C8051F500DK

C8051F500 Development Kit User's Guide

1. Relevant Devices

The C8051F500 Development Kit is intended as a development platform for the microcontrollers in the C8051F50x MCU family. The members of this MCU family are: C8051F500, C8051F501, C8051F502, C8051F503, C8051F504, C8051F505, C8051F506, and C8051F507.

- The target board included in this kit is provided with a pre-soldered **C8051F500** MCU (QFP48 package) and a **C8051F502** (QFN32 package).
- Code developed on the C8051F500 can be easily ported to the other members of this MCU family.
- Refer to the C8051F50x data sheet for the differences between the members of this MCU family.

2. Kit Contents

The C8051F500 Development Kit contains the following items:

- C8051F500 Target Board
- C8051Fxxx Development Kit Quick-Start Guide
- Silicon Laboratories IDE and Product Information CD-ROM. CD content includes:
- Silicon Laboratories Integrated Development Environment (IDE)
  - Keil 8051 Development Tools (macro assembler, linker, evaluation 'C' compiler)
  - Source code examples and register definition files
  - Documentation
  - C8051F500 Development Kit User's Guide (this document)
- AC to DC Power Adapter
- USB Debug Adapter (USB to Debug Interface)
- Two USB Cables

3. Getting Started

The necessary software to download, debug, and communicate with the target microcontroller is included in the CD-ROM. The following software is necessary to build a project, download code to, and communicate with the target microcontroller:

- Silicon Laboratories Integrated Development Environment (IDE)
- Keil 8051 Development Tools (macro assembler, linker, evaluation 'C' compiler)

Other useful software that is provided in the CD-ROM includes:

- Configuration Wizard 2
- Keil uVision Drivers
- CP210x USB to UART Virtual COM Port (VCP) Drivers

3.1. Software Installation

The included CD-ROM contains the Silicon Laboratories Integrated Development Environment (IDE), 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 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. After installing the software, see the following sections for information regarding the software and running one of the demo applications.

3.2. CP210x USB to UART VCP Driver Installation

The C8051F500 Target Board includes a Silicon Laboratories CP2102 USB-to-UART Bridge Controller. Device drivers for the CP2102 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 (P5) on the C8051F500 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 Laboratories CP210x USB to UART Bridge (Driver Removal" option in the "Add or Remove Programs" window.

4. Software Overview

4.1. Silicon Laboratories IDE

The Silicon Laboratories IDE integrates a source-code editor, a source-level debugger, and an in-system Flash programmer. See Section 6. "Using the Keil Software 8051 Tools with the Silicon Laboratories IDE," on page 5 for detailed information on how to use the IDE. The Keil Evaluation Toolset includes a compiler, linker, and assembler and easily integrates into the IDE. The use of third-party compilers and assemblers is also supported.

4.1.1. IDE System Requirements

The Silicon Laboratories IDE requirements:

- Pentium-class host PC running Microsoft Windows 2000 or newer.
- One available USB port.
- 64 MB RAM and 40 MB free HD space recommended.

4.1.2. Third Party Toolsets

The Silicon Laboratories IDE has native support for many 8051 compilers. The full list of natively supported tools is:

- Keil
- IAR
- Raisonance
- Tasking
- Hi-Tech
- SDCC

Please note that the demo applications for the C8051F500 target board are written to work with the Keil and SDCC toolsets.

4.2. Keil Evaluation Toolset

4.2.1. Keil Assembler and Linker

The assembler and linker that are part of the Keil Demonstration Toolset are the same versions that are found in the full Keil Toolset. The complete assembler and linker reference manual can be found on-line under the **Help** menu in the IDE or in the "*SiLabs\MCU\hlp*" directory (A51.pdf).

4.2.2. Keil Evaluation C51 C Compiler

The evaluation version of the C51 compiler is the same as the full version with these limitations: (1) Maximum 4 kB code generation, and (2) Floating point library not included. When installed from the CD-ROM, the C51 compiler is initially limited to a code size of 2 kB, and programs start at code address 0x0800. Please refer to the Application Note "AN104: Integrating Keil Tools into the Silicon Labs IDE" for instructions to change the limitation to 4 kB, and have the programs start at code address 0x0000.

4.3. Configuration Wizard 2

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

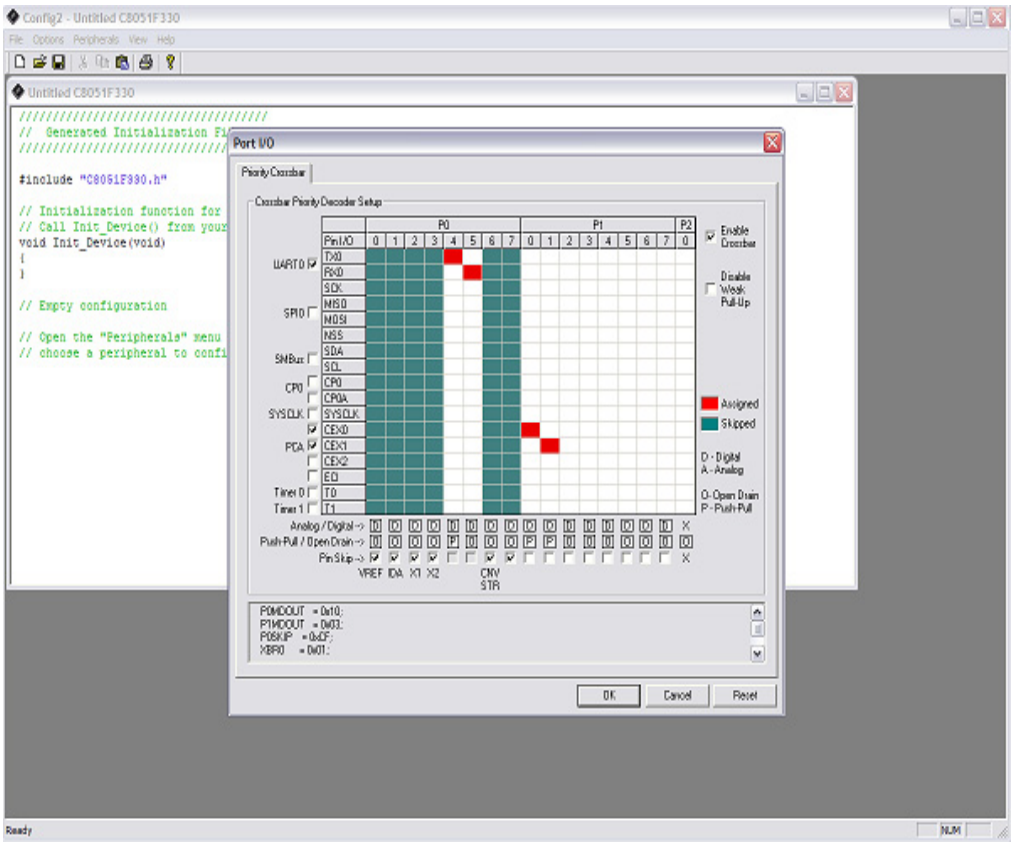

Figure 1. Configuration Wizard 2 Utility

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 Laboratories 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 Config Wizard 2.

4.4. Keil uVision2 and uVision3 Silicon Laboratories Drivers

As an alternative to the Silicon Laboratories IDE, the uVision debug driver allows the Keil uVision IDE to communicate with Silicon Laboratories on-chip debug logic. In-system Flash memory programming integrated into the driver allows for rapidly updating target code. The uVision 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 uVision driver documentation. The documentation and software are available from the Downloads webpage (www.silabs.com/mcudownloads).

5. Hardware Setup using a USB Debug Adapter

The target board is connected to a PC running the Silicon Laboratories IDE via the USB Debug Adapter as shown in Figure 2.

1. Connect the USB Debug Adapter to one of the DEBUG connector on the target board (DEBUG_A or DEBUG_B) with the 10-pin ribbon cable. The recommended connection is to DEBUG_A as this microcontroller is the primary MCU on the board and more peripherals are easily available.
2. Connect one end of the USB cable to the USB connector on the USB Debug Adapter.
3. Connect the other end of the USB cable to a USB Port on the PC.
4. Connect the AC/DC power adapter to power jack P4 on the target board.

Notes:

- Use the Reset button in the IDE to reset the target when connected using a USB Debug Adapter.
- Remove power from the target board and the USB Debug Adapter before connecting or disconnecting the ribbon cable from the target board. Connecting or disconnecting the cable when the devices have power can damage the device and/or the USB Debug Adapter.

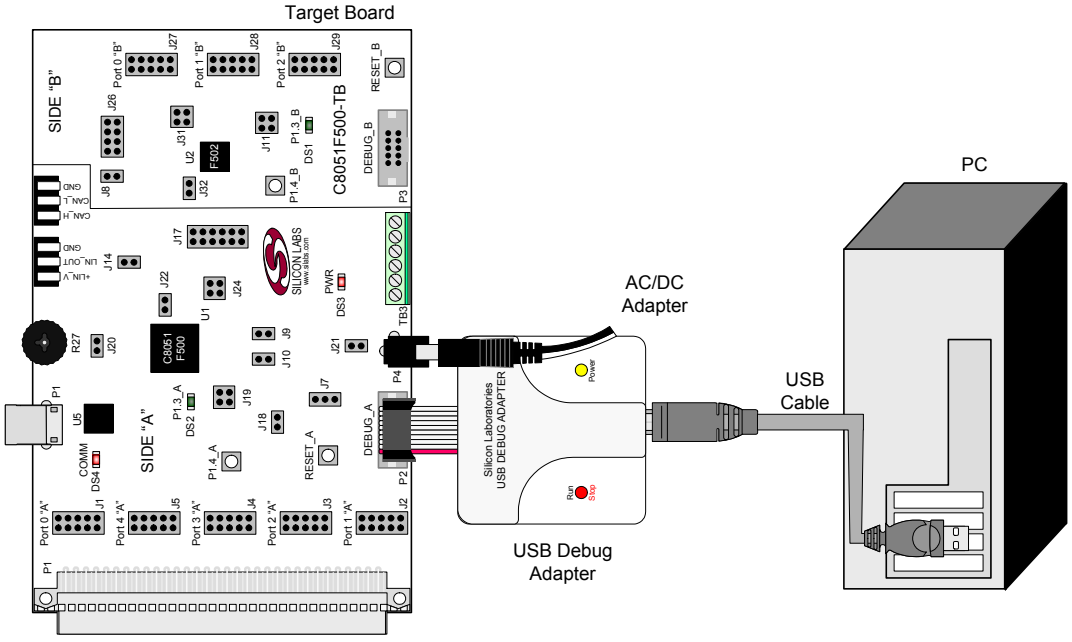

Figure 2. Hardware Setup using a USB Debug Adapter

6. 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" in the "*SiLabs\MCU\Documentation\ApplicationNotes*" directory 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.)

6.1. 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 the files. 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.

6.2. 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 appropriate adapter in the "Serial Adapter" section. Next, the correct "Debug Interface" must be selected. C8051F50x family devices use the Silicon Labs 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. Example Source Code

Example source code and register definition files are provided in the "*SiLabs\MCU\Examples\C8051F50x_1x*" 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.

7.1. Register Definition Files

Register definition files *C8051F500.inc and C8051F500_defs.h* define 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 *C8051F500_defs.h* header file with various tool chains. These files are installed into the "*SiLabs\MCU\Examples\C8051F50x_1x\Header_Files*" directory during IDE installation by default. The register and bit names are identical to those used in the C8051F50x data sheet. These register definition files are also installed in the default search path used by the Keil Software 8051 tools. Therefore, when using the Keil 8051 tools included with the development kit (A51, C51), it is not necessary to copy a register definition file to each project's file directory.

7.2. Blinking LED Example

The example source files *F500_Blinky.asm* and *F500_Blinky.c* installed in the default directory "*SiLabs\MCU\Examples\C8051F50x_1x\Blinky*" show examples of several basic C8051F500 functions. These include disabling the watchdog timer (WDT), configuring the Port I/O crossbar, configuring a timer for an interrupt routine, initializing the system clock, and configuring a GPIO port pin. When compiled/assembled and linked, this program flashes the green LED on the C8051F500 target board about five times a second using the interrupt handler with a C8051F500 timer.

8. Target Board

The C8051F500 Development Kit includes a target board with a **C8051F500** (Side A) and **C8051F502** (Side B) 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 3 for the locations of the various I/O connectors. Figure 4 on page 9 shows the factory default shorting block positions. A summary of the signal names and headers is provided in Table 11 on page 16.

J1-J5	Side A: Port 0 through Port 4 headers
J7	Header to choose between +5V from Debug Adapter (P2) or +5V from on-board regulator (U6)
J8	Side B: CAN Transceiver (U4) power connector
J9, J10	Side A: External crystal enable connectors
J11	Side B: Connects P1.3_B LED and P1.4_B Switch to MCU port pins
J14	Side A: CAN Transceiver (U3) power connector
J17	Side A: Connects MCU to three separate transceivers (UART(U5), CAN(U3) and LIN(T1))
J18	Side A: Connects VIO to VIO_A_SRC which powers the P1.2 potentiometer, the /RST_A pin pull-up, and P1.4_A Switch pull-up.
J19	Side A: Connects P1.3_A LED and P1.4_A Switch to MCU port pins
J20	Side A: Connects R27 potentiometer to port pin 1.2
J21	Connect V_HIGH node from TB1 LIN header to +5V regulator input for board power
J22	Side A: Connects decoupling capacitors C28 and C29 for MCU VREF (P0.0)
J24	Side A: Connects +5V net to VIO and VREGIN of the MCU
J26	Side B: Connects MCU to three separate transceivers (CAN (U4) and LIN (T2))
J27-J29	Side B: Port 0 through Port 2 headers
J31	Side B: Connects +5V net to VIO and VREGIN of the MCU
J32	Side B: Connects decoupling capacitors C41 and C42 for MCU VREF (P0.0)
P1	Side A: 96-pin female connector
P2	Side A: DEBUG connector for Debug Adapter interface
P3	Side B: DEBUG connector for Debug Adapter interface
P4	Power connector (accepts input from 7 to 15 VDC unregulated power adapter)
P5	USB connector (connects to PC for serial communication)
TB1	Shared LIN Connector for Side A and B MCUs for external nodes
TB2	Shared CAN Connector for Side A and B MCUs for external nodes
TB3	Side A: Power supply terminal block

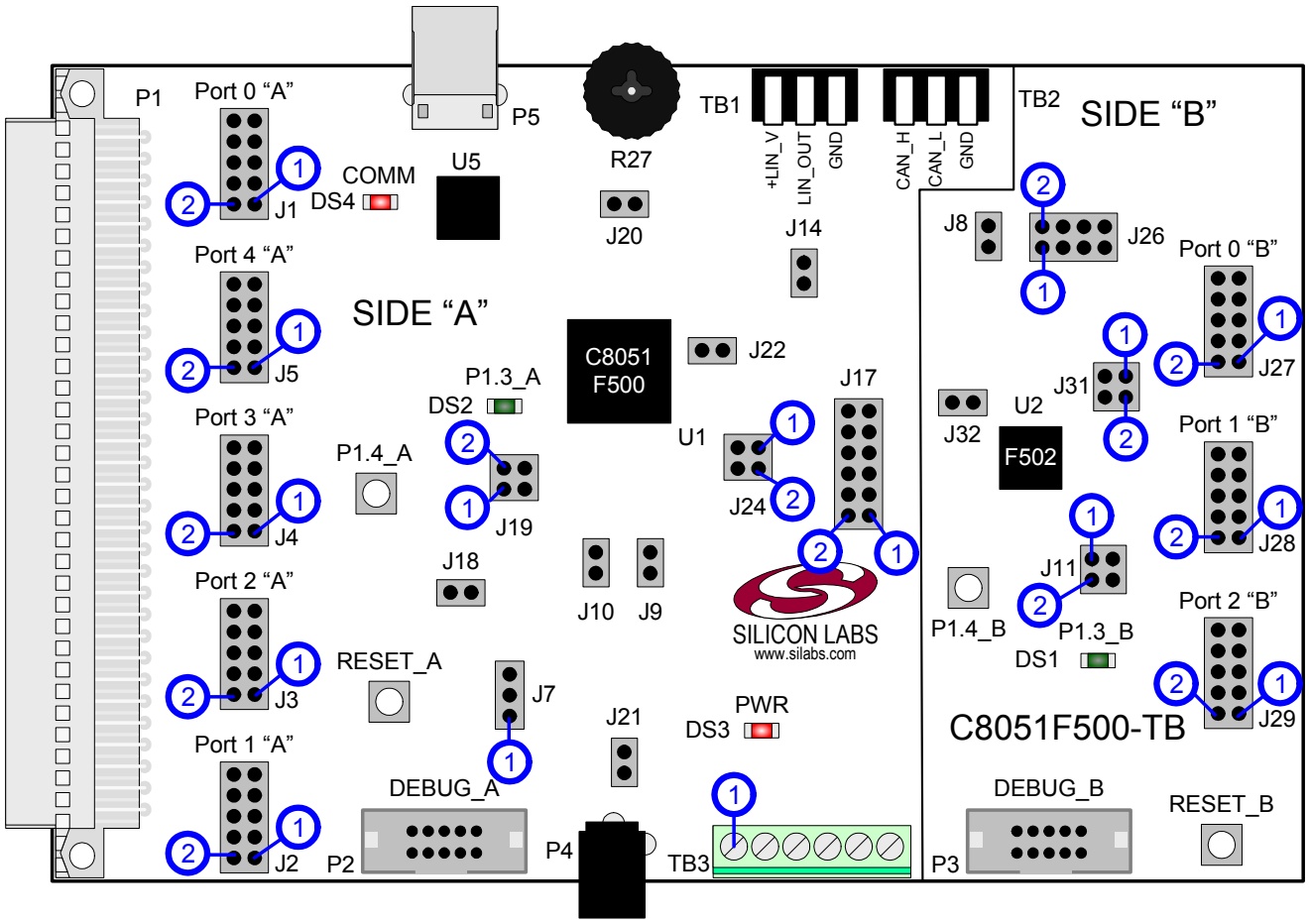

Figure 3. C8051F500 Target Board with Pin Numbers

8.1. Target Board Shorting Blocks: Factory Defaults

The C8051F500 target board comes from the factory with pre-installed shorting blocks on many headers. Figure 4 shows the positions of the factory default shorting blocks.

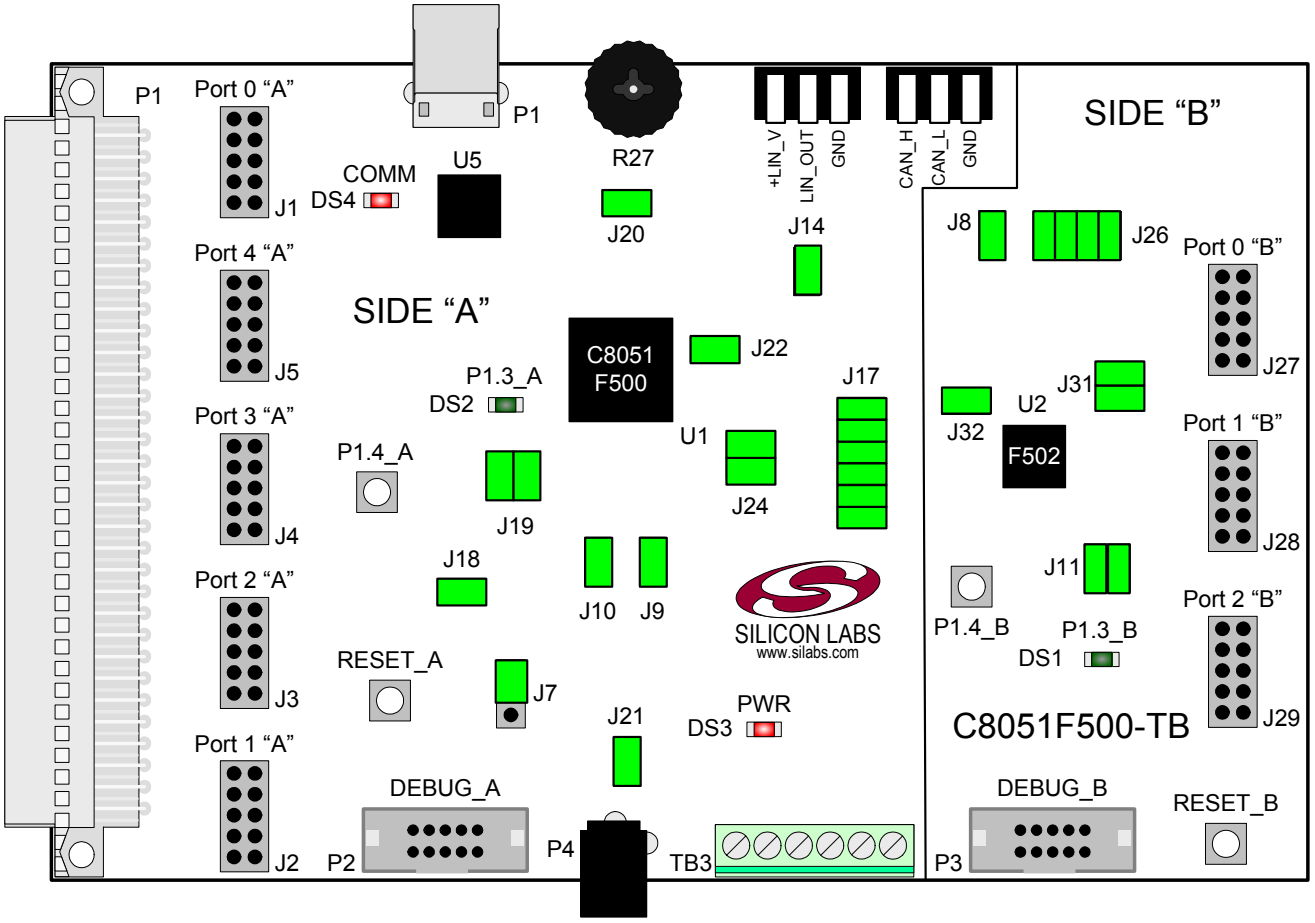

Figure 4. C8051F500 Target Board Shorting Blocks: Factory Defaults

8.2. Target Board Power Options and Current Measurement

The C8051F500 target board supports three power options:

1. 12V DC power using the AC to DC power adapter (P4)
2. 5V DC USB VBUS power from PC via the USB Debug Adapter (DEBUG_A)
3. 12V DC power from the LIN external header (TB1)

The two 12V power sources are ORed together using reverse-biased diodes (Z1 and Z2). The ORed power is regulated to a 5.0V DC voltage using a LDO regulator (U6). To power the board from the USB Debug Adapter connected to DEBUG_A instead of the 12V sources, move the shorting block on the J7 header to pins 2 and 3 to select SER_PWR. The output of the regulator powers the +5VD net on the target board, and is also connected to one end of the header J24 (SIDE A) and J31 (SIDE B). Two shorting blocks can be put on each header to connect the 5V net to the VREGIN and VIO pins on the two MCUs. With the shorting block removed, a source meter can be used across the headers to measure the current consumption of the MCU.

Note: The USB Debug Adapter does not provide the necessary peak power for the CAN transceivers to operate. One of the 12V DC sources is recommended for CAN transceiver operation.

8.3. System Clock Sources

8.3.1. Internal Oscillators

The C8051F500 and C8051F502 devices installed on the target board feature a factory calibrated programmable high-frequency internal oscillator (24 MHz base frequency, ±0.5%), which is enabled as the system clock source on reset. After reset, the internal oscillator operates at a frequency of 187.5 kHz by default but may be configured by software to operate at other frequencies. The on-chip crystal is accurate for CAN and LIN master communications and in many applications an external oscillator is not required. However, if you wish to operate the C8051F500 device (SIDE A) at a frequency not available with the internal oscillator, an external crystal may be used. Refer to the C8051F50x data sheet for more information on configuring the system clock source.

8.3.2. External Oscillator Options

The target board is designed to facilitate the installation of an external crystal. Remove shorting blocks at headers J9 and J10 and install the crystal at the pads marked Y1. Install a 10 MΩ resistor at R9 and install capacitors at C6 and C7 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 C6 for capacitor mode, and populate R3 and C6 for RC mode. Refer to the C8051F50x data sheet for more information on the use of external oscillators.

8.4. Switches and LEDs

Two push-button switches are provided on the target board for each MCU. Switch RESET_A is connected to the /RST pin of the C8051F500. Switch RESET_B is connected to the /RST pin of the C8051F502. Pressing RESET_A puts the C8051F500 device into its hardware-reset state, and similarly for RESET_B and the C8051F502 MCU. Switches P1.4_A and P1.4_B are connected to the MCU's general purpose I/O (GPIO) pins through headers. Pressing either one of these switches generates a logic low signal on the port pin. Remove the shorting block from the header to disconnect these switches from the port pins. See Table 1 for the port pins and headers corresponding to each switch.

Four LEDs are provided on the target board to serve as indicators. The red LED labeled PWR indicates presence of power to the target board. The second red LED labeled COMM indicates if the CP2102 USB-to-UART bridge (P5) is recognized by the PC. The green LED labeled with port pin name P1.3_A is connected to the C8051F500's (Side A) GPIO pin P1.3 through the header J19. Remove the shorting block from the header to disconnect the LED from the port pin. Similarly, the green LED named P1.3_B is connected to the C8051F502 (Side B) through the J11 header. See Table 1 for the port pins and headers corresponding to each LED.

Table 1. Target Board I/O Descriptions

Description	I/O	Header(s)
RESET_A	Reset (Side A)	none
RESET_B	Reset (Side B)	none
P1.4_A Switch	P1.4 (Side A)	J191–2]
P1.4_B Switch	P1.4 (Side B)	J11[1–2]
P1.3_A LED	P1.3 (Side A)	J19[3–4]
P1.3_B LED	P1.3 (Side B)	J11[3–4]
Red LED (PWR)	Power	none
Red LED (COMM)	COMM Active	none

8.5. Target Board Debug Interfaces (P2 and P3)

The debug connectors P2 (DEBUG_A) and P3 (DEBUG_B) provide access to the debug (C2) pins of the C8051F500 and C8051F502. The debug connectors are used to connect the Serial Adapter or the USB Debug Adapter to the target board for in-circuit debugging and Flash programming. Table 2 shows the DEBUG pin definitions.

Table 2. DEBUG Connector Pin Descriptions

Side A - C8051F500		Side B - C8051F502	
Pin #	**Description**	**Pin #**	**Description**
1	Not Connected	1	Not Connected
2, 3, 9	GND (Ground)	2, 3, 9	GND (Ground)
4	C2D_A	4	P3.0_C2D_B
5	/RST (Reset)	5	/RST_B (Reset)
6	Not Connected	6	P3.0_B
7	/RST/C2CK_A	7	/RST/C2CK_B
8	Not Connected	8	Not Connected
10	USB Power (+5VDC from P2)	10	Not Connected

8.6. Serial Interface (P5)

A USB-to-UART bridge circuit (U5) and USB connector (P5) are provided on the target board to facilitate serial connections to UART0 of the C8051F500 (Side A). The Silicon Labs CP2102 USB-to-UART bridge provides data connectivity between the C8051F500 and the PC via a USB port. The TX and RX signals of UART0 may be connected to the CP2102 by installing shorting blocks on header J17. The shorting block positions for connecting each of these signals to the CP2102 are listed in Table 3. To use this interface, the USB-to-UART device drivers should be installed as described in Section 3.2. "CP210x USB to UART VCP Driver Installation," on page 2.

Table 3. Serial Interface Header (J3) Description

Header Pins	UART0 Pin Description
J17[9–10]	UART_TX (P0.4_A)
J17[11–12]	UART_RX (P0.5_A)

8.7. CAN Interface and Network (TB2)

Both MCUs on the target board are connected to CAN transceivers through headers. The port pins assigned to the CAN peripheral on each MCU are P0.6 (CAN_TX) and P0.7 (CAN_RX). The C8051F500 (Side A) is connected to U3 through the J17 header and the C8051F502 (Side B) is connected to U4 through the J26 header. The two CAN transceivers are connected to each other and form a CAN network. Other external devices can be connected to the CAN network through the TB2 interface. The shorting block positions for connecting the MCUs to the CAN transceivers are listed in Table 4. The pin connections for the external CAN devices are listed in Table 5. The CAN transceivers are powered by the +5VREG node and connected through J8 and J14 headers.

Table 4. CAN Interface Headers (J17 and J26) Description

Header Pins	CAN0 Pin Description
J17[5–6]	CAN_TX (P0.6_A)
J17[7–8]	CAN_RX (P0.7_A)
J26[1–2]	CAN_TX (P0.6_B)
J26[3–4]	CAN_RX (P0.7_B)

Table 5. TB2 External CAN Interface Header Description

Pin #	Pin Description
1	CAN_H
2	CAN_L
3	GND

8.8. LIN Interface and Network (TB1)

Both MCUs on the target board are connected to LIN transceivers through headers. These headers assume that the MCU's crossbars are configured to put the LIN TX and RX pins on port pins P1.0 and P1.1 respectively. See the C8051F50x data sheet for crossbar configuration. The C8051F500 (Side A) is connected to the T1 transceiver through the J17 header and the C8051F502 (Side B) is connected to the T2 transceiver through the J26 header. The two LIN transceivers are connected to each other and form a LIN network. Other external devices can be connected to the LIN network through the TB1 interface. The TB1 interface also provides the option for connecting an external power source so that all LIN transceivers can use the same source voltage. This source voltage can also be used to power the target board. If an external voltage source is not provided, the LIN transceivers use the 12V provided through the P4 wall-wart connector. See Section 8.2. for more power option details. The shorting block positions for connecting the MCUs to the LIN transceivers are listed in Table 6. The pin connections for the external LIN devices are listed in Table 7.

Table 6. LIN Interface Headers (J17 and J26) Description

Header Pins	LIN0 Pin Description
J17[9–10]	LIN_TX (P1.0_A)
J17[11–12]	LIN_RX (P1.1_A)
J26[5-6]	LIN_TX (P1.0_B)
J26[7-8]	LIN_RX (P1.1_B)

Table 7. TB1 External LIN Interface Header Description

Pin #	Pin Description
1	+LIN_V
2	LIN_OUT
3	GND

8.9. Port I/O Connectors (J1-J5 and J27-J29)

Each of the parallel ports of the C8051F500 (Side A) and C8051F502 (Side B) has its own 10-pin header connector. Each connector provides a pin for the corresponding port pins 0-7, +5V VIO, and digital ground. The same pin-out is used for all of the port connectors.

Table 8. Port I/O Connector Pin Description

Pin #	Pin Description
1	Pn.0
2	Pn.1
3	Pn.2
4	Pn.3
5	Pn.4
6	Pn.5
7	Pn.6
8	Pn.7
9	+5V (VIO)
10	GND (Ground)

8.10. Voltage Reference (VREF) Connectors (J22 and J32)

The VREF connectors can be used to connect the VREF pin from the MCU (P0.0) to external 0.1 uF and 4.7 uF decoupling capacitors. The C8051F500 (Side A) device is connected to the capacitors through the J22 header and the C8051F502 (Side B) device connects to its own set of capacitors through J32.

8.11. Expansion Connector (P1)

The 96-pin expansion I/O connector P1 is used to connect daughter boards to the main target board. P1 provides access to many C8051F500 signal pins. Pins for VREGIN, VDD, VIO, and 3.3V are also available. See Table 9 for a complete list of pins available at P1.

The P1 socket connector is manufactured by Hirose Electronic Co. Ltd, part number PCN13-96S-2.54DS, Digi-Key part number H7096-ND. The corresponding plug connector is also manufactured by Hirose Electronic Co. Ltd, part number PCN10-96P-2.54DS, Digi-Key part number H5096-ND.

Table 9. P1 Pin Listing

Pin #	Description	Pin #	Description	Pin #	Description
A-1	+3.3V	B-1	GND	C-1	N/C
A-2	N/C	B-2	N/C	C-2	N/C
A-3	N/C	B-3	N/C	C-3	N/C
A-4	N/C	B-4	N/C	C-4	N/C
A-5	N/C	B-5	N/C	C-5	N/C
A-6	N/C	B-6	N/C	C-6	N/C
A-7	N/C	B-7	N/C	C-7	N/C
A-8	N/C	B-8	N/C	C-8	N/C
A-9	N/C	B-9	N/C	C-9	N/C
A-10	N/C	B-10	P0.7_A	C-10	P0.6_A
A-11	P0.5_A	B-11	P0.4_A	C-11	P_0.3_A
A-12	P_0.2_A	B-12	P0.1_A	C-12	P0.0_A
A-13	P4.7_A	B-13	P4.6_A	C-13	P4.5_A
A-14	P4.4_A	B-14	P4.3_A	C-14	P4.2_A
A-15	P4.1_A	B-15	P4.0_A	C-15	P3.7_A
A-16	P3.6_A	B-16	P3.5_A	C-16	P3.4_A
A-17	P3.3_A	B-17	P3.2_A	C-17	P3.1_A
A-18	P3.0_A	B-18	P2.7_	C-18	P2.6_A
A-19	P2.5_A	B-19	P2.4_	C-19	P2.3_A
A-20	P2.2_A	B-20	P2.1_A	C-20	P2.0_A
A-21	P1.7_A	B-21	P1.6_A	C-21	P1.5_A
A-22	P1.2_A	B-22	P1.3_A	C-22	P1.4_A
A-23	P1.1_A	B-23	P1.0_A	C-23	N/C
A-24	C2D_A	B-24	N/C	C-24	N/C
A-25	/RST_A	B-25	GND	C-25	N/C
A-26	GND	B-26	N/C	C-26	N/C
A-27	N/C	B-27	N/C	C-27	N/C
A-28	N/C	B-28	N/C	C-28	N/C
A-29	VREF0	B-29	VDD_A	C-29	VREGIN_A
A-30	N/C	B-30	N/C	C-30	N/C
A-31	N/C	B-31	N/C	C-31	N/C
A-32	N/C	B-32	AGND	C-32	N/C

8.12. Potentiometer (J20)

The C8051F500 (Side A) device has the option to connect port pin P1.2 to 10K linear potentiometer. The potentimeter is connected through the J20 header. The potentiometer can be used for testing the analog-to-digital (ADC) converter of the MCU.

8.13. Power Supply I/O (Side A) (TB3)

All of the C8051F500 target device's supply pins are connected to the TB3 terminal block. Refer to Table 10 for the TB3 terminal block connections.

Table 10. TB1 Terminal Block Pin Descriptions

Pin #	Description
1	VIO_A
2	VREGIN_A
3	VDD_A
4	VDDA_A
5	GNDA_A
6	GND

8.14. C2 Pin Sharing

On the C8051F500 (Side A), the debug pin C2CK is shared with the /RST pin. On the C8051F502 (Side B), the debug pins C2CK and C2D are shared with the pins /RST and P3.0 respectively. The target board includes the resistors necessary to enable pin sharing which allow the pin–shared pins (/RST and P3.) 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.

8.15. Target Board Pin Assignment Summary

Some GPIO pins of the C8051F500 MCU can have an alternate fixed function. For example, pin 46 on the C8051F500 MCU is designated P0.4, and can be used as a GPIO pin. Also, if the UART0 peripheral on the MCU is enabled using the crossbar registers, the TX signal is routed to this pin. This is shown in the "Alternate Fixed Function" column. The "Target Board Function" column shows that this pin is used as TX on the ‘F500 Target Board. The "Relevant Headers" column shows that this signal is routed to pin 3 of the J17 header and pin 5 of the J1 header. More details can be found in the C8051F50x data sheet. Some of the GPIO pins of the C8051F500 have been used for various functions on the target board. All pins of the Side A MCU also connect to the 96-pin (P1) expansion connector which is not explicitly listed below. Table 11 summarizes the C8051F500 MCU pin assignments on the target board, and also shows the various headers associated with each signal.

Table 11. C8051F500 Target Board Pin Assignments and Headers

MCU Pin Name	Pin#	Primary Function	Alternate Fixed Function	Target Board Function	Relevant Headers
P0.0	8	P0.0	VREF	VREF	J1[1], J22[1]
P0.1	1	P0.1	CNVSTR	CNVSTR	J1[2]
P0.2	48	P0.2	XTAL1	XTAL1	J1[3]*, J9[1]
P0.3	47	P0.3	XTAL2	XTAL2	J1[4]*, J10[1]
P0.4	46	P0.4	UART_TX	TX_MCU	J1[5], J17[3]
P0.5	45	P0.5	UART_RX	RX_MCU	J1[6], J17[1]
P0.6	44	P0.6	CAN_TX	CNVSTR	J1[7], J17[5]
P0.7	43	P0.7	CAN_RX	SW2 (switch)	J1[8], J17[7]
P1.0	42	P1.0		LIN_TX	J2[1], J17[9]
P1.1	41	P1.1		LIN_RX	J2[2], J17[11]
P1.2	40	P1.2		POTENTIOMETER	J2[3], J20[1]
P1.3	39	P1.3		LED	J2[4], J19[3]
P1.4	38	P1.4		SWITCH	J2[5], J19[1]
P1.5	37	P1.5		GPIO	J2[6]
P1.6	36	P1.6		GPIO	J2[7]
P1.7	35	P1.7		GPIO	J2[8]
P2.0	34	P2.0		GPIO	J3[1]
P2.1	33	P2.1		GPIO	J3[2]
P2.2	32	P2.2		GPIO	J3[3]
P2.3	31	P2.3		GPIO	J3[4]
P2.4	30	P2.4		GPIO	J3[5]
P2.5	29	P2.5		GPIO	J3[6]
P2.6	28	P2.6		GPIO	J3[7]
P2.7	27	P2.7		GPIO	J3[8]
P3.0	26	P3.0		GPIO	J4[1]
P3.1	25	P3.1		GPIO	J42]

Table 11. C8051F500 Target Board Pin Assignments and Headers (Continued)

P3.2	24	P3.2		GPIO	J4[3]
P3.3	23	P3.3		GPIO	J4[4]
P3.4	22	P3.4		GPIO	J45]
P3.5	21	P3.5		GPIO	J4[6]
P3.6	20	P3.6		GPIO	J4[7]
P3.7	19	P3.7		GPIO	J4[8]
P4.0	18	P4.0		GPIO	J5[1]
P4.1	17	P4.1		GPIO	J5[2]
P4.2	16	P4.2		GPIO	J5[3]
P4.3	15	P4.3		GPIO	J5[4]
P4.4	14	P4.4		GPIO	J5[5]
P4.5	13	P4.5		GPIO	J5[6]
P4.6	10	P4.6		GPIO	J5[7]
P4.7	9	P4.7		GPIO	J5[8]
/RST/C2CK	12	/RST	C2CK	/RST/C2CK	P2[7], P2[5]*
C2D	11	C2D		C2D	P2[4]
VIO	2	VIO		VIO	J24[4], J18[1], TB3[1] J1-J5[9]
VREGIN	3	VREGIN		VREGIN	J24[2], P2[5]*, TB3[2]
VDD	4	VDD		VDD	TB3[3]
VDDA	5	VDDA		VDDA	TB3[4]
GND	6	GND		GND	J1-J5[10], TB3[6]
GNDA	7	GNDA		VDD	TB3[5]

***Note:** Headers denoted by this symbol are not directly connected to the MCU pin; the connection might be via one or more headers and/or pin-sharing resistor(s). See board schematic for details.

9. Schematics

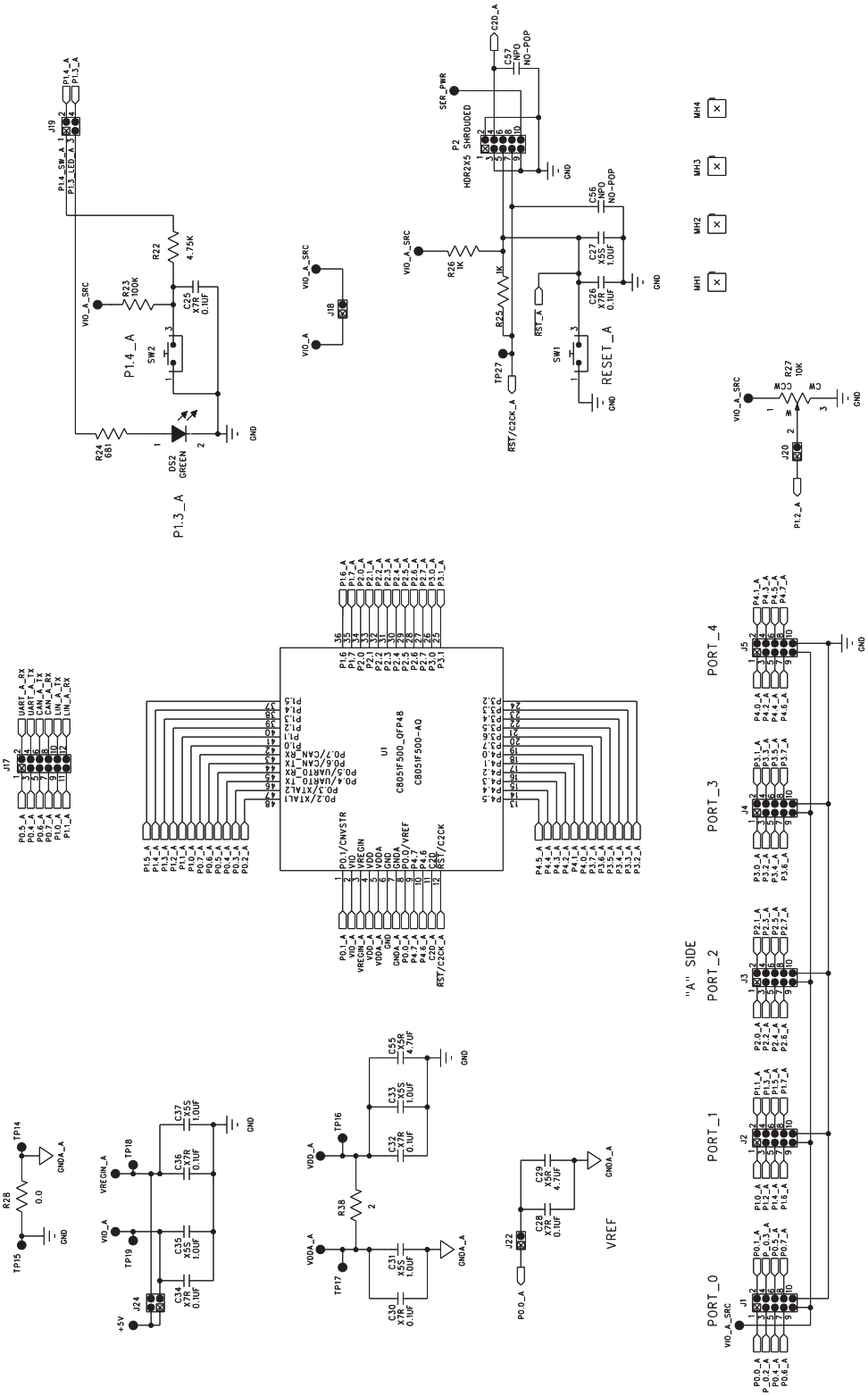

Figure 5. C8051F502 Target Board Schematic (Page 1 of 4)

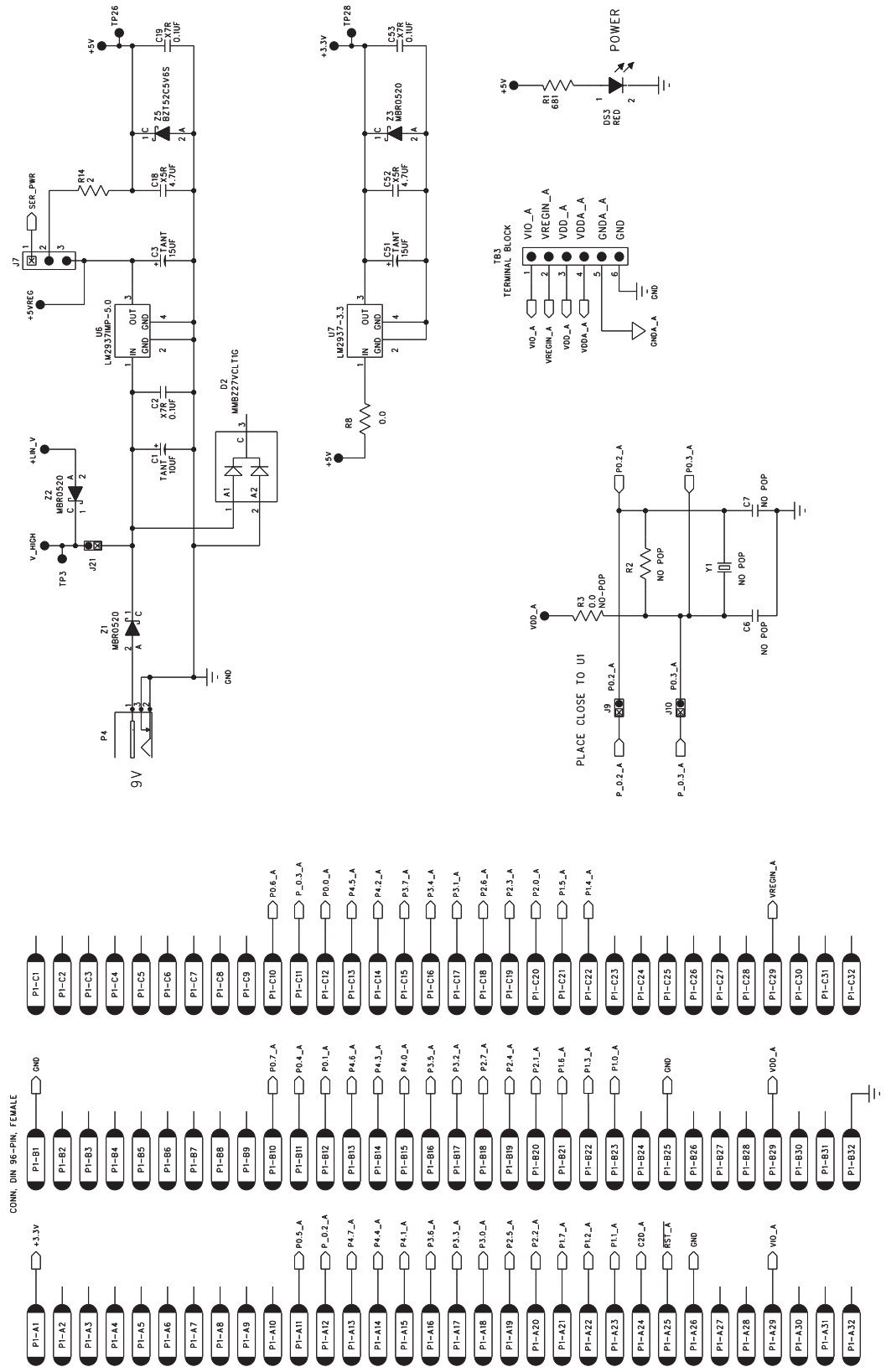

Figure 6. C8051F502 Target Board Schematic (Page 2 of 4)

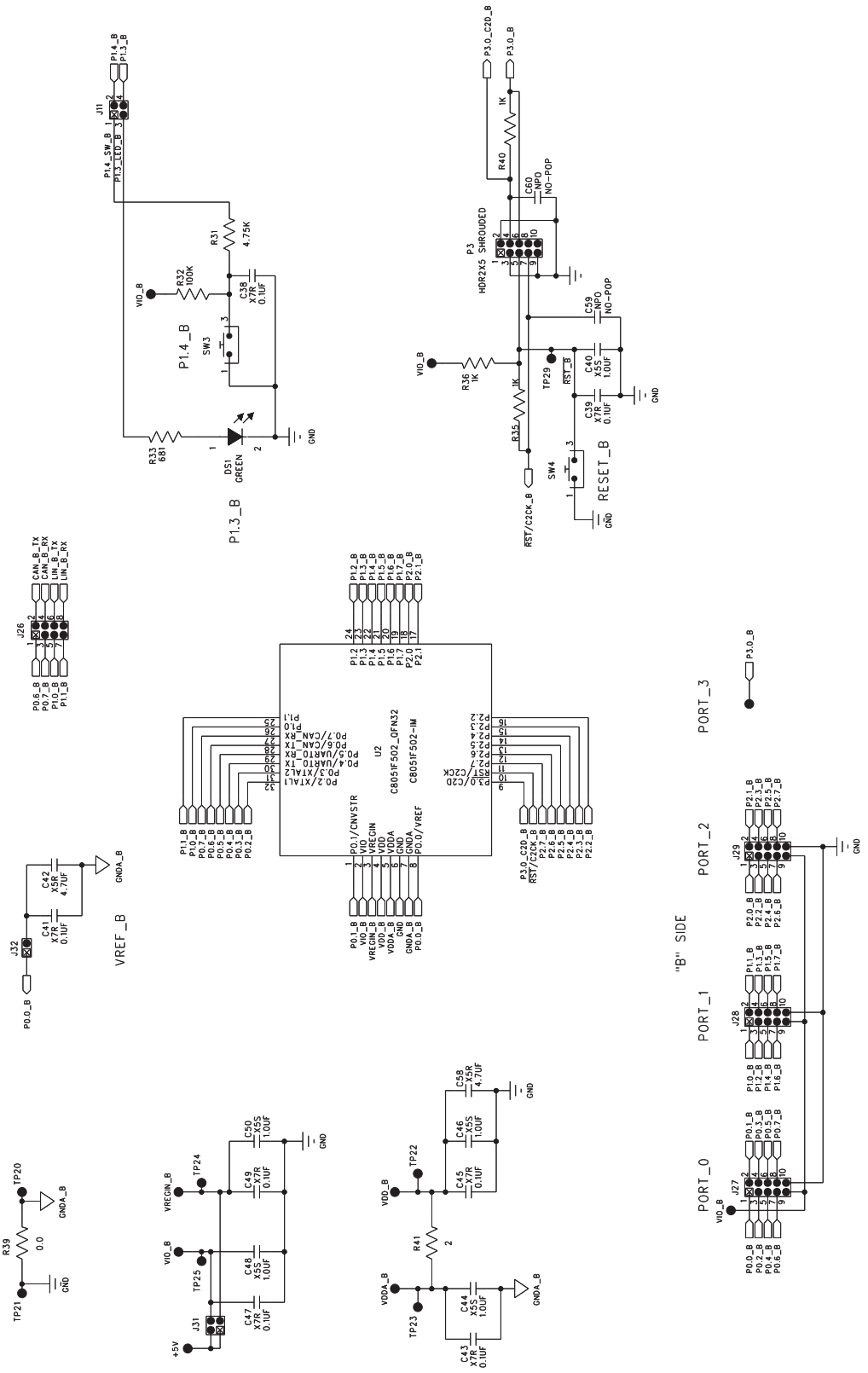

Figure 7. C8051F502 Target Board Schematic (Page 3 of 4)

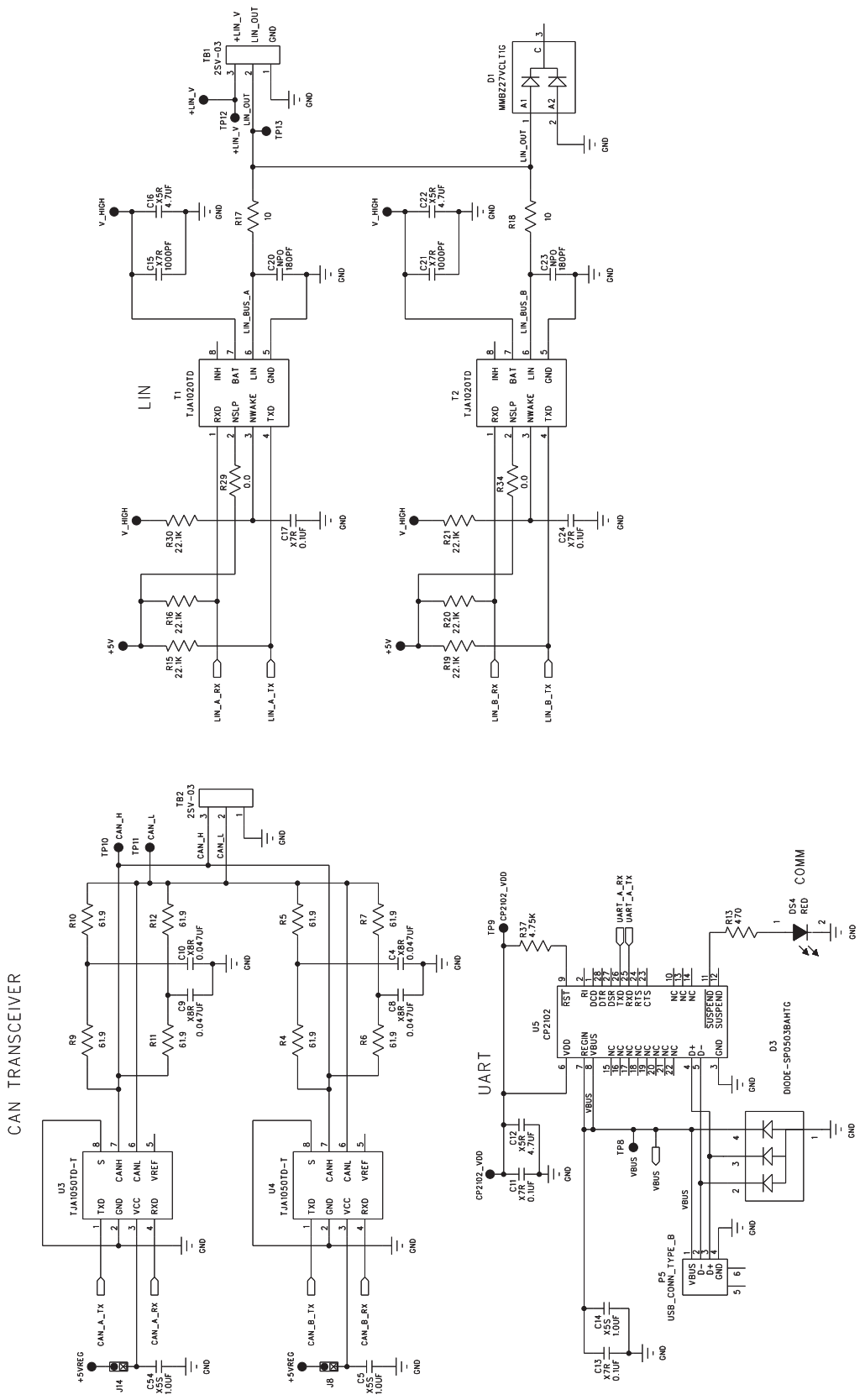

Figure 8. C8051F502 Target Board Schematic (Page 4 of 4)

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: mcutools@silabs.com
Internet: www.silabs.com