Si825x-DK

Si825x Development Kit User's Guide

1. Kit Contents

The Si825x Development Kit contains the following items:

- 35 W Si8250 Isolated Half-bridge Target Board
- USB to SMBus™ Bridge Board
- USB Debug Adapter
- USB Cable
- Two 9 V, 1.5 A Universal Power Supplies
- Silicon Laboratories IDE and Product Information CD-ROM. CD content includes the following:
  - Silicon Laboratories Integrated Development Environment (IDE)
  - Keil 8051 Development Tools (macro assembler, linker, evaluation 'C' compiler)
  - Source code examples and register definition files
  - Si825x Kernel
  - Application Builder tool suite (waveform editor, compensator, system and MCU wizards)
  - PMBus™ Monitor Software
  - Documentation: Kernel flowcharts and Si825x data sheet
- Si825x Development Kit User's Guide (this document)

Note: The full version of the Kernel when compiled is approximately 14 kB. This exceeds the 4 kB limit of the compiler that is shipped with the kit's software development tools. To avoid a compiler issue, either buy the full Keil compiler toolset or compile the limited version of the Kernel.

2. Hardware Overview

The Si8250 Target Board implements a digitally-controlled, isolated half-bridge current doubler at a PWM frequency of 400 kHz. The target board (Figure 1) contains system power stages and digital control circuits with debug connectors for both the secondary-side controller (Si8250) and the primary-side MCU (C8051F300). The user can directly access the Si8250 using PMBus by connecting a PC to the target board using the included USB to SMBus Bridge Board.

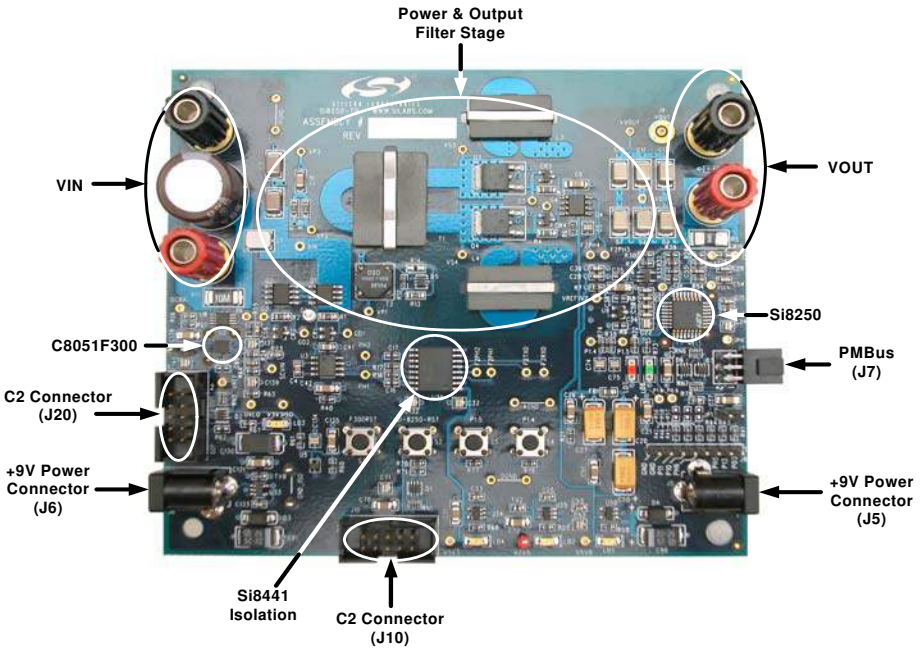

Figure 1. Si8250 Target Board

3. Target Board Stand-Alone Operation

The Si8250 Target Board comes preloaded with SMPS algorithms installed. This dc-dc converter is designed to provide 1.0 V output with up to 35 A of output current.

To operate the target board as a stand-alone power supply:

1. Connect the two 9 V, 1.5 A universal power supplies (supplied with the kit) to J5 and J6 as shown in Figure 2. This supply provides bias for the Si8250 and driver ICs.
2. Connect a third external supply (not provided in kit) to the V_{IN} terminals as shown. This supply needs to provide a voltage between 36–75 V with at least 2 A of current.
3. Connect an electronic load simulator or other load to the V_{OUT} terminals as shown in Figure 2.
4. Turn the V_{IN} supply on. The target board will start and run.

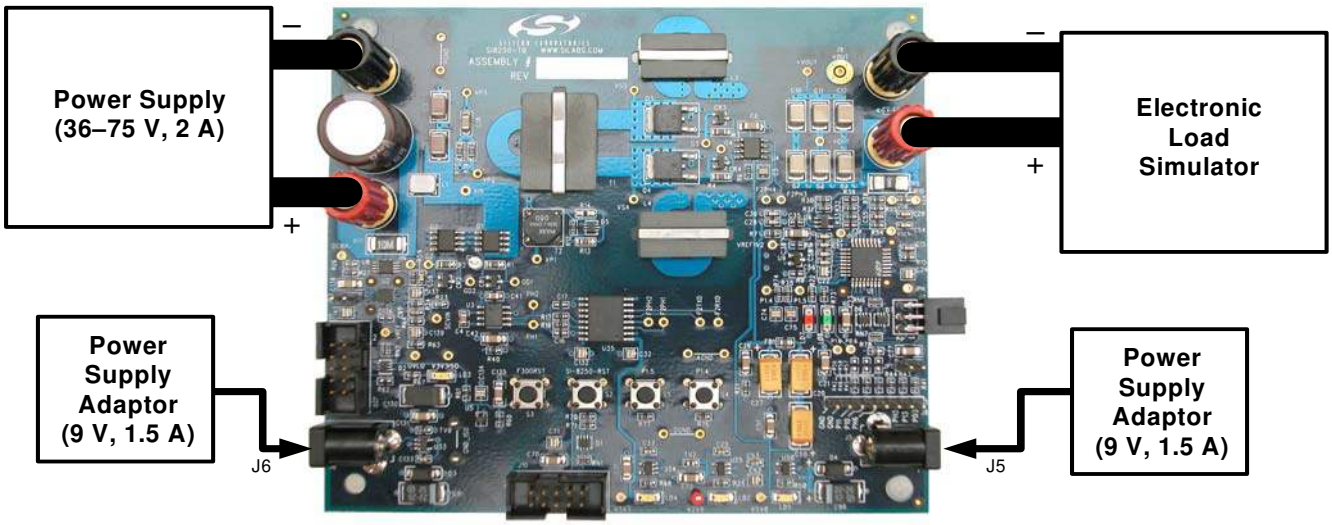

Figure 2. Board Power Configuration

4. Development/Debug Initial Hardware Setup

This section describes use of the Si8250 Target Board with the Silicon Laboratories integrated development environment (IDE) and Application Builder tools. To configure the hardware for connection to the IDE:

1. Disconnect the external supply if connected to V_{IN} and disconnect the load simulator if connected to V_{OUT}.
2. Connect one of the 9 V, 1.5 A universal power supplies (supplied with the kit) to J6. Connect the other 9 V, 1.5 A universal power supply to J5 as shown in Figure 3.These supplies provide bias for the Si825x, driver ICs, and the C8051F300.
3. Connect the USB Debug Adaptor's ribbon cable to the Si8250 Target Board at J10 as shown in Figure 3.
4. Connect the USB cable to the USB Debug Adapter's USB input plug.
5. Connect the USB cable from the USB Debug Adapter to a USB port on the PC.

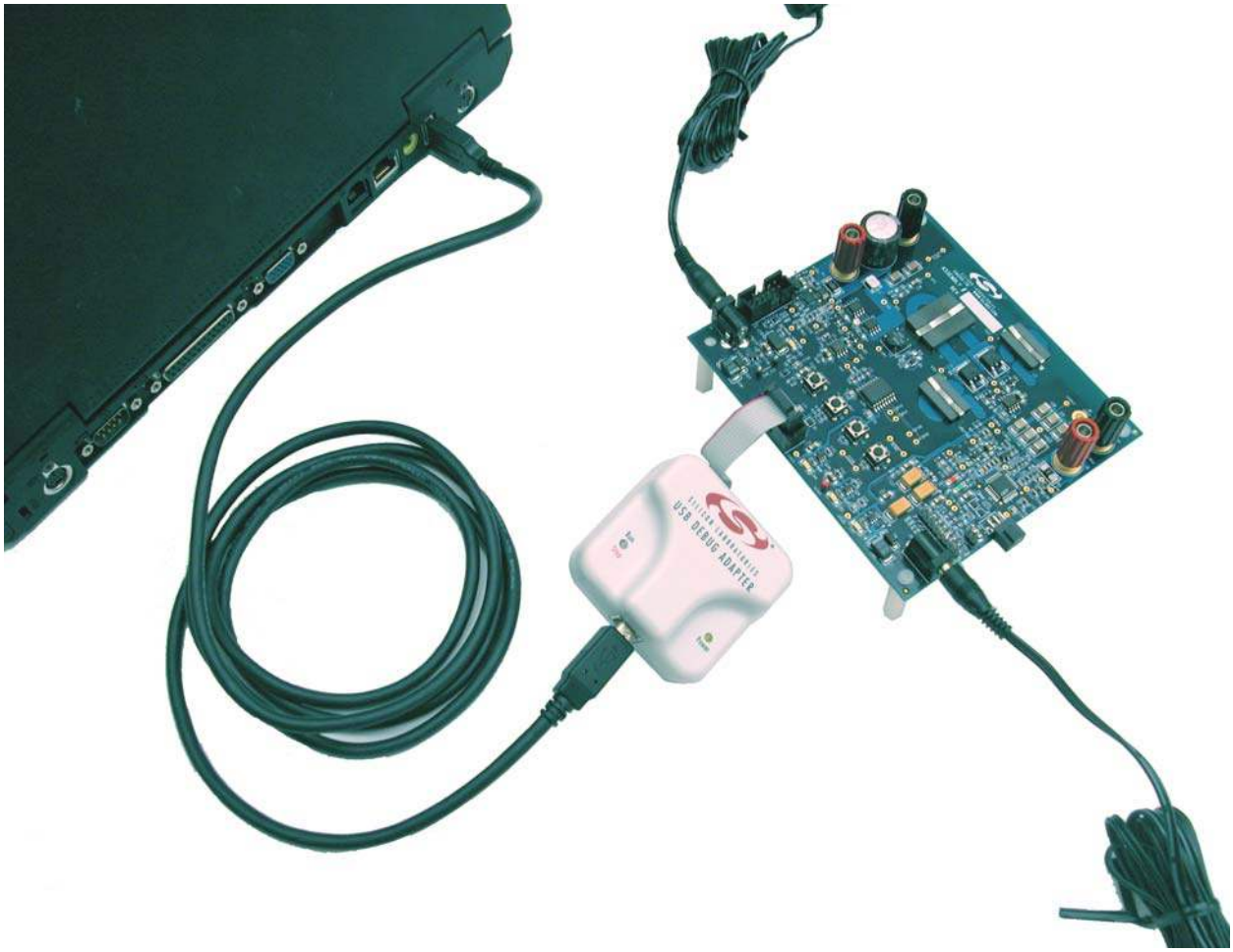

Figure 3. Overall View of the Debug Connection

5. Development/Debug Operation: Software Setup

The Si825x Development Kit comes with Application Builder software (detailed in Section "8. Si825x Application Builder" on page 11), a configurable real-time software Kernel, and a software Kernel compiled specifically for the Si8250 Target Board.

The Kernel is royalty-free application software for the Si825x family of digital power controllers that greatly reduces application program development time, effort, and engineering risk. The Application Builder is used to customize the Kernel and create C-code source level application software for the Si825x end application. The Application Builder directly modifies the source code in the Kernel, which is then compiled and downloaded to the Si825x. For more information on the Kernel, see application note "AN271: Si8250 Real-Time Kernel Overview".

The half-bridge directory *(SiLabs\Power\Si8250_Dev_Kit\Firmware)* contains the Kernel configured for the Si8250 Target Board.

Kernel software can be loaded/reloaded to the target board using the Silicon Laboratories IDE. Note that hardware must be set up as detailed in Section "4. Development/Debug Initial Hardware Setup" on page 3. Follow the instructions below to configure and download the **half-bridge** application software for the Si8250 Target Board.

Note: A thorough understanding of the IDE is required before one can use the development/Debug Mode of the kit. The IDE is detailed in Section "7. Silicon Laboratories Integrated Development Environment".

1. The included CD-ROM contains the Silicon Laboratories Integrated Development Environment (IDE), Application Builder examples, the PMBus Monitor, 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 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. See Section "7. Silicon Laboratories Integrated Development Environment" on page 8 for further information on the development tools.
2. Open the IDE by selecting **Silicon Laboratories→Silicon Laboratories IDE** from the PC programs menu.
3. Next, the example project included with the kit needs to be opened. Select **Project→Open Project...** from the IDE menus. In the Project Workspace window, browse to the "*SiLabs\Power\Si8250_Dev_Kit\Firmware\Half_bridge\source*" directory and select the ***.wsp project** file. Press **Open** to close the window and open the project.

   Note: This example will only work with the full version of the Keil compiler. If the demonstration compiler is used, use the files in "*SiLabs\Power\Si8250_Dev_Kit\Firmware\Half_bridge_basic\source".* This code will compile to less than 4 kB of code.
4. The Si8250 Target Board has several connection requirements that need to be specified before connecting to the board. Select **Options→Connection Options...** from the IDE menu. In the Connection Options window, select USB Debug Adapter in the Serial Adapter section. Next, select C2 in the Debug Interface section. The Si825x family of devices use the Silicon Laboratories 2-wire (C2) debug interface. Click OK to close the window.
5. Click the **Connect** button in the toolbar or select **Debug→Connect** from the menu to connect to the device.
6. Build the project by clicking on the **Build/Make Project** button in the toolbar or by 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.

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. Run the example by pressing the **Go** button in the IDE toolbar.
9. At this point, power can be reapplied to the drivetrain on the Si8250 Target Board. Connect a third external supply (not provided in kit) to the V_{IN} terminals as shown in Figure 2. This supply needs to provide a voltage of between 36–75 V with at least 2 A of current.
10. Connect an electronic load simulator to the V_{OUT} terminals as shown in Figure 2.
11. Turn the V_{IN} supply on. The isolated half-bridge supply will start and run.
12. Save the project when finished with the debug session to preserve the current target build configuration, editor settings, and location of all open debug views. To save the project, select **Project→Save Project** from the menu.

6. PMBus Operation

PMBus is a connectivity solution designed for connecting power supplies to a single management bus. The Real-time Kernel provided with the Si8250 design kit includes optional support for PMBus. In addition, the Si825x design kit also comes with a PMBus monitor application and USB to SMBus Bridge Board to manage the power supply through PMBus.

1. The PMBus Monitor software is installed during the initial software setup. (See Section "5. Development/Debug Operation: Software Setup" on page 4.)
2. Connect the board as shown in Section "3. Target Board Stand-Alone Operation" on page 2. Note that the PMBus monitor may also be operated with the USB Debug Adaptor. If this is desired, connect the target board as shown in Section "4. Development/Debug Initial Hardware Setup" on page 3.
3. Drivers must be installed to allow the PMBus Monitor to communicate with the USB to SMBus Bridge Board. The driver files are located by default in the "*Silabs\Power\Si825x AppBuilder\PMBus Monitor\USB-SMBus Bridge Board Drivers*" directory. Run the *PreInstaller.exe* application. This program will copy the driver files to the PC's "Program Files" directory and then will register the driver files so the board will be recognized when it is connected. Windows Logo testing warnings may appear. Press the "**Continue Anyway**" button.
4. Connect the USB to SMBus Bridge Board to an available USB slot on your PC with a USB cable.
5. Windows will open a "**Found New Hardware Wizard**" window. Press "**Next**" after selecting the (Recommended) option. Windows Logo testing warnings may appear. Press the "**Continue Anyway**" button. Press "**Finish**" to finish installing the USB-SMBus Bridge Board.
6. Connect the Si8250 Target Board to the USB to SMBus Bridge Board as shown in Figure 4.

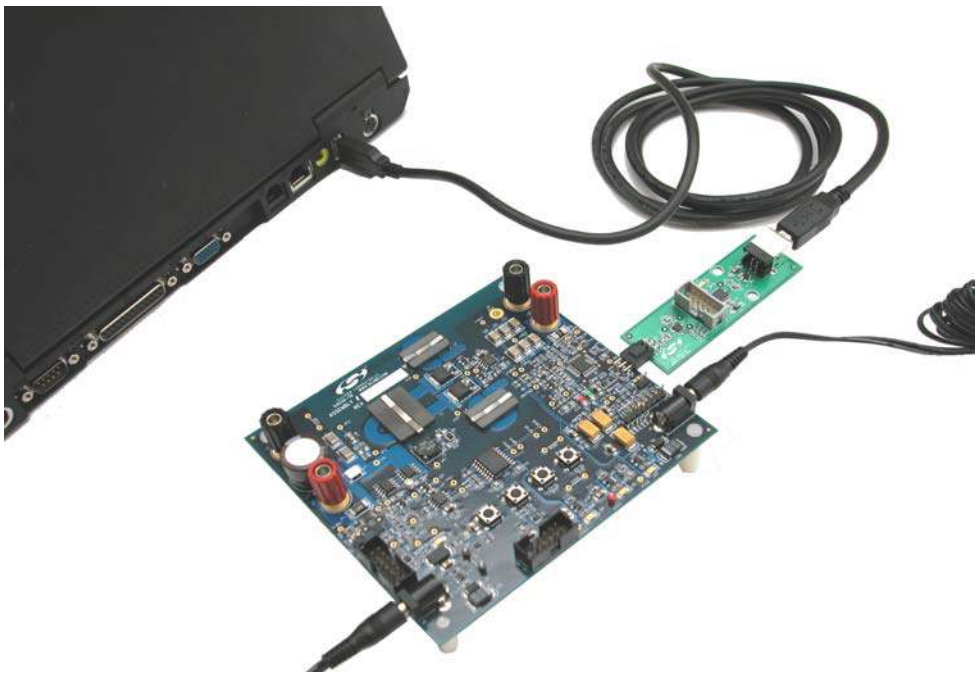

Figure 4. PMBus to USB Adapter Connection

7. Open the Application Builder by selecting **Silicon Laboratories→Si825x Application Builder** from the PC programs menu.
8. Run the PMBus Monitor application by selecting **Options→Launch PMBus Monitor Tool** from the Application Builder. The window shown in Figure 5 will appear. The PMBus Monitor can be used to control and configure the target board. The target can be enabled/disabled through the monitor. The PMBus Monitor allows parameters, such as fault thresholds, to be changed. It also reports operating conditions and problems.

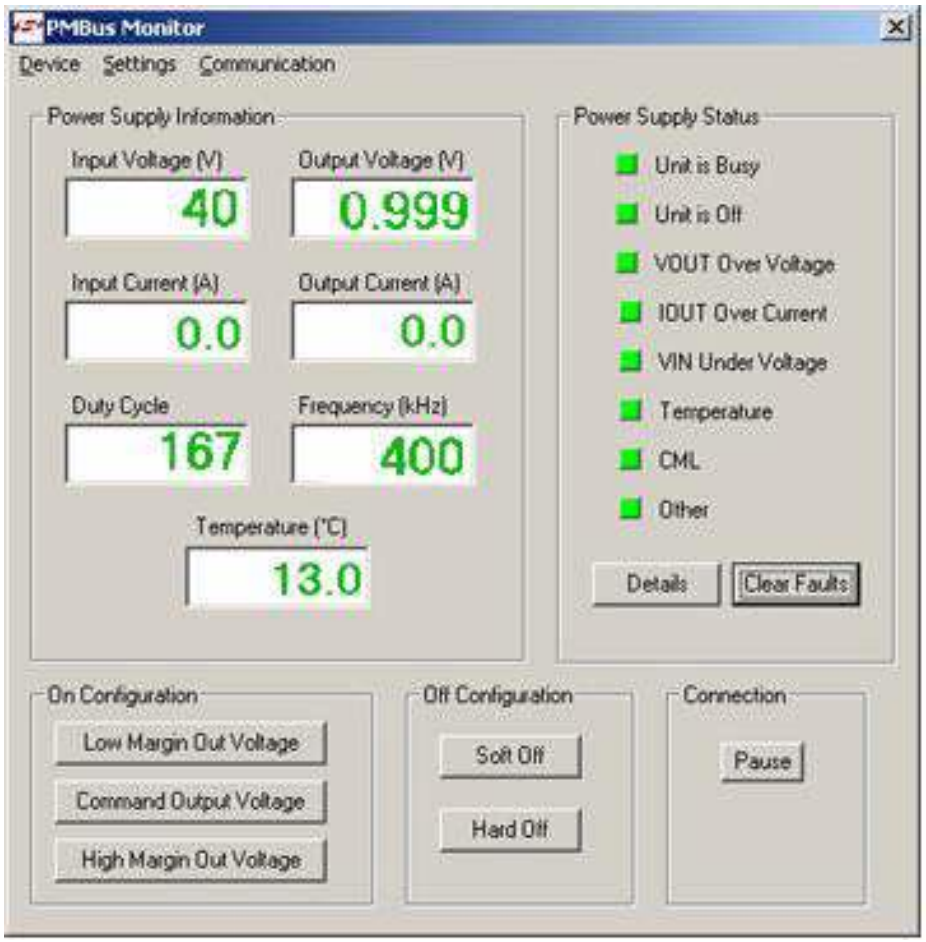

Figure 5. View of the PMBus Monitor

7. Silicon Laboratories Integrated Development Environment

The Silicon Laboratories IDE combines an editor, project manager, code development tools, and a debugger into a single, intuitive environment for code development and in-system debugging. No additional target RAM, program memory, or communications channels are required. The use of third-party compilers and assemblers is also supported. This development kit includes the Keil Software A51 macro assembler, BL51 linker, and evaluation version C51 'C' compiler. These tools can be used from within the Silicon Laboratories IDE. Figure 6 illustrates the IDE.

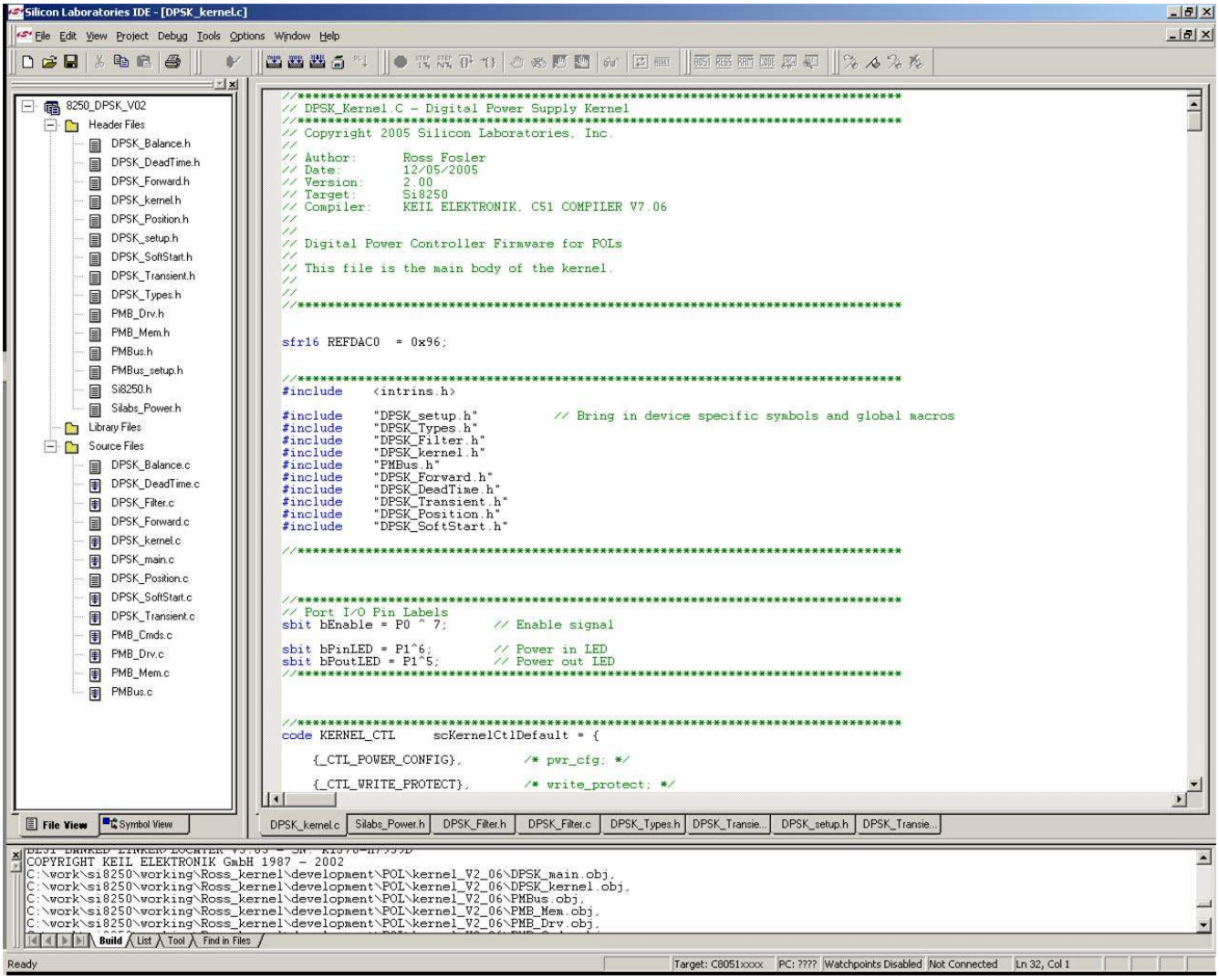

Figure 6. IDE

7.1. System Requirements

The Silicon Laboratories IDE requirements:

- Pentium-class host PC running Microsoft Windows 95 or later or Microsoft Windows NT or later.
- One available USB port.
- 64 MB RAM and 40 MB free HD space recommended.

7.2. Assembler and Linker

A full-version Keil A51 macro assembler and BL51 banking linker are included with the development kit and are installed during IDE installation. The complete assembler and linker reference manual can be found online under the **Help** menu in the IDE or in the "*SiLabs\MCU\hlp*" directory (A51.pdf).

7.3. Evaluation C51 'C' Compiler

An evaluation version of the Keil C51 'C' compiler is included with the development kit and is installed during IDE installation. The evaluation version of the C51 compiler is the same as the full professional version except that code size is limited to 4 kB, and the floating point library is not included. The C51 compiler reference manual can be found under the **Help** menu in the IDE or in the "*SiLabs\MCU\hlp*" directory (C51.pdf).

7.4. Using the Keil Software 8051 Tools with the Silicon Laboratories IDE

To perform source-level debugging with the IDE, you must configure the Keil 8051 tools to generate an absolute object file in the OMF-51 format with object extensions and debug records enabled. You may build the OMF-51 absolute object file by calling the Keil 8051 tools at the command line (e.g. batch file or make file) or by using the project manager built into the IDE. The default configuration when using the Silicon Laboratories IDE project manager enables object extension and debug record generation. Refer to application note "AN104: Integrating Keil 8051 Tools Into the Silicon Labs IDE" 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.)

7.4.1. Creating a New Project

1. Select **Project→New Project** to open a new project and reset all configuration settings to default.
2. Select **File→New File** to open an editor window. Create your source file(s) and save the file(s) with a recognized extension, such as *.c, *.h, or *.asm, to enable color syntax highlighting.
3. Right-click on "New Project" in the **Project Window**. Select **Add files to project**. Select files in the file browser and click Open. Continue adding files until all project files have been added.
4. For each of the files in the **Project Window** that you want assembled, compiled, and linked into the target build, right-click on the file name and select **Add file to build**. Each file will be assembled or compiled as appropriate (based on file extension) and linked into the build of the absolute object file.

   Note: If a project contains a large number of files, the "Group" feature of the IDE can be used to organize. Right-click on "New Project" in the **Project Window**. Select **Add Groups to project**. Add predefined 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.

7.4.2. Building and Downloading the Program for Debugging

1. Once all source files have been added to the target build, build the project by clicking on the **Build/Make Project** button in the toolbar or selecting **Project→Build/Make Project** from the menu.

   Note: After the project has been built the first time, the **Build/Make Project** command will only build the files that have been changed since the previous build. To rebuild all files and project dependencies, click on the **Rebuild All** button in the toolbar or select **Project->Rebuild All** from the menu.

2. Before connecting to the target device, several connection options may need to be set. Open the **Connection Options** window by selecting **Options→Connection Options...** in the IDE menu. First, select the adapter that was included with the kit in the "Serial Adapter" section. Next, the correct "Debug Interface" must be selected. Si825x family devices use the Silicon Laboratories 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**.

7.5. Si825x Debug Mode

The IDE contained in the Si8250 Development Kit has an Online Debug feature, which optionally enables the user to inspect or update special function registers (SFRs) in the Si8250 while it is operating. For example, filter coefficients can be optimized by simply typing in new coefficient values while the supply is connected to a network analyzer and running. The IDE can also be operated in Standard mode where SFR inspect and update is allowed only when the Si8250 is not running. As shown in Figure 7, the IDE can be set for Online Debug Mode or Standard Debug mode by clicking on the circled mode switch.

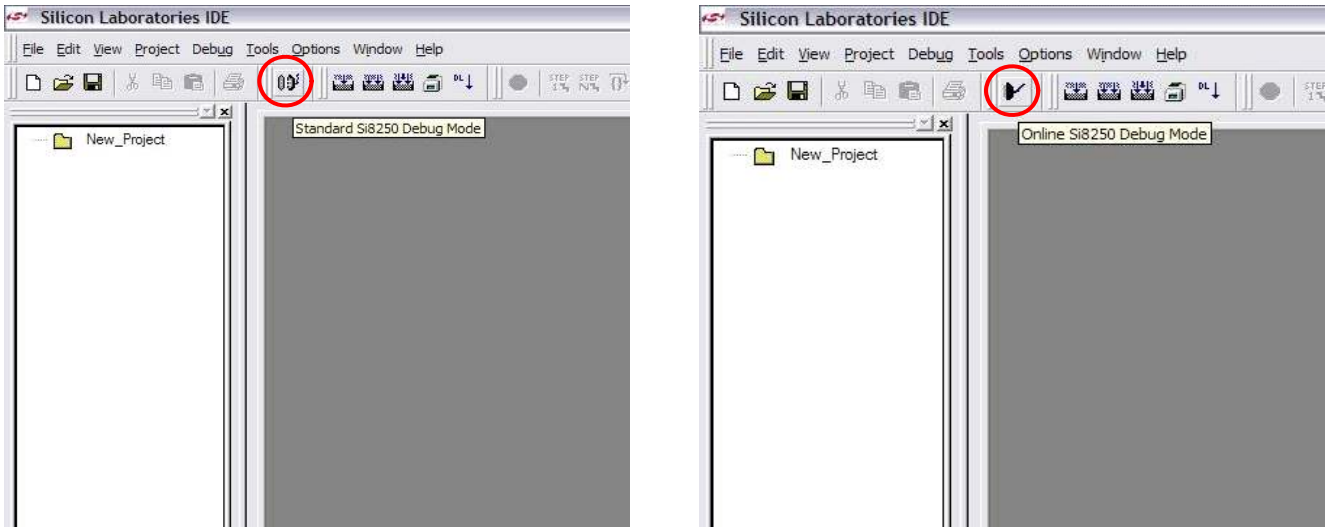

Figure 7. Si825x IDE Debug Modes

8. Si825x Application Builder

In addition to the IDE, the Si825x family is also supported with an intuitive toolset that leverages traditional power supply control design methods minimizing the digital supply design learning curve. The toolset consists of a real-time firmware Kernel (C-language source code), and the **Application Builder** (see Figure 8). The Application Builder includes device peripheral configuration options with the **Peripheral Configuration Wizard**. Additionally, the Application Builder tools are used to modify the Kernel. Three key development tools within the Application Builder are the **DPWM Timing Diagram Editor**, the **System Parameter Programmer**, and the **Compensation Editor**. These tools are detailed in the following sections. The flow diagrams for the Kernel are included on the CD.

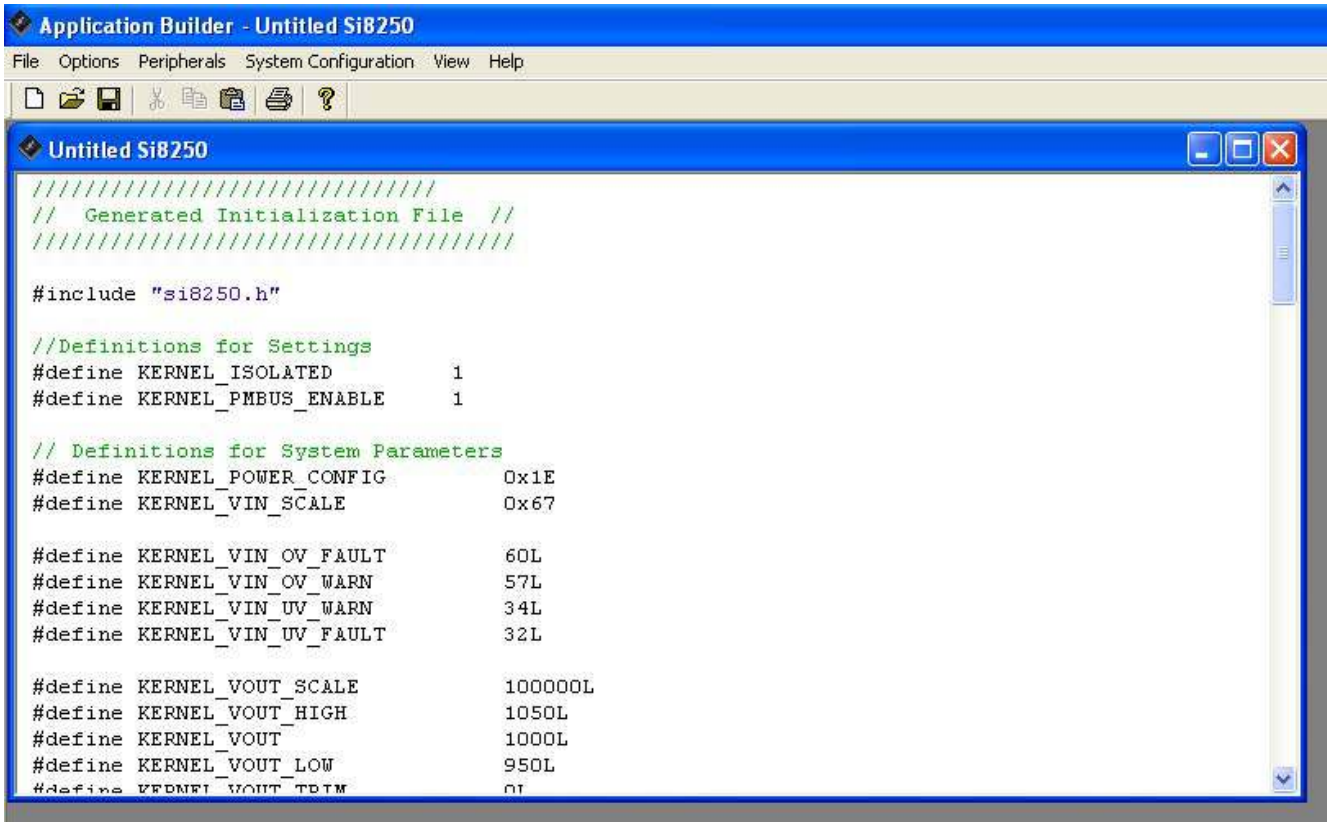

Figure 8. Si825x Application Builder

8.1. DPWM Timing Diagram Editor

The **DPWM Timing Diagram Editor** permits designers to generate DPWM initialization code by simply drawing the timing for their end system. The wizard accommodates up to six output phases and can be used to establish positive or negative dead-times, relative edges, absolute edges, and other timing required by the end system. Refer to the Si825x data sheet for a description of the different edges.

This example illustrates how to use the **DPWM Timing Diagram Editor** to create and simulate the timing for a half-bridge as well as generate the initialization code in the Kernel.

1. To open the **DPWM Timing Diagram Editor** window, open the Application Builder and select **System Configuration→DPWM Timing Diagram Editor** from the menu.
2. To create an absolute edge on Phase 1, hold the mouse above the Phase 1 zero timing line (default) at time tick 10. Then, either double click with the left mouse button or right-click at that point and select **Absolute Edge**. An absolute edge at time tick 10 will be created (see Figure 9).
3. To finish the timing for Phase 1, specify hardware modulation using **(Cu0)**. This event edge will be modulated relative to its absolute edge at time tick 10. To create this edge, hold the mouse above the Phase 1 timing line to the right of time tick 10 at time tick 60. Either double-click or right-click at that point and select **Event (Cu0) Edge**. Next, select the edge to reference by clicking on edge 1 in Phase 1. A relative falling (Cu0) edge will be created at time tick 60 since U(n) defaults to 50 (see Figure 9).

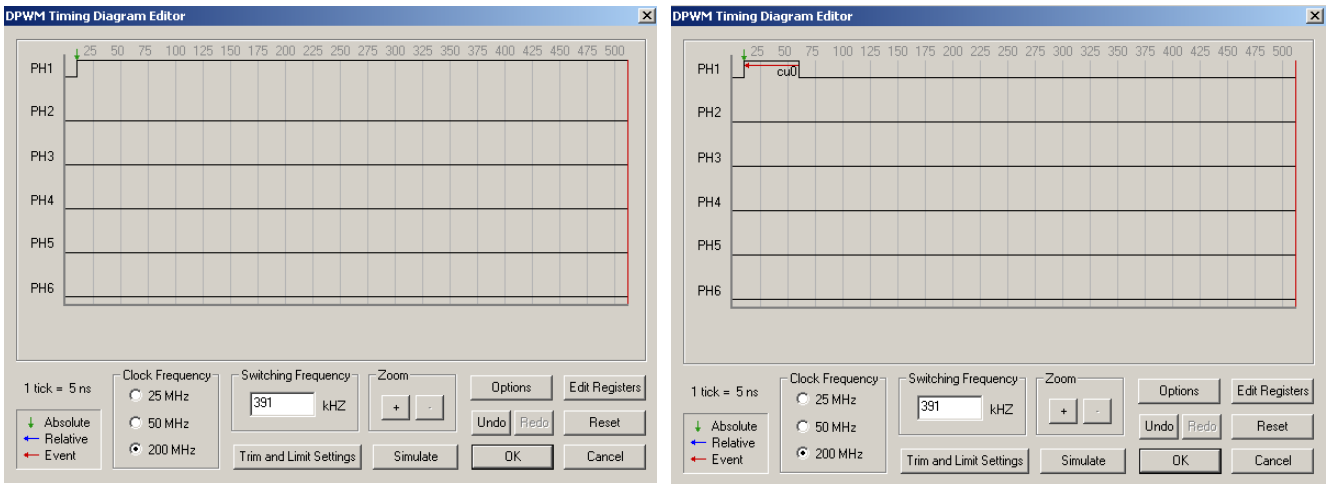

Figure 9. DPWM Timing Diagram Editor—Phase 1

1. Now, create the timing for Phase 2. For Phase 2, the goal is to create a relative rising edge relative to the falling edge of Phase 1 and an absolute falling edge on Phase2 at time tick 505. To create the relative rising edge, hold the mouse above the Phase 2 zero timing line (default) at time tick 100. Then, either double-click with the left mouse button or right-click at that point and select **Relative Edge**. Next, select the edge to reference by clicking on Phase 1's falling edge. A relative edge at approximately time tick 90 will be created (see Figure 10).
2. To finish the timing for Phase 2, create an absolute falling at time tick 505. To create this edge, hold the mouse above the Phase 2 timing line at time tick 505. Then, either double-click with the left mouse button or right-click at that point and select **Absolute Edge**. An absolute edge at time tick 505 will be created (see Figure 10).

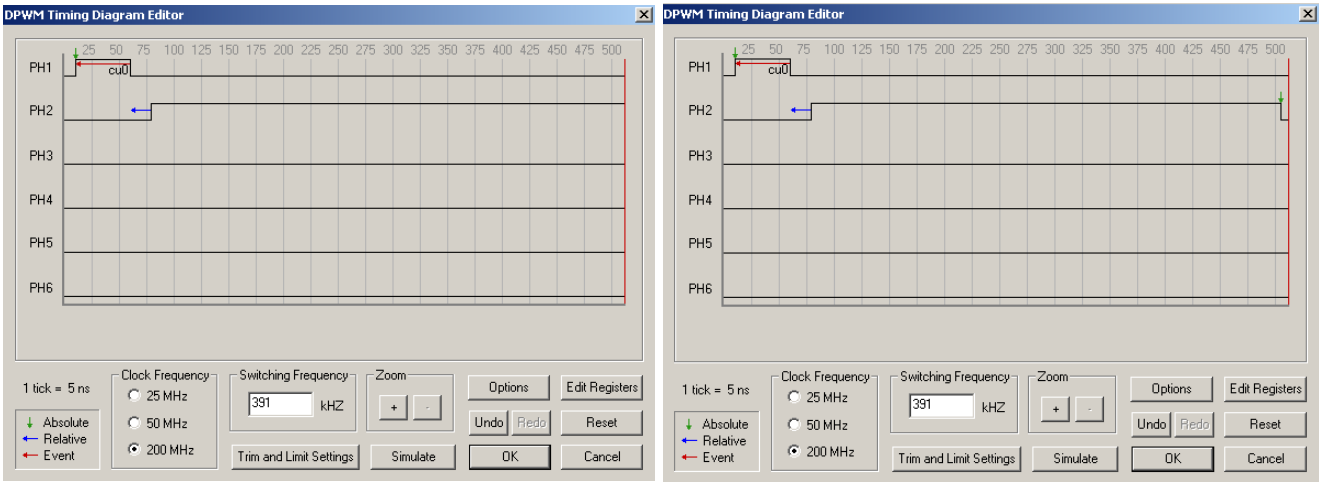

Figure 10. DPWM Timing Diagram Editor—Phase 2

3. Click the **Simulate** button to display the Simulate Window (see Figure 11). Use the arrows to increase and decrease the value of U(n). Notice that Phase 1's edge should modulate with its absolute edge starting at 10 ticks. Phase 2 should also modulate. However, its rising edge starts relative to Phase 1's falling edge.

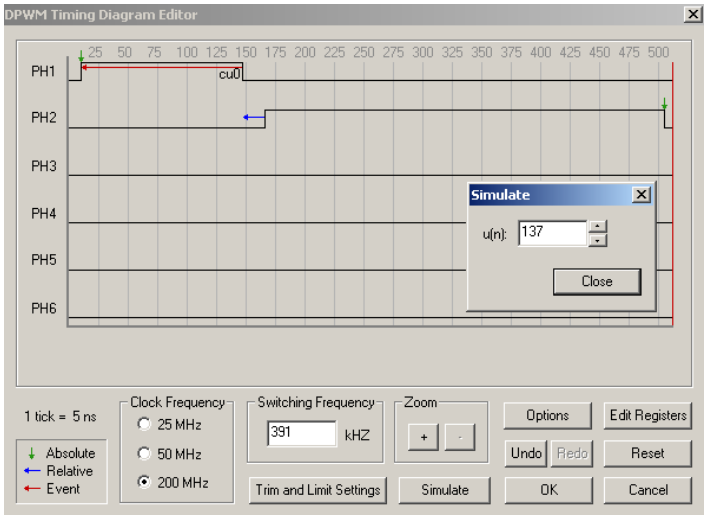

Figure 11. DPWM Timing Diagram Editor—Timing Simulator

4. Now that the desired timing for the two-phase system has been created, click on **OK**. The Application Builder will automatically extract the correct DPWM timing initialization data for the Kernel. These coefficients can be saved to a new project file if desired for later use. Click on **File→Save Project** to save this project. To generate an IDE project, select **File→Build IDE Project...** and select the directory for project generation.

8.2. Compensation Editor

The **Compensation Editor** is a loop simulation and coefficient generation tool for frequency compensating the system. To open the **Compensation Editor** window, open the Application Builder and select **System Configuration→Compensation Editor** from the menu. The Compensation Editor for a buck regulator is shown in Figure 12. As shown, this tool provides fields for the user to enter power stage parameters, such as the output filter component and parasitic values; controller parameters, such as PWM frequency and pole/zero locations; and Si8250-specific data, such as ADC sample frequency. The simulator comes prepopulated with default values for the Si8250 Target Board. These model parameters can be changed as desired. To view the gain and phase plots with the default parameters, click on the **View Graphs** button, and their plots will be generated a short time later. Moreover, clicking on the **View Graphs** button, the user can view different responses of the buck regulator. Sample graphs are shown in Figure 12.

Table 1 shows two frequency responses: one for steady-state operation and one for operation during a transient. The Si8250 automatically extends loop bandwidth by writing faster coefficients to the loop compensation filter (DSP filter engine). This nonlinear control response improves system transient response, reducing both the magnitude and duration of the output transient.

Table 1. Si8250 Target Board Frequency Response

	Steady-State Response	Transient Response
Loop Gain Bandwidth	40 kHz	64 kHz
Phase Margin	82 degrees	10 degrees

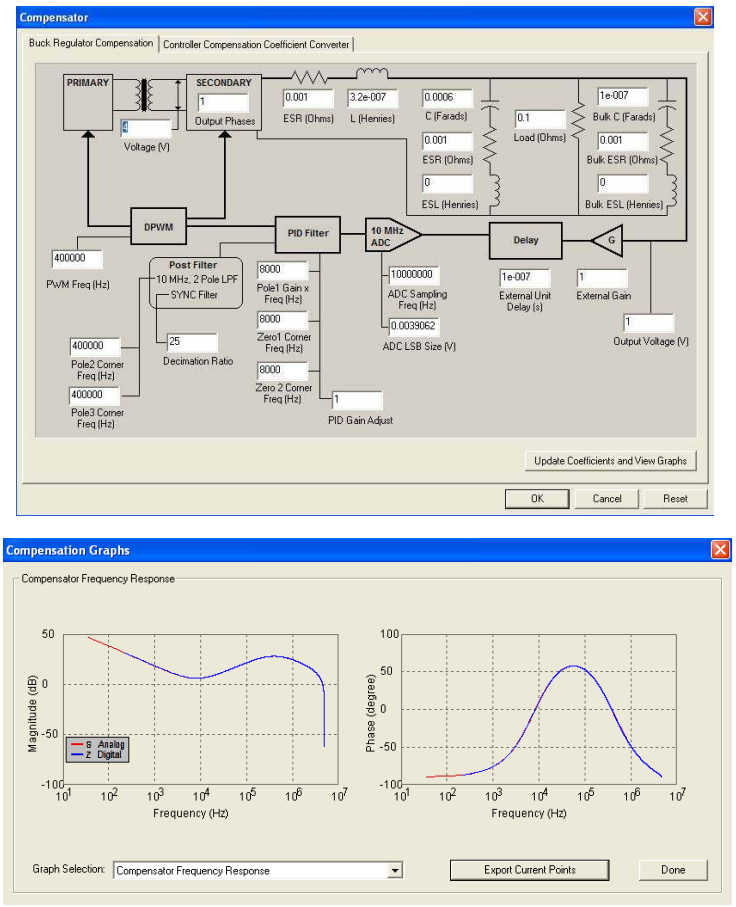

Figure 12. Compensation Editor, Input/Output Windows

8.3. System Parameter Programmer

The **System Setting Programmer** allows the designer to input all system settings (UVLO, OV, OCP, etc.) and then converts these parameters to HEX and populates the resulting initialization code in the Kernel. To open the **System Settings** window, open the Application Builder and select **System Configuration→System Settings** from the menu (see Figure 13).

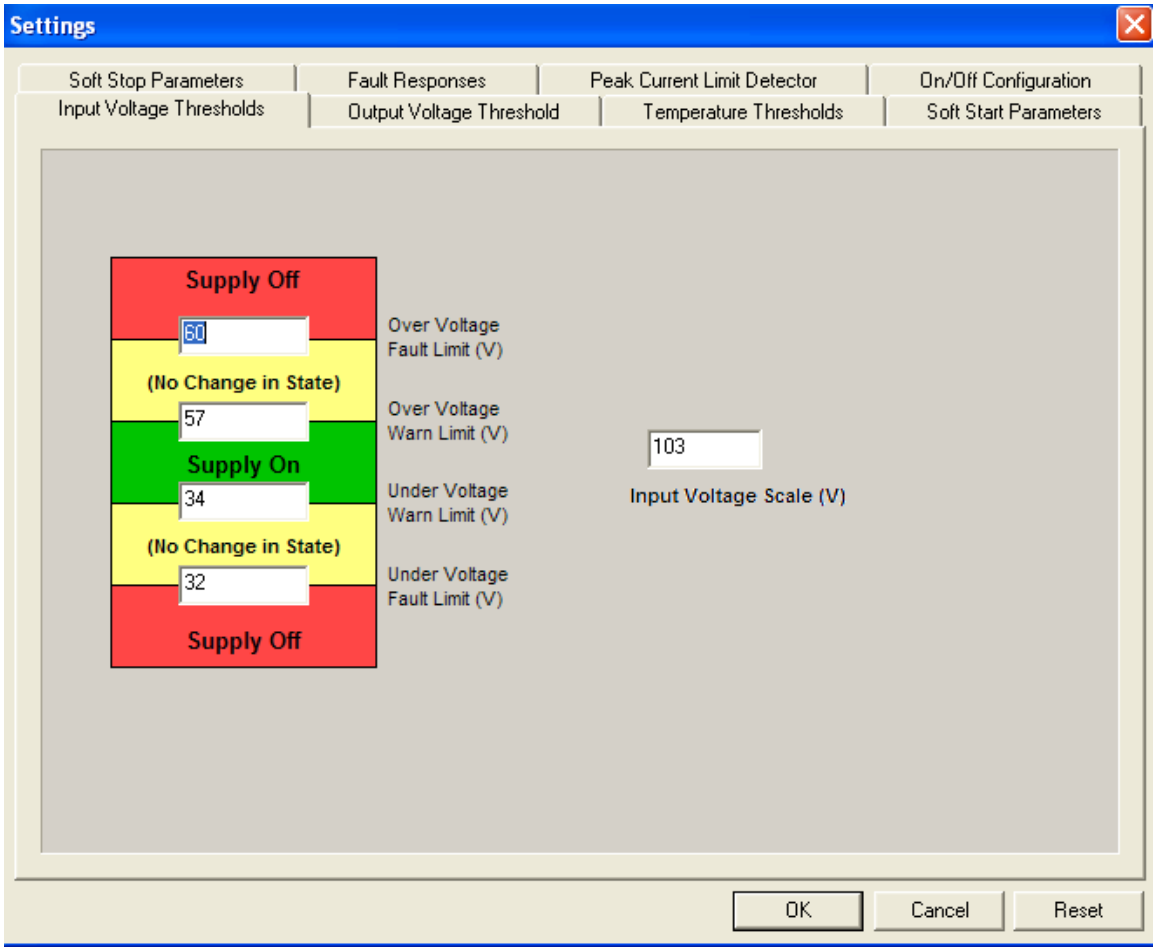

Figure 13. System Parameters Window

8.4. Peripheral Configuration Wizard

The **Peripheral Configuration Wizard** can be used to automatically generate initialization code for the Si8250's on-chip peripherals (ADC2, comparator, UART, SMBus™, etc.). The peripheral windows can be accessed by clicking on the **Peripherals** menu in the Application Builder. Figure 14 illustrates the **Port I/O** window. For more details on using this wizard, consult the "Help" file by clicking on **Help→Help...**.

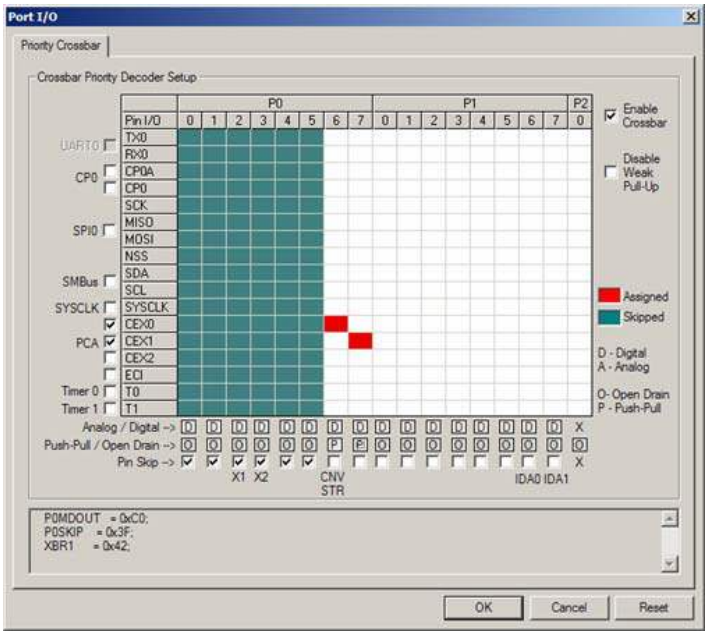

Figure 14. Peripheral Configuration Wizard - Port I/O Window

9. Restoring Factory Defaults

The Si825x Development Kit includes hex files created for the isolated half-bridge application. Downloading these hex files to the Si8250 Target Board will restore the board to its factory defaults.

9.1. Restoring the Si8250

To download the factory default Si8250 hex file (secondary side) to the target board, perform the following steps:

1. Open the IDE by selecting **Silicon Laboratories→Silicon Laboratories IDE** from the PC programs menu.
2. The Si8250 Target Board has several connection requirements that need to be specified before connecting to the board. Select **Options→Connection Options...** from the IDE menu. In the **Connection Options** window, select **USB Debug Adapter** in the Serial Adapter section. Next, select **C2** in the Debug Interface section. The Si825x family of devices use the Silicon Laboratories 2-wire (C2) debug interface. Press OK to close the window.
3. Connect the USB Debug Adaptor's ribbon cable to the Si8250 Target Board at J10 as shown in Figure 3.
4. Click the **Connect** button in the toolbar or select **Debug→Connect** from the menu to connect to the device.
5. Select **Debug→Download Object File...** from the IDE menus to open the download window.
6. Press the **Browse** button to open the **Download Filename...** window.
7. In the **List files of type:** drop down box, select the **Intel-Hex** option.
8. Browse to the *"SiLabs\Power\Si8250_Dev_Kit\Firmware\Half_bridge\hex"* directory and select the ***.hex** file. Press **OK** to close the window.
9. Press the **Download** button to download the file.
10. Click the **Disconnect** button in the toolbar, or select **Debug→Disconnect** from the menu to disconnect from the device.
11. Power cycle the device to run the downloaded program.

9.2. Restoring the C8051F300

To download the factory default C8051F300 hex file (primary side) to the target board, perform the following steps:

1. Open the IDE by selecting **Silicon Laboratories→Silicon Laboratories IDE** from the PC programs menu.
2. The Si8250 Target Board has several connection requirements that need to be specified before connecting to the board. Select **Options→Connection Options...** from the IDE menu. In the **Connection Options** window, select **USB Debug Adapter** in the Serial Adapter section. Next, select **C2** in the Debug Interface section. The Si825x family of devices use the Silicon Laboratories 2-wire (C2) debug interface. Press OK to close the window.
3. Remove jumper JP3.
4. Connect the USB Debug Adaptor's ribbon cable to the Si8250 Target Board at J20.
5. Click the **Connect** button in the toolbar or select **Debug→Connect** from the menu to connect to the device.
6. Select **Debug→Download Object File...** from the IDE menus to open the download window.
7. Press the **Browse** button to open the **Download Filename...** window.
8. In the **List files of type:** drop down box, select the **Intel-Hex** option.
9. Browse to the *"SiLabs\Power\Si8250_Dev_Kit\Firmware\half_bridge_iso\hex"* directory and select the ***.hex** file. Press **OK** to close the window.
10. Press the **Download** button to download the file.
11. Replace jumper JP3.
12. Click the **Disconnect** button in the toolbar, or select **Debug→Disconnect** from the menu to disconnect from the device.
13. Press switch **S3**, 'F300 RST button, and the downloaded firmware will begin to execute.

10. Si8250 Target Board

The Si8250 Target Board has a Si8250-IQ installed. Refer to Figure 15 for the locations of the various I/O connectors and major components.

J1, J2	V_{IN}, Primary power connection 36–75 V, 2 A
J3, J4	V_{OUT}, Supply output connection for load simulator
J5	Secondary side power connector (9 V, 1.5 A power supply adaptor)
J6	Primary side power connector (9 V, 1.5 A power supply adaptor)
J7	PMBus adapter
J8	I/O header
J10	Si8250 Debug Interface
J20	C8051F300 Debug Interface
JP1	Si8250 ENABLE Polarity Select
JP3	C8051F300 Reset Capacitor

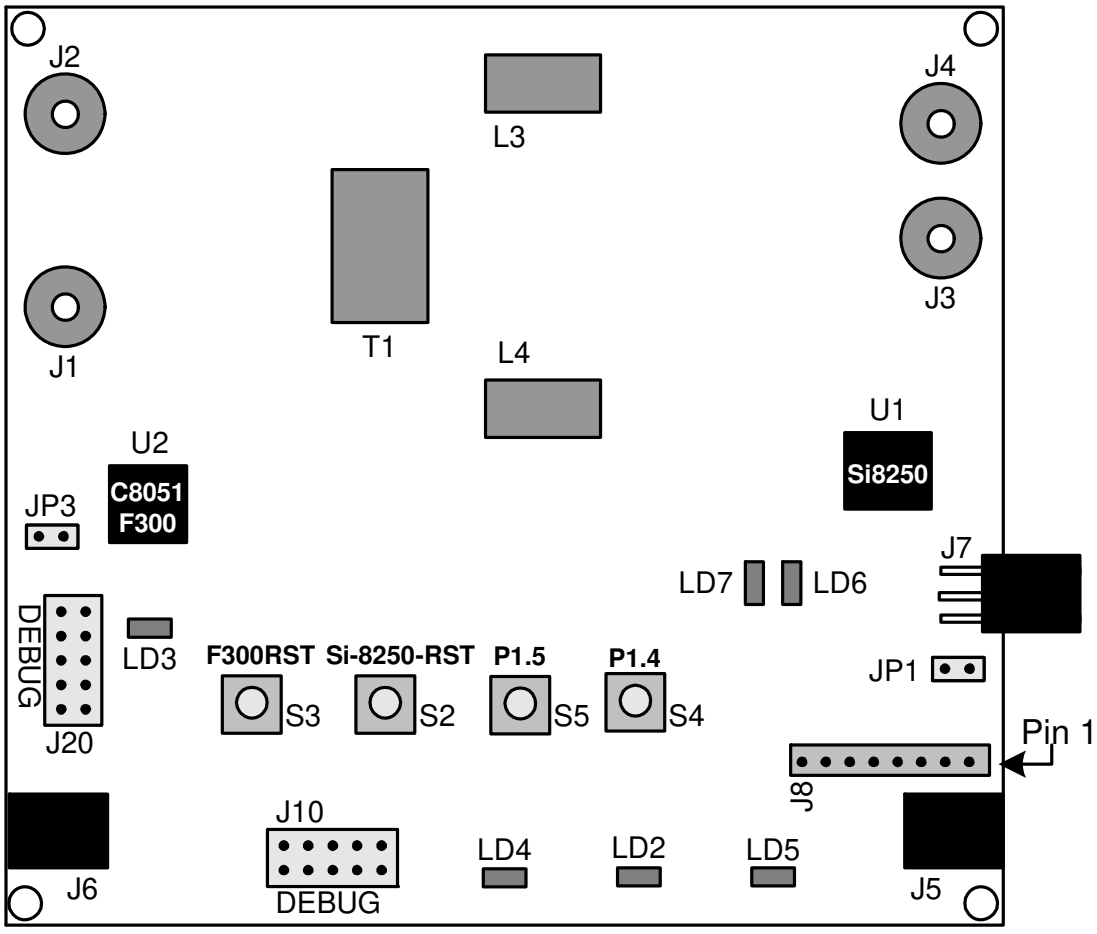

Figure 15. Si8250 Target Board

10.1. System Clock Sources

The Si8250-IQ device installed on the target board features a calibrated programmable internal oscillator that is enabled as the system clock source on reset. After reset, the device operates at a frequency of 80 kHz by default using the internal low-frequency oscillator but may be configured by software to operate at other frequencies. Refer to the Si825x family data sheet for more information on configuring the system clock source.

10.2. Switches and LEDs

Four switches are provided on the target board. Switch S2 is connected to the RESET pin of the Si8250. Switch S3 is connected to the RESET pin of the C8051F300. Pressing S2 or S3 puts the corresponding device into its hardware-reset state. Switches S4 and S5 are connected to two of the Si8250's general-purpose I/O (GPIO) pins, P1.4 and P1.5. Pressing S4 or S5 generates a logic low signal on the port pin. See Table 2 for descriptions of each switch.

Six LEDs are also provided on the target board. The red LED is used to indicate communications between the C8051F300 and the Si8250 devices on the target board. The green LED is used to indicate that power is being regulated by the Si8250. The yellow LEDs are used to indicate power connections to the target board. See Table 2 for a description of each LED.

Table 2. Target Board Switch and LED Descriptions

Label	Reference	Description
F300RST	S3	C8051F300 Reset Switch
Si-8250-RST	S2	Si8250 Reset Switch
P1.5	S5	Si8250 GPIO port pin P1.5 Switch
P1.4	S4	Si8250 GPIO port pin P1.4 Switch
none	LD7	Si8250/C8051F300 Communications Red LED
none	LD6	Power Regulation in process Green LED
V3V3ISO	LD3	3.3 V to C8051F300 from J6 Yellow LED
V3V3	LD4	3.3 V to Si8250 from J5 Yellow LED
V2V5	LD2	2.5 V to Si8250 from J5 Yellow LED
V5VO	LD5	5 V to Si8250 from J5 Yellow LED

10.3. V_{IN} (J1, J2)

The user-provided power supply should be connected to connectors J1 and J2 where J2 is the reference. The power source must be from 36–75 V with at least 2 A maximum output.

Table 3. J7 Pin Descriptions

Jumper #	Description
J1	36–75 V, 2 A
J2	GND

10.4. V_{OUT} (J3, J4)

Connectors J3 and J4 are the dc output from the half-bridge. A load should be connected, preferably an electronic load.

Table 4. J7 Pin Descriptions

Jumper #	Description
J3	1 V (Nominal), 35 A max
J4	Isolated GND

10.5. PMBus Connector (J7)

The J7 connector is the PMBus interface connector for the Si8250 Target Board. Table 5 shows the J7 pin definitions.

Table 5. J7 Pin Descriptions

Pin #	Description
1	SCL
2	DGND
3	SDA
4	DGND
5	SMBA
6	DGND

10.6. I/O Connector (J8)

The J8 connector provides access to signals that are not used in the isolated half-bridge converter. These pins can be used for other purposes, such as debugging or prototyping. Table 6 shows the J8 pin definitions

Table 6. J8 Pin Descriptions

Pin #	Description
1	P0.3
2	P1.3
3	PH5
4	PH6
5	P1.0
6	P1.1
7	GND
8	GND

10.7. Si8250 DEBUG Interface (J10)

The Si8250 DEBUG connector (J10) provides access to the DEBUG (C2) pins of the Si8250 device on the target board. It is used to connect the USB Debug Adapter to the target board for in-circuit debugging and programming. Table 7 shows the J10 Si8250 DEBUG pin definitions.

Table 7. J10 Si8250 DEBUG Connector Pin Descriptions

Pin #	Description
1	+2.5 V
2, 3, 9	DGND
4	C2D
7	RST/C2CK
5, 6, 8, 10	Not Connected

10.8. C8051F300 DEBUG Interface (J20)

The C8051F300 DEBUG connector (J20) provides access to the DEBUG (C2) pins of the C8051F300 device on the target board. It is used to connect the USB Debug Adapter to the target board for in-circuit debugging and programming. Table 8 shows the J20 C8051F300 DEBUG pin definitions.

Table 8. J20 C8051F300 DEBUG Connector Pin Descriptions

Pin #	Description
1	+2.5 V
2, 3, 9	DGND
4	C2D
7	RST/C2CK
5, 6, 8, 10	Not Connected

10.9. Si8250 Enable Polarity (JP1)

The JP1 jumper selects the polarity of the reset line for the Si8250. When installed, the reset signal is active high. Otherwise, it is active low.

10.10. C8051F300 Programmer (JP3)

The JP3 jumper allows the option to connect an onboard capacitor to the reset signal of the C8051F300. Remove this jumper when programming the C8051F300.

10.11. Voltage and Current Sense Test Points

The Si8250 Target Board has several test points for VREF, VINSENSE, IIN, VSENSE, all PHn, IPK, and more. These test points correspond to the respective pins on the Si8250-IQ integrated circuit as well as other useful inspection points. See Section “12. Schematic, Layout, and Bill of Materials” on page 24.

11. USB Debug Adapter

The USB Debug Adapter provides the interface between the PC's USB port and the Si825x's in-system debug/programming circuitry. The attached 10-pin DEBUG ribbon cable connects the adapter to the target board and the target device's debug interface signals. (The USB Debug Adapter supports both Silicon Laboratories JTAG and C2 debug interfaces.) Power is provided to the adapter from the USB connection to the PC. The USB Debug Adapter is capable of providing power to a circuit board via pin 10 of the DEBUG connector. The Si8250 Target Board is not designed to be powered from this source. Table 9 shows the pin definitions for the DEBUG ribbon cable connector.

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

Table 9. USB Debug Adapter DEBUG Connector Pin Descriptions

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

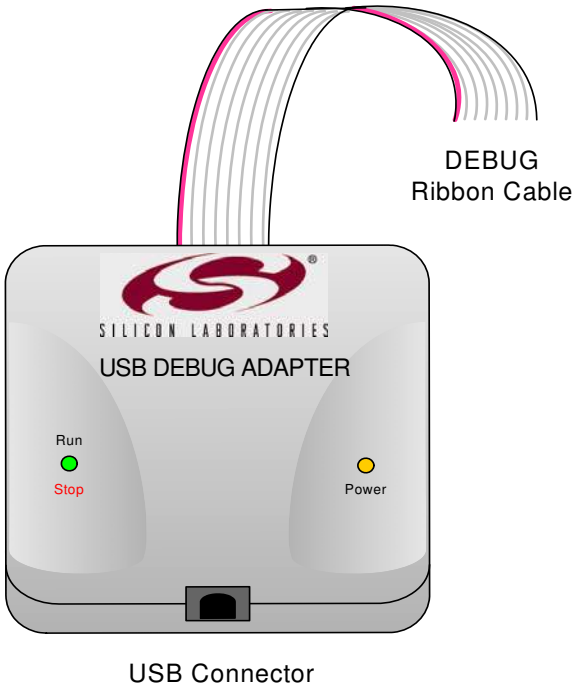

Figure 16. USB Debug Adapter

12. Schematic, Layout, and Bill of Materials

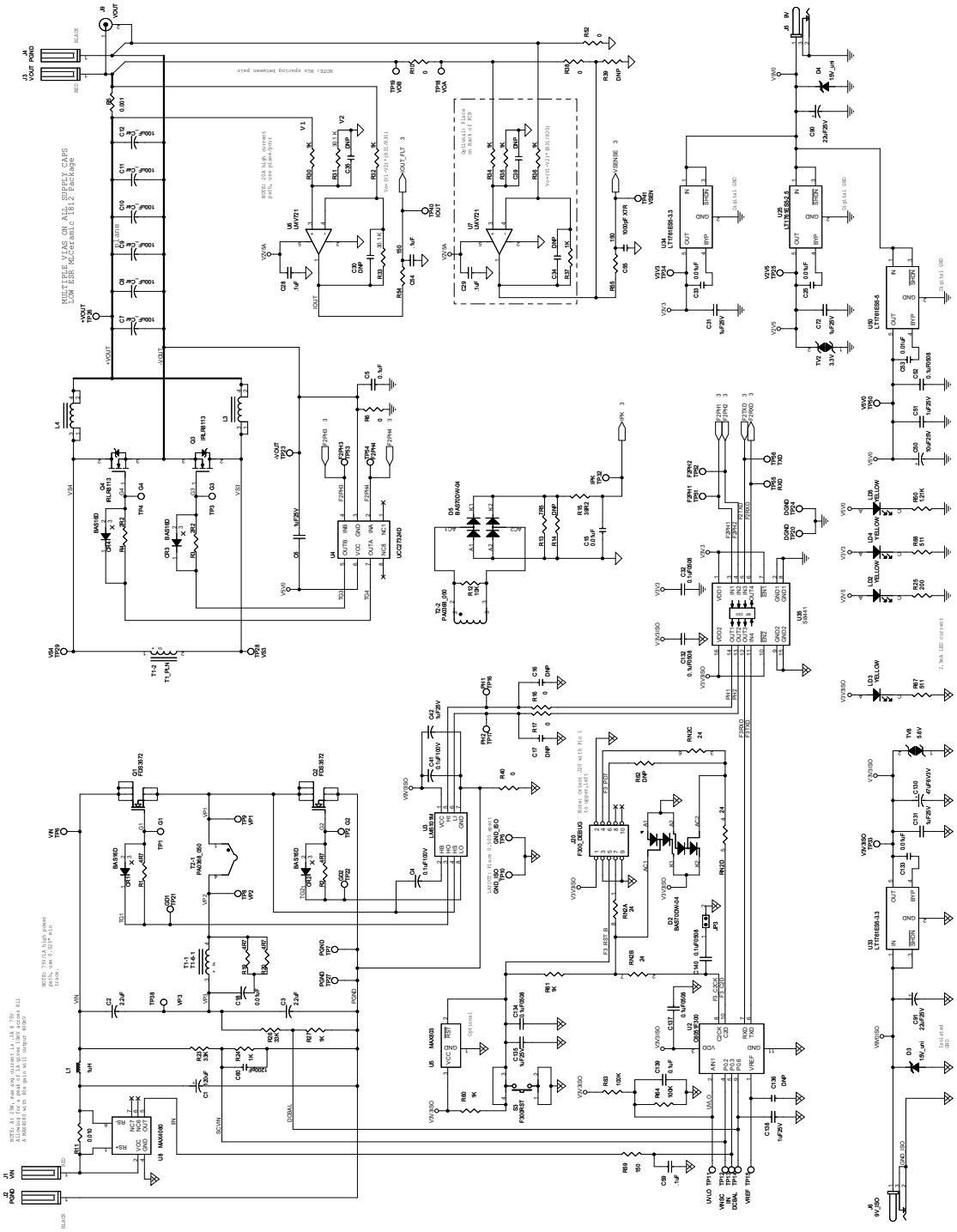

Figure 17. Si8250 Target Board Schematic (Page 1)

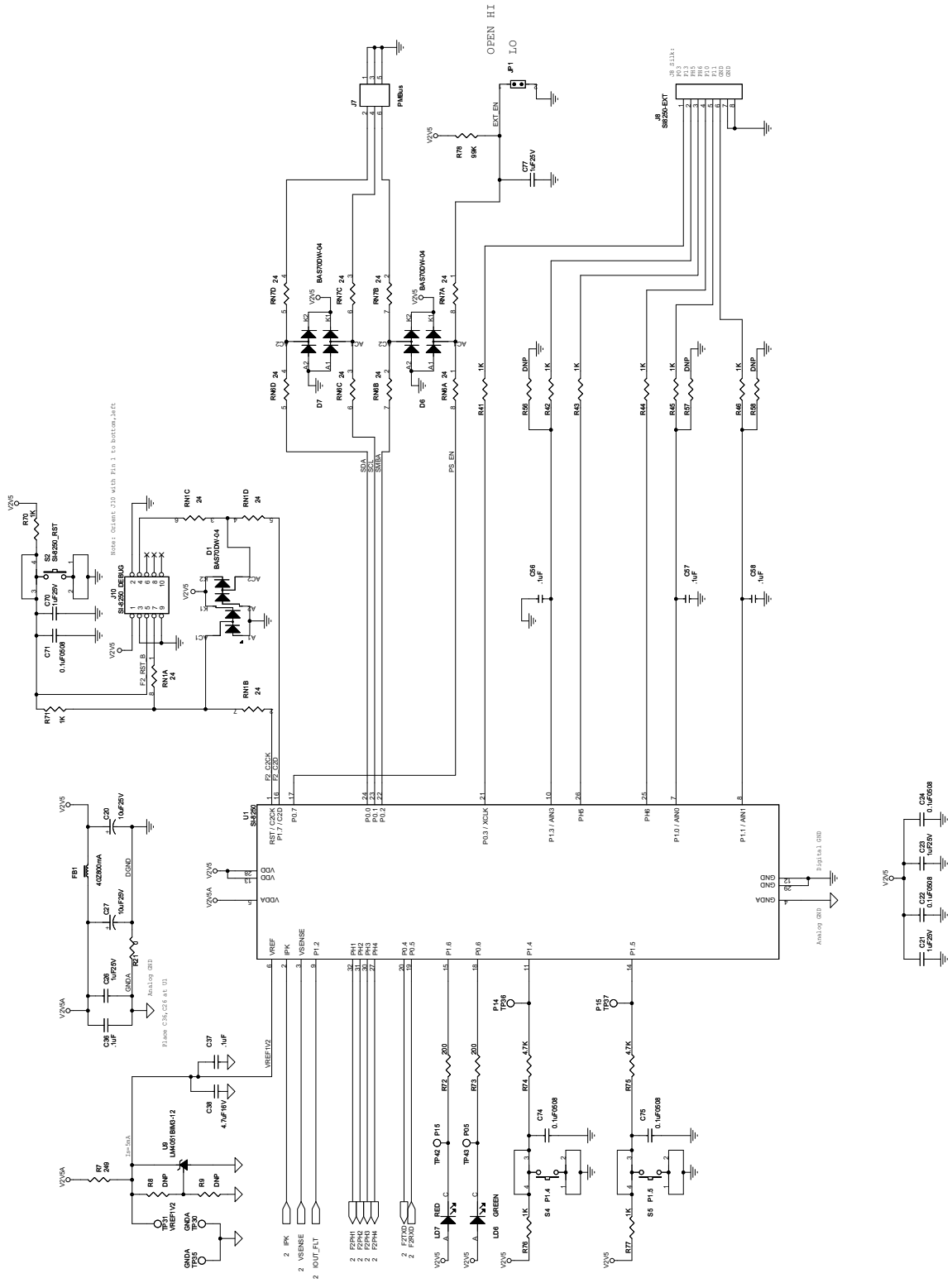

Figure 18. Si8250 Target Board Schematic (Page 2)

13. Bill of Materials

Table 10. Si825x-DK Bill of Materials

Item	Qty	Reference	Footprint	Part	MFG	MFG-PN	Dist
1	1	C55	C0603	1000 pF X7R	Panasonic	ECJ-1VB1H102K	Digi-Key
2	1	C60	C0603	1200 pF	Panasonic	ECJ-1VB1H122K	Digi-Key
3	12	C5,C22,C24,C32,C52,C71, C74,C75,C132,C134,C137, C139	C0508	0.1 uF0508	Panasonic	ECY-29RE104KV	Digi-Key
4	5	C15,C25,C33,C53,C133	C0603	0.01 uF	Panasonic	ECJ-1VB1C103K	Digi-Key
5	9	C28,C29,C36,C37,C54, C56,C57,C58,C59	C0603	.1 uF	Panasonic	ECJ-1VB1E104K	Digi-Key
6	2	C41,C4	C0805	0.1 uF100 V	TDK	C2012X7R2A104K	
7	14	C6,C18,C21,C23,C26, C31,C42,C51,C70,C72, C73,C77,C131,C135,C138	C1206	1 uF25 V	Panasonic	ECJ-3YB1E105K	Digi-Key
8	6	C7,C8,C9,C10,C11,C12	C1812	100 uF_Cer	TDK	C4532X5R0J107M	Digi-Key
9	2	C3,C2	C1812	2.2 uF	Murata	GRM43ER72A225KA01L	Digi-Key
10	3	C20,C27,C50	C7343D	10 uF25 V	Kemet	T491D106M025A5	Digi-key
11	2	C90,C91	C7343D	22 uF25 V	Panasonic	ECS-T1ED226R	Digi-Key
12	1	C38	C1206	4.7 uF16 V	TDK	C3216X7R1C475K	Digi-Key
13	1	C130	C7343D	47uF6V3V	Panasonic	EEF-HL0J470R	Digi-Key
14	1	C1	M16x15	120 uF	United Chemicon	SXE100VB121M16X15LL	
15	4(dnp)	C16,C17,C30,C35	C0603	DNP			
16	3(dnp)	C34,C39,C136	C0603	DNP			
17	4	CR1,CR2,CR3,CR4	SOT23-3	BAS16D	Diodes Inc	BAS16-7-F	Digi-Key
18	5	D1,D2,D5,D6,D7	SOT363-6	BAS70DW-04	Diodes Inc	BAS70DW-04	Digi-Key
19	2	D4,D3	SMA	15 V_uni	Diodes Inc	SMAJ15A-13	Digi-Key
20	1	FB1	FB0805	40Z800 mA	Steward	LI0805H400R-00	Digi-Key
21	4	HOL1,HOL2,HOL3,HOL4	MH_170_NP	MH			
22	2	J1.J3	BJ	VIN	Johnson Components	111-0702-001	
23	2	J2,J4	BJ	PGND	Johnson Components	111-0703-001	
24	2	J5,J6	PWR_J	9 V	Switchcraft	RAPC722	Digi-Key
25	1	JP1	BERG1X2	JUMPER1x2	Molex	22-28-4023	Digi-Key
26	1	J8	BERG1X8	SI8250-EXT	Molex	22-28-4083	Digi-Key
27	1	J7	HEADER_3X2RA	Pmbus	Sullens	PTC03DBAN	Digi-Key
28	2	J10,J20	HEADER_5X2	SI-8250_DEBUG	Molex	10-89-1101	Digi-Key
29	4	LD2,LD3,LD4,LD5	D1206	YELLOW	CML	CMD15-21VYD/TR8	Digi-Key
30	1	LD6	D1206	GREEN	CML	CMD15-21VGD/TR8	Digi-Key
31	1	LD7	D1206	RED	CML	CMD15-21VRD/TR8	Digi-Key
32	1	L1	DO1605	1 uH	CoilCraft	DO1605T-102MX	Coilcraft
33	2	L3,L4	L2525-2		Ferroxcube	E18/4/R-3F35-A160-P	

Table 10. Si825x-DK Bill of Materials (Continued)

Item	Qty	Reference	Footprint	Part	MFG	MFG-PN	Dist
34	2	Q1,Q2	DPAK	SUD15N15-95	Vishay	SUD15N15-95	
35	2	Q4,Q3	DPAK	SUD40N03-18P	Vishay	SUD40N03-18P	Mouser
36	3	R54,R55,R59	R60603	150	Panasonic	ERJ-3EKF150V	Digi-Key
37	21	R15,R24,R27,R30,R32, R34,R35,R36,R37,R41, R42,R43,R44,R45,R46, R60,R61,R70,R71,R76, R77	R0603	1 K	Panasonic	ERJ-3EKF1001V	Digi-Key
38	1	R12	R0603	10 K	Panasonic	ERJ-3EKF1002V	Digi-Key
39	5	R23,R26,R63,R64,R78	R0603	100 K	Panasonic	ERJ-3EKF1003V	Digi-Key
40	2	R31,R33	R0603	30.1 K	Panasonic	ERJ-3EKF3012V	Digi-Key
41	1	R50	R0603	1.21K	Panasonic	ERJ-3EKF1211V	Digi-Key
42	3	R25,R72,R73	R0603	200	Panasonic	ERJ-3EKF2000V	Digi-Key
43	1	R7	R0603	249	Panasonic	ERJ-3EKF2490V	Digi-Key
44	2	R74,R75	R0603	4.7 K	Panasonic	ERJ-3EKF4751V	Digi-Key
45	2	R68,R67	R0603	511	Panasonic	ERJ-3EKF5110V	Digi-Key
46	7	R6,R16,R17,R21,R38, R40,R52	R0603	0	Panasonic	ERJ-3GEY0R00V	Digi-Key
47	8	R1,R2,R3,R4,R13,R14, R19,R20	R0805	4R7	Panasonic	ERJ-3RQJ4R7V	Digi-Key
48	1	R11	R2512	0.01	Panasonic	ERJ-M1WSF10MU	Digi-Key
49	1	R5	R2512	0.001	Panasonic	ERJ-M1WTJ1M0U	Digi-Key
50	4	RN1,RN2,RN6,RN7	RPEXB28V	24	Panasonic	EXB-28V240JX	Digi-Key
51	7(dnp)	R8,R9,R39,R56,R57, R58,R62	R0603	DNP			
52	4	S2,S3,S4,S5	SWPAD04	SI-8250_RST	Panasonic	EVQ-PAD04M	Digi-Key
53	4	TP25,TP33,TP34,TP50	TP040	V2V5	Keystone	5000	Digi-Key
54	6	TP5,TP10,TP20,TP24, TP30,TP35	TP040	GND_ISO	Keystone	5001	Digi-Key
55	25	TP1,TP2,TP3,TP4,TP6, TP7,TP8,TP9,TP11,TP12, TP13,TP14,TP15,TP16, TP17,TP21,TP22,TP23, TP27,TP31,TP32,TP51, TP52,TP53,TP54	TP040	G1	Keystone	5002	Digi-Key
56	1	TV2	D0603	3.3 V	AVX	VC060303A100DP	Mouser
57	1	TV8	D0603	5.6 V	AVX	VC060305A150RP	Mouser
58	1	T1	T1_PCB	T1-6-1	Ferroxcube	E22/6/16/R-3F3	
59	1	T2	PA0282	PA0368_050	Pulse	PA0368.050	
60	1	U5	SC70-3	MAX803	Maxim	MAX803TEXR	
61	1	U8	UMAX-8	MAX4080	Maxim	MAX4080SAUA	
62	1	U9	SOT23-3	LM4051BIM3-12	National	LM4051BIM3-12	Digi-Key
63	1	U3	SO-8	LM5101AM	National	LM5101AM	Digi-Key

Table 10. Si825x-DK Bill of Materials (Continued)

Item	Qty	Reference	Footprint	Part	MFG	MFG-PN	Dist
64	2	U6,U7	SOT23-5	LMV721	National	LMV721M5	Digi-Key
65	1	U25	SOT23-5	LT1761ES5-2.5	LTC	LT1761ES5-2.5	Arrow
66	2	U33,U34	SOT23-5	LT1761ES5-3.3	LTC	LT1761ES5-3.3	Arrow
67	1	U50	SOT23-5	LT1761ES5-5	LTC	LT1761ES5-5	Arrow
68	1	U35	SO-16_WIDE	Si8441	SiLabs	Si8441BB-C-IS	SiLabs
69	1	U1	LQFP32	Si8250	SiLabs	Si8250-IQ	SiLabs
70	1	U2	MLP11	C8051F300	SiLabs	C8051F300	SiLabs
71	1	U4	SO-8	UCC27324D	TI	UCC27324D	Digi-Key

DOCUMENT CHANGE LIST

Revision 0.2 to Revision 0.3

- Updated “Contact Information” on page 30.
  - Updated disclaimer.

Revision 0.3 to Revision 0.31

- Updated Table 10 to reflect new Si84xx isolation nomenclature.

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