C8051F52x-53x-DK

C8051F52x-53x Development Kit User's Guide

1. Relevant Devices

The C8051F52x-53x Development Kit is intended as a development platform for the microcontrollers in the C8051F52x-53x MCU family. The members of this MCU family are C8051F520, C8051F521, C8051F523, C8051F524, C8051F526, C8051F527, C8051F530, C8051F531, C8051F533, C8051F534, C8051F536 and C8051F537.

Notes:

- The target board included in this kit is provided with two pre-soldered C8051F530-IT MCUs (TSSOP-20 package).
- Code developed on the C8051F530 can be easily ported to the other members of this MCU family.
- Refer to the C8051F52x-53x data sheet for the differences between the members of this MCU family.

2. Kit Contents

The C8051F52x-53x Development Kit contains the following items:

- C8051F530 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
    - C8051F52x-53x Development Kit User's Guide (this document)
- AC to DC Power Adapter
- USB Debug Adapter (USB to Debug Interface)
- USB Cable

The development kit target board contains two C8051530 microcontrollers that can communicate through a LIN network. One of the C8051F530 (U1) can also be connected to a CP2102 USB to UART bridge while the other one (U2) can be directly connected to two analog signals and a Voltage Reference Signal Input.

The kit is provided with one USB Debug Adapter but the designer can add a second USB debug adapter so that both microcontrollers can have their firmware debugged at the same time using only one PC and two instances of the Silicon Labs IDE.

3. Getting Started

All software required to develop firmware and communicate with the target microcontroller is included in the CD-ROM. The CD-ROM also includes other useful software.

Below is the software necessary for firmware development and communication 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 the following:

- 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 C8051F530 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 was selected during installation, this will launch a driver "unpacker" utility.

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 (P1) on the C8051F530 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 6 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 as follows:

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

The demo applications for the C8051F530 target board are written to work with the Keil and SDCC toolsets.

4.2. Keil Evaluation Toolset

4.2.1. Keil Assembler and Linker

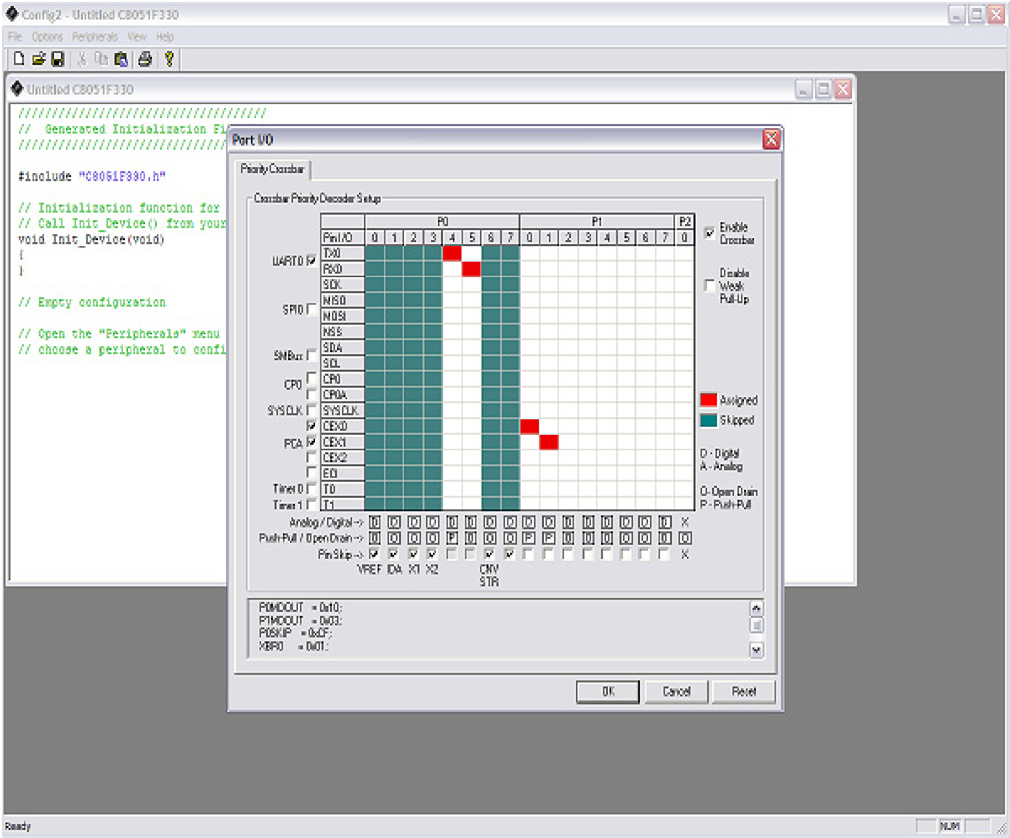
The assembler and linker that are part of the Keil Demonstration Toolset are the same versions that are found in the full Keil Toolset. The complete assembler and linker reference manual can be found on-line under the **Help** menu in the IDE or in the "*SiLabs\MCU\hlp*" directory (A51.chm).

4.2.2. Keil Evaluation C51 C Compiler

The evaluation version of the C51 compiler is the same as the full version with the following limitation: (1) Maximum 4 kB code generation. 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. 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.



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

#include "C8051F330.h"

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

// Empty configuration

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

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

Figure 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, 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.

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 connectors on the target board (HDR1 or HDR2) with the 10-pin ribbon cable. The recommended connection is to the HDR1 (connected to U1) as this microcontroller can be connected to the CP2102 USB to UART bridge.
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 P5 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.

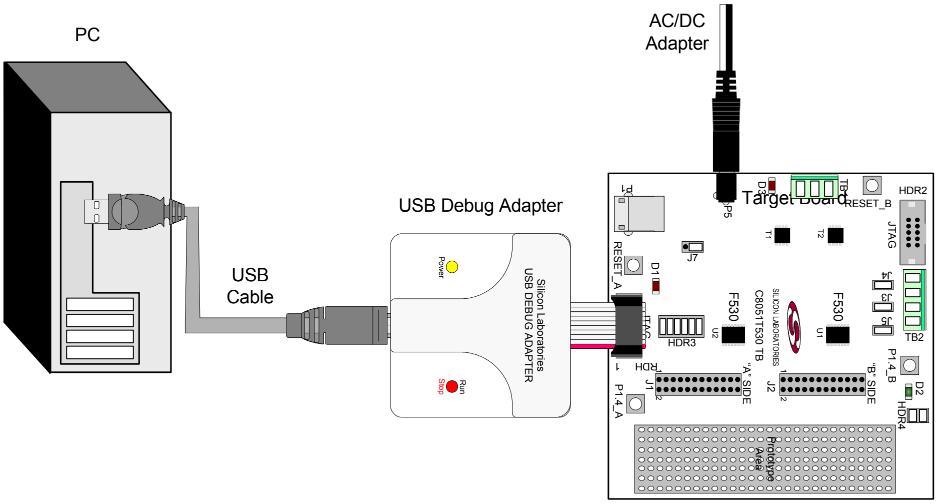


Figure 2. Hardware Setup using a USB Debug Adapter

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. 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. C8051F52x-53x 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\C8051F52x_53x*" 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, *C8051F530.inc and C8051F530.h*, define all SFR registers and bit-addressable control/ status bits. They are installed into the "*SiLabs\MCU\Examples\C8051F52x_53x*" directory during IDE installation. The register and bit names are identical to those used in the C8051F52x-53x data sheet. Both 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, *blink.asm* and *blinky.c*, show examples of several basic C8051F52x-53x 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. When compiled/assembled and linked, this program flashes the green LED on the C8051F530 target board associated with the microcontroller about five times a second using the interrupt handler with a C8051F530 timer.

8. Target Board

The C8051F52x-53x Development Kit includes a target board with two C8051F530 devices 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.

P5	Power connector (Accepts input from 7 to 15 VDC unregulated power adapter.)
D3	RED Power-on LED
TB1	LIN connector

J1	24-pin Expansion I/O connector for U2 (side A)
HDR1	Debug connector for Debug Adapter Interface (side A - U2)
P1	USB connector to serial interface (CP2102)
D1	Green LED (side A, U2), P1.3-A
Reset_A	Reset button (side A, U2)
P1.4_A	Push button (side A, U2)
HDR3	Connector block for serial port connection, Green LED, and push-button Side A
J7	Power connection block selection for the CP2102
T1	LIN transceiver (side A - U2)
U2	C8051F530 side A

J2	24-pin Expansion I/O connector for U1 (side B)
HDR2	Debug connector for Debug Adapter Interface (side B - U1)
P1.4-B	Push button (Side B, U1)
J5	VREFIN connector block
T2	LIN transceiver (side B - U1)
Reset_B	Reset button (side B, U1)
U1	C8051F530 side B
TB2	Analog input connector
HDR4	Green LED and push-button connector block, Side B
D2	Green LED (side B, U1), P1.3_B
J3	Channel 1, P1.6_B connector block
J4	Channel 2, P1.7_B connector block

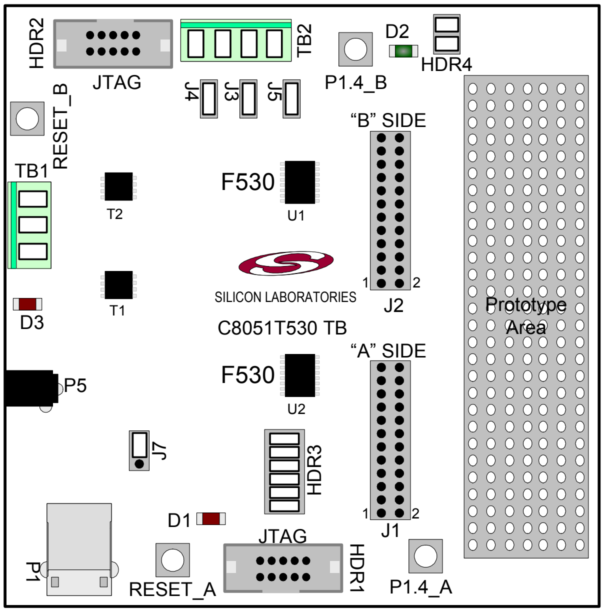


Figure 3. C8051F530 Target Board

8.1. System Clock Sources

The C8051F530 device installed on the target board features a calibrated programmable internal oscillator which is enabled as the system clock source on reset. After reset, the internal oscillator operates at a frequency of 3.0625 MHz (±0.5%) by default but may be configured by software to operate at other frequencies. Therefore, in many applications an external oscillator is not required. However, if you wish to operate the C8051F530 device at a frequency not available with the internal oscillator, an external crystal may be used. Refer to the C8051F52x-53x data sheet for more information on configuring the system clock source.

The target board is designed to facilitate the installation of external crystals. Install the crystals at the pads marked Y1 or Y2. Install a 10 MΩ resistor at R17 or R22 and install capacitors at (C29 and C30) or (C34 and C35) using values appropriate for the crystals you select. Refer to the C8051F52x-53x data sheet for more information on the use of external oscillators.

8.2. Switches and LEDs

Four switches are provided on the target board.

Switch RESET_A is connected to the RESET pin of the C8051F530 A-Side (U2).

Switch RESET_B is connected to the RESET pin of the C8051F530 B-Side (U1).

Pressing RESET_A or RESET_B puts the attached device into its hardware-reset state.

Switches P1.4_A and P1.4_B are connected to the C8051F530 parts (U1 and U2) general purpose I/O (GPIO) pins through headers. Pressing P1.4_A or P1.4_B generates a logic low signal on the port pin of the respective microcontroller.

Remove the shorting block from the header to disconnect P1.4_A or P1.4_B from the port pins. The port pin signals are also routed to pins on the J1 and J2 I/O connectors. See Table 1 for the port pins and headers corresponding to each switch.

Three LEDs are also provided on the target board. The red LED labeled WR is used to indicate a power connection to the target board. The green LEDs labeled D1 and D2 are connected to the C8051F530's GPIO pins through headers. Remove the shorting blocks from the headers to disconnect the LEDs from the port pins. The port pin signals are also routed to pins on the J1 and J2 I/O connectors. See Table 1 for the port pins and headers corresponding to each LED.

Table 1. Target Board I/O Descriptions

Description	I/O	Header
Reset_A	U2-Reset	none
Reset_B	U1-Reset	none
P1.4_A	U2-P1.4	HDR3[3–4]
P1.4_B	U1-P1.4	HDR4[3–4]
Green LED D1	U2-P1.3	HDR3[1–2]
Green LED D2	U1-P1.3	HDR4[1–2]
Red LED	PWR	none

8.3. Expansion I/O Connectors (J1, J2)

The two 24-pin Expansion I/O connectors J1 and J2 provide access to all signal pins of the C8051F530 devices. Pins for V_{DD}, GND, 5 V, Reset, Vbat, LIN, 3.3 V and Vrefin are also available. A small through-hole prototyping area is also provided.

All I/O signals routed to connectors J1 and J2 are also routed to through-hole connection points between J1 and J2 and the prototyping area (see Figure 3 on page 8). Each connection point is labeled indicating the signal available at the connection point. See Table 2 for a list of pin descriptions for J1 and J2.

Table 2. Pin Descriptions for J1 and J2

J1					J2			
Pin #	Description	Pin #	Description		Pin #	Description	Pin #	Description
1	P0.0_A	13	P1.4_A		1	P0.0_B	13	P1.4_B
2	P0.1_A	14	P1.5_A		2	P0.1_B	14	P1.5_B
3	P0.2_A	15	P1.6_A		3	P0.2_B	15	P1.6_B
4	P0.3_A	16	P1.7_A		4	P0.3_B	16	P1.7_B
5	P0.4_A	17	+5V		5	P0.4_B	17	+5V
6	P0.5_A	18	RST/C2CLK_A		6	P0.5_B	18	RST/C2CLK_B
7	P0.6_A	19	VBAT		7	P0.6_B	19	VBAT
8	P0.7_A	20	LIN		8	P0.7_B	20	LIN
9	P1.0_A	21	+3.3V		9	P1.0_B	21	VREFIN
10	P1.1_A	22	V_{DD} of U1		10	P1.1_B	22	V_{DD} of U2
11	P1.2_A	23	GND		11	P1.2_B	23	GND
12	P1.3_A	24	GND		12	P1.3_B	24	GND

8.4. Target Board DEBUG Interface (HDR1, HDR2)

The DEBUG connectors (HDR1 and HDR2) provide access to the DEBUG (C2) pins of the C8051F530 parts. They are used to connect the USB Debug Adapter to the target board for in-circuit debugging and Flash programming. Table 3 shows the DEBUG pin definitions.

Table 3. DEBUG Connector Pin Descriptions

Pin #	Description
1	+3 VD (+3.3 VDC)
2, 3, 9	GND (Ground)
4	C2D
5	/RST (Reset)
6	P3.0
7	C2CK
8	Not Connected
10	USB Power

8.5. USB to Serial Connector (P1)

A USB-to-Serial bridge interface is provided. A B type connector (P1), a CP2102, and related circuits are provided to facilitate the serial connection between a PC and one of the C8051F530 (U2) microcontrollers on the target board. The RX, TX, CTS, and RTS signals of the UART side of the Bridge (CP2102) may be connected to the microcontroller by installing shorting blocks on HDR3 as follows:

Table 4. UART Connections

HDR3	
Connection	**Signals**
5–6	P0.4_A_TX to TX_A
7–8	P0.4_A_RX to RX_A
9–10	P1.1_A to RTS_A
11–12	P1.2_A to CTS_A

The selection of a power source for the CP2102 is made through the J7 header. The option between BUS-Powered and SELF-Powered is selected as follows:

Table 5. CP210x Power Connections

J7	
Connection	**Signals**
1–2	BUS-Powered
2–3	SELF_Powered

The BUS-Powered option allows the CP2102 to use the 5 V provided by the USB interface while the SELF-Powered option allows the CP2102 to use the 5 V supply from the target board.

8.6. Analog I/O (TB2)

The Analog connector block (TB2) provides Analog inputs to the C8051F530 (U1) as follows:

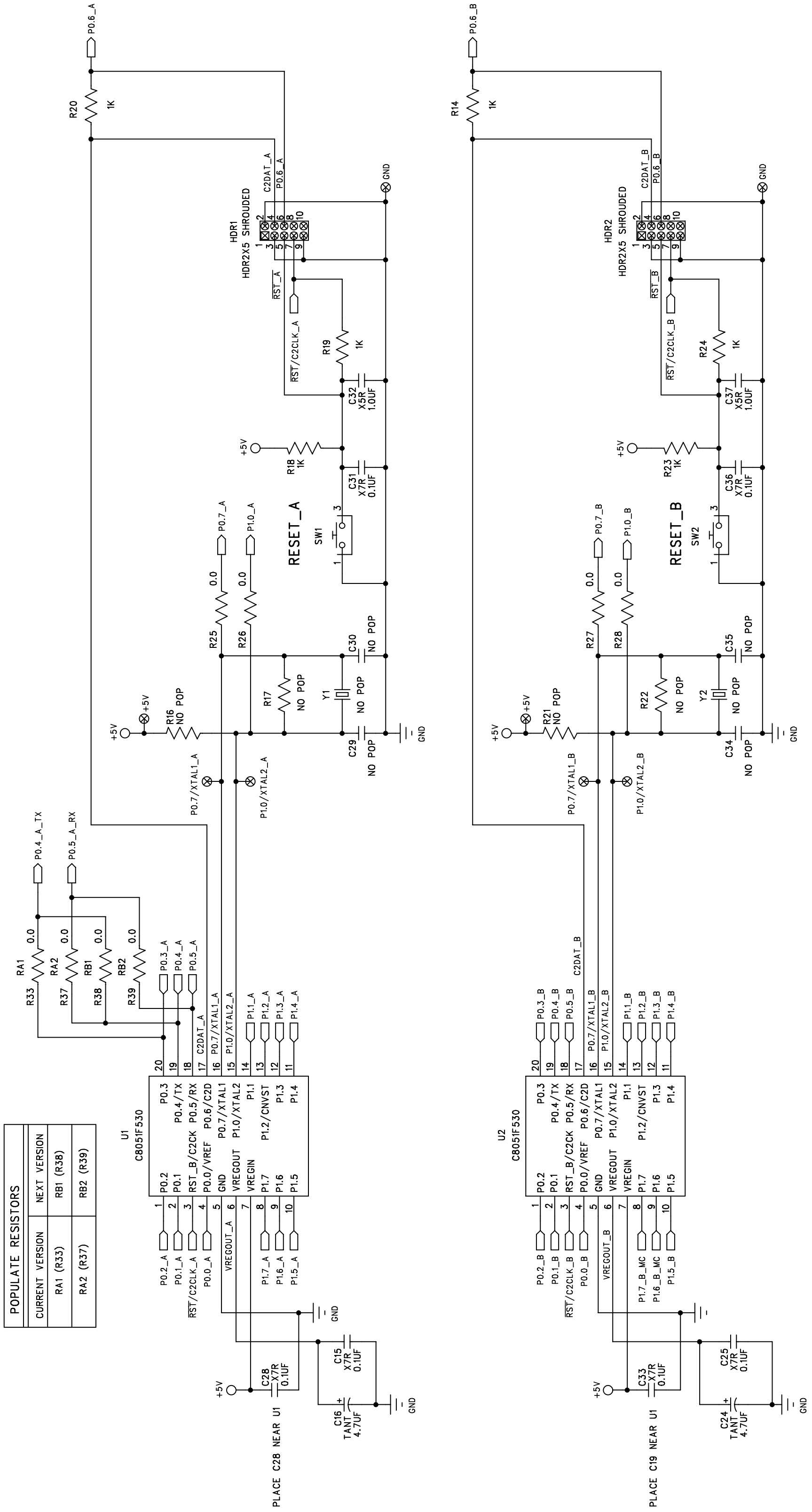
Table 6. Analog I/O Connections

TB2			
Signal	**Connection**	**I/O**	**Shorting Block**
Vrefin	External Reference Input or Internal Reference Output	P0.0_B	J5
CH1	Analog Input 1	P1.6_B_MC	J3
CH2	Analog Input 1	P1.7_B_MC	J4
GND	Ground	GND	-

8.7. Power Supply Options

The target board provides two options of power supply. The first option is to use the provided 9 V power supply attached to the P5 connector. The second option is to use an external 12 V (7.5 V minimum) connected to the TB1 (pins 1 and 3).

9. Schematics



POPULATE RESISTORS	
CURRENT VERSION	NEXT VERSION
RA1 (R33)	RB1 (R38)
RA2 (R37)	RB2 (R39)

Figure 4. C8051F52x-53x Schematic (1 of 3)

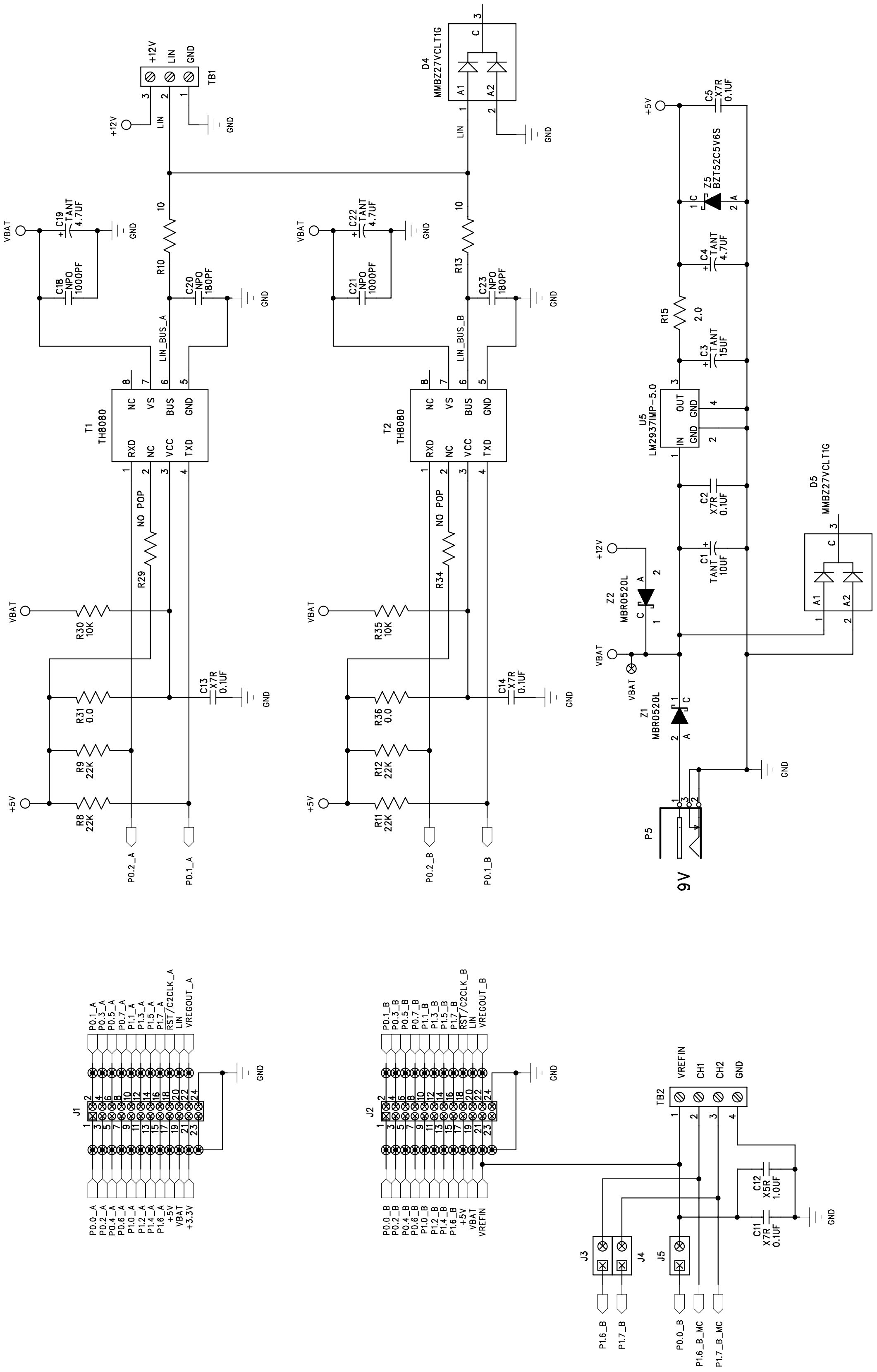


Figure 5. C8051F52x-53x Schematic (2 of 3)

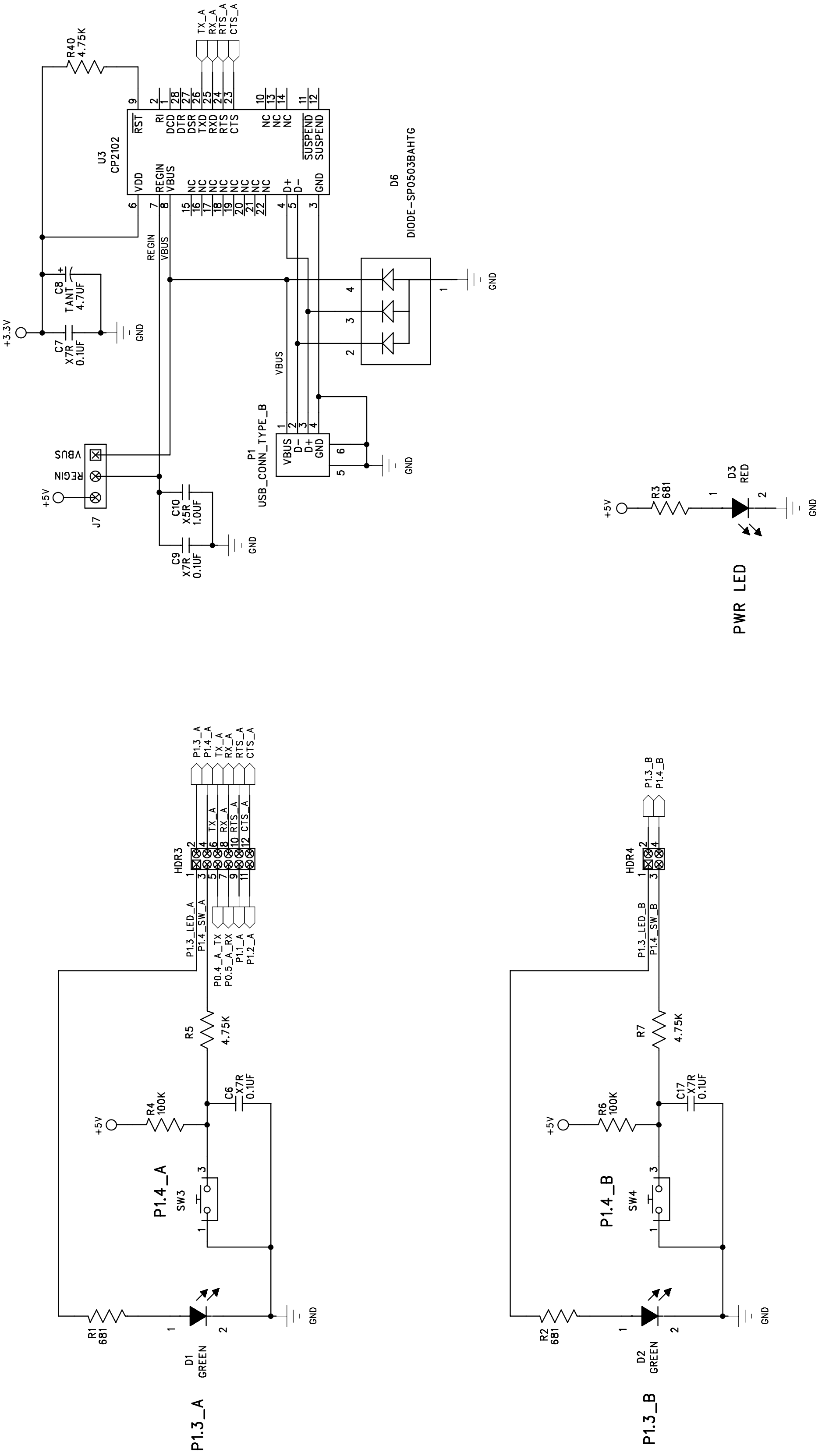


Figure 6. C8051F52x-53x Schematic (3 of 3)

DOCUMENT CHANGE LIST

Revision 0.1 to Revision 0.2

- Added Relevant Devices section.
- Section 2 moved to Section 5.
- Changed section 3 to "Getting Started."
- Updated section 3 to include latest VCP driver installation instructions.
- Changed section 4 to "Software Overview."
- Updated Evaluation Compiler restrictions in section 4.2.2.
- Added overview of Configuration Wizard 2 and Keil uVision Drivers to section 4.
- Created new section 6.

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

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.