C8051F41x Development Kit User's Guide

1. Kit Contents

The C8051F41x Development Kit contains the following items:

- C8051F410 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
  - C8051F41x Development Kit User's Guide (this document)
- AC to DC Power Adapter
- USB Debug Adapter (USB to Debug Interface)
- USB Cable

2. 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 1.

1. Connect the USB Debug Adapter to the DEBUG connector on the target board with the 10-pin ribbon cable.
2. Connect one end of the USB cable to the USB connector on the USB Debug Adapter.
3. 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 P1 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.

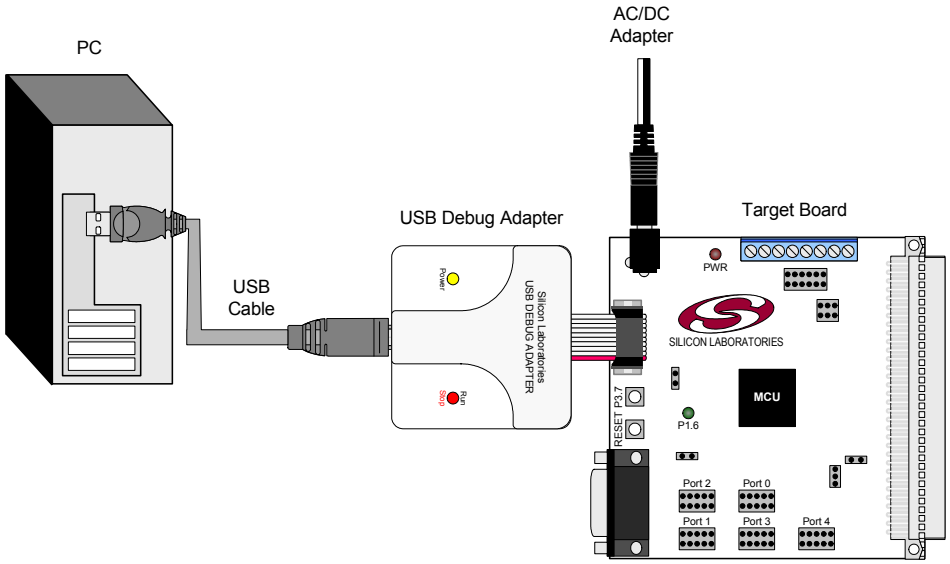

Figure 1. Hardware Setup using a USB Debug Adapter

3. Software Setup

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 *readme.txt* file on the CD-ROM for the latest information regarding known IDE problems and restrictions.

4. Silicon Laboratories Integrated Development Environment

The Silicon Laboratories IDE integrates a source-code editor, source-level debugger and in-system Flash programmer. 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.

4.1. System Requirements

The Silicon Laboratories IDE requirements:

- Pentium-class host PC running Microsoft Windows 98SE or later.
- One available COM or USB port.
- 64 MB RAM and 40 MB free HD space recommended.

4.2. 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 under the **Help** menu in the IDE or in the "*SiLabs\MCU\hlp*" directory (A51.pdf).

4.3. 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 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).

4.4. 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 Applications Note **AN104 - Integrating Keil 8051 Tools Into the Silicon Labs IDE** in the "*SiLabs\MCU\Documentation\Appnotes*" directory on the CD-ROM 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.)

4.4.1. 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 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.

4.4.2. 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 appropriate adapter in the "Serial Adapter" section. Next, the correct "Debug Interface" must be selected. C8051F41x 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**.

5. Example Source Code

Example source code and register definition files are provided in the "*SiLabs\MCU\Examples\C8051F41x*" 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.

5.1. Register Definition Files

Register definition files *C8051F410.inc and C8051F410.h* define all SFR registers and bit-addressable control/status bits. They are installed into the "*SiLabs\MCU\Examples\C8051F41x*" directory during IDE installation. The register and bit names are identical to those used in the C8051F41x 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.

5.2. Blinking LED Example

The example source files *blink.asm* and *blinky.c* show examples of several basic C8051F41x 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 C8051F410 target board about five times a second using the interrupt handler with a C8051F410 timer.

6. Target Board

The C8051F41x Development Kit includes a target board with a C8051F410 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 2 for the locations of the various I/O connectors.

P1	Power connector (accepts input from 7 to 15 VDC unregulated power adapter)
J1	22-pin Expansion I/O connector
J3	Port I/O Configuration Jumper Block
J4	DEBUG connector for Debug Adapter interface
J5	DB-9 connector for UART0 RS232 interface
J6	Analog I/O terminal block
J7	Connector for IDAC0 voltage circuit
J8	USB Debug Adapter target board power connector
J9,	J10 External crystal enable connectors
J11	Connector for IDAC1 voltage circuit
J12	Connector block for Thermistor circuitry
J13, J14	ADC external voltage reference connectors

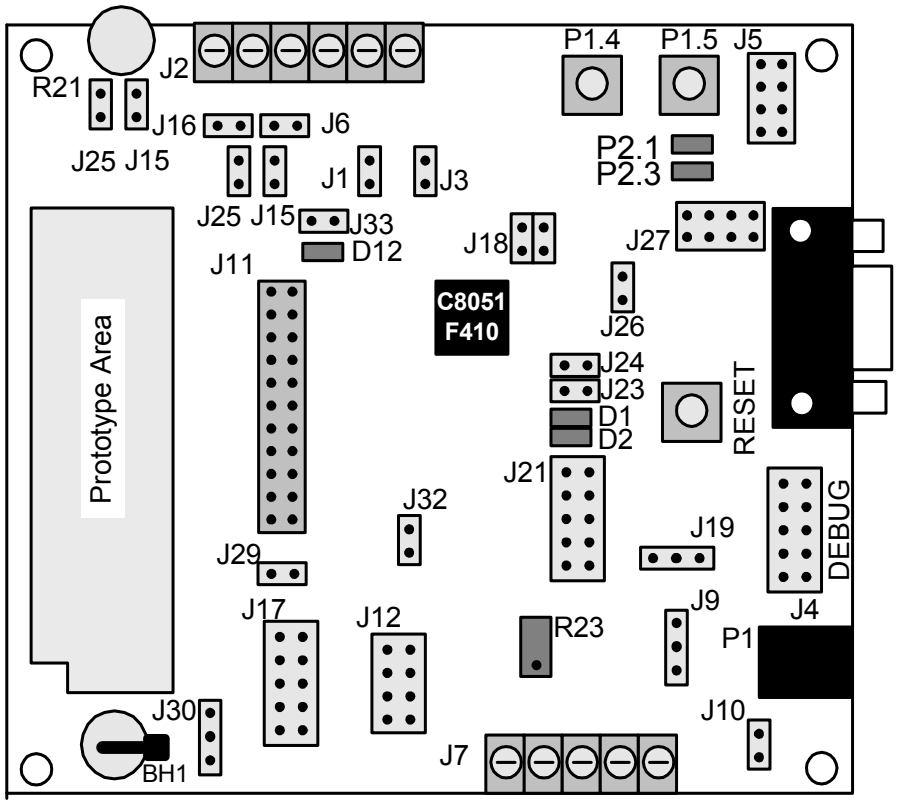

Figure 2. C8051F410 Target Board

6.1. System Clock Sources

The C8051F410 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 191.4 kHz (±2%) 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 C8051F410 device at a frequency not available with the internal oscillator, an external crystal may be used. Refer to the C8051F41x data sheet for more information on configuring the system clock source.

The target board is designed to facilitate the installation of an external crystal. Remove shorting blocks at headers J9 and J18 and install the crystal at the pads marked Y2. Install a 10 MΩ resistor at R4 and install capacitors at C44 and C43 using values appropriate for the crystal you select. Refer to the C8051F41x data sheet for more information on the use of external oscillators. The target board also has a 32.768 kHz watch crystal installed to provide a timebase for the smaRTClock. Jumper J26 may be used to short the XTAL3 and XTAL4 pins if internal clock mode is desired.

6.2. Power Options

The C8051F41x Target Board has many power options. This allows the user to exercise the different operating modes of the C8051F410. The board has 2 voltage regulators. A 3.3 V LDO and a 1.2–5.25 V variable regulator.

To use the 3.3 V regulator, pin 2–3 of J9 should be shorted and a jumper installed in J10.

To use the variable regulator, pin 1–2 of J19 should be shorted and an output voltage should be selected using J21. After selecting the appropriate voltage, pin 1–2 of J21 should be shorted to enable the output of the variable regulator. If the voltage "VAR" is selected, the potentiometer R23 should be adjusted until the desired voltage is reached. The input to the variable regulator can be obtained from the unregulated 9 V supply or from the 5 V USB VBUS supply available when using a USB debug adapter.

Note: Before enabling either voltage regulator, the user should check the 4 supply rail selection headers (J29+J30, J17, J12, J31+32) to ensure the correct voltage is being routed to the correct power pin. An incorrect jumper setting may permanently damage the board. Note that VDD cannot exceed 2.5 V and is typically derived from the on-chip regulator. Do not connect the 5.25 or 3.3 V output directly to VDD.

The three power LEDs for VIO, VREG, and VDD indicate if the appropriate supply rail is connected to a power supply. Check to make sure all supply rails (with exception of VREG if not using the on-chip regulator) are powered.

For the VIO voltage rail, the user may choose from the 3 V regulator (+3 VD), the variable regulator, or the on-chip regulator. The selections are marked on the target board silkscreen.

For the VREG on-chip voltage regulator input, the user may choose from the 3 V regulator (+3VD), the 5 V USB VBUS source obtained from the USB debug adapter (5VEC3), or from the variable regulator (VREG). The selections are marked on the target board silkscreen.

For VDD, the user may choose the output of the on-chip regulator (VDD_) or the output of the variable regulator (VREG).

6.3. Switches and LEDs

Three switches are provided on the target board. Switch SW1 is connected to the RESET pin of the C8051F410. Pressing SW1 puts the device into its hardware-reset state. Switch SW2 and SW3 are connected to the C8051F410's general purpose I/O (GPIO) pins through headers. Pressing SW2 or SW3 generates a logic low signal on the port pin. Remove the shorting block from the jumper J5 to disconnect SW2 and/or SW3 from their associated port pins. The port pin signals are also routed to pins on the J11 I/O connector. 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 PWR is used to indicate a power connection to the target board. The green LEDs labeled with port pin names are connected to the C8051F410'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 I/O connector. See Table 1 for the port pins and headers corresponding to each LED.

Table 1. Target Board I/O Descriptions

Description	I/O	Jumper
SW1	Reset	none
SW2	P1.4	J5[3–4]
SW3	P1.5	J5[7–8]
Green LED D3	P2.1	J5[1–2]
Green LED D5	P2.3	J5[5–6]
Red LED D1	VREGIN	J24
Red LED D2	VDD	J23
Red LED D12	VIO	J33

6.4. Expansion I/O Connector (J1)

The 24-pin Expansion I/O connector J11 provides access to all signal pins of the C8051F410 device. A small through-hole prototyping area is also provided. All I/O signals routed to connector J1 are also routed to through-hole connection points between J11 and the prototyping area (see Figure 2 on page 5). Each connection point is labeled indicating the signal available at the connection point. See Table 2 for a list of pin descriptions for J1.

Table 2. J1 Pin Descriptions

Pin #	Description	Pin #	Description
1	P0.0	13	P1.4
2	P0.1	14	P1.5
3	P0.2	15	P1.6
4	P0.3	16	P1.7
5	P0.4	17	P2.0
6	P0.5	18	P2.1
7	P0.6	19	P2.2
8	P0.7	20	P2.3
9	P1.0	21	P2.4
10	P1.1	22	P2.5
11	P1.2	23	P2.6
12	P1.3	24	P2.7

6.5. Target Board DEBUG Interface (J4)

The DEBUG connector (J4) provides access to the DEBUG (C2) pins of the C8051F410. It is used to connect the Serial Adapter or the USB Debug Adapter to the target board for in-circuit debugging and Flash programming. Table 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

6.6. Serial Interface (J5)

A RS232 transceiver circuit and DB-9 (J5) connector are provided on the target board to facilitate serial connections to UART0 of the C8051F410. The TX, RX, RTS and CTS signals of UART0 may be connected to the DB-9 connector and transceiver by installing shorting blocks on header J3.

J27[1–2] - Install shorting block to connect UART0 TX (P0.4) to transceiver.
J27[3–4] - Install shorting block to connect UART0 RX (P0.5) to transceiver.
J27[5–6] - Install shorting block to connect UART0 RTS (P1.4) to transceiver.
J27[7–8] - Install shorting block to connect UART0 CTS (P1.5) to transceiver.

6.7. Analog I/O (J6)

Many of the C8051F410 target device's port pins are connected to the J2 terminal block. Connections for VDDA, AGND, ADC external voltage references, IDAC outputs and ADC inputs are available. Refer to Table 4 for the J6 terminal block connections.

Table 4. J6 Terminal Block Pin Descriptions

Pin #	Description
1	P0.0/IDAC0
2	P0.1/IDAC1
3	VREFIN
4	GND
5	AIN0
6	AIN1

6.8. IDAC Connectors (J7, J11)

The C8051F410 target board also features two Current-to-Voltage 750 Ω load resistors that may be connected to the 2-bit current-mode Digital-to-Analog Converters (IDACs) on port pins P1.0 and P1.1. Install a shorting block on J13 to connect the IDAC0/P1.0 pin of the target device to a load resistor. Install a second shorting block on J14 to connect the IDAC1/P1.1 pin of the target device to a load resistor. The IDAC signals are then routed to the J1 and J13 and J14 connectors.

6.9. USB Debug Adapter Target Board Power Connector (J8)

The USB Debug Adapter includes a connection to provide power to the target board. This connection is routed from J4[10] to J8[1]. Place a shorting block at header J8[2–3] to power the board directly from an ac/dc power adapter. Place a shorting block at header J8[1-2] to power the board from the USB Debug Adapter. The second option is not supported with either the EC1 or EC2 Serial Adapters.

6.10. smaRTClock (Real Time Clock)

The C8051F41x Target Board is designed for developing system using the on-chip smaRTClock. A 32 kHz watch crystal is installed in Y1 and a battery holder (BH1) is installed for use as a backup power source. To connect the battery to the VBAT input on the MCU, connect pin 1 and 2 of J30 and short J29. The variable voltage regulator with potentiometer (R23) may also be used to generate the battery voltage and simulate a battery in its various charge conditions. See the C8051F41x data sheet for more details about the smaRTClock.

7. Schematics

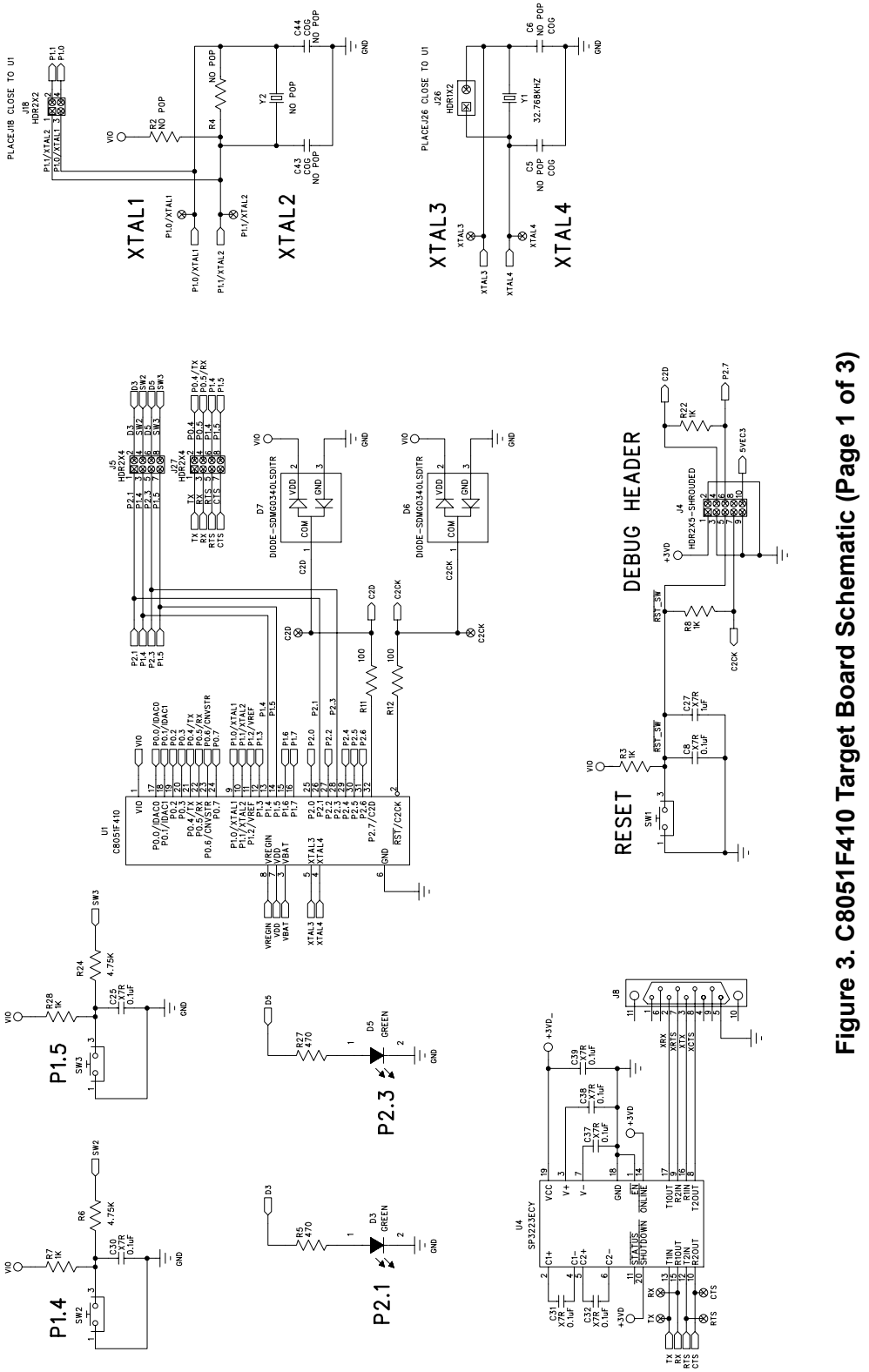

Figure 3. C8051F410 Target Board Schematic (Page 1 of 3)

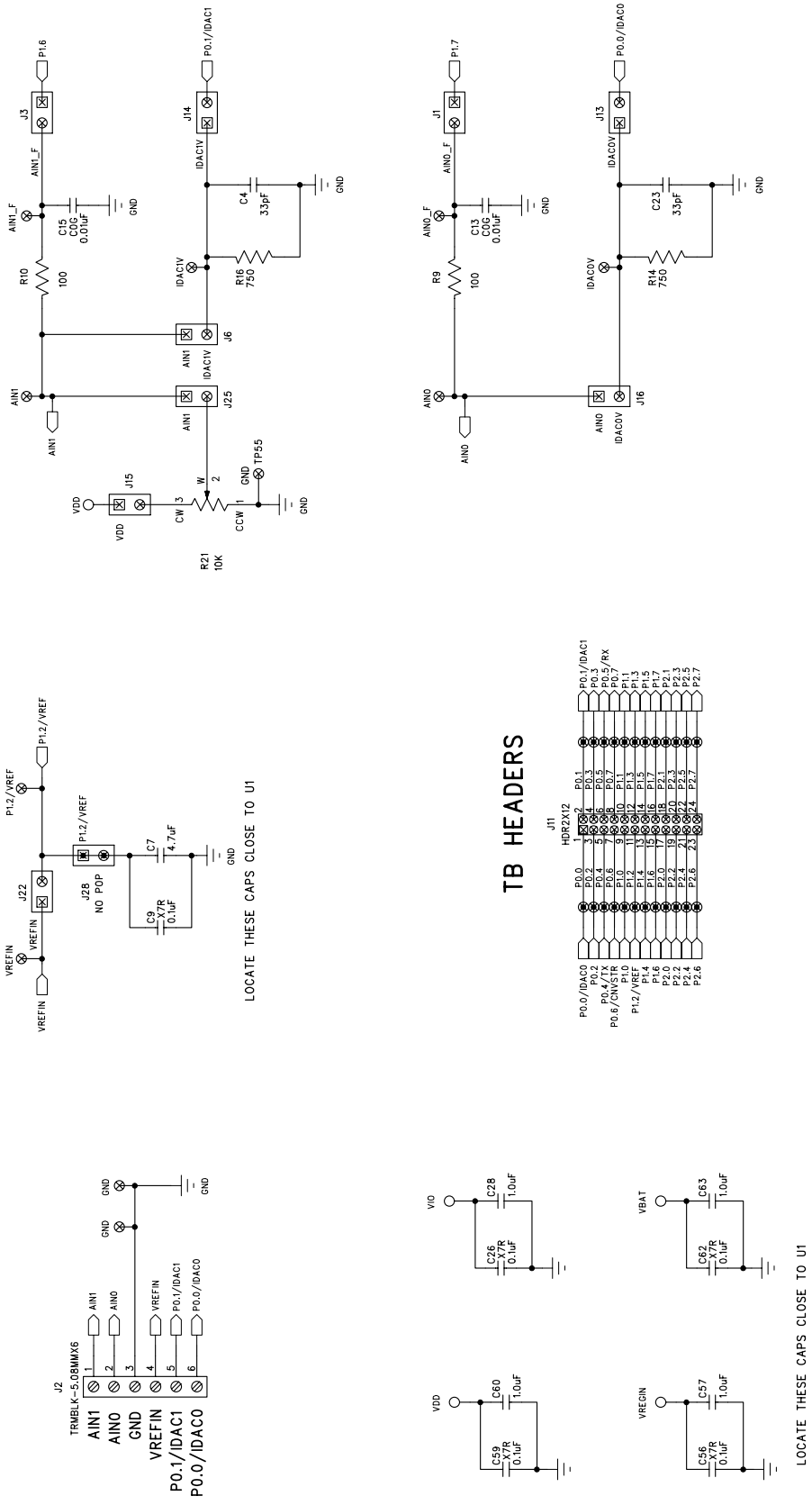

Figure 4. C8051F410 Target Board Schematic (Page 2 of 3)

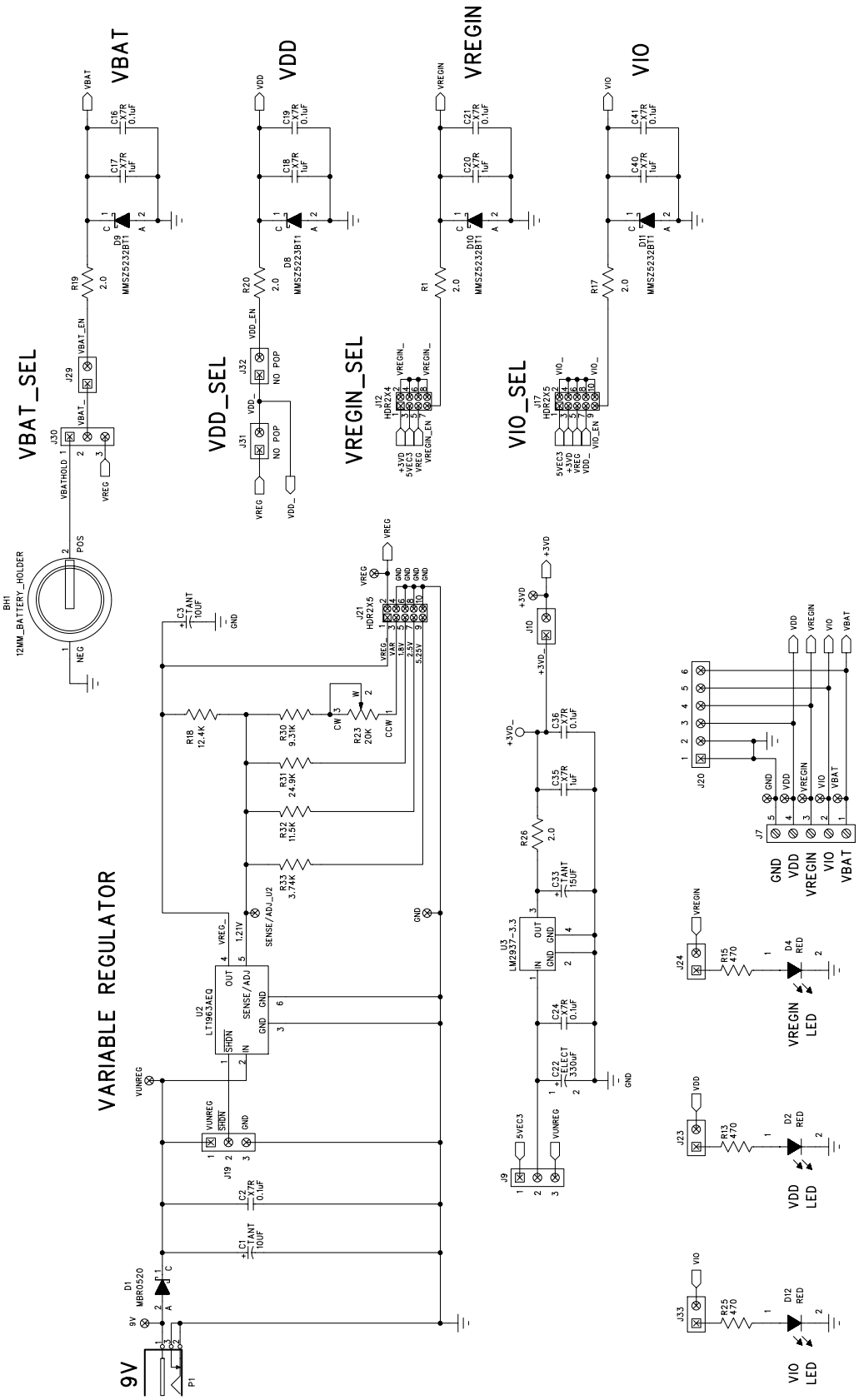

Figure 5. C8051F410 Target Board Schematic (Page 3 of 3)

DOCUMENT CHANGE LIST

Revision 0.2 to Revision 0.3

- Removed Section 9. USB Debug Adapter. See USB Debug Adapter User's Guide.

CONTACT INFORMATION

Silicon Laboratories Inc.

4635 Boston Lane
Austin, TX 78735
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

Email: MCUinfo@silabs.com
Internet: www.silabs.com