C8051F996-DK

C8051F996 Development Kit User's Guide

1. Relevant Devices

The C8051F996 Development Kit is intended as a development platform for the microcontrollers in the C8051F99x-C8051F98x MCU family. The members of this MCU family are C8051F990, C8051F991, C8051F996, C8051F997, C8051F980, C8051F981, C8051F982, C8051F983, C8051F985, C8051F986, C8051F987, C8051F988, and C8051F989.

Notes:

- The target board included in this kit is provided with a pre-soldered C8051F996 MCU (QSOP24 package).
- Code developed on the C8051F996 can be easily ported to the other members of this MCU family.
- Refer to the C8051F99x-C8051F98x data sheet for the differences between the members of this MCU family.

2. Kit Contents

The C8051F996 Development Kit contains the following items:

- C8051F996 Target Board
- C8051Fxxx Development Kit Quick-Start Guide
- Silicon Laboratories IDE and Product Information CD-ROM. CD content includes the following:
  - Silicon Laboratories Integrated Development Environment (IDE)
  - Evaluation version of 8051 development tools (macro assembler, linker, C compiler)
  - Source code examples and register definition files
  - Documentation
  - C8051F996 Development Kit User's Guide (this document)
- Third Party Tools CD
- AC to DC Power Adapter
- USB Debug Adapter (USB to Debug Interface)
- 2 USB Cables
- 2 AAA Batteries

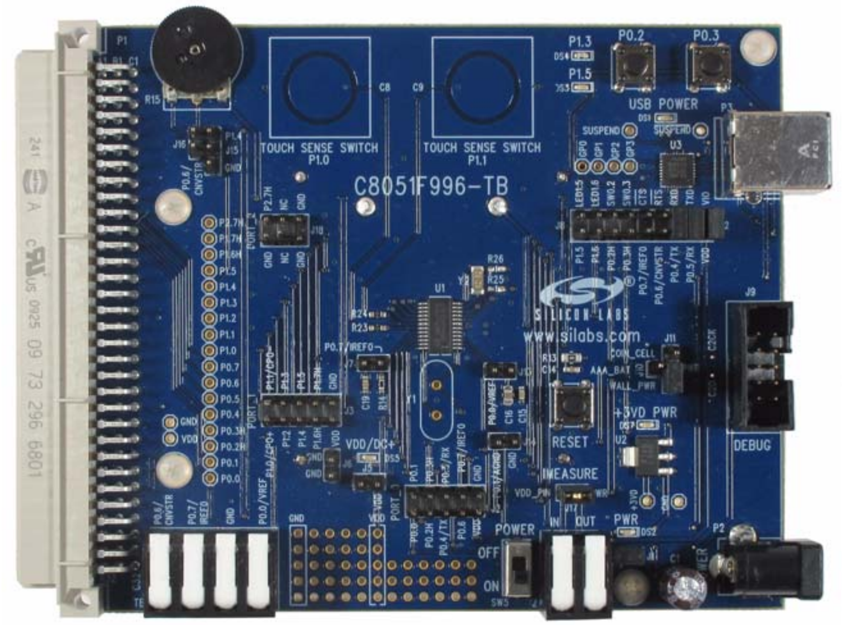


Figure 1. C8051F996 Target Board

3. Hardware Setup

Refer to Figure 2 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 a shorting block is installed on J17 and that SW5 is in the ON position.
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 P2 on the target board (Optional).

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.

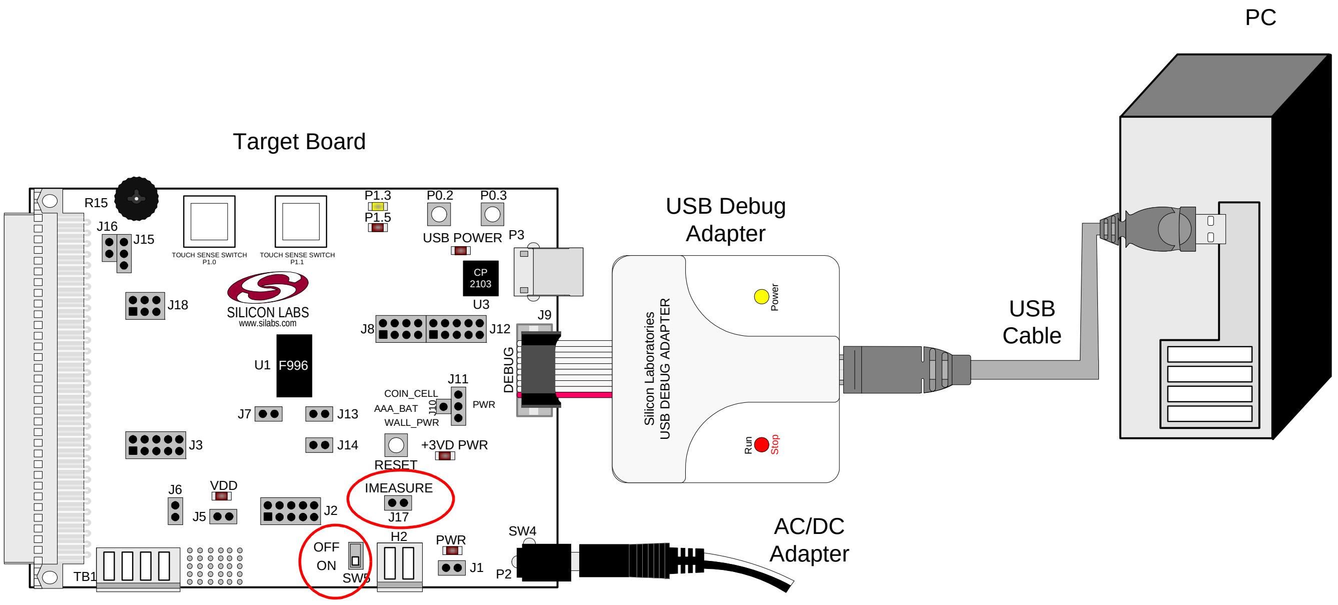


Figure 2. Hardware Setup using a USB Debug Adapter

4. Software Installation

The included CD-ROM contains the Silicon Labs Integrated Development Environment (IDE), 8051 evaluation toolset, optional software utilities, 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.

4.1. System Requirements

The following are the system requirements necessary to run the debug and programming tools:

- Pentium-class host PC running Microsoft Windows 2000 or newer.
- One available USB port.

4.2. Development Tools Installation

To install the IDE, utilities, and code examples, perform the following steps:

1. Click on the "Install Development Tools" button on the installation utility's startup screen.
2. In the Kit Selection box that appears, choose the C8051F996-DK development kit from the list of options.
3. In the next screen, choose “Components to be Installed”. The programs necessary to download and debug on the MCU are the **Silicon Labs IDE** and the **8051 Evaluation Toolset**. The **CP210x Drivers** are necessary to use the UART capabilities of the target board. See Section 4.3. for more information about installing the CP210x drivers. See Section 5. for an overview of all applicable software included on the CD-ROM.
4. Installers selected in Step 3 will execute in sequence, prompting the user as they install programs, documentation, and drivers.

4.3. CP210x USB to UART VCP Driver Installation

The C8051F996 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:\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 (P3) on the C8051F996 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.

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

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

- Configuration Wizard 2
- Keil µVision2 and µVision3 Drivers
- MCU Production Programmer and Flash Programming Utilities

5.1. 8051 Evaluation Toolset

The Silicon Labs IDE has native support for many third party 8051 toolsets. Included with this kit are several 8051 evaluation assembler, compiler, and linker toolsets. For further information on the tools, including limitations, see the corresponding application note. Application notes can be found in the documentation section of the Development Kit CD, or on the Silicon Labs website (http://www.silabs.com/appnotes). See Table 1 for a list of supported toolsets and associated application notes.

Table 1. Supported Third Party 8051 Toolsets

Toolset	Application Note
Keil	AN104, “Integrating Keil 8051 Tools into the Silicon Labs IDE”
Raisonance	AN125, “Integrating Raisonance 8051 Tools into the Silicon Labs IDE”
Tasking	AN126, “Integrating Tasking 8051 Tools into the Silicon Labs IDE”
HI-TECH	AN140, “Integrating Hi-TECH 8051 Tools into the Silicon Labs IDE”
SDCC	AN198, “Integrating SDCC 8051 Tools into the Silicon Labs IDE”
IAR	AN236, “Integrating IAR 8051 Tools into the Silicon Labs IDE”

5.2. Silicon Labs IDE

The Silicon Labs IDE integrates a source-code editor, source-level debugger and in-system programmer. The following sections discuss how to open an example project in the IDE, build the source code, and download it to the target device.

5.2.1. Running the Fxxx_Blinky example program

The **Fxxx_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\C8051F99x_98x\Blinky* directory (default) and select the Fxxx_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. C8051Fxxx/xx 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.

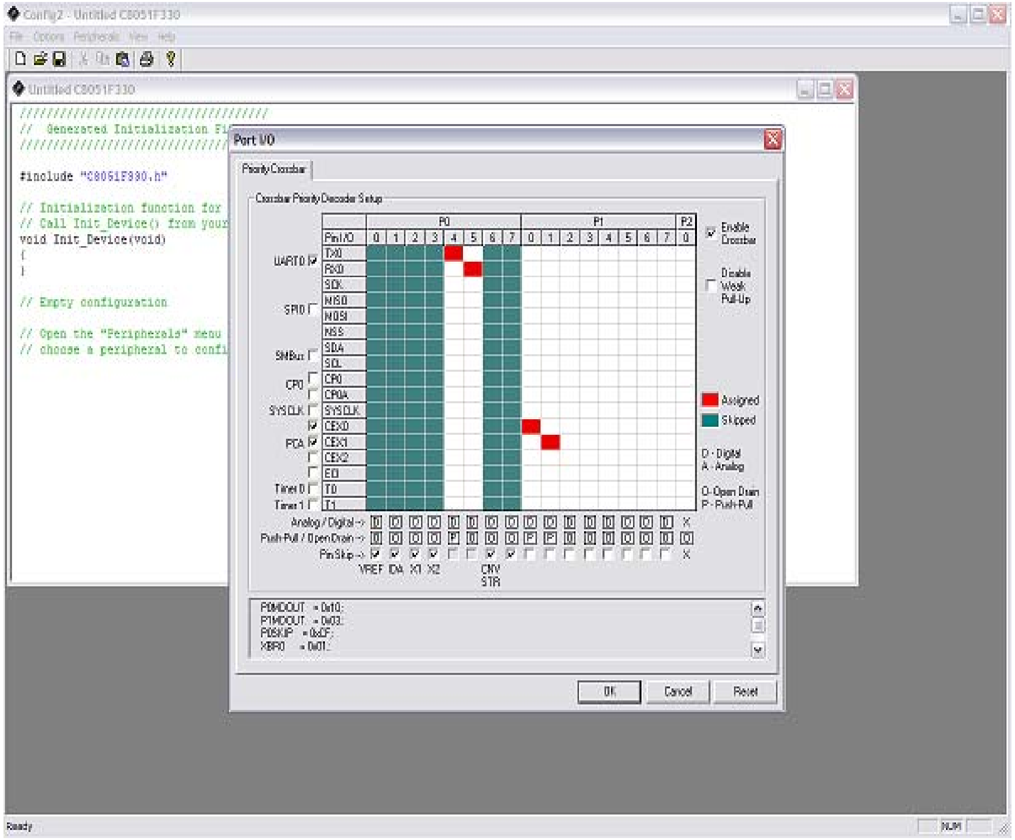
5.2.2. Creating a New Project

Use the following steps to create a new projects. Once steps 1-5 in this section are complete, continue at Step 3 in Section 5.2.1.

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**.

5.3. 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.



```
/////////////////////////////////////
//  Generated Initialization Fi
/////////////////////////////////

#include "C8051F330.h"

// Initialization function for
// Call Init_Device() from your
void Init_Device(void)
{
}

// Empty configuration

// Open the "Peripherals" menu
// choose a peripheral to confi
```

```
P0MDOUT  = 0x10;
P1MDOUT  = 0x03;
P0SKIP   = 0xCF;
XBR0     = 0x01;
```

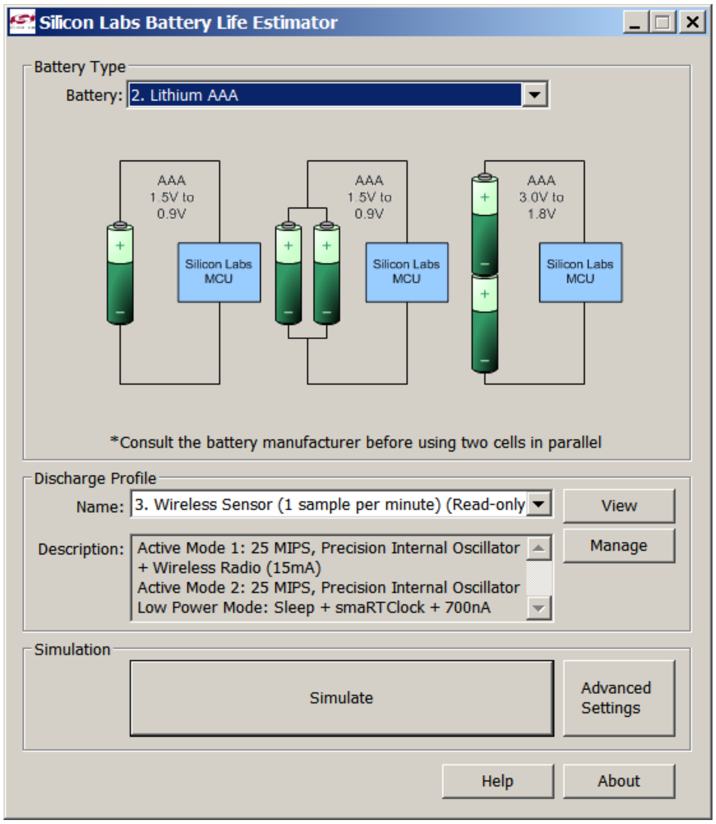
Figure 3. Configuration Wizard 2 Utility

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).

5.4. Silicon Labs Battery Life Estimator

The Battery Life Estimator is a system design tool for battery operated devices. It allows the user to select the type of battery they are using in the system and enter the supply current profile of their application. Using this information, it performs a simulation and provides an estimated system operating time. The Battery Life Estimator is shown in Figure 4.



Battery: 2. Lithium AAA

*Consult the battery manufacturer before using two cells in parallel

Name: 3. Wireless Sensor (1 sample per minute) (Read-only

Description: Active Mode 1: 25 MIPS, Precision Internal Oscillator + Wireless Radio (15mA)
Active Mode 2: 25 MIPS, Precision Internal Oscillator
Low Power Mode: Sleep + smaRTClock + 700nA

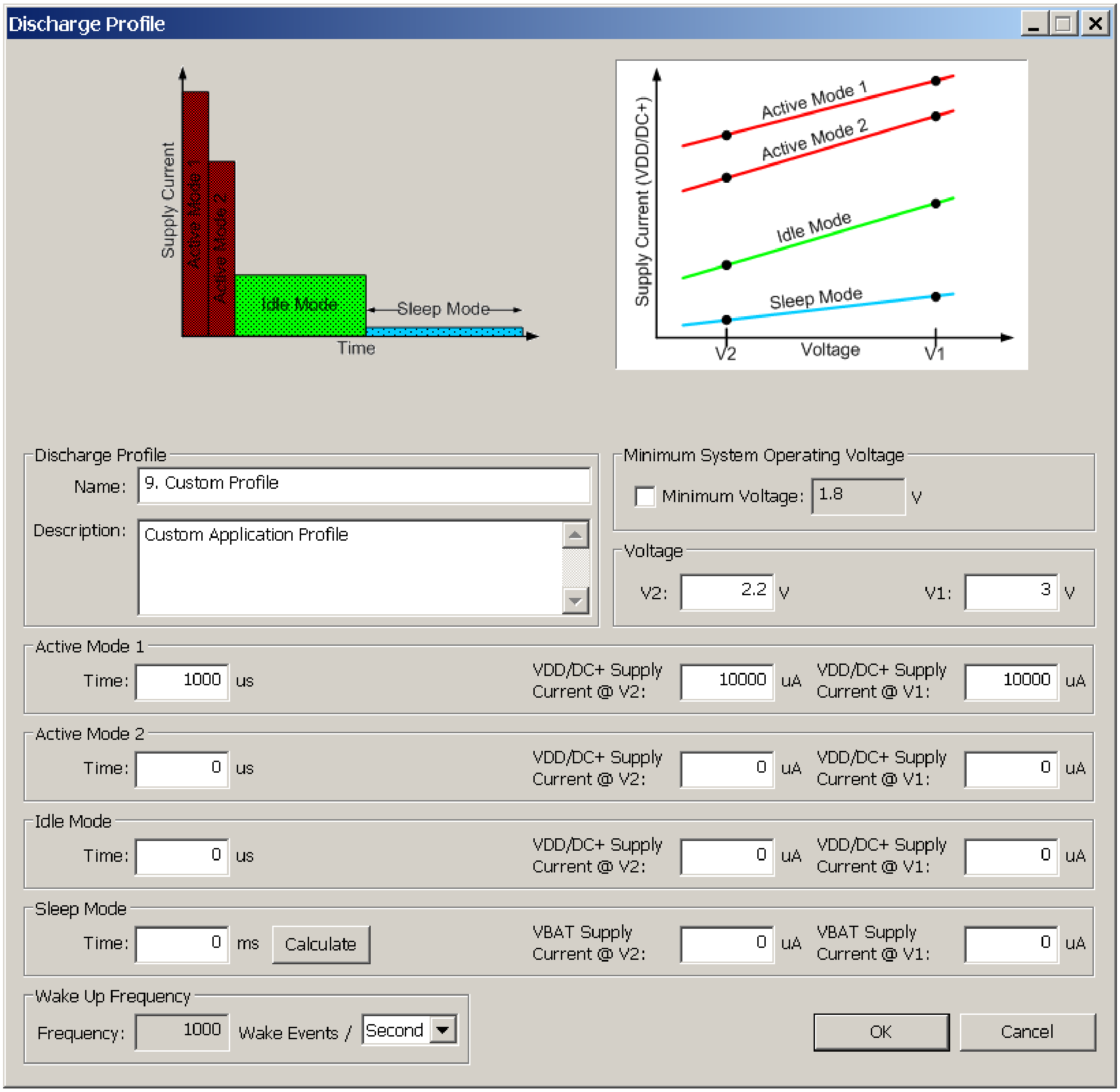
Figure 4. Battery Life Estimator Utility

From Figure 4, the two inputs to the Battery Life Estimator are battery type and discharge profile. The utility includes battery profiles for common battery types such as AAA, AA, A76 Button Cell, and CR2032 coin cell. The discharge profile is application-specific and describes the supply current requirements of the system under various supply voltages and battery configurations. The discharge profile is independent of the selected power source. Several read-only discharge profiles for common applications are included in the pulldown menu. The user may also create a new profile for their own applications.

To create a new profile:

1. Select the profile that most closely matches the target application or choose the "Custom Profile".
2. Click Manage
3. Click Duplicate
4. Click Edit

Profiles may be edited with the easy-to-use GUI (shown in Figure 5).



Discharge Profile

Name: 9. Custom Profile

Description: Custom Application Profile

Minimum System Operating Voltage

☐ Minimum Voltage: 1.8 V

Voltage

V2: 2.2 V V1: 3 V

Active Mode 1

Time: 1000 us VDD/DC+ Supply Current @ V2: 10000 uA VDD/DC+ Supply Current @ V1: 10000 uA

Active Mode 2

Time: 0 us VDD/DC+ Supply Current @ V2: 0 uA VDD/DC+ Supply Current @ V1: 0 uA

Idle Mode

Time: 0 us VDD/DC+ Supply Current @ V2: 0 uA VDD/DC+ Supply Current @ V1: 0 uA

Sleep Mode

Time: 0 ms VBAT Supply Current @ V2: 0 uA VBAT Supply Current @ V1: 0 uA

Wake Up Frequency

Frequency: 1000 Wake Events / Second

Figure 5. Battery Life Estimator Discharge Profile Editor

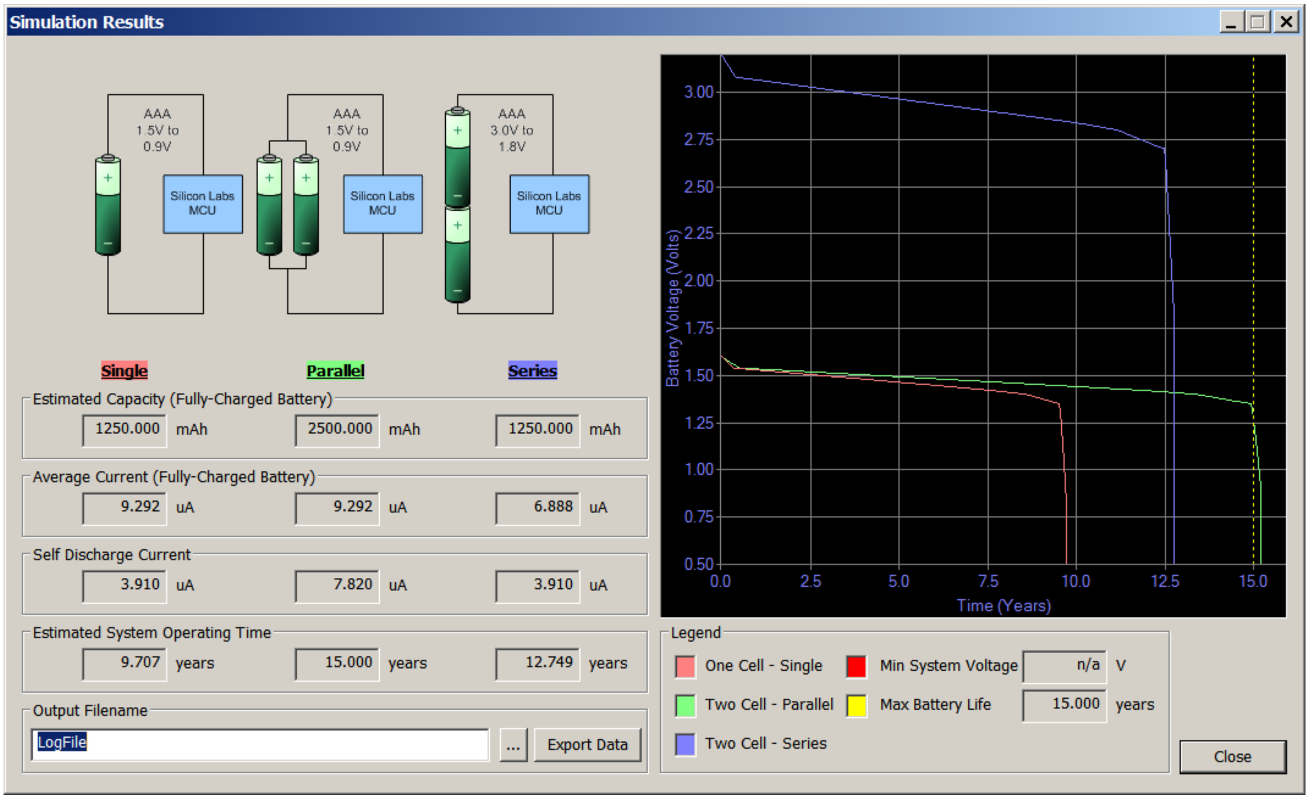
The Discharge Profile Editor allows the user to modify the profile name and description. The four text entry boxes on the left hand side of the form allow the user to specify the amount of time the system spends in each power mode. On the right hand side, the user may specify the supply current of the system in each power mode.

Since supply current is typically dependent on supply voltage, the discharge profile editor provides two columns for supply current. The V2 and V1 voltages at the top of the two columns specify the voltages at which the current measurements were taken. The Battery Life Estimator creates a linear approximation based on the input data and is able to feed the simulation engine with an approximate supply current demand for every input voltage.

The minimum system operating voltage input field allows the system operating time to stop increasing when the simulated battery voltage drops below a certain threshold. This is primarily to allow operating time estimates for systems that cannot operate down to 1.8 V, which is the voltage of two fully drained single-cell batteries placed in series.

The wakeup frequency box calculates the period of a single iteration through the four power modes and displays the system wake up frequency. This is typically the "sample rate" in low power analog sensors.

Once the battery type and discharge profile is specified, the user can click the "Simulate" button to start a new simulation. The simulation engine calculates the estimated battery life when using one single-cell battery, two single-cell batteries in series, and two single-cell batteries in parallel. Figure 6 shows the simulation output window.



	Single		Parallel		Series	
Estimated Capacity (Fully-Charged Battery)	1250.000	mAh	2500.000	mAh	1250.000	mAh
Average Current (Fully-Charged Battery)	9.292	uA	9.292	uA	6.888	uA
Self Discharge Current	3.910	uA	7.820	uA	3.910	uA
Estimated System Operating Time	9.707	years	15.000	years	12.749	years

Output Filename: LogFile

Legend

🟥 One Cell - Single
🟩 Two Cell - Parallel
🟦 Two Cell - Series
🟥 Min System Voltage n/a V
🟨 Max Battery Life 15.000 years

Figure 6. Battery Life Estimator Utility Simulation Results Form

The primary outputs of the Battery Life Estimator are an estimated system operating time and a simulated graph of battery voltage vs. time. Additional outputs include estimated battery capacity, average current, self-discharge current, and the ability to export graph data to a comma delimited text file for plotting in an external graphing application.

5.5. Keil µVision2 and µVision3 Silicon Labs Drivers

As an alternative to the Silicon Labs IDE, the µVision debug driver allows the Keil µVision 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 µVision 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 µVision driver documentation. The documentation and software are available from the Downloads webpage (www.silabs.com/mcudownloads).

5.6. 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).

6. Example Source Code

Example source code and register definition files are provided in the "*SiLabs\MCU\Examples\C8051F99x_98x*\" default 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

The register definition file *C8051F990_defs.h* defines 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 *C8051F996_defs.h* header file with various tool chains. These files are installed into the "*SiLabs\MCU\Examples\C8051F99x_98x\Header_Files*\" directory during IDE installation by default. The register and bit names are identical to those used in the C8051F99x-C8051F98x data sheet.

6.2. Blinking LED Example

The simple LED blinking example *F99x_Blinky.c* installed in the default examples directory "*SiLabs\MCU\Examples\C8051F99x_98x\Blinky*" shows several basic C8051F996 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 LED on the C8051F996 Target Board about five times a second using the interrupt handler with a C8051F996 timer.

6.3. Current Measurement Example

Example code to place the device in all its low power modes is installed in the examples folder located at "*SiLabs\MCU\Examples\C8051F99x_98x\F9xx_Imeasure".* This example code is described in application note 358 and allows the specified device power consumption to be measured on the bench.

7. Target Board

The C8051F996 Development Kit includes a target board with a C8051F996 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 7 for the locations of the various I/O connectors. Figure 9 on page 13 shows the factory default shorting block positions.

P1	Expansion connector (96-pin)
P2	Power connector (accepts input from 7 to 15 VDC unregulated power adapter)
P3	USB connector (connects to PC for serial communication)
J1	Enable/Disable VBAT Power LED
J2, J3, J18	Port I/O headers (provide access to Port I/O pins)
J5	Enable/Disable VDD Power LED
J6	Provides an easily accessible ground clip
J7	Connects pin P0.7 (IREF0 Output) to resistor R14 and capacitor C19
J8	Connects P0.2 and P0.3 to switches and P1.5 and P1.3 to LEDs
J9	DEBUG connector for Debug Adapter interface
J10, J11	Selects the power supply source (Wall Power, AAA Battery, or Coin Cell)
J12	Connects Port I/O to UART0 interface
J13	Connects external VREF capacitor to the P0.0/VREF
J14	Connects the PCB ground plane to P0.1/AGND
J15	Connects negative potentiometer (R15) terminal to pin P1.4 or to GND
J16	Connects the potentiometer (R15) wiper to P0.6/CNVSTR
J17	Creates an open in the power supply path to allow supply current measurement
TB1	Analog I/O terminal block
H2	Provides terminal block access to the input and output nodes of J17
SW5	Turns power to the MCU on or off

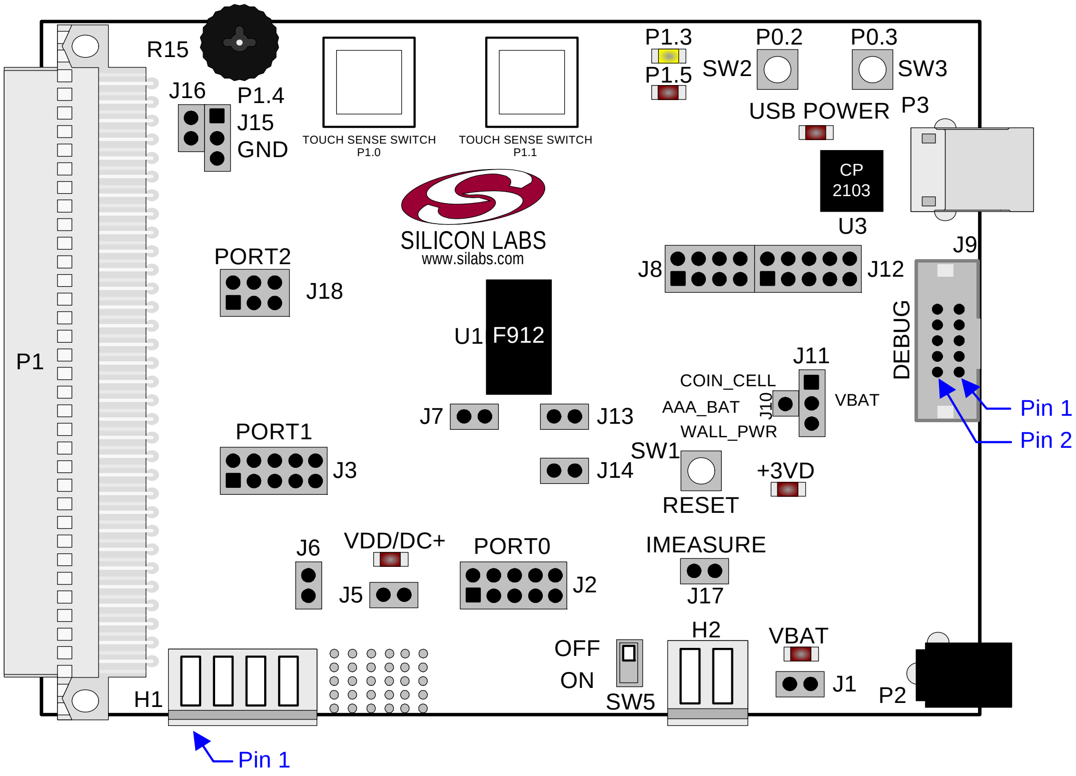


Figure 7. C8051F996 Target Board

The following items are located on the bottom side of the board. See Figure 8.

BH1	Battery Holder for 1.5 V AAA.
BH2	Battery Holder for 1.5 V AAA.
BH3	Battery Holder for 3 V Coin Cell (CR2032).

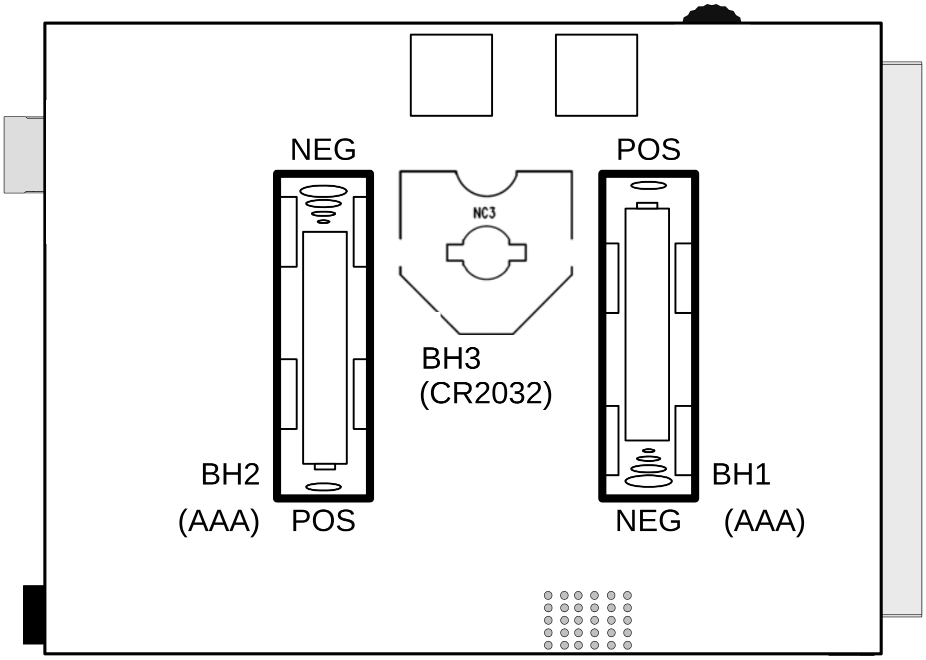


Figure 8. Bottom of C8051F996 Target Board

7.1. Target Board Shorting Blocks: Factory Defaults

The C8051F996 Target Board comes from the factory with pre-installed shorting blocks on many headers. Figure 9 shows the positions of the factory default shorting blocks.

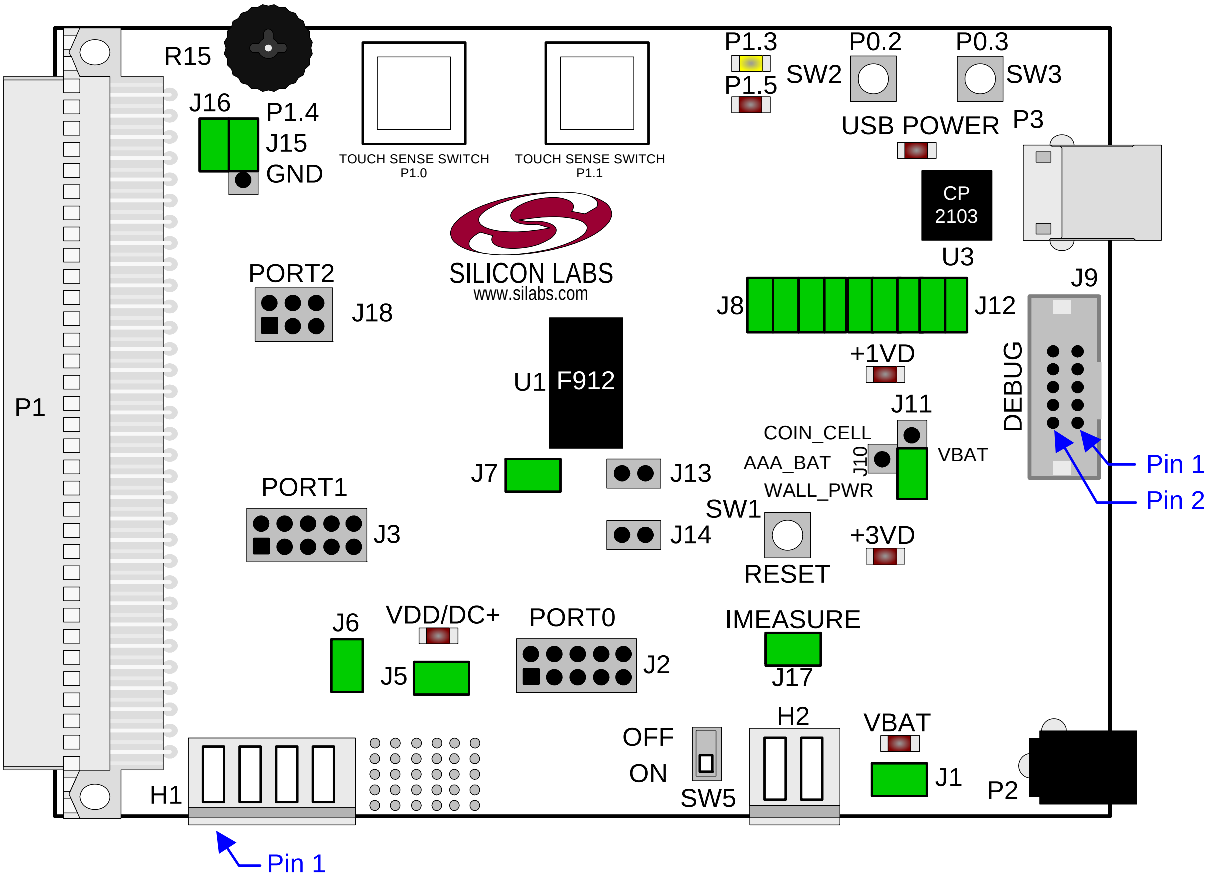


Figure 9. C8051F996 Target Board Shorting Blocks: Factory Defaults

7.2. Target Board Power Options and Current Measurement (J10, J11, J17, H2, P2, P3, SW5)

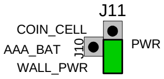
The C8051F996 Target Board supports three power options, selectable by the three-way header (J10/J11). Power to the MCU may be switched on/off using the power switch (SW5). The power options are described in the paragraphs below.

7.2.1. Wall Power

When the J10/J11 three-way header is set to WALL_PWR, the C8051F996 Target Board may be powered from the following power sources:

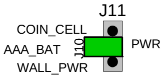
- 9 VDC power using the ac to dc power adapter (P2)
- 5 VDC USB VBUS power from PC via the USB Debug Adapter (J9)
- 5 VDC USB VBUS power from PC via the CP2103 USB connector (P3)

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. The VBAT supply net on the target board is powered directly from the +3 VD net.



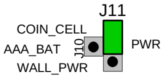
7.2.2. AAA Battery

When the J10/J11 three-way header is set to AAA_BAT, the C8051F996 Target Board is powered from the series combination of the AAA batteries inserted in BH1 and BH2.



7.2.3. Coin Cell Battery

When the J10/J11 three-way header is set to COIN_CELL, the C8051F996 Target Board may be powered from a 3 V Lithium (CR2032) coin cell inserted in BH3.



7.2.4. Measuring Current

The header (J17) and terminal block (H2) provide a way to measure the total supply current flowing from the power supply source to the MCU. The measured current does not include any current from the PWR LED (DS2) or the quiescent current from the power supply; however, it does include the current used by any LEDs powered from the VDD supply net or sourced through a GPIO pin. See the target board schematic in Figure 10 through Figure 12 for additional information.

7.3. System Clock Sources

7.3.1. Internal Oscillators

The C8051F996 device installed on the target board features a factory calibrated programmable high-frequency internal oscillator (24.5 MHz base frequency, ±2%) and a low power internal oscillator (20 MHz ±10%). After each reset, the low power oscillator divided by 8 results in a default system clock frequency of 2.5 MHz (±10%). The selected system clock and the system clock divider may be configured by software for operation at other frequencies. For low-frequency operation, the C8051F996 features a smaRTClock real time clock. A 32.768 kHz Watch crystal (Y2) is included on the target board. If you wish to operate the C8051F996 device at a frequency not available with the internal oscillators, an external crystal may be used. Refer to the C8051F99x-C8051F98x 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 (Y1). Install a 10 MΩ resistor at R17 and install capacitors at C20 and C21 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 C21 for capacitor mode, and populate R16 and C21 for RC mode. Refer to the C8051F99x-C8051F98x data sheet for more information on the use of external oscillators.

7.4. Port I/O Headers (J2, J3, J18, J6)

Access to all Port I/O on the C8051F996 is provided through the headers J2, J3, and J18. The header J6 provides access to the ground plane for easy clipping of oscilloscope probes.

7.5. Switches and LEDs (J1, J5, J8, J15, J16, SW1, SW2, SW3)

Three push-button switches are provided on the target board. Switch SW1 (RESET) is connected to the reset pin of the C8051F996. Pressing SW1 puts the device into its hardware-reset state. Switches SW2 (P0.2) and SW3 (P0.3) are connected to the C8051F996's general purpose I/O (GPIO) pins through headers. Pressing SW2 or SW3 generates a logic low signal on the port pin. Remove the shorting block from the header (J8) to disconnect the switches from the port pins. The port pin signal is also routed to pins on the J2 and P1 I/O connectors. See Table 2 for the port pins and headers corresponding to each switch.

Two touch sensitive (contactless) switches are provided on the target board. The operation of these switches require appropriate firmware running on the C8051F996 MCU that can sense the state of the switch.

Four power LEDs are provided on the target board to serve as indicators. The 3.3 V regulator has a red LED used to indicate the presence of power at the output of the regulator. A red USB Power LED turns on when a USB cable is plugged into the USB connector P3. One power LED is also added to each of the two primary supply nets powering the MCU (VDD and PWR). The PWR net is before the power measurement header and the VDD net is after the power measurement header. The LEDs connected to the supply nets may be disabled by removing the shorting blocks from J1 and J5.

Two LEDs are connected to GPIO pins P1.5 and P1.3 for use by application software. See Table 2 for the port pins and headers corresponding to each LED.

A potentiometer (R15) is also provided on the target board for generating analog signals. Place a shorting block on J16 to connect the wiper to P0.6/CNVSTR. The header J15 allows the negative terminal of the potentiometer to be tied to GND or to P1.4. When tied to GND, the potentiometer is always enabled and will draw a measurable amount of supply current. When tied to P1.4, it only draws current when P1.4 is driving a logic 0 and draws no current when P1.4 is driving a logic 1.

Table 2. Target Board I/O Descriptions

Description	I/O	Header(s)
SW1	Reset	none
SW2	P0.2	J8[5–6]
SW3	P0.3	J8[7–8]
P1.0 (Touch Sense Switch)	P1.0	none
P1.1 (Touch Sense Switch)	P1.1	none
Red LED (P1.5)	P1.5	J8[1–2]
Yellow LED (P1.3)	P1.3	J8[3–4]
Red LED (VDD)	VDD Supply Net	J5
Red LED (PWR)	PWR Supply Net	J1
Red LED (USB Power)	USB VBUS	none
Red LED (+3 VD Power)	+3 VD Regulator Output	none
Potentiometer (R15)	P0.6/P1.4	J15, J16

7.6. Expansion I/O Connector (P1)

The 96-pin Expansion I/O connector P1 provides access to all signal pins of the C8051F996 device (except the C2 debug interface signals). In addition, power supply and ground pins are included. A small through-hole prototyping area is also provided. See Table 3 for a list of pin descriptions for P1.

Table 3. P1 Pin Descriptions

Row A Pin #	Description	Row B Pin #	Description	Row C Pin #	Description
1	+3 VD	1	GND	1	nc
2	nc	2	nc	2	nc
3	nc	3	nc	3	nc
4	nc	4	nc	4	nc
5	nc	5	nc	5	nc
6	nc	6	nc	6	nc
7	nc	7	nc	7	nc
8	nc	8	nc	8	nc
9	nc	9	nc	9	nc
10	nc	10	P0.7/IREF0	10	P0.6/CNVSTR
11	P0.5/RX	11	P0.4/TX	11	P0.3H
12	P0.2H	12	P0.1/AGND	12	P0.0/VREF
13	P2.7H	13	P1.6H	13	P1.5
14	P1.4	14	P1.3	14	P1.2
15	P1.1/CP0-	15	P1.0/CP0+	15	nc
16	nc	16	nc	16	nc
17	nc	17	nc	17	nc
18	nc	18	nc	18	nc
19	nc	19	nc	19	nc
20	nc	20	nc	20	P1.7H
21	nc	21	nc	21	nc
22	nc	22	nc	22	nc
23	nc	23	nc	23	nc
24	nc	24	nc	24	nc
25	nc	25	GND	25	nc
26	GND	26	nc	26	nc
27	nc	27	nc	27	nc
28	nc	28	VDD	28	nc
29	nc	29	nc	29	nc
30	nc	30	nc	30	nc
31	nc	31	nc	31	nc
32	nc	32	GND	32	nc

7.7. Target Board DEBUG Interface (J9)

The DEBUG connector J9 provides access to the DEBUG (C2) pins of the C8051F996. 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 4 shows the DEBUG pin definitions.

Table 4. DEBUG Connector Pin Descriptions

Pin #	Description
1	+3 VD (+3.3 VDC)
2, 3, 9	GND (Ground)
4	P2.7/C2D
5	$\overline{\text{RST}}$ (Reset)
6	P2.7
7	$\overline{\text{RST}}$/C2CK
8	Not Connected
10	USB Power (+5 VDC from J9)

7.8. Serial Interface (J12)

A USB-to-UART bridge circuit (U3) and USB connector (P3) are provided on the target board to facilitate serial connections to UART0 of the C8051F996. The Silicon Labs CP2103 (U3) USB-to-UART bridge provides data connectivity between the C8051F996 and the PC via a USB port. The VIO power supply and TX, RX, RTS and CTS signals of UART0 may be connected to the CP2103 by installing shorting blocks on header J12. The shorting block positions for connecting each of these signals to the CP2103 are listed in Table 5. To use this interface, the USB-to-UART device drivers should be installed as described in 4.3. "CP210x USB to UART VCP Driver Installation," on page 3.

Table 5. Serial Interface Header (J12) Description

Header Pins	UART0 Pin Description
J12[9–10]	CP2103_VIO (VDD)
J12[7–8]	TX_MCU (P0.5)
J12[5–6]	RX_MCU (P0.4)
J12[3–4]	RTS (P0.6)
J12[1–2]	CTS (P0.7)

7.9. Analog I/O (H1)

Several of the C8051F996 target device’s port pins are connected to the H1 terminal block. Refer to Table 6 for the H1 terminal block connections.

Table 6. H1 Terminal Block Pin Descriptions

Pin #	Description
1	P0.6/CNVSTR
2	P0.7/IREF0
3	GND (Ground)
4	P0.0/V_{REF} (Voltage Reference)

7.10. IREF Connector (J7)

The C8051F996 Target Board also features a current-to-voltage 1 kΩ load resistor that may be connected to the current reference (IREF0) output that can be enabled on port pin (P0.7). Install a shorting block on J7 to connect port pin P0.7 of the target device to the load resistor. If enabled by software, the IREF0 signal is then routed to the J2[8] and TB1[2] connectors.

7.11. VREF and AGND Connector (J13, J14)

The C8051F996 Target Board also features 4.7 µF capacitor in parallel with a 0.1 µF that can be connected to P0.0/VREF when using an external reference. The capacitors are connected to P0.0/VREF when a shorting block is installed on J13. Using an external reference is optional since C8051F99x-C8051F98x devices have an on-chip High-Speed Voltage Reference.

A shorting block on J14 allows P0.1/AGND to be connected to ground. This provides a noise-free ground reference to the analog-to-digital Converter. The use of this dedicated analog ground is optional.

7.12. C2 Pin Sharing

On the C8051F996, the debug pins C2CK and C2D are shared with the pins $\overline{\text{RST}}$ and P2.7, respectively. The target board includes the resistors necessary to enable pin sharing which allow the $\overline{\text{RST}}$ and P2.7 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. Schematics

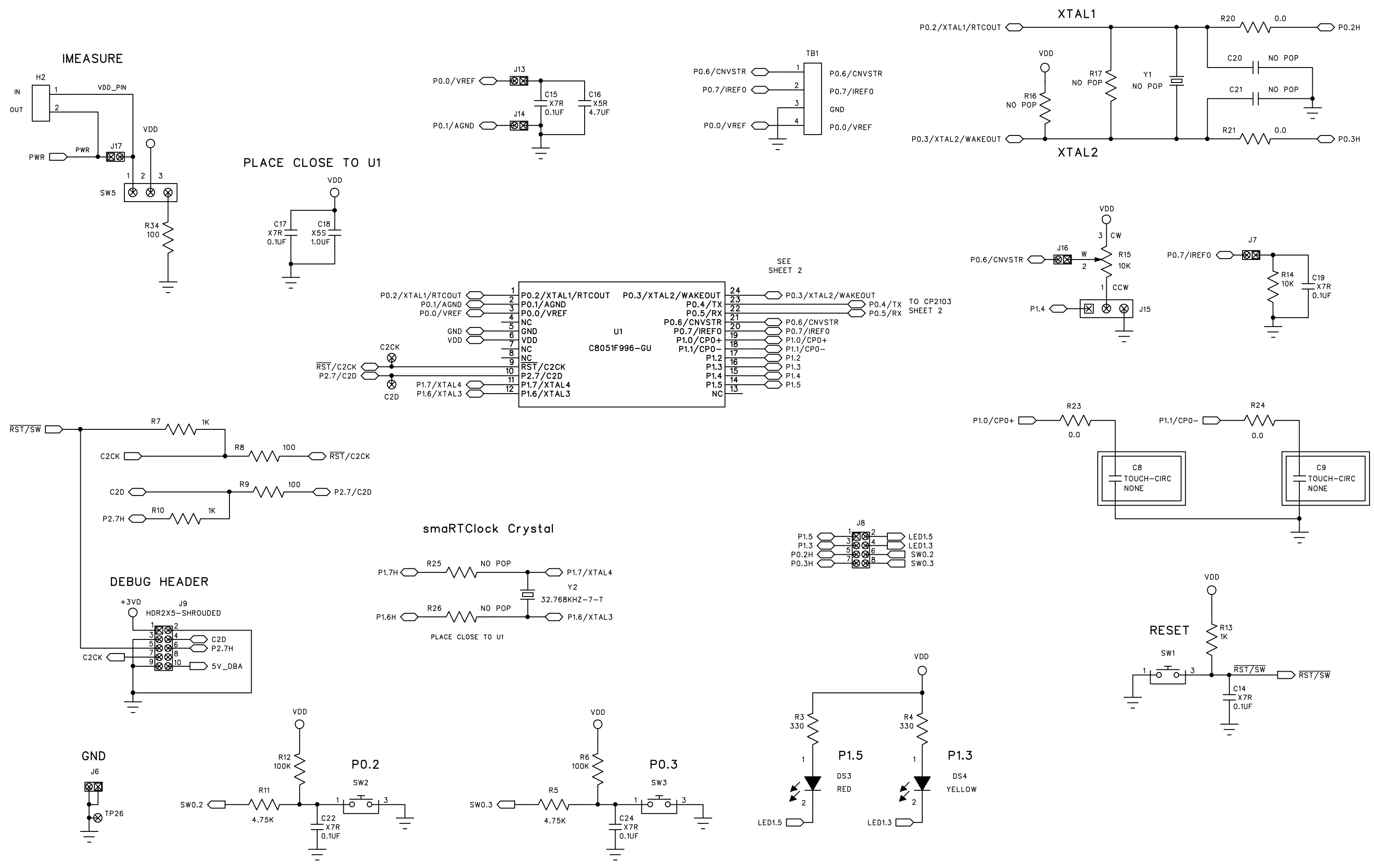


Figure 10. C8051F996 Target Board Schematic (1 of 3)

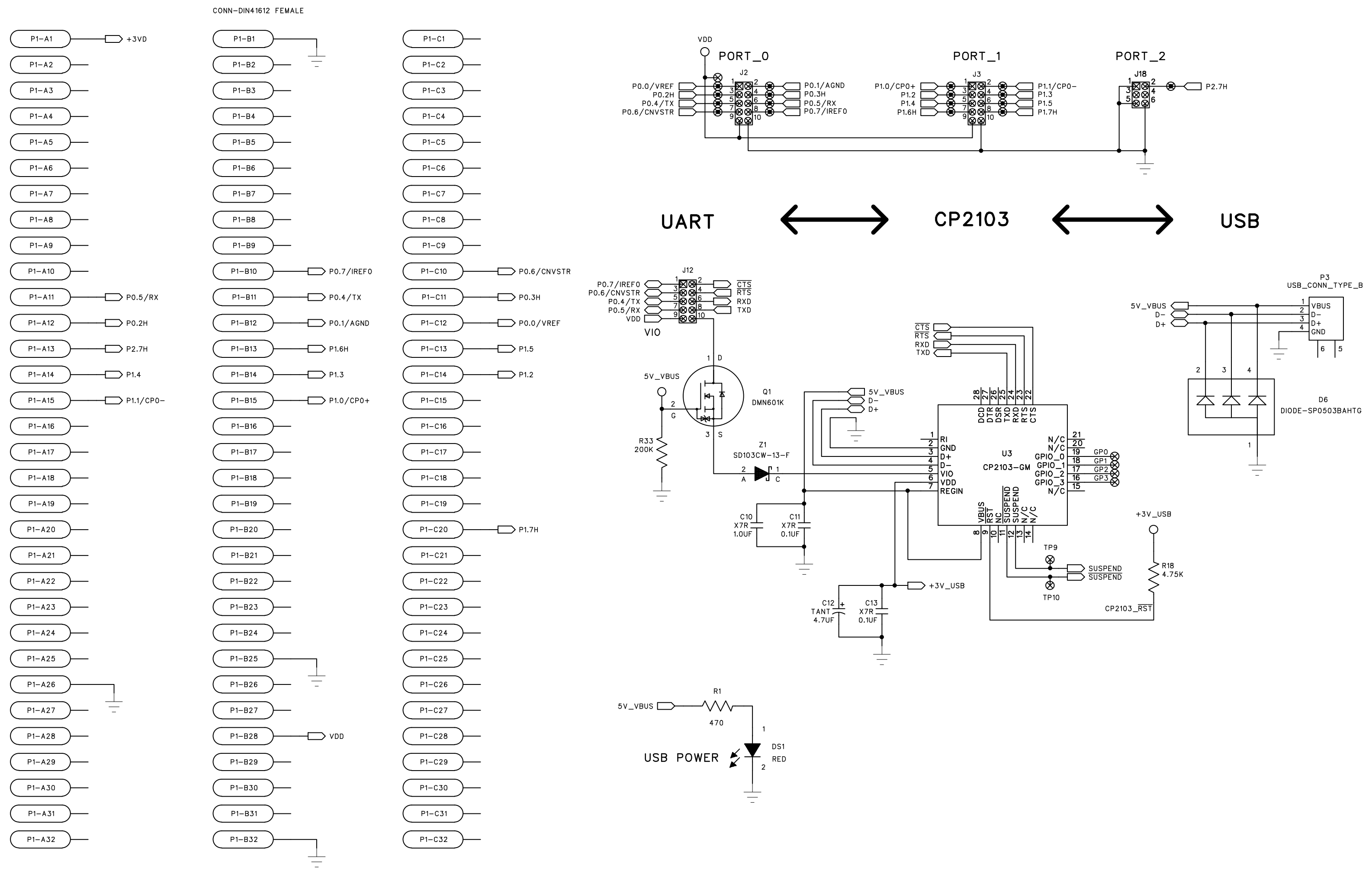


Figure 11. C8051F996 Target Board Schematic (2 of 3)

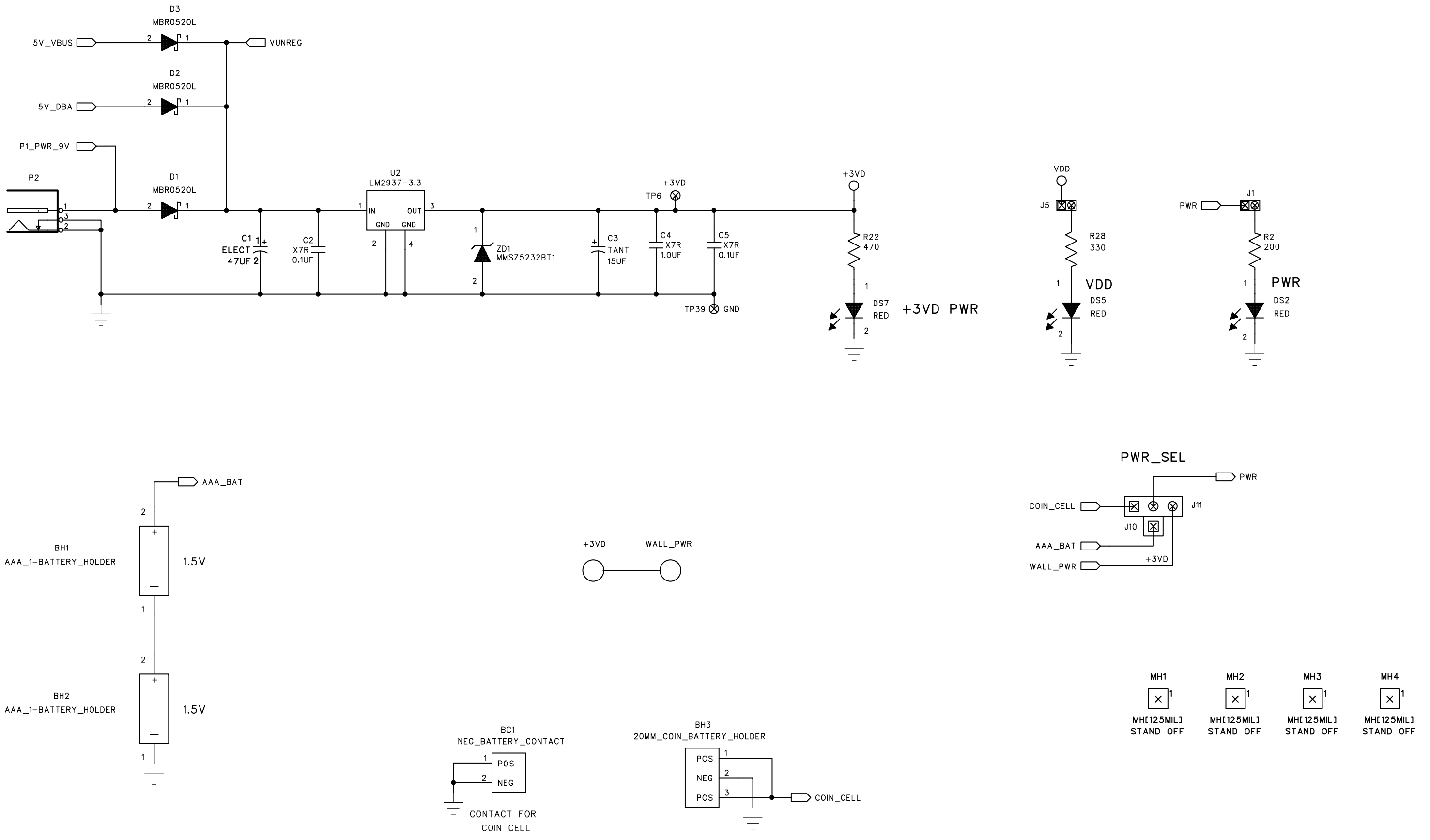


Figure 12. C8051F996 Target Board Schematic (3 of 3)

NOTES:

CONTACT INFORMATION

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