C8051T61x Development Kit User's Guide

1. Kit Contents

The C8051T61x development kit contains the following items:

- C8051T610 main board
- C8051T610 TQFP 32-pin socketed daughter board for programming TQFP devices
- C8051T610 emulation daughter board with C8051F310 installed
- Five C8051T610-GQ samples
- C8051T61x development kit quick-start guide
- Product information CD-ROM including:
  - Silicon Laboratories Integrated Development Environment (IDE)
  - Evaluation version of Keil 8051 development tools (macro assembler, linker, "C" compiler)
  - Source code examples and register definition files
  - Documentation
- AC-to-DC Universal Power adapter
- Two USB cables

Also available for purchase separately are C8051T610 QFN Socket Daughter Boards for programming QFN-28 and QFN-24 devices.

2. About the Daughter Boards

The C8051T61x development kit includes an Emulation Daughter Board (EDB) and a TQFP Socket Daughter Board (TQFP-DB). The EDB has an installed C8051F310 device, which is a FLASH-based device that can be used for the majority of C8051T61x code development. The TQFP-DB, as well as the 28-pin and 24-pin QFN daughter boards (available separately), are intended to allow both programming and system-level debugging of C8051T61x devices.

A C8051T61x device cannot be erased once it has been programmed; so, it is advisable to use the C8051F310 for the majority of code development. Refer to "AN330: C8051F310 to C8051T610 Porting Guide" for more details on how the C8051F310 can be used to develop code for the C8051T61x device family.

3. Software Overview

This section provides an introduction to the software tools included with the C8051T61x Development Kit. Software covered in this section includes the Silicon Laboratories IDE, the Keil Toolset, the Configuration Wizard 2, Keil uVision, and the ToolStick Terminal application.

3.1. Silicon Laboratories IDE

The Silicon Laboratories IDE integrates a source-code editor, source-level debugger, and an in-system Flash programmer. The Keil demonstration toolset includes a compiler, linker, and assembler and easily integrates into the IDE. The use of third-party compilers and assemblers is also supported.

3.1.1. IDE System Requirements

Silicon Laboratories IDE requirements include:

- Pentium-class host PC running Microsoft Windows 2000 or later.
- One available USB port.

3.1.2. Third-Party Toolsets

The Silicon Laboratories IDE has native support for many 8051 compilers. Natively-supported tools are:

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

Please note that the demonstration applications for the C8051T61x development kit are written for the Keil toolset.

3.2. Keil Demonstration Toolset

3.2.1. Keil Assembler and Linker

The Keil Demonstration Toolset assembler and linker place no restrictions on code size. The complete assembler and linker reference manual can be found online under the **Help** menu in the IDE or in the "*SiLabs\MCU\hlp*" directory (A51.pdf).

3.2.2. Keil Demonstration C51 C Compiler

The demonstration version of the C51 compiler is the same as the full version, except code size is limited to 2 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). The code size limitation can be increased to 4 kB by registering the compiler with Keil. See "AN104: Integrating Keil 8051 Tools into the Silicon Labs IDE" for details on registering the evaluation compiler.

3.3. Configuration Wizard

Configuration Wizard is a code generation tool for all Silicon Laboratories devices. Code is generated through the use of dialog boxes for each device peripheral as shown in Figure 1.

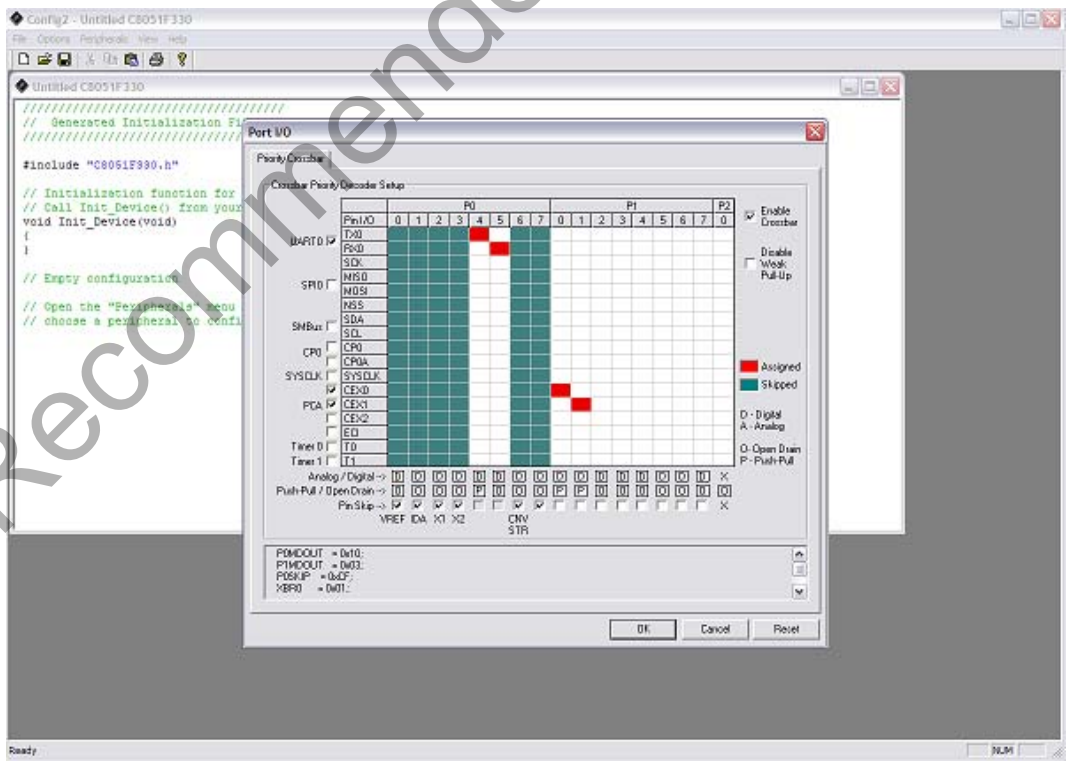

Figure 1. Configuration Wizard 2 Utility

The Configuration Wizard utility helps accelerate development by automatically generating initialization source code to configure and enable the on-chip resources needed by most design projects. In just a few steps, the wizard creates complete startup code for a specific Silicon Laboratories MCU. The program is configurable to provide the output in C or assembly language.

For more information, refer to the Configuration Wizard documentation. Documentation and software is available from the downloads webpage, www.silabs.com/mcudownloads.

3.4. Keil uVision2 and uVision3 Silicon Laboratories Drivers

As an alternative to the Silicon Laboratories IDE, the uVision debug driver allows the Keil uVision2 and uVision3 IDEs to communicate with Silicon Laboratories on-chip debug logic. In-system Flash memory programming integrated into the driver allows for rapid updating of target code. The uVision2 and uVision3 IDEs 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, refer to the uVision driver documentation. The documentation and software are available from the downloads webpage, www.silabs.com/mcudownloads.

3.5. ToolStick Terminal

The onboard debug circuitry provides both an in-system programming and debugging interface and a communications interface to the target microcontroller's UART. The ToolStick Terminal software can access the debug hardware's communications path and provides a terminal-like interface on the PC. Note that, for concurrent debugging and UART communications, the CP2103 USB-to-UART bridge is also included onboard.

In addition to the standard terminal functions (Send File, Receive File, Change Baud Rate), two GPIO pins on the target microcontroller can be controlled using the terminal for either RTS/CTS handshaking or software-configurable purposes. The ToolStick Terminal software is available on the ToolStick webpage, www.silabs.com/toolstick.

4. Hardware Setup

See Figure 1 for a diagram of the hardware configuration.

1. Attach the desired daughter board to the main board at connectors P1 and P2.
2. If using the TQFP Socket Daughter Board or either of the two QFN daughter boards, place the device to be programmed into the socket.
3. Place shorting blocks on J7 and the +3VD-VDD_PWR jumper pair on J6, as shown in Figure 1.
4. Connect the main board’s P5 USB connector to a PC running the Silicon Laboratories IDE using the USB Cable.
5. Connect the ac-to-dc power adapter to connector P3 on the main board.

Notes:

- Use the Reset icon in the IDE to reset the target when connected during a debug session.
- Remove power from the main board and remove the USB cable before removing a daughter board from the main board. Connecting or disconnecting a daughter board when the power adapter or USB cable are connected can damage the main board, the daughter board, or the socketed device.
- Remove power from the main board and remove the USB cable before removing a C8051T61x device from the socket. Inserting or removing a device from the socket when the power adapter or USB cable are connected can damage the main board, the daughter board, or the socketed device.
- The above hardware setup instructions configure the development system to be powered through the onboard 3.3 V regulator. For other power options, see 7.3.

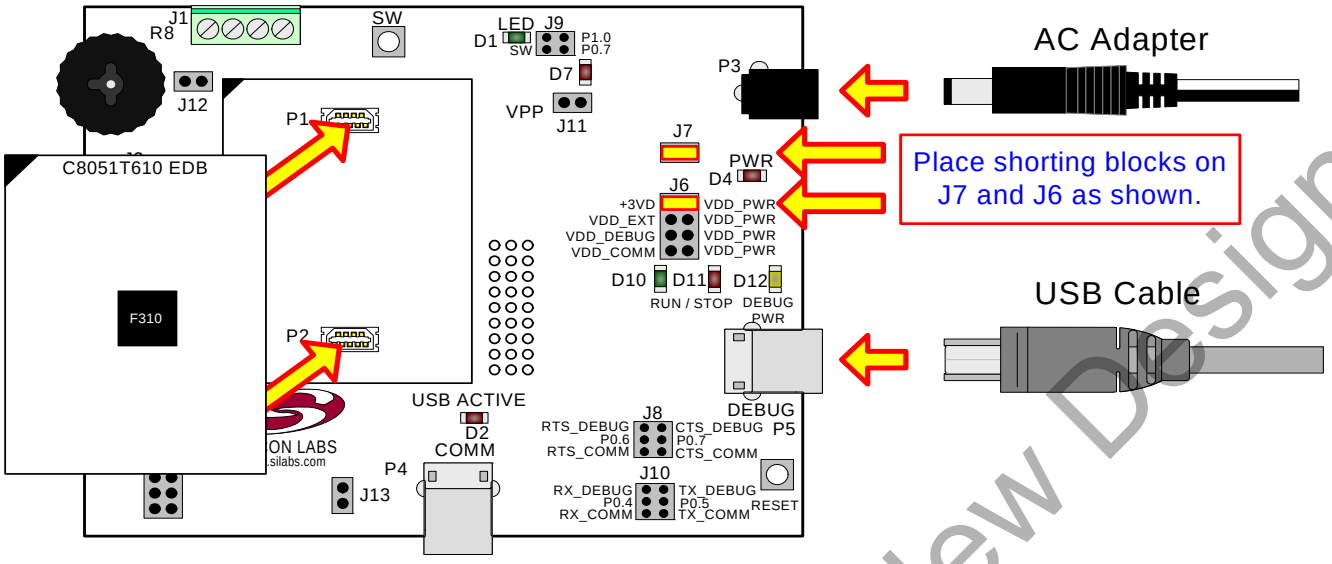

Figure 2. Hardware Setup (Emulation Daughter Board)

5. Software Setup

The included CD-ROM contains the Silicon Laboratories Integrated Development Environment (IDE), Keil software 8051 tools, the Virtual COM Port Drivers, 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 the CD contents.

5.1. Development Tools Installation

To install the IDE, utilities, and code examples, perform the following steps:

1. Click on the "Install Development Tools" button on the installation utility's startup screen.
2. In the Kit Selection box that appears, choose the C8051T610-DK development kit from the list of options.
3. In the next screen, choose “Components to be Installed”.
   Note: selecting the action that reads "Install CP210x Drivers" will launch a program described in “5.2. CP210x USB to UART VCP Driver Installation” .
4. Installers selected in Step 3 will execute in sequence, prompting the user as they install programs, documentation, and drivers.

5.2. CP210x USB to UART VCP Driver Installation

The C8051T610 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 (P2) on the C8051T610 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.

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 "AN104: Integrating Keil 8051 Tools into the Silicon Labs IDE" in the *SiLabs\MCU\Documentation\Appnotes* directory on the CDROM 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 it 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 them. Right-click on "New Project" in the Project Window. Select “Add Groups to Project”. Add predefined 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.

6.1.1. 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 by 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 "USB Debug Adapter" option. The USB Debug circuitry is integrated onto the C8051T610 main board.

Next, the correct debug interface must be selected. C8051T61x family devices and the C8051F310 all use the Silicon Labs "C2" 2-wire 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\C8051T610* directory during IDE installation. These files may be used as a template for code development.

7.1. Register Definition Files

Register definition files, *C8051T610.inc*, *C8051T610_defs.h*, and *compiler_defs.h*, define all SFR registers and bit addressable control/status bits. They are installed into the *SiLabs\MCU\Examples\C8051T610* directory during IDE installation. The register and bit names are identical to those used in the C8051T61x datasheet. The 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 C8051T61x 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, these programs flash the green LED on the C8051T610 main board about ten times a second using the interrupt handler with a timer.

8. Development Boards

The C8051T61x Development Kit includes a main board that interfaces to various daughter boards. The C8051T610 Emulation Daughter Board contains a C8051F310 device to be used for preliminary software development. The socketed C8051T610 daughter boards allow programming and evaluation of the actual C8051T61x family of devices. Numerous input/output (I/O) connections are provided on the main board to facilitate prototyping. Refer to Figure 2 for the locations of various connectors on the main board. Figure 3 shows the C8051T610 mother board and indicates locations for various I/O connectors. Figure 4 shows the factory default shorting block positions. Figures 5, 6, 7, and 8 show the available C8051T610 daughter boards.

P1, P2	Daughter board connection
P3	Power connector that accepts input from 7.5 to 15 Vdc unregulated power adapter
P4	USB connector for UART to USB communications interface
P5	USB Debug interface connector
J1	Analog I/O terminal block
J2	Port 0 header
J3	Port 1 header
J4	Port 2 header
J5	Port 3 header
J6	Power supply selection header (See “8.3. Power Supply Headers (J6 and J7)”)
J7	Power supply enable header that connects power source selected on J6 to the board's main power supply net
J8	Communications Interface Control Signal header
J9	Connects port pin P0.7 to the switch labeled "SW" and port pin P1.0 to the LED labeled "LED"

J10	Communications Interface Data Signal header
J11	VPP supply connection used when programming OTP devices
J12	Connects potentiometer to the port pin, P1.2, on the device
J13	Additional connections to ground

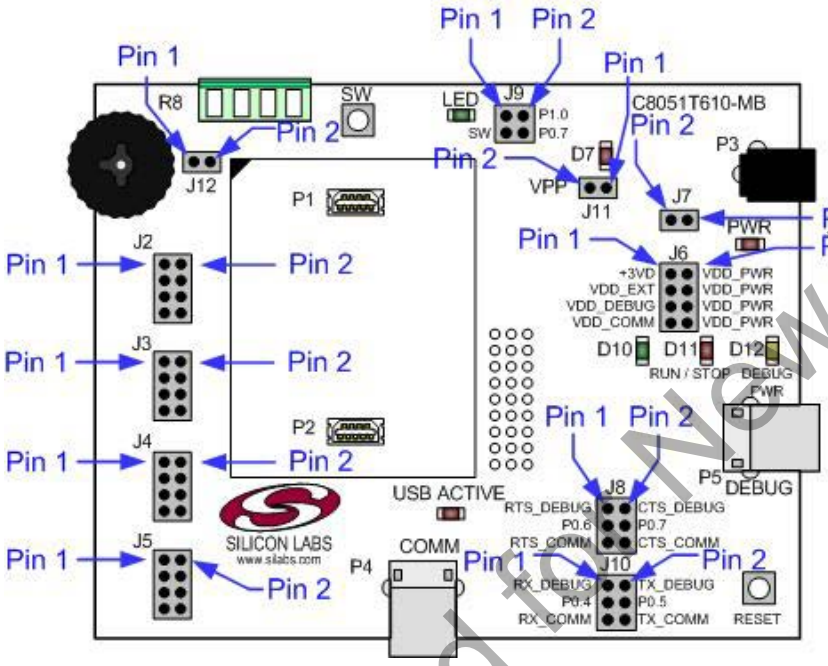

Figure 3. C8051T610 Main Board (Included in Kit)

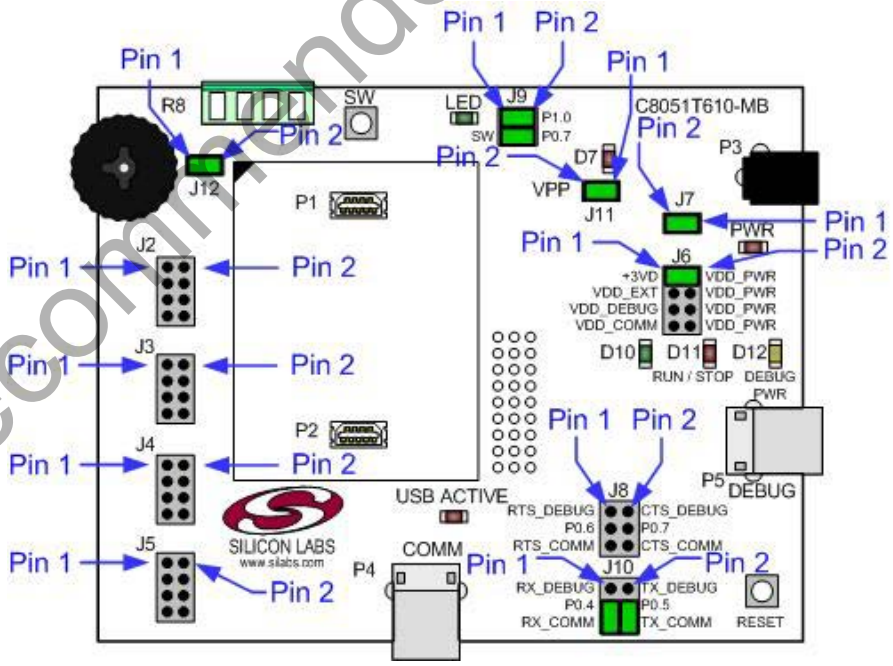

Figure 4. C8051T610 Mother Board Shorting Blocks (Factory Defaults)

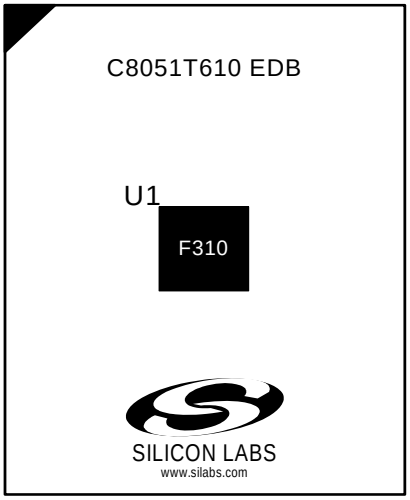

Figure 5. C8051T610 Evaluation Daughter Board (Included in Kit)

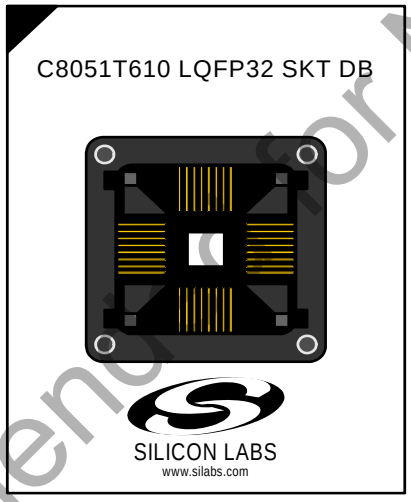

Figure 6. C8051T610 TQFP Socket Daughter Board (Included in Kit)

Figure 7. C8051T610 28-Pin QFN Socket Daughter Board (Available Separately)

Figure 8. C8051T610 24-Pin QFN Socket Daughter Board (Available Separately)

8.1. System Clock Sources

The C8051T61x devices feature a calibrated internal oscillator that is enabled as the system clock source on reset. After reset, the internal oscillator operates at a frequency of 24.5 MHz (±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 C8051T61x device at a frequency not available with the internal oscillator, an external oscillator source may be used. Refer to the C8051T61x datasheet for more information on configuring the system clock source.

8.2. Switches, LEDs and Potentiometer

Two switches are provided on the main board. The RESET switch is connected to the RST pin of the C8051T61x. Pressing RESET puts the device into its hardware-reset state. The switch labeled “SW” can be connected to the C8051T61x's general purpose I/O (GPIO) pin P1.0 through header J9. Pressing this switch generates a logic low signal on the port pin. Remove the shorting block from the J9 header to disconnect the switch from the port pin.

Six LEDs are also provided on the main board. The red LED, labeled “PWR” (D4), is used to indicate a power connection to the main board. The green LED labeled “D10” is the "run" light for the debug circuitry; the red LED, labeled “D11”, is the "stop" light for the debug circuitry, and the orange LED, labeled “D12”, indicates whether the debug adapter is being powered through P5's USB connector. The red LED, labeled “D7”, indicates when the VPP programming voltage is being applied to the device. The green LED, labeled "LED" (D1), can be connected to the C8051T61x's GPIO pin through header J9. Remove the shorting block from the header to disconnect the LED from the port pin. The "USB ACTIVE" LED, labeled “D2”, will turn on whenever the CP2103 USB-to-UART bridge is connected to a PC and has successfully completed enumeration.

Also included on the C8051T61x main board is a 10 kΩ thumbwheel rotary potentiometer, reference number R8. The potentiometer can be connected to the C8051T61x's P1.2 pin through the J12 header. Remove the shorting block from the header to disconnect the potentiometer from the port pin.

Table 1 lists the port pins and headers corresponding to the switches, LEDs, and potentiometer.

Table 1. Main Board I/O Descriptions

Description	Component Name	I/O	Header
SW	SW1	Daughter Card's P0.7	J9 [3-4]
RESET	SW2	Daughter Card's RST/C2CK	None
Green LED labeled "LED"	D1	Daughter Card's P1.0	J9 [1-2]
Green LED labeled "USB Active"	D2	U2 CP2103's SUSPEND	None
Red LED labeled "PWR"	D4	Daughter Card's VDD	J6, J7
Red LED connected to VPP	D7	Daughter Card's P0.2	J11
Green LED labeled "RUN"	D10	Debug Adapter Signal	None
Red LED labeled "STOP"	D11	Debug Adapter Signal	None
Orange LED labeled "DEBUG PWR"	D12	Debug Adapter Signal	None
Potentiometer	R8	Daughte Card's P1.5	J12

8.3. Power Supply Headers (J6 and J7)

The main power supply of the main board, which is used to power the daughter board, can be provided by either the USB Debug Adapter's on-chip voltage regulator, the CP2103 USB-to-UART bridge's on-chip voltage regulator, P3 and its associated circuitry, or an external voltage applied to the VDD_EXT connection on J1. To select a power supply , place a shorting block on J6 across the appropriate pin pair, as shown in Figures 7 thru 10. To connect the main power supply to an attached daughter board, place a shorting block across J7.

Notes:

1. One and only one shorting block should be placed on J6 at a time.
2. In order to use the CP2103's voltage regulator as the board's power supply, a USB cable must be connected to P4, and the USB Active LED (D2) must be on.
3. In order to use the USB Debug Adapter's voltage regulator as the board's power supply, a USB cable must be connected to P5, and the DEBUG PWR LED (D12) must be on.

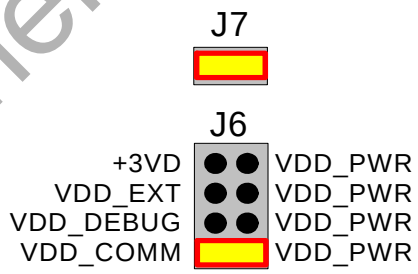

Figure 9. J6 and J7 Shorting Block Configuration for Power Using the CP2103 and P4

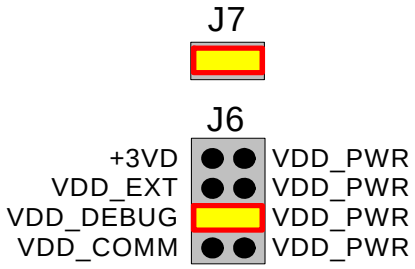

Figure 10. J6 and J7 Shorting Block Configuration for Power Using USB Debug Adapter Power and P5

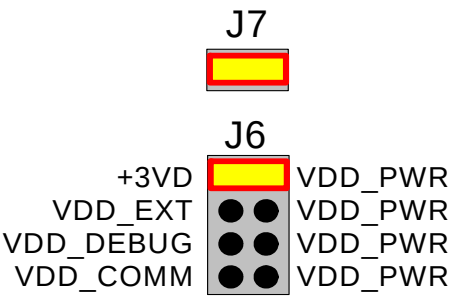

Figure 11. J6 and J7 Shorting Block Configuration for Power Using 3.3 V Regulator and P3

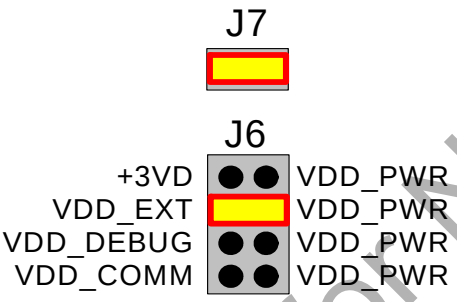

Figure 12. J6 and J7 Shorting Block Configuration for External Power Using J1

8.4. USB Debug Adapter (DEBUG / P5)

A Universal Serial Bus (USB) connector (P5) provides the onboard debug and programming interface. The debug/programming MCU and associated circuitry are powered through the USB connector, which can also supply the rest of the main board by routing the USB Debug Adapter's power through J6. The USB Debug Adapter also provides a data communications interface that can be used when the debug adapter is not debugging or programming a C8051T61x device.

8.5. UART to USB Communications Interfaces

The C8051F610 main board provides UART to USB communications interfaces through both the CP2103 USB-to-UART bridge device and the communications interface of the USB Debug Adapter.

The CP2103 bridge device connects to a PC through the USB connector labeled “P4”. This USB connector supplies power to the CP2103 and can supply power to the rest of the main board by configuring J6 and J7 as shown in Figure 9. To use the CP2103 as a communications interface, the CP2103 virtual COM port drivers must be installed on a PC.

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

8.6. Communications Interface Selector Headers (J10 and J8)

The C8051T610 main board routes the C8051T61x's P0.4 (UART TX) and P0.5 (UART RX) to J10, where those signals can be connected to either the CP2103 USB-To-UART bridge or the USB Debug Adapter. The main board also allows the C8051T61x's P0.6 and P0.7 to be used as the UART control signals, RTS and CTS. These two signals are routed to J8, where they can be connected to either the CP2103 or the USB Debug Adapter.

To connect the C8051T61x's UART to the CP2103, shorting blocks on J10 and J8 should be placed in the positions shown in Figure 13. To connect the microcontroller's UART to the USB Debug Adapter, shorting blocks on J10 and J8 should be placed as shown in Figure 12.

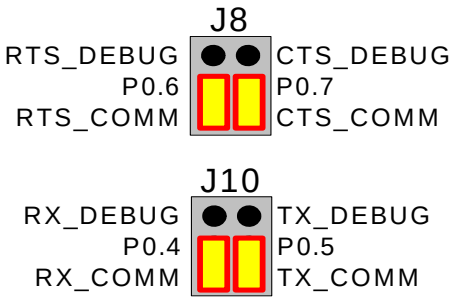

Figure 13. Shorting Block Configuration when Using the CP2103

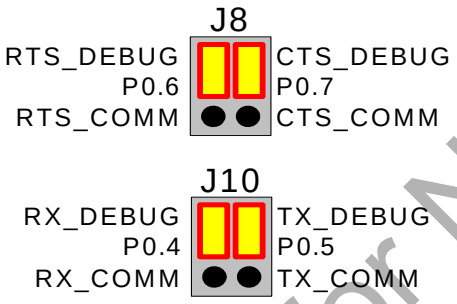

Figure 14. Shorting Block Configuration when Using the USB Debug Adapter

8.7. PORT I/O Connector (J2, J3, J4, and J5)

Each of the C8051T61x's I/O pins, as well as +3VD and GND, are routed to headers J2 thru J5. J2 connects to the microcontroller's Port 0 pins. J3 connects to Port 1; J4 connects to J4, and J5 connects to Port 3.

8.8. Analog I/O (J1)

Two of the C8051T61x target device's port pins are connected to the J1 terminal block. The terminal block also allows users to input an external voltage that can be used as the power supply of the board. Refer to Table 2 for the J1 terminal block connections.

Table 2. J1 Terminal Block Descriptions

Pin #	Description
1	GND (Ground)
2	P1.2
3	P0.0 (VREF)
4	VDD_EXT (routed to header J6)

8.9. VPP Connection (J10)

The C8051T61x devices require a special 6.5 V programming voltage applied to the VPP pin during device programming. The VPP pin on these devices is shared with P0.2. During programming, the VPP voltage is automatically enabled when needed. Header J11 is provided to allow the user to disconnect the programming circuitry from the VPP/P0.2 pin to avoid interfering with the normal application operation of P0.2. When programming the device, J11 should be shorted with a shorting block. When running normal application code, J11 can be removed. Note that the C8051T610 Emulation Daughter Board does not connect the main board's VPP and P0.2 signals; so, removing the shorting block is not necessary when using the Emulation Daughter Board.

8.10. Using Alternate Supplies with the C8051T61x Development Kit

For most evaluation purposes, the onboard 3.3 V supply regulator is sufficient to be used as a VDD power supply. However, in applications where a different supply voltage is desired (for example, 1.8 V), an external supply voltage can be applied to the board at the analog connector (J1).

1. When programming a C8051T61x device, the onboard 3.3 V regulator should be used for VDD.
2. If a VDD supply voltage between 2.7 and 3.6 V is desired, the shorting block on J6 should be placed so that the pin 3 (VDD_EXT) is shorted to pin 4 (VDD_PWR), and the desired supply voltage can be applied directly to the GND and VDD terminals of terminal block J6. No board modification is necessary.

9. Schematics

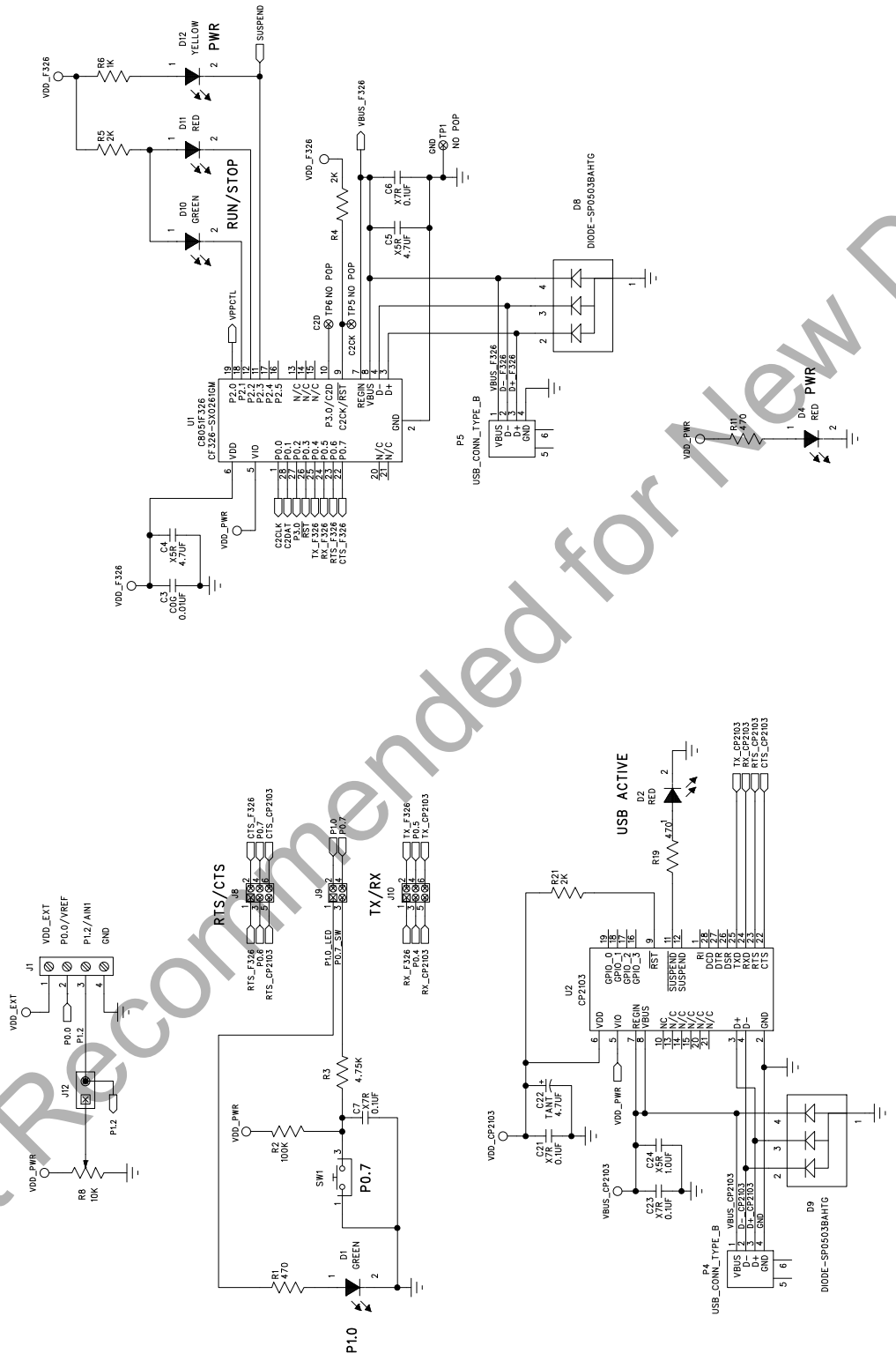

Figure 15. C8051T610 Main Board Schematic (1 of 2)

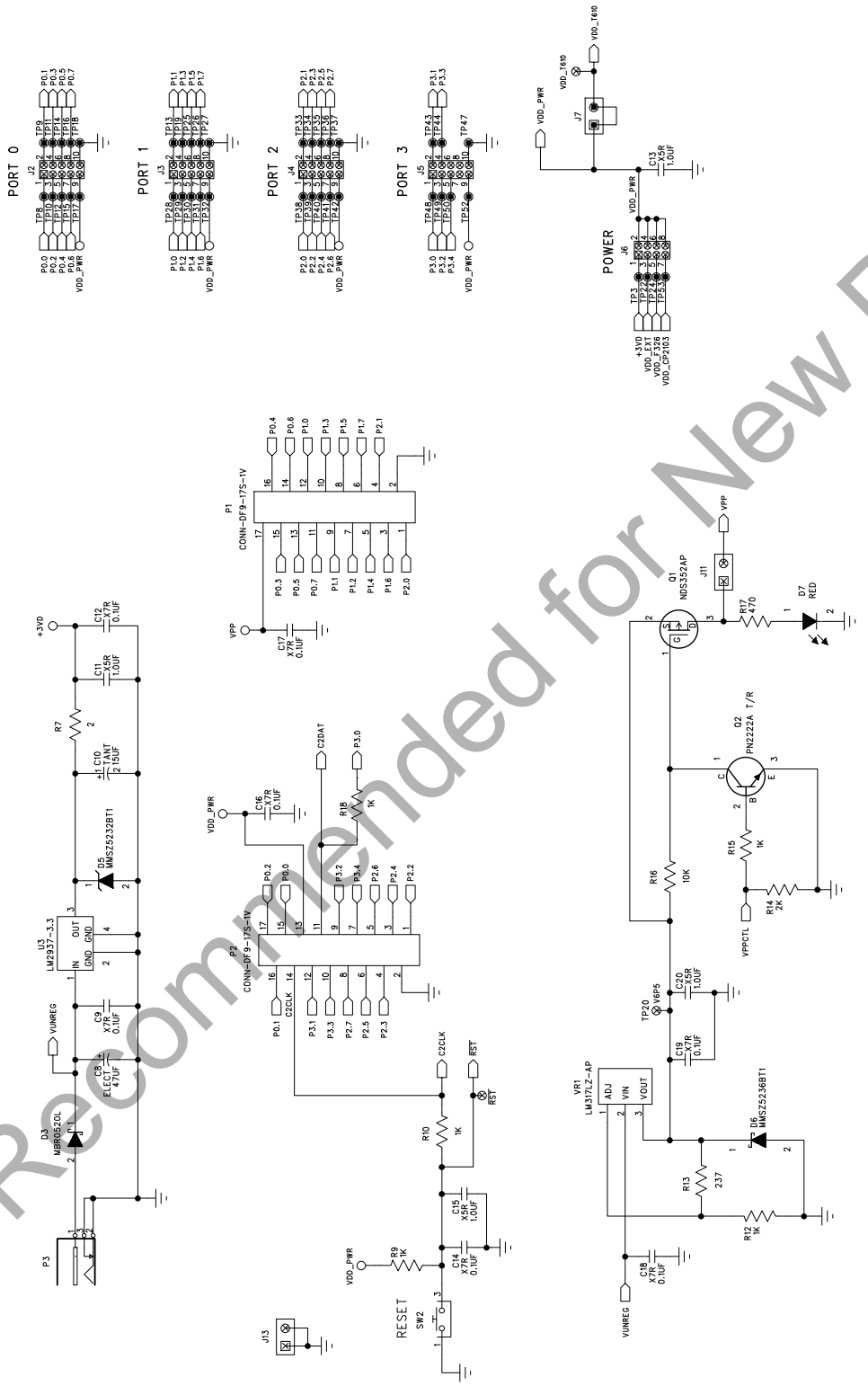

Figure 16. C8051T610 Main Board Schematic (2 of 2)

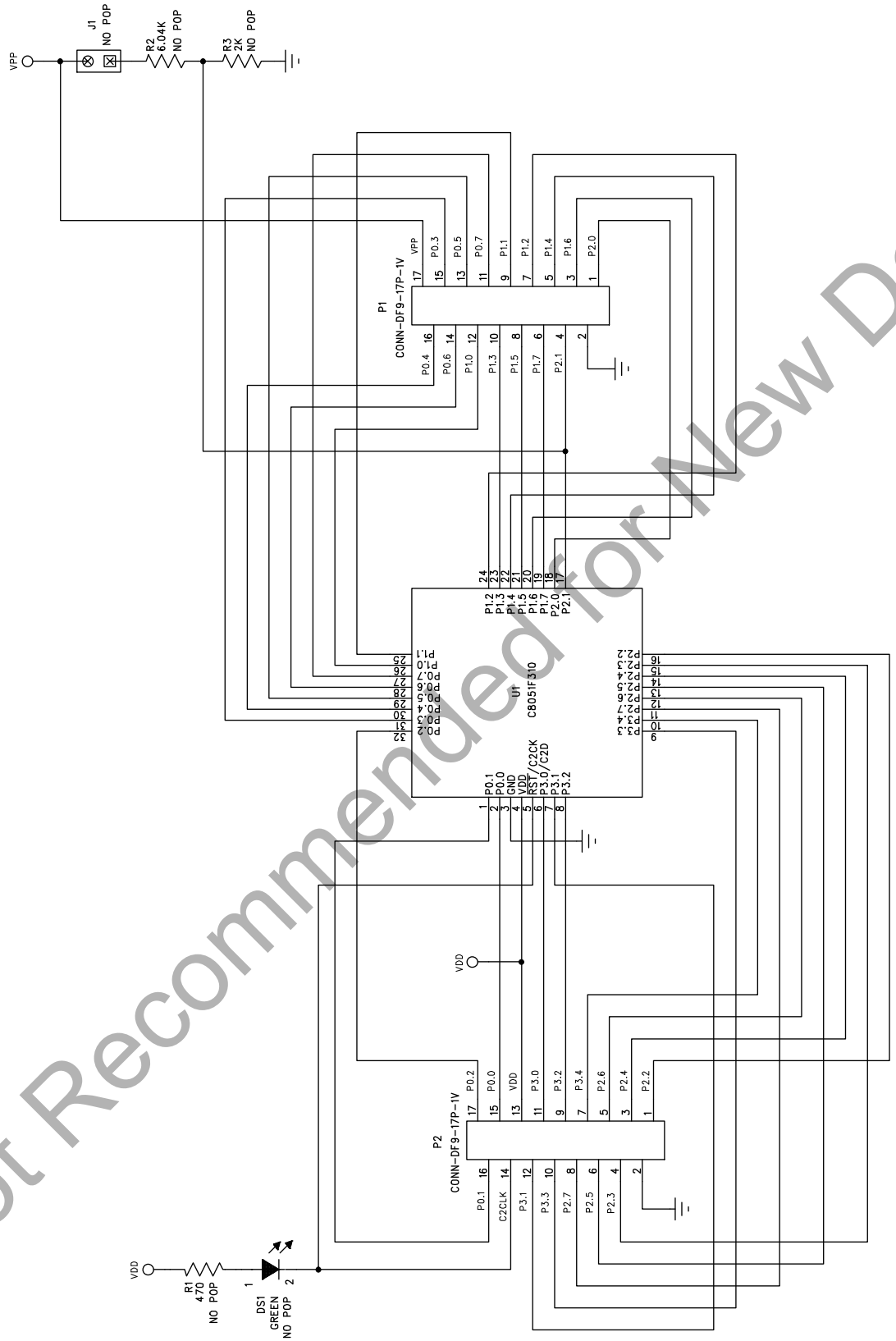

Figure 17. C8051T610 Emulation Daughter Board Schematic

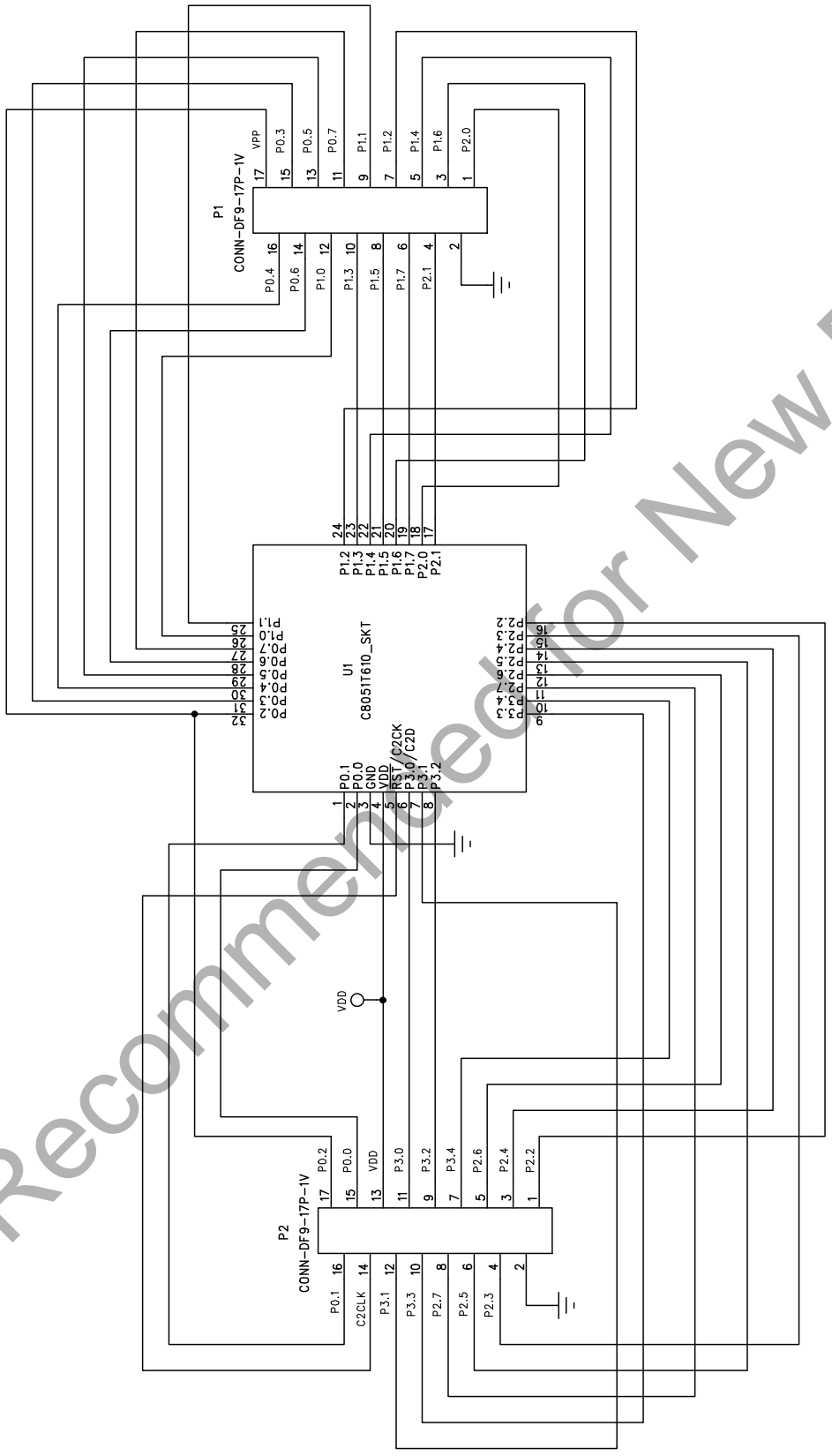

Figure 18. C8051T610 TQFP Socket Daughter Board Schematic

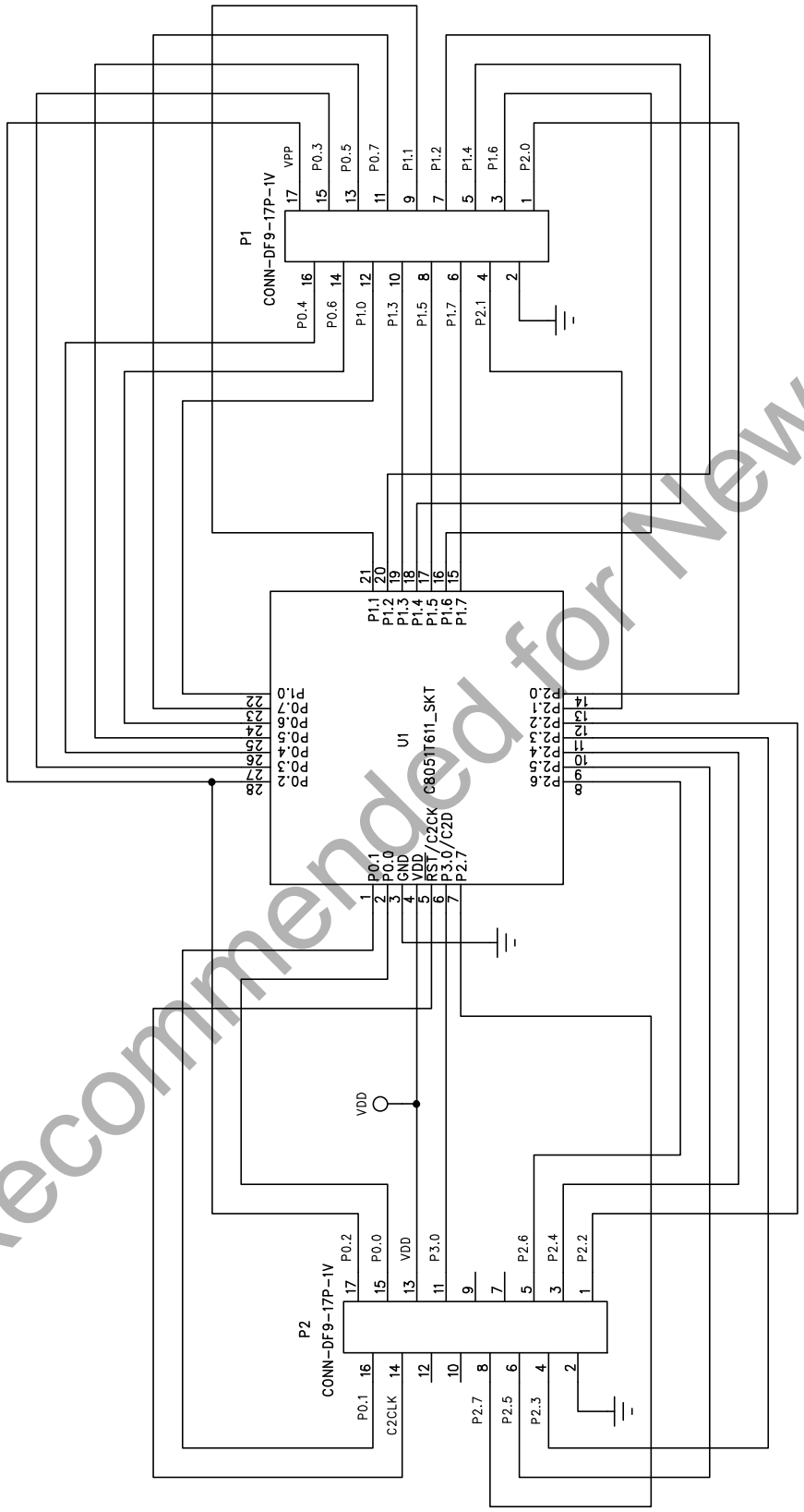

Figure 19. C8051T610 QFN-28 Socket Daughter Board Schematic

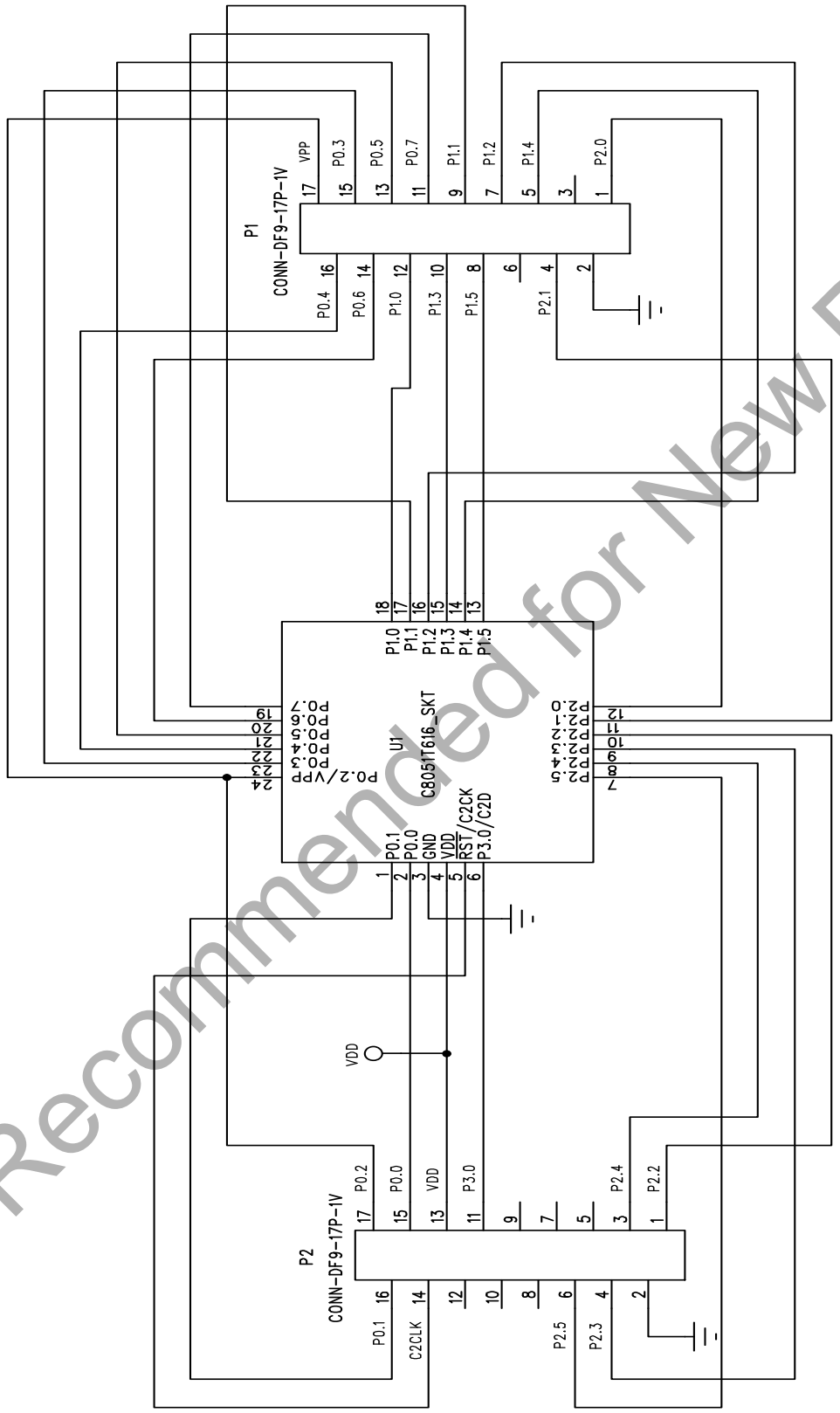

Figure 20. C8051T610 QFN-24 Daughter Board Schematic