C8051F12x Development Kit User's Guide

1. Kit Contents

The C8051F12x Development Kit contains the following items:

- C8051F120 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
  - C8051F12x 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 JTAG 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.

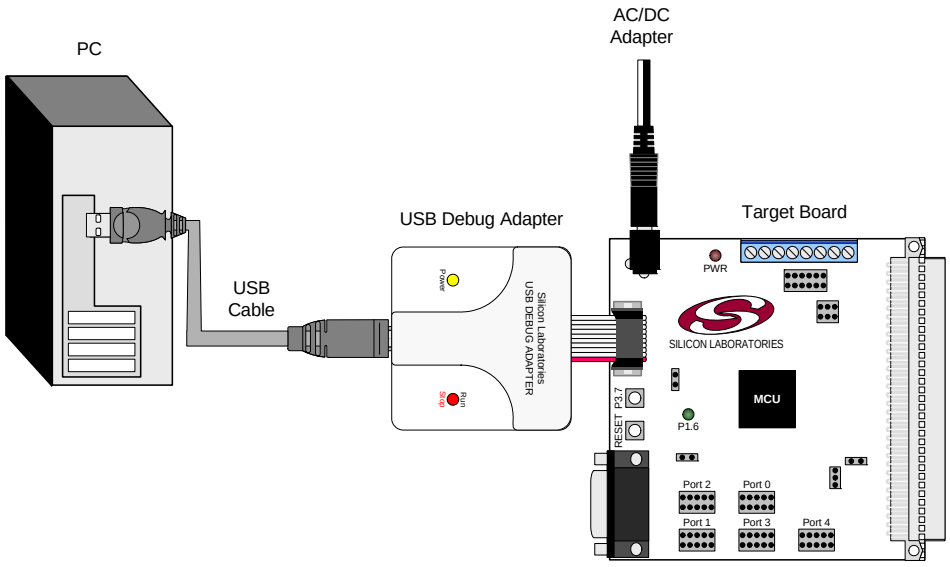


Figure 1. Hardware Setup using a USB Debug Adapter

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 Laboratories 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. C8051F12x family devices use the JTAG 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\C8051F12x*" 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 *C8051F120.inc and C8051F120.h* define all SFR registers and bit-addressable control/status bits. They are installed into the "*SiLabs\MCU\Examples\C8051F12x*" directory during IDE installation. The register and bit names are identical to those used in the C8051F12x 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. Code Banking Files

In order to utilize code banking in a project there are two files that will need to be changed. You can either custom edit L51_bank.a51 and startup.a51 or use the copies installed into the "*IDEfiles\C51\Lib*" directory during IDE installation. Both of these files MUST be added to your project and linked with your project in order for code banking to function properly. For more information on code banking, please see Application Note **AN130 - Code Banking Using the Keil 8051 Tools**.

5.3. Blinking LED Example

The example source files *blink.asm* and *blinky.c* show examples of several basic C8051F12x 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 C8051F120 target board about five times a second using the interrupt handler with a C8051F120 timer.

6. Target Board

The C8051F12x Development Kit includes a target board with a C8051F120 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	Connects SW2 to P3.7 pin
J3	Connects LED D3 to P1.6 pin
J4	JTAG connector for Debug Adapter interface
J5	DB-9 connector for UART0 RS232 interface
J6	Connector for UART0 TX (P0.0)
J8	Connector for UART0 RTS (P4.0)
J9	Connector for UART0 RX (P0.1)
J10	Connector for UART0 CTS (P4.1)
J11	Analog loopback connector
J12–J19	Port 0–7 connectors
J20	Analog I/O terminal block
J22	VREF connector
J23	VDD Monitor Disable
J24	96-pin Expansion I/O connector

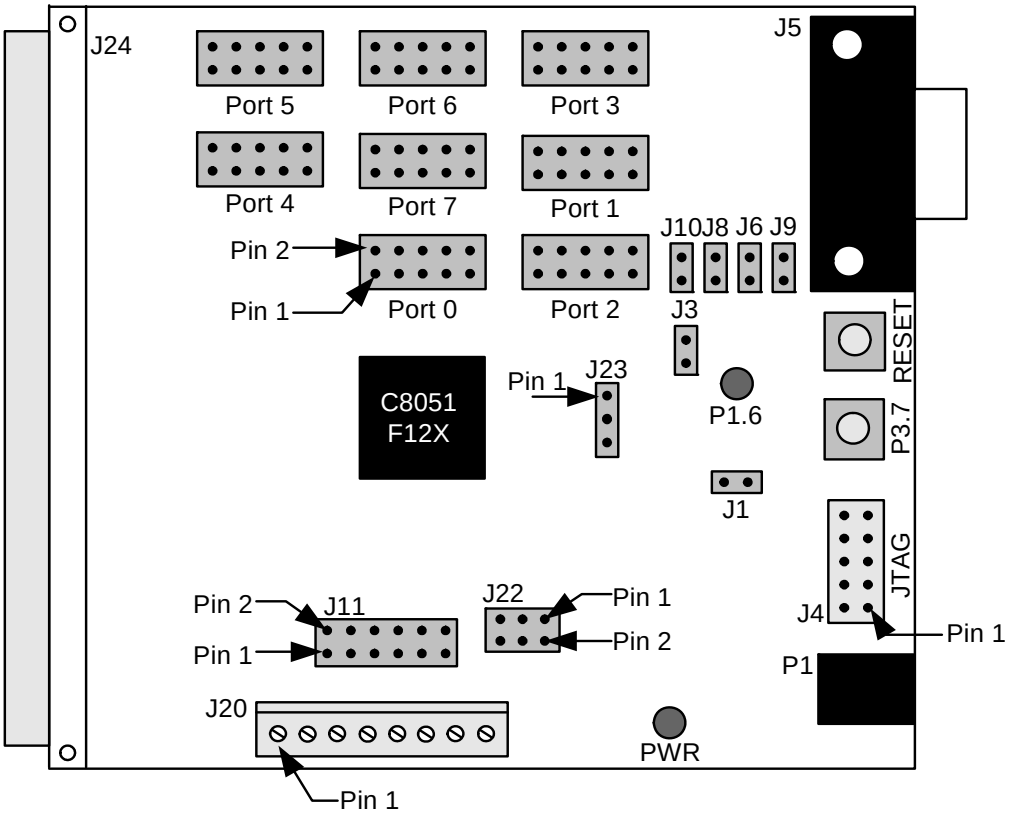


Figure 2. C8051F120 Target Board

6.1. System Clock Sources

The C8051F120 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 3.0625MHz (+/-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, an external 22.1184 MHz crystal is installed on the target board for additional applications. Refer to the C8051F12x data sheet for more information on configuring the system clock source.

6.2. Switches and LEDs

Two switches are provided on the target board. Switch SW1 is connected to the RESET pin of the C8051F120. Pressing SW1 puts the device into its hardware-reset state. Switch SW2 is connected to the C8051F120's general purpose I/O (GPIO) pin through headers. Pressing SW2 generates a logic low signal on the port pin. Remove the shorting block from the header to disconnect SW2 from the port pins. The port pin signal is also routed to a pin on the J24 I/O connector. See Table 1 for the port pins and headers corresponding to each switch.

Two 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 LED labeled with a port pin name is connected to the C8051F120's GPIO pin through headers. Remove the shorting block from the header to disconnect the LED from the port pin. The port pin signal is also routed to a pin on the J24 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	Header
SW1	Reset	none
SW2	P3.7	J1
Green LED	P1.6	J3
Red LED	PWR	none

6.3. Target Board JTAG Interface (J4)

The JTAG connector (J4) provides access to the JTAG pins of the C8051F120. 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 2 shows the JTAG pin definitions.

Table 2. JTAG Connector Pin Descriptions

Pin #	Description
1	+3VD (+3.3VDC)
2, 3, 9	GND (Ground)
4	TCK
5	TMS
6	TDO
7	TDI
8, 10	Not Connected

6.4. 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 C8051F120. The TX, RX, RTS and CTS signals of UART0 may be connected to the DB-9 connector and transceiver by installing shorting blocks on headers J6, J8, J9 and J10.

J6 - Install shorting block to connect UART0 TX (P0.0) to the transceiver.
J9 - Install shorting block to connect UART0 RX (P0.1) to the transceiver.
J8 - Install shorting block to connect UART0 RTS (P4.0) to the transceiver.
J10 - Install shorting block to connect UART0 CTS (P4.1) to the transceiver.

6.5. Analog I/O (J11, J20)

Several C8051F120 analog signals are routed to the J20 terminal block and the J11 header. The J11 connector provides the ability to connect DAC0 and DAC1 outputs to several different analog inputs by installing a shorting block between a DAC output and an analog input on adjacent pins of J11. Refer to Table 3 for J20 terminal block connections and Table 4 for J11 pin definitions.

Table 3. J20 Terminal Block Pin Descriptions

Pin #	Description
1	CP0+
2	CP0-
3	DAC0
4	DAC1
5	AIN0.0
6	AIN0.1
7	VREF0
8	ADND (Analog Ground)

Table 4. J11 Connector Pin Descriptions

Pin #	Description
1	CP0+
2	CP0-
3	DAC0
4	DAC1
5	CP1+
6	CP1-
7	AIN0.0
8	AIN0.1
9	DAC0
10	DAC1
11	AIN0.6
12	AIN0.7

6.6. PORT I/O Connectors (J12 - J19)

In addition to all port I/O signals being routed to the 96-pin expansion connector, each of the eight parallel ports of the C8051F120 has its own 10-pin header connector. Each connector provides a pin for the corresponding port pins 0-7, +3.3 VDC and digital ground. Table 5 defines the pins for the port connectors. The same pin-out order is used for all of the port connectors.

Table 5. J12- J19 Port Connector Pin Descriptions

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	+3 VD (+3.3 VDC)
10	GND (Ground)

6.7. VDD Monitor Disable (J23)

The VDD Monitor of the C8051F120 may be disabled by moving the shorting block on J23 from pins 1-2 to pins 2-3, as shown in Figure 3.

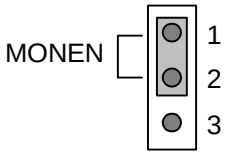


Figure 3. VDD Monitor Hardware Setup

6.8. VREF Connector (J22)

The VREF connector (J22) can be used to connect the VREF (Voltage Reference) output of the C8051F120 to any (or all) of its voltage reference inputs. Install shorting blocks on J22 in the following manner:

1–2 to connect VREF to VREFD
3–4 to connect VREF to VREF0
5–6 to connect VREF to VREF1

6.9. Expansion I/O Connector (J24)

The 96-pin expansion I/O connector J24 is used to connect daughter boards to the main target board. J24 provides access to many C8051F120 signal pins. Pins for +3 V, digital ground, analog ground and the unregulated power supply (VUNREG) are also available. The VUNREG pin is connected directly to the unregulated +V pin of the P1 power connector. See Table 6 for a complete list of pins available at J24.

The J24 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 6. J24 Pin Descriptions

Pin #	Description	Pin #	Description	Pin #	Description
A-1	+3 VD2 (+3.3 VDC)	B-1	DGND (Digital Gnd)	C-1	XTAL1
A-2	MONEN	B-2	P1.7	C-2	P1.6
A-3	P1.5	B-3	P1.4	C-3	P1.3
A-4	P1.2	B-4	P1.1	C-4	P1.0
A-5	P2.7	B-5	P2.6	C-5	P2.5
A-6	P2.4	B-6	P2.3	C-6	P2.2
A-7	P2.1	B-7	P2.0	C-7	P3.7
A-8	P3.6	B-8	P3.5	C-8	P3.4
A-9	P3.3	B-9	P3.2	C-9	P3.1
A-10	P3.0	B-10	P0.7	C-10	P0.6
A-11	P0.5	B-11	P0.4	C-11	P0.3
A-12	P0.2	B-12	P0.1	C-12	P0.0
A-13	P7.7	B-13	P7.6	C-13	P7.5
A-14	P7.4	B-14	P7.3	C-14	P7.2
A-15	P7.1	B-15	P7.0	C-15	P6.7
A-16	P6.6	B-16	P6.5	C-16	P6.4
A-17	P6.3	B-17	P6.2	C-17	P6.1
A-18	P6.0	B-18	P5.7	C-18	P5.6
A-19	P5.5	B-19	P5.4	C-19	P5.3
A-20	P5.2	B-20	P5.1	C-20	P5.0
A-21	P4.7	B-21	P4.6	C-21	P4.5
A-22	P4.4	B-22	P4.3	C-22	P4.2
A-23	P4.1	B-23	P4.0	C-23	TMS
A-24	TCK	B-24	TDI	C-24	TDO
A-25	/RST	B-25	DGND (Digital Gnd)	C-25	VUNREG
A-26	AGND (Analog Gnd)	B-26	DAC1	C-26	DAC0
A-27	CP1-	B-27	CP1+	C-27	CP0-
A-28	CP0+	B-28	VREF	C-28	VREFD
A-29	VREF0	B-29	VREF1	C-29	AIN0.7
A-30	AIN0.6	B-30	AIN0.5	C-30	AIN0.4
A-31	AIN0.3	B-31	AIN0.2	C-31	AIN0.1
A-32	AIN0.0	B-32	AGND (Analog Gnd)	C-32	AV+ (+3.3 VDC Analog)

7. Schematic

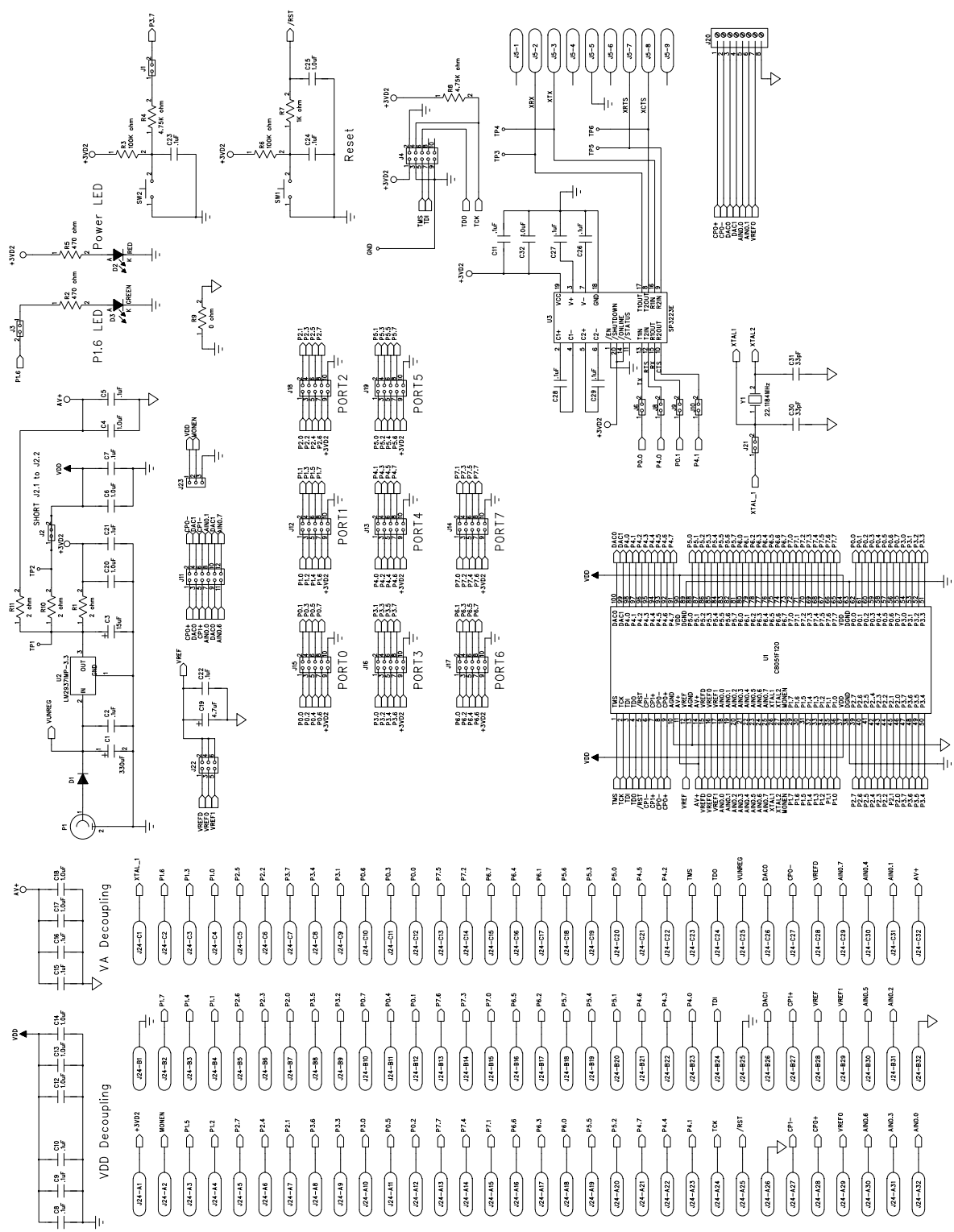


Figure 4. C8051F120 Target Board Schematic

Document Change List

Revision 0.4 to Revision 0.5

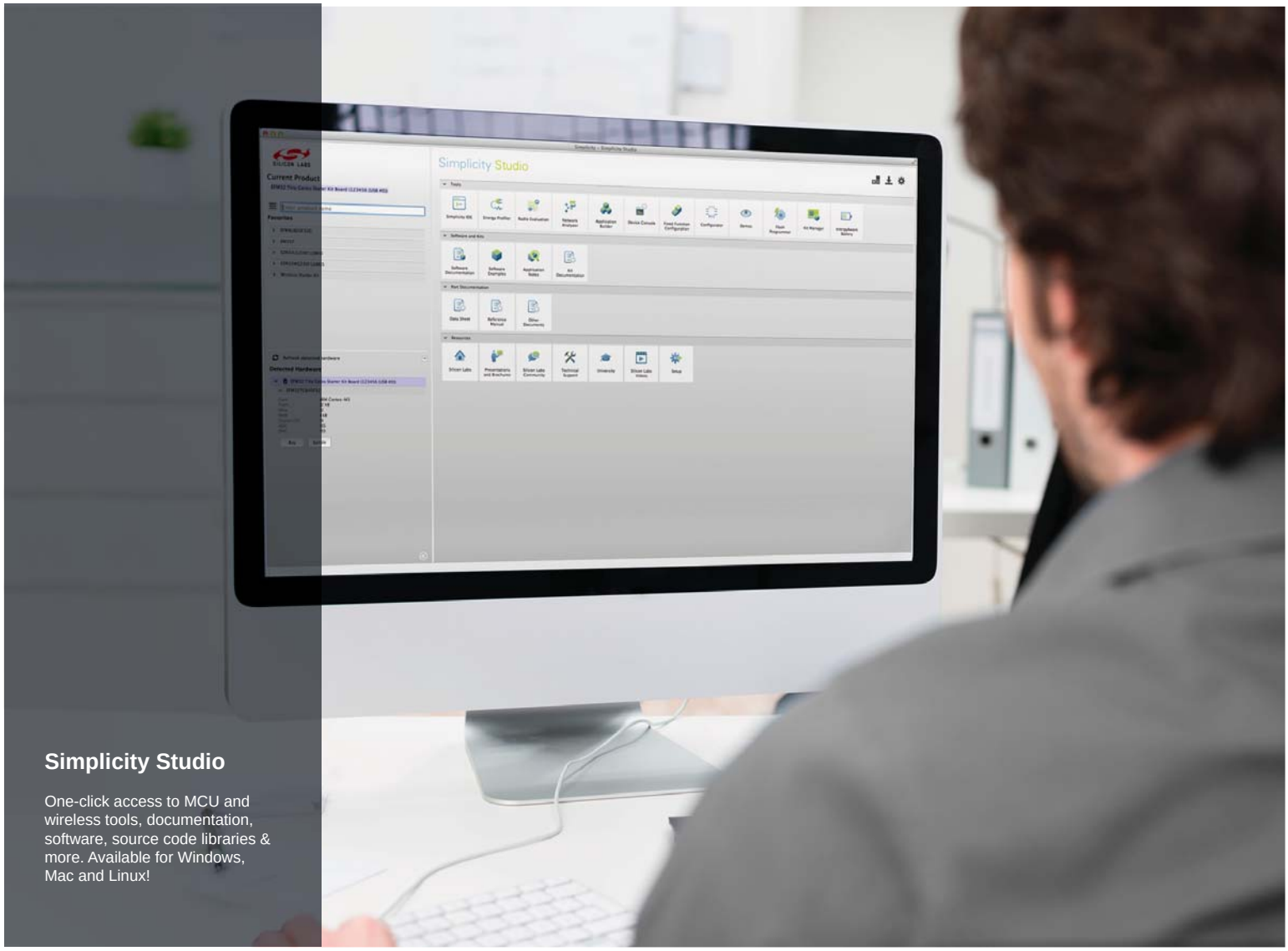
- Changed C8051F124 to C8051F120.

Revision 0.5 to Revision 0.6

- Section 1, added USB Debug Adapter and USB Cable.
- Section 2, changed name from "Hardware Setup" to "Hardware Setup using an EC2 Serial Adapter".
- Section 2, added 2 Notes bullets.
- Section 2, removed Note from bottom of page.
- Added Section 3, "Hardware Setup using a USB Debug Adapter".
- Section 5.4.2, changed step 2 to include new instructions.
- Section 7, J4, changed "Serial Adapter" to "Debug Adapter".
- Target Board DEBUG Interface Section, added USB Debug Adapter.
- DEBUG Connector Pin Descriptions Table, changed pin 4 to C2D.
- Changed "jumper" to "header".
- EC2 Serial Adapter section, added EC2 to the section title, table title and figure title.
- EC2 Serial Adapter section, changed "JTAG" to "DEBUG".
- Added "USB Debug Adapter" section.
- Added J8 and J10 to the figure in the Target Board section.
- Added J8 and J10 to the connector list.

Revision 0.6 to Revision 0.7

- Removed EC2 Serial Adapter from Kit Contents.
- Removed Section 2. Hardware Setup using an EC2 Serial Adapter. See RS232 Serial Adapter (EC2) User's Guide.
- Removed Section 8. EC2 Serial Adapter. See RS232 Serial Adapter (EC2) User's Guide.
- Removed Section 9. USB Debug Adapter. See USB Debug Adapter User's Guide.



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Silicon Laboratories:

C8051F120DK