C8051T620/2 Development Kit User's Guide

1. Kit Contents

The C8051T620 and C8051T622 Development Kits contain the following items:

- C8051T62x Motherboard
- C8051T62x Emulation Daughter Board with C8051F34A installed
- Socket Daughter Board (one of the following):
  - C8051T620DK: C8051T62x QFN 32-pin
  - C8051T622DK: C8051T622 QFN 24-pin
- Device samples (one of the following):
  - C8051T620DK: C8051T620-GM (Qty: 10) and C8051T626-B-GM (Qty: 10)
  - C8051T622DK: C8051T622-GM (Qty: 10)
- C8051Txxx Development Kit Quick-Start Guide
  - Quick Start Guide includes steps to download the following development software and documentation (these instructions can also be found in this document. See "4. Software Installation" on page 3.):
    - Silicon Laboratories Integrated Development Environment (IDE)
    - Evaluation version of 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

2. About the Daughter Boards

The C8051T620 and C8051T622 Development Kits include an Emulation Daughter Board (EDB) and a QFN Socket Daughter Board (QFN-DB). The EDB has an installed C8051F34A device, which is a Flash-based device that can be used for the majority of C8051T62x/32x code development. The QFN-DB is intended to allow both programming and system-level debugging of C8051T62x/32x devices directly.

A C8051T62x/32x device cannot be erased once it has been programmed; so, it is advisable to use the C8051F34A for the majority of code development. Refer to "AN368: Differences between the C8051F34A and the C8051T62x and C8051T32x Device Families" for more details on how the C8051F34A can be used to develop code for the C8051T62x/32x device families. Application notes can be found on the Silicon Labs web site: www.silabs.com/appnotes

3. Hardware Setup

See Figure 1 for a diagram of the hardware configuration.

1. Attach the desired daughter board to the motherboard at connectors P1 and P2.
2. If using the QFN Socket Daughter Board, 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 motherboard'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 motherboard.

Notes:

1. Use the Reset icon in the IDE to reset the target when connected during a debug session.
2. Remove power from the motherboard and remove the USB cable before removing a daughter board from the motherboard. Connecting or disconnecting a daughter board when the power adapter or USB cable are connected can damage the motherboard, the daughter board, or the socketed device.
3. Remove power from the motherboard and remove the USB cable before removing a C8051T62x/32x device from the socket. Inserting or removing a device from the socket when the power adapter or USB cable are connected can damage the motherboard, the daughter board, or the socketed device.
4. 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. Power Supply Headers (J6 and J7)" on page 15.

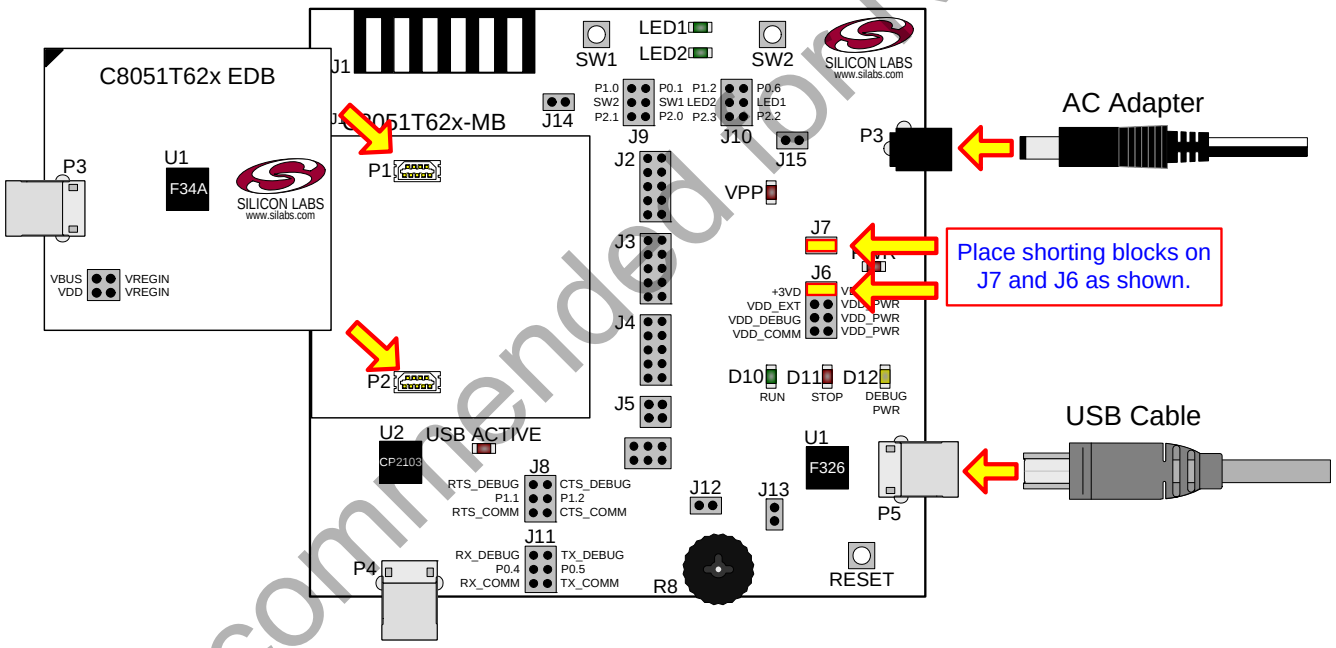


Figure 1. Hardware Setup (Emulation Daughter Board)

4. Software Installation

All the latest software tools and documentation can be downloaded from www.silabs.com/mcudownloads.

The Silicon Labs 8-bit Development Tools download package available on that web page includes the Silicon Laboratories Integrated Development Environment (IDE), 8051 evaluation toolset, Virtual COM Port drivers for the CP210x USB to UART Bridge, and additional documentation. Download the package and install the components by following the on-screen instructions.

4.1. System Requirements

The following are the system requirements necessary to run the debug and programming tools:

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

4.2. Development Tools Installation

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

1. Run the Silicon Labs 8-bit Development Tools installer downloaded from www.silabs.com/mcudownloads.
2. In the screen titled “Select Installation Packages”, the Silicon Labs IDE, 8051 Evaluation Toolset and CP210x Virtual COM Port drivers will be checked by default. The programs necessary to download and debug on the MCU are the Silicon Labs IDE and the 8051 Evaluation Toolset. The CP210x Drivers are necessary to use the UART capabilities of the target board. See “4.3. CP210x USB to UART VCP Driver Installation” for more information about installing the CP210x drivers. See “5. Software Overview” for an overview of all applicable software included in the web download.
3. Installers selected in Step 3 will execute in sequence, prompting the user as they install programs, documentation, and drivers.

4.3. CP210x USB to UART VCP Driver Installation

The C8051T62x Motherboard includes a Silicon Laboratories CP2103 USB-to-UART Bridge Controller. Device drivers for the CP2103 need to be installed before PC software, such as HyperTerminal, can communicate with the 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 of this installer will provide an option to install the driver on the target system. Select the “Launch the CP210x VCP Driver Installer” option in that window if you are ready to install the driver. You do not need to have the motherboard's USB connector plugged into the PC's USB port for this step.
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*). At this location, run *CP210xVCPInstaller.exe*.
5. To complete the installation process, connect the included USB cable between the host computer and the COMM USB connector (P4) on the C8051T62x Motherboard. 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 the “Silicon Laboratories CP210x USB to UART Bridge (Driver Removal)” option in the “Add or Remove Programs” window.

5. Software Overview

The following software is necessary to build a project, download code to, and communicate with the target microcontroller.

- 8051 Evaluation Toolset
- Silicon Labs Integrated Development Environment (IDE)

Other useful software included in the Silicon Labs 8-bit Development Tools package from the Silicon Labs Downloads web site (www.silabs.com/mcudownloads) includes:

- Configuration Wizard 2
- Keil µVision2, µVision3, and µVision4 Drivers
- MCU Production Programmer and Flash Programming Utilities

5.1. 8051 Evaluation Toolset

The Silicon Labs IDE has native support for many third-party 8051 toolsets. Included with this kit is an 8051 evaluation assembler, compiler, and linker. For further information on the tools, including limitations, see the corresponding application note. Application notes can be found on the Silicon Labs web site (http://www.silabs.com/appnotes). See Table 1 for a list of supported toolsets and associated application notes.

Table 1. Supported Third Party 8051 Toolsets

Toolset	Application Note
Keil	“AN104: Integrating Keil 8051 Tools into the Silicon Labs IDE”
Raisonance	“AN125: Integrating Raisonance 8051 Tools into the Silicon Labs IDE”
Tasking	“AN126: Integrating Tasking 8051 Tools into the Silicon Labs IDE”
HI-TECH	“AN140: Integrating Hi-TECH 8051 Tools into the Silicon Labs IDE”
SDCC	“AN198: Integrating SDCC 8051 Tools into the Silicon Labs IDE”
IAR	“AN236: Integrating IAR 8051 Tools into the Silicon Labs IDE”

5.2. Silicon Labs IDE

The Silicon Labs IDE integrates a source-code editor, source-level debugger, and in-system programmer. The following sections discuss how to open an example project in the IDE, build the source code, and download it to the target device.

5.2.1. Running the T620_Blinky or T622_Blinky example program

The T620_Blinky or T622_Blinky example program blinks an LED on the target board.

1. Open the Silicon Labs IDE from the Start menu.
2. Select **Project→Open Project** to open an existing project.
3. Browse to the *C:\SiLabs\MCU\Examples\C8051T620_1_6_7_T320_3\Blinky or SiLabs\MCU\Examples\C8051T622_3_T326_7\Blinky* directory (default) and select the T620_Blinky_C.wsp or T622_Blinky_C.wsp project file. Click **Open**.
4. Once the project is open, 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.
5. 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. Next, the correct “Debug Interface” must be selected. C8051T62x/32x devices use Silicon Labs “C2” 2-wire debug interface. Once all the selections are made, click the **OK** button to close the window.

6. Click the **Connect** button in the toolbar or select **Debug→Connect** from the menu to connect to the device.
7. 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.
8. Click on the **Go** button (green circle) in the toolbar or by selecting **Debug→Go** from the menu to start running the firmware. The LED on the target board will start blinking.

5.2.2. Creating a New Project

Use the following steps to create a new project. Once steps 1–5 in this section are complete, continue with Step 3 from Section 5.2.1.

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 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.
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**.

5.3. Configuration Wizard 2

Configuration Wizard 2 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 2.

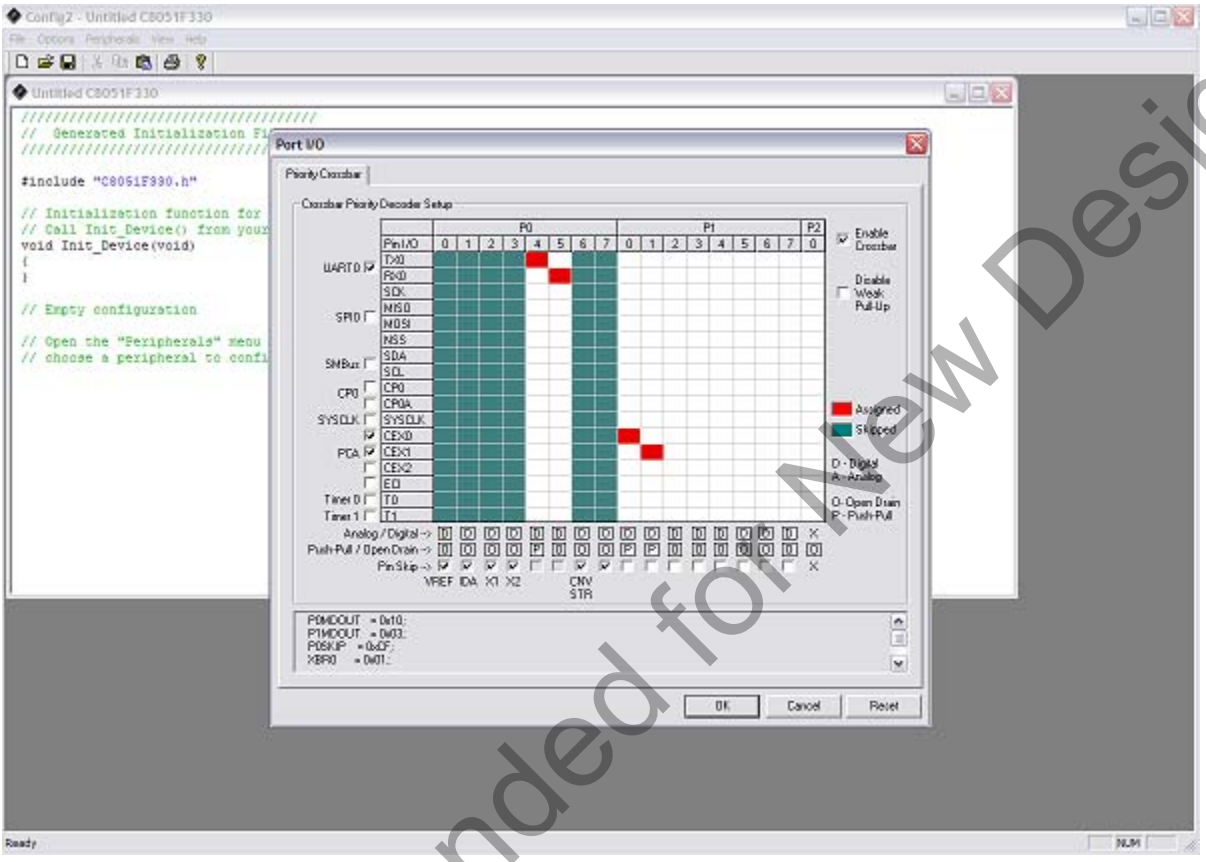


Figure 2. Configuration Wizard 2 Utility

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 2 help available under the Help menu in Configuration Wizard 2 or refer to the Configuration Wizard 2 documentation. Documentation and software are available from the downloads webpage: www.silabs.com/mcudownloads.

5.4. Keil uVision2, uVision3, and uVision4 Silicon Laboratories Drivers

As an alternative to the Silicon Laboratories IDE, the µVision debug driver allows the Keil µVision2, µVision3, and µVision4 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 µVision2, µVision3, and µVision4 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 µVision driver documentation. The documentation and software are available from the downloads webpage: www.silabs.com/mcudownloads.

5.5. Programming Utilities

The Silicon Labs IDE is the primary tool for downloading firmware to the MCU during development. There are two software programming tools that are intended for use during prototyping or in the field: the MCU Production Programmer and the Flash Programming Utilities. The MCU Production Programmer is installed with the IDE to the directory, C:\Silabs\MCU\Utilities\Production Programmer\ (default). The Flash Programming Utilities can be optionally installed from the web installer, and are installed to C:\Silabs\MCU\Utilities\FLASH Programming\ (default).

5.6. 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 downloads webpage: www.silabs.com/mcudownloads.

6. Example Source Code

Example source code and register definition files are provided by default in the *SiLabs\MCU\Examples\C8051T620_1_6_7_T320_3 or SiLabs\MCU\Examples\C8051T622_3_T326_7* directory during IDE installation. These files may be used as a template for code development.

6.1. Register Definition Files

Register definition files *C8051T620.inc*, *C8051T622.inc*, *C8051T620_defs.h*, *C8051T622_defs.h*, and *compiler_defs.h* define all SFR registers and bit-addressable control/status bits. They are installed by default into the *SiLabs\MCU\Examples\C8051T620_1_6_7_T320_3 or SiLabs\MCU\Examples\C8051T622_3_T326_7* directory during IDE installation. The register and bit names are identical to those used in the C8051T620-21-26-27_T320-3 or C8051T622-23_T326-27 data sheet.

6.2. Blinking LED Example

The example source files T620_Blinky.asm and T620_Blinky.c or T622_Blinky.asm and T622_Blinky.c show examples of several basic C8051T62x 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 C8051T62x Motherboard about five times a second using the interrupt handler with a timer.

7. Development Boards

The C8051T620/2 Development Kit includes a motherboard that interfaces to various daughter boards. The C8051T62x Emulation Daughter Board contains a C8051F34A device to be used for preliminary software development. The C8051T620 Socket Daughter Board and C8051T622 Socket Daughter Board allow programming and evaluation of the actual C8051T62x devices. Numerous input/output (I/O) connections are provided on the motherboard to facilitate prototyping. Figure 3 shows the C8051T62x Motherboard and indicates locations for various I/O connectors. Figure 4 shows the factory default shorting block positions. Figures 5, 6, and 7 show the available C8051T62x daughter boards. Figures 8, 9, 10, and 11 show the available C8051T32x daughter boards.

- P1, P2 Daughter board connection
- P3 Power connector that accepts input from 7.5 to 15 V dc 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 with access to VDD and GND
- J6 Power supply selection header (See "7.3. Power Supply Headers (J6 and J7)" on page 15)
- 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 pins to the switches labeled “SW1” and “SW2”
- J10 Connects port pins to the LEDs labeled “LED1” and “LED2”
- J11 Communications interface data signal header
- J12 Connects potentiometer to the port pin, P2.5
- J13 Additional connections to ground
- J14 Connects an external VREF from J1 to P0.7
- J15 VPP supply connection used when programming EPROM devices

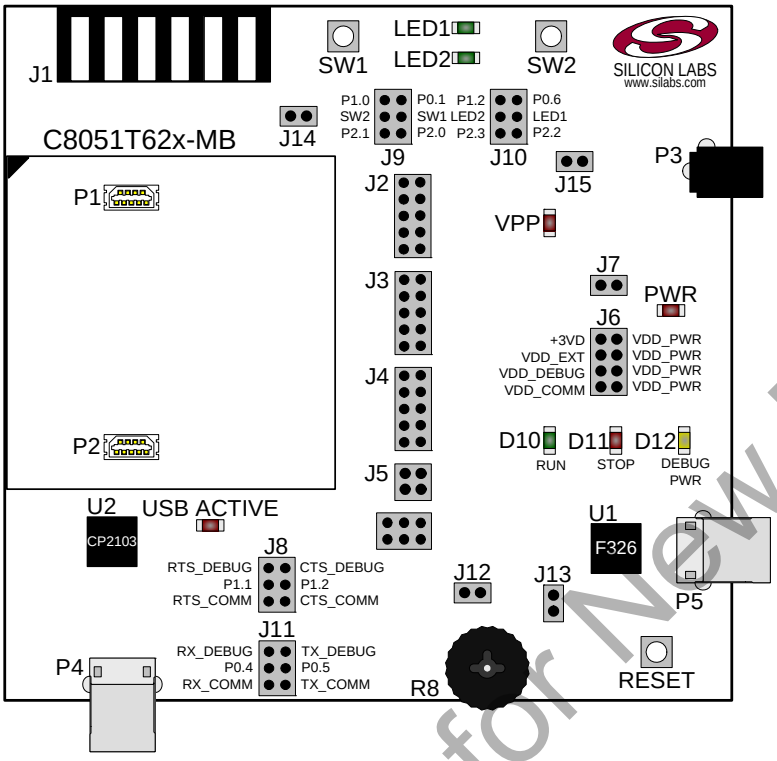


Figure 3. C8051T62x Motherboard

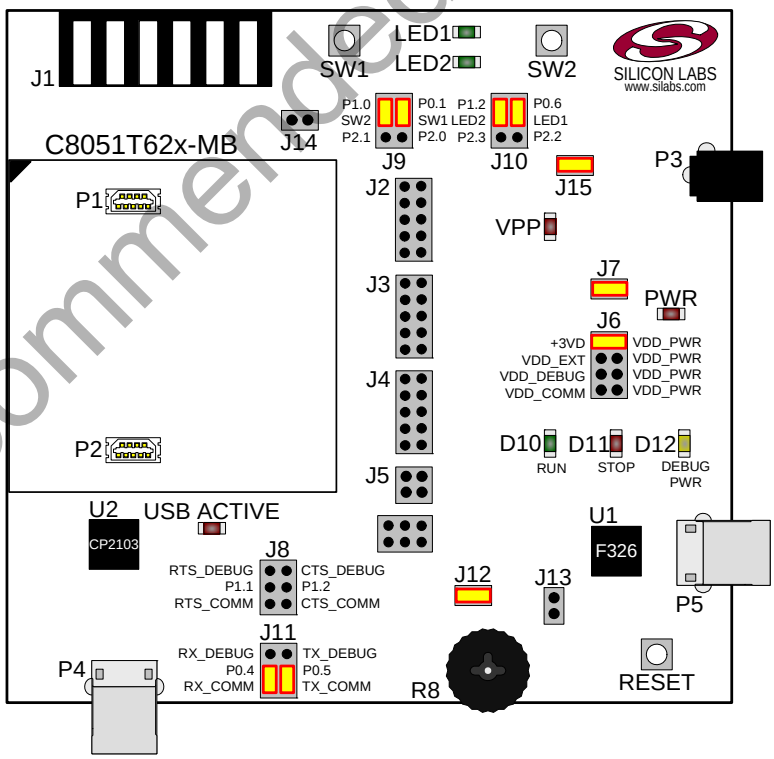


Figure 4. C8051T62x Motherboard Default Shorting Block Positions

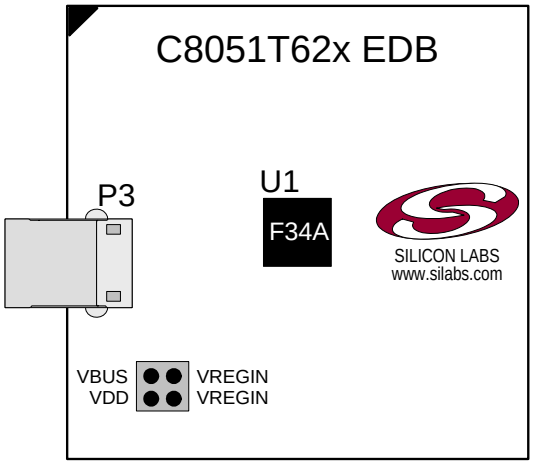


Figure 5. C8051T62x Emulation Daughter Board

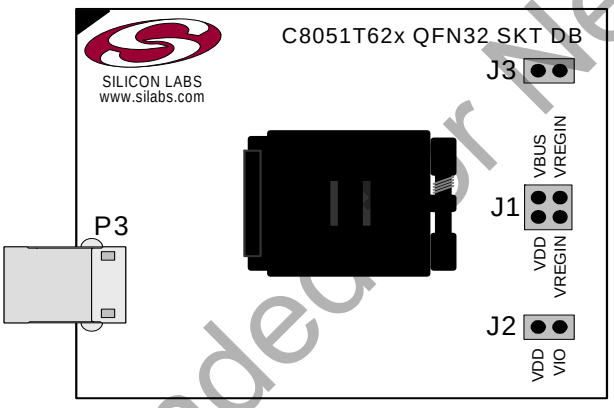


Figure 6. C8051T620 QFN32 Socket Daughter Board

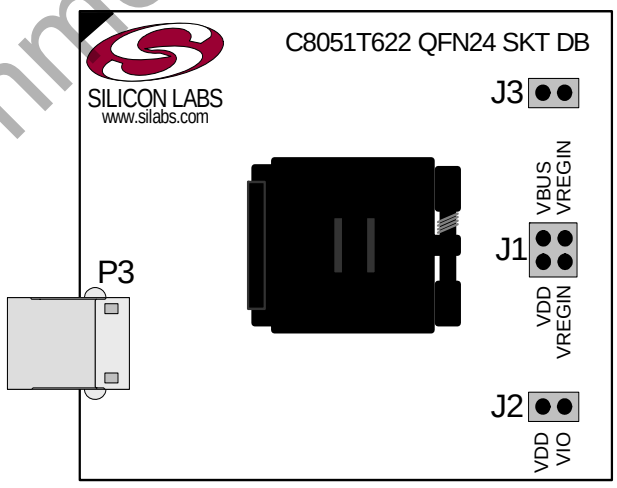


Figure 7. C8051T622 QFN24 Socket Daughter Board

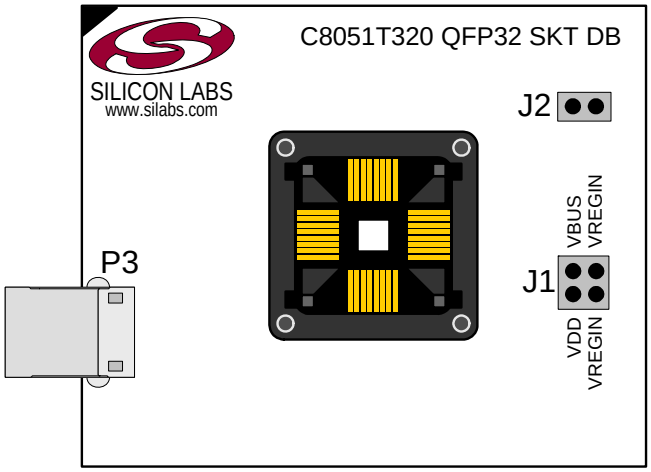


Figure 8. C8051T320 QFP32 Socket Daughter Board

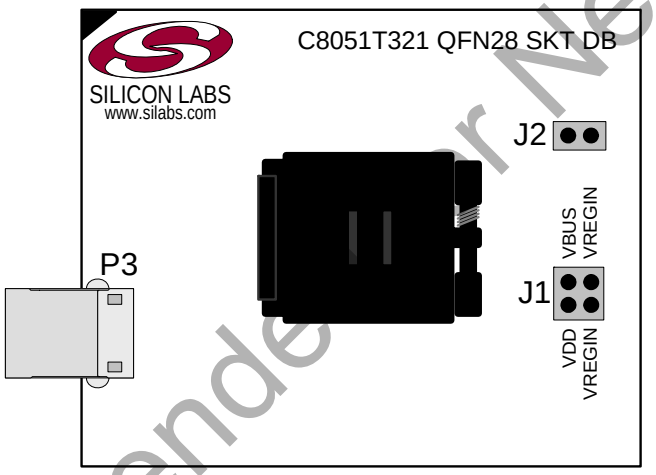


Figure 9. C8051T321 QFN28 Socket Daughter Board

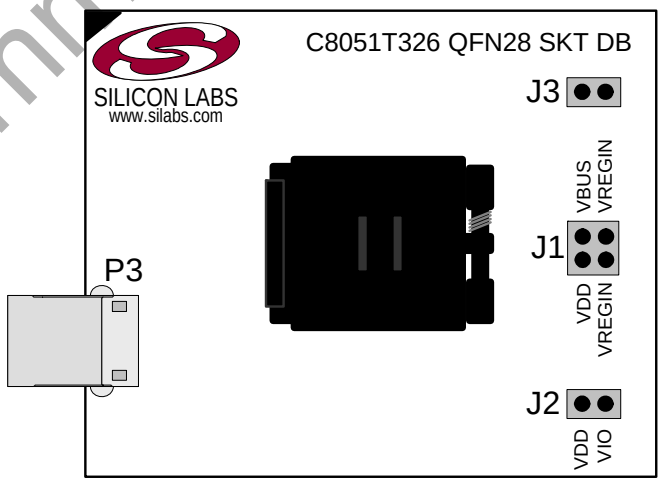


Figure 10. C8051T326 QFN28 Socket Daughter Board

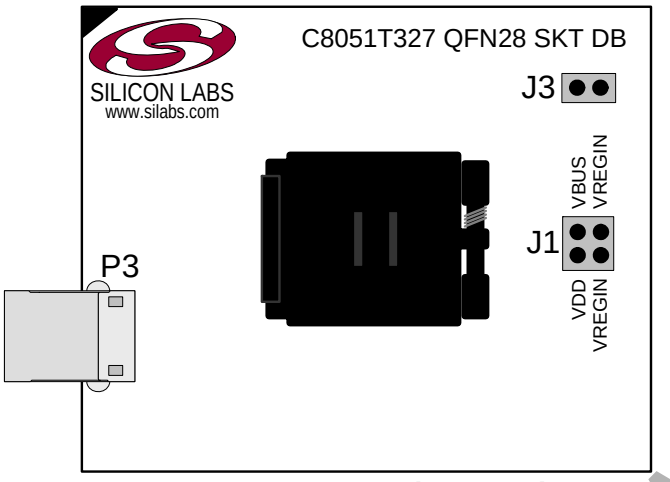


Figure 11. C8051T327 QFN28 Socket Daughter Board

7.1. System Clock Sources

The C8051T62x/32x 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 48 MHz (±1.5%) 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 C8051T62x/32x device at a frequency not available with the internal oscillator, an external oscillator source may be used. Refer to the C8051T620-21-26-27_T320-3 or C8051T622-23_T326-27 data sheet for more information on configuring the system clock source.

7.2. Switches, LEDs, and Potentiometer (J9, J10, and J12)

Three switches are provided on the motherboard. The RESET switch is connected to the RST pin of the C8051T62x/32x. Pressing RESET puts the device into its hardware-reset state. The switch labeled “SW1” can be connected to the C8051T62x/32x's general-purpose I/O (GPIO) pins P0.1 and P2.0, and “SW2” can be connected to the C8051T62x/32x's general-purpose I/O (GPIO) pins P1.0 and P2.1 through header J9. Pressing a switch generates a logic low signal on the port pin. Remove its shorting block from the J9 header to disconnect the switch from the port pin.

Seven LEDs are also provided on the motherboard. The red LED labeled “PWR” (D4) is used to indicate a power connection to the motherboard. The green LED labeled “RUN” (D10) turns on when the debug circuitry is in a running state; the red LED labeled “STOP” (D11) turns on when the debug circuitry is in a halted state, and the orange LED labeled “DEBUG PWR” (D12) indicates whether the debug adapter circuit is being powered through P5's USB connector. The red LED labeled “VPP” (D7) indicates when the VPP programming voltage is being applied to the device. The green LEDs, labeled “LED1” (D1) and “LED2” (D2), can be connected to C8051T62x/32x's GPIO pins through header J10. Remove its shorting block from the header to disconnect an LED from the port pin. The red LED labeled “USB ACTIVE” (D13) will turn on whenever the CP2103 USB-to-UART bridge is connected to a PC and has successfully completed enumeration.

Also included on the C8051T62x Motherboard is a 10 kΩ thumbwheel rotary potentiometer, reference number R8. The potentiometer can be connected to the C8051T62x/32x's P2.5 pin through the J12 header. Remove the shorting block from the header to disconnect the potentiometer from the port pin.

Table 2 lists the port pins and headers corresponding to the switches, LEDs, and potentiometer.

Table 2. Motherboard I/O Descriptions

Description	Component Name	I/O	Header
Switch	SW1	Daughter Card's P0.1 Daughter Card's P2.0	J9 [2-4] J9 [4-6]
Switch	SW2	Daughter Card's P1.0 Daughter Card's P2.1	J9 [1-3] J9 [3-5]
RESET	SW3	Daughter Card's RST/C2CK	None
Green LED labeled "LED1"	D1	Daughter Card's P0.6 Daughter Card's P2.2	J10 [2-4] J10 [4-6]
Green LED labeled "LED2"	D2	Daughter Card's P1.2 Daughter Card's P2.3	J10 [1-3] J10 [3-5]
Red LED labeled "PWR"	D4	Daughter Card's VDD	J6, J7
Red LED labeled "VPP"	D7	Daughter Card's VPP pin (See "VPP Pin Sharing" on page 17)	J15
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
Green LED labeled "USB ACTIVE"	D13	U2 CP2103's SUSPEND	None
Potentiometer	R8	Daughter Card's P2.5	J12

7.3. Power Supply Headers (J6 and J7)

The main power supply of the motherboard, 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 Figure 12. To connect the main power supply to an attached daughter board, place a shorting block across J7.

Notes:

1. Only one shorting block should be placed on J6 at a time.
2. 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. 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.

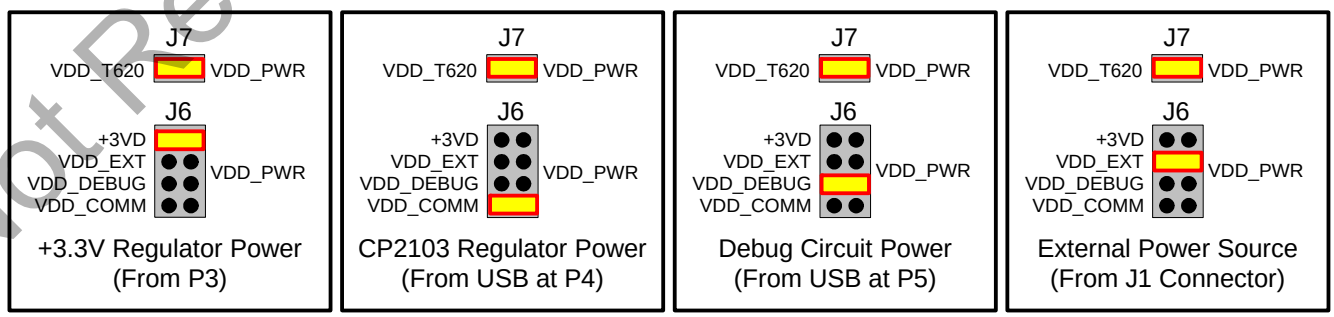


Figure 12. J6 and J7 Shorting Block Configuration for Power Options

7.4. 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 motherboard 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 C8051T62x/32x device.

7.5. UART to USB Communications Interfaces (COMM/P4)

The C8051T62x Motherboard 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 “COMM” (P4). This USB connector supplies power to the CP2103 and can supply power to the rest of the motherboard by configuring J6 and J7 as shown in Figure 12. 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.

7.6. Communications Interface Selector Headers (J8 and J11)

The C8051T62x Motherboard routes the C8051T62x/32x's P0.4 (UART TX) and P0.5 (UART RX) to J11 where those signals can be connected to either the CP2103 USB-To-UART bridge or the USB Debug Adapter. The motherboard also allows the C8051T62x/32x's P1.1 and P1.2 to be used as the UART control signals, CTS and RTS. These two signals are routed to J8, where they can be connected to either the CP2103 or the USB Debug Adapter.

The jumper options for using either the CP2103 or the Debug Adapter circuit for UART communications can be found in Figure 13.

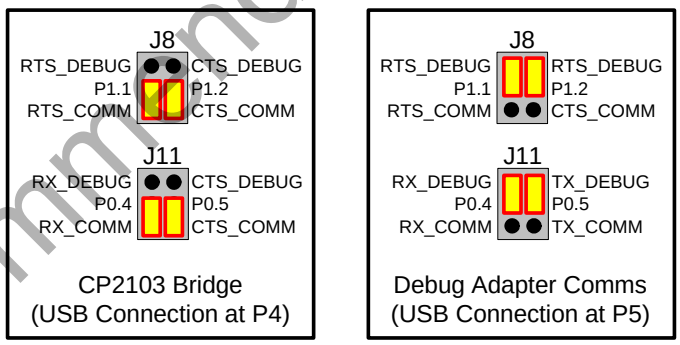


Figure 13. Shorting Block Configuration for UART Communication Options

7.7. PORT I/O Connectors (J2, J3, J4, and J5)

Each of the C8051T62x/32x's I/O pins, as well as +3VD and GND, are routed to headers J2 through J5. J2 connects to the microcontroller's Port 0 pins; J3 connects to Port 1; J4 connects to Port 2, and J5 connects to Port 3.

7.8. Analog I/O (J1 and J14)

Three of the C8051T62x/32x 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 3 for the J1 terminal block connections. Placing a shorting block on J14 will connect the P0.7/VREF signal on J1 to the P0.7 pin of the device.

Table 3. J1 Terminal Block Descriptions

Pin #	Description
1	VREGIN
2	VIO
3	GND
4	P2.5 (Analog Input)
5	P0.7/VREF (routed to header J14)
6	VDD_EXT (routed to header J6)

7.9. VPP Connection (J15)

The C8051T62x/32x devices require an external 6.0 V programming voltage applied to the VPP pin during device programming. The VPP pin on these devices is shared with P1.5 or P1.1 depending on the device. During programming, the VPP voltage is automatically enabled when needed. Header J15 is provided to allow the user to disconnect the programming circuitry from the VPP pin to avoid interfering with the normal application operation of the GPIO pin. When programming the device, J15 should be shorted with a shorting block. When running normal application code, J15 can be removed. See Table 4 for more information on which port pins are shared with VPP.

Table 4. VPP Pin Sharing

Device	Pin Shared with VPP
C8051T620 C8051T621 C8051T626 C8051T627 C8051T320 C8051T321 C8051T322 C8051T323	P1.5
C8051T622 C8051T623 C8051T326 C8051T327	P1.1

7.10. Using Alternate Supplies with the C8051T62x 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 (e.g., 1.8 V), an external supply voltage can be applied to the board at the analog connector (J1). Some devices in the C8051T62x/32x family also support a separate voltage input for the input/output voltage of the port pins. This Voltage Input/Output (VIO) should be input to J1 on Pin 2. See the C8051T620-21-26-27_T320-3 or C8051T622-23_T326-27 data sheet for more information about VIO usage and constraints.

Notes:

1. When programming a C8051T62x/32x device, VDD must be at least 3.3 V. VDD can be supplied directly to the device, or the on-chip 5 V regulator can be used.
2. If an external supply voltage is desired, the shorting block on J6 should be placed so that the Pin 3 (VDD_EXT) is shorted to Pin 4 (VDD_PWR).

8. Schematics

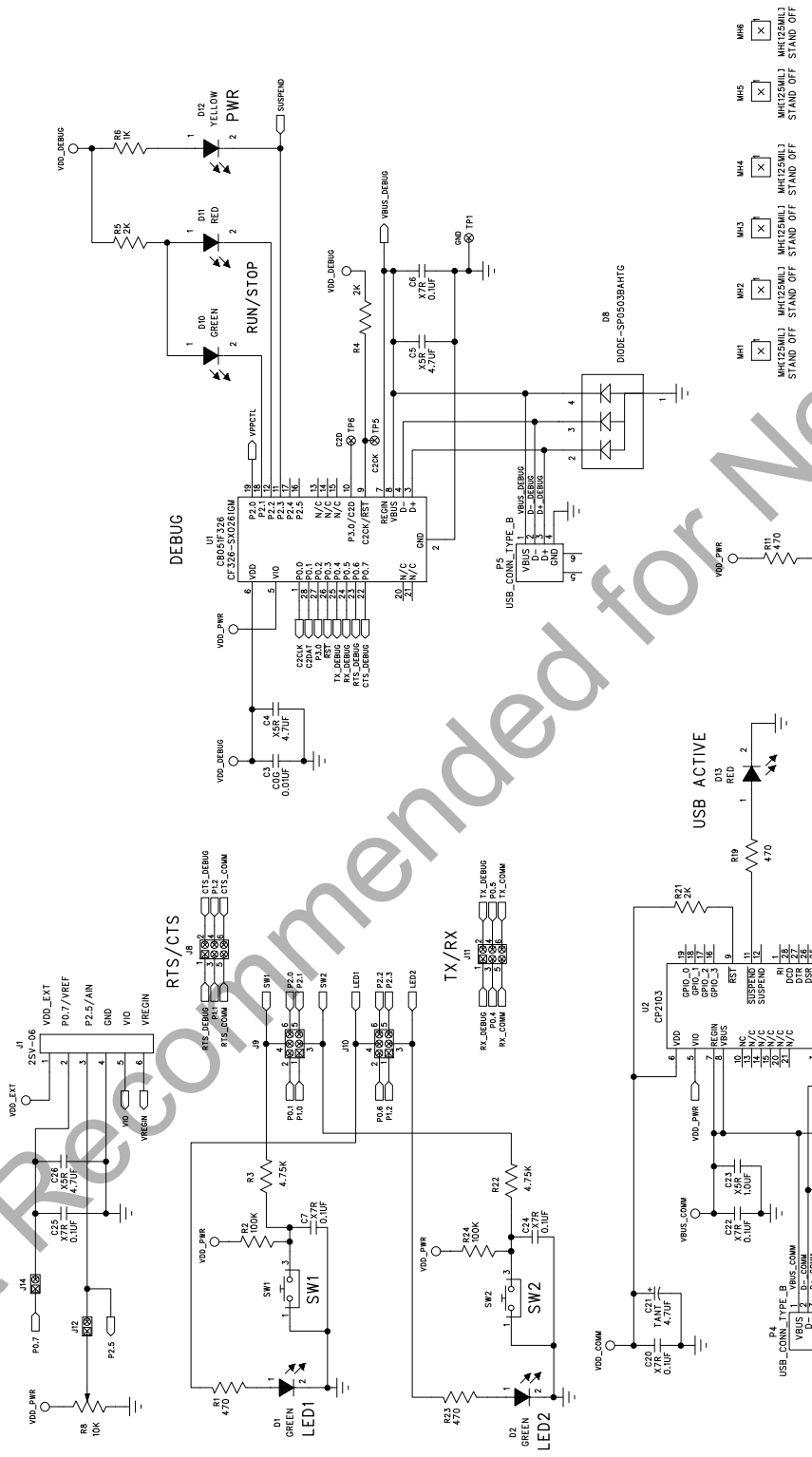


Figure 14. C8051T62x Motherboard Schematic (1 of 2)

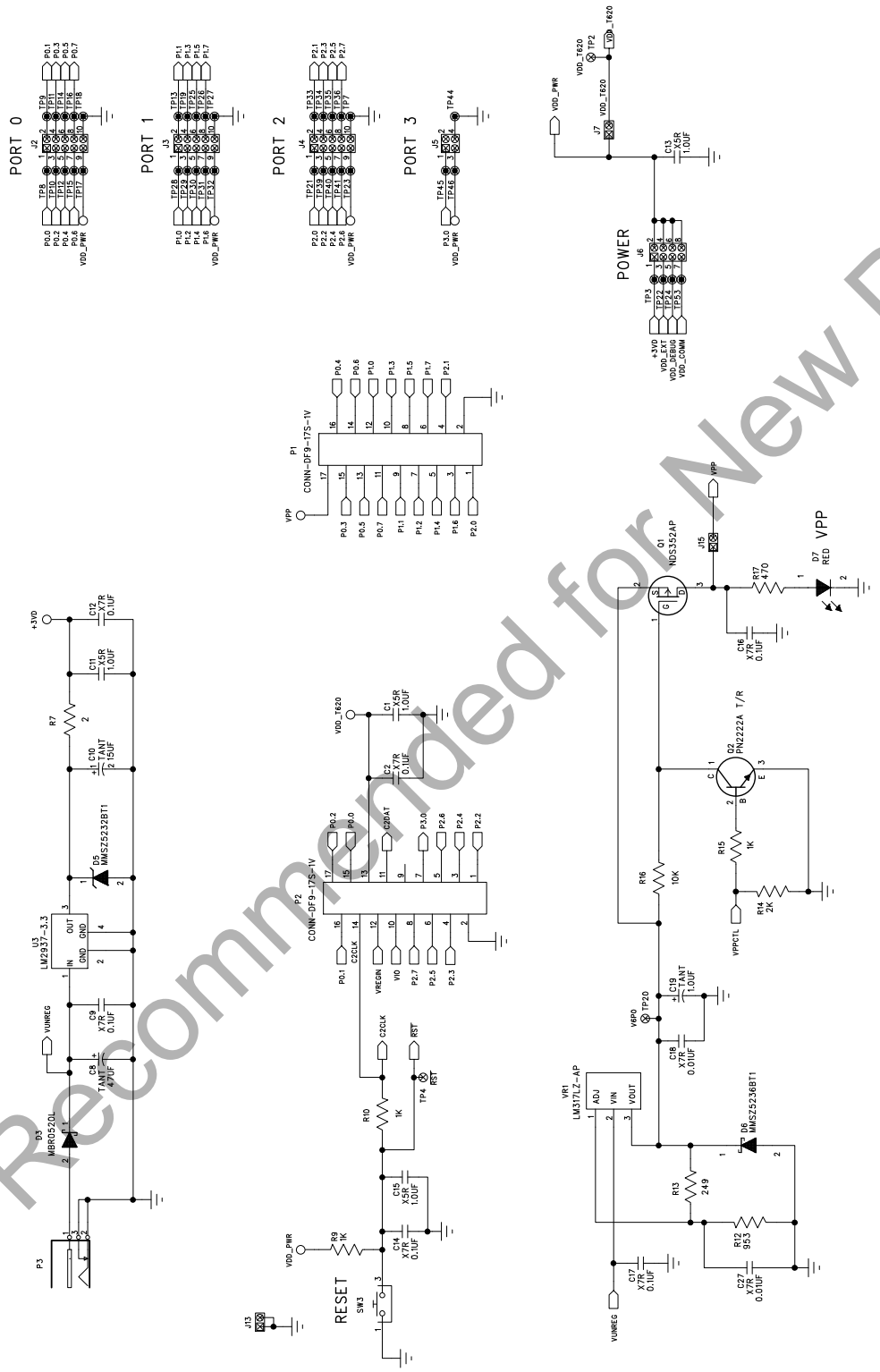


Figure 15. C8051T62x Motherboard Schematic (2 of 2)

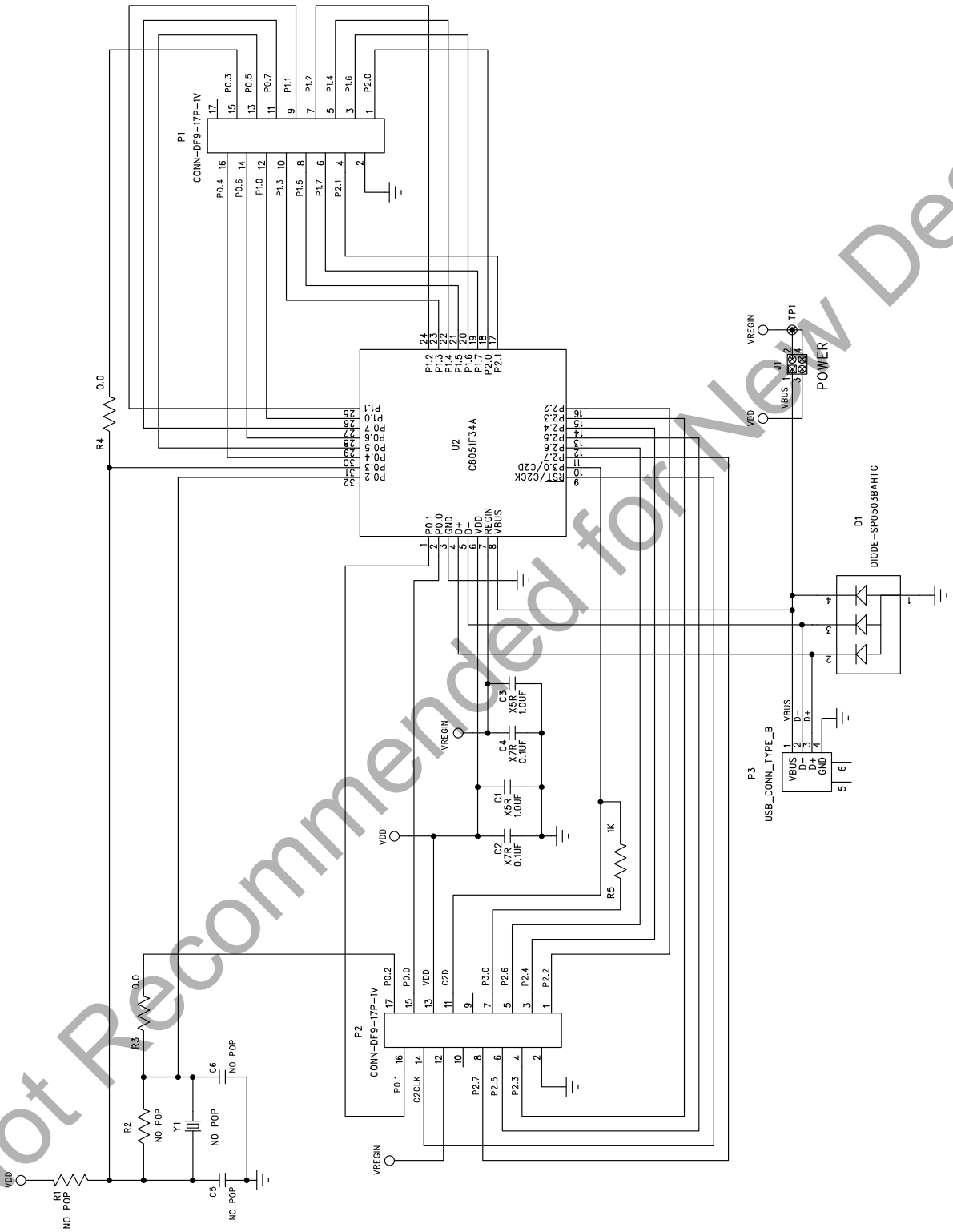


Figure 16. C8051T62x Emulation Daughter Board Schematic

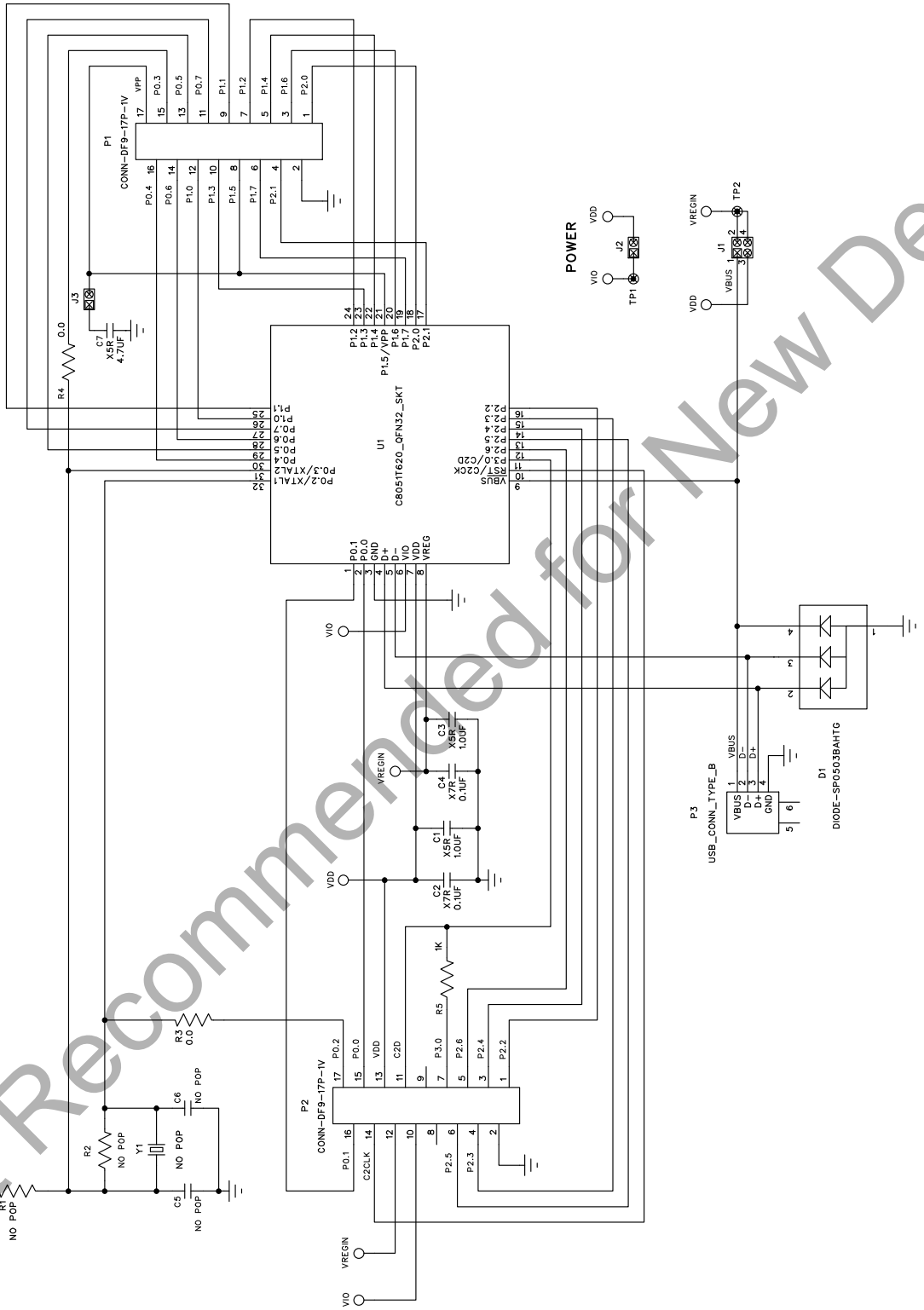


Figure 17. C8051T620 QFN-32 Daughter Board Schematic

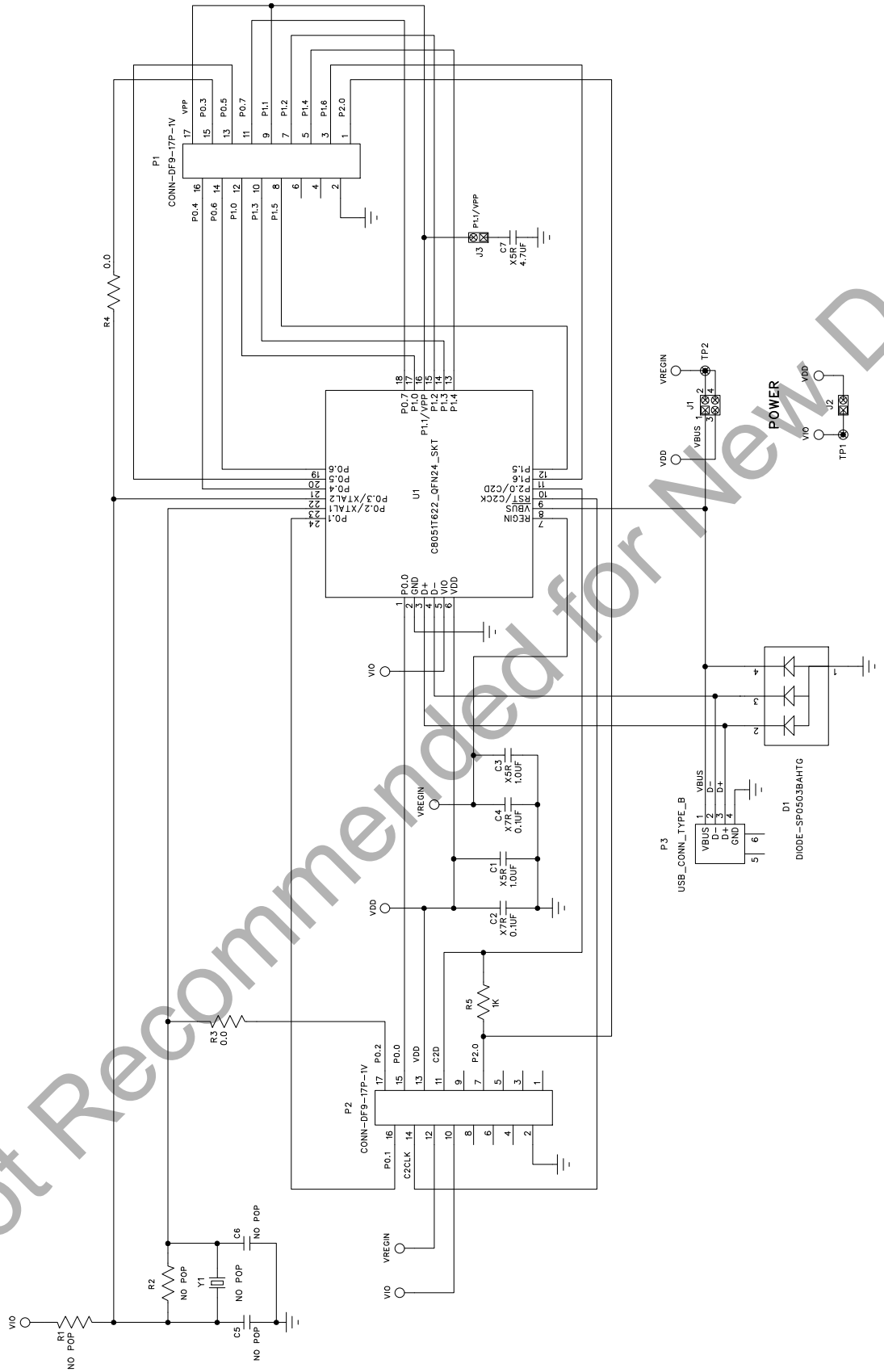


Figure 18. C8051T622 QFN-24 Daughter Board Schematic

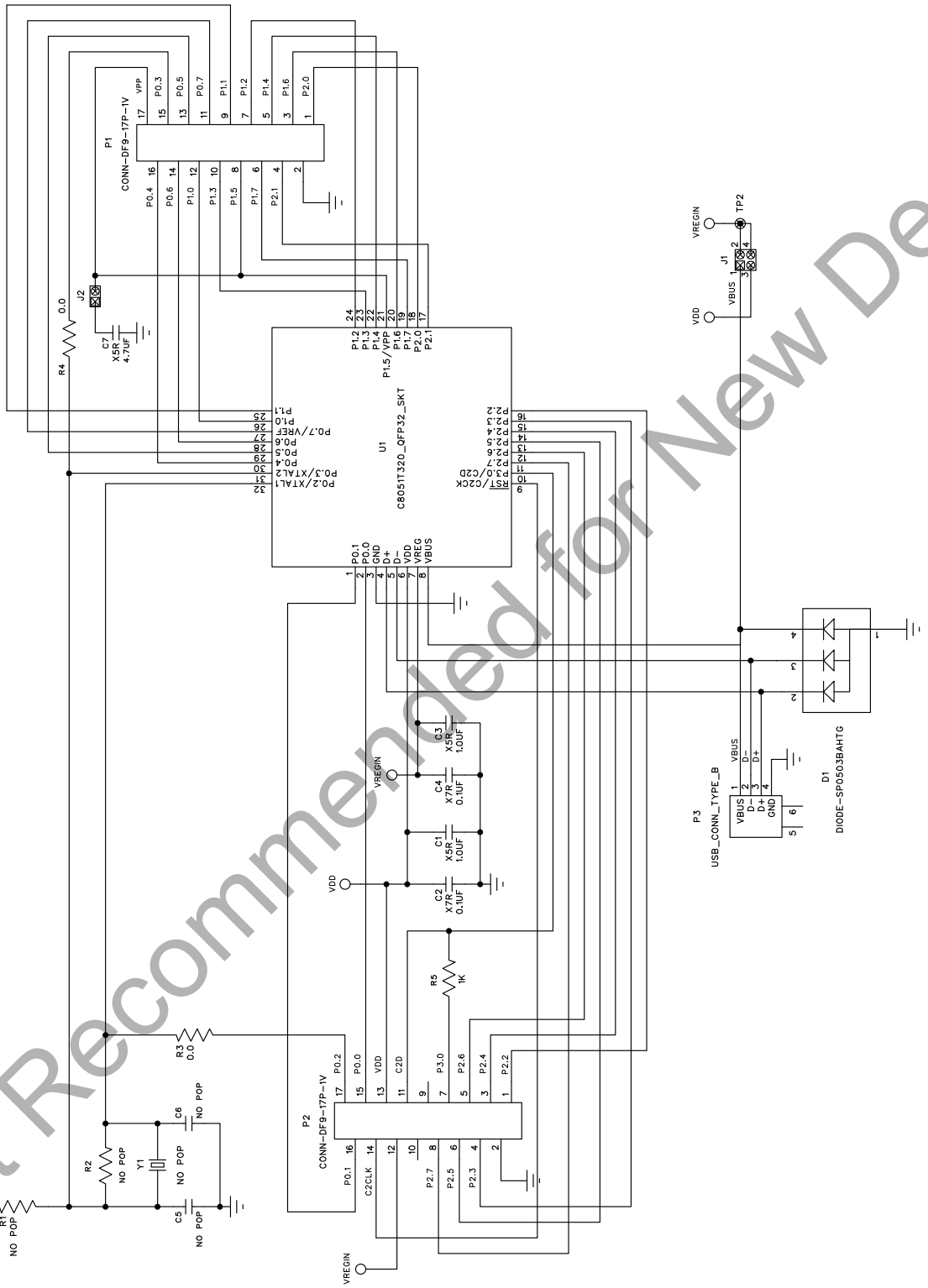


Figure 19. C8051T320 QFP-32 Daughter Board Schematic

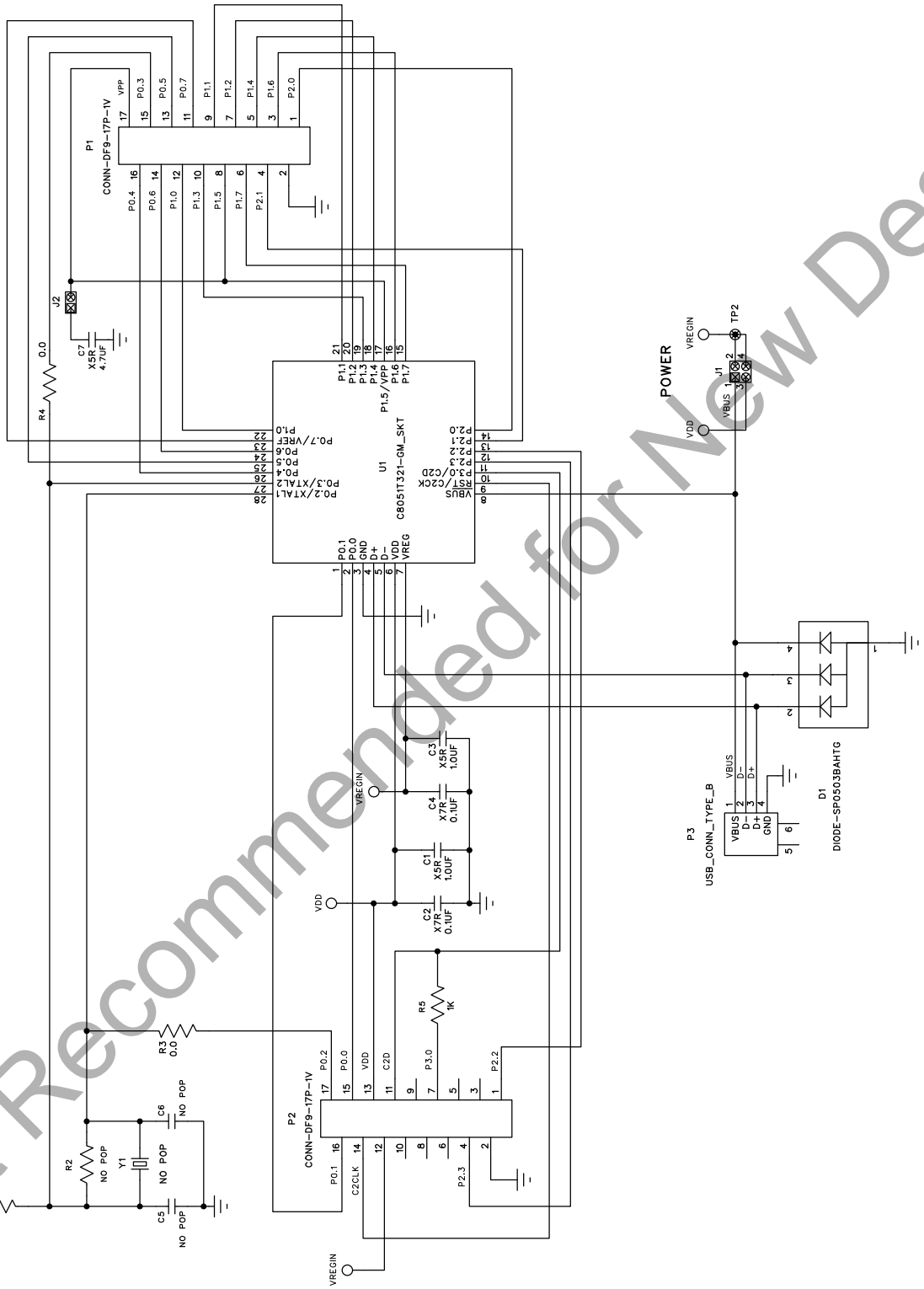


Figure 20. C8051T321 QFN-28 Daughter Board Schematic

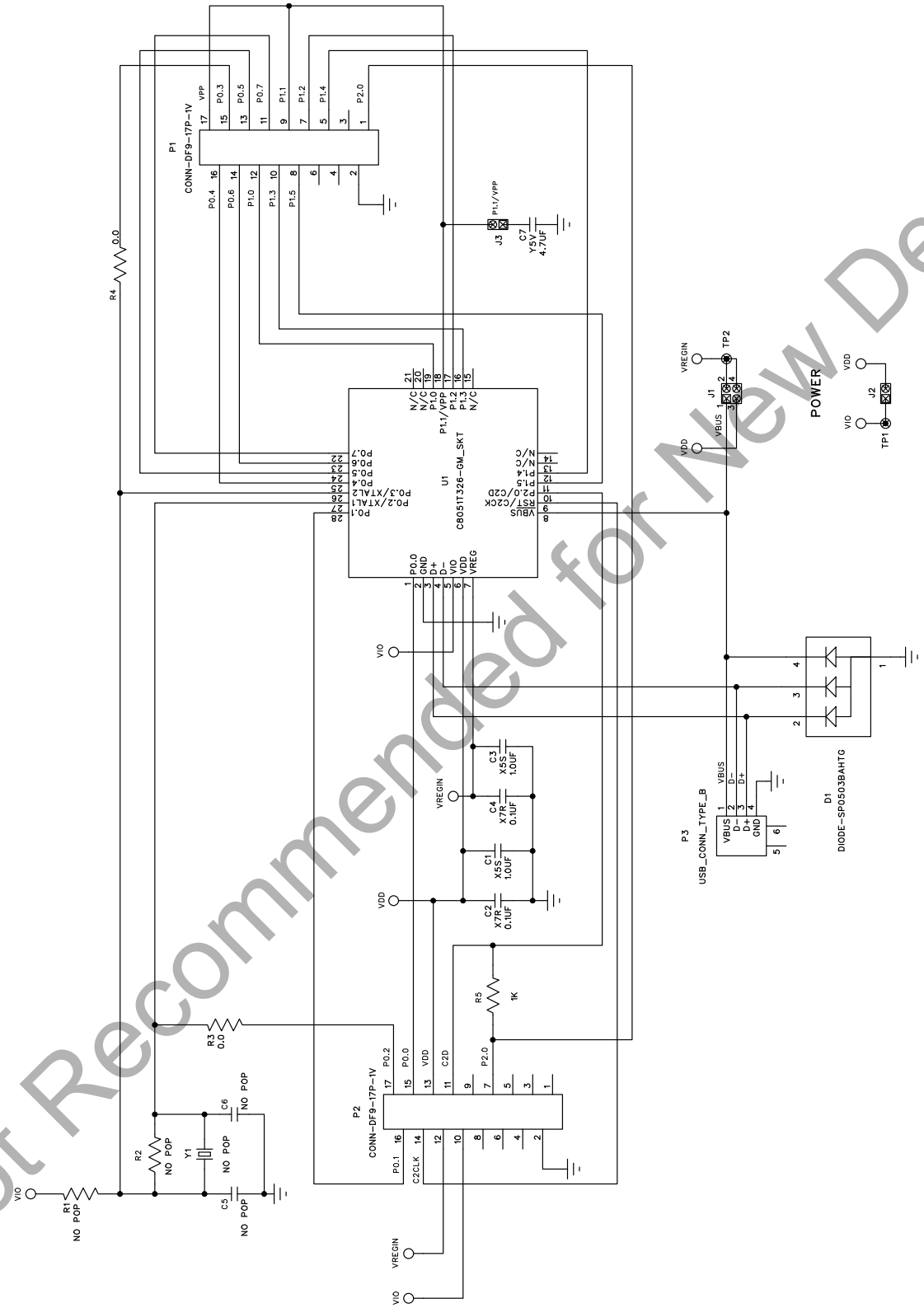


Figure 21. C8051T326 QFN-28 Daughter Board Schematic

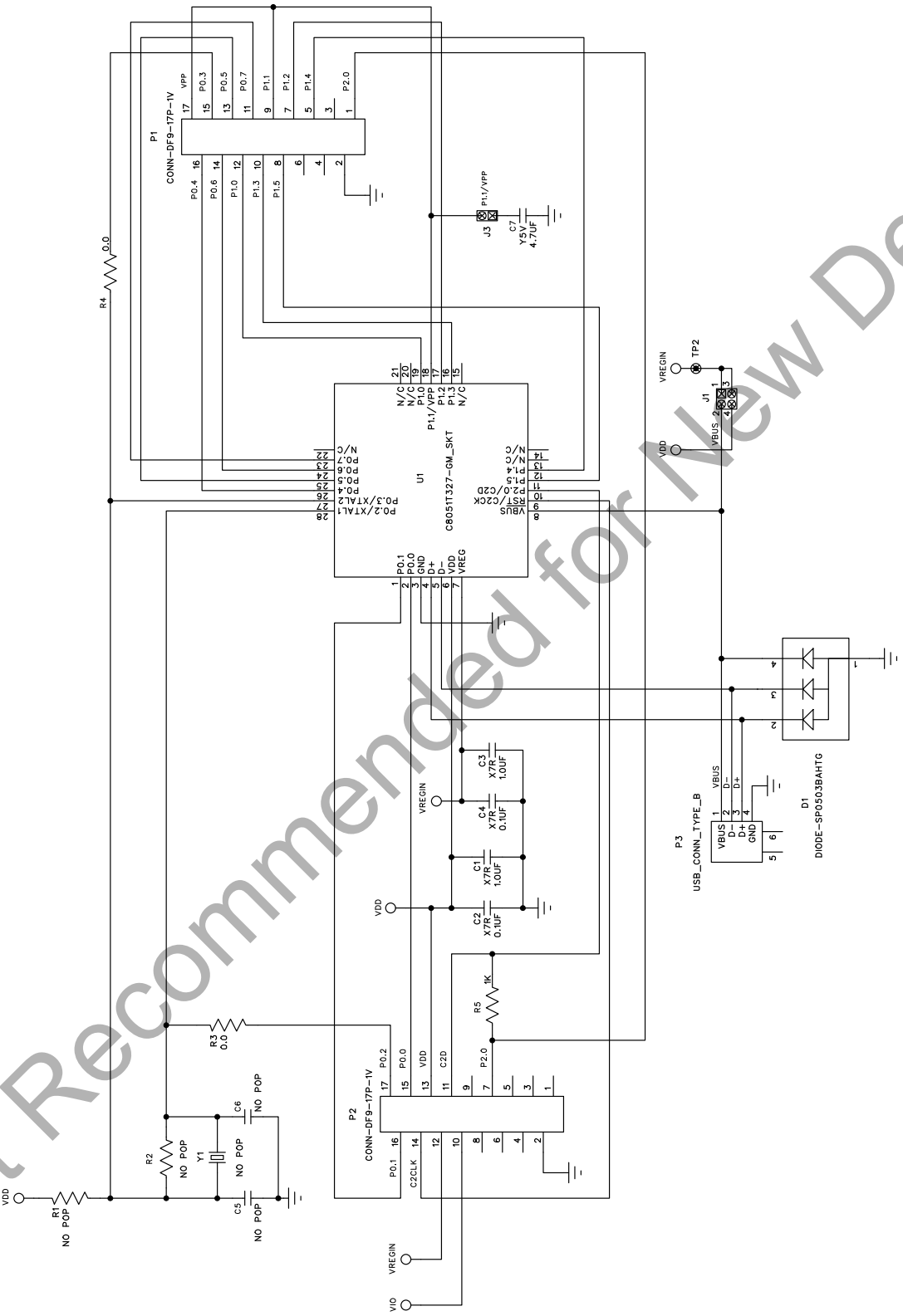


Figure 22. C8051T327 QFN-28 Daughter Board Schematic

Document Change List

Revision 0.2 to Revision 0.3

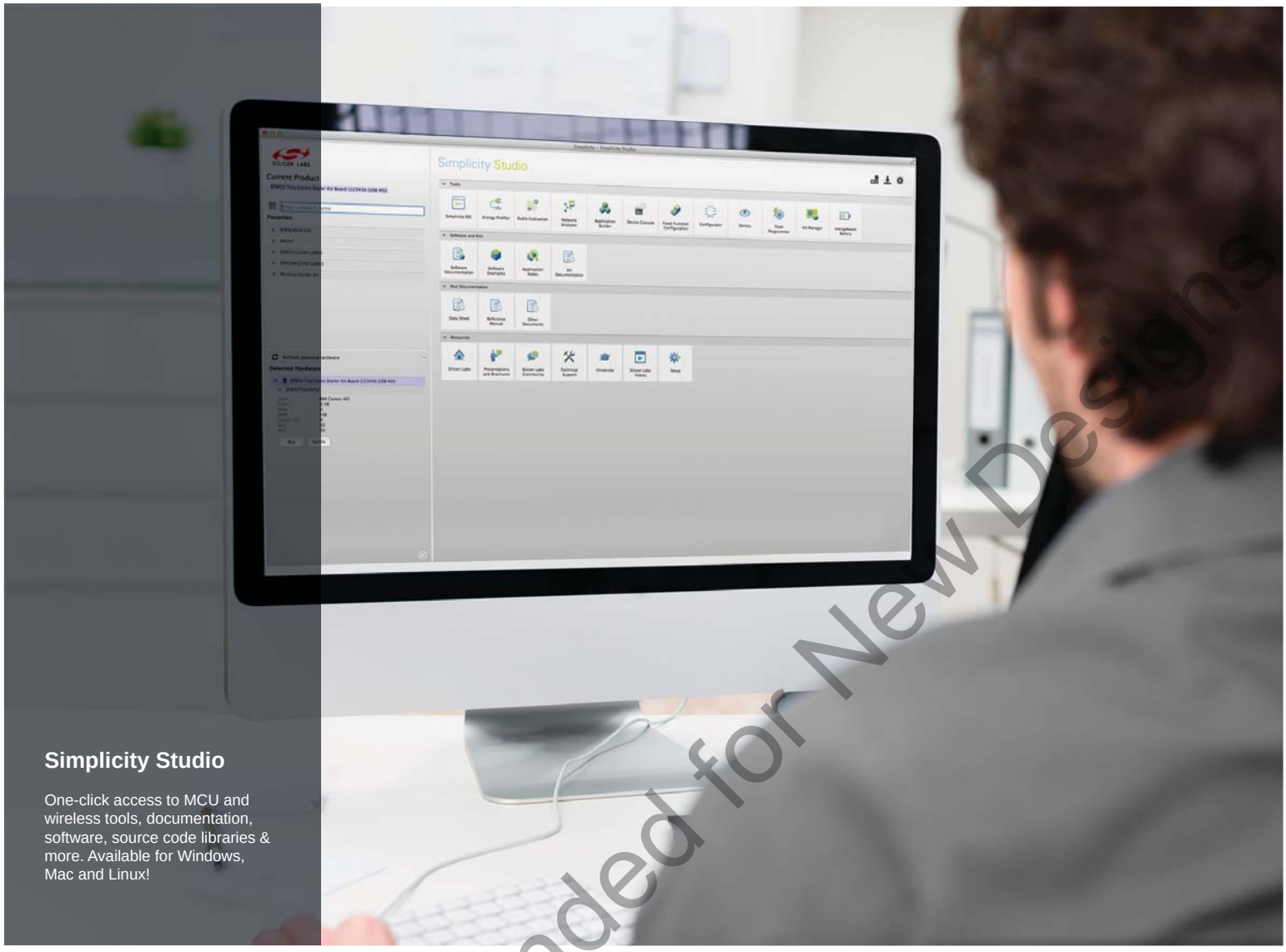
- Updated "4.2. Development Tools Installation" on page 3.
- Updated "4.3. CP210x USB to UART VCP Driver Installation" on page 3.
- Updated Figure 17 on page 22.
- Updated Figure 18 on page 23.
- Updated Figure on page 28.

Revision 0.3 to Revision 0.4

- Updated "1. Kit Contents" on page 1.
- Updated "2. About the Daughter Boards" on page 1.
- Updated "6. Example Source Code" on page 8.
  - Updated project paths
- Updated "7. Development Boards" on page 9.
- Updated "8. Schematics" on page 19.
  - Added Figures 19, 20, 21, and 22.
- Updated C8051T62x references to include C8051T32x devices.
- Updated data sheet references.

Revision 0.4 to Revision 0.5

- Updated "1. Kit Contents" on page 1.
- Updated document to reflect change from CD-ROM based installation to web-based installer.
- Updated various sections to include the new C8051T626/7 part numbers.



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