

---

## C8051F800 DEVELOPMENT KIT USER'S GUIDE

---

### 1. Relevant Devices

The C8051F800 Development Kit is intended as a development platform for the microcontrollers in the C8051F80x-83x MCU family. The members of this MCU family are as follows: C8051F800, C8051F801, C8051F802, C8051F803, C8051F804, C8051F805, C8051F806, C8051F807, C8051F808, C8051F809, C8051F810, C8051F811, C8051F812, C8051F813, C8051F814, C8051F815, C8051F816, C8051F817, C8051F818, C8051F819, C8051F820, C8051F821, C8051F822, C8051F823, C8051F824, C8051F825, C8051F826, C8051F827, C8051F828, C8051F829, C8051F830, C8051F831, C8051F832, C8051F833, C8051F834, and C8051F835.

- The target board included in this kit is provided with a pre-soldered C8051F800-GM MCU (QFN20 package).
- Code developed on the C8051F800 can be easily ported to the other members of this MCU family.
- Refer to the C8051F80x-83x data sheet for the differences between the members of this MCU family.
- The C8051F80x-GM and C8051F81x-GM devices (QFN20 package) are pin-compatible with the C8051F330/1/2/3/4/5/6/7 devices.

### 2. Kit Contents

The C8051F800 Development Kit contains the following items:

- C8051F800 Target Board
- C8051Fxxx Development Kit Quick-Start Guide
- Silicon Labs IDE and Product Information CD-ROM. CD content includes the following:
  - Silicon Labs Integrated Development Environment (IDE)
  - Evaluation assembler, compiler, and linker tools
  - Source code examples and register definition files
  - Documentation
- Optional Third Party Tools CD
- AC to DC Power Adapter
- USB Debug Adapter
- Two USB Cables

### 3. Hardware Setup

Refer to Figure 1 for a diagram of the hardware configuration.

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. Verify that shorting blocks are installed on the target board as shown in Figure 5 on page 10.
4. Connect the other end of the USB cable to a USB Port on the PC.
5. Connect the ac/dc power adapter to power jack P4 on the target board.

#### Notes:

- Use the **Reset** icon in the IDE to reset the target when connected during a debug session.
- 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.



## 4.3. CP210x USB to UART VCP Driver Installation

The C8051F800 Target Board includes a Silicon Labs CP2103 USB-to-UART Bridge Controller. Device drivers for the CP2103 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 is selected during installation, a driver "unpacker" utility will launch.

1. Follow the steps to copy the driver files to the desired location. The default directory is *C:\SiLabsMCU\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:\SiLabsMCU\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 (P5) on the C8051F800 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 Labs CP210x USB to UART Bridge Driver Removal" option in the "Add or Remove Programs" window.

## 5. Software Overview

The following software is necessary to build a project, download code to, and communicate with the target microcontroller.

- Silicon Labs Integrated Development Environment (IDE)
- 8051 Toolset

Other useful software that is provided on the development kit CD and the Silicon Labs Downloads ([www.silabs.com/mcudownloads](http://www.silabs.com/mcudownloads)) website includes:

- Configuration Wizard 2
- Keil  $\mu$ Vision2 and  $\mu$ Vision3 Drivers
- MCU Production Programmer and Flash Programming Utilities
- QuickSense™ Studio (available at [www.silabs.com/quicksense](http://www.silabs.com/quicksense))

### 5.1. Silicon Labs IDE

The Silicon Labs IDE integrates a source-code editor, source-level debugger and in-system programmer. The use of third-party compilers, assemblers, and linkers is also supported. This development kit includes evaluation versions of commercial C compilers and assemblers which can be used from within the Silicon Labs IDE.

#### 5.1.1. Third Party Toolsets

The Silicon Labs IDE has native support for many 8051 compilers. Natively-supported tools are as follows:

- Keil
- IAR
- Raisonance
- Tasking
- SDCC

Specific instructions for integrating each of the supported tools can be found in the application notes section of the CD, or on the Silicon Labs website (<http://www.silabs.com>).

# C8051F800-DK

---

## 5.1.2. Getting Started With the Silicon Labs IDE

The following sections discuss how to open an example project in the IDE, build the source code, and download it to the target device.

### 5.1.2.1. Running the F800\_Blinky example program

The **F800\_Blinky** example program blinks an LED on the target board.

1. Open the Silicon Labs IDE from the Start menu.
2. Select **Project**→**Open Project** to open an existing project.
3. Browse to the *C:\SiLabs\MCU\Examples\C8051F80x\_83x\Blinky* directory (default) and select the *F800\_Blinky\_C.wsp* project file and click **Open**.
4. Once the project is open, 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.
5. 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 “USB Debug Adapter” option. Next, the correct “Debug Interface” must be selected. C8051F80x-83x devices use Silicon Labs “C2” 2-wire debug interface. Once all the selections are made, click the **OK** button to close the window.
6. Click the **Connect** button in the toolbar or select **Debug**→**Connect** from the menu to connect to the device.
7. 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.
8. Click on the **Go** button (green circle) in the toolbar or by selecting **Debug**→**Go** from the menu to start running the firmware. The LED on the target board will start blinking.

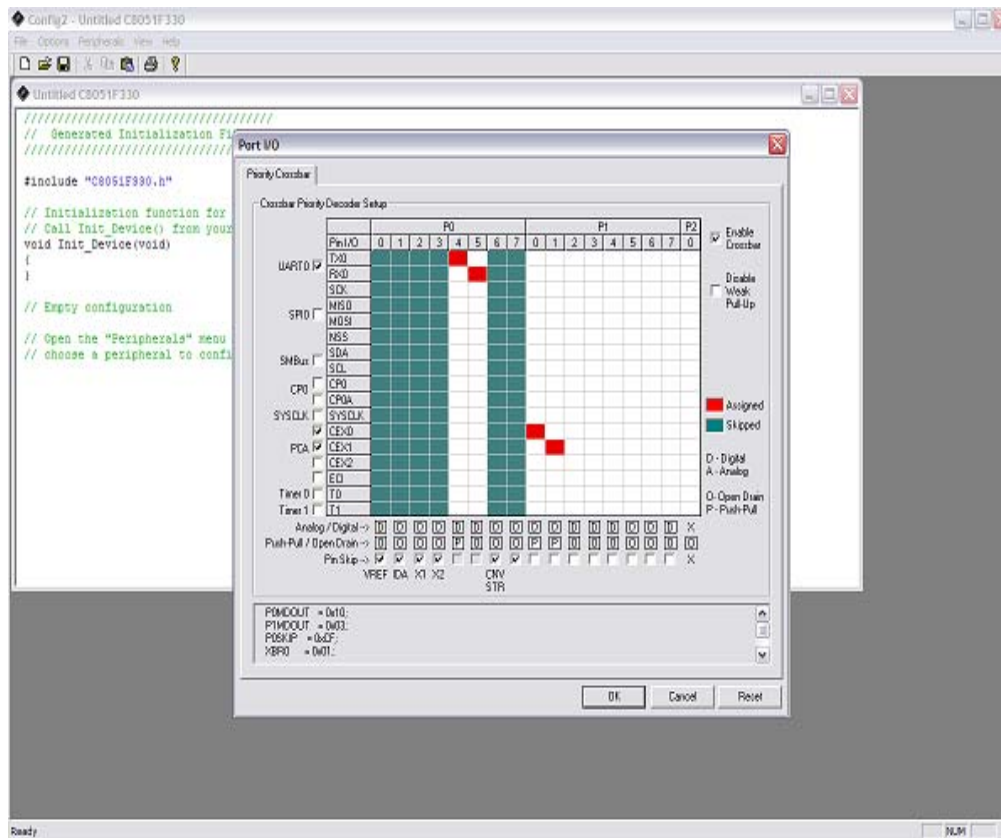
### 5.1.2.2. Creating a New Project

Use the following steps to create a new projects. Once steps 1–5 in this section are complete, continue at Step 3 in Section 5.1.2.1.

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. 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**.

## 5.2. Configuration Wizard 2

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



**Figure 2. 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 Labs 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 Configuration Wizard 2.

For more information, please refer to the Configuration Wizard 2 documentation. The documentation and software are available from the Downloads webpage ([www.silabs.com/mcudownloads](http://www.silabs.com/mcudownloads)).

## 5.3. Keil $\mu$ Vision2 and $\mu$ Vision3 Silicon Labs Drivers

As an alternative to the Silicon Labs IDE, the  $\mu$ Vision debug driver allows the Keil  $\mu$ Vision IDE to communicate with Silicon Labs on-chip debug logic. In-system Flash memory programming integrated into the driver allows for rapidly updating target code. The  $\mu$ Vision 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.

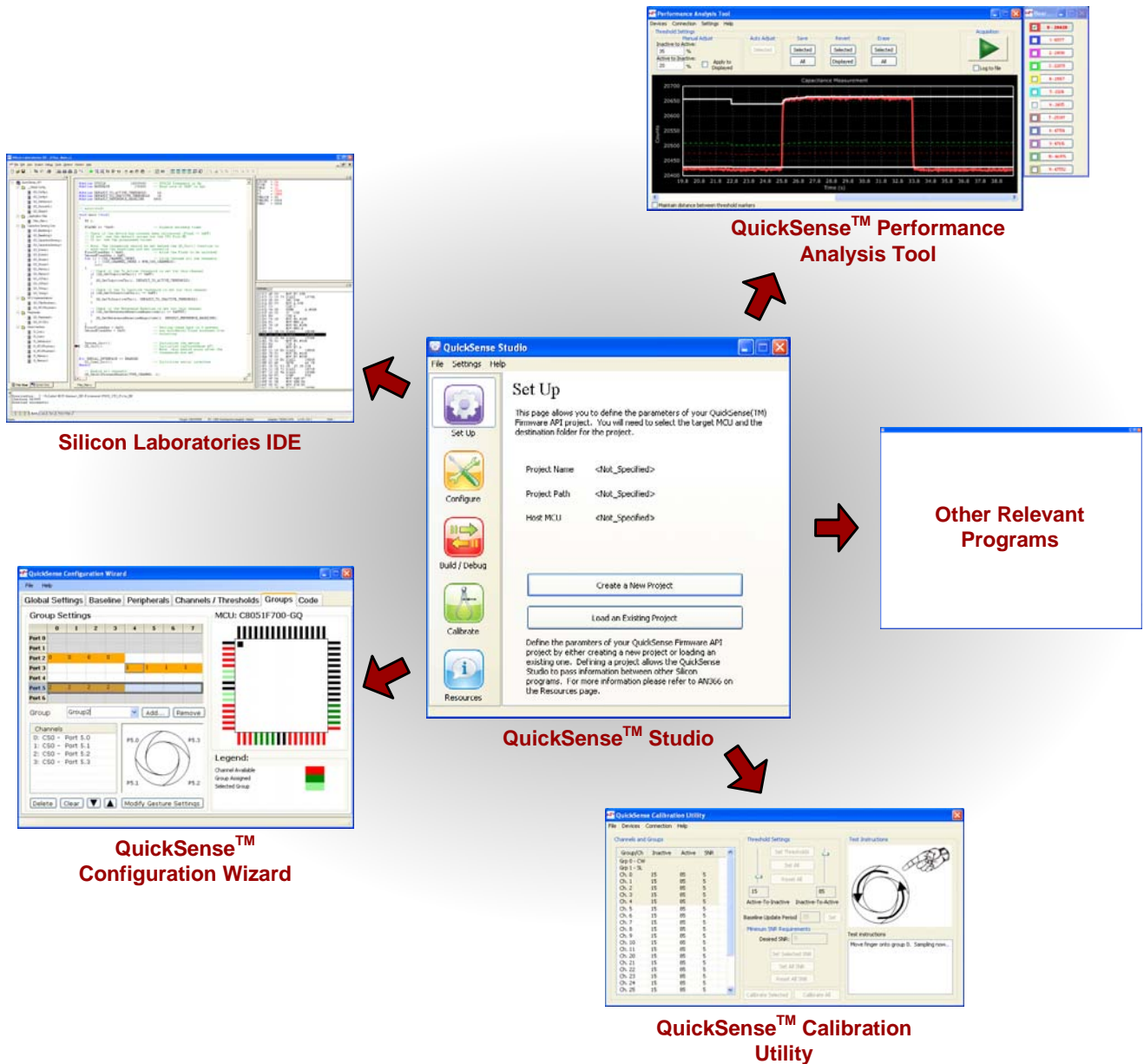
For more information, please refer to the  $\mu$ Vision driver documentation. The documentation and software are available from the Downloads webpage ([www.silabs.com/mcudownloads](http://www.silabs.com/mcudownloads)).

## 5.4. Programming Utilities

The Silicon Labs IDE is the primary tool for downloading firmware to the MCU during development. There are two software programming tools that are intended for use during prototyping or in the field: the MCU Production Programmer and the Flash Programming Utilities. The MCU Production Programmer is installed with the IDE to the directory *C:\Silabs\MCU\Utilities\Production Programmer\* (default). The Flash Programming Utilities can be optionally installed from the CD and is installed to *C:\Silabs\MCU\Utilities\FLASH Programming\* (default).

### 5.5. QuickSense™ Studio

The QuickSense Studio software facilitates rapid code development and analysis for applications using the Silicon Labs family of QuickSense products. The Studio is comprised of multiple programs that guide users through an intuitive development flow, including graphical configuration wizards, firmware templates, and performance monitoring tools. These programs interface with the QuickSense Firmware API, a highly configurable open-source firmware library that provides support for many different applications, from simple buttons to complex algorithms like gesture recognition.



**Figure 3. Silicon Labs QuickSense Studio Software**

For detailed information on the QuickSense Studio software, please see the QuickSense Studio User’s Guide available on the Silicon Labs QuickSense webpage ([www.silabs.com/quicksense](http://www.silabs.com/quicksense)) in the QuickSense Studio section.

## 6. Example Source Code

Example source code and register definition files are provided in the “*SiLabs\MCU\Examples\C8051F80x\_83x*” default directory during IDE installation. These files may be used as a template for code development. The comments in each example file indicate which development tool chains were used when testing. Example applications include a blinking LED example which configures the green LED on the target board to blink at a fixed rate. Also included are examples for each of peripherals of the MCU such as the UART.

### 6.1. Register Definition Files

Register definition files *C8051F800.inc*, *C8051F800\_defs.h* and *compiler\_defs.h* define all SFR registers and bit-addressable control/status bits. These files are installed into the “*SiLabs\MCU\Examples\C8051F80x\_83x\Header\_Files*” default directory during IDE installation. The register and bit names are identical to those used in the C8051F80x-83x data sheet.

### 6.2. Blinking LED Example

The example source files *F800\_Blinky.asm* and *F800\_Blinky.c* installed in the default directory “*SiLabs\MCU\Examples\C8051F80x\_83x\Blinky*” show examples of several basic C8051F800 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 C8051F800 Target Board about five times a second using the interrupt handler with a C8051F800 timer.

### 6.3. Capacitive Sense Switch Example

The example source file *F80x\_CS0.c* demonstrates the configuration and usage of the capacitive sense switches labeled P1.5 and P1.6. Refer to the source file for step-by-step instructions to build and test this example. This is installed in the “*SiLabs\MCU\Examples\C8051F80x\_83x\CS0*” directory by default.

### 6.4. QuickSense™ Firmware API Example

The QuickSense Studio software install (available at [www.silabs.com/quicksense](http://www.silabs.com/quicksense)) includes a QuickSense Firmware API example for the C8051F800 Target Board. This is installed in the “*SiLabs\MCU\QuickSense\_Studio\Firmware\F800\_TargetBoard*” directory by default. In addition to the source files, a pre-built Intel Hex file (*F800TB.hex*) is also included for quick evaluation. This firmware uses the QuickSense Firmware API to measure capacitance on the two sensing pads and application layer code indicates touch using the LEDs on the board. The firmware provides the following functionality:

- When the capacitive sense button labeled P1.5 is touched, the four LEDs labeled P1.0 through P1.3 are sequentially lit from bottom to top.
- When the capacitive sense button labeled P1.6 is touched, the four LEDs are sequentially lit from top to bottom.
- When both the buttons P1.5 and P1.6 are touched simultaneously, the LED sequencing is paused.
- When neither of the two buttons is touched, all the four LEDs are not lit.

For a more detailed description of the QuickSense Firmware API or the Serial Interface, see “AN366: QuickSense API.” For a more detailed description of active/inactive thresholds, see “AN367: Understanding Capacitive Sensing Signal to Noise Ratios.” For a discussion on baselining in the QuickSense Firmware API, see “AN418: Baselining in the QuickSense Firmware API.”



## 7. Target Board

The C8051F800 Development Kit includes a target board with a C8051F800-GM 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 10 shows the factory default shorting block positions. A summary of the signal names and headers is provided in Table 5 on page 15.

- P1 Power connector (accepts input from 7 to 15 VDC unregulated power adapter)
- P2 USB connector (connects to PC for serial communication)
- J1 22-pin Expansion I/O connector
- J2 MCU power header (VDD)
- J3 Port I/O configuration header
- J4 DEBUG connector for Debug Adapter interface
- J5 Connects pin P0.0 to the VREF bypass capacitors and TB1
- J6, J7 Connects the potentiometer (R14) to pin P0.7 and +3VD
- J8, J9 Connects pins P0.2 (XTAL1) and P0.3 (XTAL2) to J1
- J10 Connects pin P0.1 to GND and TB1
- TB1 Analog I/O terminal block

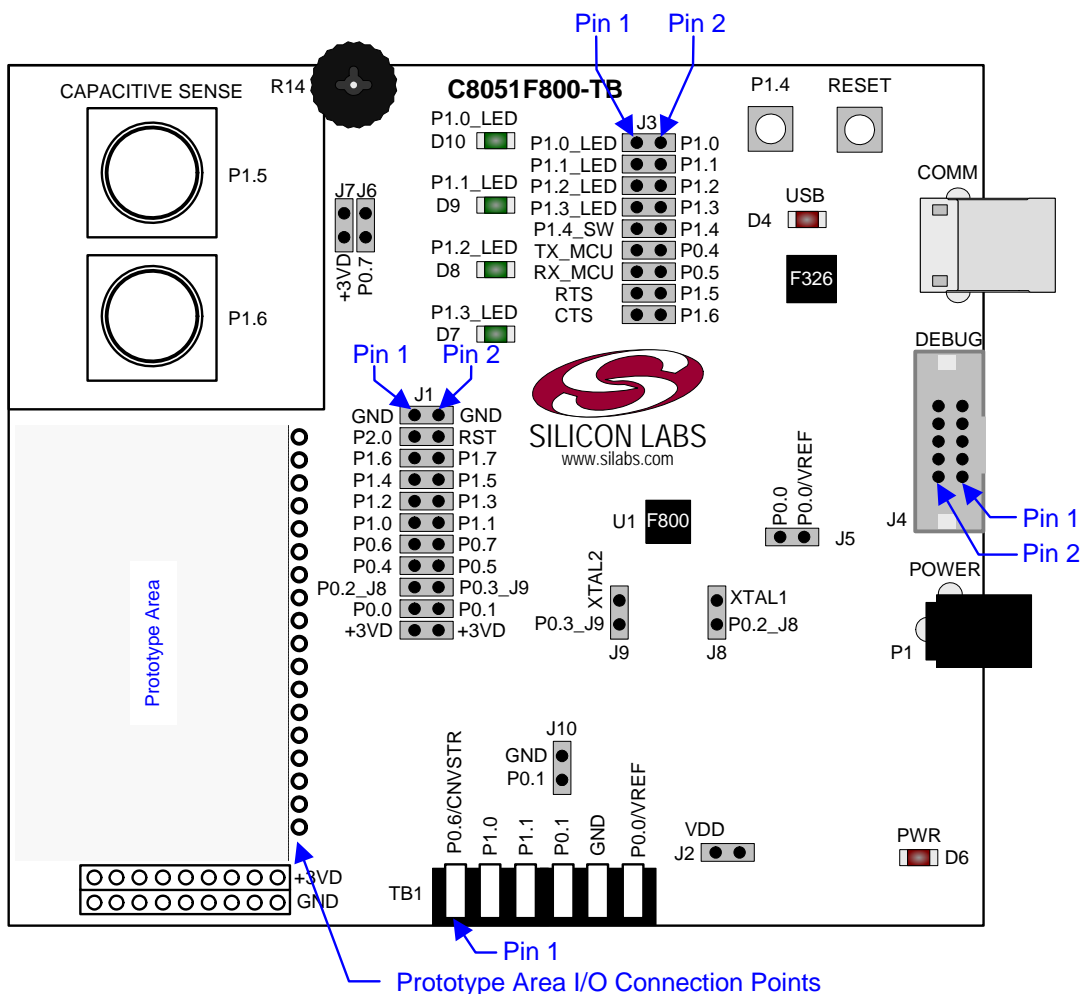


Figure 4. C8051F800 Target Board with Pin Numbers

# C8051F800-DK

## 7.1. Target Board Shorting Blocks: Factory Defaults

The C8051F800 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. It should be noted that a shorting block on J2 to connect power to the MCU is required for normal operation.

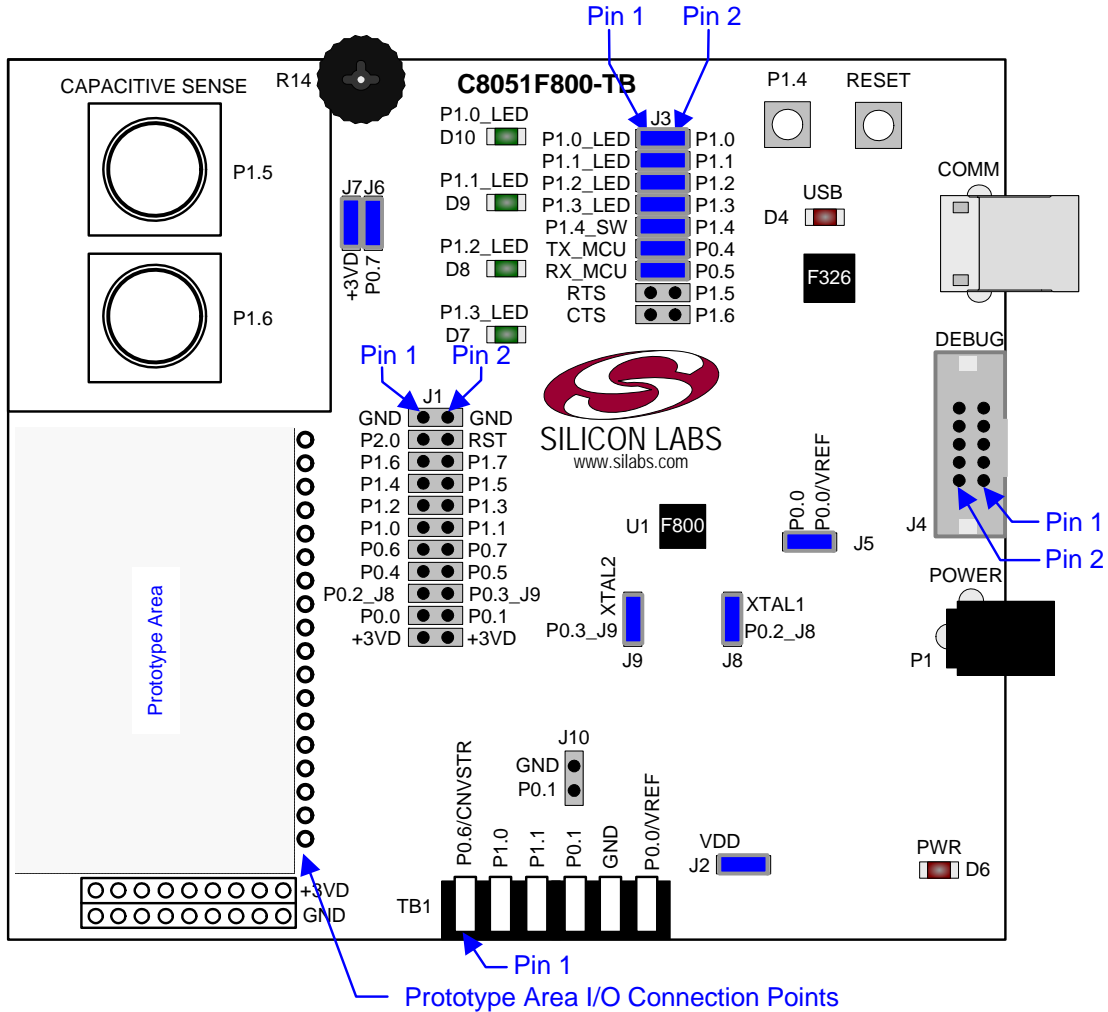


Figure 5. C8051F800 Target Board Shorting Blocks: Factory Defaults

## 7.2. Target Board Power Options and Current Measurement (P1, P2, J2, J4)

The C8051F800 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 CP2103 USB connector (P2, labeled “COMM”)

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 3.3 V dc voltage using a LDO regulator (U2). The output of the regulator powers the +3 VD 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 C8051F800 MCU. With the shorting block removed, a multimeter can be used across J2 to measure the current consumption of the MCU.

## 7.3. System Clock Sources (J8, J9)

### 7.3.1. Internal Oscillators

The C8051F800 device installed on the target board features a factory-calibrated, programmable high-frequency internal oscillator (24 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 by default but may be configured by software to operate at other frequencies. The on-chip crystal is accurate for many serial communications (UART, SPI, SMBus) and an external oscillator is not required depending on the bit rate. However, if you wish to operate the C8051F800 device at a frequency not available with the internal oscillator, an external crystal may be used. Refer to the C8051F80x-83x data sheet for more information on configuring the system clock source.

### 7.3.2. External Oscillator Options

The target board is designed to facilitate the installation of an external crystal. Remove shorting blocks at headers J8 and J9 and install the crystal at the pads marked Y1. Install a 10 M $\Omega$  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 C13 for capacitor mode, and populate R7 and C13 for RC mode. Refer to the C8051F80x-83x data sheet for more information on the use of external oscillators.

## 7.4. Switches and LEDs (J3)

Two push-button switches are provided on the target board. The switch labeled 'RESET' (SW1 in schematic) is connected to the reset pin ( $\overline{RST}$ ) of the C8051F800. Pressing this switch puts the device into its hardware-reset state. The switch labeled "P1.4" (SW2 in schematic) is connected to the C8051F800's general purpose I/O (GPIO) pin through headers. Pressing this switch generates a logic low signal on the port pin. Remove the shorting block from the header to disconnect this switch from the port pin. The port pin signal is also routed to a pin on the J1 I/O connector. See Table 1 for the port pins and headers corresponding to each switch.

Two capacitive sense switches are also provided on the target board. The operation of these switches needs appropriate code running on the C8051F800 MCU that can sense the state of the switch. Note that no shorting blocks should be present on J3[15-16] and J3[17-18] for proper operation of these switches. See Section 6.3. "Capacitive Sense Switch Example," on page 8 for details about example source code.

Six 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. Another red LED labeled "USB" is used to indicate a valid USB connection via the USB connector labeled "COMM." Note that this LED will light up only after CP210x device drivers are loaded successfully on the PC. The four green LEDs labeled with port pin names P1.0\_LED through P1.3\_LED are connected to the C8051F800's GPIO pins P1.0 through P1.3, respectively, via 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. 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)
RESET (SW1)	Reset	none
P1.4 (SW2)	P1.4	J3[9-10]
P1.5 (Capacitive Sense)	P1.5	J3[15-16]*
P1.6 (Capacitive Sense)	P1.6	J3[17-18]*
P1.0_LED (Green LED)	P1.0	J3[1-2]
P1.1_LED (Green LED)	P1.1	J3[3-4]
P1.2_LED (Green LED)	P1.2	J3[5-6]
P1.3_LED (Green LED)	P1.3	J3[7-8]
PWR (Red LED)	Power	none
USB (Red LED)	USB Active	none
<b>*Note:</b> Shorting blocks should NOT be present at these headers for the proper operation of the capacitive sense switches.		

## 7.5. Target Board Debug Interface (J4)

The DEBUG connector J4 provides access to the DEBUG (C2) pins of the C8051F800. 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 DEBUG pin definitions.

**Table 2. DEBUG Connector Pin Descriptions**

Pin #	Description
1	+3 VD (+3.3 VDC)
2, 3, 9	GND (Ground)
4	P2.0/C2D
5	$\overline{\text{RST}}$ (Reset)
6	P2.0
7	$\overline{\text{RST}}/\text{C2CK}$
8	Not Connected
10	USB Power (+5 VDC)

## 7.6. Serial Interface (P2, 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 C8051F800. The Silicon Labs CP2103 USB-to-UART bridge provides data connectivity between the C8051F800 and the PC via a USB port. The TX, RX, RTS and CTS signals of UART0 may be connected to the CP2103 by installing shorting blocks on header J3. Note that the use of the optional RTS/CTS handshaking signals will prevent the simultaneous use of the capacitive sense switches because those GPIO pins are shared on this board. The shorting block positions for connecting each of these signals to the CP2103 are listed in Table 3. To use this interface, the USB-to-UART device drivers should be installed as described in Section 4.3. "CP210x USB to UART VCP Driver Installation," on page 3.

**Table 3. Serial Interface Header Description**

Header Pins	UART0 Pin Description
J3[11–12]	TX_MCU (P0.4)
J3[13–14]	RX_MCU (P0.5)
J3[15–16]	RTS (P1.5)
J3[17–18]	CTS (P1.6)

## 7.7. Potentiometer (J6, J7)

The C8051F800 MCU has the option to connect port pin P0.7 to a 10K linear potentiometer (R14). The potentiometer is connected through the J6 and J7 headers. The potentiometer can be used for testing the analog-to-digital (ADC) converter of the MCU.

## 7.8. Analog I/O, Voltage and Ground Reference Options (TB1, J5, J10)

Several of the C8051F800 target device's port pins are connected to the TB1 terminal block. Refer to Table 4 for the TB1 terminal block connections. The J5 header connects the MCU VREF pin (P0.0) to the VREF bypass capacitors C18 and C19, and also to TB1 pin 6 for an optional external VREF input. The J10 header connects P0.1 to GND, and is useful if the P0.1/AGND option is enabled via the REF0CN register in the C8051F800. Refer to the C8051F80x-83x data sheet for more information on configuring the voltage and ground reference options.

**Table 4. TB1 Terminal Block Pin Descriptions**

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

## 7.9. C2 Pin Sharing

On the C8051F800, the debug pins C2CK and C2D are shared with the pins  $\overline{\text{RST}}$  and P2.0 respectively. The target board includes the resistors necessary to enable pin sharing which allow the  $\overline{\text{RST}}$  and P2.0 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](http://www.silabs.com) for more information regarding pin sharing.

## 7.10. Target Board Pin Assignment Summary

Some GPIO pins of the C8051F800 MCU can have an alternate fixed function. For example, pin 17 on the C8051F800-GM 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 C8051F800 Target Board. The "Relevant Headers" column shows that this signal is routed to pin 15 of the J1 header and pin 12 of the J3 header. More details can be found in the C8051F80x-83x data sheet. Some of the GPIO pins of the C8051F800 have been used for various functions on the target board. Table 5 summarizes the MCU pin assignments on the target board, and also shows the various headers associated with each signal.

**Table 5. C8051F800 Target Board Pin Assignments and Headers**

MCU Pin Name	Pin# ('F800-GM)	Primary Function	Alternate Fixed Function	Target Board Function	Relevant Headers
P0.0	1	P0.0 (GPIO)	VREF	VREF	J1[19], TB1[6]*, J5[1]
P0.1	20	P0.1 (GPIO)	AGND	GPIO/AGND	J1[20], TB1[4], J10[1]
P0.2	19	P0.2 (GPIO)	XTAL1	XTAL1	J1[17]*, J8[2]
P0.3	18	P0.3 (GPIO)	XTAL2	XTAL2	J1[18]*, J9[2]
P0.4	17	P0.4 (GPIO)	TX_MCU	TX_MCU	J1[15], J3[12]
P0.5	16	P0.5 (GPIO)	RX_MCU	RX_MCU	J1[16], J3[14]
P0.6	15	P0.6 (GPIO)	CNVSTR	CNVSTR	J1[13], TB1[1]
P0.7	14	P0.7 (GPIO)		Potentiometer	J1[14], J6[1]
P1.0	13	P1.0 (GPIO)		LED	J1[11], J3[2], TB1[2]
P1.1	12	P1.1 (GPIO)		LED	J1[12], J3[4], TB1[3]
P1.2	11	P1.2 (GPIO)		LED	J1[9], J3[6]
P1.3	10	P1.3 (GPIO)		LED	J1[10], J3[8]
P1.4	9	P1.4 (GPIO)		Switch (SW2)	J1[7], J3[10]
P1.5	8	P1.5 (GPIO)		Capacitive Sense Switch / RTS	J1[8], J3[16]
P1.6	7	P1.6 (GPIO)		Capacitive Sense Switch / CTS	J1[5], J3[18]
P1.7	6	P1.7 (GPIO)		GPIO	J1[6]
P2.0/C2D	5	P2.0 (GPIO)	C2D	P2.0/C2D	J1[3]*, J4[6]*, J4[4]
$\overline{\text{RST}}/\text{C2CK}$	4	$\overline{\text{RST}}$ (Reset)	C2CK	$\overline{\text{RST}}/\text{C2CK}$	J1[4]*, J4[5]*, J4[7]
VDD	3	VDD (Power)		VDD	J2[2]
GND	2	GND (Ground)		GND	J1[1], J1[2], 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.





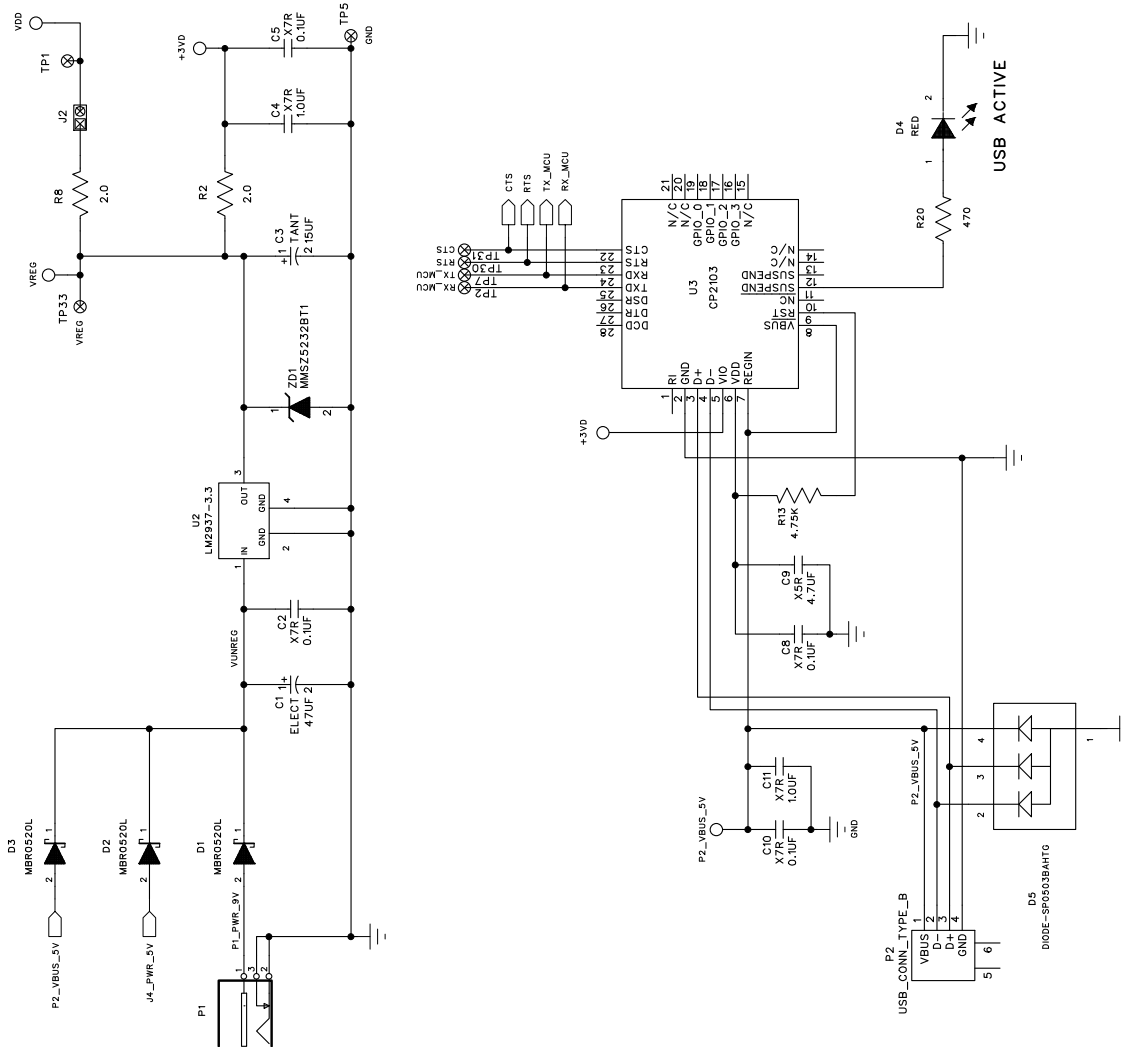


Figure 7. C8051F800 Target Board Schematic (Page 2 of 2)

## 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

Please visit the Silicon Labs Technical Support web page:  
<https://www.silabs.com/support/pages/contacttechnicalsupport.aspx>  
and register to submit a technical support request.

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.