
CP250X DEVELOPMENT KIT USER'S GUIDE

1. Kit Contents

The CP250x Development Kit contains the following items:

- CP250x Motherboard
- CP2501 Development Daughter Board
- CP250x Development Kit Quick-Start Guide
- Product information CD-ROM including the following:
 - Silicon Laboratories Integrated Development Environment (IDE)
 - SDCC Development Tools
 - Raisonance Development Tools (evaluation version)
 - Keil 8051 Development tools (evaluation version)
 - CP250x Configuration Wizard and Bootloader
 - Source code examples
 - Documentation
- AC-to-DC universal power adapter
- Two USB cables

2. Hardware Setup

See Figure 1 for a diagram of the hardware configuration.

1. Attach the daughter board to the motherboard at connectors P1 and P2.
2. Place shorting blocks on J7 and J3 [+3VD, VDD_PWR], as shown in Figure 1.
3. Connect the motherboard's P5 USB connector to a PC using the USB Cable.
4. Connect the ac-to-dc power adapter to connector P3 on the motherboard.

Notes:

1. Use the Reset icon in the IDE to reset the target when connected during a debug session.
2. Remove power from the motherboard and remove the USB cable before removing a daughter board from the motherboard. Connecting or disconnecting a daughter board when the power adapter or USB cable are connected can damage the motherboard or the daughter board.

CP2501-DK

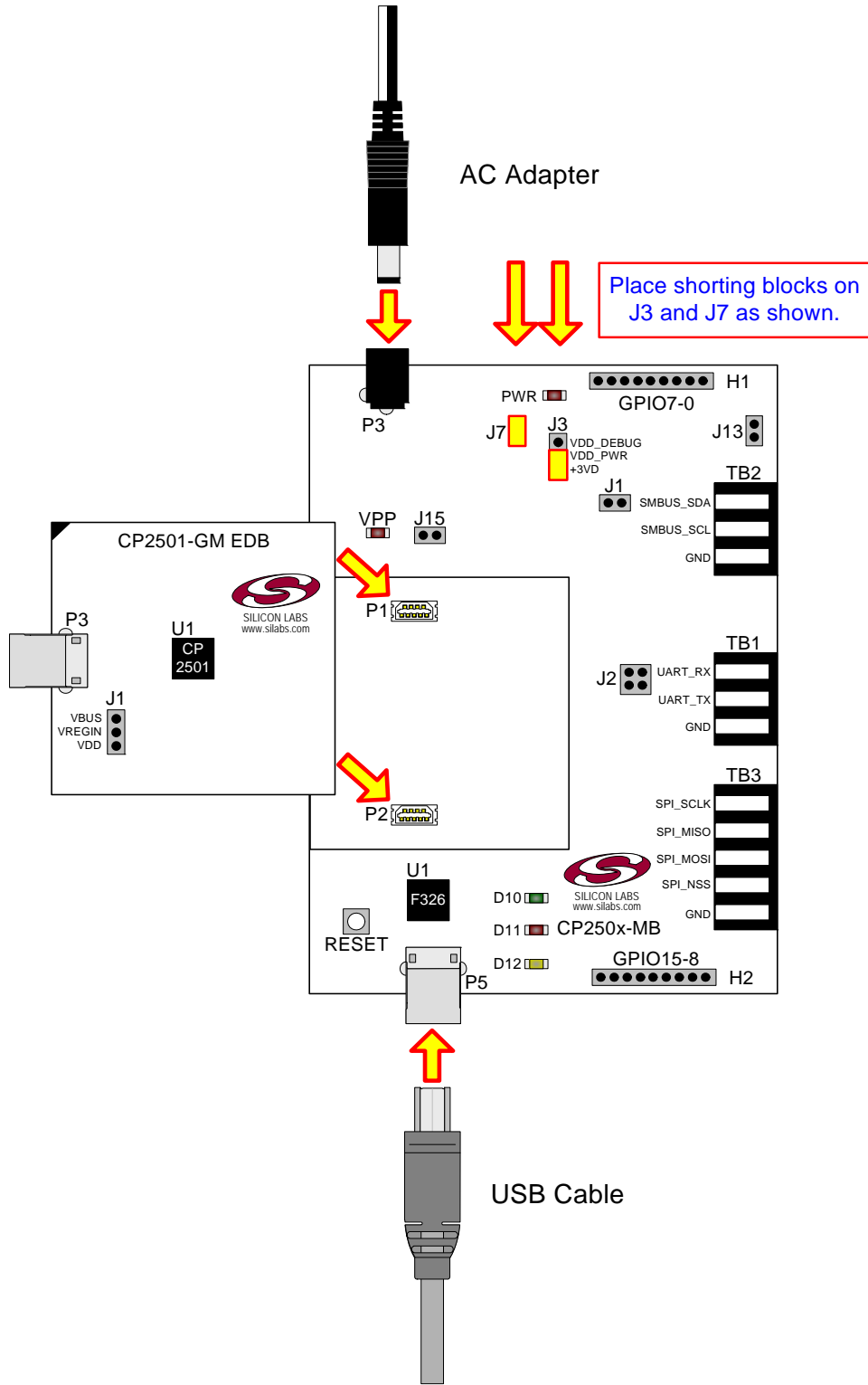


Figure 1. Hardware Setup

3. 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.

3.1. System Requirements

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

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

3.2. 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 CP250x 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**. See Section 4. 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. 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 the following:

- CP250x Configuration Wizard
- CP250x USB Bootloader

4.1. 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/products/mcu/Pages/ApplicationNotes.aspx>). 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"
SDCC	AN198, "Integrating SDCC 8051 Tools into the Silicon Labs IDE"

4.2. 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.

4.2.1. Running the CP250x_NoScreen example program

The **CP250x_NoScreen** example program drags a simulated finger starting from the top left corner of the screen towards the middle of the screen and releases.

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\CP250x\NoScreen` directory (default) and select the `CP250x_NoScreen.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. CP2501 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.
9. Connect the USB connector on the daughter board to a Windows Vista or Windows 7 PC that supports HID touchscreens.
10. The device will enumerate and then drag the simulated finger across the screen.

4.2.2. Creating a New Project

Use the following steps to create a new project. Once steps 1-5 in this section are complete, continue at Step 4 in Section 4.2.1.

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**.

4.3. CP250x Configuration Wizard

The CP250x Configuration Wizard generates the initial firmware project for CP250x devices based on the user's system requirements. Code is generated through the use of dialog boxes as shown in Figure 2.

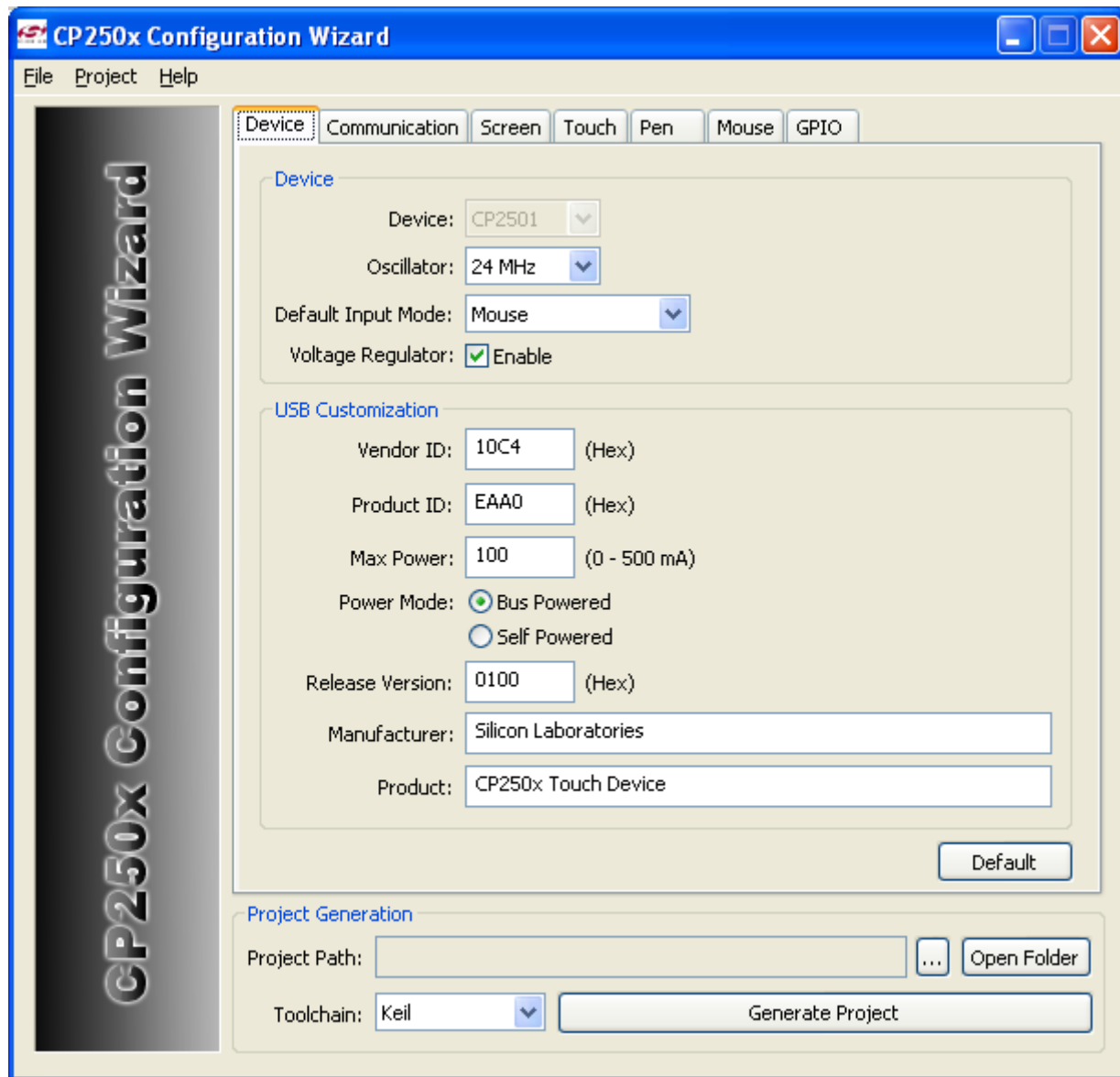


Figure 2. CP250x Configuration Wizard

The CP250x Configuration Wizard 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 firmware that will enumerate over USB as a HID touch screen device. The program is configurable to output projects for Raisonance, SDCC, and Keil.

For more information, refer to the “AN464: CP250x Programmer’s Guide and API specification.” Documentation and software is available on the kit CD and from the downloads webpage: www.silabs.com/mcudownloads.

5. Development Boards

The CP2501 Development Kit includes a motherboard that interfaces to a daughter board. The CP2501 Development Daughter Board contains a CP2501 device to be used for preliminary software development. Numerous input/output (I/O) connections are provided on the motherboard to facilitate prototyping. Figure 3 shows the CP250x Motherboard and indicates locations for various I/O connectors and shows the factory default shorting block positions. Figure 4 shows the available CP250x daughter board.

CP250x Motherboard

P1, P2	Daughter board connection
P3	Power connector that accepts input from 7.5 V dc to 15 V dc unregulated power adapter
P5	USB Debug interface connector
H1	GPIO 7:0 header
H2	GPIO 15:8 header
J1	SMBus Pull-up Resistor Power Header
J2	UART Connection to Debug Adapter
J3	Power Supply Selection Header
J7	Power supply enable header that connects power source selected on J3 to the board's main power supply net
J13	Additional connections to ground
TB1	UART Interface Header
TB2	SMBus Interface Header
TB3	SPI Interface Header

CP2501 Development Daughter Board

J1	CP2501 VREGIN Input Selection Header
P3	USB connector for touch-screen interface

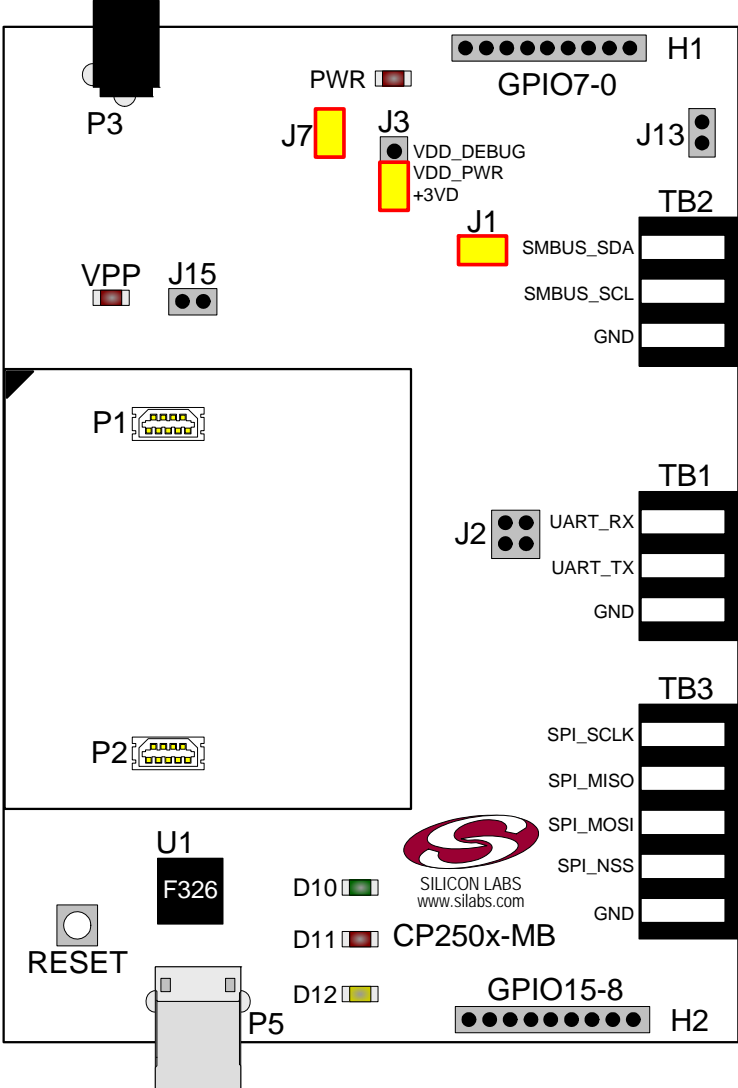


Figure 3. CP250x Motherboard with Default Shorting Block Positions

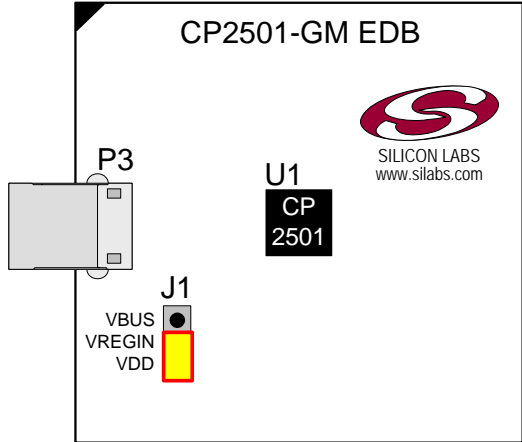


Figure 4. CP2501 Development Daughter Board

5.1. Switch and LEDs

The RESET switch is connected to the $\overline{\text{RST}}$ pin of the CP250x. Pressing RESET puts the device into its hardware-reset state.

Four LEDs are also provided on the motherboard. The red LED labeled “PWR” (D4), indicates a power connection to the motherboard. The green LED labeled “RUN” (D10) turns on when the debug circuitry is in a running state; the red LED labeled “STOP” (D11) turns on when the debug circuitry is in a halted state, and the orange LED labeled “DEBUG PWR” (D12) indicates whether the debug adapter circuit is being powered through P5’s USB connector. Table 2 lists the port pins and headers corresponding to the switch and LEDs.

Table 2. Motherboard LED and Switch Descriptions

Description	Component Name	I/O	Header
RESET	SW3	Daughter Board’s RST/C2CK	None
Red LED labeled "PWR"	D4	Daughter Board’s VDD	J3, J7
Green LED labeled "RUN"	D10	Debug Adapter Signal	None
Red LED labeled "STOP"	D11	Debug Adapter Signal	None
Orange LED labeled "DEBUG PWR"	D12	Debug Adapter Signal	None

5.2. Power Supply Headers (P3, J3, and J7)

The main power supply of the motherboard, which is used to power the daughter board, can be provided by either the USB debug device’s on-chip voltage regulator or P3 and its associated circuitry. To select a power supply, place a shorting block on J3 across the appropriate pin pair, as shown in Figure 5. To connect the main power supply to an attached daughter board, place a shorting block across J7.

Notes:

1. Only two of the pins on the J3 header should be connected at any one time
2. To use the USB debug device’s voltage regulator as the board’s power supply, a USB cable must be connected to P5, and the DEBUG PWR LED (D12) must be on.

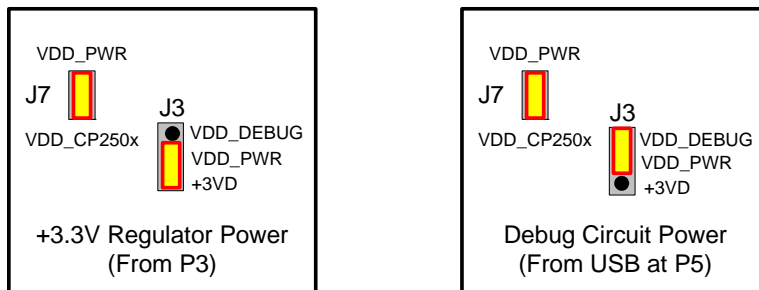


Figure 5. J3 and J7 Shorting Block Configuration for Power Options

5.3. USB Debug Device (DEBUG/P5)

A universal serial bus (USB) connector (P5) provides the onboard debug and programming interface. The debug/programming MCU and associated circuitry are powered through the USB connector, which can also supply the rest of the motherboard by routing the USB debug device's power through J3. The USB debug device also provides a data communications interface that can be used when the debug adapter is not debugging or programming a CP250x device.

5.4. UART Communications Interfaces (TB1, J2)

The CP250x Motherboard provides UART to USB communications interfaces through both the TB1 interface and the communications interface of the USB Debug Adapter. Any external touch screen module using the UART interface can connect to the CP250x through either header.

The USB debug device's communications interface connects to a PC through P5. Access to the USB debug device's communications interface is provided by the Windows program called "ToolStick Terminal", which is available for download for free from the Silicon Laboratories website. See the ToolStick Terminal help file for information on how to use ToolStick Terminal.

The CP250x Motherboard routes the CP250x's UART_TX and UART_RX pins to J2, where those signals can be optionally connected to TB1 header.

5.5. SMBus Communications Interface (TB2, J1)

The CP250x Motherboard connects the SMBus interface pins SMBUS_SDA and SMBUS_SCL to the TB2 header. These two pins can be pulled up to the VIO net through the two pull-up resistors, R1 and R2, by adding the shorting block to the J1 header. Only one set of pull-up resistors is required on any SMBus network. The shorting block on the J1 header should be removed if the pull-up resistors are already present on the network.

The R1 and R2 pull-up resistors are 1K resistors and are located on the backside of the mother board near the TB2 header.

5.6. SPI Communications Interfaces (TB3)

The CP250x Motherboard connects the SPI interface pins, SPI_SCLK, SPI_MISO, SPI_MOSI, and SPI_NSS to the TB3 header.

5.7. GPIO Interfaces (H1, H2)

The CP2501 devices support 16 GPIO pins (GPIO0 through GPIO15). These GPIO pins are connected to the two headers H1 and H2.

5.8. Daughter Board (P3, J1)

The P3 header on the daughter board is the USB connection for the CP2501. The J3 header selects how the CP2501 on the daughter board is powered. If a shorting block is connected to J1[VBUS-VREGIN], the CP2501 is powered from the P3 USB connector. If a shorting block is connected to J1[VDD-VREGIN], the CP2501 is powered from the motherboard.

6. Schematics

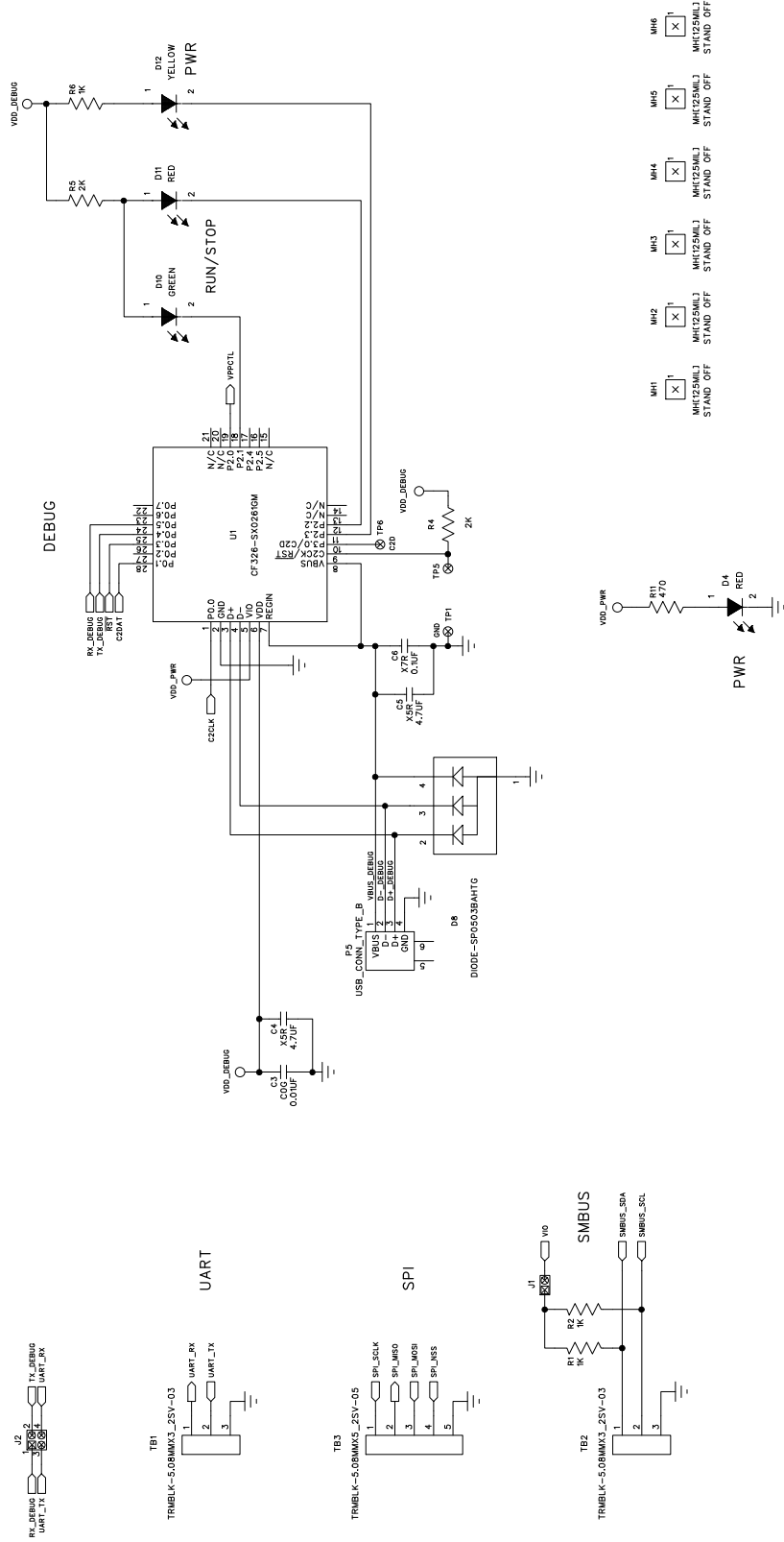


Figure 6. CP250x Motherboard Schematic (1 of 2)

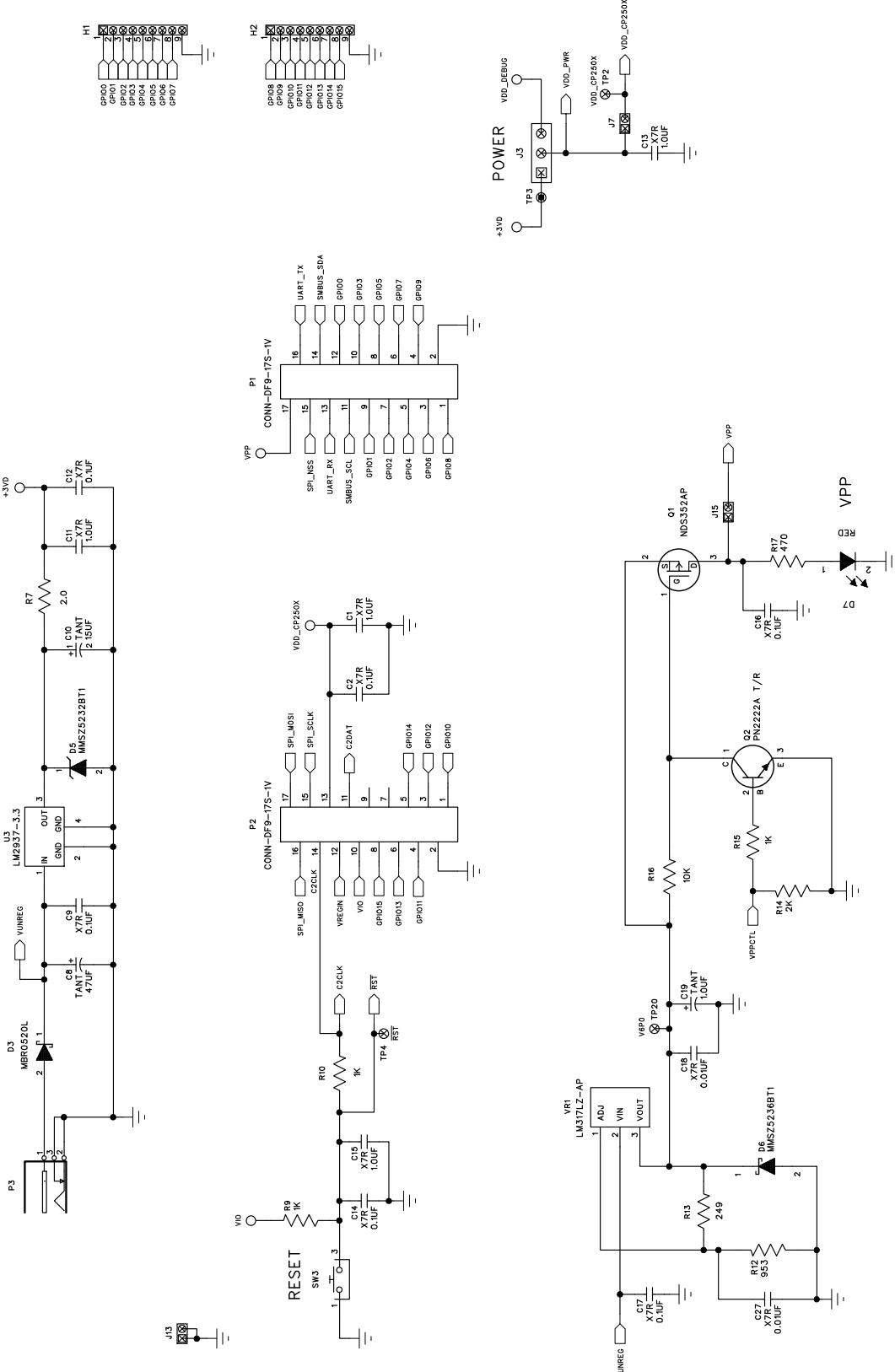


Figure 7. CP250x Motherboard Schematic (2 of 2)

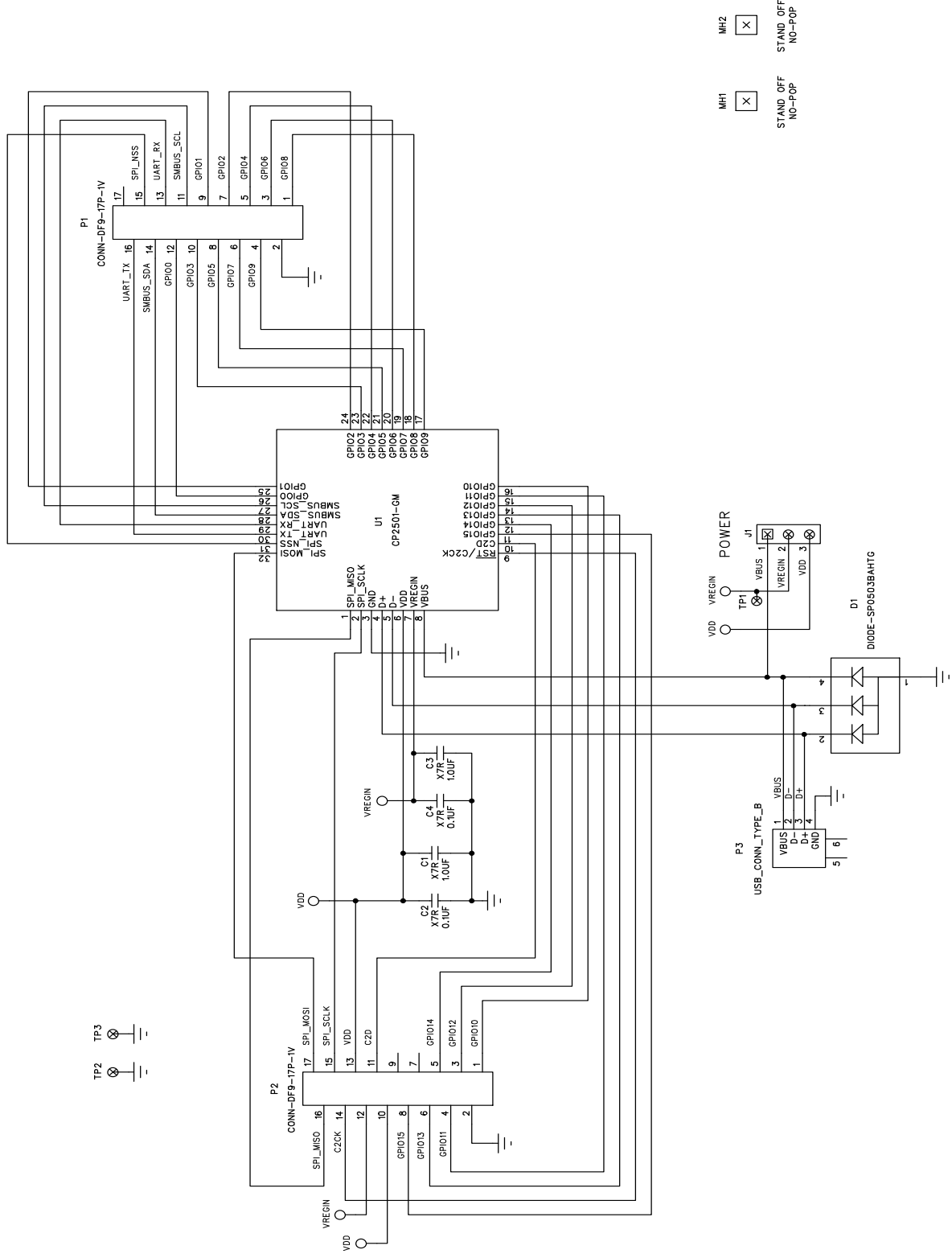


Figure 8. CP2501 Development Daughter Board Schematic

NOTES:

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 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.