



CY3240-I2USB

USB-I2C Bridge Guide

Doc. # 001-66660 Rev.*B

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone (USA): 800.858.1810
Phone (Intl): 408.943.2600
<http://www.cypress.com>

Copyrights

© Cypress Semiconductor Corporation, 2011-2013. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems, where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use, and in doing so indemnifies Cypress against all charges.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress), and is protected by, and subject to worldwide patent protection (United States and foreign), United States copyright laws, and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and/or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application, or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems, where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use, and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.

PSoC Designer™ and Programmable System-on-Chip™ are trademarks and PSoC® is a registered trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Flash Code Protection

Cypress products meet the specifications contained in their particular Cypress PSoC Data Sheets. Cypress believes that its family of PSoC products is one of the most secure families of its kind on the market today, regardless of how they are used. There may be methods, unknown to Cypress, that can breach the code protection features. Any of these methods, to our knowledge, would be dishonest and possibly illegal. Neither Cypress nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Cypress is willing to work with the customer who is concerned about the integrity of their code. Code protection is constantly evolving. We at Cypress are committed to continuously improving the code protection features of our products.

Contents



1. Introduction	5
1.1 Kit Contents	5
1.1.1 Prerequisites	5
1.2 Additional Learning Resources	6
1.3 Document History	6
1.4 Documentation Conventions	6
2. Getting Started	9
2.1 Kit Installation	9
2.2 PSoC Designer	13
2.3 PSoC Programmer	14
2.4 Bridge Control Panel	15
3. Kit Operation	17
3.1 Introduction	17
3.2 Connect Bridge to Device	17
3.2.1 Program USB-I2C Bridge	18
3.3 Connect Demonstration Board to Bridge	19
3.3.1 Run Demonstration Board Test	19
4. Hardware	25
4.1 System Block Diagram	25
4.2 Functional Description	26
4.2.1 PSoC CY8C24894 Chip	27
4.2.2 USB Mini B Connector	28
4.2.3 ISSP Programming Header	28
4.2.4 GPIO Pins	29
4.2.5 I2C Slave Interface Connector	30
4.2.6 Demo Board	31
5. Code Examples	33
5.1 My First Code Example	33
5.1.1 Project Objective	33
5.1.2 USB-I2C Demo Board Flowchart	34
5.1.3 Creating My First PSoC 1 Project	35
5.1.4 Verify Output	51
5.2 USBtoIIC Code Example	53
5.2.1 Project Description	53
5.2.2 PSoC Designer Interconnected View	54
5.2.3 USBtoIIC Flowchart	55
5.2.4 Verify Output	57

A. Appendix 59

- A.1 Schematic.....59
 - A.1.1 USB-I2C Bridge Schematic.....59
 - A.1.2 Demo Board Schematic.....61
- A.2 USB-I2C Bridge Board Layout.....62
 - A.2.1 Demo Board Layout.....62
- A.3 Bill of Materials (BOM).....63
 - A.3.1 USB-I2C Bridge BOM63
 - A.3.2 Demo Board.....64

1. Introduction



Thank you for your interest in the CY3240 USB-I2C Bridge Kit. The CY3240 USB-I2C Bridge can be used with the Bridge Control Panel that is installed with PSoC Programmer. The kit is designed to test, tune, and debug programs that have an I2C slave interface. This document describes the working of the CY3240 USB-I2C Bridge and includes code examples, which will help you get started with the bridging tool. You can also develop custom programs using PERL, Python, C++, and C# languages. These custom applications allow generating complex testing, debugging, and validation systems using PSoC Programmer.

You can evaluate the kit using the code examples provided with the kit. The code examples help to:

- Explore I2USB communication between the PC and the slave, interfaced via I2C slave interface connector of the bridge
- Customize the designs provided along with the kit

The CY3240 USB-I2C Bridge board is configured with the “USBtoIIC” code example when shipped. The I2C demo board is configured with “My First Code Example”. See Chapter 5 for details. Evaluate the examples provided with the kit and then experiment with the kit hardware and software to create your own designs.

1.1 Kit Contents

The CY3240 USB-I2C Bridge Kit contains:

- USB-I2C Bridge
- I2C demo board with temperature and ambient light sensor
- USB A to Mini B cable
- CY3240 USB-I2C Bridge kit CD
 - PSoC Designer installation file
 - PSoC Programmer installation file
 - Bridge Control Panel installation file (packaged along with PSoC Programmer)
 - Code examples
 - Hardware file
 - Kit guide
 - Quick start guide
 - Release notes

Inspect the contents of the kit. If any parts are missing, contact your nearest Cypress sales office for further assistance.

1.1.1 Prerequisites

The CY3217-MiniProg1 kit is required to reprogram the CY3240 USB-I2C Bridge and I2C demo board. MiniProg is a small and compact programming tool that connects to your PC using the

provided USB 2.0 cable. During prototyping, the MiniProg can be used as an in-system serial programmer (ISSP) to program all PSoC 1 devices on your PCB. This kit can be purchased from <http://www.cypress.com/go/CY3217-MiniProg1>.

1.2 Additional Learning Resources

Visit www.cypress.com for additional learning resources in the form of data sheets, technical reference manual, and application notes.

- Application Note - I2C-USB Bridge Usage
<http://www.cypress.com/?rID=2853>
- Application Note - Using Cypress I2C Port Expander with Flash Storage:
<http://www.cypress.com/?rID=2694>
- PSoC CY8C24894 - Features and Chip functionality:
<http://www.cypress.com/?rID=37765>
- PSoC CY8C21123 - Features and Chip functionality:
<http://www.cypress.com/?rID=3335>
- For more information regarding PSoC Designer functionality and releases:
www.cypress.com/go/psocdesigner
- For more information regarding PSoC Programmer, supported hardware and COM layer:
www.cypress.com/go/psocprogrammer
- For a list of PSoC Designer-related trainings:
<http://www.cypress.com/?rID=40543>

1.3 Document History

Revision	PDF Creation Date	Origin of Change	Description of Change
**	01/25/2011	RKPM	Initial version of kit guide
*A	04/14/2011	RKPM	Added My First Code Example section. Content edits throughout the document
*B	01/17/2013	KUK	Updated images in sections 2 and 5 and minor content updates through the document.

1.4 Documentation Conventions

Table 1-1. Document Conventions for Guides

Convention	Usage
Courier New	Displays file locations, user entered text, and source code: C:\ ...cd\icc\
<i>Italics</i>	Displays file names and reference documentation: Read about the <i>sourcefile.hex</i> file in the <i>PSoC Designer User Guide</i> .
[Bracketed, Bold]	Displays keyboard commands in procedures: [Enter] or [Ctrl] [C]

Table 1-1. Document Conventions for Guides

Convention	Usage
File > Open	Represents menu paths: File > Open > New Project
Bold	Displays commands, menu paths, and icon names in procedures: Click the File icon and then click Open .
Times New Roman	Displays an equation: $2 + 2 = 4$
Text in gray boxes	Describes cautions or unique functionality of the product.

2. Getting Started



This chapter describes the installation and configuration of the CY3240 USB-I2C Bridge kit.

2.1 Kit Installation

To install the kit software, follow these steps:

1. Insert the kit CD into the CD drive of your PC. The CD is designed to auto-run and the kit installer startup screen appears.

Note You can also download the latest kit installer from <http://www.cypress.com/go/CY3240-I2USB>. Three different types of installers are available for download.

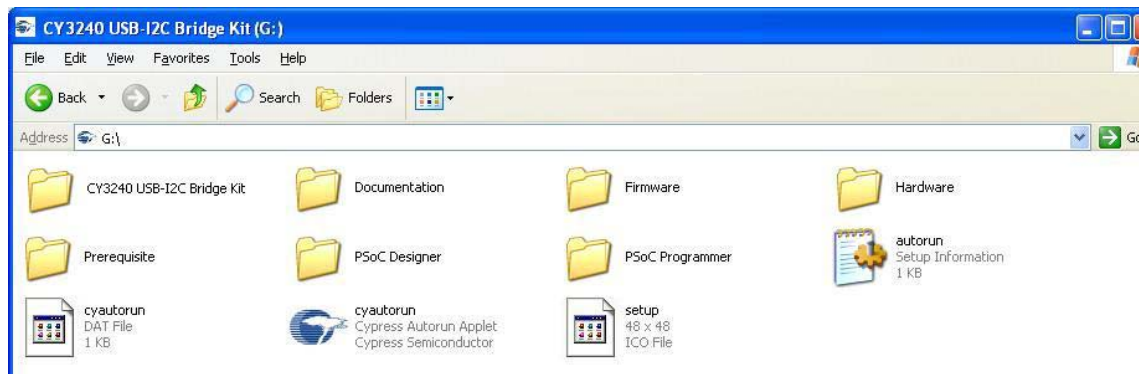
- a. CY3240 USB-I2C Bridge_ISO: This file (ISO image) is an archive of the optical disc provided with the kit. You can use this to create an installer CD or extract information using WinRar or similar tools.
 - b. CY3240 USB-I2C Bridge_Single Package: This executable file installs the kit CD contents, including PSoC Programmer, PSoC Designer, code examples, hardware files, and user documents.
 - c. CY3240 USB-I2C Bridge_Single Package (without prerequisites): This executable file installs only the kit contents, which includes kit code examples, hardware files, and user documents.
2. Click **Install CY3240 USB-I2C Bridge** to start the installation, as shown in [Figure 2-1](#).

Figure 2-1. CY3240 USB-I2C Bridge Kit Menu



Note If auto-run does not execute, double-click *cyautorun.exe* file on the root directory of the CD, as shown in [Figure 2-2](#).

Figure 2-2. Root Directory of CD



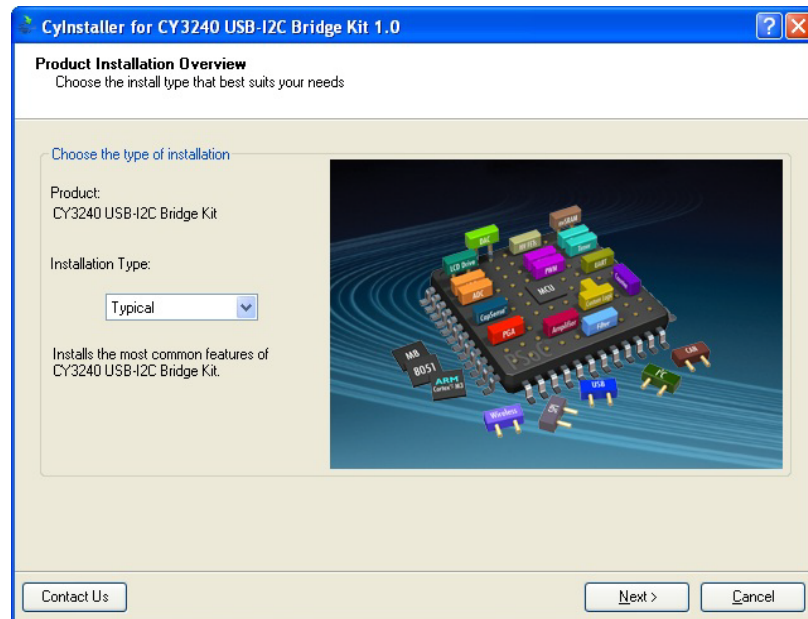
3. On the startup screen, click **Next** to start the installer.
4. In the **InstallShield Wizard**, choose the folder location to install the setup files. You can change the location of the folder using **Change**, as shown in [Figure 2-3](#).
5. Click **Next** to launch the installer.

Figure 2-3. InstallShield Wizard



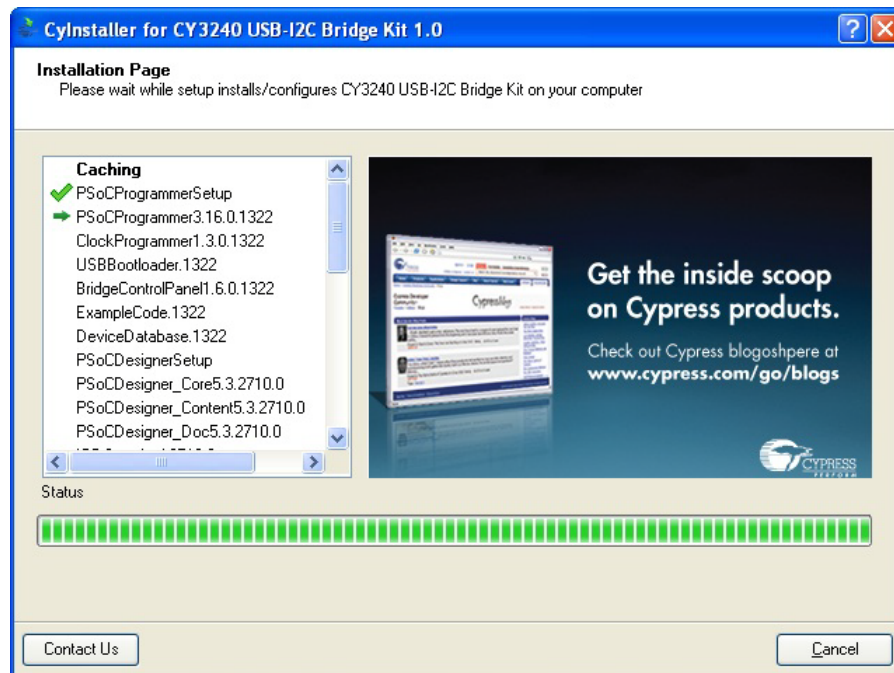
6. On the **Product Installation Overview** screen, select the installation type that best suits your requirement. The drop-down menu has three options: **Typical**, **Complete**, and **Custom**, as shown in [Figure 2-4](#).
7. Click **Next** to start the installation.

Figure 2-4. Installation Type Options



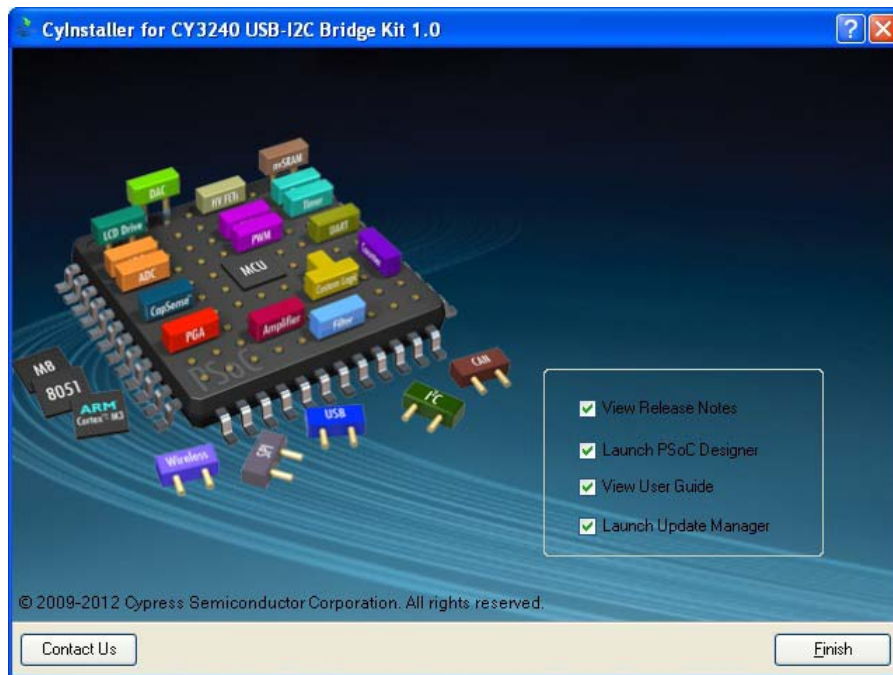
8. When the installation begins, a list of packages appears on the **Installation Page**.
9. A green check mark appears adjacent to every package that is downloaded and installed, as shown in Figure 2-5.
10. Wait until all the packages are downloaded and installed successfully.

Figure 2-5. Installation Page



11. Click **Finish** to complete the installation, as shown in Figure 2-6.

Figure 2-6. Installation Complete Page



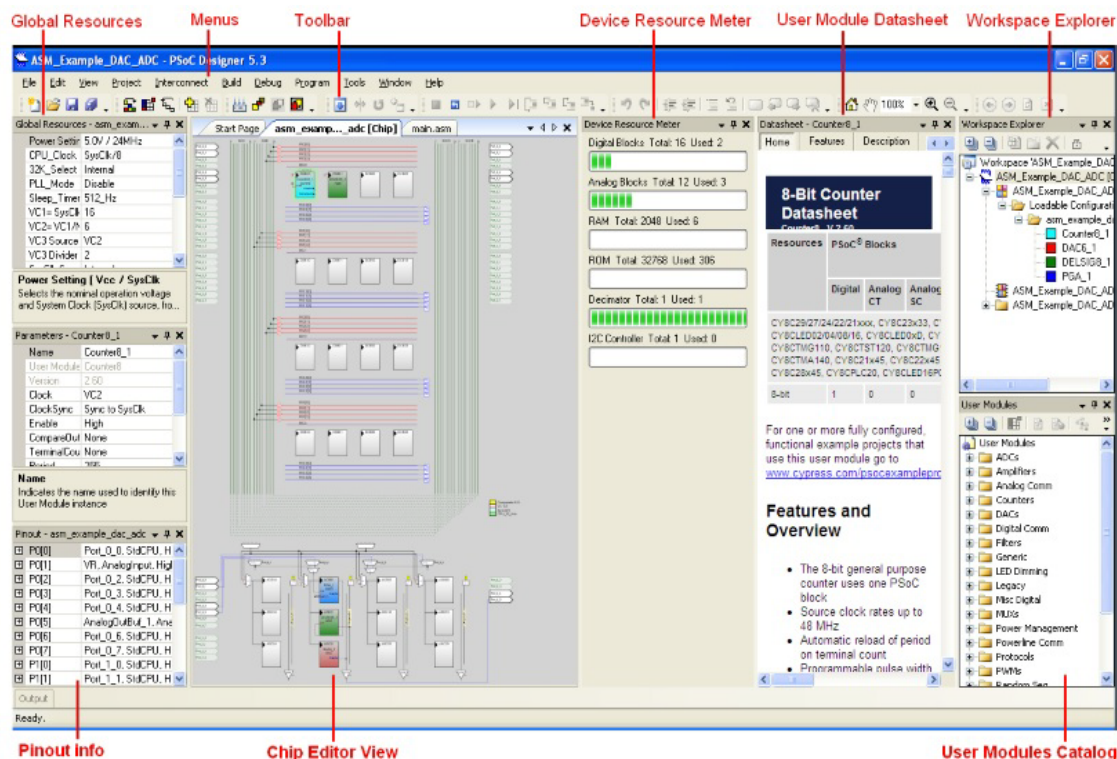
After the software installation, verify that you have all hardware and drivers set up for the CY3240 USB-I2C Bridge kit by connecting the bridge to your PC via its USB interface. Because this is the first time you have connected this board to the PC, initial drivers are installed. Follow the instructions to complete the installation process. Also, verify your installation by opening PSoC Programmer with the CY3240 USB-I2C Bridge board attached to the USB port of the PC.

2.2 PSoC Designer

PSoC Designer is the revolutionary integrated design environment (IDE) that helps to customize PSoC to meet your specific application requirements. PSoC Designer software accelerates system bring-up and time-to-market.

1. To open PSoC Designer, click **Start > All Programs > Cypress > PSoC Designer <version> > PSoC Designer <version>**
2. To create a new project, click **File > New Project**
3. To open an existing project, click **File > Open**
4. To experiment with the code examples, go to [Code Examples on page 33](#)

Figure 2-7. PSoC Designer Interconnect View



Note For more details on PSoC Designer, see the PSoC Designer IDE Guide located at: <Install_Directory>\Cypress\PSoC Designer\<version>\Documentation.

You can also see the PSoC Designer Quick Start Guide at: <http://www.cypress.com/?rID=47954>

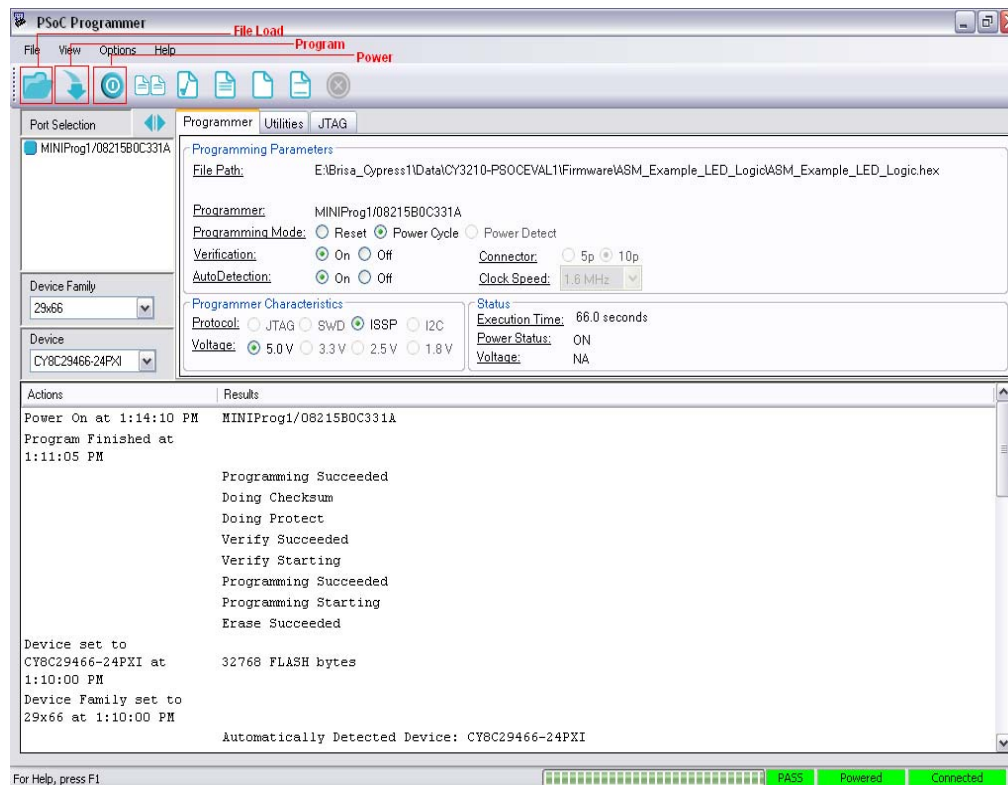
2.3 PSoC Programmer

To open PSoC Programmer, click **Start > All Programs > Cypress > PSoC Programmer <version> > PSoC Programmer <version>**.

To successfully program the device, follow these steps:

1. Select the MiniProg1 in **Port Selection**, as shown in [Figure 2-8](#).

Figure 2-8. PSoC Programmer Window



2. Click the **File Load** button to load a hex file.
3. Click the **Program** button to program the hex file into the chip.
4. Close PSoC Programmer.

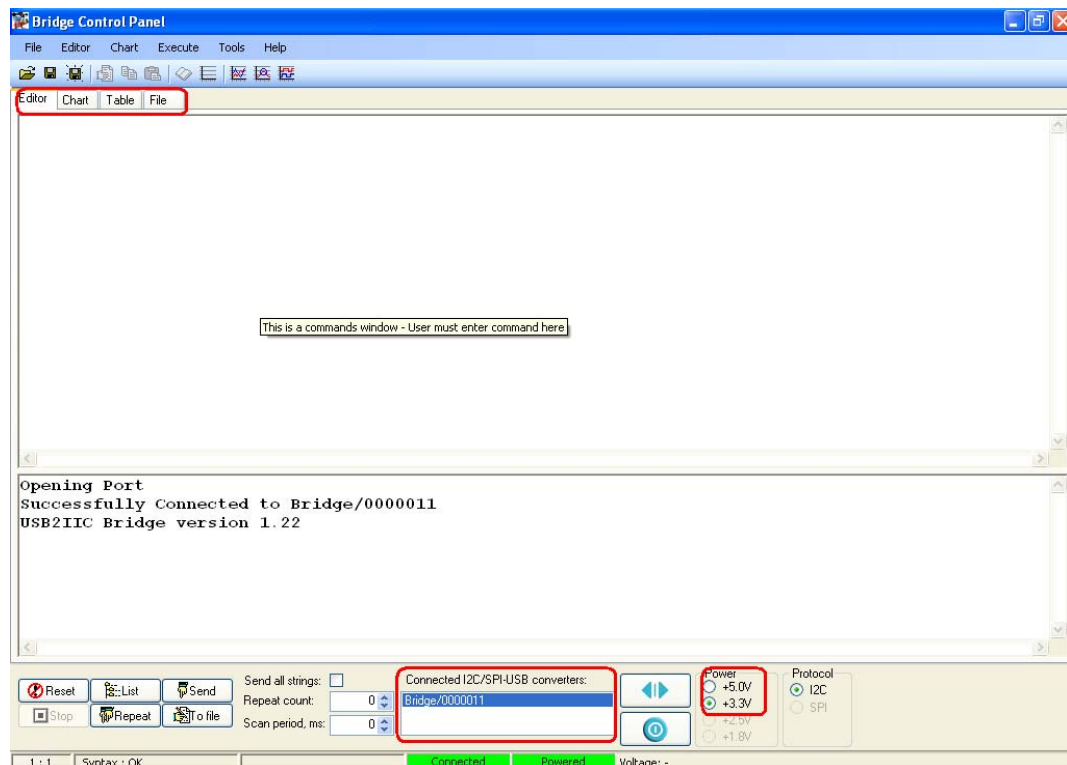
For more details on PSoC Programmer, see the user guide at the following location:
 <Install_Directory>:\Cypress\Programmer\<version>\Documents

2.4 Bridge Control Panel

The Bridge Control Panel is used with CY3240 USB-I2C Bridge to enable communication with I2C slave devices. This program is used to configure I2C devices and also to acquire and process data received from I2C slave devices. The Bridge Control Panel helps in optimizing, debugging, and calibrating the target application.

1. Click **Start > All Programs > Cypress > Bridge Control Panel <version> > Bridge Control Panel**.
2. Select **Power Supply**, as highlighted in [Figure 2-9](#).
3. Select the port connectivity bridge from the **Connected I2C/SPI-USB converters**.

Figure 2-9. Select Bridge



Note For more information, go to **Bridge Control Panel > Help > Help Contents**.

3. Kit Operation



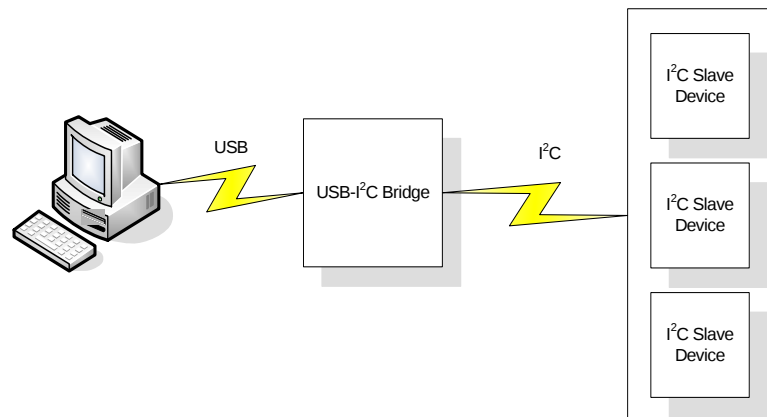
3.1 Introduction

This chapter provides key information regarding the CY3240 USB-I2C Bridge specifications. It also gives a brief overview of how to verify that you have set up the kit correctly. The device meets the requirements of the I2C Specifications for standard and fast speed I2C devices, and supports USB human interface devices (HID). The bridge is powered by the USB and consumes less than 500 mA. The device can be configured for several I2C clock rates such as 50 kHz, 100 kHz, and 400 kHz.

The number of devices that can be connected is constrained only by the I2C address limit and physical ability of the I2C bus. For more details, see the [I2C Specifications](#).

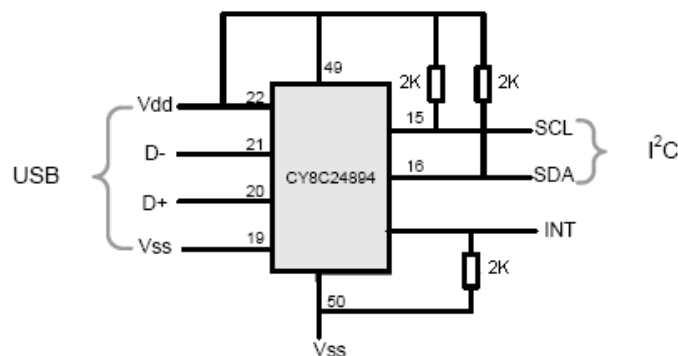
The USB communication function uses two 64-byte packets: one for input data flow and the other for output data flow. The maximum bandwidth of this configuration is 64 bytes. This is sufficient for most USB-I2C bridge applications ([Figure 3-1](#)).

Figure 3-1. USB-I2C Bridge



3.2 Connect Bridge to Device

Figure 3-2. Simplified Schematic of USB-I2C Bridge



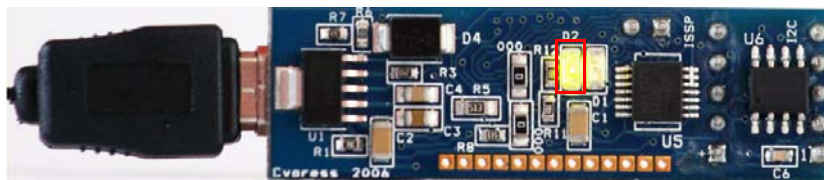
Perform the following steps to connect the device to CY3240 USB-I2C Bridge, as shown in [Figure 3-2](#):

1. Connect GND of the device to GND of the bridge.
2. Connect the SDA and SCL lines to the bridge.

The CY3240 USB-I2C Bridge has 2.2 k pull-up resistors connected to +5 V. The INT pin is a pull-down bidirectional pin that can be used as an additional signal between the bridge and the I2C slave device for functions such as sleep mode control.

3. Power the device from the Vdd pin on the bridge, if it does not have its own power supply. Note that the Vdd connection between the bridge and demo board is required, even if the board is self-powered. Optionally, the bridge can provide 3.3 V or 5 V, or work with an externally powered board using 2.4 V to 5.6 V.
4. When the connection between the bridge and USB is successful, the LED (green) lights up, as shown in [Figure 3-3](#).

Figure 3-3. Connecting Bridge to USB.

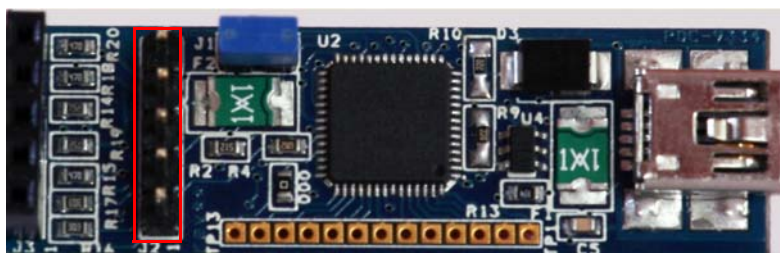


5. Open the Bridge Control Panel from the PC to work with the bridge.

3.2.1 Program USB-I2C Bridge

The CY3240 USB-I2C Bridge can be programmed using a MiniProg at the programming header of the bridge. To use MiniProg, use the ISSP programming header (J2) on the board, as highlighted in [Figure 3-4](#).

Figure 3-4. ISSP Programming Header



3.3 Connect Demonstration Board to Bridge

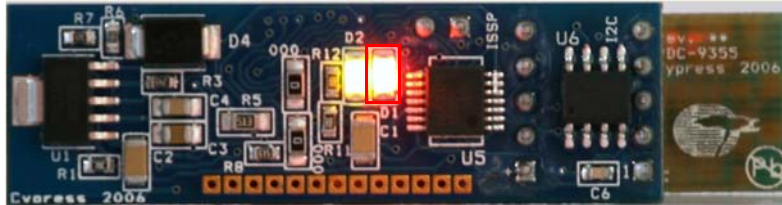
1. Connect the I2C slave demonstration board to the bridge, as shown in [Figure 3-5](#).

Figure 3-5. Demonstration Board and Bridge Connection



2. Select **+5 V** as power supply to the board from the Bridge Control Panel and click on the **Toggle Power** button. The LED (red) on the bridge board lights up, as shown.

Figure 3-6. LED (Red) on the Bridge Board



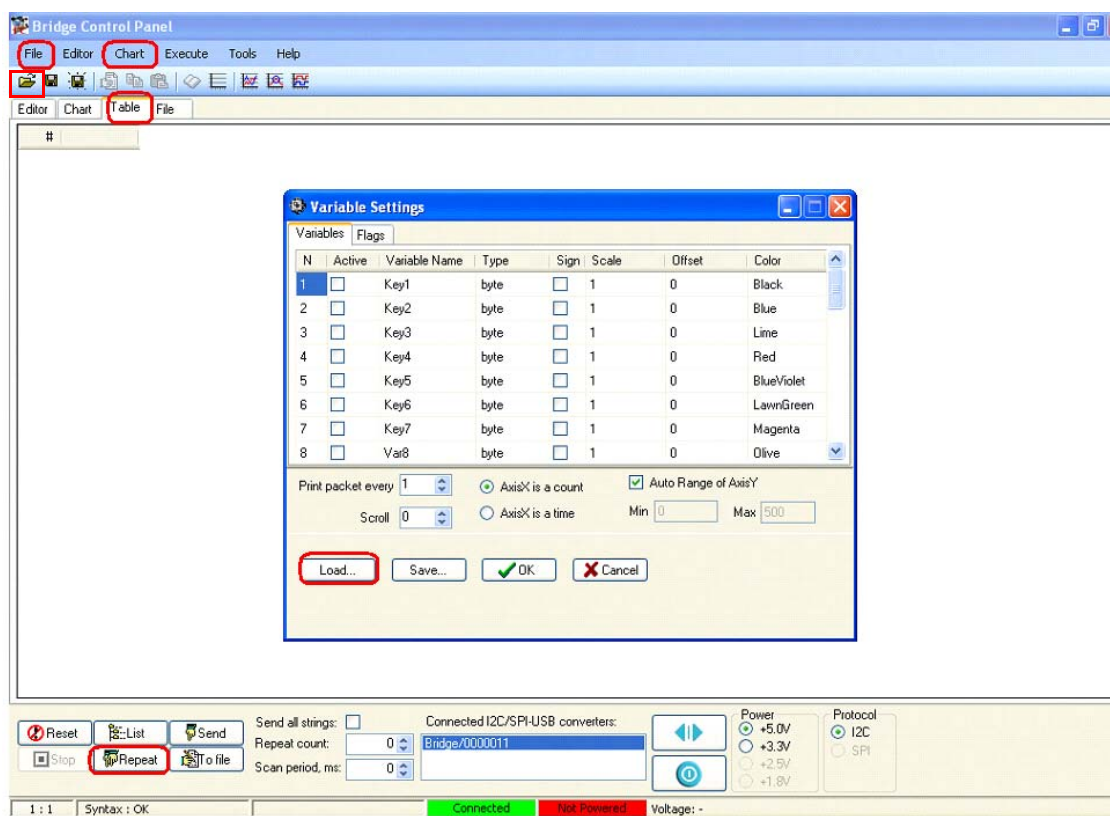
3. Click **List**; the bridge will find the I2C slave demonstration board at 0x00 address.

3.3.1 Run Demonstration Board Test

The demonstration board has a built-in temperature sensor and photo diode. The measurement results of these are sent over I2C.

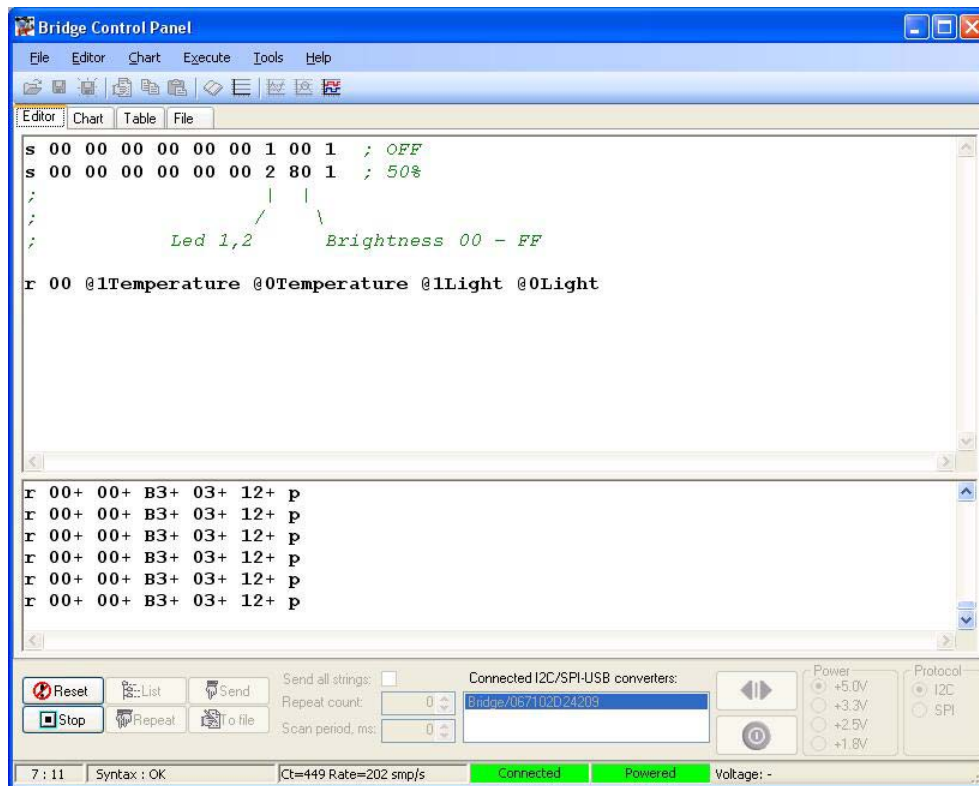
1. Open the **Variable Setting** dialog box from the **Chart** menu.

Figure 3-7. Variable Setting Box



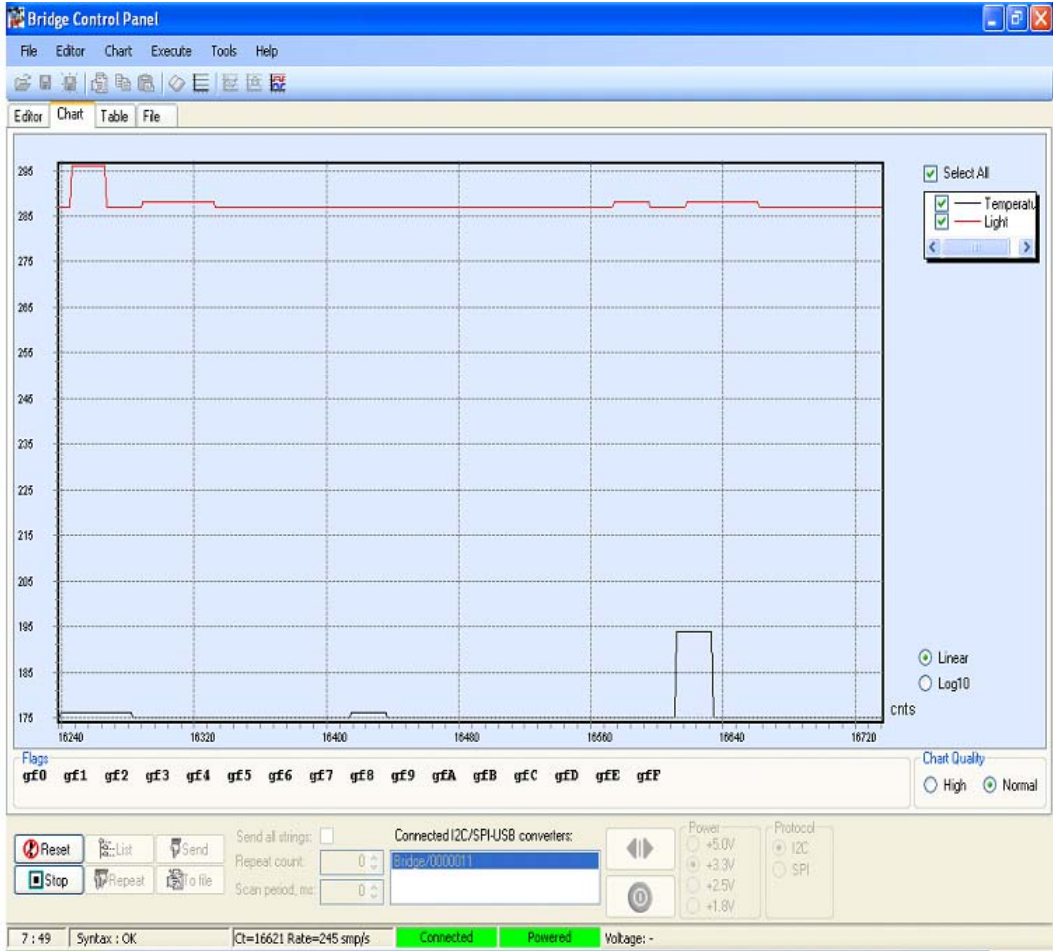
2. The *demo.ini* file initializes light and temperature variables. Load the *demo.ini* file by following these steps:
 - a. Click the **Load** button, as shown in Figure 3-7
 - b. Navigate to <Install_Directory>:\Cypress\CY3240 USB-I2C Bridge Kit\<version> \Firmware\Configuration files
 - c. Select *demo.ini*
 - d. Click **Open**
 - e. Click **OK**
3. The *demo.iic* file contains the iic commands that can be sent to the demo board. Load the *demo.iic* file by following these steps:
 - a. Click on the **Open Commands File** button
 - b. Navigate to <Install_Directory>:\Cypress\CY3240 USB-I2C Bridge Kit\<version> \Firmware\Configuration files
 - c. Select *demo.iic*
 - d. Click **Open**
4. The first two lines in the *demo.iic* file show how to control the LEDs on the board. If the two red LEDs do not light up, click on the **Toggle Power** button. Position the cursor in the first line and press **[Enter]** to send the command. Repeat for the second command line. Observe that, on sending the first command, LED1 turns off. The second command reduces the intensity of LED2 by 50 percent. Experiment with the intensity settings.

Figure 3-8. Bridge Control Panel Editor Screen View



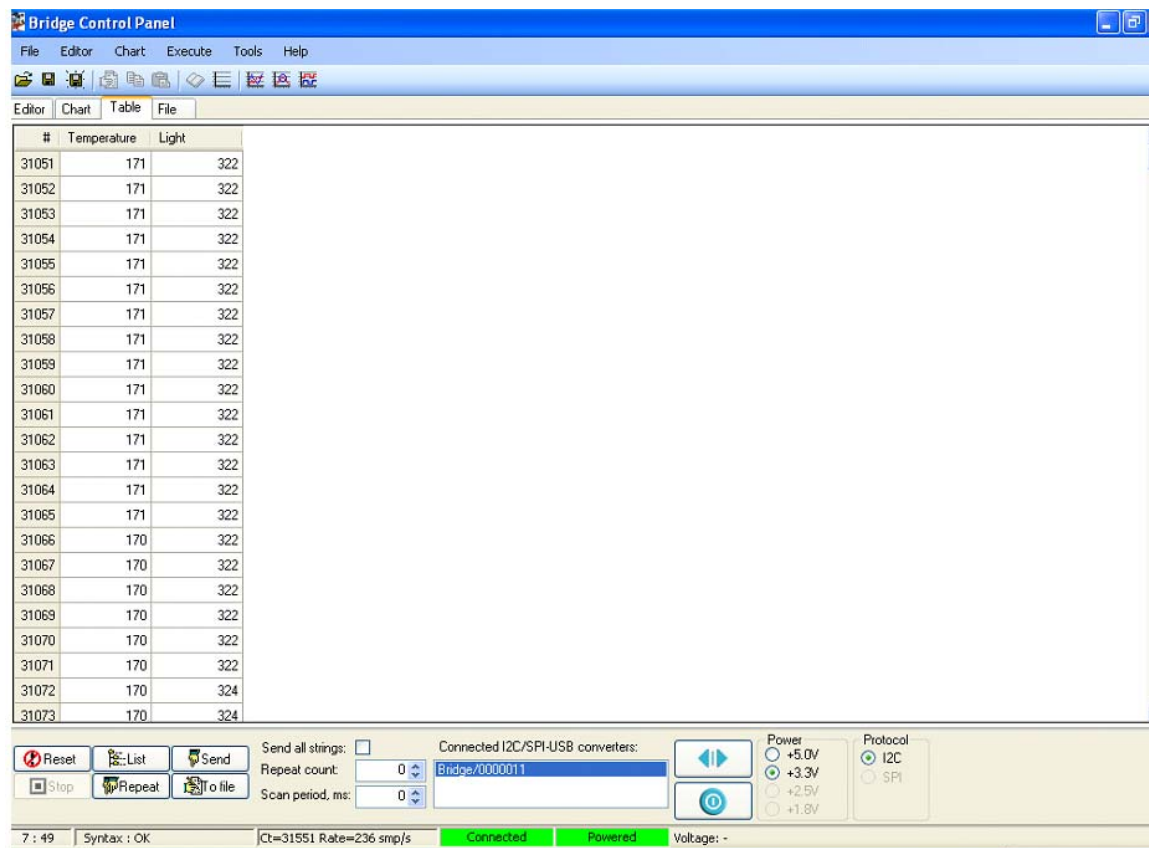
5. Click in the last line, which reads temperature and light data from the device and then click the **Repeat** button. On clicking **Repeat**, the command that is sent last is repeated until **Stop** is clicked. This makes data collection easier. The data received from slave can be viewed either graphically or in a tabular form.
6. Click the **Chart** tab to view the data graphically, as shown in [Figure 3-9](#). Experiment with displaying only temperature or only light.
7. Move your hand over the photo diode on the demo board to vary the light intensity; relative values are reflected on the graph. Similarly, vary the temperature around the board by placing a hot or cold object near the board and observe that the temperature values varies accordingly.

Figure 3-9. Bridge Control Panel Graphical Screen View



8. Click the **Stop** button to stop scanning.
9. Click the **Table** tab to view data in a tabular form, as shown in [Figure 3-10](#).

Figure 3-10. Bridge Control Panel Tabular Screen View



The **File** menu has these options:

- **Save Send Data** - To save data sent to the demo board from the Bridge Control Panel.
- **Save Receive Data** - To save received data shown on the status window of the Bridge Control Panel.

Go to **Bridge Control Panel Help** from the **Help** menu for more information on the Bridge Control Panel and iic command format.

4. Hardware



This section provides an overview of the hardware, including power system, jumper setting, and programming interface. To start using the board, go to [Code Examples on page 33](#).

4.1 System Block Diagram

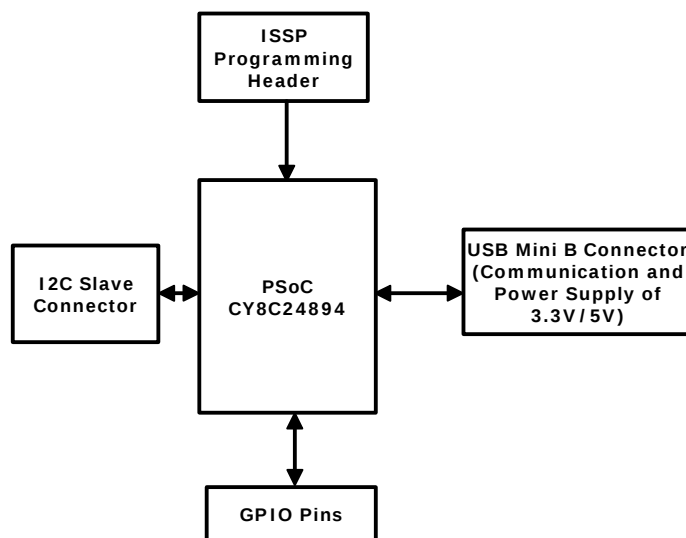
The CY3240 USB-I2C Bridge kit has two boards:

- USB-I2C Bridge board
- Demo board

The USB-I2C Bridge board consists of:

- PSoC CY8C24894 chip
- USB Mini B connector
- ISSP programming header
- GPIO pins
- I2C slave connector

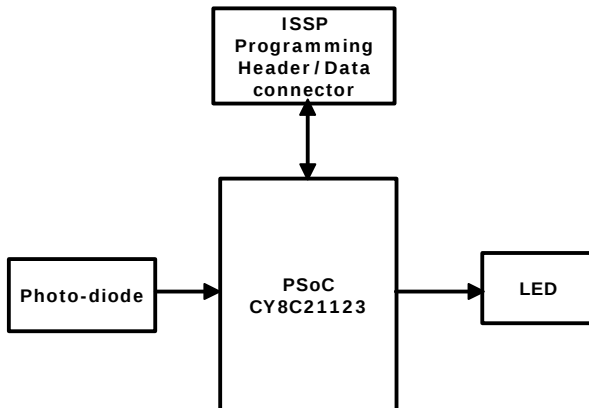
Figure 4-1. USB-I2C Bridge Board



Demo board consists of:

- CY8C21123 chip
- LED
- Photo diode
- ISSP programming header or data connector (with I2C lines)

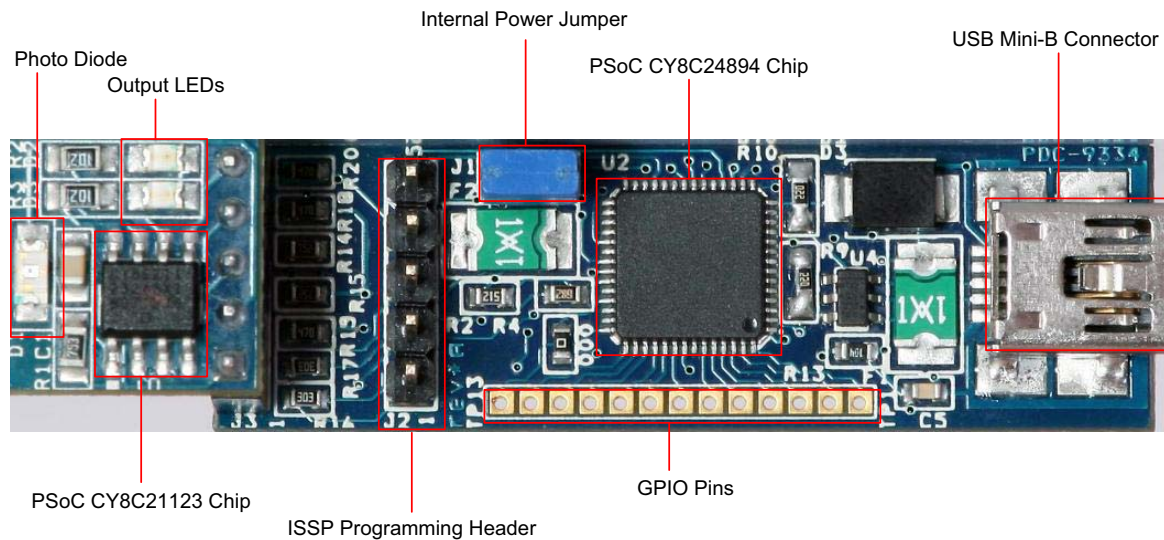
Figure 4-2. Demo Board



4.2 Functional Description

The USB-I2C Bridge is connected to the PC in the same way as an HID device. It requires no additional driver when connected to a PC installed with Windows. This USB-I2C Bridge works as a master in the I2C bus and is controlled by the PC program via USB. In addition, a demonstration PC program is included with the project to demonstrate bridge operation with connected I2C slave.

Figure 4-3. USB-I2C Bridge with Demo Board



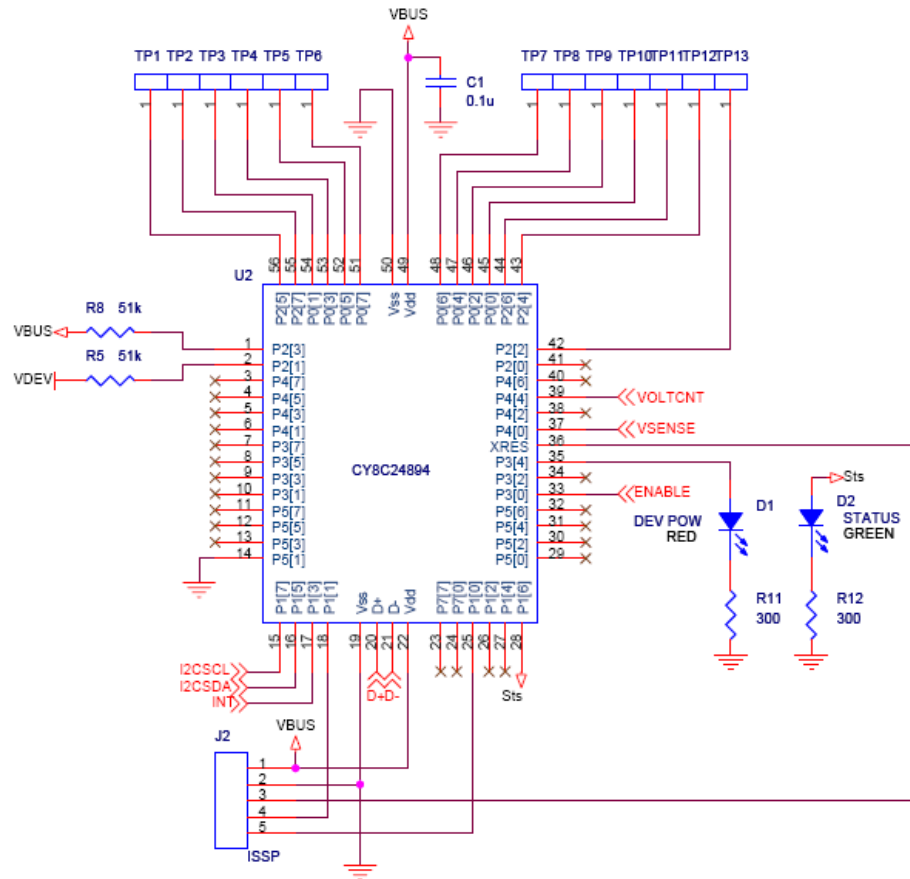
4.2.1 PSoC CY8C24894 Chip

The PSoC CY8C24894 on the CY3240 USB-I2C Bridge board is preprogrammed to function as a full-featured, full-speed (12 Mbps) USB, and an I2C master. This device enables creating customized peripheral (I/O) configurations that match the requirements of individual applications.

The PSoC device also performs the following functions:

- Lights up LED (green) on connecting the bridge to the host via USB
- Lights up LED (red) on external power supply to the target device
- Acts as an interface between the host and the target device

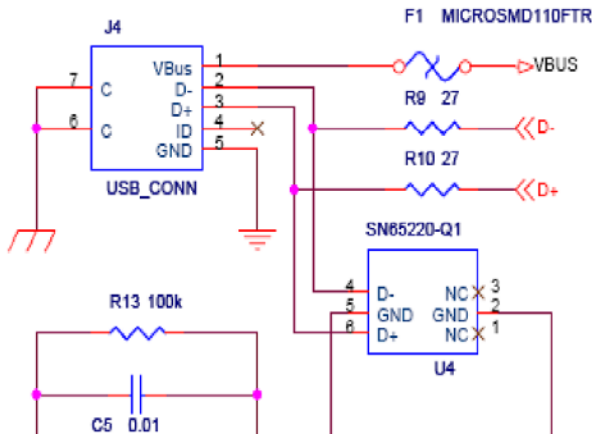
Figure 4-4. PSoC CY8C24894 Hardware Schematic



4.2.2 USB Mini B Connector

USB Mini B connector communicates between the PC and the bridge. It is used to power up the bridge and supply a voltage range of 3.3 V or 5 V to the demo board. These plugs are always oriented downstream towards the USB device. It has SN65220, a single transient voltage suppressor, to provide electrical noise transient protection to the USB port.

Figure 4-5. USB Mini Port Schematic



4.2.3 ISSP Programming Header

The in-system serial programming header is used to:

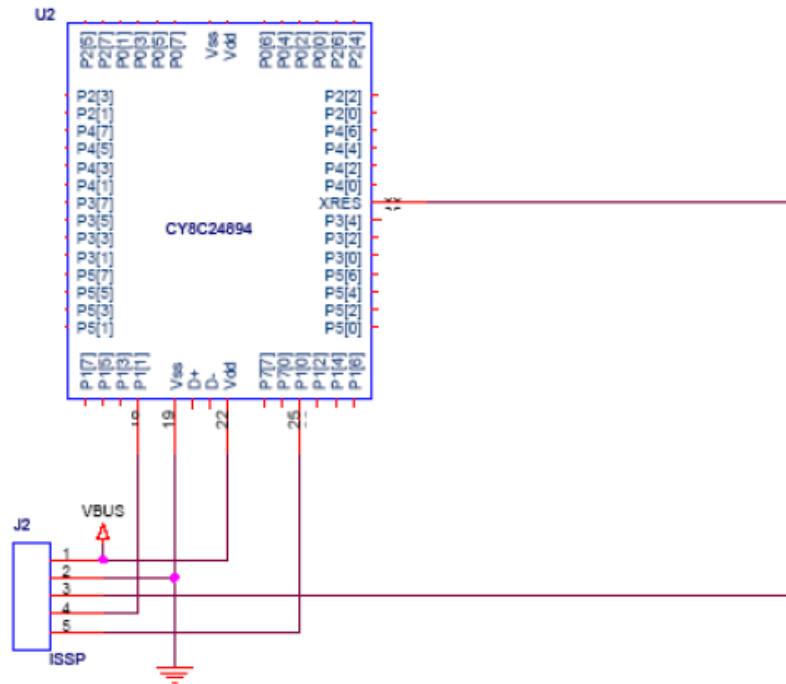
- Burn the hex code onto the chip
- Connect the MiniProg to program the chip

The ISSP connector consists of the following:

Table 4-1. ISSP Header Connection Details

Pin No	Connection	Description
1	+V Device	To supply voltage to the bridge
2	GND	Ground pin of the bridge
3	INT	Interrupt pin
4	SCL	Serial clock line
5	SDA	Serial data line

Figure 4-6. ISSP Header Schematic



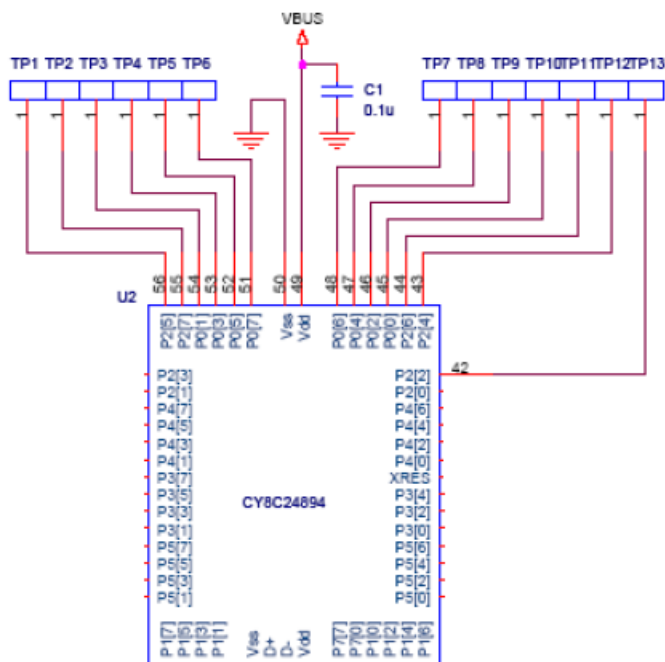
4.2.4 GPIO Pins

GPIO plays an important role in customizing the applications.

Table 4-2. GPIO Pin Connectivity Table

Terminal Connection	Port	Pin No.	Description
TP12	P2[4]	43	External analog ground (AGND) input
TP11	P2[6]	44	External voltage reference (VREF) input
TP10	P0[0]	45	Analog column mux input
TP9	P0[2]	46	Analog column mux input
TP8	P0[4]	47	Analog column mux input
TP7	P0[6]	48	Analog column mux input
TP6	P0[7]	51	Analog column mux input
TP5	P0[5]	52	Analog column mux input and column output
TP4	P0[3]	53	Analog column mux input and column output
TP3	P0[1]	54	Analog column mux input
TP2	P2[7]	55	GPIO
TP1	P2[5]	56	GPIO
TP13	P2[2]	42	GPIO

Figure 4-7. GPIO Pin Schematic



4.2.5 I2C Slave Interface Connector

The I2C slave interface connector is used to communicate data between the target device and the bridge board. It consists of two devices, MAX3378 and SN721.

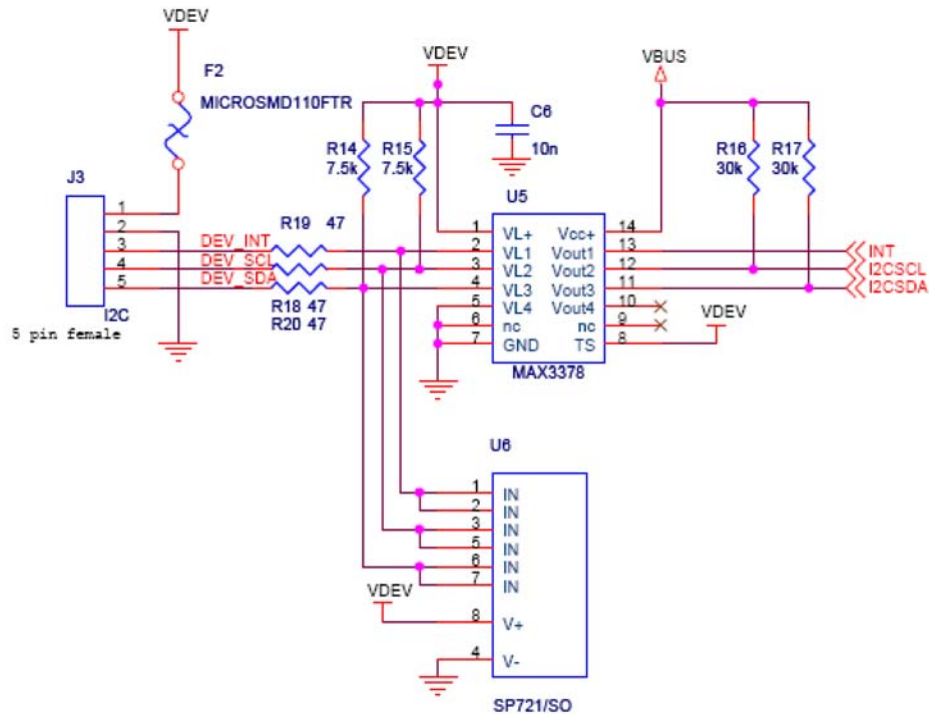
MAX3378

- Is used to convert voltage level between the target device (V_{CC}) and the host level voltage (V_L)
- Has bidirectional level translation, accepts $V_L + 1.2$ V (Min) to +5.5 V (Max); V_{out} (High) is 0.6 times V_L (if source current is 0.02 mA) and V_{out} (Low) is 0.4 V

SN721

- An array of SCR/diode bipolar structure for ESD and over-voltage protection

Figure 4-8. I2C Slave Interface Schematic



4.2.6 Demo Board

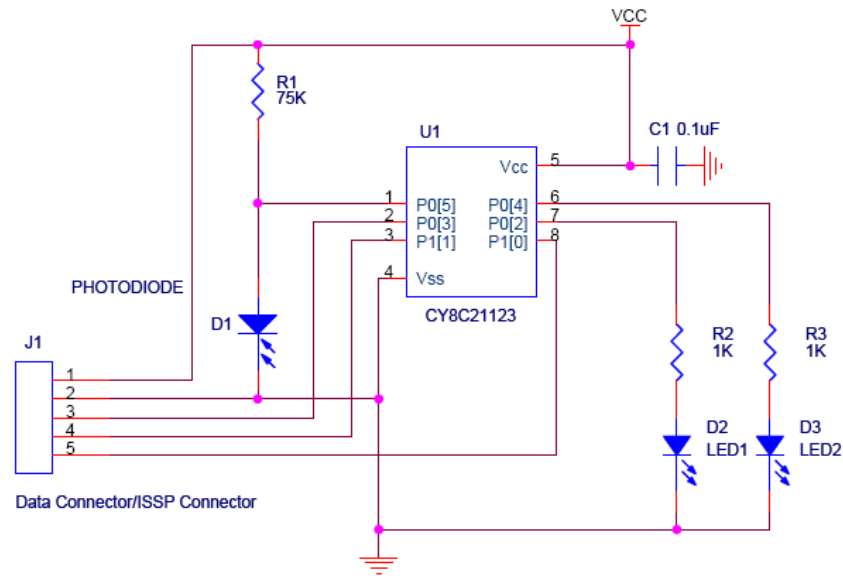
The demo board lights the two LEDs upon receiving data from the host and transfers the data, acquired from the photo diode to the host.

The PSoC device (CY8C21123) is used to control the two LEDs based on input from the host using the Bridge Control Panel. The PSoC device updates the I2C register with the sensor information, which can be read by any I2C master.

Table 4-3. CY8C21123 Pin Connectivity

Pin No.	Port No	Description
6	P0[4]	LED1
7	P0[2]	LED2
1	P0[5]	Photo-diode
8	P1[0]	Voltage supply
3	P1[1]	Ground
2	P0[3]	Interrupt
5	Vcc	Serial clock
4	GND	Serial data

Figure 4-9. Demo Board Chip Schematic



5. Code Examples



All code examples are available in the CY3240 USB-I2C Bridge kit CD or at the following location:
<Install_Directory>:\Cypress\CY3240 USB-I2C Bridge Kit\<version>\Firmware.

5.1 My First Code Example

5.1.1 Project Objective

This code example demonstrates data transfer between the demo board and the host using I2C communication. The demo board outputs temperature and light data to the host whenever the host requests this data. Similarly, the host board sends commands to the demo board to control the LEDs on the board.

The demo board has a photo diode to measure the light intensity. The voltage value analogous to the die temperature is available on a PSoC analog block array. This temperature input can be used in applications such as battery chargers and data acquisition without any external components. In this code example, this temperature input is used to read the die temperature value. See the Technical Reference Manual available in **Help > Documentation** for more details.

The following user modules are used in the design to achieve the objective of this project:

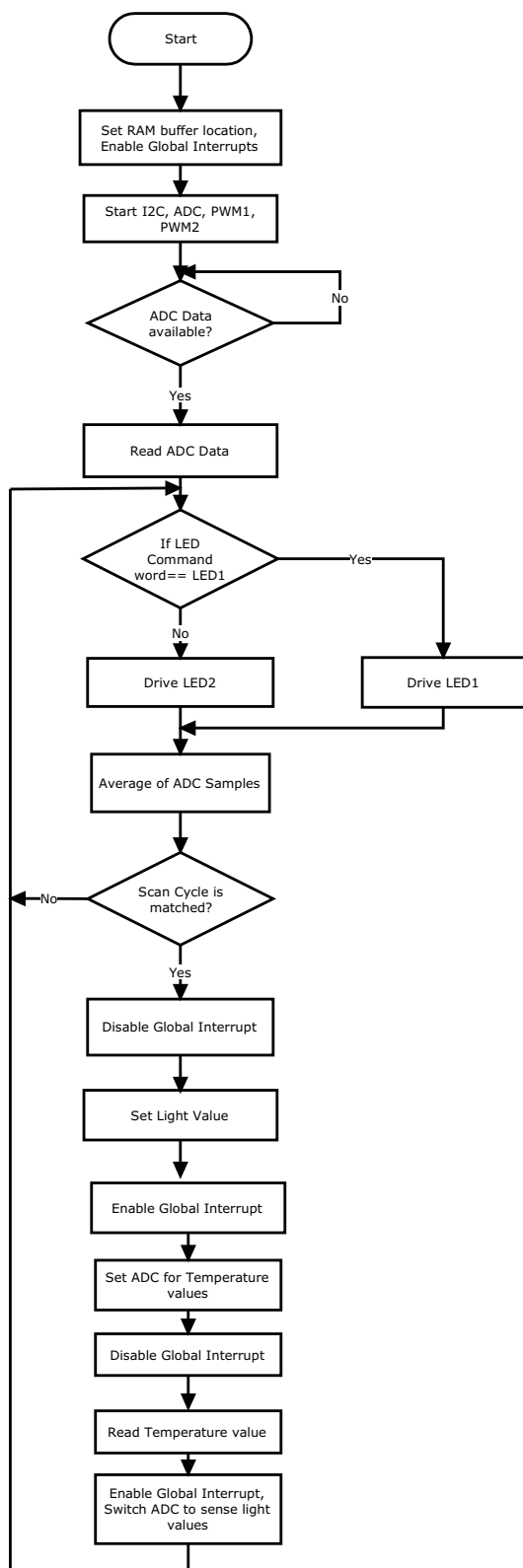
ADC10: Used to obtain the digital values for light intensity and temperature. Input to the ADC module is switched between light input and temperature input once every 551 scan cycles.

EzI2C: Used to configure the PSoC on the demo board as I2C slave and to transfer data to the bridge board where the PSoC is configured as I2C master.

PWM8: Used to handle the LED command from the host and vary the LED intensity accordingly.

When data is requested by the I2C master, the ADC values analogous to the die temperature and light intensity incident on the board are output.

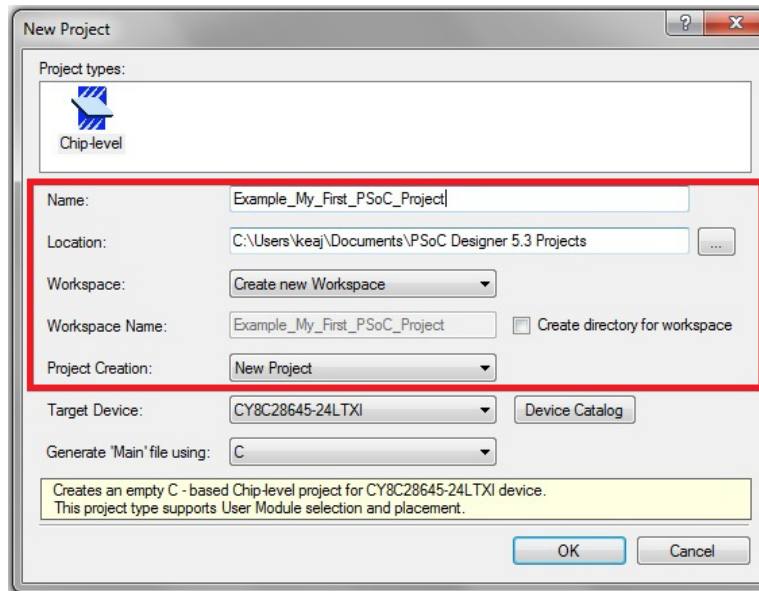
5.1.2 USB-I2C Demo Board Flowchart



5.1.3 Creating My First PSoC 1 Project

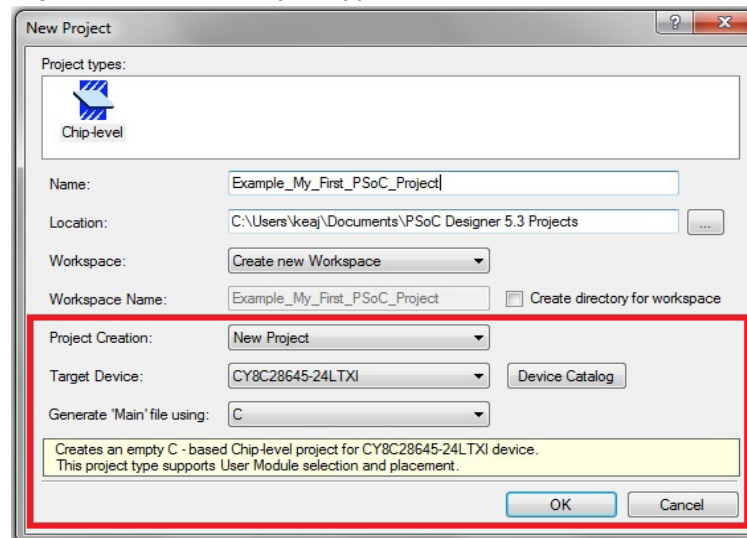
1. Open PSoC Designer.
2. To create a new project, click **File > New Project**.
3. In the New Project window, select the chip-level icon. Name the project **Example_My_First_PSoC_Project**, as shown in [Figure 5-1](#).
4. Click **Browse** to navigate to the directory in which the project is created.

Figure 5-1. New Project Window



5. In the project creation drop down menu select new project.
6. In the Select Target Device option, click **Device Catalog**, as shown in [Figure 5-2](#).

Figure 5-2. Select Project Type



7. The Device Catalog window opens and then scroll down to the **CY8C21123, CY8C21223, CY8C21323** section.

8. For this project, click **CY8C21123-24SXI**; then click **Select**.

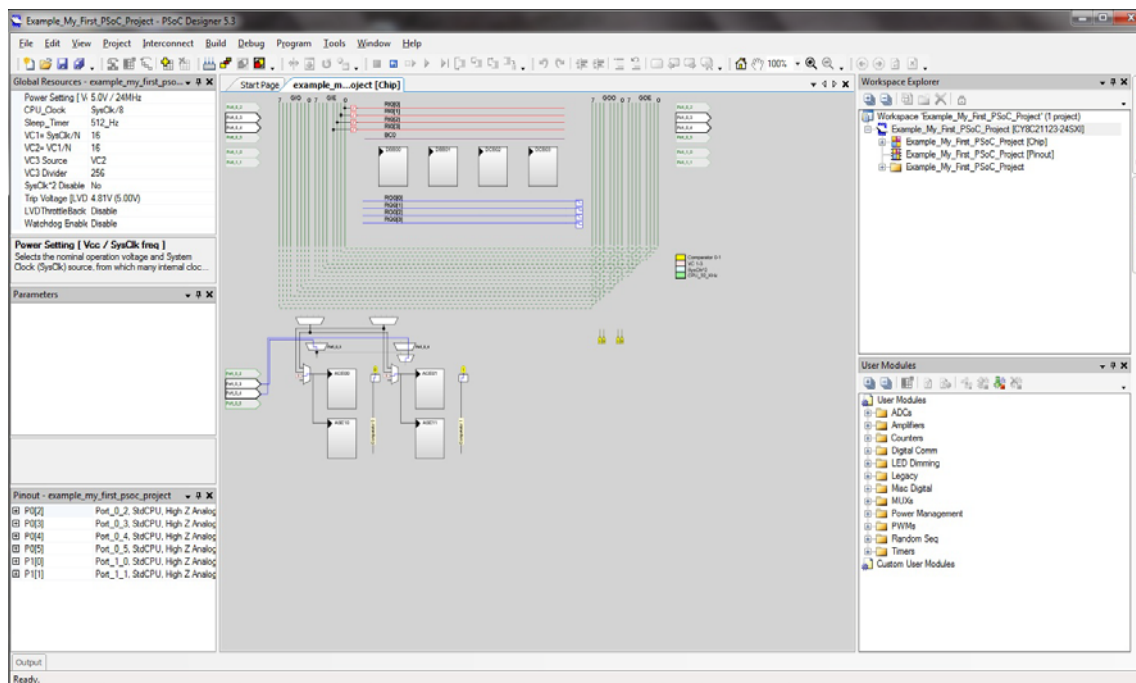
Figure 5-3. Device Catalog Window

Device Catalog - Chip-level														
Device Type: All Devices		Compare Devices		Reset		Find...		User Modules		Part Image		Favorites		
Compare	Part Number	Pin Count	Package Type	Analog Blocks	Digital Blocks	CapSense	Flash	RAM	IO Count	Supply Voltage	SMP	USB Interface	Wireless Interface	Temperature
	Filters:													
☐	CY8C20566-24PVXI	48	SSOP	1	0	Yes	32K	2K	36	1.71 to 5.5	N/A	N/A	N/A	Ind(-40 to 85C)
☐	CY8C20636-24LTXI	48	QFN	1	0	Yes	8K	1K	36	1.71 to 5.5	N/A	N/A	N/A	Ind(-40 to 85C)
☐	CY8C20646-24LTXI	48	QFN	1	0	Yes	16K	2K	36	1.71 to 5.5	N/A	Full-Speed	N/A	Ind(-40 to 85C)
☐	CY8C20666-24LTXI	48	QFN	1	0	Yes	32K	2K	36	1.71 to 5.5	N/A	Full-Speed	N/A	Ind(-40 to 85C)
☐	CY8C21312-24PVXA	20	SSOP	1	1	Yes	8K	512	16	3.0 to 5.25	N/A	N/A	N/A	Auto(-40 to 85C)
☐	CY8C21512-24PVXA	28	SSOP	1	1	Yes	8K	512	24	3.0 to 5.25	N/A	N/A	N/A	Auto(-40 to 85C)
☐	CY8C21312-12PVXE	20	SSOP	1	1	Yes	8K	512	16	4.75 to 5.25	N/A	N/A	N/A	Auto(-40 to 125C)
☐	CY8C21512-12PVXE	28	SSOP	1	1	Yes	8K	512	24	4.75 to 5.25	N/A	N/A	N/A	Auto(-40 to 125C)
☒	CY8C21123-24SXI	8	SOIC	0 + *4	4	No	4K	256	6	2.4 to 5.25	N/A	N/A	N/A	Ind(-40 to 85C)
☐	CY8C21223-24SXI	16	SOIC	0 + *4	4	No	4K	256	12	2.4 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)
☐	CY8C21223-24LGXI	16	COL	0 + *4	4	No	4K	256	12	2.4 to 5.25	N/A	N/A	N/A	Ind(-40 to 85C)
☐	CY8C21323-24PVXI	20	SSOP	0 + *4	4	No	4K	256	16	2.4 to 5.25	N/A	N/A	N/A	Ind(-40 to 85C)
☐	CY8C21323-24LFXI	24	MLF	0 + *4	4	No	4K	256	16	2.4 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)
☐	CY8C21234-24SXI	16	SOIC	0 + *4	4	Yes	8K	512	12	2.4 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)
☐	CY8C21334-24PVXI	20	SSOP	0 + *4	4	Yes	8K	512	16	2.4 to 5.25	N/A	N/A	N/A	Ind(-40 to 85C)
☐	CY8C21534-24PVXI	28	SSOP	0 + *4	4	Yes	8K	512	24	2.4 to 5.25	N/A	N/A	N/A	Ind(-40 to 85C)
☐	CY8C21434-24LFXI	32	MLF	0 + *4	4	Yes	8K	512	28	2.4 to 5.25	N/A	N/A	N/A	Ind(-40 to 85C)
☐	CY8C21434-24LTXI	32	MLF	0 + *4	4	Yes	8K	512	28	2.4 to 5.25	N/A	N/A	N/A	Ind(-40 to 85C)
☐	CY8C21434-24LQXI	32	MLF	0 + *4	4	Yes	8K	512	28	2.4 to 5.25	N/A	N/A	N/A	Ind(-40 to 85C)
☐	CY8C21634-24LFXI	32	MLF	0 + *4	4	Yes	8K	512	26	2.4 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)
☐	CY8C21634-24LTXI	32	MLF	0 + *4	4	Yes	8K	512	26	2.4 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)
Selected Device: CY8C21123-24SXI Filters applied: 0 Devices found: 353 of 353														
												Create Project with 'CY8C21123-24SXI'		
												Cancel		

9. In the **Generate 'Main' File Using:** option, select **C**, then click **OK**.

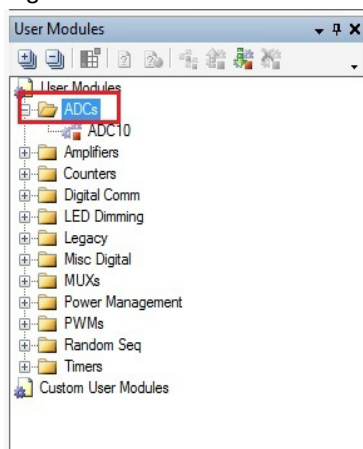
10. By default, the project opens in Chip view, as shown in [Figure 5-4](#).

Figure 5-4. Default View



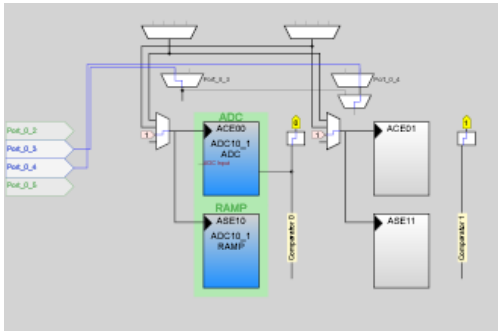
11. Configure the modules required for this design. Also, connect the modules together and to the pins on the PSoC. In the **User Modules** section, expand the **ADCs** folder.

Figure 5-5. User Modules Window



12. In this folder, right-click on **ADC10** and select **Place**. The user module (UM) is placed, as shown in the following figure.

Figure 5-6. Default ADC User Module Placement



13. The **ADC** UM is placed in **ACE00** and **ASE10** of the analog block. Click on **Next Allowable Placement** (Figure 5-7) to change the ADC module from **ACE00** to **ACE01**. Click on the **Place User Module** icon (Figure 5-7). Similarly, change the ADC module in the digital block from **DBB00** to **DCB02**, as shown in Figure 5-8.

Figure 5-7. PSoC Designer Toolbar

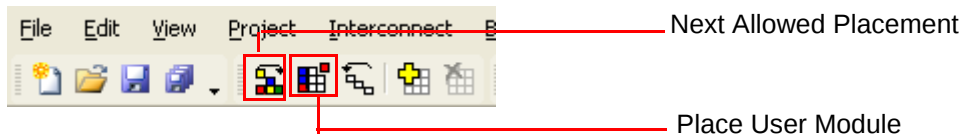
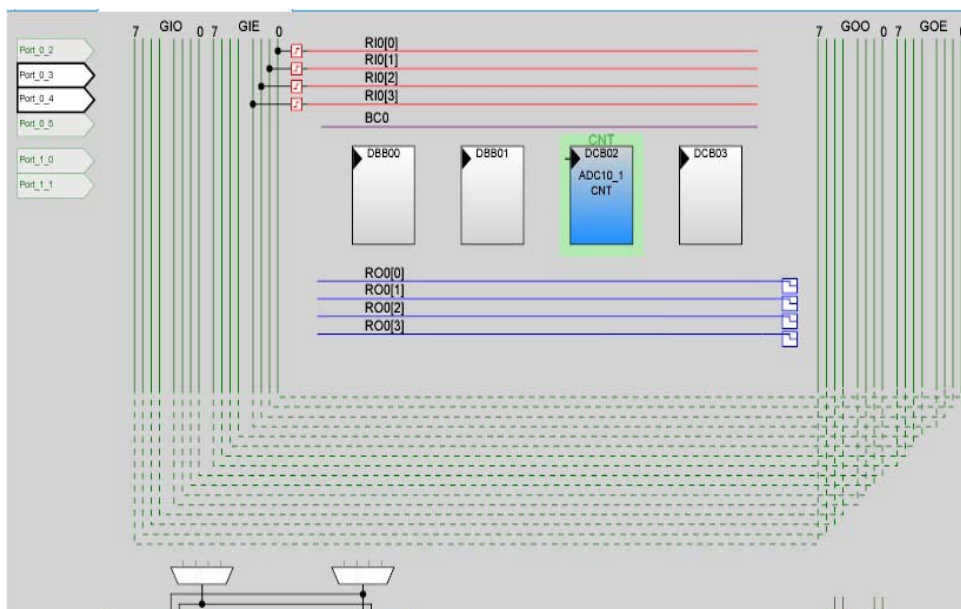


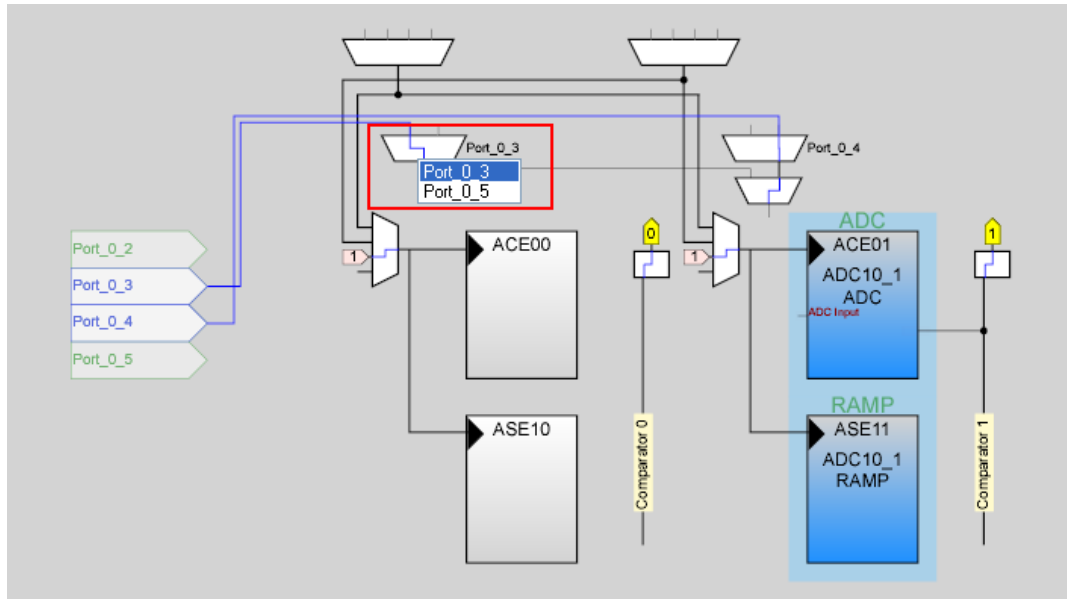
Figure 5-8. ADC10 User Module Placement



Note A temperature-sensitive voltage, derived from bandgap sensing on the die, is buffered and available as an analog input into the continuous time block ACE01. So, the ADC module placement is changed to ACE01 block.

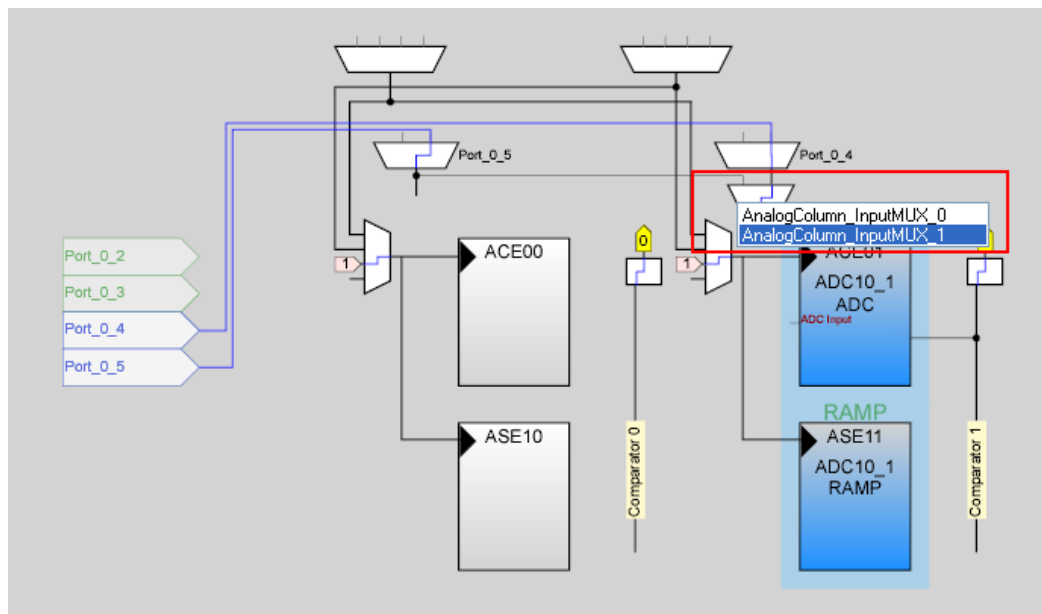
14. Click on **AnalogColumn_InputMUX_0** to change from **Port_0_3** to **Port_0_5**.

Figure 5-9. Change Port Connectivity to ADC Module



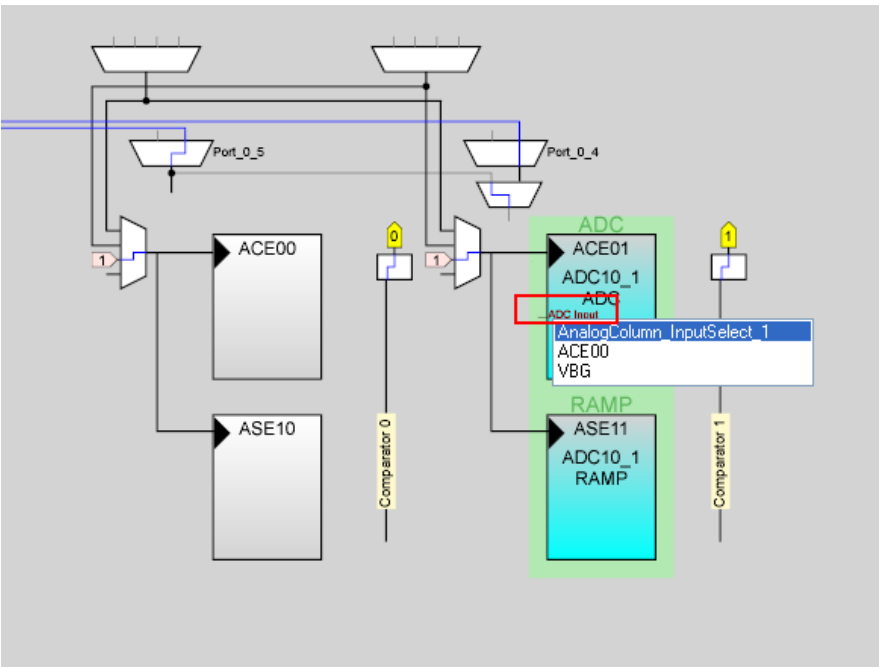
15. Click on **AnalogColumn_InputSelect_1** to select **AnaloColumn_InputMUX_0**.

Figure 5-10. Select Input Mux to ADC



16. Select **Analog ADC Module** and click on **ADC Input** to select **AnalogColumn_InputSelect_1** to connect the module to **Port_0_5**.

Figure 5-11. Select Port_0_5 to Compare ADC



17. Configure the ADC10_1 properties, as shown in Figure 5-12.

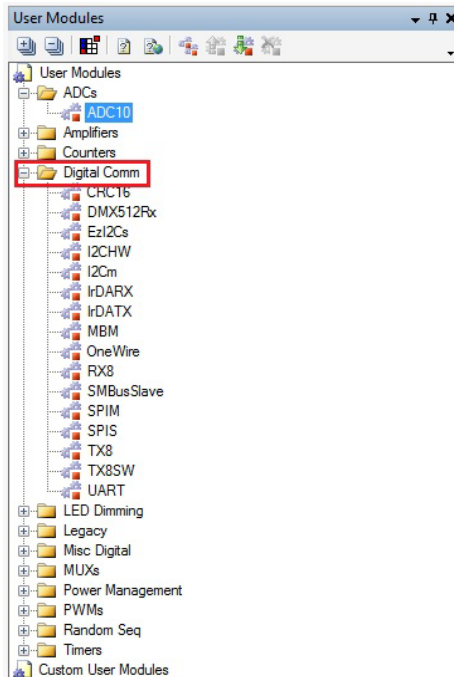
Figure 5-12. ADC10 User Module Properties

Parameters - ADC10_1	
Name	ADC10_1
User Module	ADC10
Version	1.20
ADC Input	AnalogColumn_InputSelect_1
Data Clock	VC1
Current Setting	Normal Current
PWM_High	16 VC3 Periods
PWM_Low	1 VC3 Period
Die Temp Measurem	Enabled

Die Temp Measurement
Defines whether to use the existing ADC capability to measure the VTEMP voltage and convert the voltag...

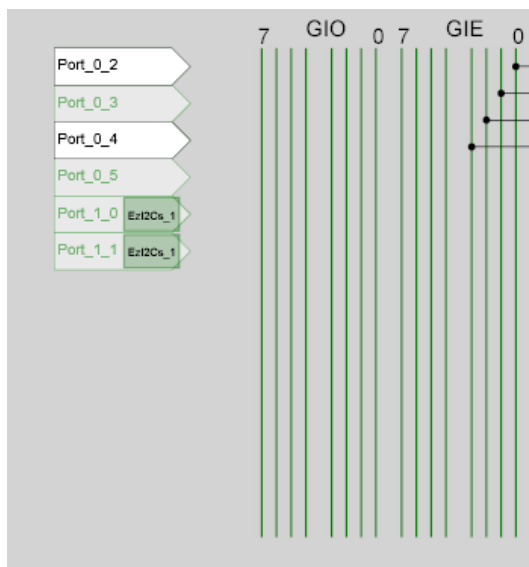
18. In the User Modules window, expand the **Digital Comm** folder and double-click on **EzI2Cs** to place an EzI2C in the design.

Figure 5-13. User Module Window



19. By default, the EzI2C is placed at **Port_1_0** and **Port_1_1**, as shown in [Figure 5-14](#).

Figure 5-14. EzI2C User Module Placement



20. Default properties of EzI2C_1 is shown in the following figure.

Figure 5-15. Default EzI2C_1 Parameters

Parameters - EzI2Cs_1	
Name	EzI2Cs_1
User Module	EzI2Cs
Version	1.30
Slave_Addr	0
Address_Type	Static
ROM_Registe	Disable
I2C Clock	
I2C Pin	
User Module	
User Module name.	

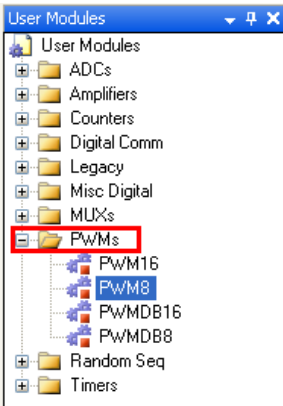
21. Configure the properties of EzI2Cs_1 and rename as **EzI2Cs** to build the project successfully.

Figure 5-16. EzI2C User Module Properties

Parameters - EzI2Cs	
Name	EzI2Cs
User Module	EzI2Cs
Version	1.2
Slave_Addr	0
Address_Type	Static
ROM_Registers	Disable
I2C Clock	400K Fast
I2C Pin	P[1]0-P[1]1
Name	
Indicates the name used to identify this User Module instance	

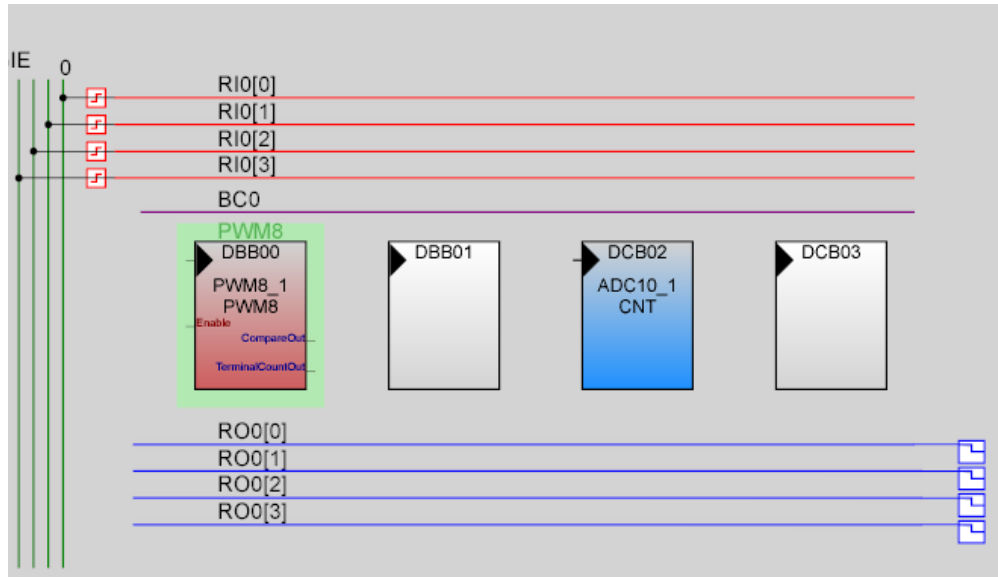
22. In the User Modules window, expand the **PWMs** folder and double-click on **PWM8** to place a PWM8 in the design.

Figure 5-17. User Module Window



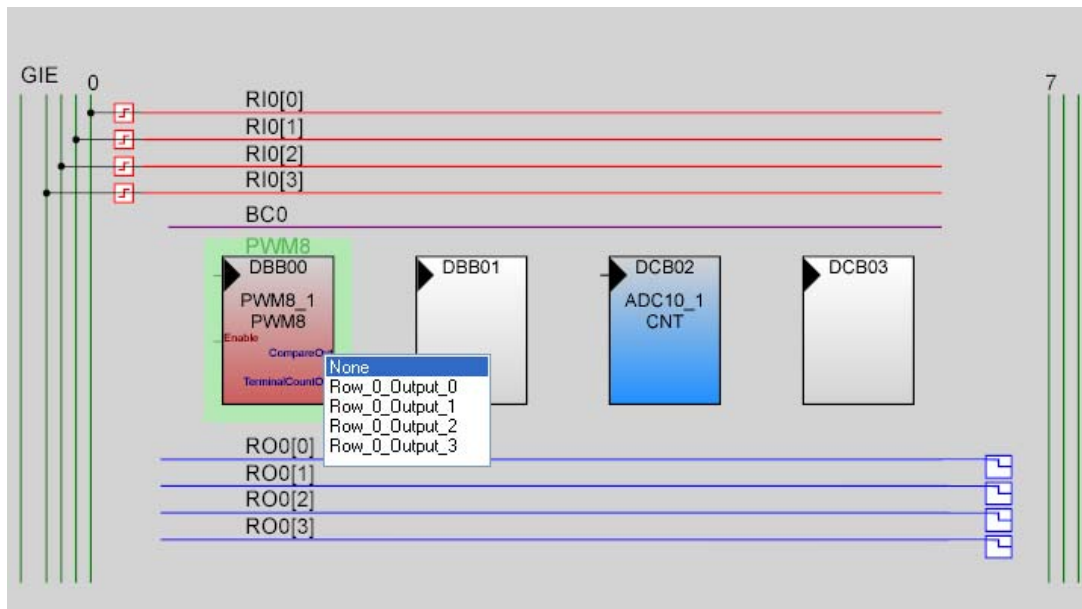
23. By default, the PWM8 is placed as shown in the following figure.

Figure 5-18. PWM Module Placement



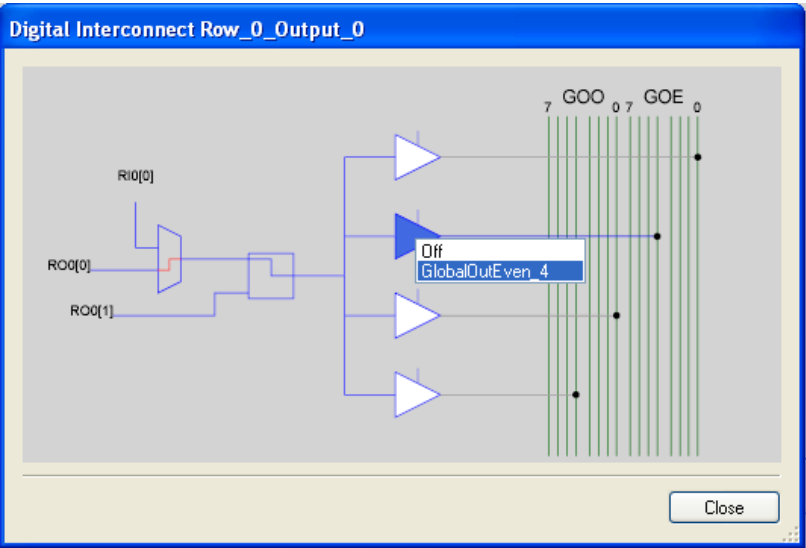
24. In the PWM module, to connect **Compare Out** to P0[4], configure the lookup table (LUT) on Row_0_Output_0 to **GlobalOut (GOE[4])** bus. To do this, click on **Compare Out** of PWM8_1 Module to select the Row_0_Output_0, as shown in [Figure 5-19](#).

Figure 5-19. Connect PWM8 to Output Line



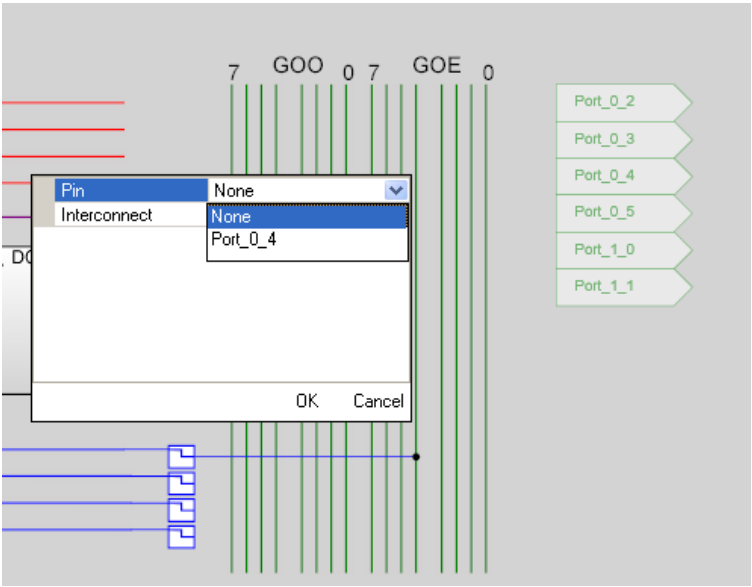
25. Click on **Row_0_Output_0** bus to open the Digital Interconnect window; configure the bus to **GlobalOutEven_4**, as shown in Figure 5-20.

Figure 5-20. Select Global Output Line for Row_0_Ouput_0



26. Click on **GlobalOutEven_4** bus to connect it output pin Port_0_4.

Figure 5-21. Select Global Output Even Line for Port Selection



27. Configure parameters of **PWM8_1**.

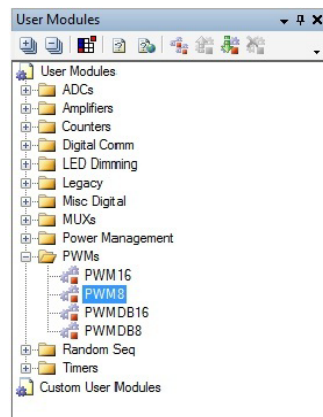
Figure 5-22. Parameter Window of PWM8_1

Parameters - PWM8_1	
Name	Pwm8_1
User Module	PWM8
Version	2.60
Clock	VC1
Enable	High
CompareOut	Row_0_Output_0
TerminalCountOut	None
Period	255
PulseWidth	255
CompareType	Less Than
InterruptType	Terminal Count
ClockSync	Sync to SysClk
InvertEnable	Normal

Name
Indicates the name used to identify this User Module instance

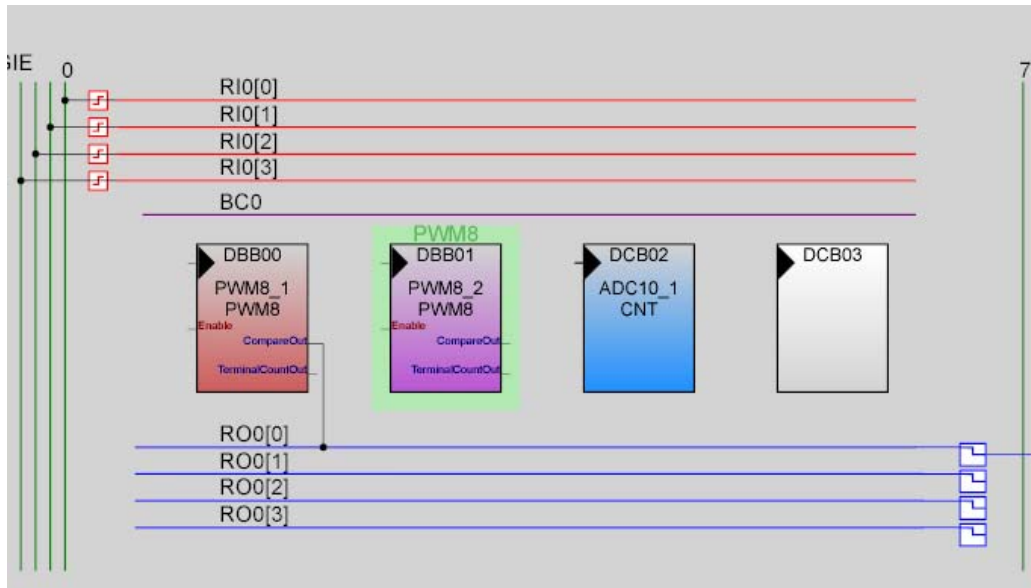
28. In the User Modules window, expand the **PWMs** folder and double-click on **PWM8** to place a PWM8 in the design.

Figure 5-23. User Module Window



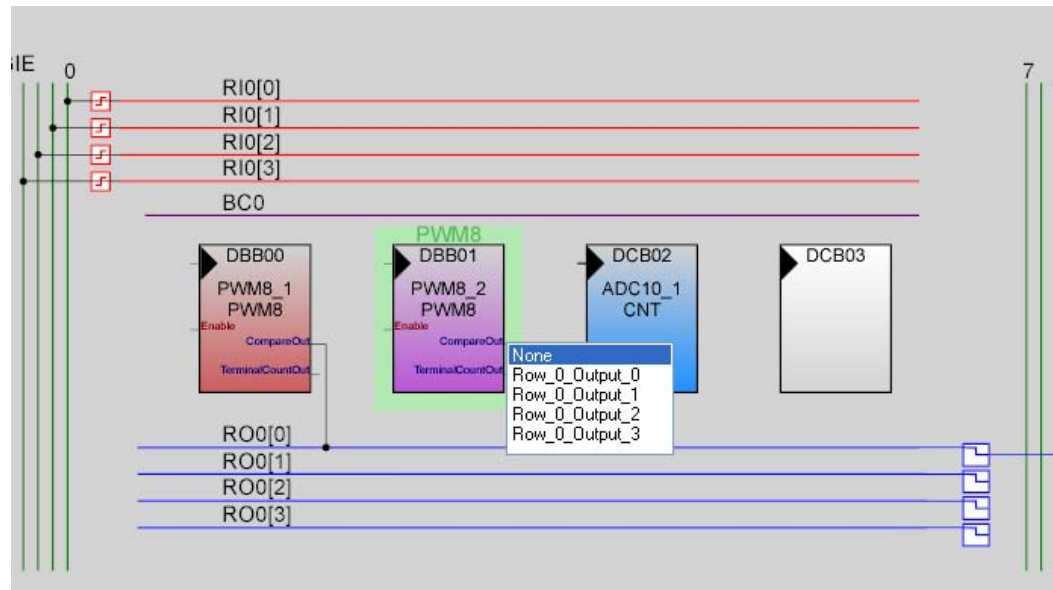
29. By default, the PWM8_2 is placed, as shown in [Figure 5-24](#).

Figure 5-24. Placement of PWM8_2 User Module



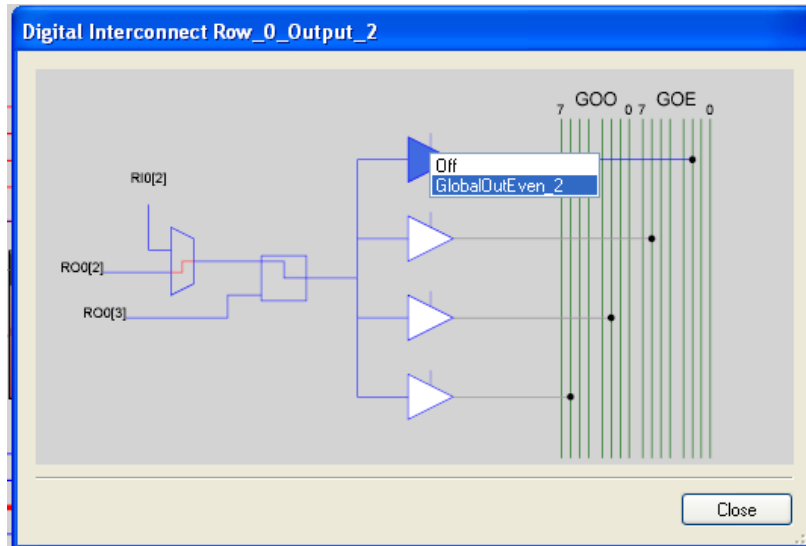
30. In the PWM Module, to connect the Compare Out to P0[2], configure the LUT on Row_0_Output_2 to **GlobalOut Even (GOE[2])** bus. To do this, click on **Compare Out** of PWM8_2 Module to select the **Row_0_Output_2**, as shown in [Figure 5-25](#).

Figure 5-25. Connect PWM8_2 to Row_0_Output_2 Bus



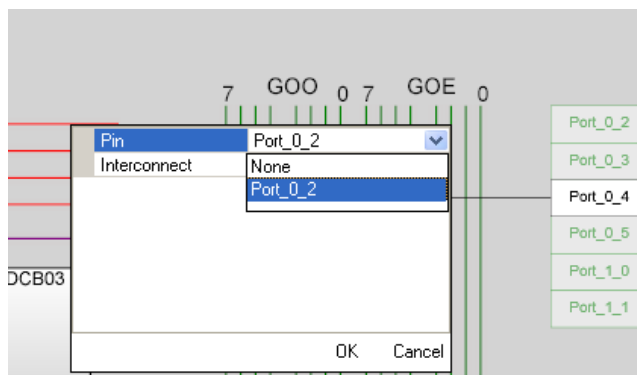
31. Click on **Row_0_Output_2** bus to open the Digital Interconnect window; configure the bus to **GlobalOutEven_2**.

Figure 5-26. Select Global Output Line for Row_0_Output_2 Bus



32. Select **GlobalOutEven_2** bus line to configure Output Pin to Port_0_2.

Figure 5-27. Select GlobalOutputEvenline for Port Selection



33. Configure the parameters of PWM8_2.

Figure 5-28. Parameter Window of PWM8_2

Parameters - PWM8_2	
Name	Pwm8_2
User Module	Pwm8
Version	2.60
Clock	VC1
Enable	High
CompareOut	Row_0_Output_2
TerminalCountOut	None
Period	255
PulseWidth	255
CompareType	Less Than
InterruptType	Compare True
ClockSync	Sync to SysClk
InvertEnable	Normal

Name
Indicates the name used to identify this User Module instance

34. Configure the Global Resources window to match the following figure.

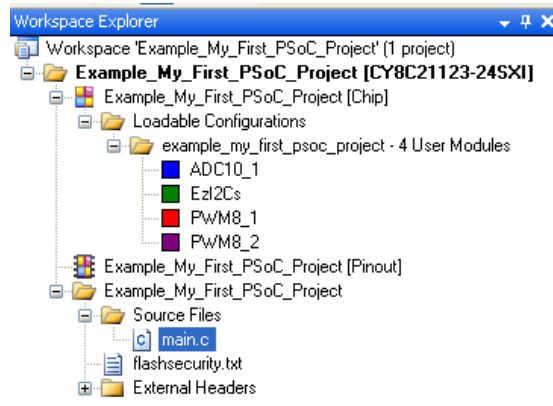
Figure 5-29. Global Resources Window

Global Resources - example_my_first_psoc_project	
Power Setting [Vcc / SysClk freq]	5.0V / 24MHz
CPU_Clock	SysClk/2
Sleep_Timer	512_Hz
VC1= SysClk/N	6
VC2= VC1/N	1
VC3 Source	VC1
VC3 Divider	64
SysClk*2 Disable	Yes
Trip Voltage [LVD (SMP)]	4.81V (5.00V)
LVDThrottleBack	Disable
Watchdog Enable	Disable

Power Setting [Vcc / SysClk freq]
Selects the nominal operation voltage and System Clock (SysClk) source, from which many internal clocks (V1, V2, V3, and CPU clocks) are derived.
Registers Affected:
CPU_SCR1fVBG_TRIMMO_TR

35. Open the existing *main.c* file in Workspace Explorer. Replace the existing *main.c* content with the content of the embedded *Example_My_First_PSoC_Project_Main.c* file, which is attached to this document.

Figure 5-30. Workspace Explorer Window



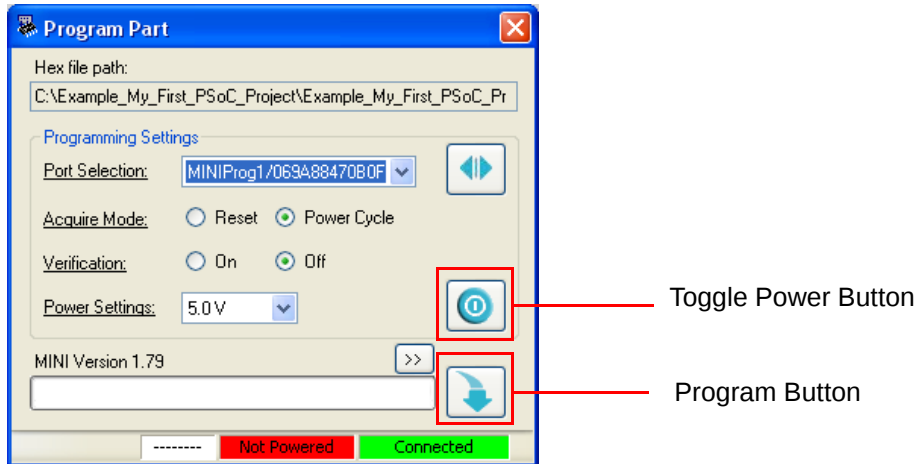
36. Save the project.
37. To build the project, click **Build >Generate/Build 'Example_My_First_PSoC_Project'**.
38. Connect the CY3240 demo board to the PC through a MiniProg1 and program the board.
39. The board can be programmed through PSoC Designer IDE or by launching PSoC Programmer.

Note To program the board using PSoC Programmer, see [PSoC Programmer on page 14](#).
40. To program the board through PSoC Designer follows these steps.
 - a. Click on **Program > Program Part**.

Note When programming the board through PSoC Designer, close any open instance of PSoC Programmer.
 - b. In the Program Part window, set the following:
 - In Port Selection drop-down box, select **MiniProg1/<MiniProg_Number>**; ensure it is **Connected**.
 - Acquire Mode: **Power Cycle**
 - Verification: **Off**
 - Power Settings: **5.0 V**

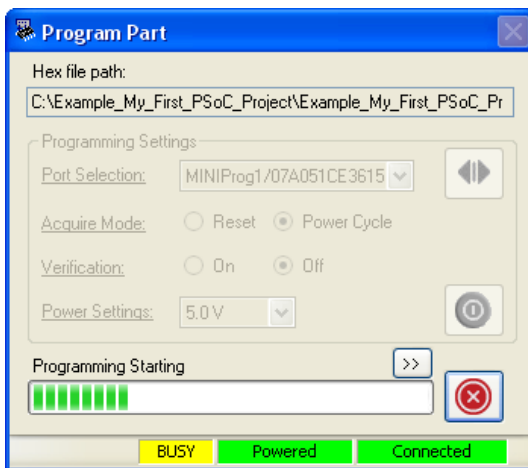
41. Click on the **Program** button to start programming the board.

Figure 5-31. Program Part Window



42. The programming of the board starts and the status can be observed on the progress bar.

Figure 5-32. Programming Status



43. When the programming is done successfully, the **Operation Succeeded!** message is displayed.

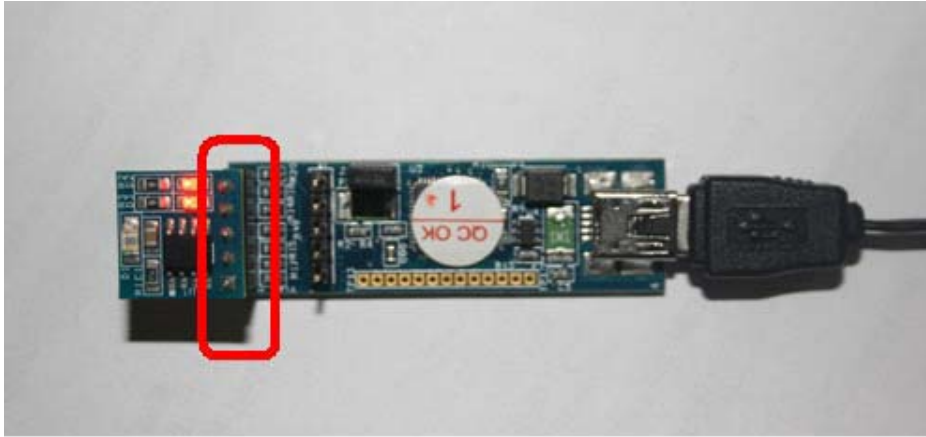
Figure 5-33. Operation Succeeded! Message



5.1.4 Verify Output

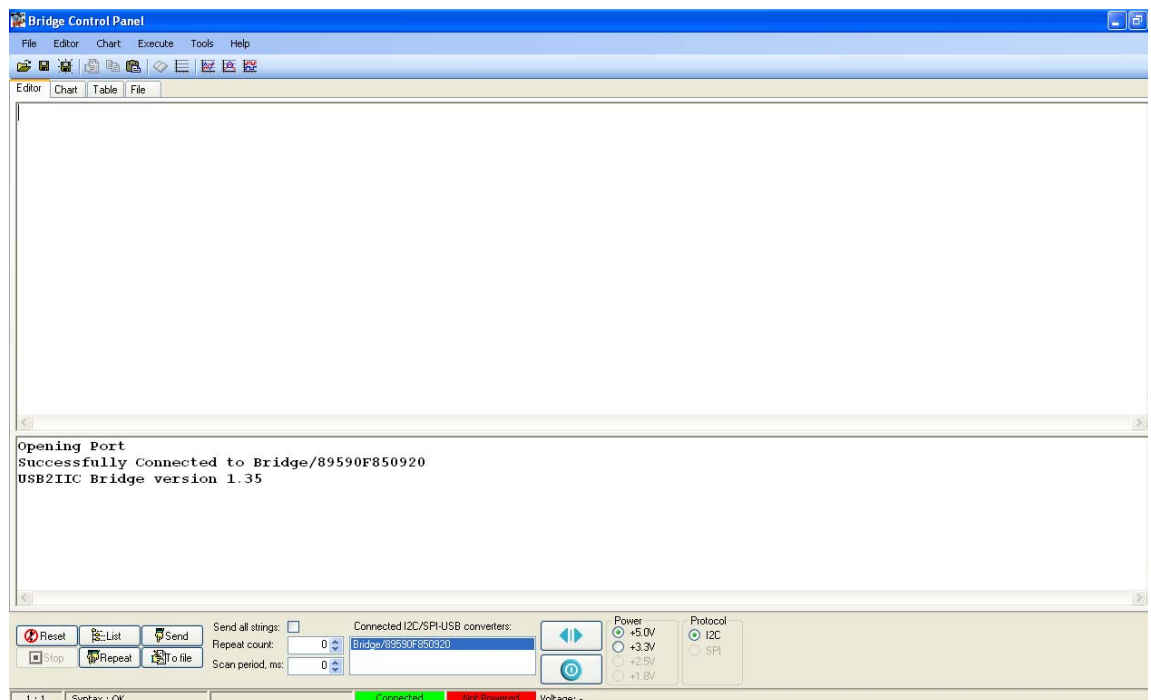
1. Connect the demo board to the bridge board.

Figure 5-34. Demo Board to Bridge Board Connection



2. Open Bridge Control Panel from **Start > All Programs > Cypress > Bridge Control Panel <version> > Bridge Control Panel**.
3. Select **+5V** as power supply to the board from the Bridge Control Panel; the red LED on the bridge board lights up.

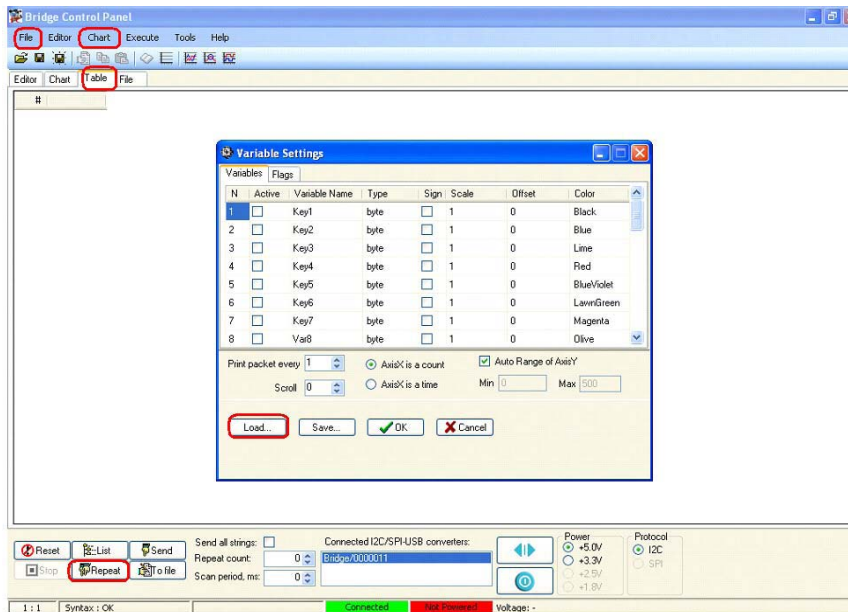
Figure 5-35. Bridge Control Panel



4. Verify the presence of *demo.ini* and *demo.iic* in the CY3240 USB-I2C Bridge Kit CD or `<Install_Directory>\Cypress\CY3240 USB-I2C Bridge Kit\<version>\Firmware\ Configuration file`
5. In the Bridge Control Panel, open the **Variable Setting** dialog box from the Chart menu.

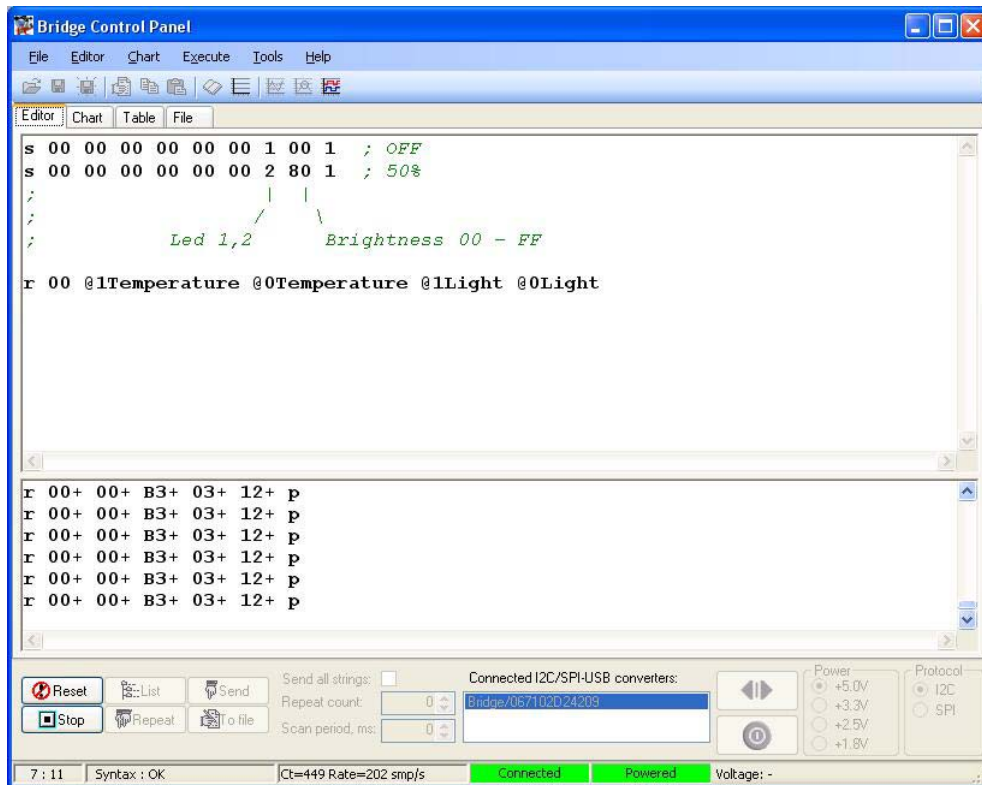
6. Load the *demo.ini* file, by clicking the **Load** button to initialize light and temperature variables, as shown in [Figure 5-36](#).

Figure 5-36. Load Demo.ini file from Bridge Control Panel



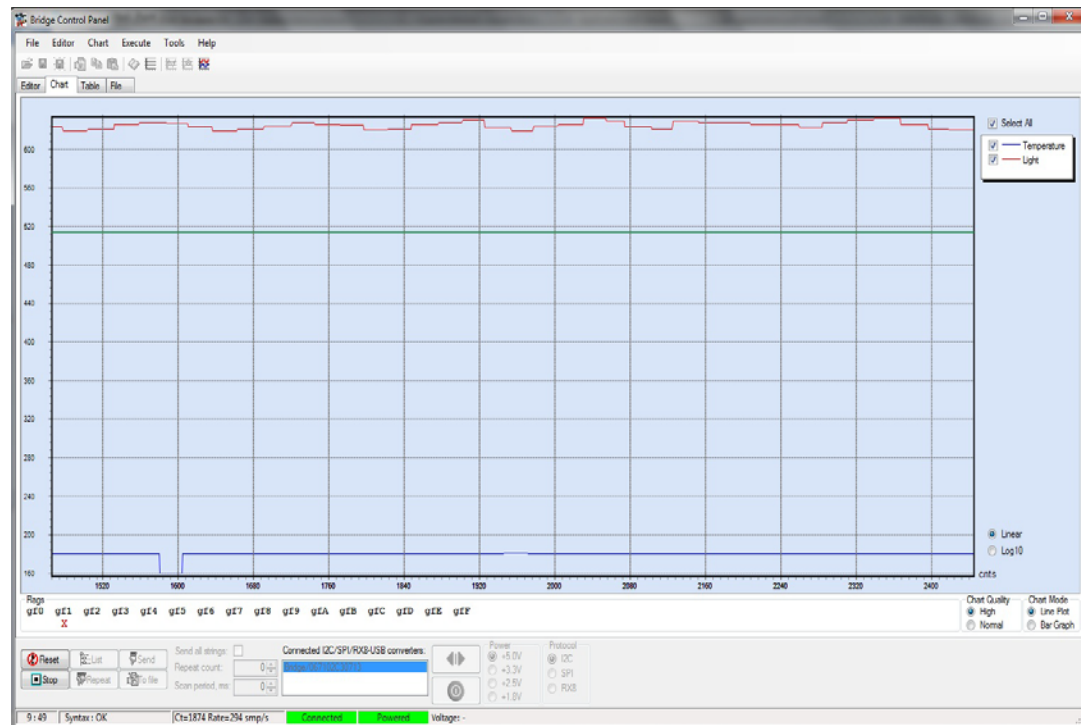
7. In the Bridge Control Panel, click **File > Open File** to select the *demo.iic* file for iic commands. The *demo.iic* file is used to transfer iic commands to the demo board, as shown in [Figure 5-37](#).

Figure 5-37. Bridge Control Panel Loaded with demo.iic File



8. The first two lines in the demo.iic file show how to control the LEDs on the board. Position the cursor in the first line and press **[Enter]** to send the command. Repeat for the second command line. Observe that, on sending the first command, the LED1 turns off. The second command reduces the LED intensity by 50%, as shown in [Figure 5-37](#).
9. Vary the Intensity value in the eighth byte (hex) value of first and second command lines of the Bridge Control Panel from 00-FF; observe the change in LED intensity on the demo board.
10. Click in the last line, which reads temperature and light data from the device. Click **Repeat** to send the last command repeatedly.
11. Click the **Chart** tab to view the data graphically, as shown in [Figure 5-38](#).

Figure 5-38. Bridge Control Panel with Plot of Temperature and Light Values



12. The ADC value analogous to the die temperature and light intensity incident on the board are plotted on the Y-axis of the graph versus the sample number on the X-axis of the graph.
13. Save and close the project.

5.2 USBtoIIC Code Example

5.2.1 Project Description

This project demonstrates communication between the bridge and the host. After each packet is received by the bridge, an acknowledgement is sent to the host before the next transfer operation begins.

This project uses the following modules:

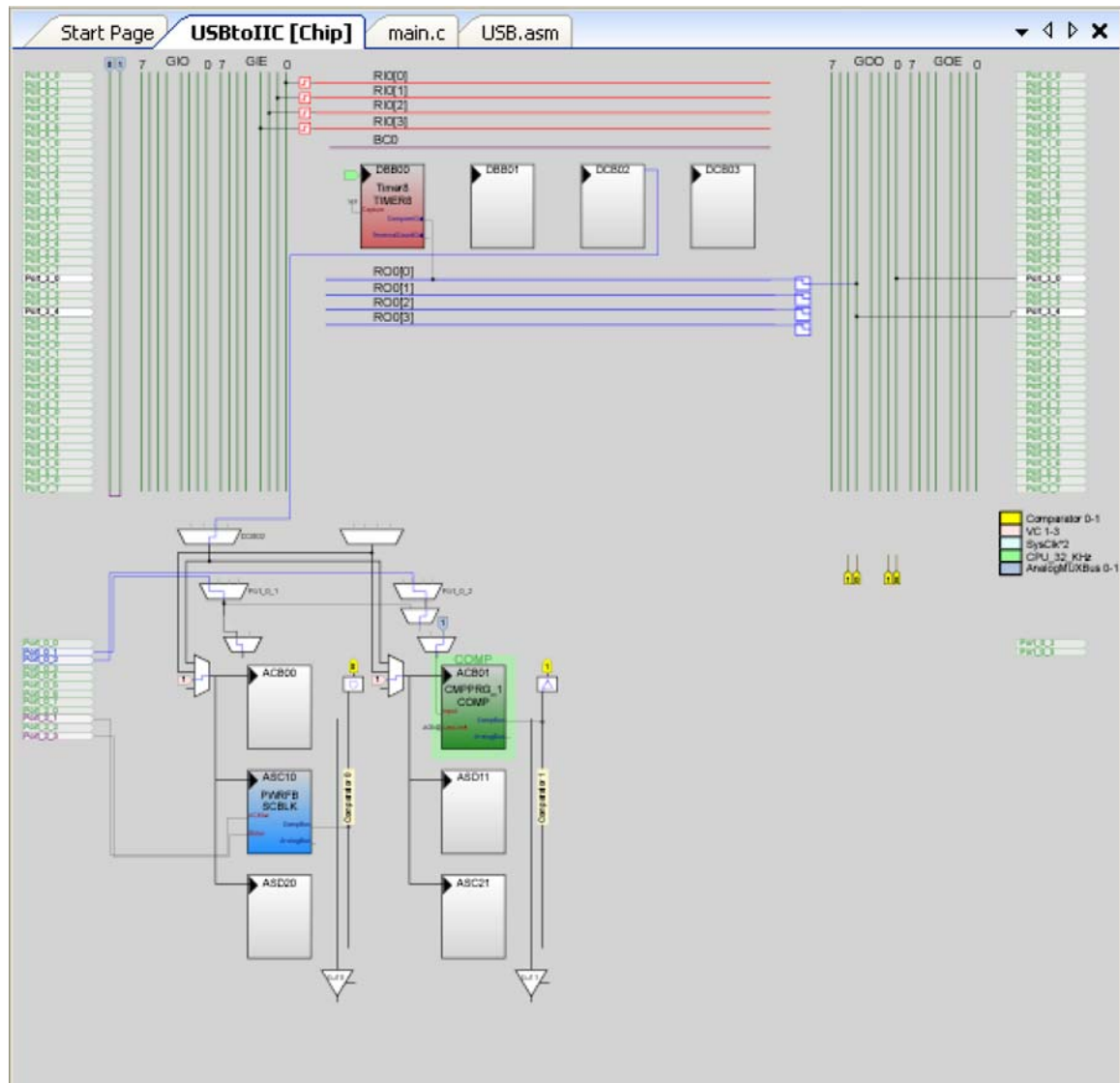
CMPRG: Used to compare the data programmable reference threshold.

PWRFB: Used to set the power level for this application.

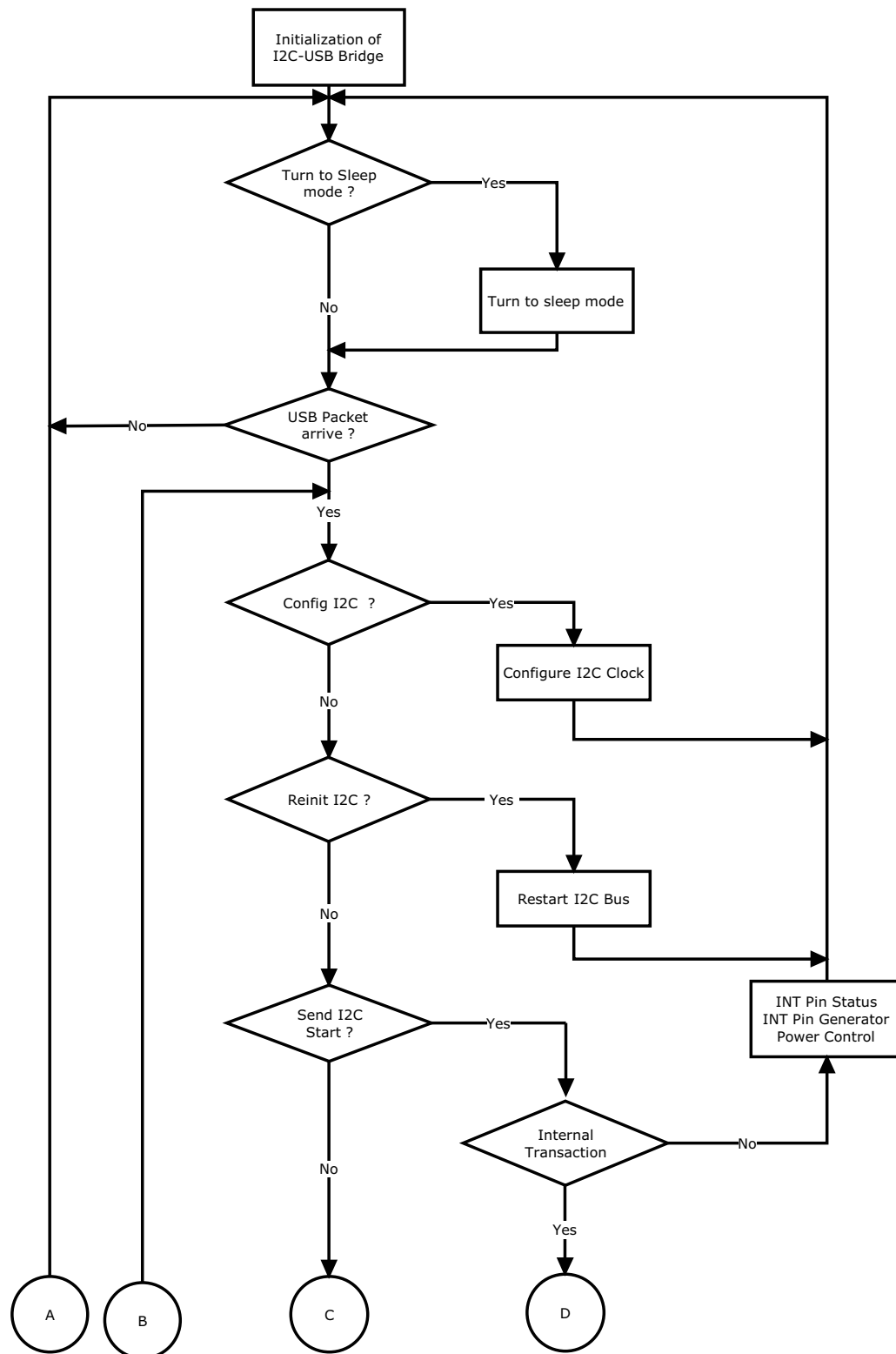
TIMER8: Used to wake up the application from sleep mode.

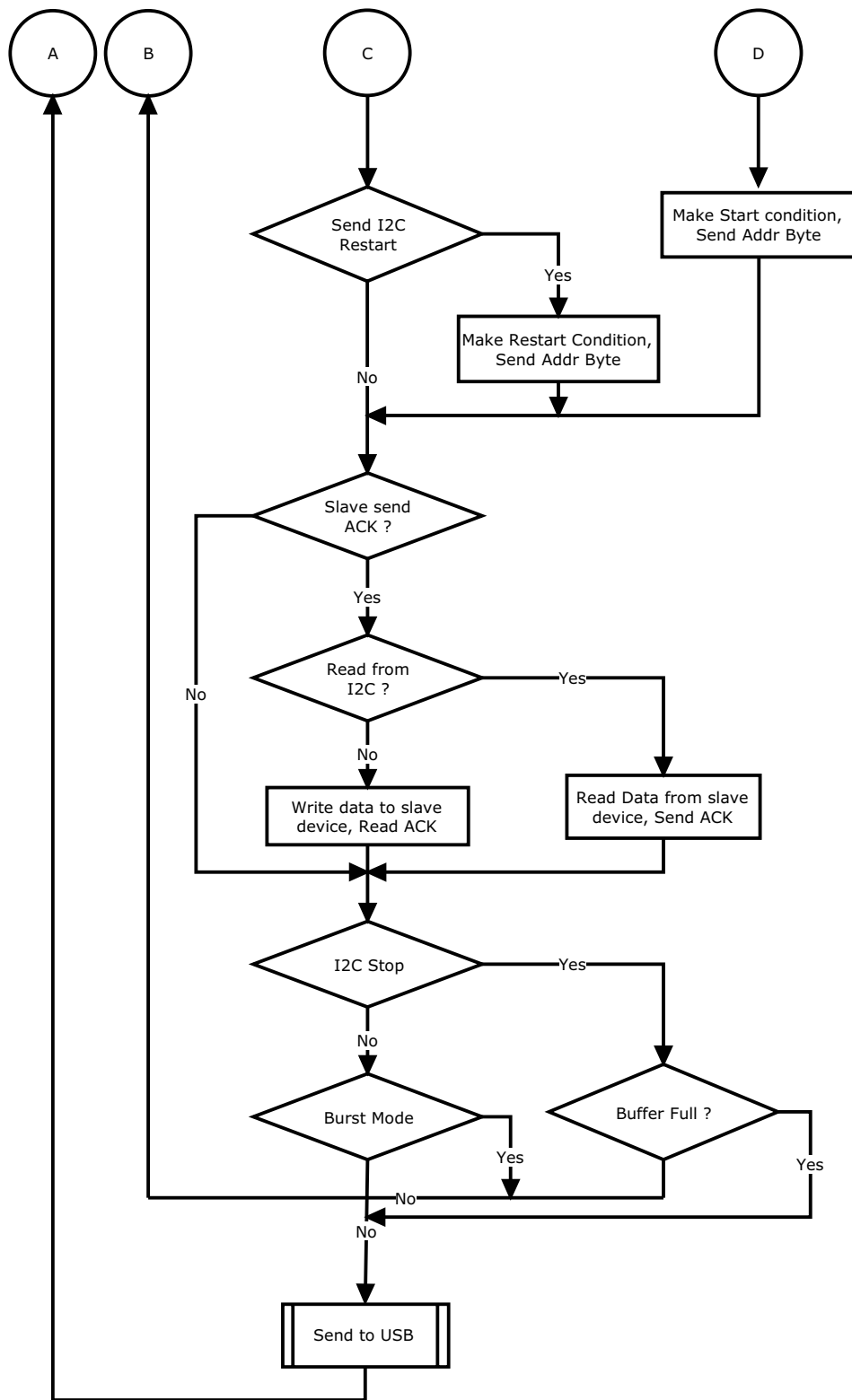
5.2.2 PSoC Designer Interconnected View

Figure 5-39. Functional Blocks



5.2.3 USBtoIIC Flowchart

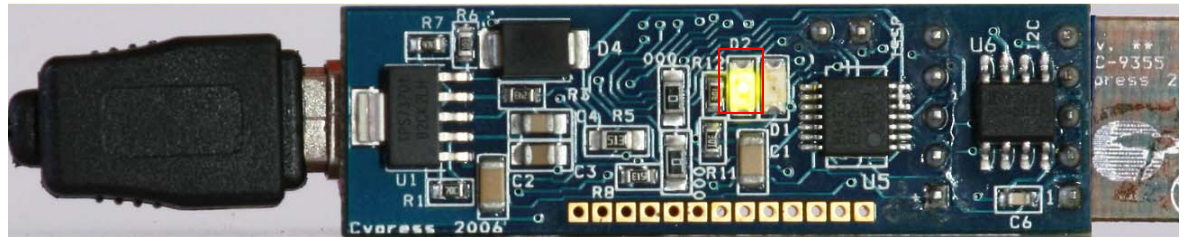




5.2.4 Verify Output

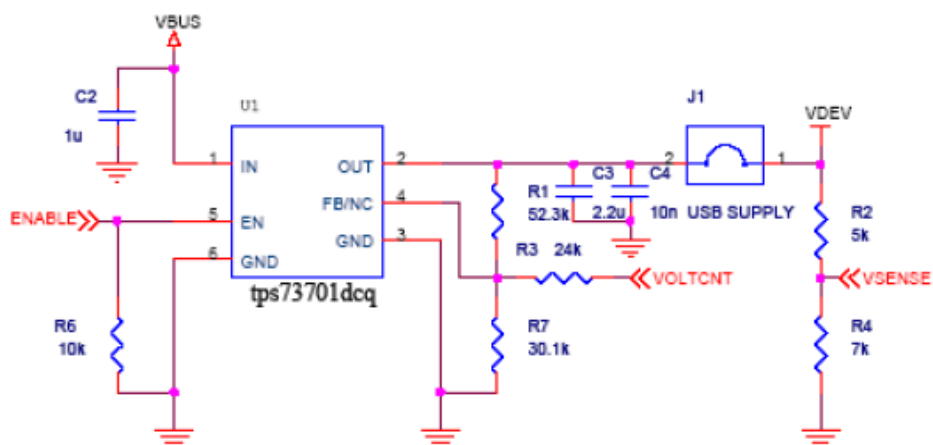
When connected to the demonstration board, the green LED on the bridge blinks indicating transfer operation between the bridge and the PC.

Figure 5-40. Verify Output



A.1 Schematic

Figure A-1. Reverse Current Protection Schematic



A.2 USB-I2C Bridge Board Layout

Figure A-7. Top View

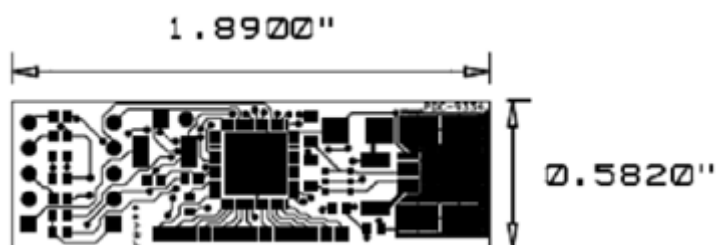
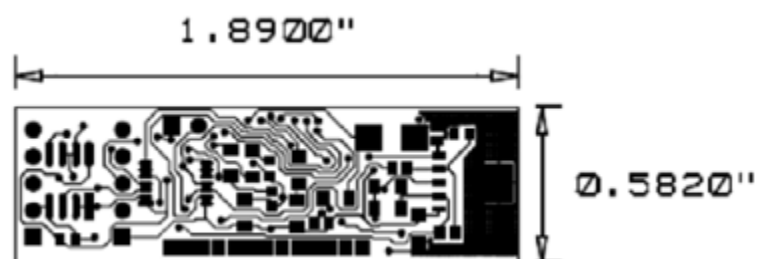
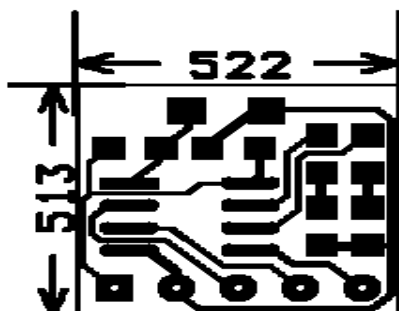


Figure A-8. Bottom View



A.2.1 Demo Board Layout

Figure A-9. Top View



A.3 Bill of Materials (BOM)

A.3.1 USB-I2C Bridge BOM

Item	Qty	Reference	Description	Manufacturer	Mfr Part Number
			PCB	Cypress Semiconductor	PDC-9334
1	1	C1	CAP CER .10UF 25V X7R 10% 0805	TDK	C2012X7R1E104K
2	1	C2	CAP CER 1.0UF 10V 10% X7R 0805	Murata Electronics North America	GRM21BR71A105KA01L
3	1	C3	CAP CER 2.2UF 16V Y5V 1206	Murata	GRM31MF51C225ZA01L
4	1	C4	CAP CER 10000PF 50V 10% X7R 1206	Murata	GRM319R71H103KA01D
5	2	C5, C6	CAP CERAMIC .01UF 100V X7R 0603	Kemet	C0603C103K1RACTU
6	1	R1	RES 52.3K OHM 1/10W 1% 0603 SMD	Yageo	RC0603FR-0752K3L
7	1	R2	RES 5.1K OHM 1/10W 5% 0603 SMD	Yageo	RC0603JR-075K1L
8	1	R3	RES 24.0K OHM 1/10W 1% 0603 SMD	Yageo	RC0603FR-0724KL
9	1	R4	RES 6.8K OHM 1/10W 5% 0603 SMD	Yageo	RC0603JR-076K8L
10	1	R5	RES 51K OHM 1/8W 5% 0805 SMD	Yageo	RC0805JR-0751KL
11	1	R6	RES 10K OHM 1/10W 5% 0603 SMD	Yageo	RC0603JR-0710KL
12	1	R7	RES 30.1K OHM 1/10W 1% 0603 SMD	Yageo	RC0603FR-0730K1L
13	1	R8	RES 51K OHM 1/10W 5% 0603 SMD	Yageo	RC0603JR-0751KL
14	2	R9, R10	RES 22 OHM 1/8W 5% 0805 SMD	Panasonic - ECG	ERJ-6GEYJ220V
15	2	R11, R12	RES 300 OHM 1/10W 5% 0603 SMD	Yageo	RC0603JR-07300RL
16	1	R13	RES 100K OHM 1/10W 5% 0603 SMD	Rohm	MCR03EZPJ104
17	2	R14, R15	RES 7.5K OHM 1/10W 5% 0603 SMD	Yageo	RC0603JR-077K5L
18	2	R16, R17	RES 30K OHM 1/10W 5% 0603 SMD	Yageo	RC0603JR-0730KL
19	3	R18, R19, R20	RES 47 OHM 1/10W 5% 0603 SMD	Yageo	RC0603JR-0747RL
20	1	J1	CONN HEADER VERT 2POS .100 TIN	Molex/Waldom Electronics Corp	22-28-4020
21	1	J2	CONN HEADER VERT 5POS .100 TIN	Molex/Waldom Electronics Corp	22-28-4050
22	1	J3	CONN HEADER .100 SNGL STR 5POS	3M Electronics	929850-01-05-10
23	1	J4	CONN USB RCPT MINI-B 5POS RT ANG	Delphi Connection Systems	15430262-110
24	2	F1, F2	POLY SWITCH 1.10A RESET FUSE SMD	Tyco Electronics/Raychem Circuit Protection	MINISMDC110-2
25	1	D1 *	LED 660NM RED WTR CLR 1206 SMD	Lumex Opto/Components Inc	SML-LX1206SRC-TR
26	1	D2 *	LED 565NM WTR CLR GREEN 1206 SMD	Lumex Opto/Components Inc	SML-LX1206GC-TR
27	2	D3, D4	Supressor 5V SMB package	Littelfuse Inc	SMBJ5.0CA
28	1	U1	IC LDO REG 1A SOT223-6 TPS73701	Texas Instruments	TPS73701DCQ
29	1	U2	CY8C24894	Cypress Semiconductor	CY8C24894-24LFXI
30	1	U4 *	IC Single USB Port TVS SOT-23-6	Texas Instruments	SN65220DBVT
31	1	U5	IC LVL XLTR LV 8MBPS 14-TSSOP	Maxim	MAX3378EEUD+
32	1	U6	TVS Array ESD 6 Input 8-SOIC	Littelfuse Inc	SP721ABG
33	2	* 000	RES 0.0 OHM 1/8W 5% 0805 SMD	Rohm	MCR10EZJHJ000
34	1	* 000	RES 0.0 OHM 1/8W 5% 0603 SMD	Rohm	MCR03EZPJ000
35	1	J1 *	Shunt	3M	929950-00

A.3.2 Demo Board

Item	Qty	Reference	Description	Manufacturer	Mfr Part Number
			PCB	Cypress Semiconductor	PDC-9355
1	1	C1	CAP 100nF 25V CERAMIC X7R 0805	Panasonic - ECG	ECJ-2VB1E104K
2	1	R1	RES 75K OHM 1/8W 5% 0805 SMD	Yageo America	RC0805JR-0775KL
3	2	R2, R3	RES 1.0K OHM 1/8W 5% 0805 SMD	Yageo America	RC0805JR-071KL
4	1	D1	Visible Light Sensor	Microsemi Inc	LX1972IBC
5	2	D2, D3	LED RED CLEAR 0805 SMD	LITE-ON INC	LTST-C170CKT
6	1	U1	IC PSoC 21x23 8SOIC	Cypress Semiconductor	CY8C21123-24SXI
7	1	J1	CONN HEADER .100 SNGL STR 5POS	Samtec	TSW-105-07-T-S