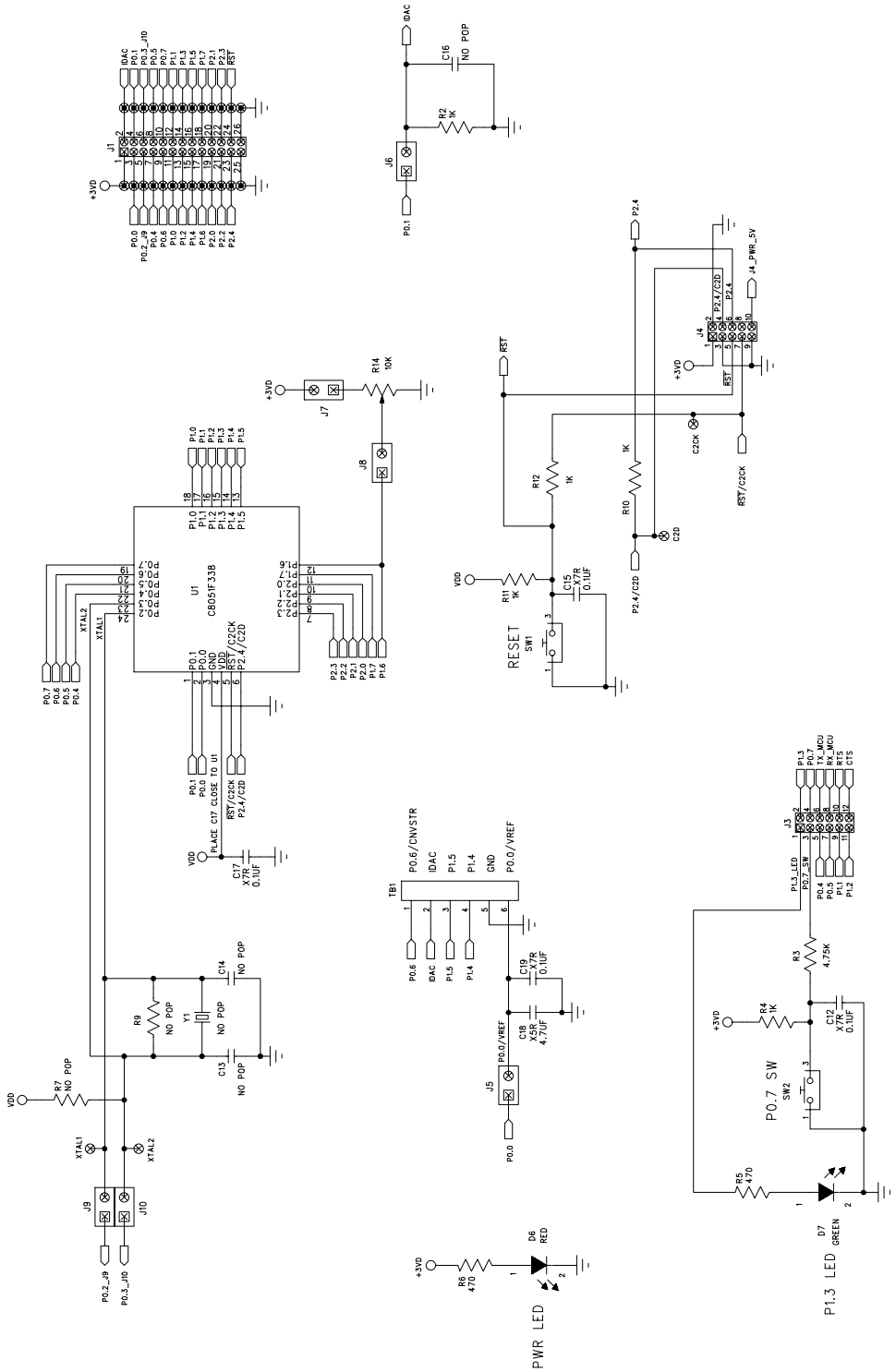

Figure 5. C8051F338 Target Board Schematic (Page 1 of 2)

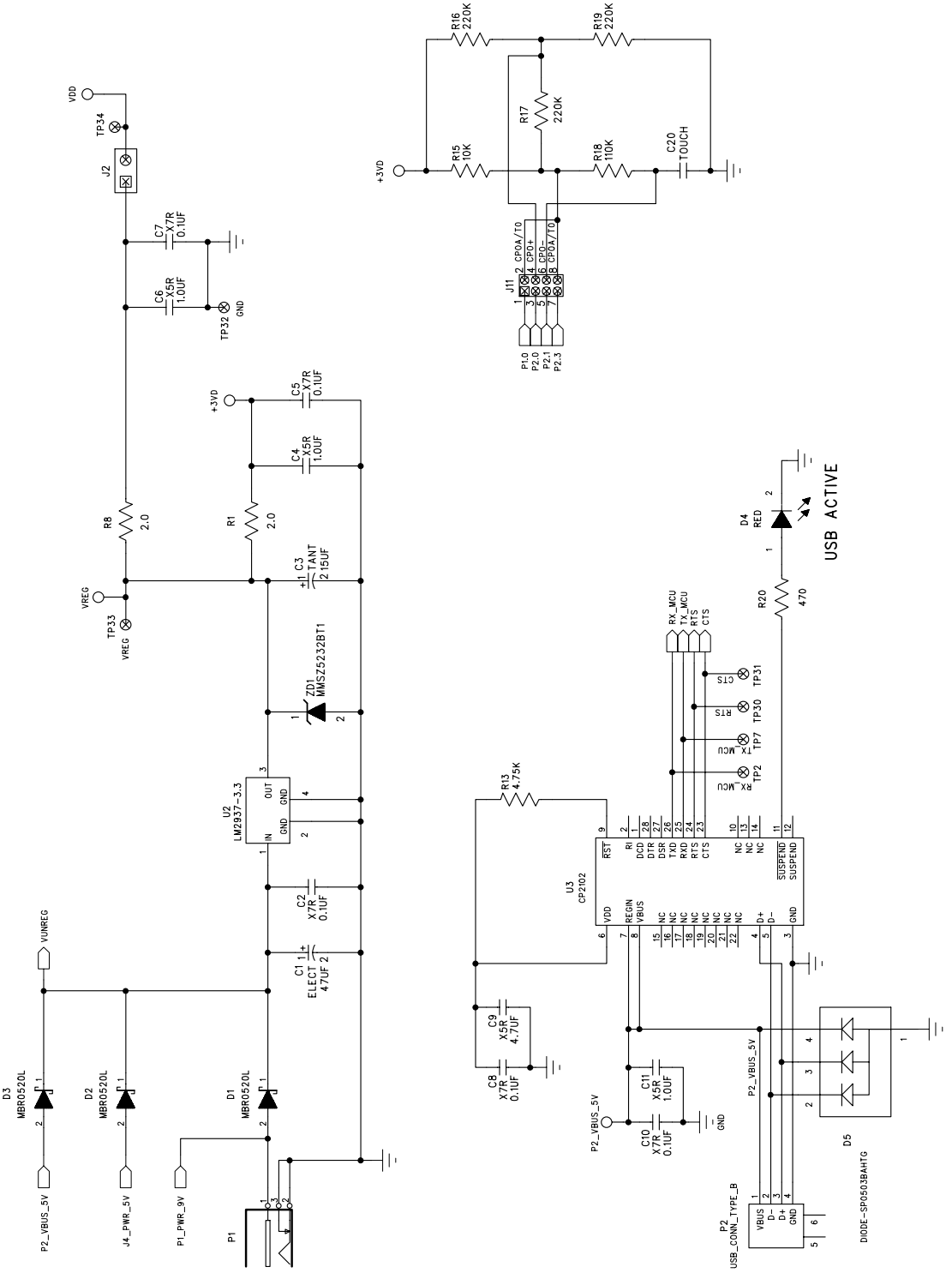

Figure 6. C8051F338 Target Board Schematic (Page 2 of 2)

CONTACT INFORMATION

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

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