

C8051F336 DEVELOPMENT KIT USER'S GUIDE

1. Relevant Devices

The C8051F338 Development Kit is intended as a development platform for the microcontrollers in the C8051F336/7/8/9 MCU family.

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
- AC to DC Power Adapter
- USB Debug Adapter (USB to Debug Interface)
- USB Cable
- CD-ROM

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 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.

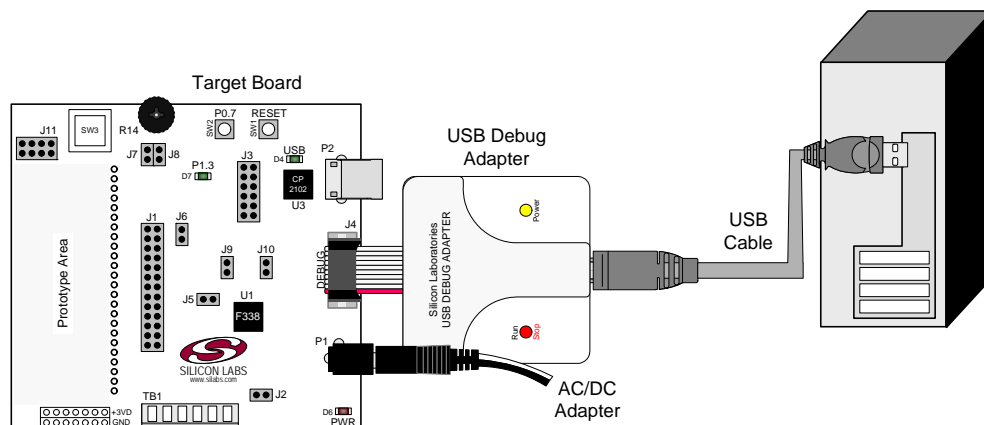


Figure 1. Hardware Setup Using a USB Debug Adapter

4. Software Setup

Simplicity Studio greatly reduces development time and complexity with Silicon Labs EFM32 and 8051 MCU products by providing a high-powered IDE, tools for hardware configuration, and links to helpful resources, all in one place.

Once Simplicity Studio is installed, the application itself can be used to install additional software and documentation components to aid in the development and evaluation process.

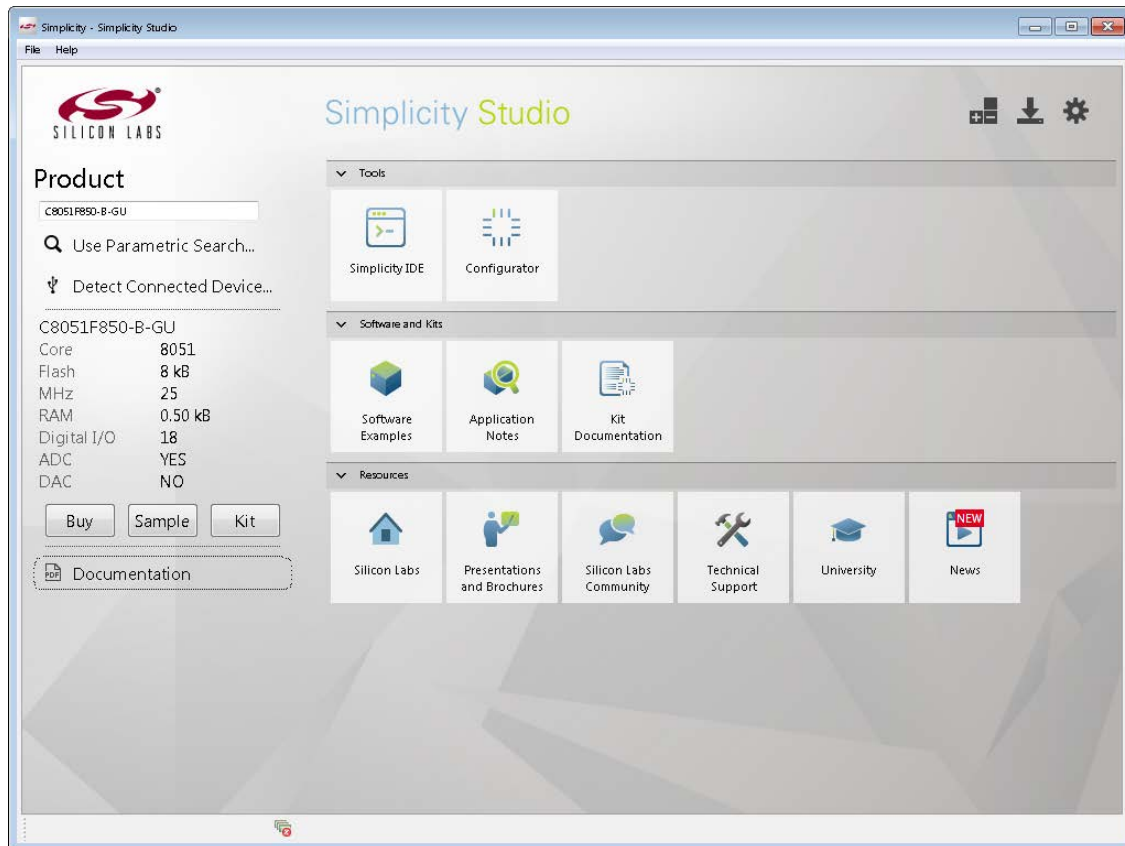


Figure 2. Simplicity Studio

The following Simplicity Studio components are required for the C8051F338 Development Kit:

- 8051 Products Part Support
- Simplicity Developer Platform

Download and install Simplicity Studio from www.silabs.com/8bit-software or www.silabs.com/simplicity-studio. Once installed, run Simplicity Studio by selecting **Start**→**Silicon Labs**→**Simplicity Studio**→**Simplicity Studio** from the start menu or clicking the **Simplicity Studio** shortcut on the desktop. Follow the instructions to install the software and click **Simplicity IDE** to launch the IDE.

The first time the project creation wizard runs, the **Setup Environment** wizard will guide the user through the process of configuring the build tools and SDK selection.

In the **Part Selection** step of the wizard, select from the list of installed parts only the parts to use during development. Choosing parts and families in this step affects the displayed or filtered parts in the later device selection menus. Choose the C8051F33x family by checking the **C8051F33x** check box. Modify the part selection at any time by accessing the **Part Management** dialog from the **Window**→**Preferences**→**Simplicity Studio**→**Part Management** menu item.

Simplicity Studio can detect if certain toolchains are not activated. If the **Licensing Helper** is displayed after completing the **Setup Environment** wizard, follow the instructions to activate the toolchain.

4.1. Running Blinky

Each project has its own source files, target configuration, SDK configuration, and build configurations such as the **Debug** and **Release** build configurations. The IDE can be used to manage multiple projects in a collection called a workspace. Workspace settings are applied globally to all projects within the workspace. This can include settings such as key bindings, window preferences, and code style and formatting options. Project actions, such as build and debug are context sensitive. For example, the user must select a project in the **Project Explorer** view in order to build that project.

To create a project based on the Blinky example:

1. Click the **Simplicity IDE** tile from the Simplicity Studio home screen.
2. Click the **Create new project** link from the welcome screen or go to **File→New→Silicon Labs MCU Project**.
3. In the **Kit** drop-down, select **C8051F338 Development Kit**, in the **Part** drop-down, select **C8051F338**, and in the **SDK** drop-down, select the desired SDK. Click **Next**.
4. Select **Example** and click **Next**.
5. Under **C8051F338 Development Kit** in the **Blinky** folder, select **F336-9 Blinky** and click **Finish**.
6. Click on the project in the **Project Explorer** and click **Build**, the hammer icon in the top bar. Alternatively, go to **Project→Build Project**.
7. Click **Debug** to download the project to the hardware and start a debug session.
8. Press the **Resume** button to start the code running. The LED should blink.



9. Press the **Suspend** button to stop the code.



10. Press the **Reset the device** button to reset the target MCU.



11. Press the **Disconnect** button to return to the development perspective.



4.2. Simplicity Studio Help

Simplicity Studio includes detailed help information and device documentation within the tool. The help contains descriptions for each dialog window. To view the documentation for a dialog, click the question mark icon in the window:



This will open a pane specific to the dialog with additional details.

The documentation within the tool can also be viewed by going to **Help→Help Contents** or **Help→Search**.

4.3. Legacy 8-bit IDE

Note: Using the Simplicity Studio tools with the C8051F338 Development Kit is recommended. See section 4. "Software Setup," on page 2 for more information.

Download the 8-bit software from the website (www.silabs.com/8bit-software) or use the provided installer on the CD-ROM to install the software tools for the C8051F336/7/8/9 devices. After installation, examples can be found in ...\\Examples\\C8051F336_9 in the installation directory. At a minimum, the C8051F338 DK requires:

- **Silicon Labs IDE**—Software enabling initial evaluation, development, and debugging.
- **Configuration Wizard 2**—Initialization code generation software for the C8051F336/7/8/9 devices.
- **Keil C51 Tools**—Keil 8051 Compiler/Assembler/Linker toolchain.
- **CP210x Drivers**—Virtual COM Port (VCP) drivers for the CP210x COM interface. More information on this installation process can be found in Section 4.4.

Other software available includes:

- **Keil μ Vision Driver**—Driver for the Keil μ Vision IDE that enables development and debugging on C8051Fxxx MCUs.
- **Flash Programming Utilities and MCU Production Programmer**—Programming utilities for the production line. More information on the available programming options can be found on the website: <http://www.silabs.com/products/mcu/Pages/ProgrammingOptions.aspx>.
- **ToolStick Development Tools**—Software and examples for the ToolStick development platform. More information on this platform can be found at www.silabs.com/toolstick.

The development kit includes the latest version of the C51 Keil 8051 toolset. This toolset is initially limited to a code size of 2 kB and programs start at code address 0x0800. After registration, the code size limit is removed entirely and programs will start at code address 0x0000.

To register the Keil toolset:

1. Find the **Product Serial Number** printed on the CD-ROM. If you no longer have this serial number, register on the Silicon Labs website (www.silabs.com/8bit-software) to obtain the serial number.
2. Open the Keil μ Vision4 IDE from the installation directory with administrative privileges.
3. Select **File**→**License Management** to open the License Management window.

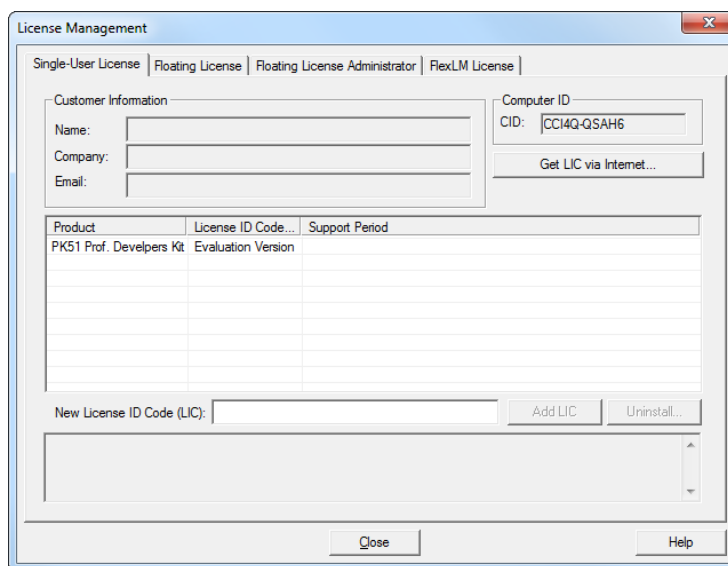


Figure 3. Keil μ Vision4 IDE License Management Window

4. Click on the **Get LIC via Internet...** button to open the Obtaining a License IDE Code (LIC) window.
5. Press **OK** to open a browser window to the Keil website. If the window doesn't open, navigate to

www.keil.com/license/install.htm.

6. Enter the Silicon Labs **Product Serial Number** printed on the CD-ROM, along with any additional required information.
7. Once the form is complete, click the **Submit** button. An email will be sent to the provided email address with the license activation code.
8. Copy the License ID Code (LIC) from the email.
9. Paste the LIC into the **New License ID Code (LIC)** text box at the bottom of the License Management window in μ Vision4.
10. Press the **Add LIC** button. The window should now list the **PK51 Prof. Developers Kit for Silabs** as a licensed product.
11. Click the **Close** button.

4.4. CP210x USB to UART VCP Driver Installation

The MCU Card includes a Silicon Labs CP210x USB-to-UART Bridge Controller. Device drivers for the CP210x need to be installed before the PC software can communicate with the MCU through the UART interface. Use the drivers included CD-ROM or download the latest drivers from the website (www.silabs.com/interface-software).

1. If using the CD-ROM, the **CP210x Drivers** option will launch the appropriate driver installer. If downloading the driver package from the website, unzip the files to a location and run the appropriate installer for the system (x86 or x64).
2. Accept the license agreement and follow the steps to install the driver on the system. The installer will let you know when your system is up to date. The driver files included in this installation have been certified by Microsoft.
3. To complete the installation process, connect the included USB cable between the host computer and the **COM PORT** USB connector (J5) on the MCU Card. Windows will automatically finish the driver installation. Information windows will pop up from the taskbar to show the installation progress.
4. If needed, the driver files can be uninstalled by selecting **Windows Driver Package—Silicon Laboratories...** option in the **Programs and Features** window.

5. 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 4 for the locations of the various I/O connectors. Figure 5 on page 7 shows the factory default shorting block positions. A summary of the signal names and headers is provided in Table 6 on page 12.

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

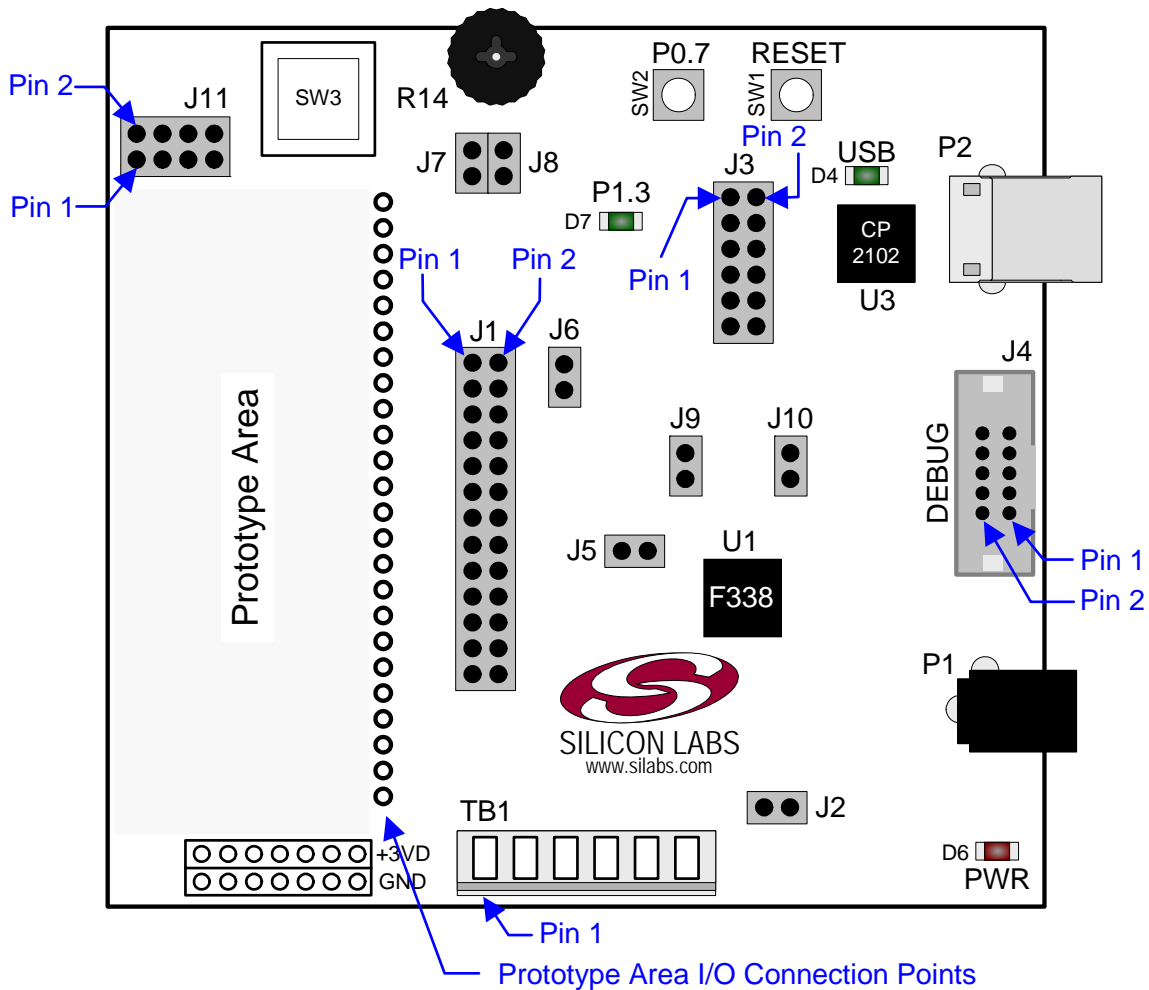


Figure 4. C8051F338 Target Board

5.1. Target Board Shorting Blocks: Factory Defaults

The C8051F338 target board comes from the factory with pre-installed shorting blocks on many headers. Figure 5 shows the positions of the factory default shorting blocks.

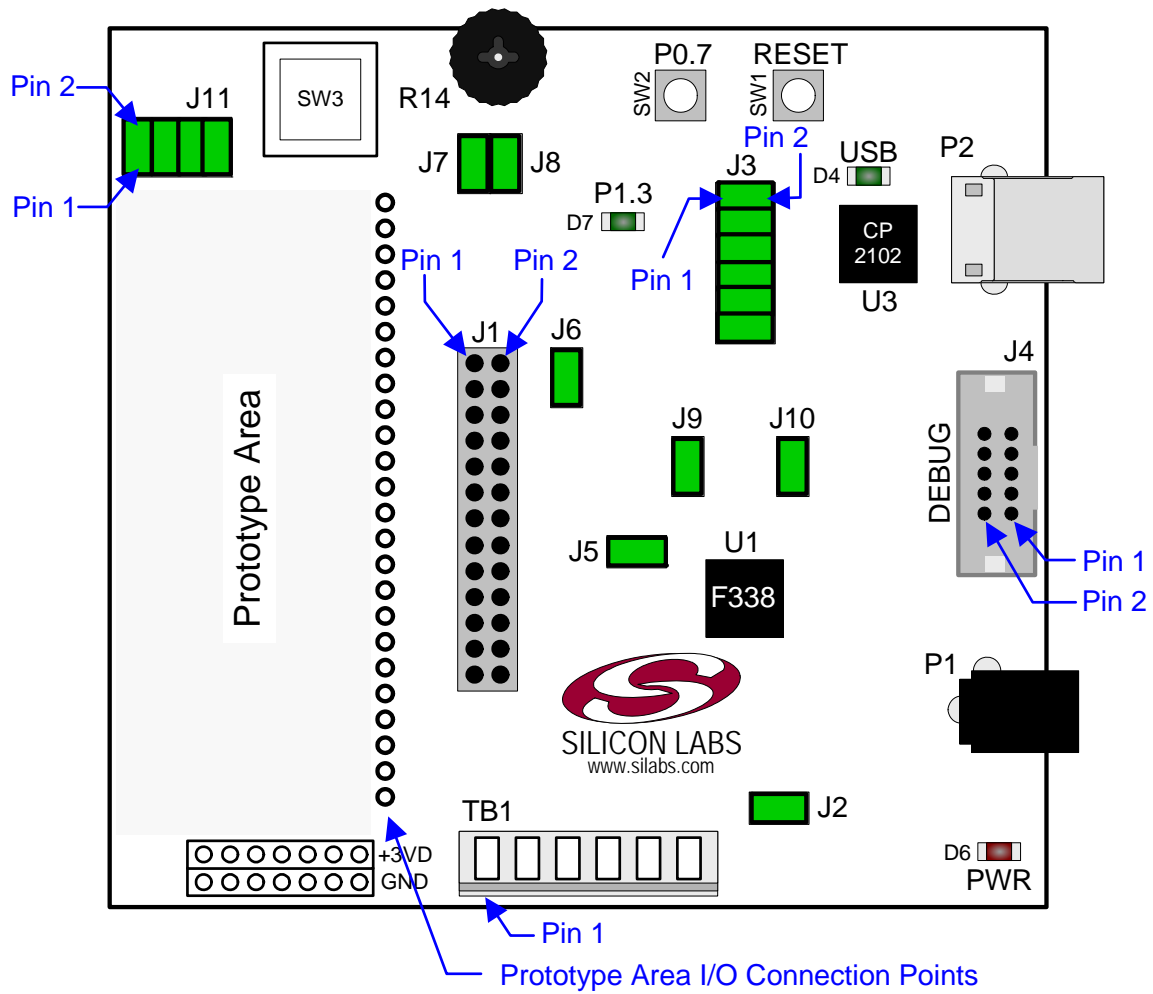


Figure 5. C8051F338 Target Board Shorting Blocks: Factory Defaults

5.2. 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.

5.3. System Clock Sources

5.3.1. Internal Oscillators

The C8051F338 device installed on the target board features a factory calibrated programmable high-frequency internal oscillator (24.5 MHz base frequency, $\pm 2\%$), 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. For low-frequency operation, the C8051F338 features a programmable low-frequency internal oscillator (80 kHz base frequency, $\pm 10\%$). 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.

5.3.2. 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 M Ω 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.

5.4. 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.

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

5.5. 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 4 on page 6). 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

5.6. 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)

5.7. 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 4.4. "CP210x USB to UART VCP Driver Installation," on page 5.

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)

5.8. 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)

5.9. IDAC Connector (J6)

The C8051F338 target board also features a Current-to-Voltage 1 K Ω 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.

5.10. 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.

C8051F336/7/8/9

5.11. 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.

6. Schematics

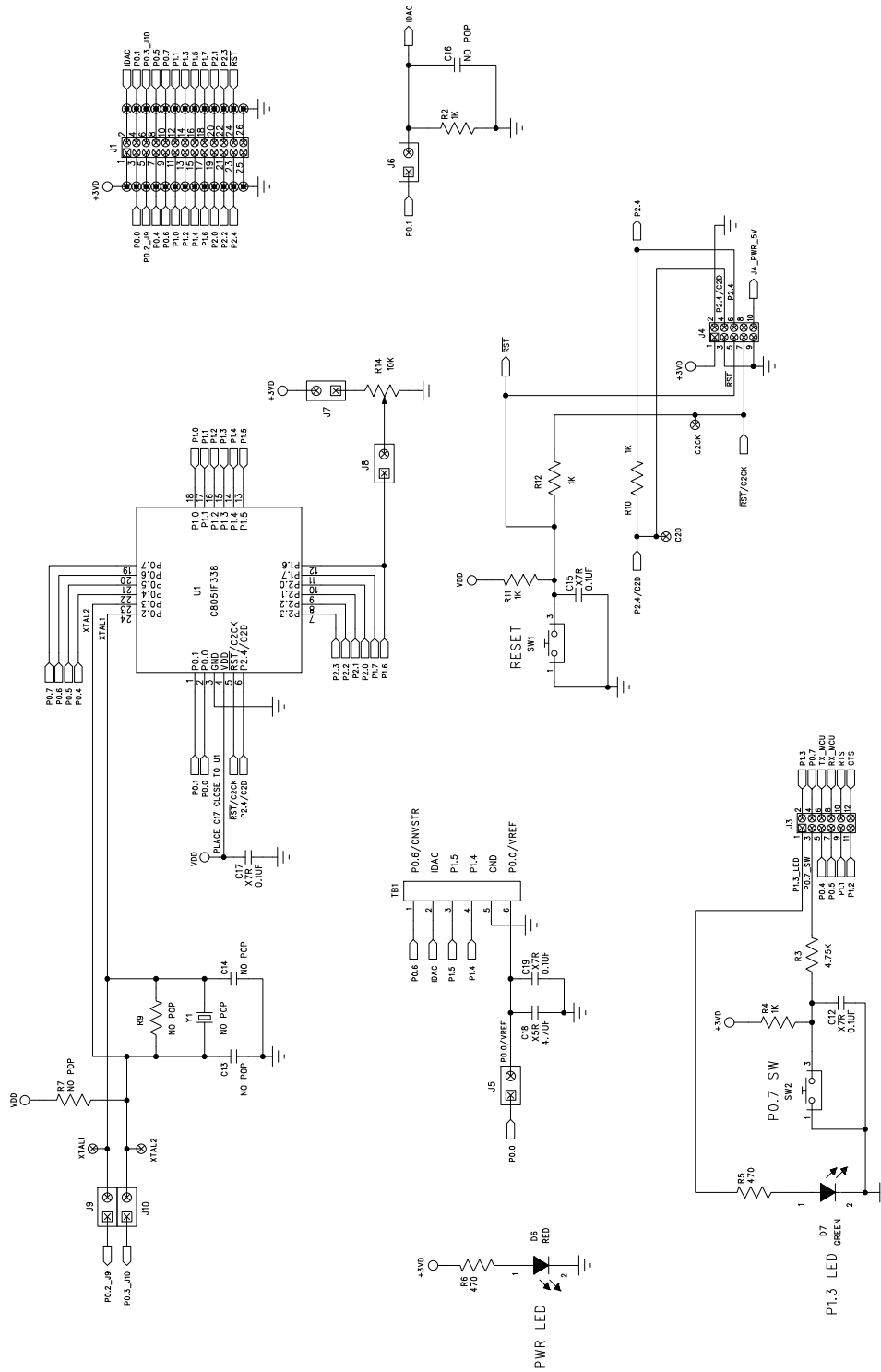


Figure 6. C8051F338 Target Board Schematic (Page 1 of 2)

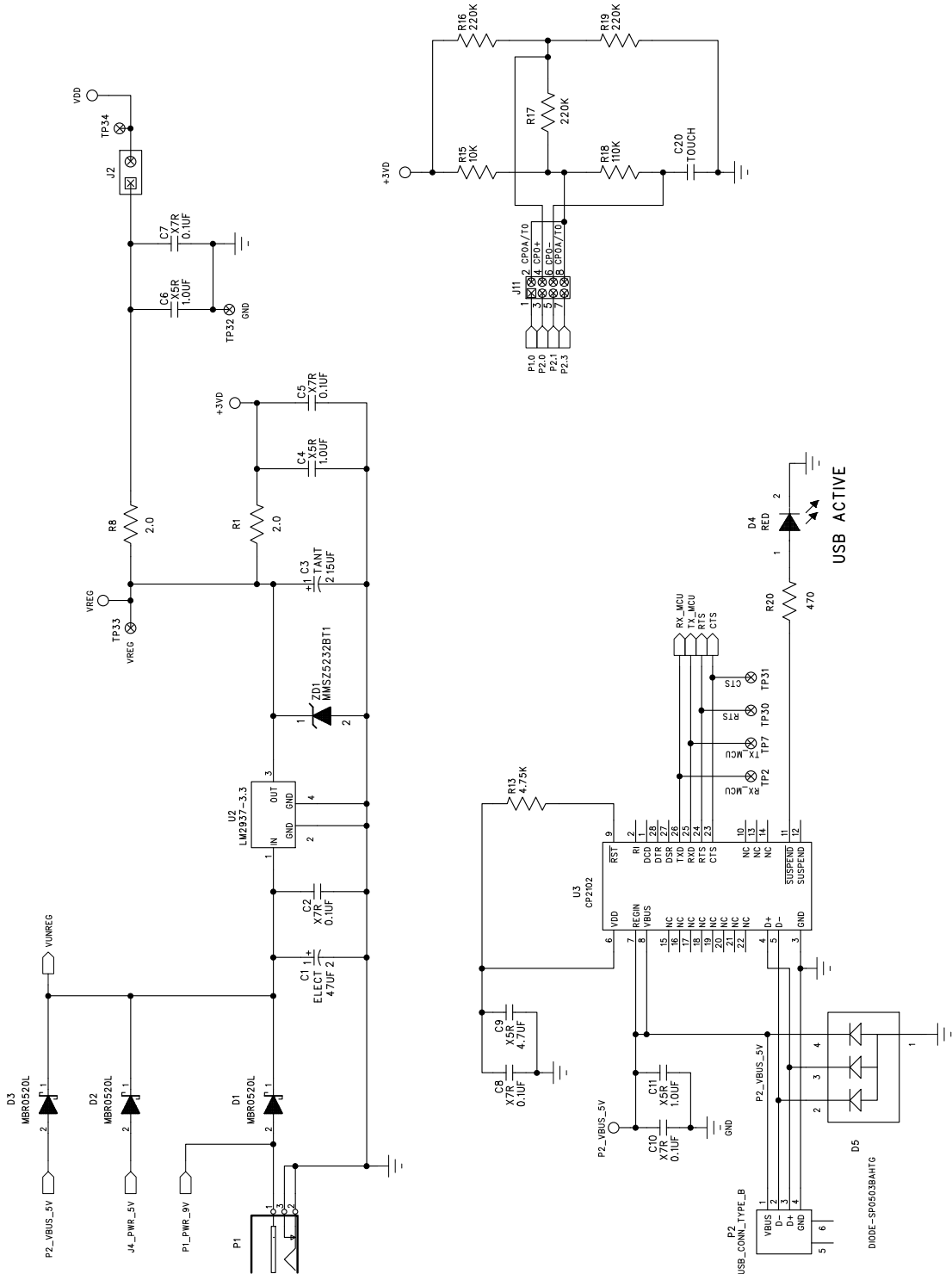


Figure 7. C8051F338 Target Board Schematic (Page 2 of 2)