CY3210-PSoCEVAL1

PSoC® 1 Evaluation Kit Guide

Doc. #: 001-66768 Rev. *D

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

Copyrights

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

Thank you for your interest in PSoC® 1. This user guide will help you get started with the CY3210-PSoCEVAL1 PSoC 1 Evaluation Kit; provide hardware description, instructions on software installation, and kit operation; and walk you through the code examples.

The code examples implement commonly used peripherals such as analog-to-digital converter (ADC), digital-to-analog converter (DAC), universal asynchronous receiver/transmitter (UART), pulse-width modulator (PWM), pseudo random sequencer (PRS), and LCD. They are functional in PSoC Designer™, which is PSoC 1's GUI-based Integrated Design Environment (IDE). Peripherals in PSoC Designer are implemented as preprogrammed precharacterized 'User Modules'. Evaluate the library of over 100 user modules in PSoC Designer and experience simpler and faster designs.

The evaluation board features a pluggable character LCD module with contrast control, status LEDs, a potentiometer, push button switches, a UART, an RS-232 interface, an ISSP programming header, and prototyping area. The MiniProg unit, included in this kit is required to program PSoC 1 devices directly on the evaluation board. Kit schematics, layout, and bill-of-materials (BOM) are provided in the Appendix on page 65.

1.1 Kit Contents

The CY3210-PSoCEVAL1 Evaluation Kit contains:

- PSoCEVAL1 evaluation board
- MiniProg programmer
- CY8C29466-24PXI 28-pin DIP sample
- CY8C27443-24PXI 28-pin DIP sample
- CY3210-PSoCEVAL1 kit DVD
  - PSoC Designer installation file
  - PSoC Programmer installation file
  - Bridge Control Panel installation file (packaged along with PSoC Programmer)
  - Code examples
  - Hardware files
  - Kit guide
  - Quick start guide
  - Release notes
- USB cable
- LCD module
- Single strand jumper wire pack

Inspect the contents of the kit; if any parts are missing, contact your nearest Cypress sales office for help.

1.2 Additional Learning Resources

Visit http://www.cypress.com for additional learning resources in the form of datasheets, technical reference manual, and application notes. For the latest information about this kit, visit http://www.cypress.com/go/CY3210-PSoCEval1.

- *PSoC Designer: PSoC Designer Overview*

  http://www.cypress.com/go/psocdesigner
- *PSoC Designer Training: PSoC Designer On-Demand Training Series and Videos*

  http://www.cypress.com/psoctraining
- *PSoC Programmer, COM Hardware Layer Supported Languages*

  http://www.cypress.com/go/psocprogrammer
- *PSoC Programmable System-on-Chip™ Datasheets*

  http://www.cypress.com/?mpn=CY8C29466-24PXI

  http://www.cypress.com/?mpn=CY8C27443-24PXI
- AN75320 - Getting Started with PSoC 1

  This application note describes the capabilities of PSoC 1 devices and the PSoC Designer development environment used to configure and program these devices. An introductory project is included to help you develop PSoC 1 applications.
- AN73212 - Debugging with PSoC 1

  This application note introduces the elements of the PSoC 1 debugger system and explains how to configure and use them effectively.
- AN2010 - PSoC 1 Best Practices and Recommendation

  This application note provides introductory guidelines and best practices for developing PSoC 1 systems; it exposes some common mistakes designers make.
- AN74170 - PSoC 1 Analog Structure and Configuration with PSoC Designer

  This application note explains the analog structure of standard PSoC 1 devices and how the global analog parameters affect many of the analog user modules.
- AN2027 - Using the PSoC Microcontroller External Crystal Oscillator

  The external crystal oscillator in the PSoC microcontroller has specific requirements for correct operation in different configurations. This application note details these requirements.
- AN2014_Introduction to PSoC 1 Programming

  This application note discusses how to design an application to enable ISSP with PSoC Designer using either the device reset or power cycle programming mode.

1.3 Document History

Revision	PDF Creation Date	Origin of Change	Description of Change
**	01/21/2011	RKPM	Initial version of kit guide
*A	03/29/2011	RKPM	Added section 5.1 My First Code Example. Other updates to text and figures throughout the document.
*B	04/11/2011	SASH	Updated Figures 5-22, 5-23, and 5-26.
*C	09/14/2012	RKPM	Multiple updates throughout the document
*D	11/20/2012	ARVI	Updated images

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\`
Italics	Displays file names and reference documentation: Read about the *sourcefile.hex* file in the *PSoC Designer User Guide*.
[**Bracketed, Bold**]	Displays keyboard commands in procedures: [**Enter**] or [**Ctrl**] [**C**]
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 how to install and configure the CY3210-PSoCEVAL1 kit.

2.1 Kit Installation

To install the kit software, follow these steps:

1. Insert the kit DVD into the DVD drive of your PC. The DVD 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/CY3210-PSoCEval1. Three different types of installers are available for download.

   a. CY3210-PSoCEval1_ISO: This file (ISO image) is an archive file of the optical disc provided with the kit. You can use this to create an installer DVD or extract information using WinRar or similar tools.

   b. CY3210-PSoCEval1_ Single Package: This executable file installs the contents of the kit DVD, which includes PSoC Programmer, PSoC Designer, kit code examples, kit hardware files, and user documents.

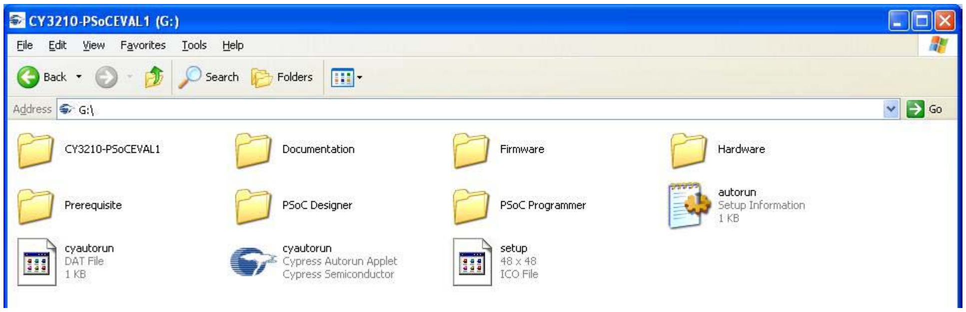
   c. CY3210-PSoCEval1_Single Package (without prerequisites): This executable file installs only the kit contents, which includes kit code examples, hardware files, and user documents.

2. Click **Install the CY3210-PSoCEVAL1** to start the kit installation, as shown in Figure 2-1.

   Figure 2-1. Kit Installer Startup Screen

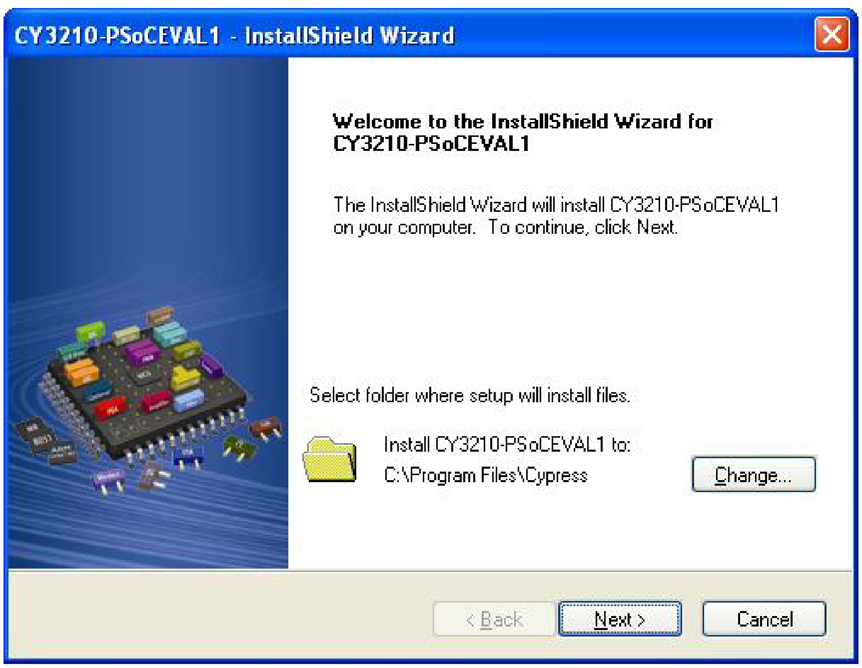
Note If auto-run does not execute, double-click *cyautorun.exe* file on the root directory of the DVD, as shown in Figure 2-2. To access the root directory, click **Start > My Computer > CY3210-PSoCEVAL1 <drive:>**.

Figure 2-2. Root Directory of DVD



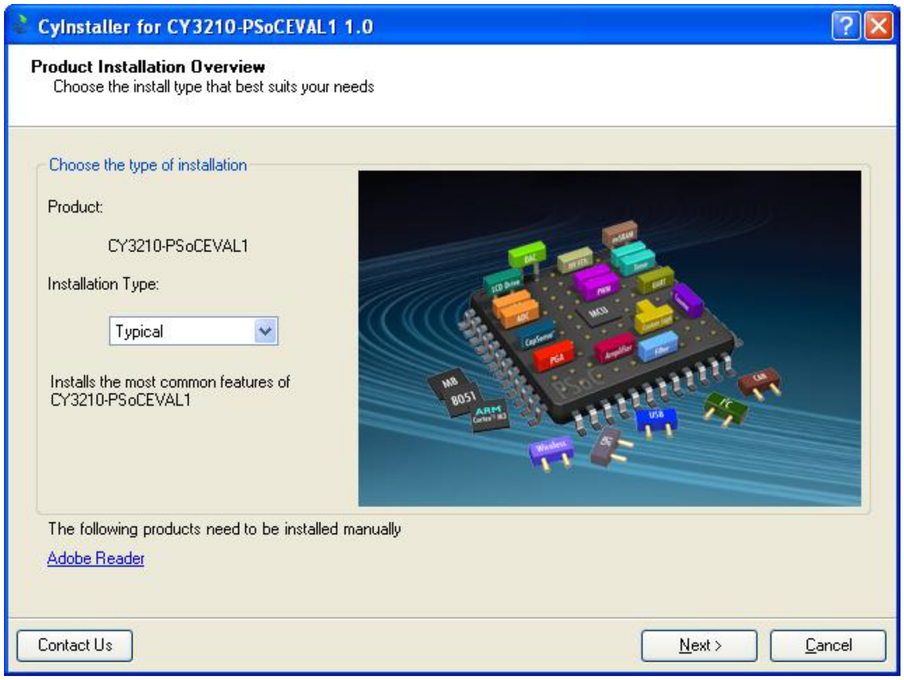
3. On the startup screen, click **Next** to start the installer.
4. The **InstallShield Wizard** screen appears. On this screen, choose the folder location to install the setup files. You can change the folder location for setup files using **Change**, as shown in Figure 2-3.
5. Click **Next** to launch the kit installer.

Figure 2-3. InstallShield Wizard



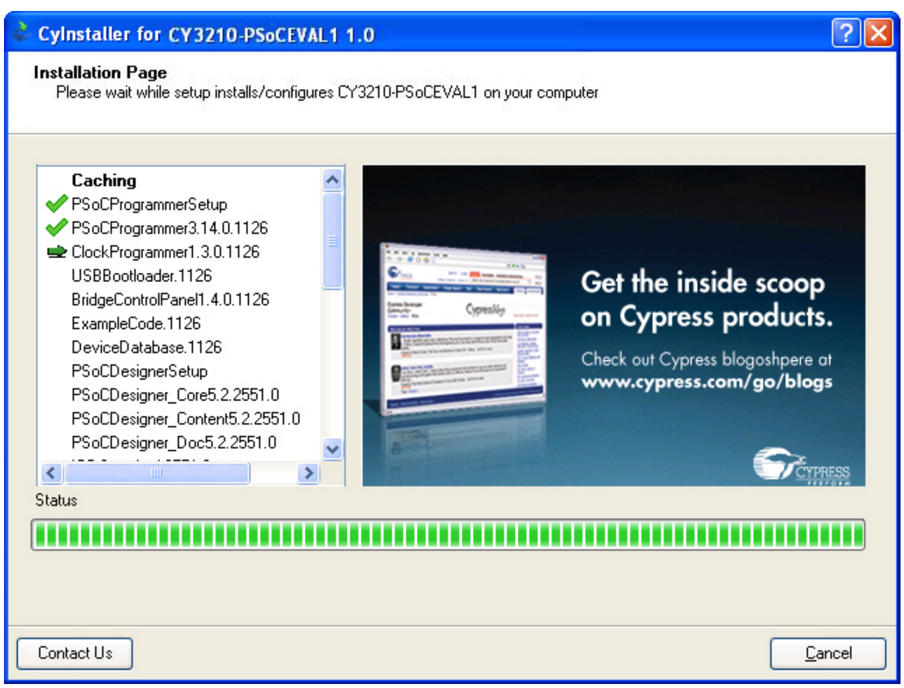
6. On the **Product Installation Overview** screen, select the installation type that best suits your requirement. The drop-down menu has three options: **Typical**, **Complete**, and **Custom**, as shown in Figure 2-4. If you are uncertain, proceed with the default setting (Typical).
7. Click **Next**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

Figure 2-4. Installation Type Options



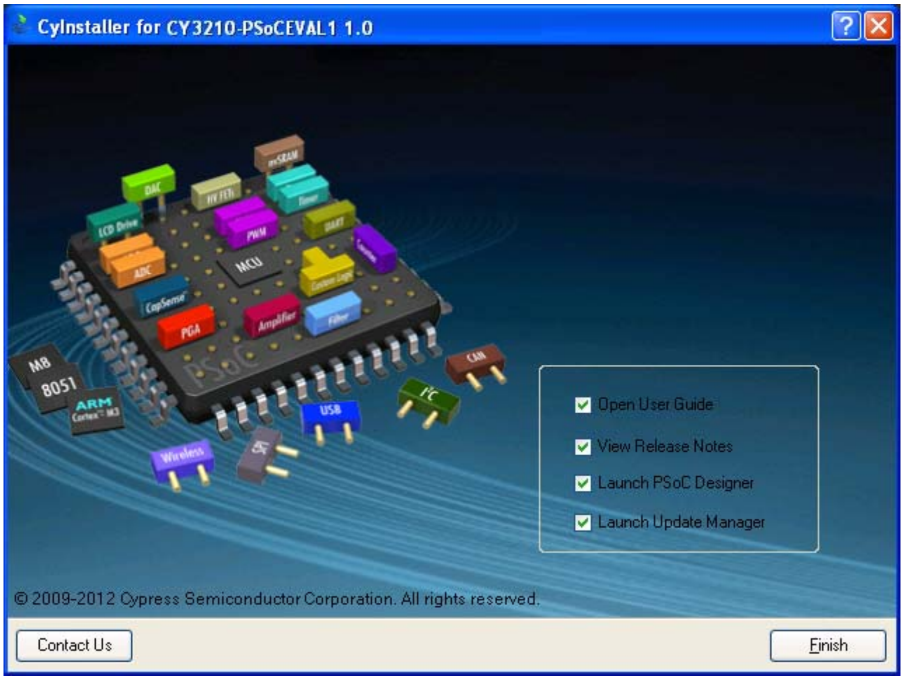
8. When the installation begins, a list of all packages appear on the **Installation Page**. A green check mark appears against every package that is downloaded and installed, as shown in Figure 2-5.
9. Wait until all the packages are downloaded and installed successfully.

Figure 2-5. Installation Page



10. Click **Finish** to complete the installation.

Figure 2-6. Installation Complete



After software installation, drivers are installed when MiniProg1 is connected to the PC for the first time. Verify driver installation by opening PSoC Programmer with the MiniProg connected to the USB port of the PC. The connected device will be listed under the Port Selection window in PSoC Programmer.

Note Advanced users can skip to Code Examples on page 27.

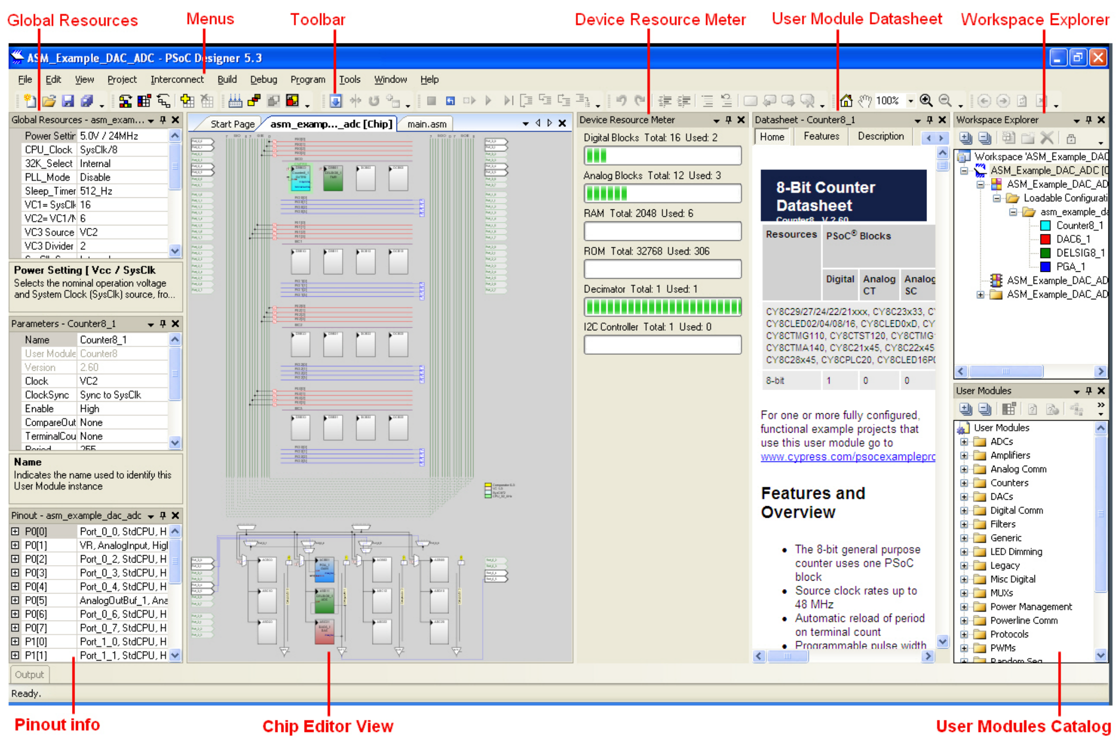
2.2 PSoC Designer

PSoC Designer is the integrated design environment (IDE) that you can use to customize your PSoC application. The latest version of PSoC Designer has several new features, bug fixes, and support for new PSoC devices.

This section gives a brief introduction on the PSoC Designer Interconnect View. Additional details on the PSoC Designer software is given in the Code Examples chapter on page 27.

1. Click **Start > All Programs > Cypress > PSoC Designer <version> > PSoC Designer <version>**.

Figure 2-7. PSoC Designer Interconnect View



Note The **Datasheet** and **Resource Meter** windows are hidden by default. To include them in the view, click on **View** and select the required windows.

Note For more details on PSoC Designer, see the PSoC Designer IDE Guide at the following location: `<Install_directory>:\PSoC Designer\<version>\Documentation`. You can also access this document via the Help menu (**Help > Documentation**).

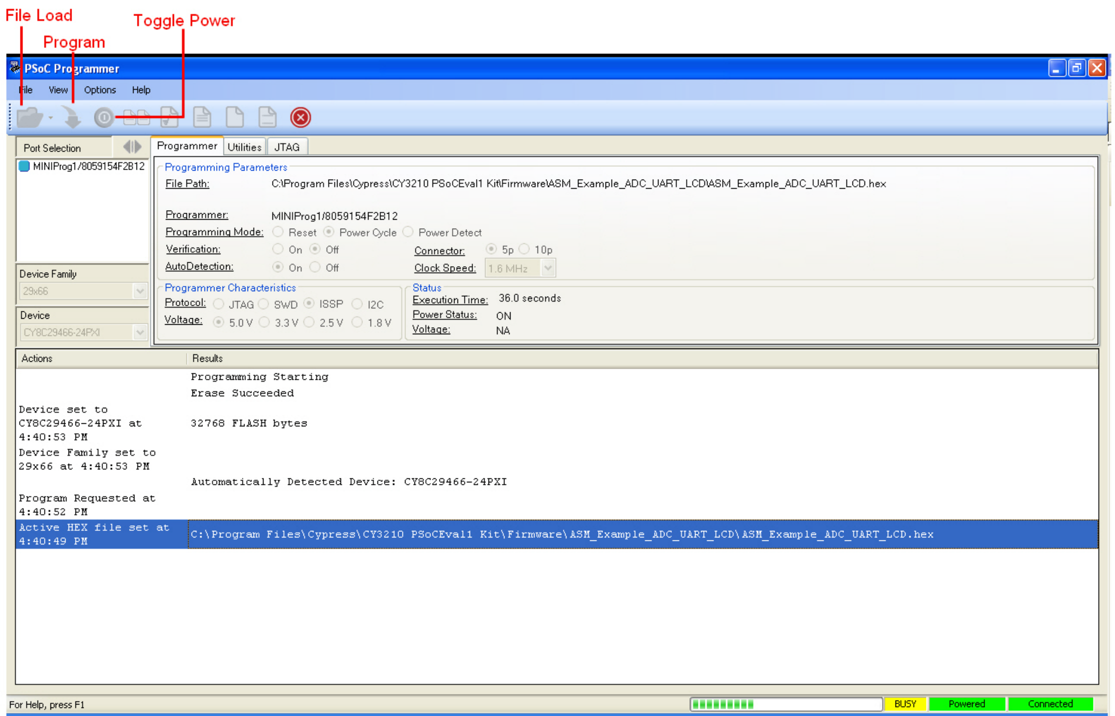
See Additional Learning Resources on page 6 for links to PSoC Designer training. The PSoC Designer quick start guide is available at: http://www.cypress.com/?rID=47954

2.3 PSoC Programmer

This section gives a brief introduction on programming PSoC; for kit specific programming instruction, see Programming Specifications and Connections on page 16.

1. Click **Start > All Programs > Cypress > PSoC Programmer <version> > PSoC Programmer <version>**.
2. Select the MiniProg from **Port Selection**.

Figure 2-8. PSoC Programmer Window



3. Click the **File Load** button to load the hex file.
4. Use the **Program** button to program the hex file on to the chip.
5. When programming is successful, the “Programming Succeeded” message appears in the Action pane.
6. Close PSoC Programmer.

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

3. Kit Operation

The CY3210-PSoCEVAL1 kit helps in evaluating applications using the PSoC 1 family of devices. This kit can be used to work with the different user modules provided in the PSoC Designer software and explore hardware features that are integrated into the PSoC 1 device.

3.1 Evaluating the PSoC 1 Device

The CY3210-PSoCEVAL1 board is populated with a preprogrammed CY8C29466-24PXI part. To evaluate the default project programmed on the board, make the following hardware connections:

- Connect the jumper (shunt) at JP1 and JP2.
- Connect the potentiometer (VR on connector J5) and port P0[1] on connector J6 using one of the single-strand jumper wires (see Figure 3-1). This connects one of the PSoC pins to the potentiometer.

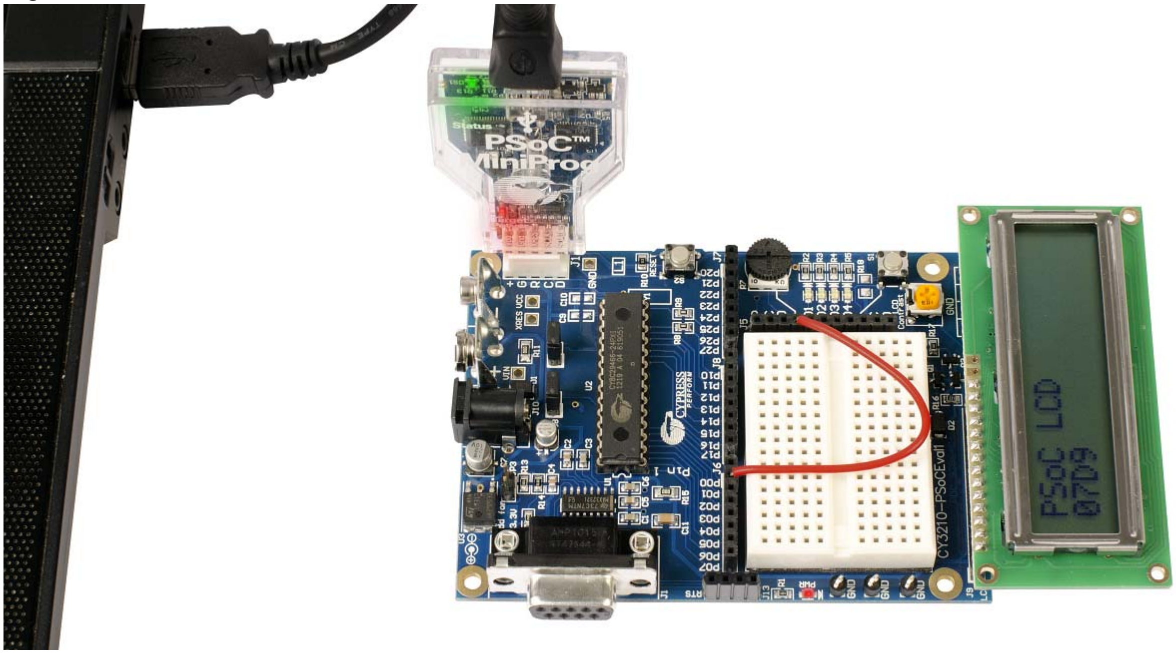
After the connections are made, plug in the MiniProg to the ISSP header. Connect the MiniProg to the PC using the USB cable provided. Power the kit using PSoC Programmer. For more information on using PSoC Programmer, see Programming Specifications and Connections on page 16. The CY3210-PSoCEVAL1 can be powered from a DC supply jack, battery terminals, or using the MiniProg. The PWR LED5 lights up (red) when the board is powered.

You can vary the analog input by varying the potentiometer (R7). The output is displayed on the LCD module. LCD contrast control potentiometer (R6) can be used to vary the LCD contrast.

The PSoC MiniProg gives you the ability to program PSoC parts quickly and easily. It is small and compact, and connects to your PC using the USB cable. During prototyping, the MiniProg can be used as in-system serial programming (ISSP) to program PSoC devices, as shown in Figure 3-2.

MiniProg1 does not support I2C communication. The CY3240-I2USB and MiniProg3 can be used as a USB-I2C Bridge for debugging I2C serial connections and communicating to PSoC devices.

Figure 3-1. PSoCEVAL1 Evaluation Board



3.1.1 Programming Specifications and Connections

When the MiniProg is connected, you can use PSoC Programmer to program the CY3210-PSoCEVAL1 Evaluation kit. Plug in the USB cable into the MiniProg before attaching it to the ISSP header on the board. When using a USB cable with MiniProg, keep the length under six feet to avoid signal integrity issues.

When using MiniProg, the LEDs blink at a variable rate to track connection status. The green LED near the USB connector turns on after MiniProg is plugged into the computer and is configured by the operating system. If MiniProg cannot find the correct driver in the system, this LED does not turn on. After the device is configured, the LED stays on at about a 4-Hz blink rate. This changes during programming, where the blink duty cycle increases.

The red LED (Figure 3-2) at the bottom turns on when the MiniProg powers the part. The LED is off when power is provided by the target board.

Figure 3-2. Programming PSoC Device

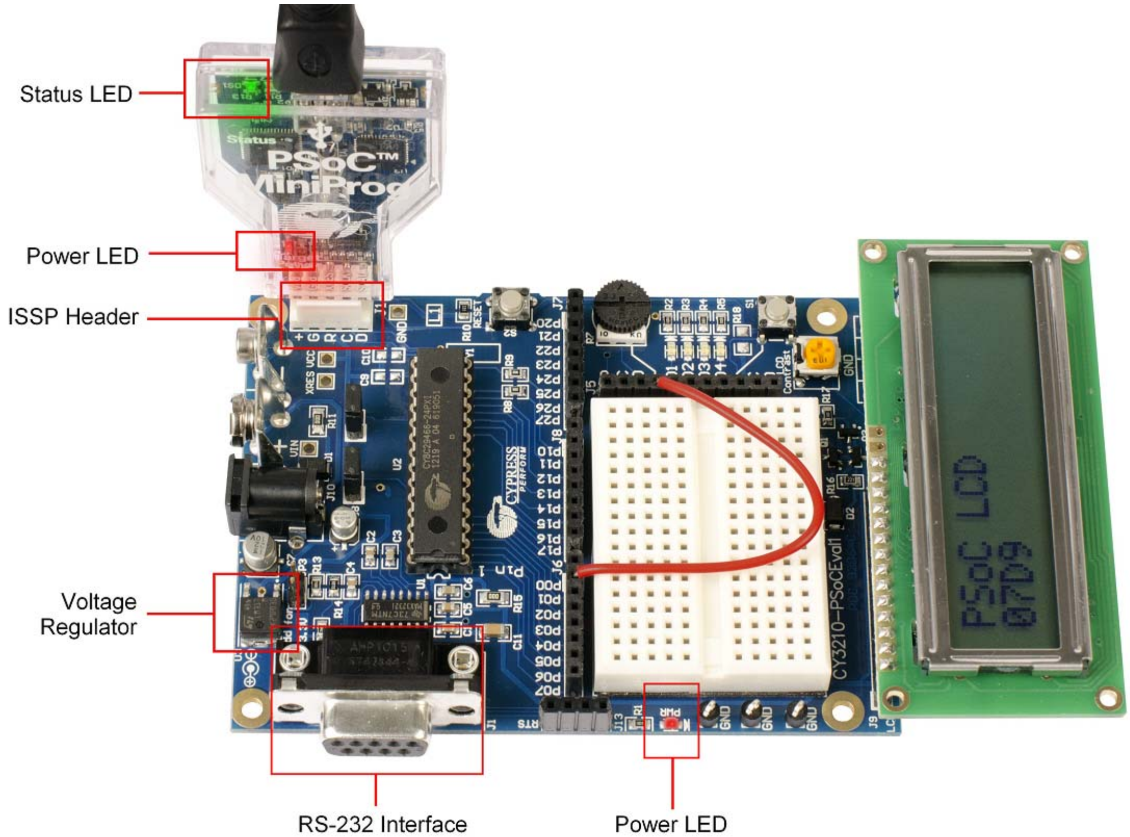
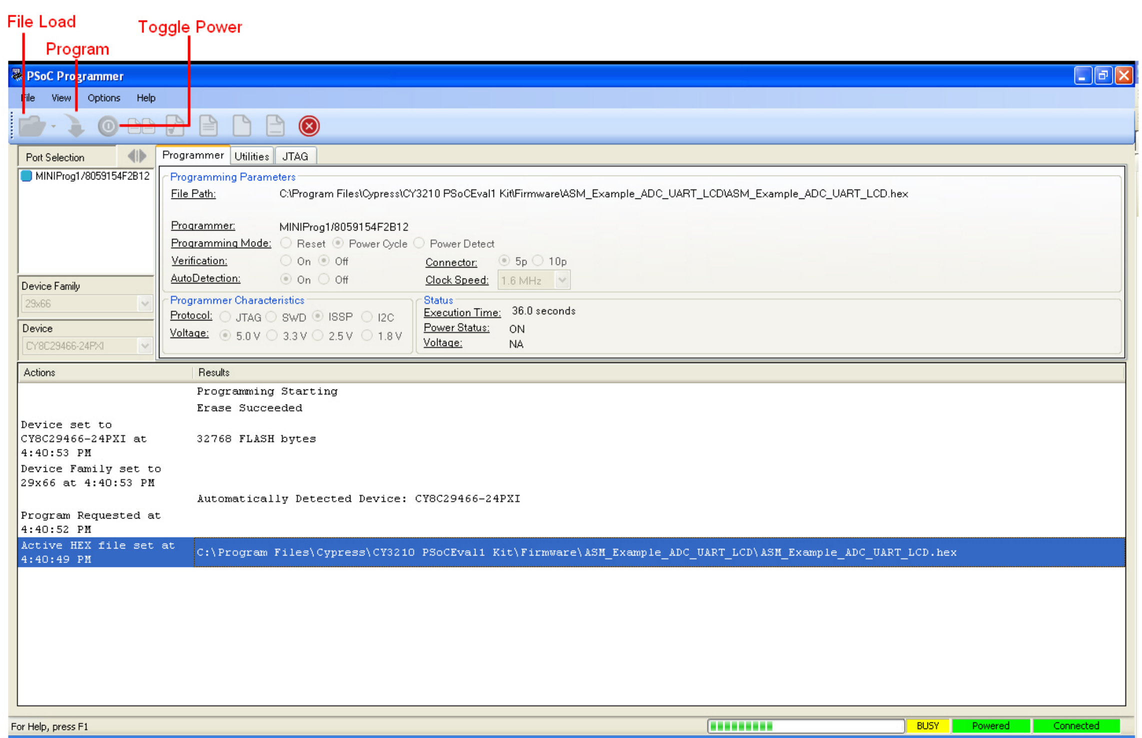


Figure 3-3. PSoC Programmer Screen



Follow these steps to program using MiniProg1:

1. Connect the MiniProg1 to the PC using the USB cable.
2. Plug in the MiniProg1 to the ISSP header on the CY3210-PSoCEVAL1 board.
3. When USB is connected to the MiniProg1, LED (green) glows in the MiniProg1.
4. Open PSoC Programmer.
5. Click the **Load File** button and browse to the hex file location (<Install_directory>:\Cypress\CY3210-PSoCEVAL1\<version>\Firmware\ASM_Example_ADC_UART_LCD\ASM_Example_ADC_UART_LCD.hex). Click **Open** to select the hex file.
6. Click **Connect** or double-click on the respective MiniProg under Port Selection to select or connect to MiniProg.
7. Make sure the **Power Cycle** radio button is selected to power the kit using USB power.
8. Click **Program** or press **[F5]** to initiate programming.
9. The green LED on the MiniProg1 blinks to indicate the progress of programming.
10. After successful programming, the red LED on MiniProg1 is powered off.
11. Select the **Toggle Power** button in PSoC Programmer to power the board and verify output.

See the section Replace CY8C29466-24PXI with CY8C27443-24PXI on page 70 for instructions on how to replace the CY8C29466-24PXI PSoC part with CY8C27443-24PXI.

4. Hardware

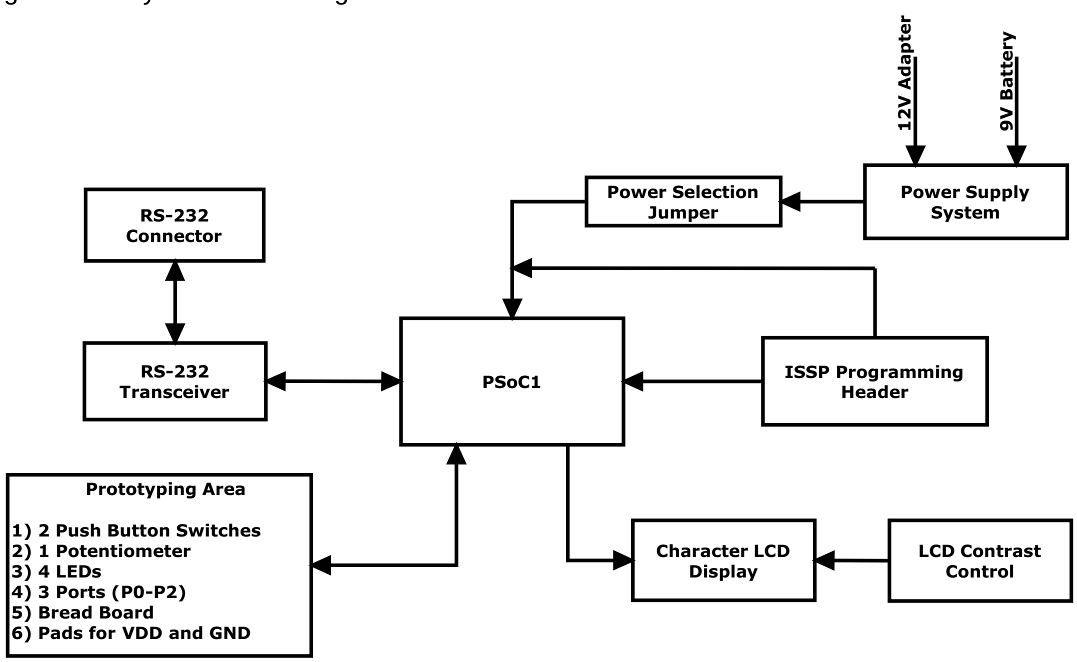
This section provides an overview of the development kit hardware, including power system, jumper setting, and programming interface. To start using the board, go to Code Examples on page 27.

4.1 CY3210-PSoCEVAL1 System Block Diagram

The CY3210-PSoCEVAL1 Evaluation Kit consists of:

- Power supply system
- Programming interface
- PSoC 1
- RS-232 interface
- Prototyping area
- Character LCD interface

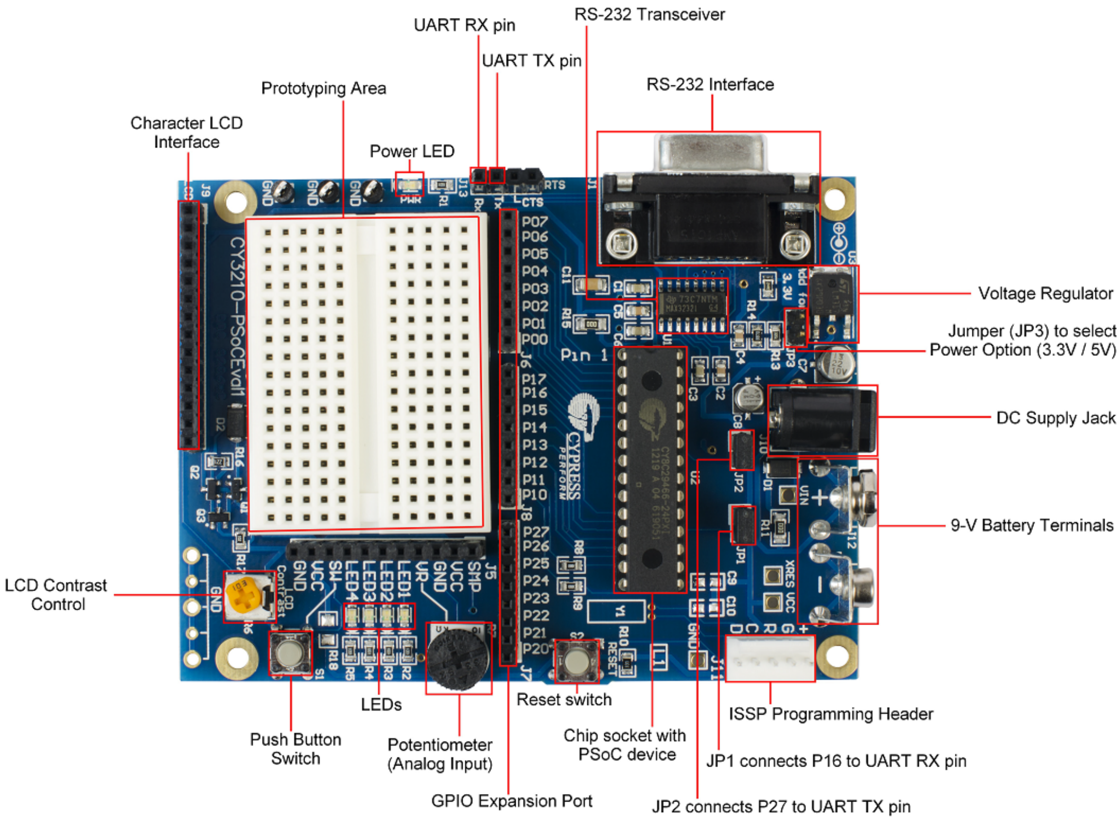
Figure 4-1. System Block Diagram



4.2 Functional Description

The CY3210-PSoCEVAL1 Evaluation Kit demonstrates the function of PSoC 1 devices. Connect the device to onboard peripherals such as potentiometer, LEDs, LCD, and RS-232. The board also has additional features such as a general prototype area (bread board) and an ISSP programming header. It provides different voltage domains; see 4.2.1 Power Supply System.

Figure 4-2. CY3210-PSoCEVAL1 Evaluation Board



4.2.1 Power Supply System

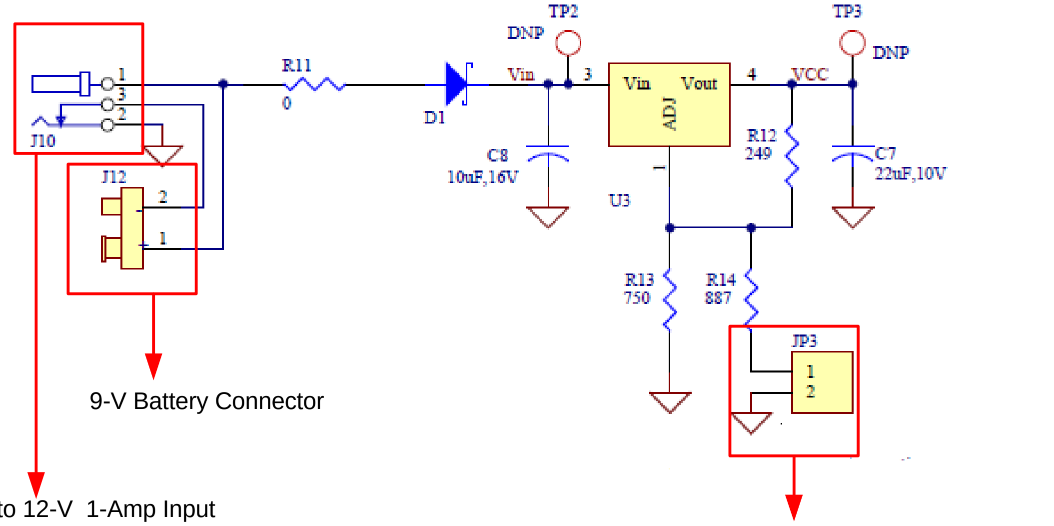
The power supply system on this board is versatile. The kit can be powered in three ways: using a MiniProg unit, DC wall adaptor, or a 9-V battery. The DC wall adaptor must be of 9 V to 12 V, 1-A rating unit. The onboard voltage regulator converts input 9 V/12 V into 3.3 V/ 5 V. Selection between 3.3 V and 5 V is done using jumper JP3. When the jumper (shunt) is not connected, the regulator gives the output voltage of 5 V; installing the jumper gives a 3.3-V output.

Note Use only one power supply at a time. Do not use a power supply that is less than 7 V or exceeds 12 V.

If the board needs to be powered via external 3.3-V or 5-V supply, the power input should be connected to the VCC sockets of the J5 connector (Figure 4-9). The external voltage must be less than 5.5 V.

Warning Any voltage input greater than 12 V connected on these headers will permanently damage the board components.

Figure 4-3. Power Supply Schematic

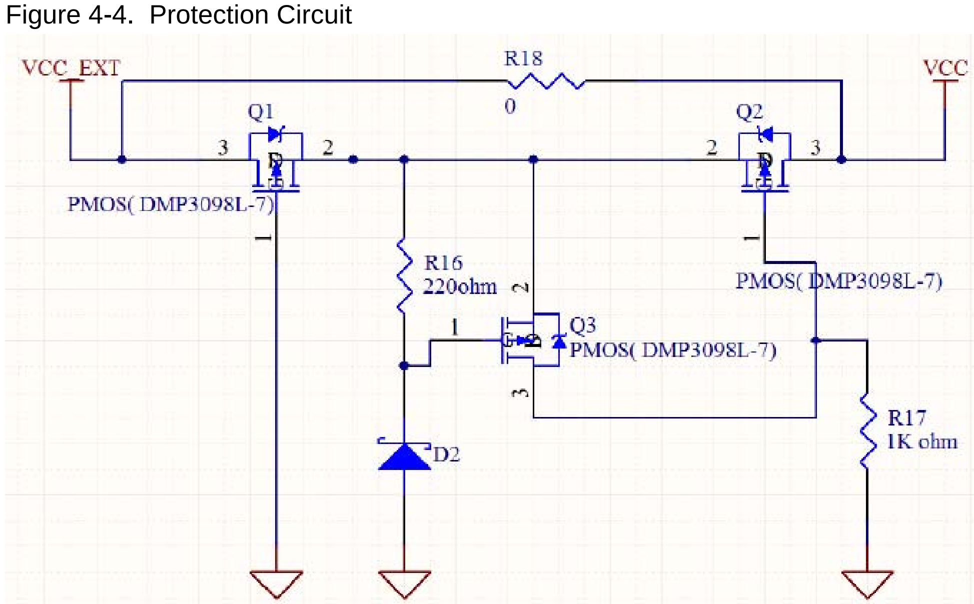


4.2.1.1 Protection Circuit

A reverse-voltage and over-voltage protection circuit is added at the expansion port on VCC lines. The protection circuit consists of two P-channel MOSFETs on the power line allowing the power/current to flow from input to output depending on the voltages applied at the external board connector. Figure 4-4 is the protection circuit placed between the VCC domain near the prototype board (on J5 connector) and the onboard components. This circuit disconnects or behaves similar to an open switch if an external voltage above 5.6 V is applied on the VCC pins of the J5 connector.

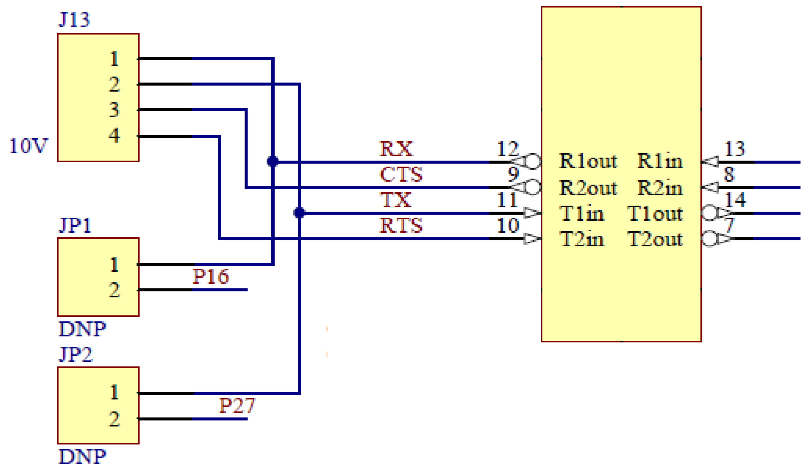
Warning Any voltage input greater than 12 V connected on these headers will permanently damage the board components.

Figure 4-4. Protection Circuit



4.2.1.2 Jumper Settings

Figure 4-5. Jumper Settings



The functions of JP1, JP2, and JP3 are as follows:

- JP1 connects P16 to Rx pin for UART communication and should be removed for normal I/O operation.
- JP2 connects P27 to Tx pin for UART communication and should be removed for normal I/O operation.
- JP3 controls the voltage regulator settings and should be removed for normal 5 V operation.

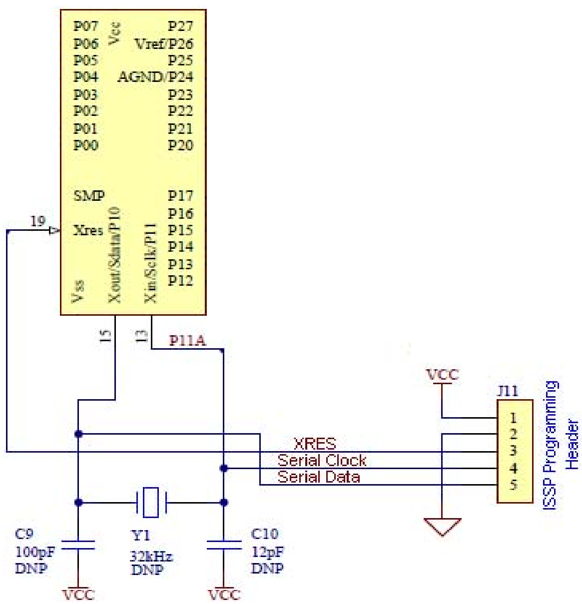
  When the jumper (shunt) is **inserted**, the board is regulated with a voltage of 3.3 V.

  When the jumper (shunt) is **removed**, the board is regulated with a normal 5-V operating voltage.

4.2.2 Programming Interface

This kit allows programming of the PSoC device via the ISSP programming header using a MiniProg.

Figure 4-6. Programming Interface Schematic



4.2.3 PSoC 1 Parts

Two parts are available with the CY3210-PSoCEVAL1 Evaluation Kit.

- PSoC CY8C29466-24PXI 28-Pin DIP
- PSoC CY8C27443-24PXI 28-Pin DIP

These parts incorporate 12-bit ADC, 6-bit DAC, flexible internal clock generators, 8-bit PWM, and 8-bit counter.

Table 4-1. Pin Description

Pin No.	Pin Name	Description	Connected To
1	P0[7]	Analog column mux input	J6.8
2	P0[5]	Analog column mux input and column output	J6.6
3	P0[3]	Analog column mux input and column output	J6.4
4	P0[1]	Analog column mux input	J6.2
5	P2[7]		J7.8
6	P2[5]		J7.6
7	P2[3]	Direct switched capacitor block input	J7.4
8	P2[1]	Direct switched capacitor block input	J7.2
9	SMP	Switch mode pump (SMP) connection to external components required	SMP
10	P1[7]	I^2C serial clock (SCL)	J8.8
11	P1[5]	I^2C serial data (SDA)	J8.6

Table 4-1. Pin Description *(continued)*

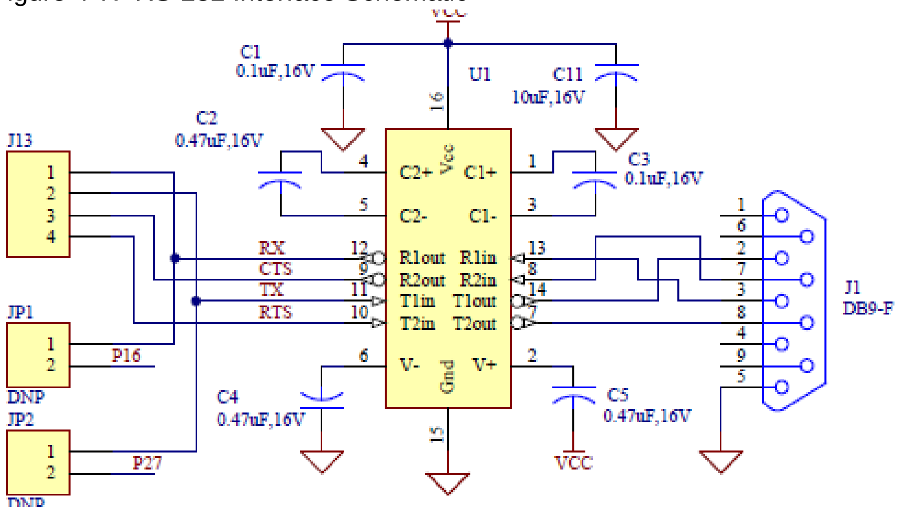
Pin No.	Pin Name	Description	Connected To
12	P1[3]		J8.4
13	P1[1]	Crystal (XTALin), I^{2}C Serial Clock (SCL), ISSP-SCLK	J8.2 and $XTAL_{IN}$ /Sclk
14	V_{SS}	Ground connection	V_{ss}
15	P1[0]	Crystal (XTALout), I^{2}C Serial Data (SDA), ISSP-SDATA	J8.1 and $XTAL_{out}$/ Sdata
16	P1[2]		J8.3
17	P1[4]	Optional external clock input (EXTCLK)	J8.5
18	P1[6]		J8.7
19	XRES	Active high external reset with internal pull-down	XRES/ TP4 DNP
20	P2[0]	Direct switched capacitor block input	J7.1
21	P2[2]	Direct switched capacitor block input	J7.3
22	P2[4]	External analog ground (AGND)	J7.5
23	P2[6]	External voltage reference (VREF)	J7.7
24	P0[0]	Analog column mux input	J6.1
25	P0[2]	Analog column mux input and column output	J6.3
26	P0[4]	Analog column mux input and column output	J6.5
27	P0[6]	Analog column mux input	J6.7
28	V_{DD}	Supply voltage	V_{CC}

4.2.4 RS-232 Interface

The RS-232 interface is a serial communication physical interface through which information transfers in or out one bit at a time. The board has an RS-232 transceiver for evaluation, using RS-232 (UART) for low-power designs. The RS-232 transceiver has a Tx and Rx configuration.

Supply voltage is 3.3 V to 5 V; output voltage Vout (High) is Vcc–0.6 V; and Vout (Low) is 0.4 V.

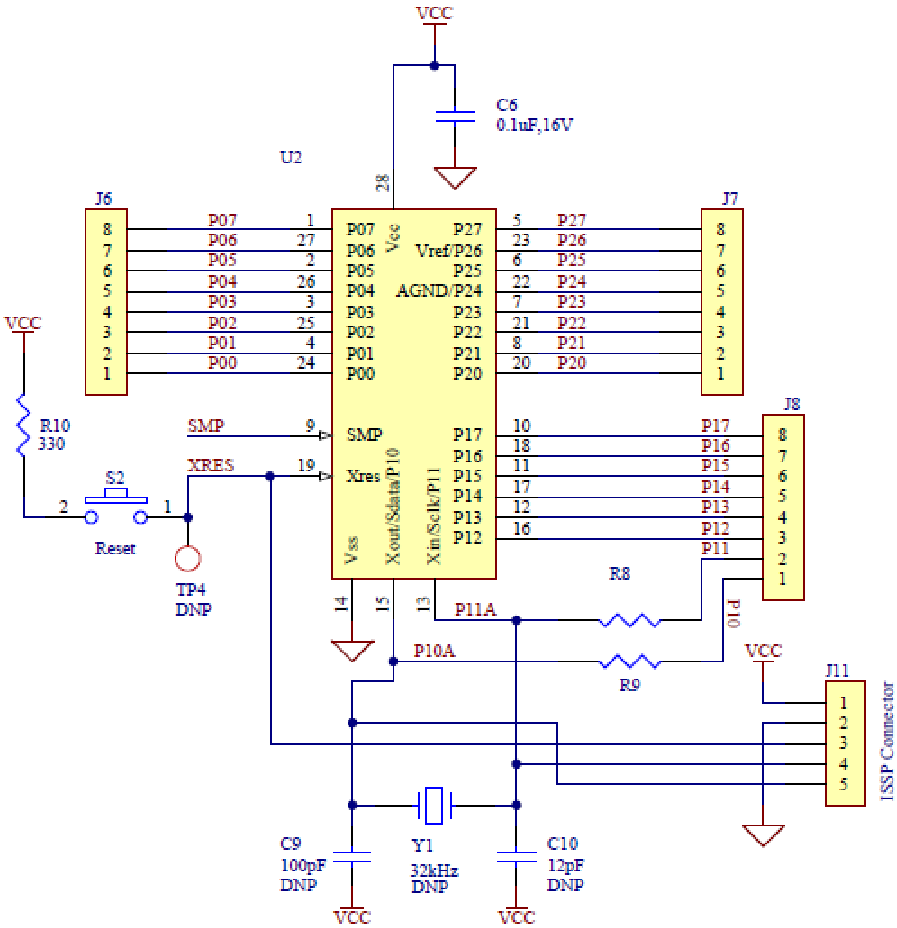
Figure 4-7. RS-232 Interface Schematic



4.2.5 Prototyping Area

The prototyping area has three complete ports for custom circuit development: port 0, port 1, and port 2. These ports can be used with the prototyping area to create simple yet elegant analog designs.

Figure 4-8. Prototyping Area Schematic

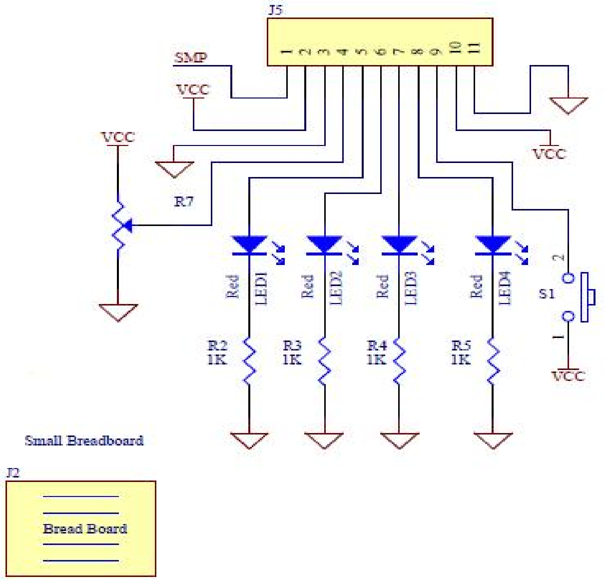


There are power and ground connections close to the prototyping area for convenience. The area also has four LEDs and two push button switches for application evaluation, including the Reset switch. This area also includes a potentiometer to be used for analog system evaluation work.

Note If the board needs to be powered via external 3.3-V or 5-V supply, the power input should be connected to the VCC sockets of the J5 connector. The external voltage needs to be less than 5.5 V.

Warning Any voltage input greater than 12 V connected on these headers will permanently damage the board components.

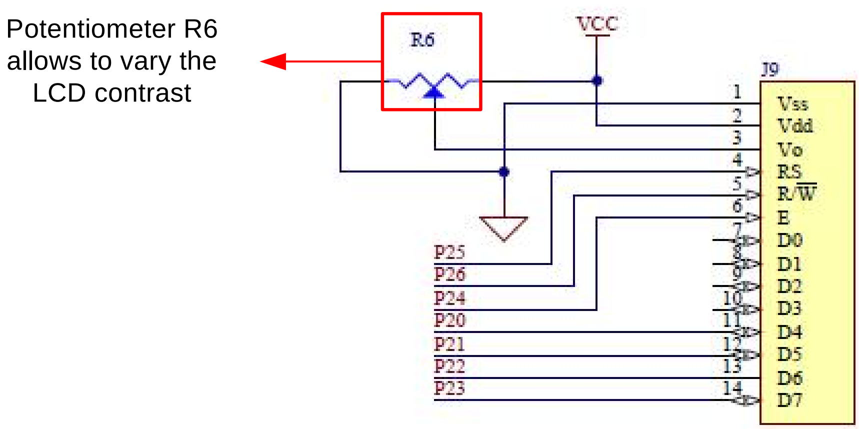
Figure 4-9. Prototyping Area Schematic



4.2.6 Character LCD Interface

The kit has a character 2×16 alphanumeric LCD module, which goes into the character LCD header, J9. The LCD runs on a 5-V supply and can function regardless of the voltage on which PSoC is powered.

Figure 4-10. LCD Interfacing Schematic



5. Code Examples

Six code examples are included in the following sections. All code examples are available on the CY3210-PSoCEVAL1 kit DVD or at this location: <Install_directory>:\Cypress\CY3210-PSoCEVAL1\<version>\Firmware.

5.1 My First Code Example

5.1.1 Project Description

This project demonstrates a 12-bit incremental ADC by measuring the voltage of the potentiometer, transmitting the conversion result on the UART, and displaying it on the LCD. The project uses the following modules:

ADCINC: This module processes the programmable gain amplifier (PGA) output at the rate of 180 samples per second and produces the corresponding digital output.

PGA: This module is used at unity gain to supply the input to ADC.

UART: This is an 8-bit universal asynchronous receiver transmitter (UART). The clock divider VC3 generates the baud clock for the UART by dividing 24 MHz by 156. The UART internally divides VC3 by 8, resulting in a baud rate of 19,200 bps. The ADC output is sent to the PC using UART module.

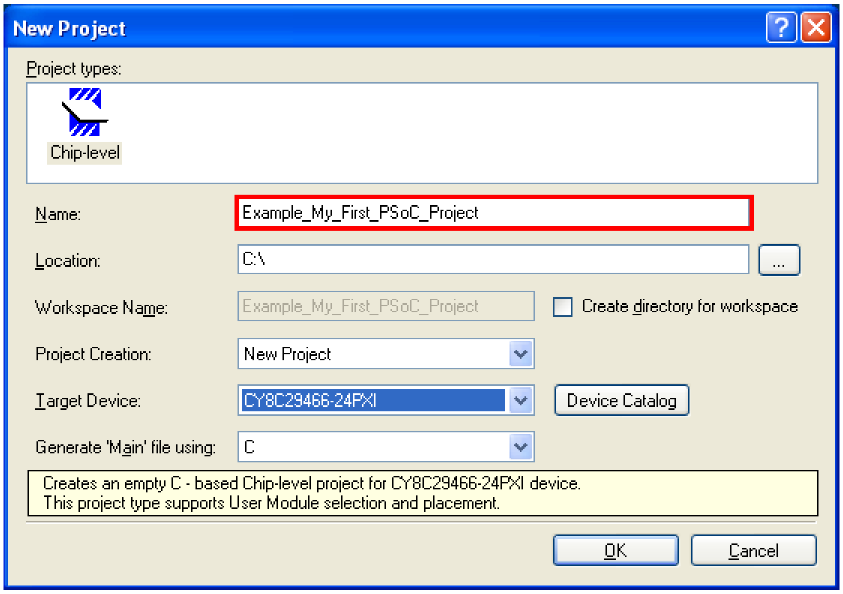
LCD: This module is used to display the ADC output (hex) values. If the ADC has completed conversion, the output is displayed on the LCD as ASCII text. The same is transmitted to the PC through a RS-232 cable.

Note This code example (**C_Example_ADC_UART_LCD**) is located in the Firmware folder.

5.1.2 Creating My First PSoC 1 Project

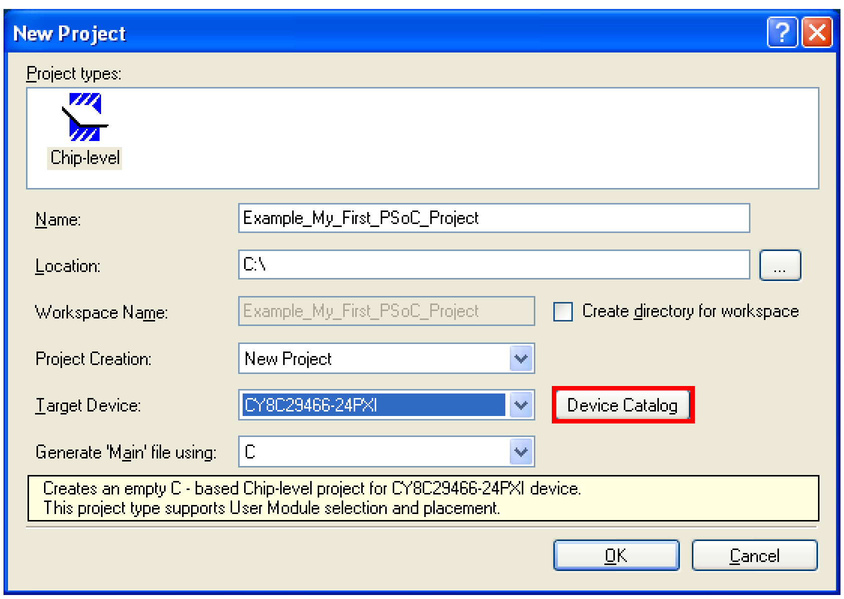
1. Open PSoC Designer.
2. To create a new project, click **File > New Project**.
3. In the New Project window, select the chip-level icon. Name the project **Example_My_First_PSoC_Project,** as shown in Figure 5-1.
4. In the **Location** field, enter the path in which the project is to be created.

Figure 5-1. New Project Window



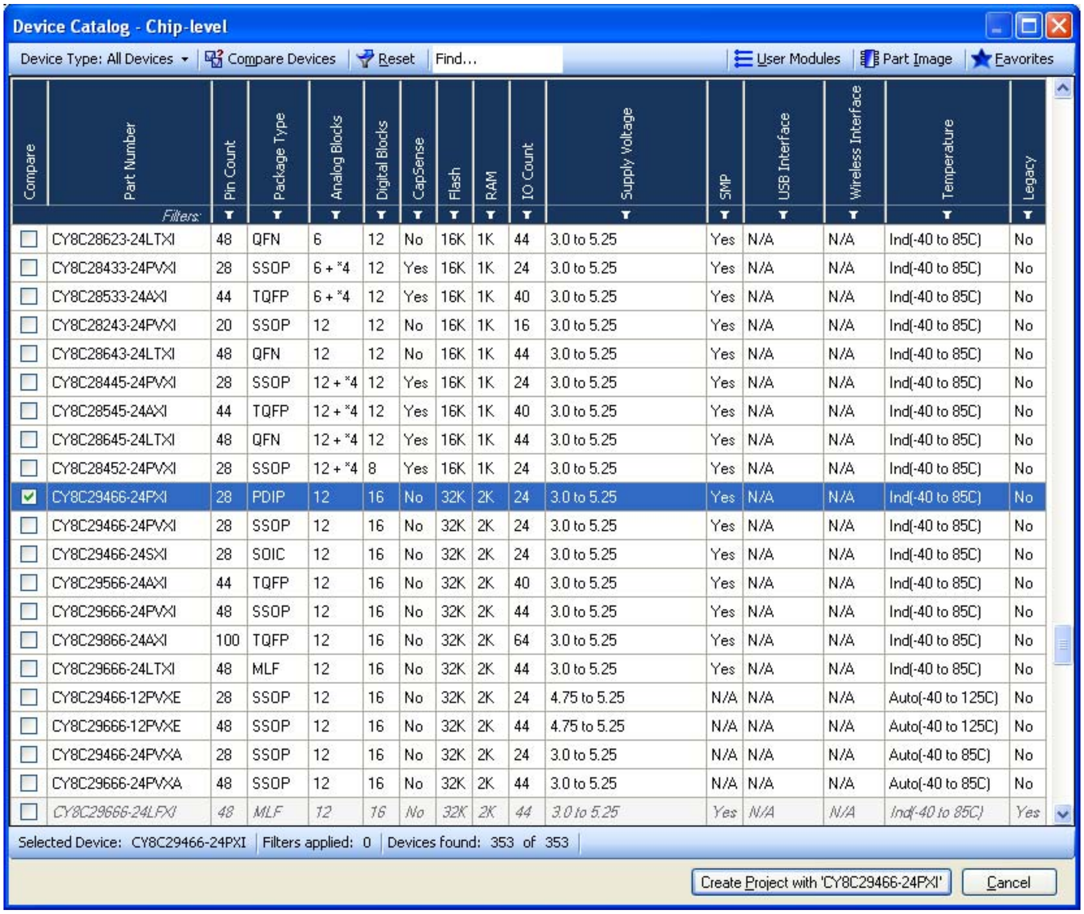
5. To select the target device, click **Device Catalog**, as shown in Figure 5-2.

Figure 5-2. Select Device Catalog



6. The Device Catalog window opens. Select **CY8C29466-24PXI** from the list and click **Create Project with 'CY8C29466-24PXI'**.

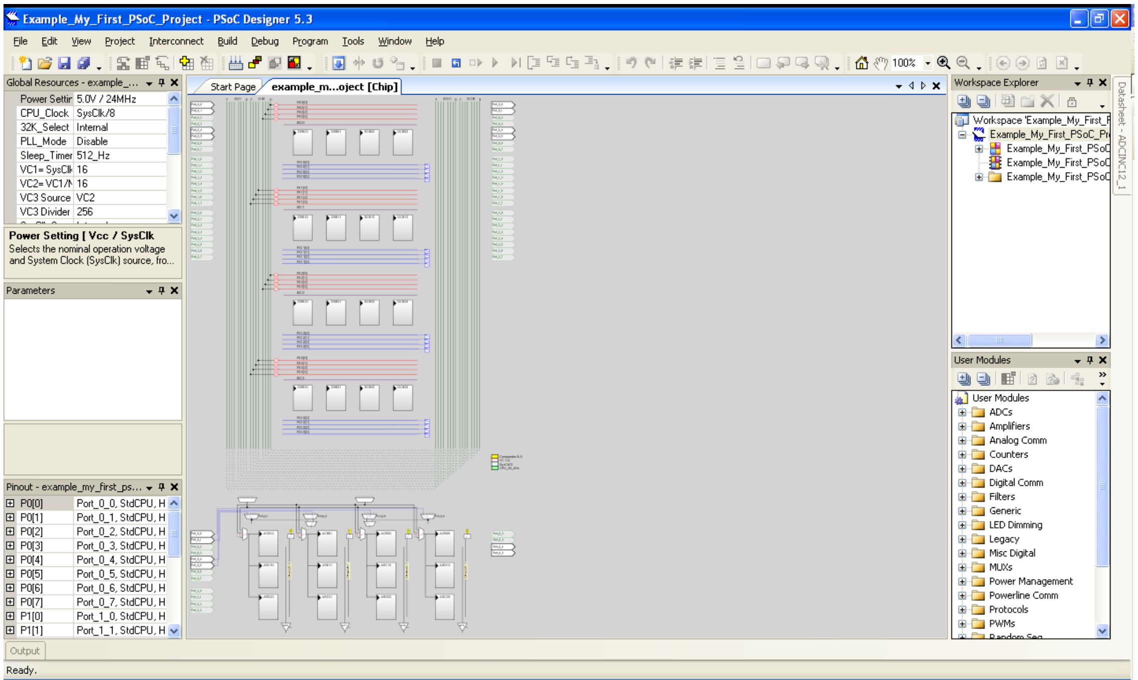
Figure 5-3. Device Catalog Window



Compare	Part Number	Pin Count	Package Type	Analog Blocks	Digital Blocks	CapSense	Flash	RAM	IO Count	Supply Voltage	SMP	USB Interface	Wireless Interface	Temperature	Legacy
	CY8C28623-24LTXI	48	QFN	6	12	No	16K	1K	44	3.0 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)	No
	CY8C28433-24PVXI	28	SSOP	6 + *4	12	Yes	16K	1K	24	3.0 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)	No
	CY8C28533-24AXI	44	TQFP	6 + *4	12	Yes	16K	1K	40	3.0 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)	No
	CY8C28243-24PVXI	20	SSOP	12	12	No	16K	1K	16	3.0 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)	No
	CY8C28643-24LTXI	48	QFN	12	12	No	16K	1K	44	3.0 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)	No
	CY8C28445-24PVXI	28	SSOP	12 + *4	12	Yes	16K	1K	24	3.0 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)	No
	CY8C28545-24AXI	44	TQFP	12 + *4	12	Yes	16K	1K	40	3.0 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)	No
	CY8C28645-24LTXI	48	QFN	12 + *4	12	Yes	16K	1K	44	3.0 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)	No
	CY8C28452-24PVXI	28	SSOP	12 + *4	8	Yes	16K	1K	24	3.0 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)	No
✓	CY8C29466-24PXI	28	PDIP	12	16	No	32K	2K	24	3.0 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)	No
	CY8C29466-24PVXI	28	SSOP	12	16	No	32K	2K	24	3.0 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)	No
	CY8C29466-24SXI	28	SOIC	12	16	No	32K	2K	24	3.0 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)	No
	CY8C29566-24AXI	44	TQFP	12	16	No	32K	2K	40	3.0 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)	No
	CY8C29666-24PVXI	48	SSOP	12	16	No	32K	2K	44	3.0 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)	No
	CY8C29866-24AXI	100	TQFP	12	16	No	32K	2K	64	3.0 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)	No
	CY8C29666-24LTXI	48	MLF	12	16	No	32K	2K	44	3.0 to 5.25	Yes	N/A	N/A	Ind(-40 to 85C)	No
	CY8C29466-12PVXE	28	SSOP	12	16	No	32K	2K	24	4.75 to 5.25	N/A	N/A	N/A	Auto(-40 to 125C)	No
	CY8C29666-12PVXE	48	SSOP	12	16	No	32K	2K	44	4.75 to 5.25	N/A	N/A	N/A	Auto(-40 to 125C)	No
	CY8C29466-24PVXA	28	SSOP	12	16	No	32K	2K	24	3.0 to 5.25	N/A	N/A	N/A	Auto(-40 to 85C)	No
	CY8C29666-24PVXA	48	SSOP	12	16	No	32K	2K	44	3.0 to 5.25	N/A	N/A	N/A	Auto(-40 to 85C)	No
	CY8C29666-24LFXI	*48*	*MLF*	*12*	*16*	*No*	*32K*	*2K*	*44*	*3.0 to 5.25*	*Yes*	*N/A*	*N/A*	*Ind(-40 to 85C)*	*Yes*

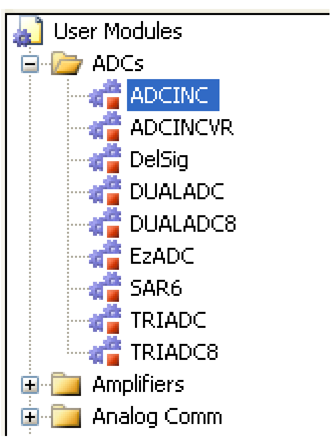
7. In the **Generate 'Main' File Using:** option, select **C** and click **OK**.
8. By default, the project opens in Chip view, as shown in Figure 5-4.

Figure 5-4. Default View



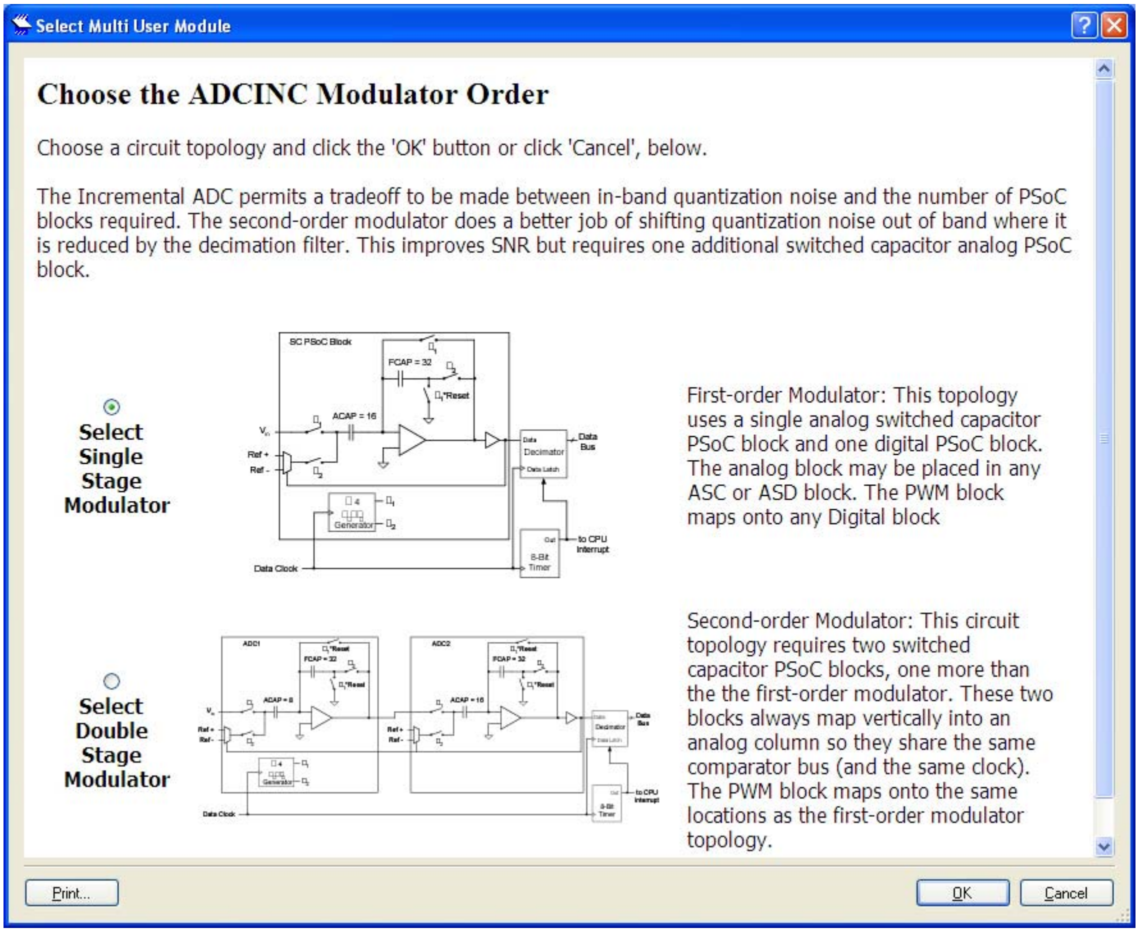
9. Configure the modules required for this design. Also, connect the modules together and to the pins on the PSoC. In the User Modules section, expand the **ADCs** folder.

Figure 5-5. User Modules Window



10. In this folder, right-click on **ADCINC** and select **Place**. Choose the **Single Stage Modulator** option for the Multi User Module dialog.

Figure 5-6. Multi User Module



The user module (UM) is placed in the first available analog block. By default, the UM is placed on an Analog Switched Capacitor Type C block. It should be shifted to a Switched Capacitor Type D block, as explained in the next step.

Note The analog block in ADCINC is configured as an integrator. Switched Capacitor Type C is a filter and Type D is an integrator. Refer to the ADCINC UM data sheet and Technical Reference Manual from **Help** > **Documentation** for more details on analog blocks.

11. To change the UM placement, click on **Next Allowed Placement** icon (see Figure 5-7). The next possible module for placement is highlighted. Click on the icon again so that the ASD11 analog block is highlighted. Now, click on the **Place User Module** icon (see Figure 5-7) to place the user module in ASD11 analog block.

Figure 5-7. PSoC Designer Toolbar

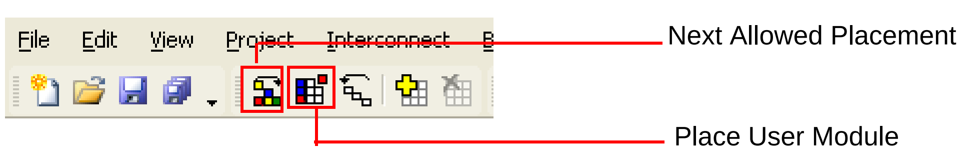
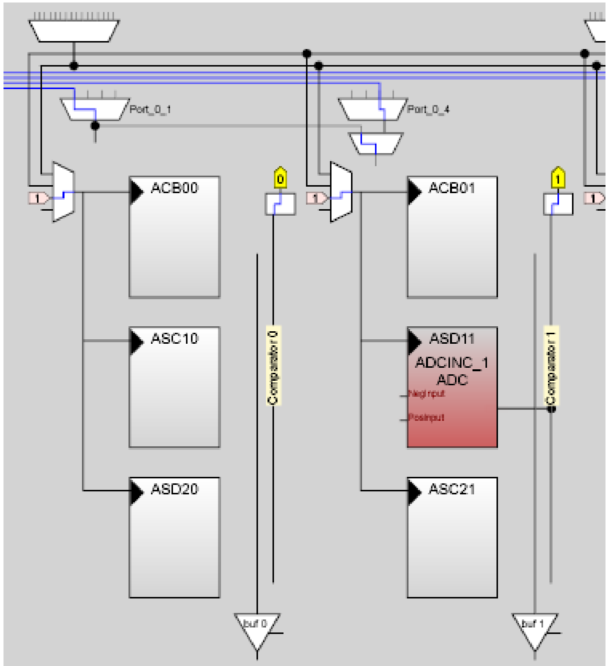


Figure 5-8. ADCINC User Module Placement



12. Configure the ADCINC_1 properties, as shown in Figure 5-9.

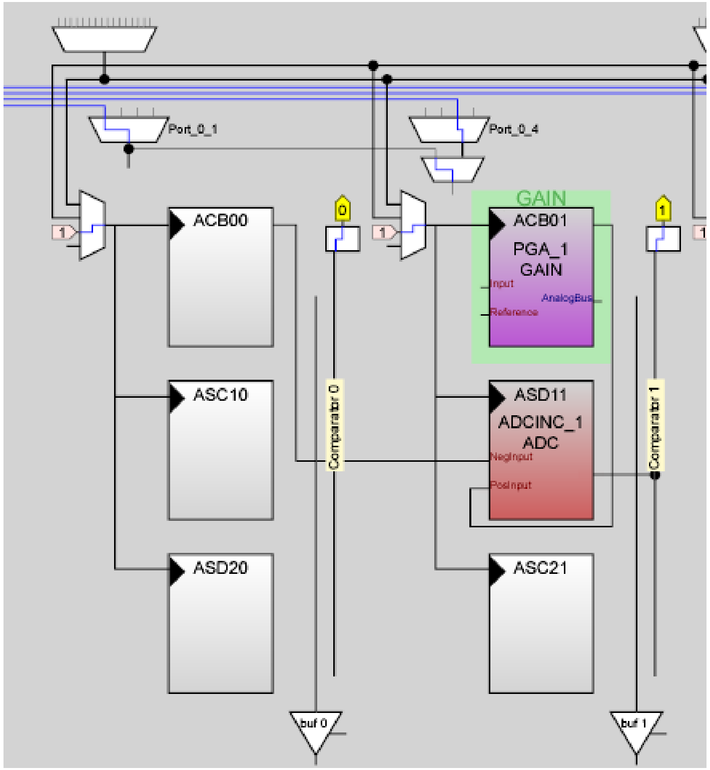
Figure 5-9. ADCINC User Module Properties

Parameters - ADCINC_1	
Name	ADCINC_1
User Module	ADCINC
Version	1.20
DataFormat	Unsigned
Resolution	12 Bit
Data Clock	VC1
ClockPhase	Normal
PosInput	ACB01
NegInput	ACB00
NegInputGain	Disconnected
PulseWidth	1
PWM Output	None

13. In the User Modules window, expand the **Amplifiers** folder and double-click on **PGA** to place a programmable gain amplifier (PGA) in the design.
14. By default, the PGA is placed on the ACB00 analog block. To change the placement to ACB01 analog block, click on the **Next Allowed Placement** icon (see Figure 5-7).

    Note Move the PGA User Module to a block above the ADCINC UM. This is because, the input to the ADCINC UM is routed through a PGA.

Figure 5-10. PGA User Module Placement



15. Configure the PGA properties, as shown in Figure 5-11.

Figure 5-11. PGA User Module Properties

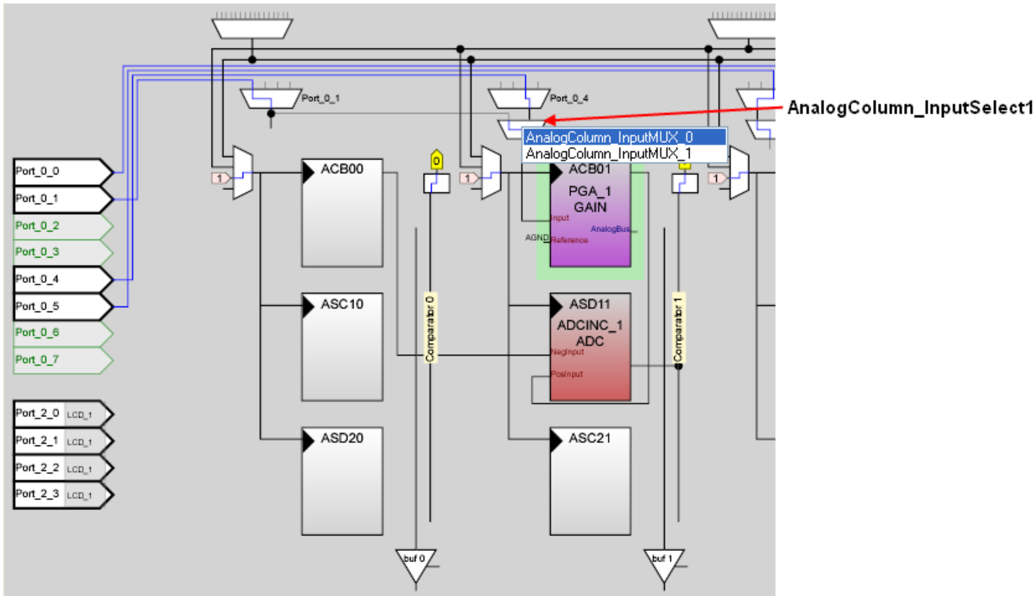
Parameters - PGA_1

Name	PGA_1
User Module	PGA
Version	3.2
Gain	1.000
Input	AnalogColumn_InputSelect_1
Reference	AGND
AnalogBus	Disable

AnalogBus

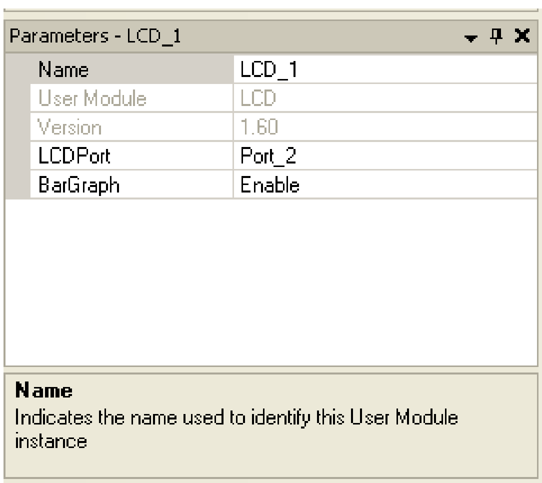
16. Click on **AnalogColumn_InputSelect_1** multiplexer in the Design window and change the input multiplexer to AnalogColumn_InputMUX_0. The analog input from P0[1] is routed to the ADC module through a PGA at unity gain (Gain is a configurable parameter in the PGA module).

Figure 5-12. Connecting PGA to Port Pin



17. In the User Modules window, expand the **Misc Digital** folder and double-click on **LCD** to place an LCD in the design. Configure the properties of the LCD UM. Enable the **BarGraph** property to include additional APIs into the project and allow the bar graph display on the LCD.

Figure 5-13. LCD User Module Properties


Parameters - LCD_1

Name	LCD_1
User Module	LCD
Version	1.60
LCDPort	Port_2
BarGraph	Enable

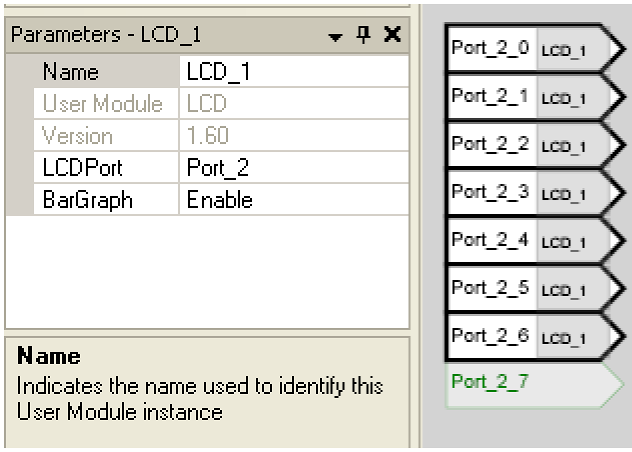
Name
Indicates the name used to identify this User Module instance

Notes

a. This UM does not use digital or analog blocks. On mapping the LCD to a port, it can be viewed in the design, as shown in Figure 5-14.

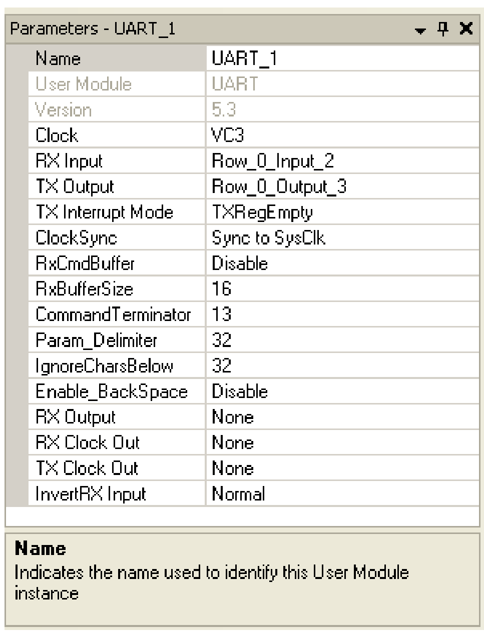
b. The LCD module can be placed on ports 0, 1, or 2. Port 2 is selected because some of the port pins of port 0 and port 1 have other uses.

Figure 5-14. LCD User Module Placement



18. In the User Modules window, expand the **Digital Comm** folder and double-click on **UART** to place a UART in the design. Configure the properties of the UART UM, as shown in Figure 5-15.

Figure 5-15. UART User Module Properties



Parameters - UART_1

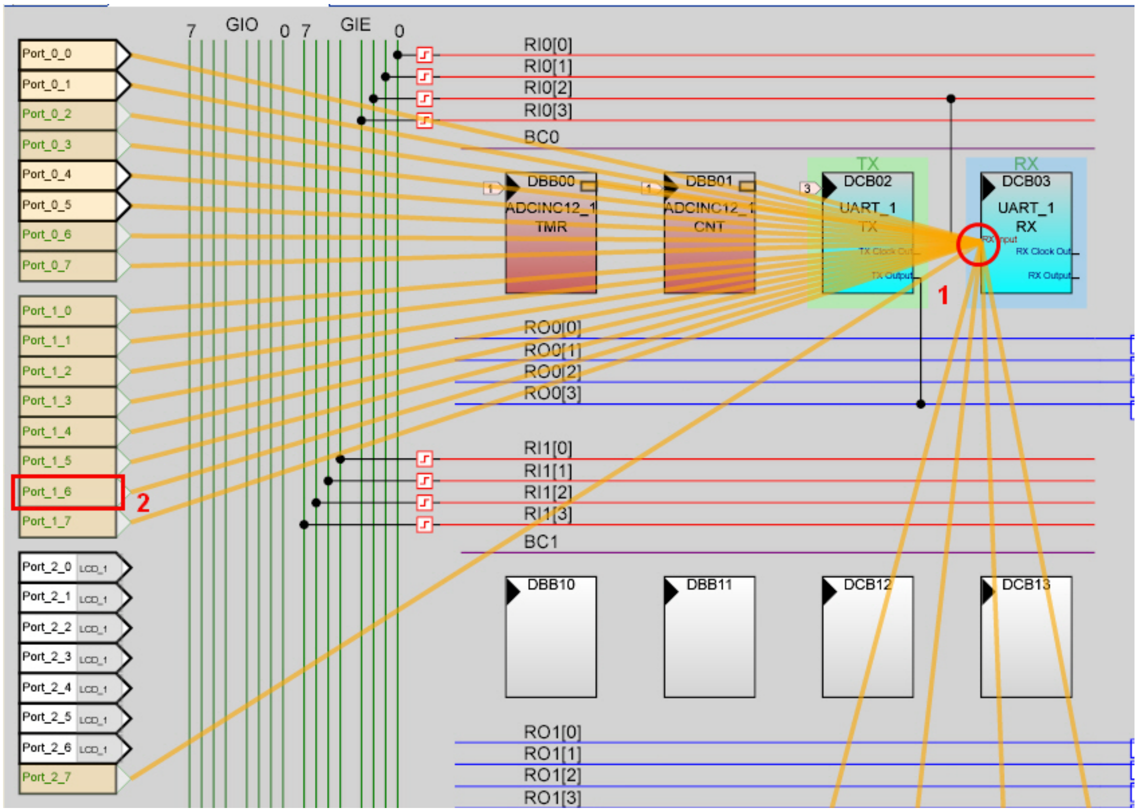
Name	UART_1
User Module	UART
Version	5.3
Clock	VC3
RX Input	Row_0_Input_2
TX Output	Row_0_Output_3
TX Interrupt Mode	TXRegEmpty
ClockSync	Sync to SysClk
RxCmdBuffer	Disable
RxBufferSize	16
CommandTerminator	13
Param_Delimiter	32
IgnoreCharsBelow	32
Enable_BackSpace	Disable
RX Output	None
RX Clock Out	None
TX Clock Out	None
InvertRX Input	Normal

Name
Indicates the name used to identify this User Module instance

19. Route the RX signal of UART to P1[6]. There are two methods to do this:

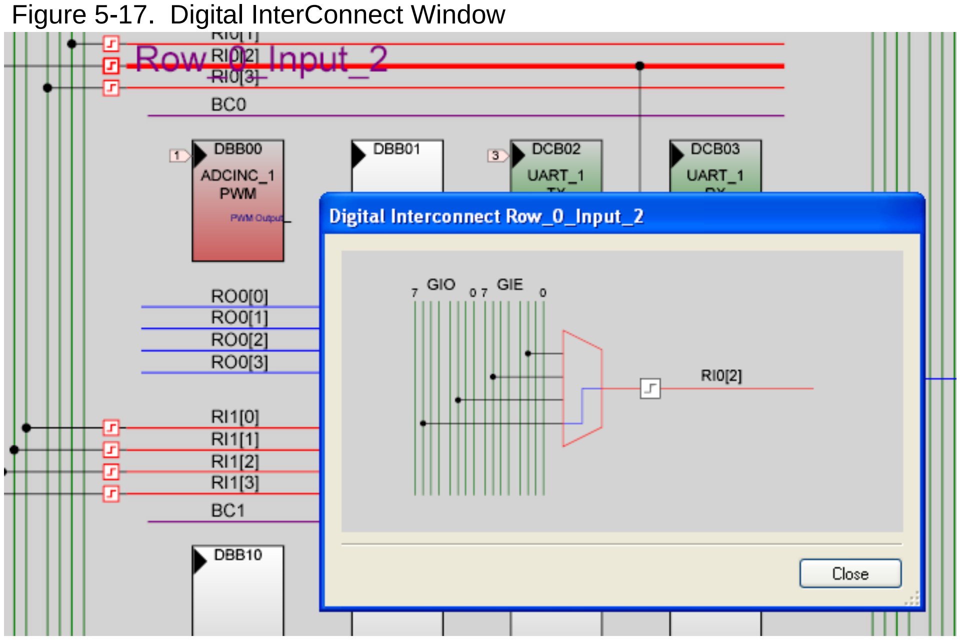
a. Auto Routing: While holding the **[Shift]** key on the keyboard, click on the **RX Input** terminal. PSoC Designer highlights the available pins/terminals to which routing is possible. Without releasing the [Shift] key, click on the **Port_1_6** pin on the left. The Row and Column interconnects are automatically configured to connect the selected terminals.

Figure 5-16. Auto Routing



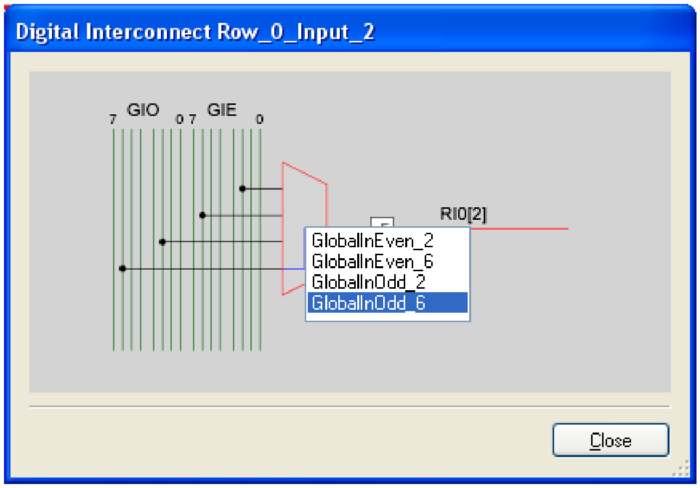
b. Manual Routing: Configure the look-up table (LUT) on Row_0_Input2 to GlobalOddEven bus. To do so, click on the **Row_0_Input_2** bus to open the Digital Interconnect window.

Figure 5-17. Digital InterConnect Window



20. Click on **Row_0_Input_2** demultiplexer in the Interconnect window and select **GlobalInOdd_6**; click **Close**.

Figure 5-18. Configure Row_0_Input_2 to GlobalInOdd_6



21. Click on **GlobalInOdd_6**. Select **Port_1_6** from the drop-down list in the Pin field; click **OK**.

Figure 5-19. Pin Select

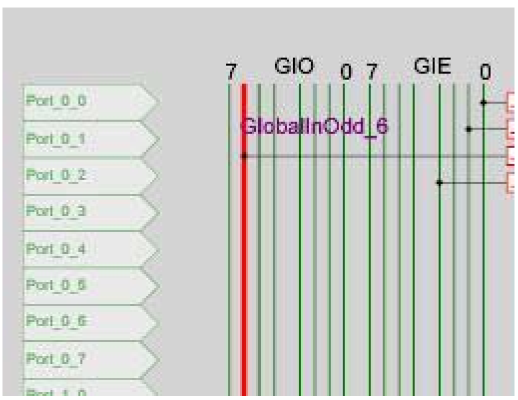
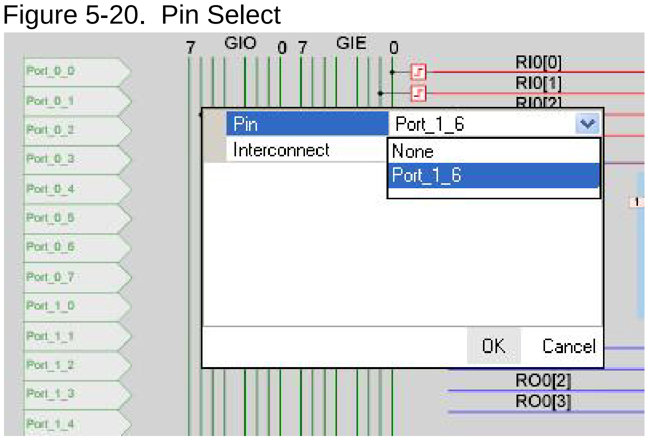


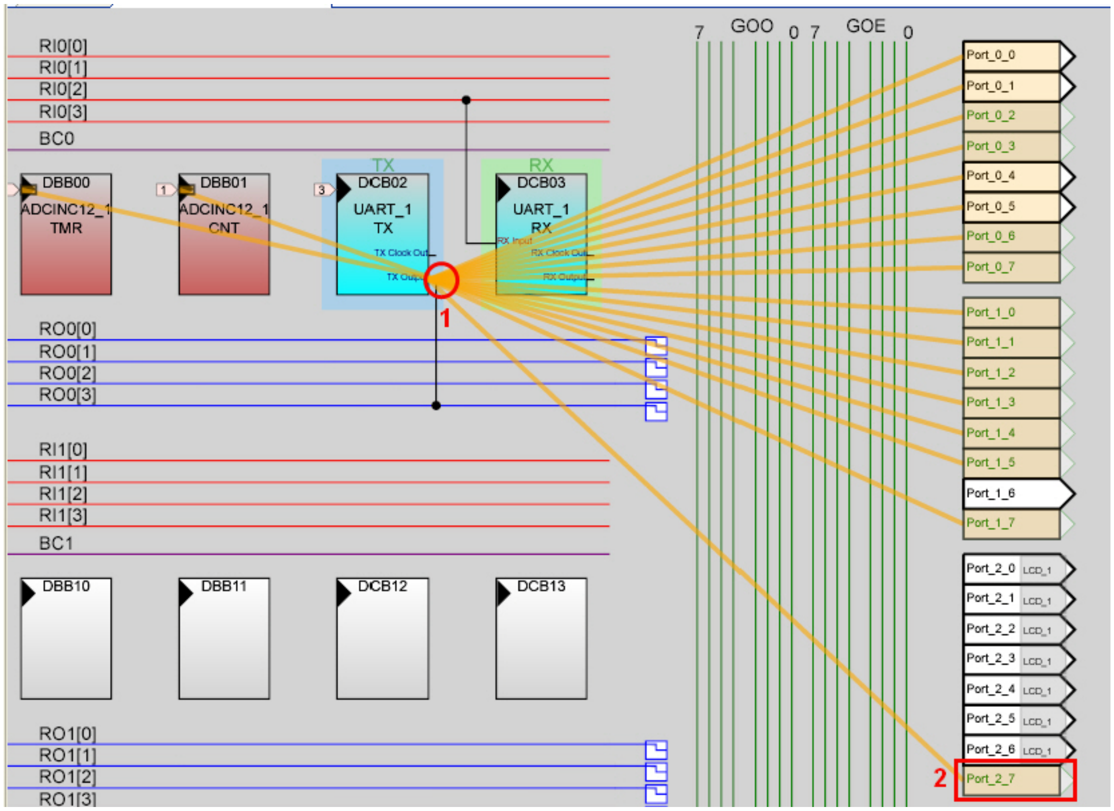
Figure 5-20. Pin Select



22. Route the TX signal of UART to P2[7]. There are two methods to do this:

a. Auto Routing: Press the **[Shift]** key on the keyboard and click on the **TX Output** terminal. Without releasing the [Shift] key, click on the **Port_2_7** pin on the right hand side.

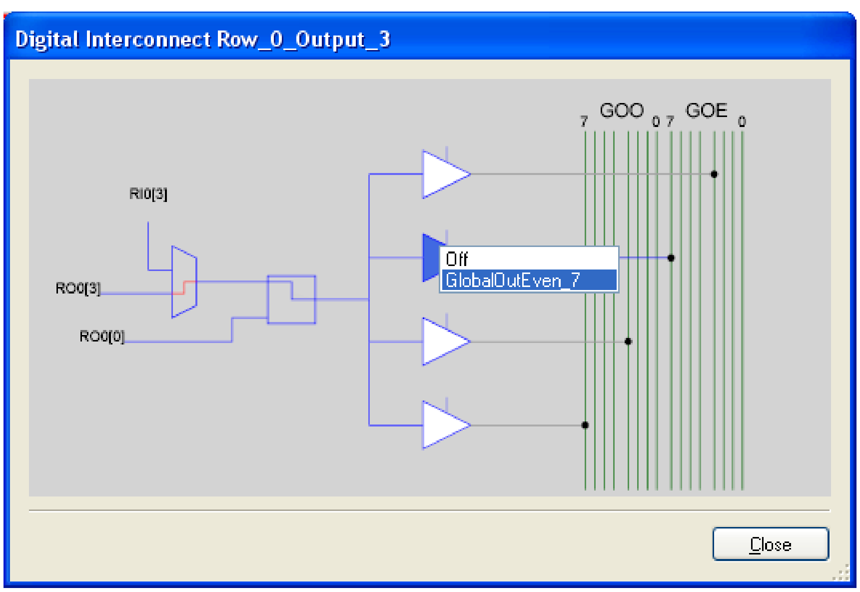
Figure 5-21. Auto-Routing



b. Manual Routing: Configure the LUT on Row_0_Output3. To do so, click on **Row_0_Output3** to open the Digital Interconnect window.

23. In this window, enable **Row_0_Output_3_Drive_1** to connect to GlobalOutEven_7.

Figure 5-22. Digital InterConnect Window



24. Click **Close**.

25. Click on **GlobalOutEven_7**. Select **Port_2_7** from the drop-down list in the Pin field; click **OK.**

Figure 5-23. Pin Select

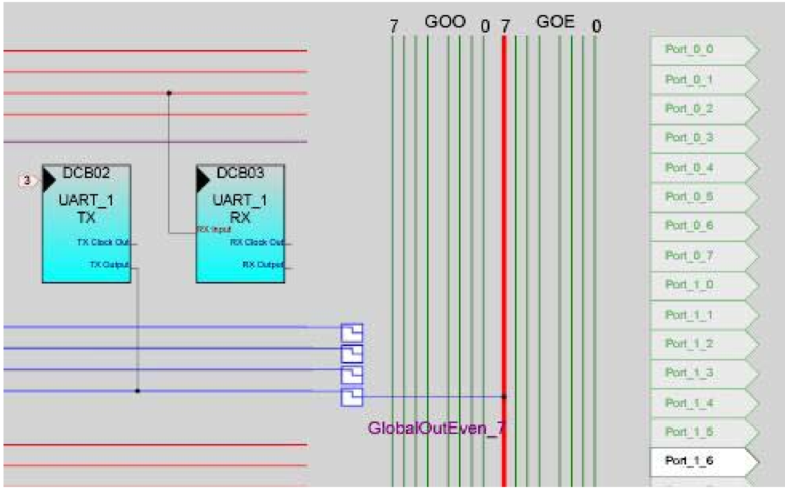
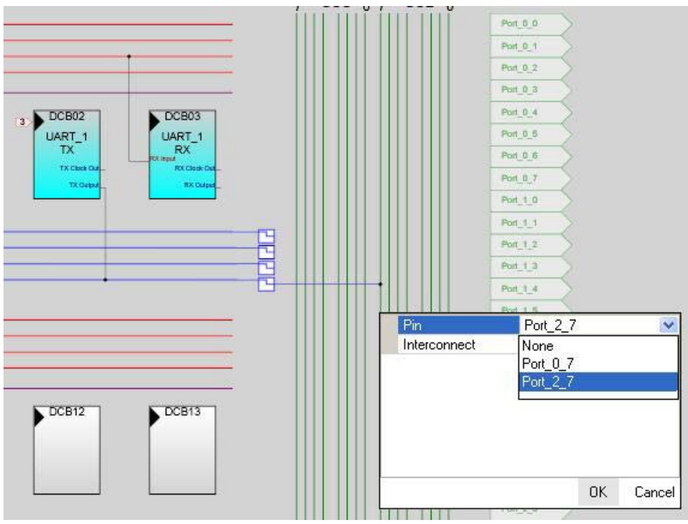
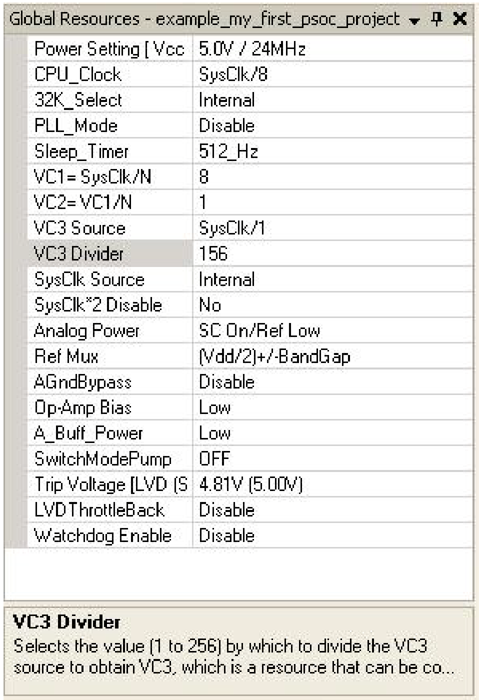


Figure 5-24. Pin Select



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

Figure 5-25. Global Resources Window



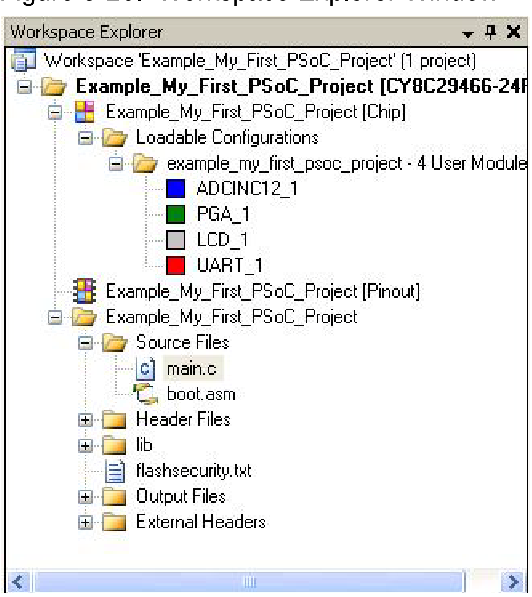
Note

a. The clock divider VC1 provides a 3-MHz sample clock to the ADCINC, resulting in a sample rate of 180 samples per second.

b. The clock divider VC3 generates the baud clock for UART by dividing 24 MHz by 156. The UART internally divides UART clock (VC3 in this example) by 8, resulting in a baud rate of 19200 bits per second. See the UART UM data sheet for details.

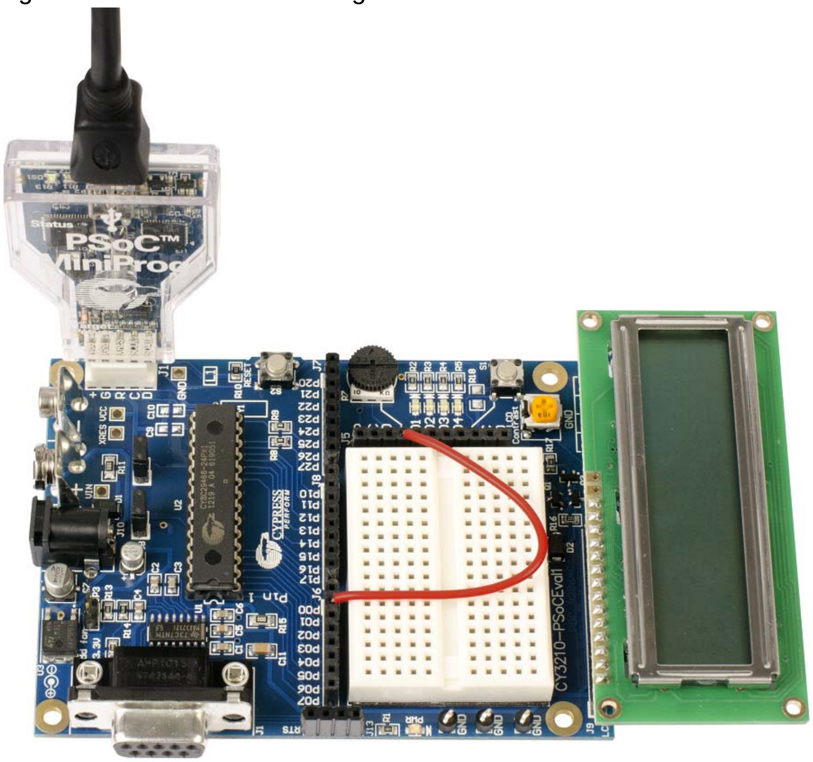
27. Open the existing *main.c* file in Workspace Explorer. Replace the existing *main.c* content with the content of the embedded *My_First_Example_Project_Main.c* file, which is available within the attachments feature of this PDF document.

Figure 5-26. Workspace Explorer Window



28. Save the project.
29. To build the project, click **Build > Generate/Build 'Example_My_First_PSoC_Project'**.
30. Connect the CY3210 PSoCEVAL1 board to a PC through a MiniProg1.

Figure 5-27. Connect MiniProg1 to Board

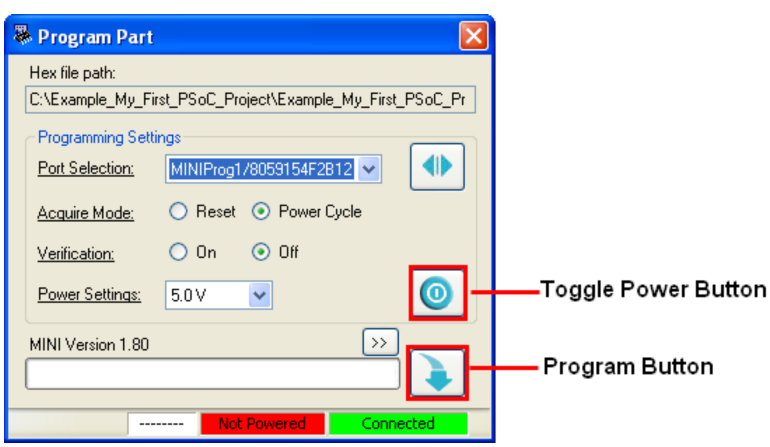


The board can be programmed either through PSoC Designer IDE or by launching PSoC Programmer. To program the board using PSoC Programmer, see Programming Specifications and Connections on page 16. To program the board through PSoC Designer, follow these steps.

Note When programming the board through PSoC Designer, close any open instance of PSoC Programmer.

a. Click on **Program > Program Part**.

Figure 5-28. Program Part Window

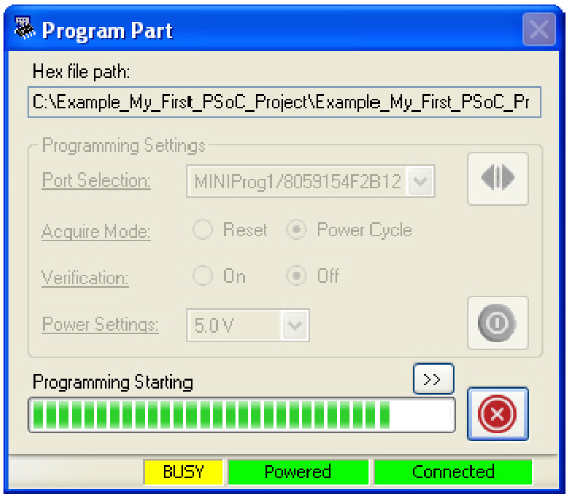


b. In the Program Part window, configure the following settings:
   Port Selection: Select **MiniProg1/xxxxxxxxxxx** and then **Connected**
   Acquire Mode: **Power Cycle**
   Verification: **Off**
   Power Settings: **5.0 V**

c. Click on the **Program** button (see Figure 5-28) to start programming the board.

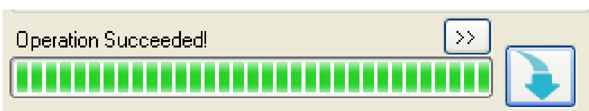
d. Observe the programming status on the progress bar.

Figure 5-29. Programming Status



e. When programming is successful, the 'Operation Succeeded!' message is displayed.

Figure 5-30. Operation Succeeded!' Message

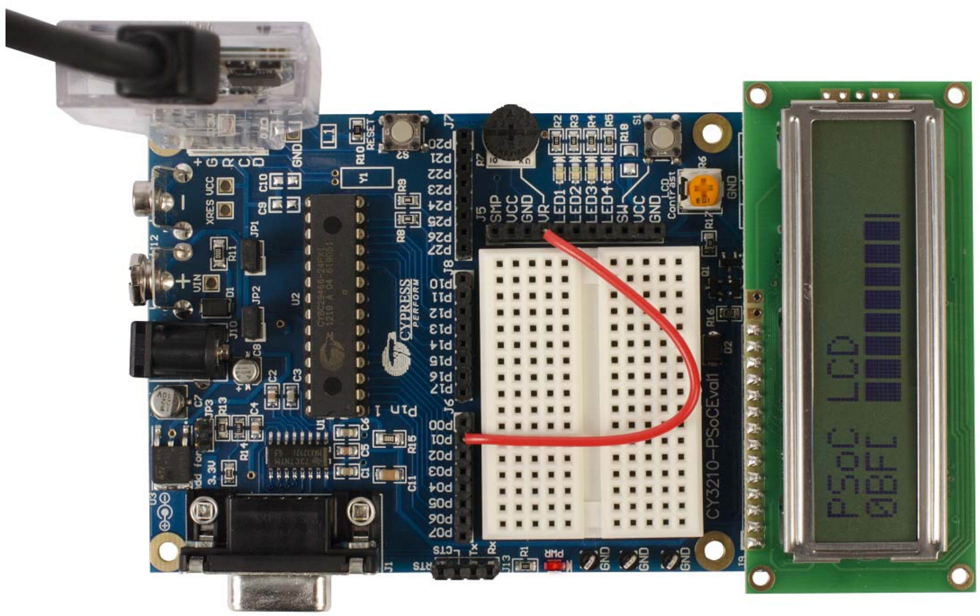


5.1.3 Verifying Output using LCD

1. Set up the board with the following connections using the jumpers (shunts) and single strand jumper wires:
   a. Connect P01 to VR using a single strand jumper wire. This connects one of the PSoC pins to the potentiometer.
   b. Place jumper (shunt) on JP1 to connect P16 and Rx.
   c. Place jumper (shunt) on JP2 to connect P27 and Tx.
   d. Connect an RS-232 cable from connector J1 to a COM port on the PC.
   e. Remove jumper (shunt) JP3 to operate the board at 5 V. The LCD display is seen for 5 V operation only.
2. Power the board by clicking on the **Toggle Power** button (see Figure 5-28) in the Program Part window.
3. The ADC value is shown on the LCD display. A bar graph corresponding to the ADC value is also displayed. The ADC value varies from 0000-0FFF for input voltage of 0 V to 5 V. The input voltage can be varied by rotating the potentiometer (R7) connected to VR.

Note The measured value might have an error of upto 10 counts due to ADC offset or potentiometer inaccuracy.

Figure 5-31. LCD Displaying ADC Value



4. Vary the potentiometer and observe the change in the value of LCD.

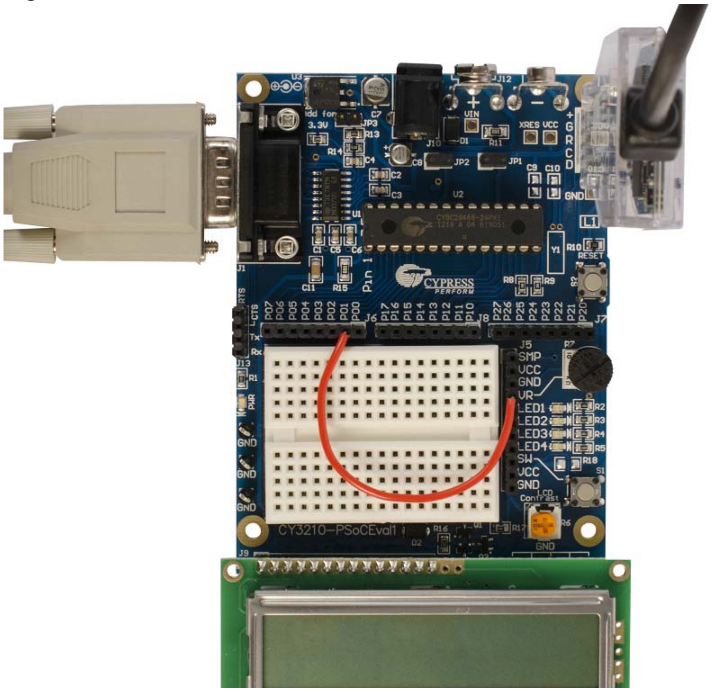
   Note ADC values may fluctuate several counts due to system noise or if the potentiometer voltage is at the edge of an ADC count.

5. Save and close the project.

5.1.4 Verifying Output using UART

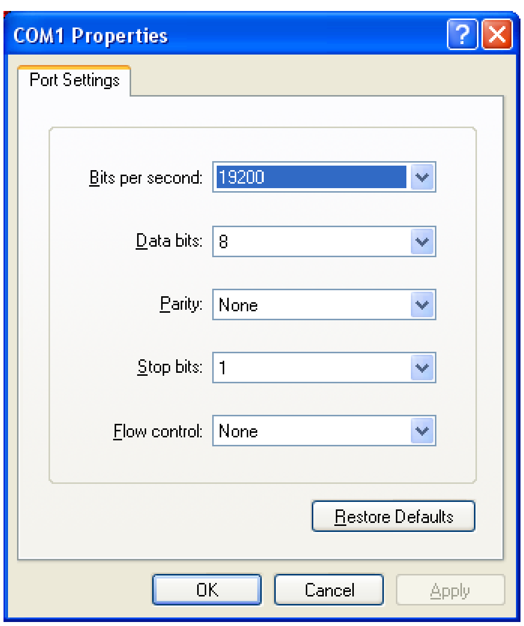
The PSoC project in Creating My First PSoC 1 Project on page 27 uses an UART module in the design. The ADC value that is displayed on the LCD can be viewed on a terminal application such as HyperTerminal or TeraTerm on a PC. To view this output, apart from the hardware connections in Verifying Output using LCD on page 42, a RS-232 cable needs to be connected from the CY3210-PSoCEVAL1 board to a COM port on a PC, as shown in Figure 5-32.

Figure 5-32. CY3210-PSoCEVAL1 Connected to RS-232 Cable



1. Connect the hardware as explained in Verifying Output using LCD on page 42 section.
2. Open a terminal application such as HyperTerminal or TeraTerm with these parameters:
   a. Baud Rate: **19200**
   b. Data: **8-bit**
   c. Parity: **None**
   d. Stop: **1-bit**
   e. Flow Control: **None**

   Figure 5-33. HyperTerminal Settings



3. Power the board by clicking on the **Toggle Power** button in the Program Part window
4. The ADC value is displayed on the HyperTerminal and on the LCD, as shown in Figure 5-34 and Figure 5-35

Figure 5-34. Verify Output on LCD

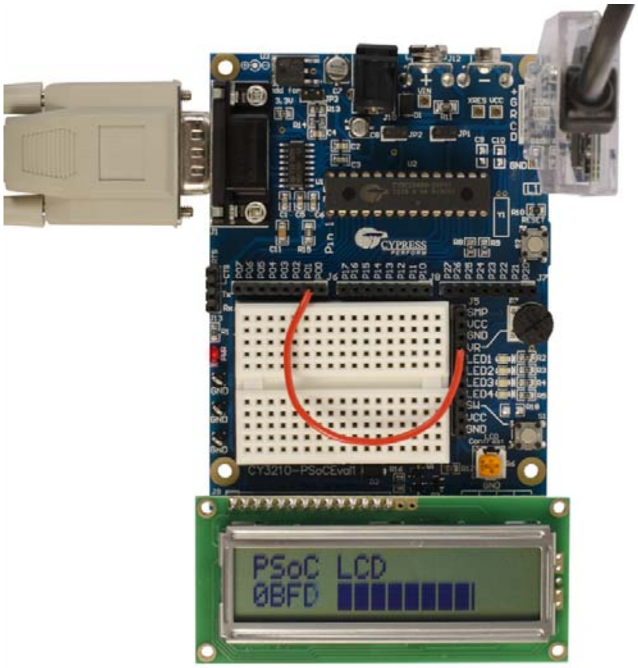
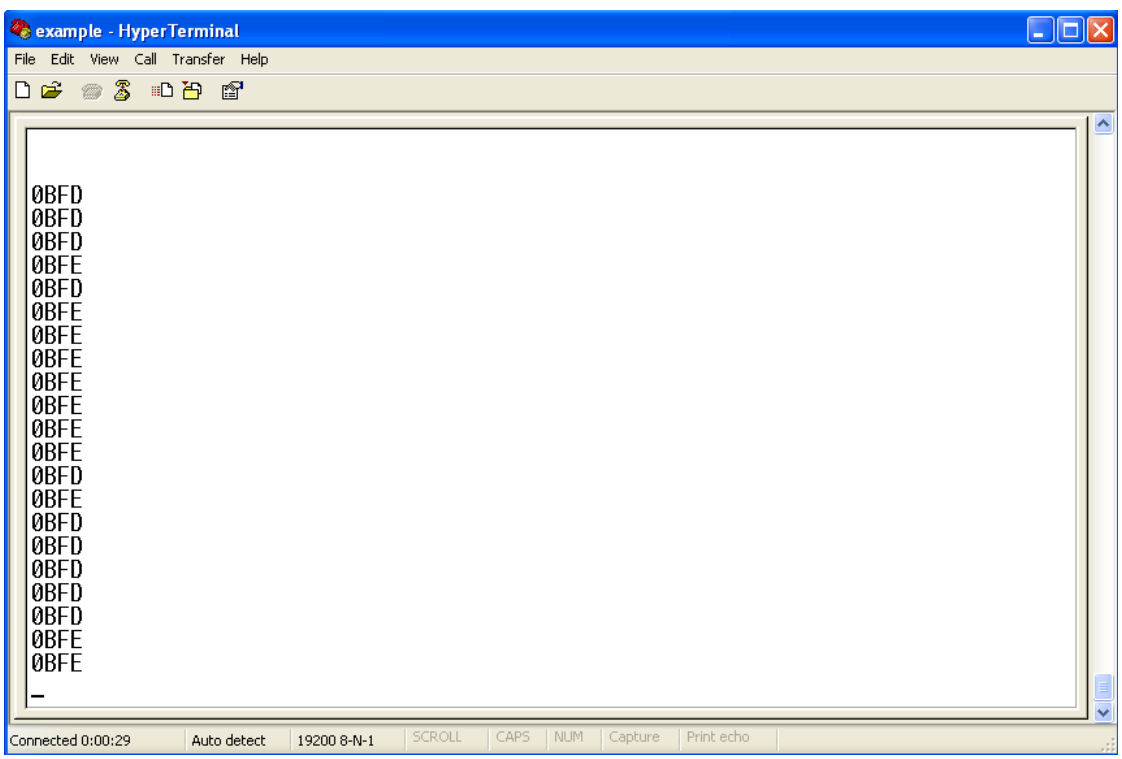


Figure 5-35. Verify Output on HyperTerminal



5.2 Code Example 2: ASM_Example_ADC_UART_LCD

5.2.1 Project Description

This project demonstrates a 12-bit incremental ADC by measuring the voltage of the potentiometer, transmitting the conversion result on the UART, and displaying it on the LCD. The project uses the following modules:

ADCINC: This module processes the programmable gain amplifier (PGA) output at the rate of 180 samples per second and produces the corresponding digital output.

PGA: This module is used at unity gain to supply the input to ADC.

UART: This is an 8-bit universal asynchronous receiver transmitter (UART). The clock divider VC3 generates the baud clock for the UART by dividing 24 MHz by 156. The UART internally divides VC3 by 8, resulting in a baud rate of 19,200 bps. The ADC output is sent to the PC using UART module.

LCD: This module is used to display the ADC output (hex) values. If the ADC has completed conversion, the output is displayed on the LCD as ASCII text. The same is transmitted to the PC through a RS-232 cable.

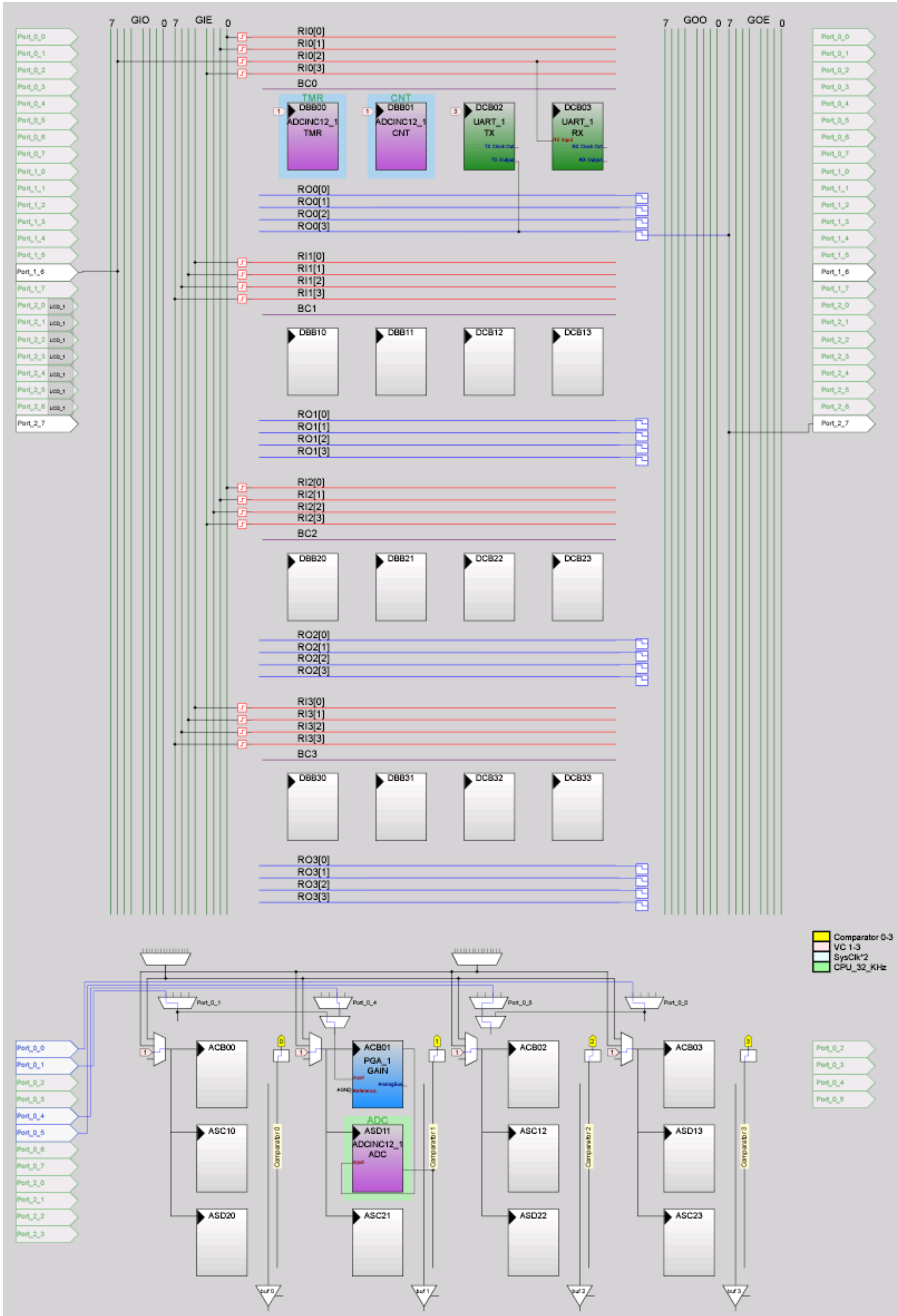
5.2.2 Creating the Project

The procedure for creating this project is similar to the flow described in 5.1.2 Creating My First PSoC 1 Project. The previous example demonstrated coding in C language, while this example implements the same functionality in Assembly language. The LCD bargraph function is disabled to keep the code compact.

To create the project:

1. Follow steps 1 to 8 from 5.1.2 Creating My First PSoC 1 Project.
2. In the **Generate 'Main' File Using:** option, select **Assembly** and click **OK**.
3. Follow steps 10 through 28 from 5.1.2 Creating My First PSoC 1 Project.
4. Open the existing *main.asm* file in Workspace Explorer. Replace the existing *main.asm* content with the content from the file located at <Install_directory>:\Cypress\CY3210-PSoCEVAL1\<version>\Firmware\ASM_Example_ADC_UART_LCD\ASM_Example_ADC_UART_LCD\main.asm.
5. Follow steps 30 through 32 from 5.1.2 Creating My First PSoC 1 Project to build the project and program the device.

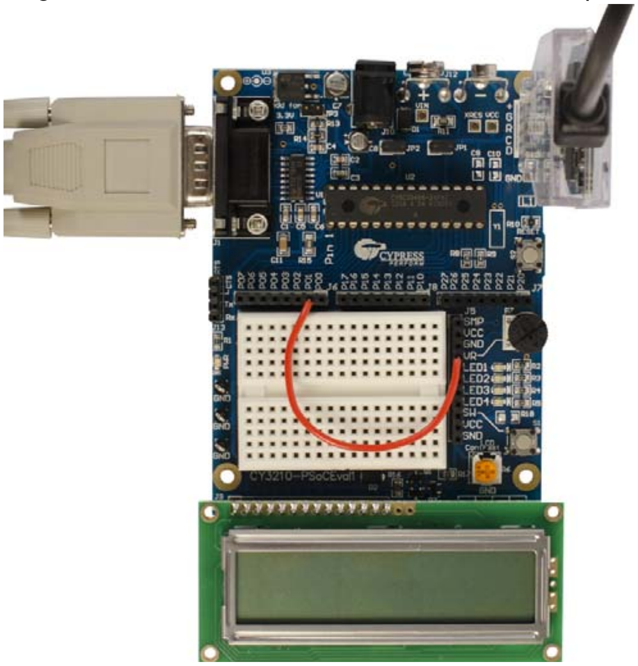
Figure 5-36. Device Configuration for ADC Conversion and LCD Display



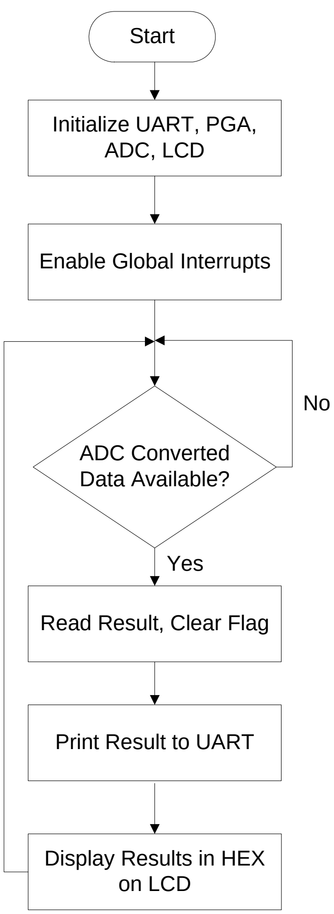
5.2.3 Hardware Connections

- ADC input (0–Vdd): Connect P01 to VR
- Serial Rx: Place jumper (shunt) on JP1 to connect P16 and Rx
- Serial Tx: Place jumper (shunt) on JP2 to connect P27 and Tx
- Connect RS-232 cable to the PC

Figure 5-37. Hardware Connection: Code Example 2



5.2.4 Code Example 2 Flowchart



5.2.5 Verifying Output using LCD

When the example code is built and programmed into the device, reset the device by pressing the RESET button or power cycling the board. The voltage of the potentiometer is measured by the ADC and is output as a four-digit hex value on the LCD. The value displayed on the LCD should change as the potentiometer is turned (see Figure 5-38).

Note Remove jumper JP3 to verify output at 5 V. The LCD display will not be seen at 3.3 V operation.

Figure 5-38. Verify Output - Code Example 2

5.2.6 Verifying Output using UART

Open a terminal application such as HyperTerminal or TeraTerm with these setup parameters:

- Baud Rate: **19200**
- Data: **8-bit**
- Parity: **None**
- Stop: **1-bit**
- Flow Control: **None**

The ADC value is displayed on the HyperTerminal and on the LCD.

Figure 5-39. HyperTerminal Settings

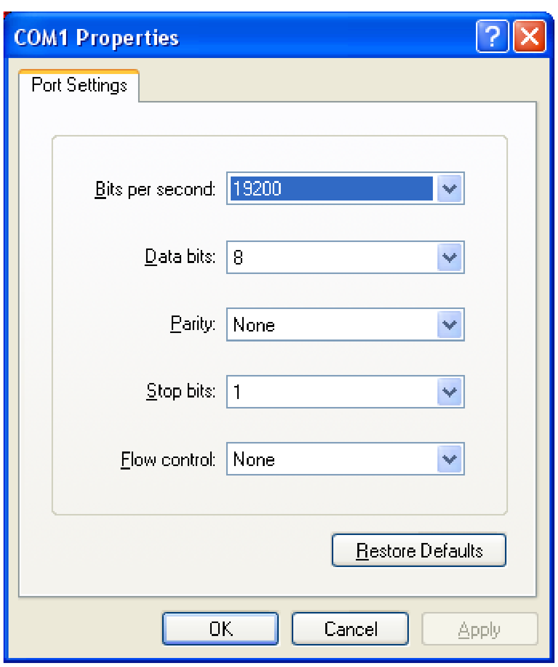
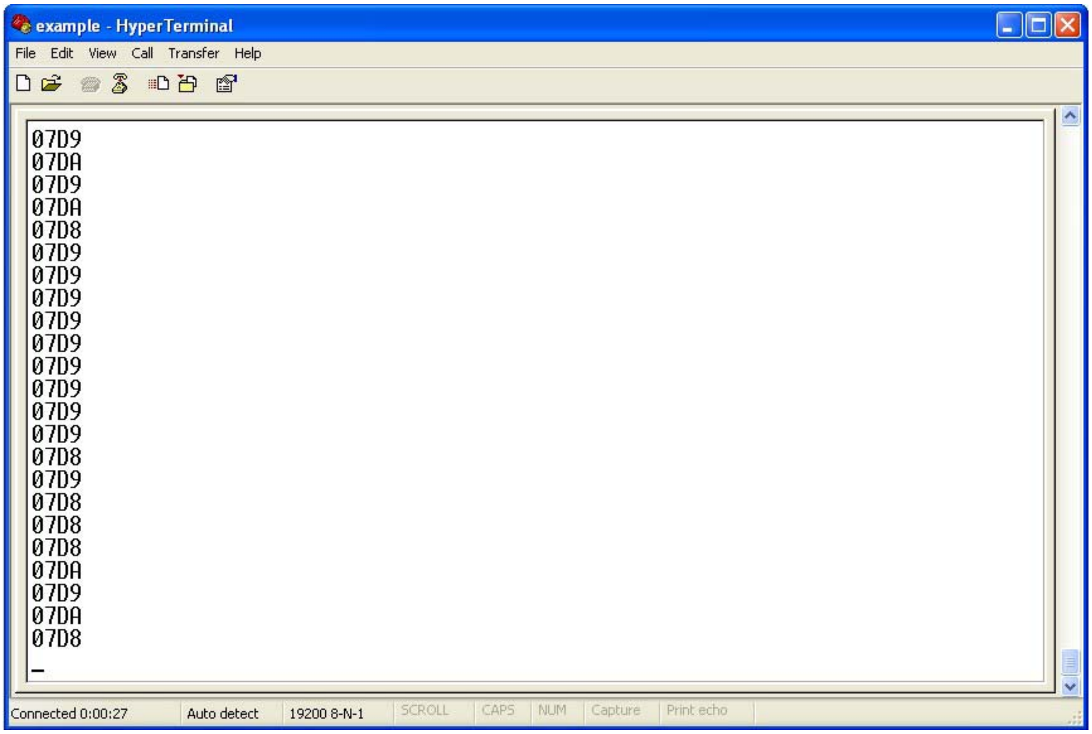


Figure 5-40. Verify Output on HyperTerminal



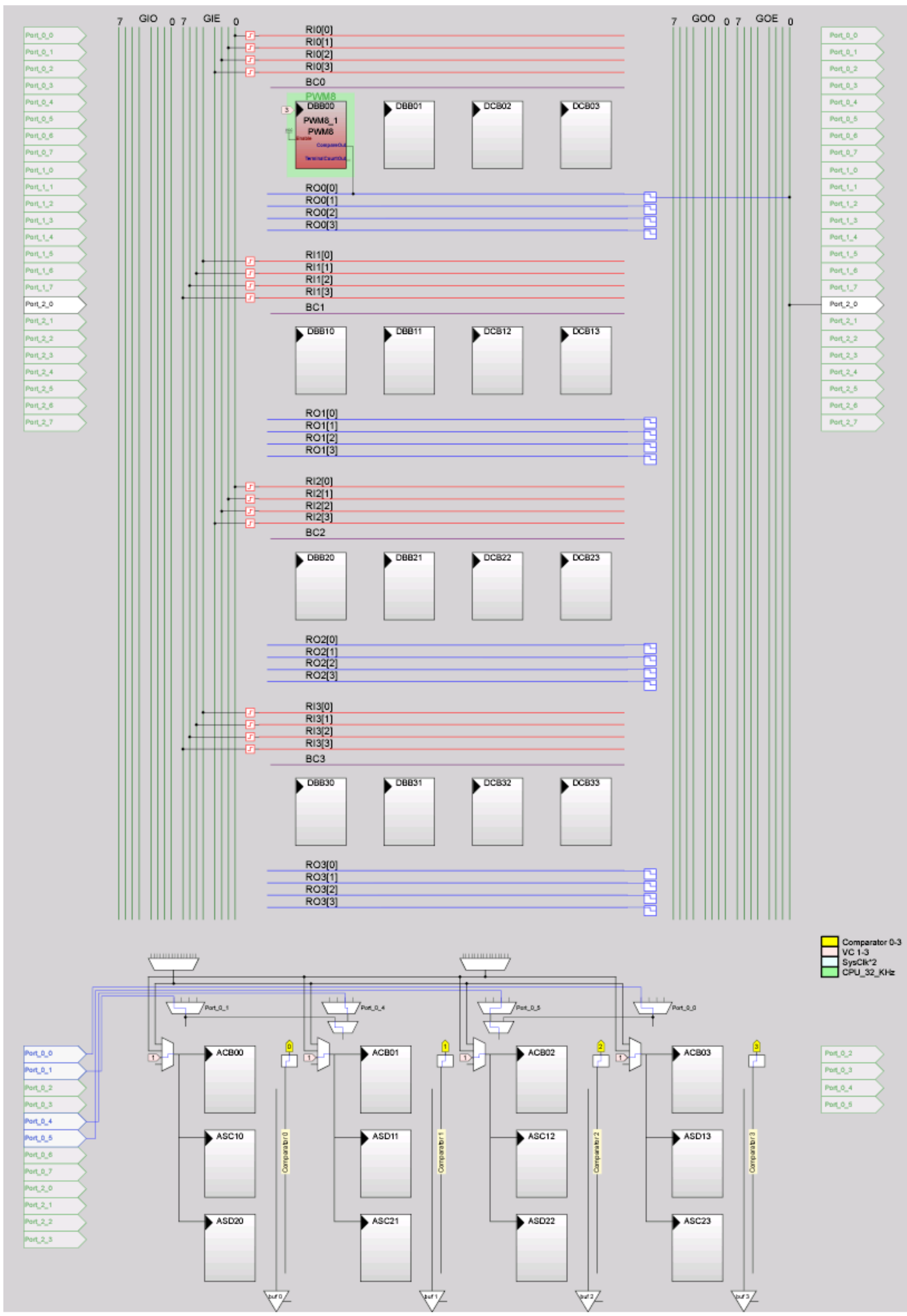
5.3 Code Example 3: ASM_Example_Blink_LED

5.3.1 Project Description

This project demonstrates how an LED blinks at a constant duty cycle using a hardware PWM.

PWM8: The clock dividers VC1, VC2, and VC3 are used to divide the 24-MHz system clock by 16, 16, and 256, respectively. The resulting 366-Hz clock is used as the input to an 8-bit PWM. This in turn produces an LED blink period of 1.4 Hz.

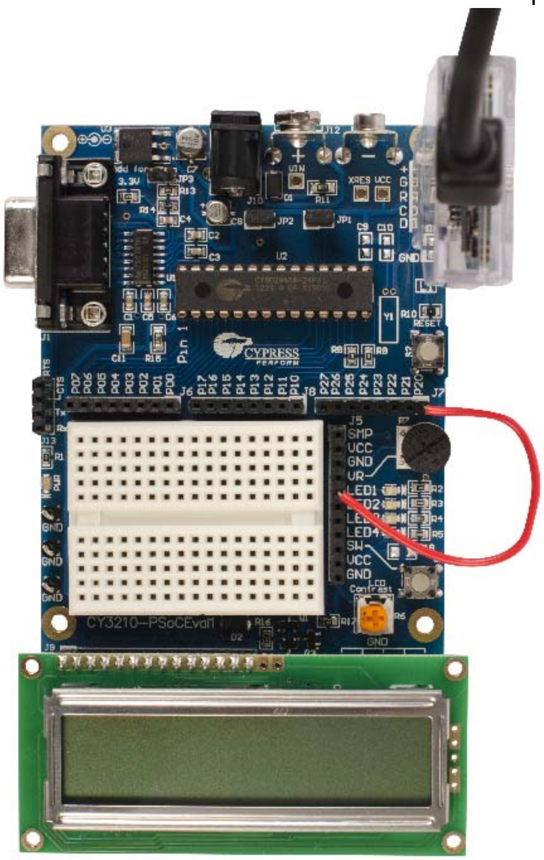
Figure 5-41. Device Configuration to Blink an LED



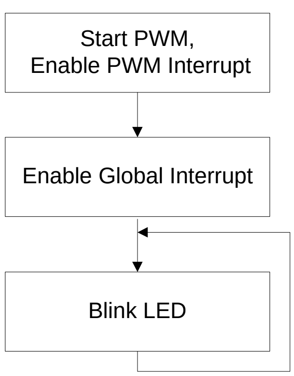
5.3.2 Hardware Connections

- Connect P20 to LED1.

Figure 5-42. Hardware Connection: Code Example 3



5.3.3 Code Example 3 Flowchart

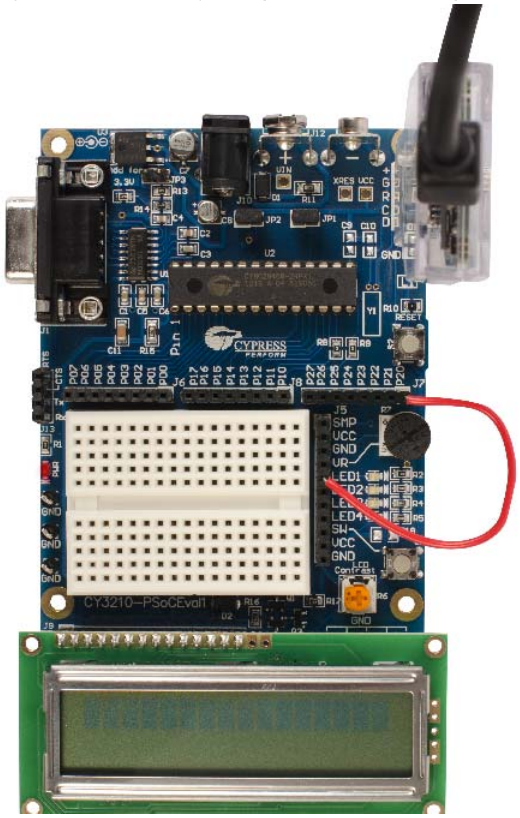


5.3.4 Verifying Output

When the example code is built and programmed into the device, make all the hardware connections and reset the board by pressing the RESET button or by power cycling the board. The LED1 blinks with a blink period of 1.4 Hz.

Note Remove the jumper JP3 to verify output at 5 V.

Figure 5-43. Verify Output: Code Example 3



5.4 Code Example 4: ASM_Example_DAC_ADC

5.4.1 Project Description

This project generates a sine wave using a 6-bit DAC. The sine wave period is based on the current ADC value of the potentiometer. The project uses the following user modules:

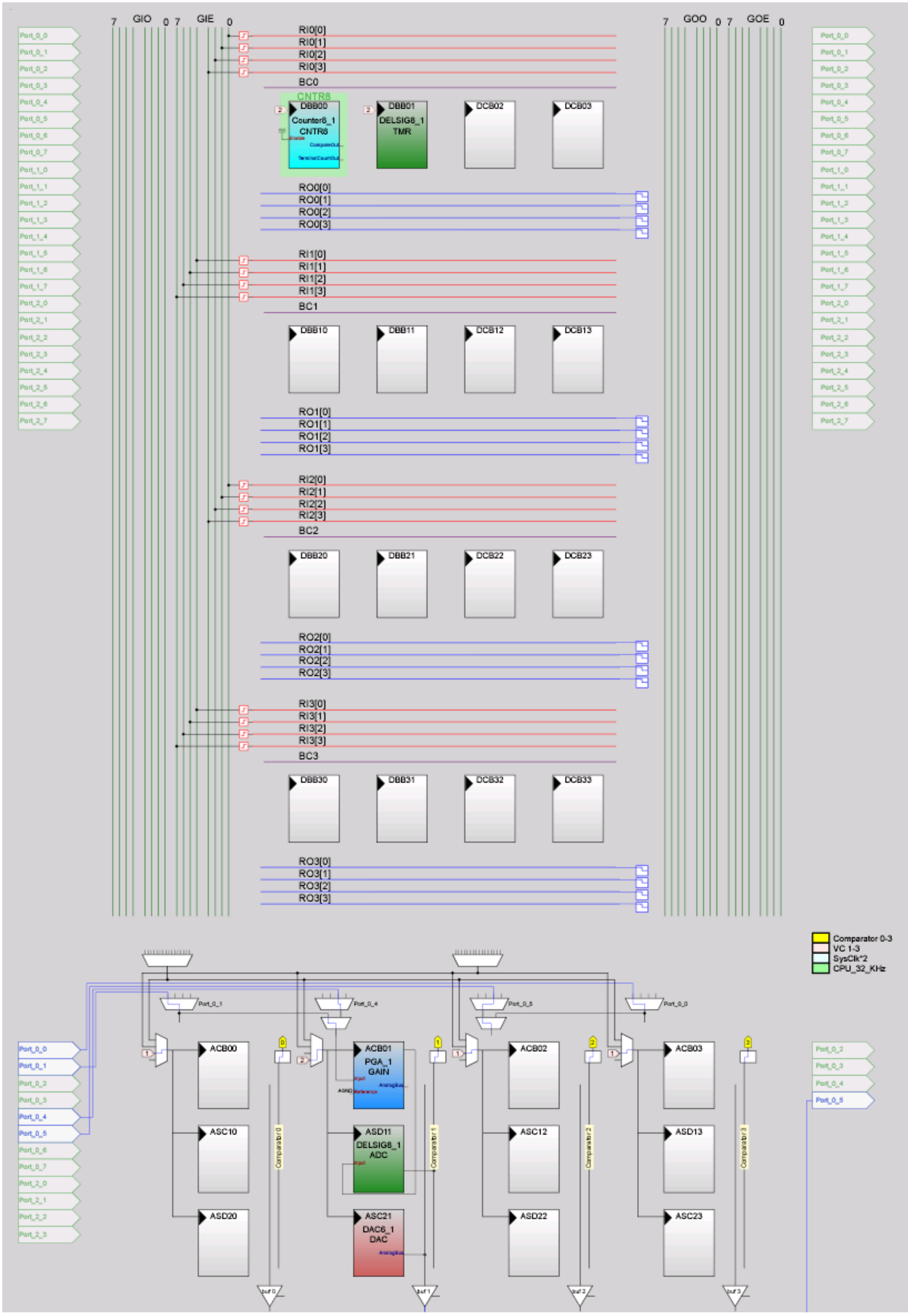
Counter8: An 8-bit counter is used to generate an interrupt at the DAC update rate (1/64 sine wave period). By adjusting the counter period, the DAC frequency and the resulting sine frequency can be modified.

DELSIG8: Current ADC conversion values are used to reload counter period. ADC input voltage is between 0 V and Vdd, depending on the potentiometer.

PGA: This module is implemented as buffer with user-programmable gain.

DAC6: This module converts digital input to analog output, which is used to generate sine wave. The DAC output is routed to P0[5] in the chip design. When P0[5] is connected to an LED on the board using single strand jumper wire, the LED blink period varies with the position of potentiometer. The sine wave pattern can be observed on an oscilloscope when the DAC output (P0[5]) is connected to the oscilloscope.

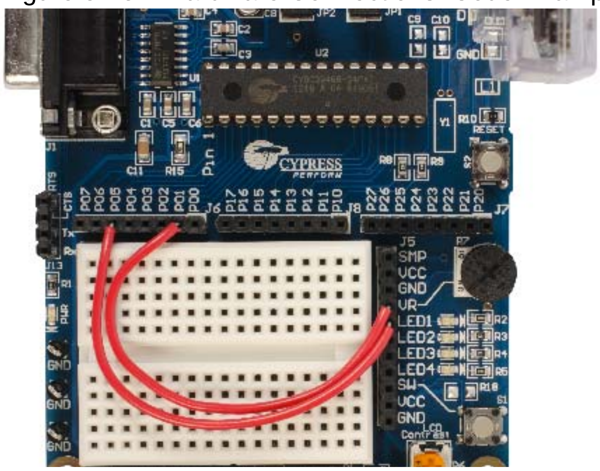
Figure 5-44. Device Configuration to Output a Sine Wave



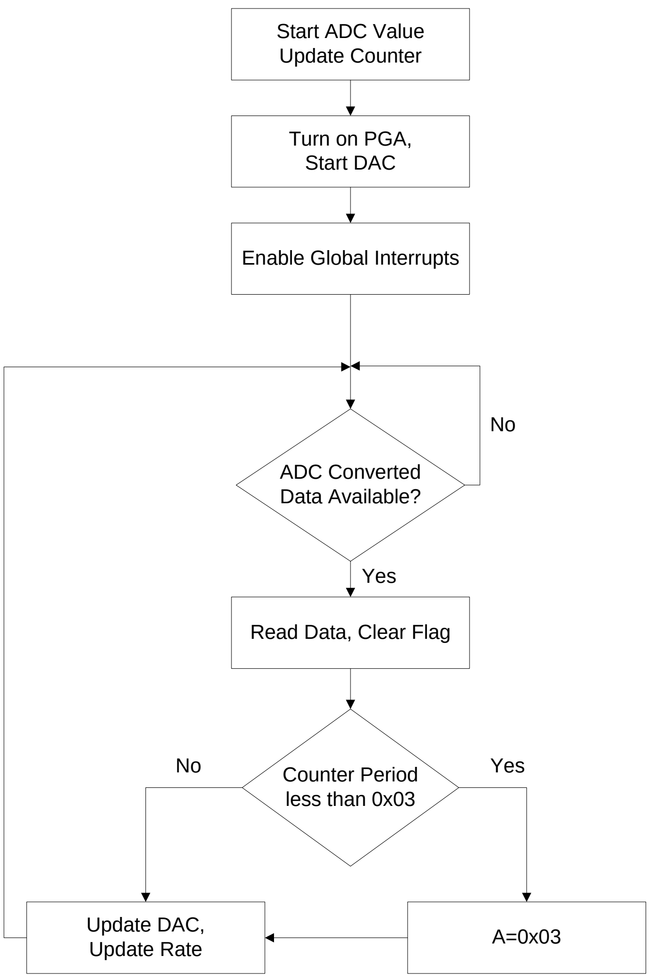
5.4.2 Hardware Connections

- ADC Input (0–Vdd): Connect P01 to VR
- DAC Output (0–Vdd): Connect P05 to LED1 and CRO

Figure 5-45. Hardware Connections: Code Example 4



5.4.3 Code Example 4 Flowchart



5.4.4 Verifying Output

After the example code is built and programmed into the device, make all the hardware connections and reset the board by pressing the RESET button or by power cycling the board. LED1 is a sine wave output whose period is based on the ADC. Turning the potentiometer changes the ADC value and controls the frequency of the sine wave. The sine wave output can be viewed on an oscilloscope.

Note Remove the jumper JP3 to verify output at 5 V.

Figure 5-46. Output Sine Wave

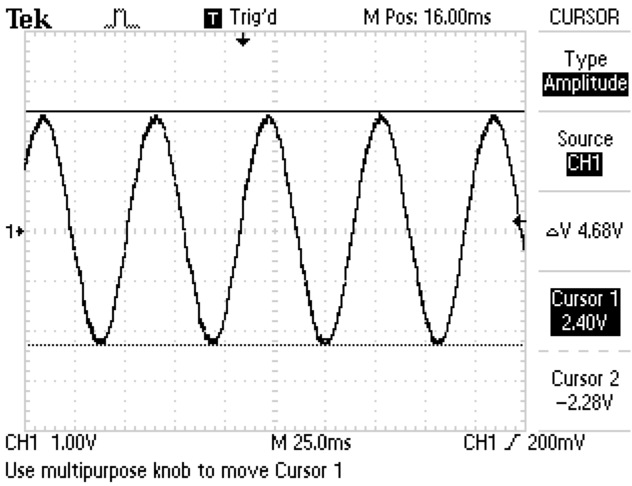
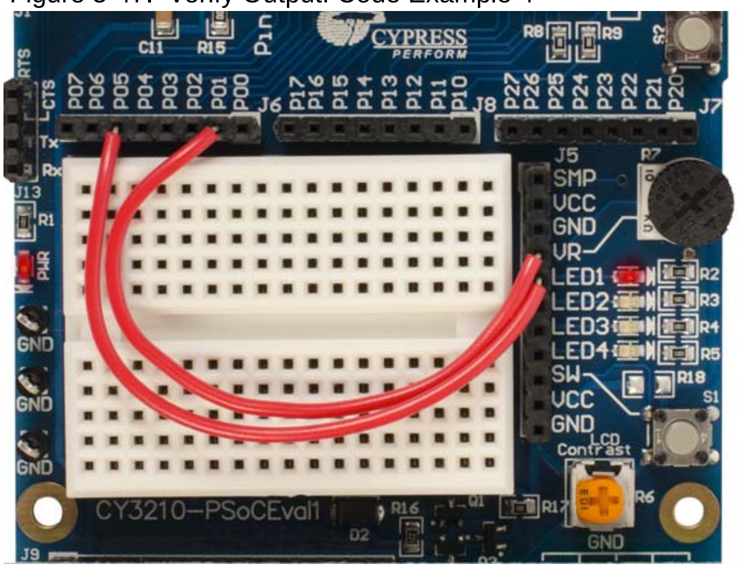


Figure 5-47. Verify Output: Code Example 4

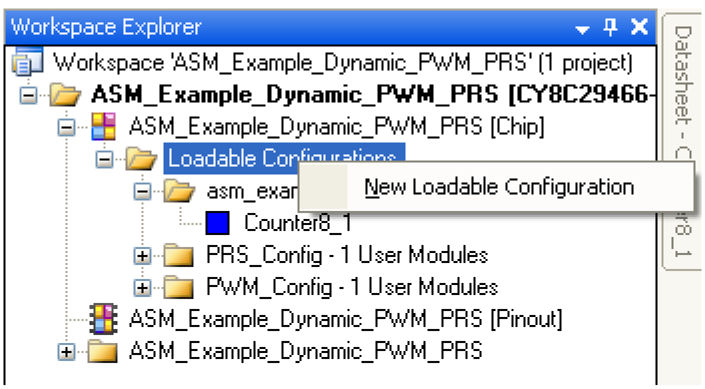


5.5 Code Example 5: ASM_Example_Dynamic_PWM_PRS

5.5.1 Project Description

This project demonstrates the Dynamic Reconfiguration capability of PSoC Designer. A set of user modules is called a configuration. A loadable configuration consists of one or more UMs placed with module parameters, global resources, pinouts, and generated application files. The UMs used and the register settings can be changed easily. The application can switch in and out of configurations in real-time, allowing greater use of the chip resources. This is similar to memory overlaying or using more than 100 percent of the resources in an FPGA design. Imagine a vending machine where a PSoC monitors and controls environmental conditions such as temperature and humidity, dispenses drinks, counts money, makes change, controls LEDs and an LCD display, and implements capacitive sensing buttons or a touchscreen interface. Now, imagine that once a day for about a minute, it reconfigures itself such that it can send information back to a central office. This is the power of PSoC with dynamic reconfiguration.

To add loadable configurations to a PSoC project, right-click the **Loadable Configuration** folder in the Workspace Explorer and select **New Loadable Configuration**. A new folder is created with a default name, Configx, where x is the number of alternate configurations.



The new configuration can be renamed based on the project. All PSoC resources (digital and analog blocks) can be reused. A loadable configuration can be deleted if no longer required. However, the base configuration (the default configuration) cannot be deleted.

When using dynamic reconfiguration, global parameters are set in the same manner as single configurations. However, changes to the base configuration global parameters are propagated to all additional configurations. Therefore, global parameter changes made to an additional configuration are done locally to that particular configuration. For instance, if global parameter #1 has a specific value in "Base" loadable configuration, it will have the same value in "New" loadable configuration. But not vice versa: if global parameter #2 has a specific value in "New" loadable configuration, it will have the default (or base-specific) value in "Base" loadable configuration." The same also applies to port pin settings.

In this project, dynamic reconfiguration is demonstrated by configuring a PWM in the digital block DBB01 in one configuration. The same block is used to configure a PRS in the second configuration. A Counter8 UM is placed in the base configuration that drives the PWM or PRS module based on the configuration that is loaded. A switch on the CY3210-PSoCEVAL1 board is used to toggle between configurations.

When the SW switch is released, the PRS configuration is unloaded (if already loaded), and the PWM configuration is loaded. In this configuration, LED1 and LED2 are used to output the PWM Pulse Width and PWM Terminal Count, respectively.

When the SW switch is pressed, the PWM configuration is unloaded and the PRS configuration is loaded. In this configuration, LED1 and LED3 are used to output the PRS Pulse Density and PRS Bitstream, respectively.

The APIs UnloadConfig_<configname> and LoadConfig_<configname> are used to unload and load the required configurations

Counter8: In the base configuration, it takes a clock of 732 Hz as an input.

PRS8: The PRS configuration contains a PRS with pulse density (analogous to pulse width) and a bitstream output that is shifted out to LED pin.

PWM8: The PWM configuration contains a standard 8-bit PWM with a duty cycle of 50 percent. Both the pulse width and terminal count outputs are displayed on LEDs.

Figure 5-48. Base Configuration

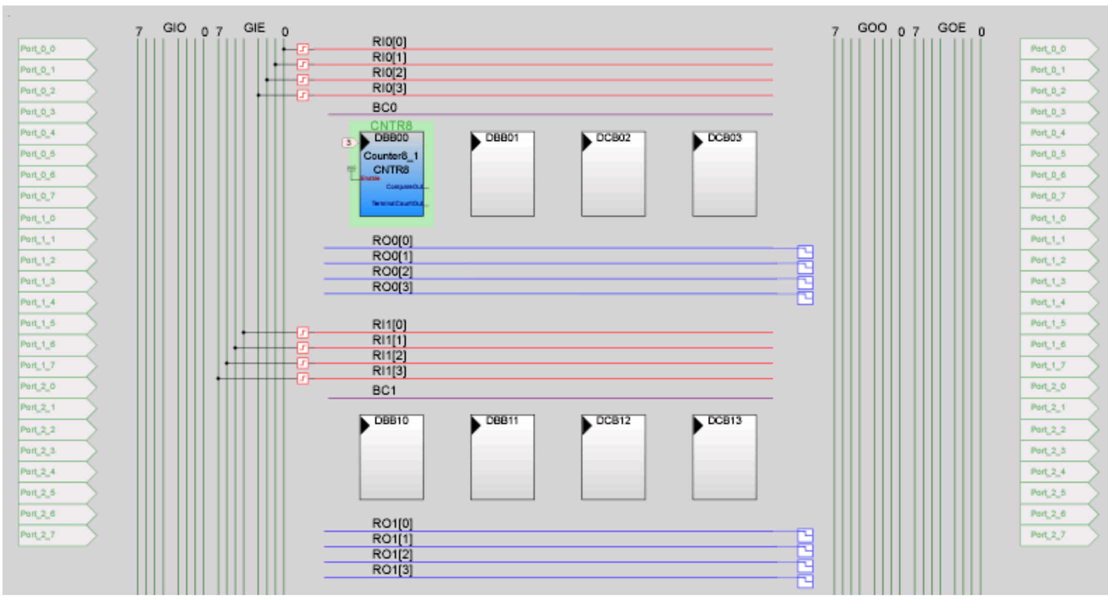


Figure 5-49. PWM Configuration

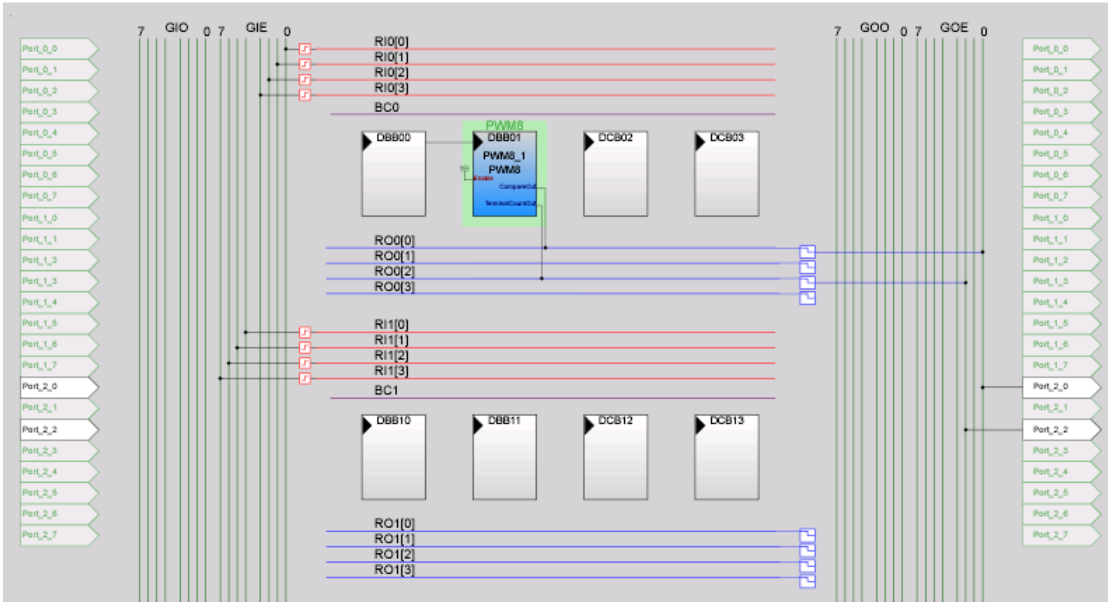
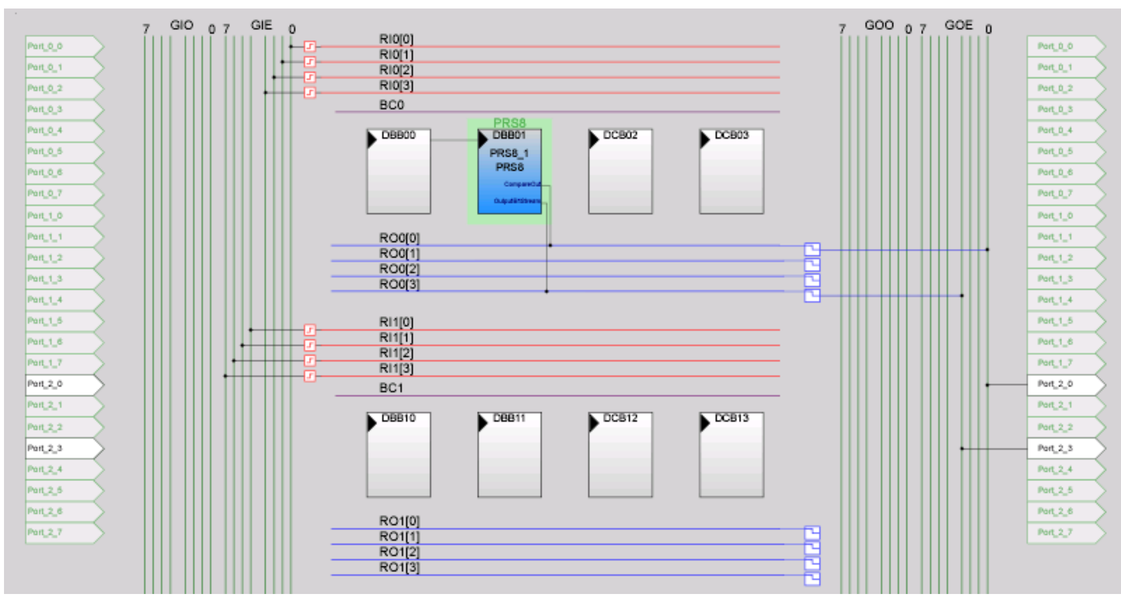


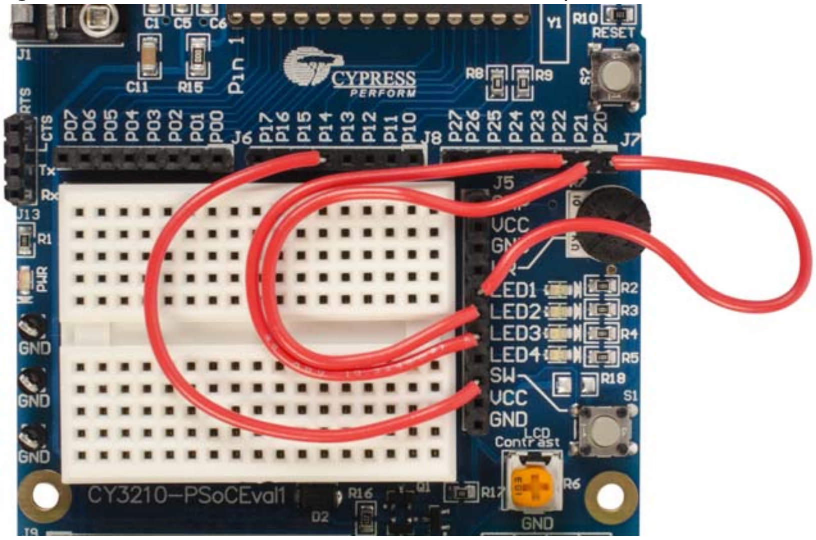
Figure 5-50. PRS Configuration



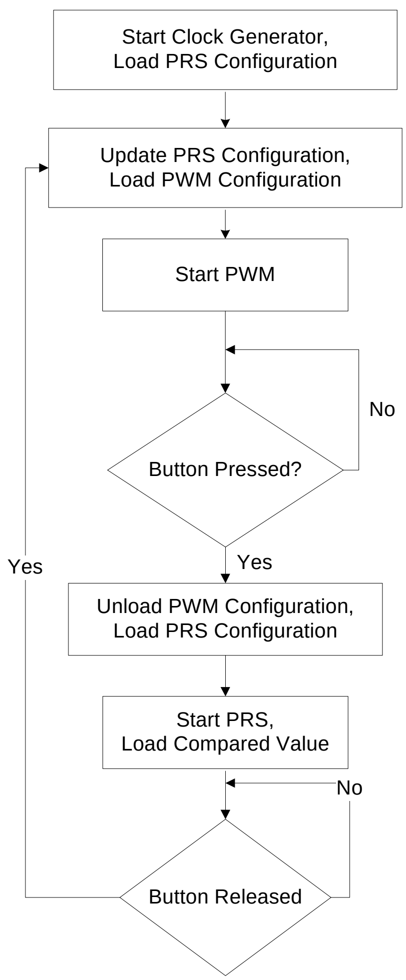
5.5.2 Hardware Connections

- User button: Connect P14 to SW
- PWM pulse width or PRS pulse density: Connect P20 to LED1
- PWM terminal count: Connect P22 to LED2
- PRS bitstream: Connect P23 to LED3

Figure 5-51. Hardware Connection: Code Example 5



5.5.3 Code Example 5 Flowchart

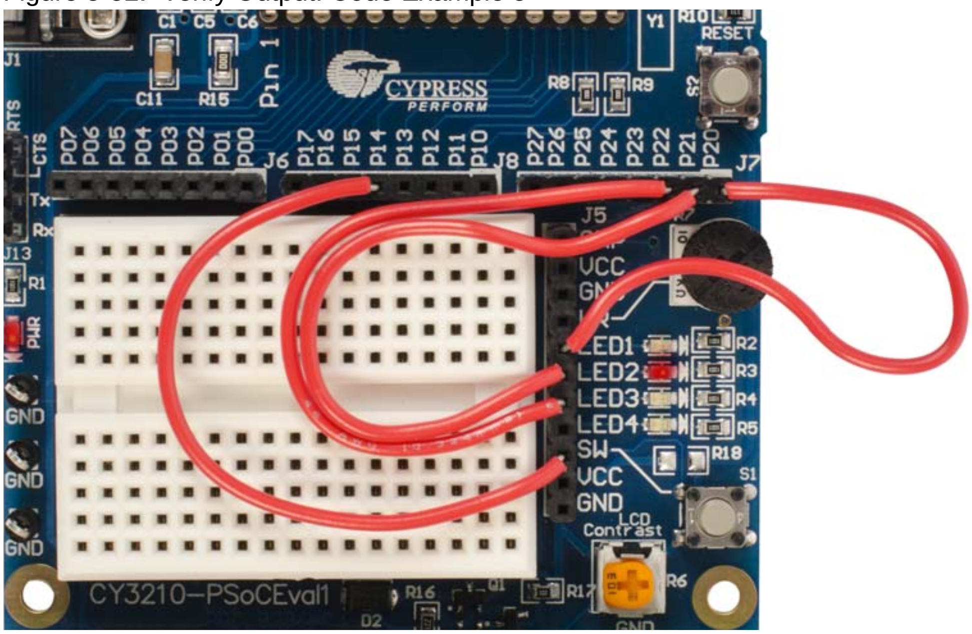


5.5.4 Verifying Output

Program the board with the hex file; disconnect power and make all hardware connections. Remove the LCD module and power the board. When the switch is released, the PWM configuration is loaded and LED1 and LED2 blink with PWM Pulse Width and PWM Terminal Count, respectively. When the switch is pressed, the PRS configuration is loaded and LED1 and LED3 blinks with PRS Pulse Density and PRS Bitstream, respectively.

Note Remove jumper JP3 to verify output at 5 V.

Figure 5-52. Verify Output: Code Example 5



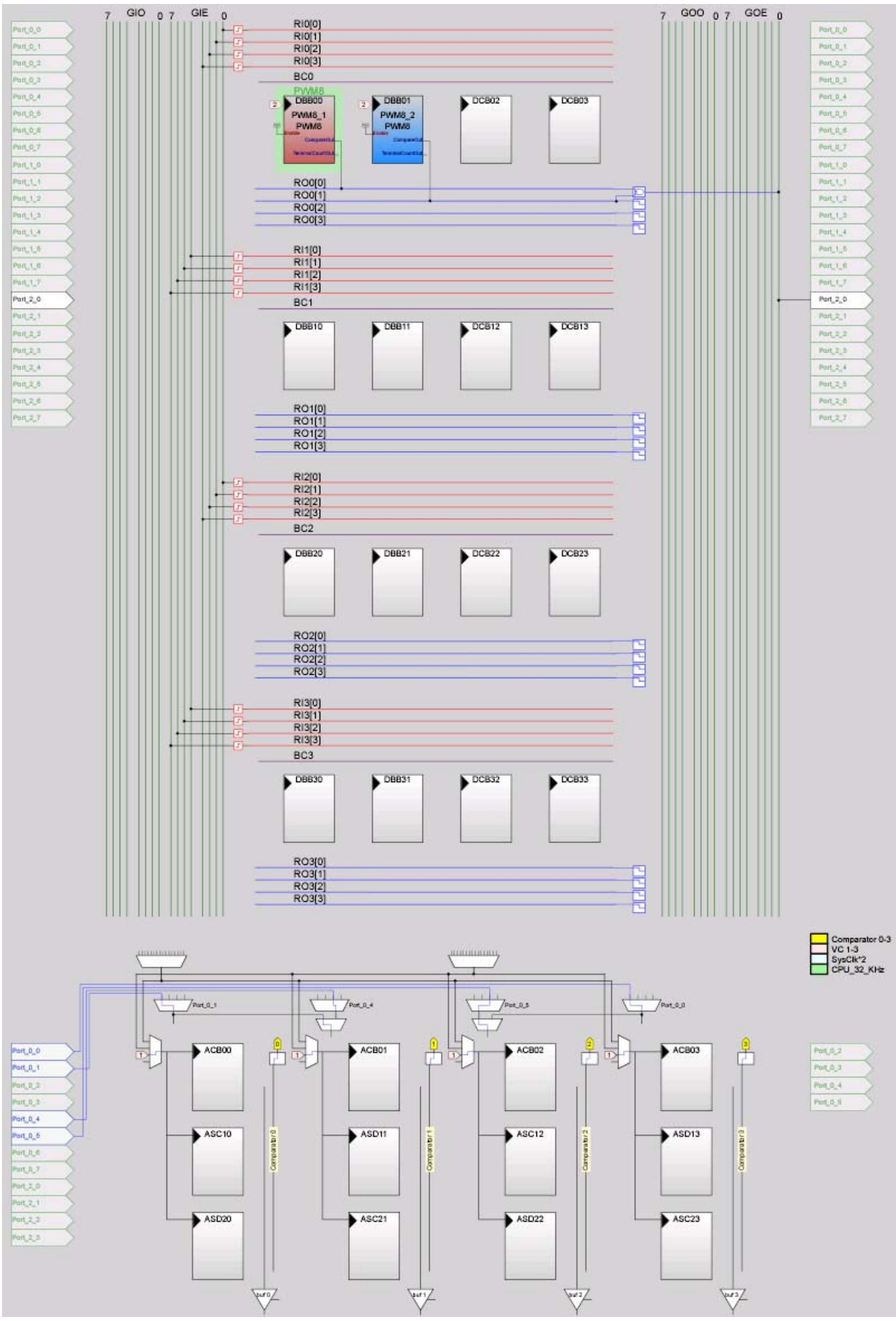
5.6 Code Example 6: ASM_Example_LED_Logic

5.6.1 Project Description

This project demonstrates a PSoC project designed to blink an LED using the output of two PWMs. The outputs are combined using an AND gate in an output bus logic block. This logical combination results in a beat frequency of 1.4 Hz.

PWM8: Two 8-bit PWM UMs process a 93.37-kHz clock with periods of 256 and 255, respectively. This produces the LED beat frequency of 1.4 Hz.

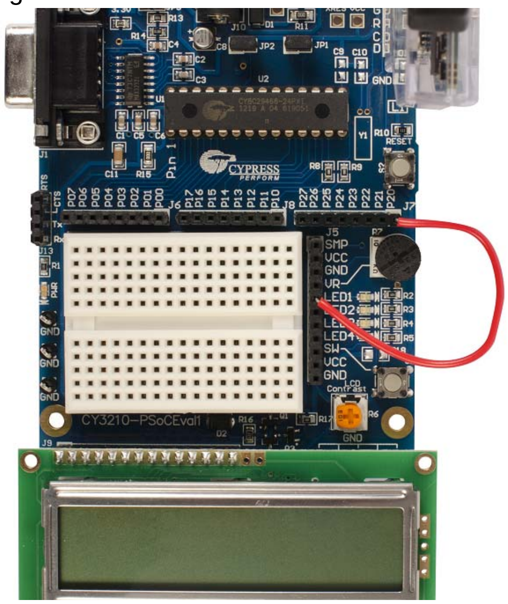
Figure 5-53. Device Configuration to Combine PWMs using Output Logic



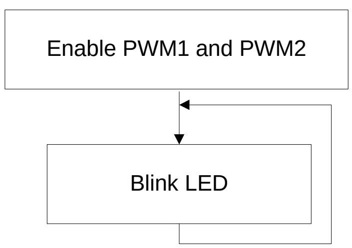
5.6.2 Hardware Connections

- Connect P20 to LED1

Figure 5-54. Hardware Connection: Code Example 6



5.6.3 Code Example 6 Flowchart

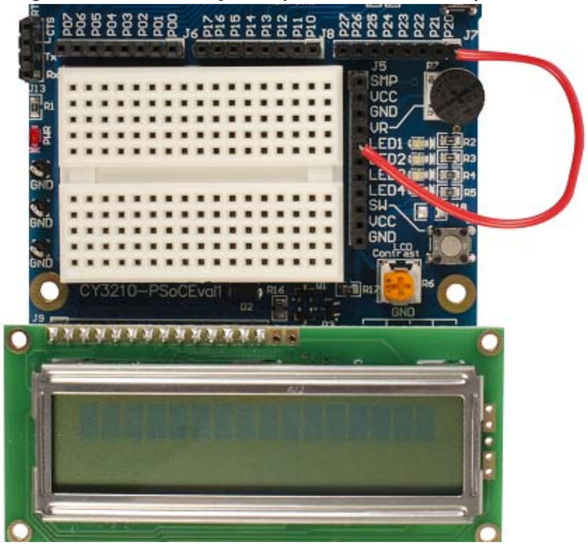


5.6.4 Verifying Output

After the program is built and programmed into the device, make all the hardware connections and reset the board by either pressing the **RESET** button or by power cycling the board. The LED1 blinks at a frequency of 1.4 Hz.

Note Remove jumper JP3 to verify output at 5 V.

Figure 5-55. Verify Output: Code Example 6



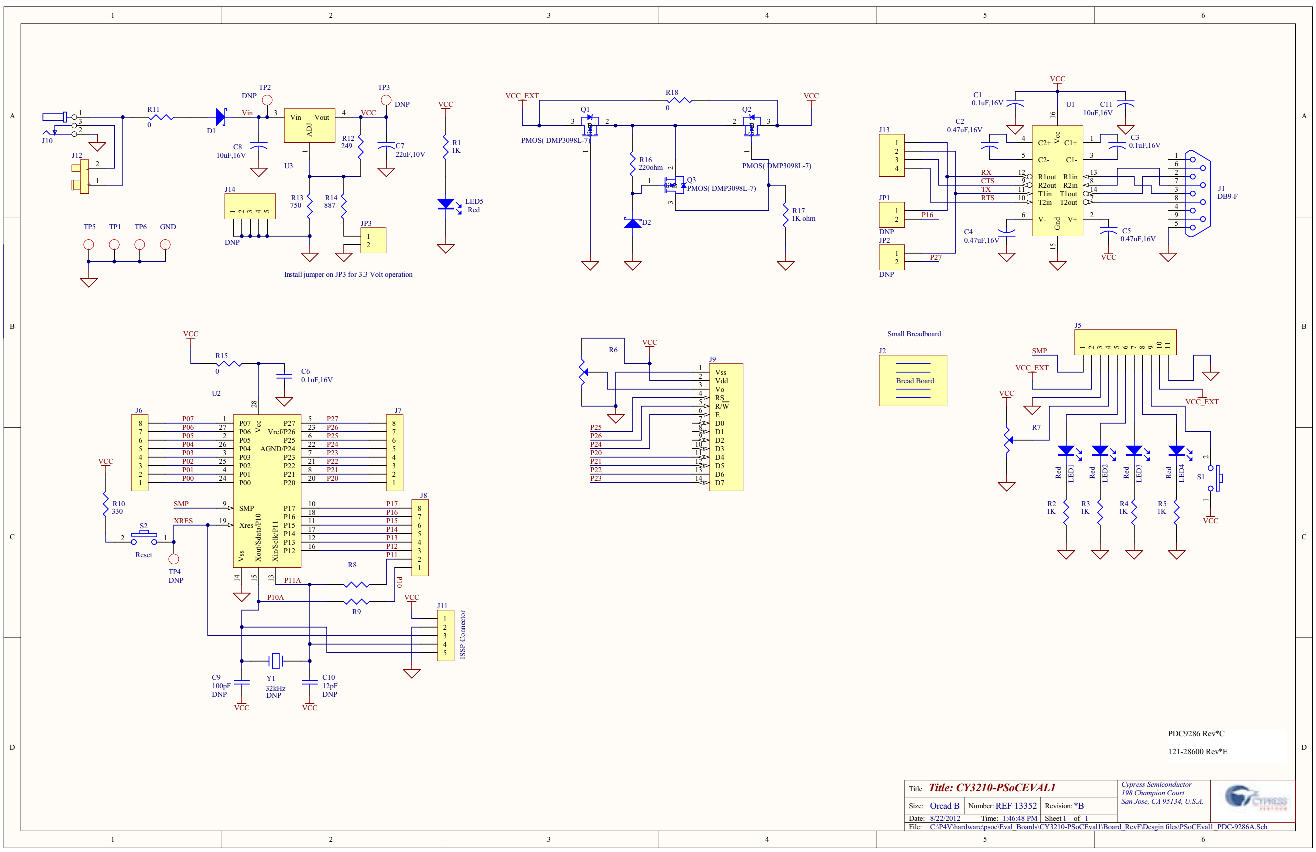
See Replace CY8C29466-24PXI with CY8C27443-24PXI on page 70 for instructions on how to replace the CY8C29466-24PXI PSoC part with CY8C27443-24PXI.

A. Appendix

The schematic and board layouts are available on the CY3210-PSoCEVAL1 kit DVD or at this location: `<Install_directory>:\Cypress\CY3210-PSoCEVAL1\<version>\Hardware.`

A.1 Schematic

A.1.1 CY3210-PSoCEVAL1 Board Schematic



A.2 Board Layout

Figure A-1. Top Copper Layer

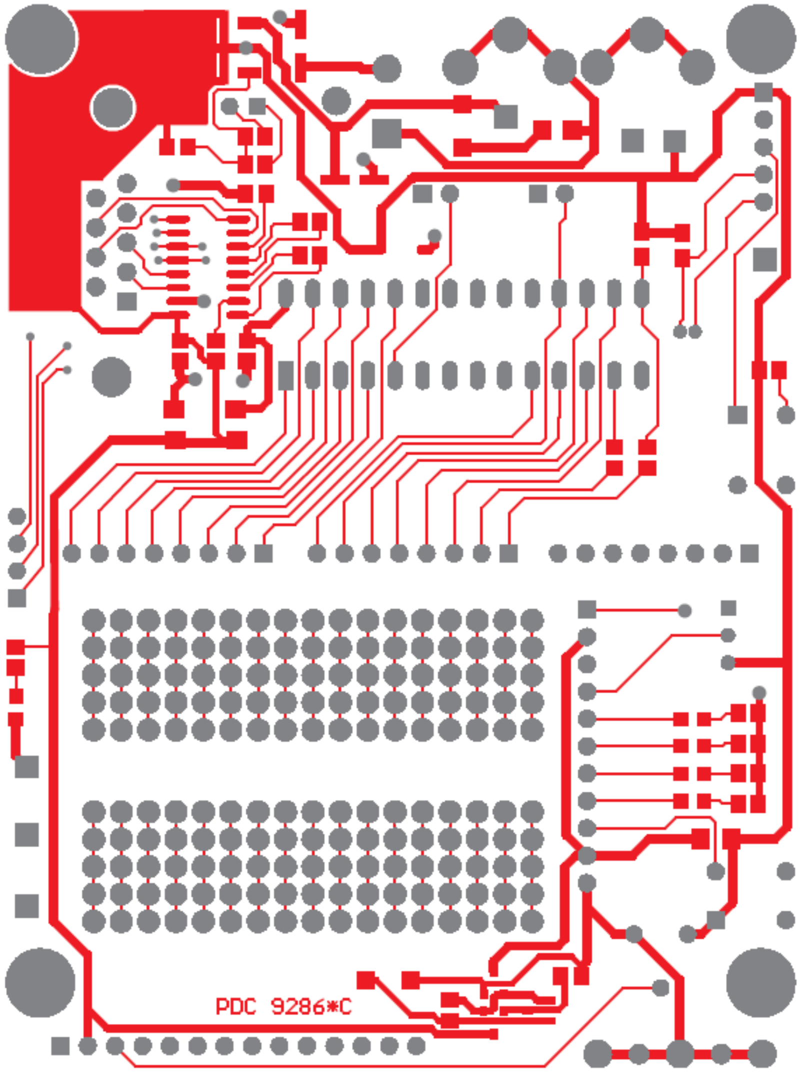


Figure A-2. Bottom Copper Layer

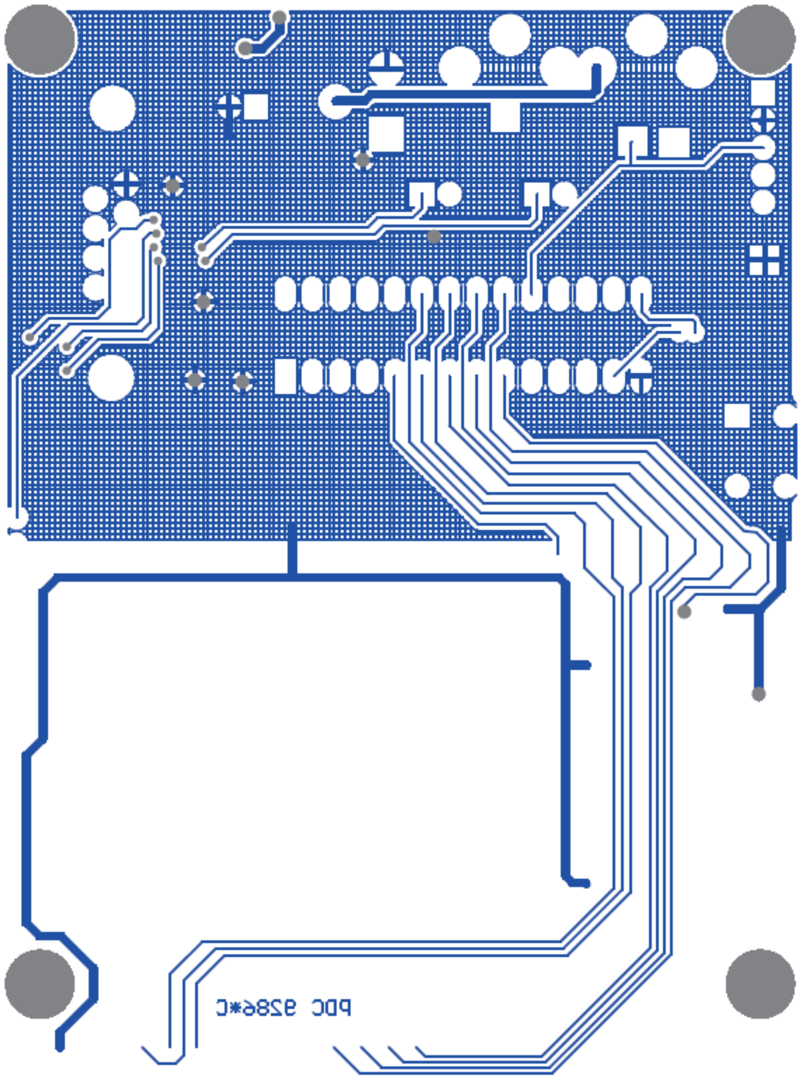
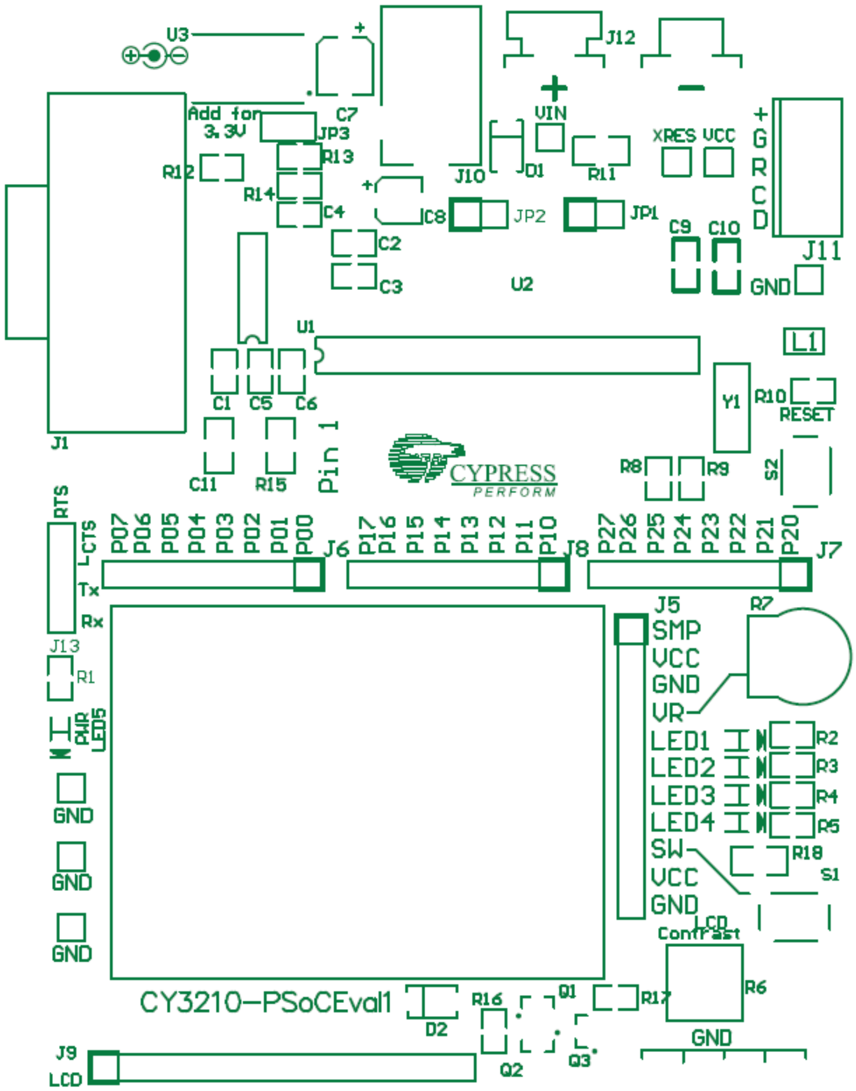


Figure A-3. Top Overlay



A.3 Bill of Materials

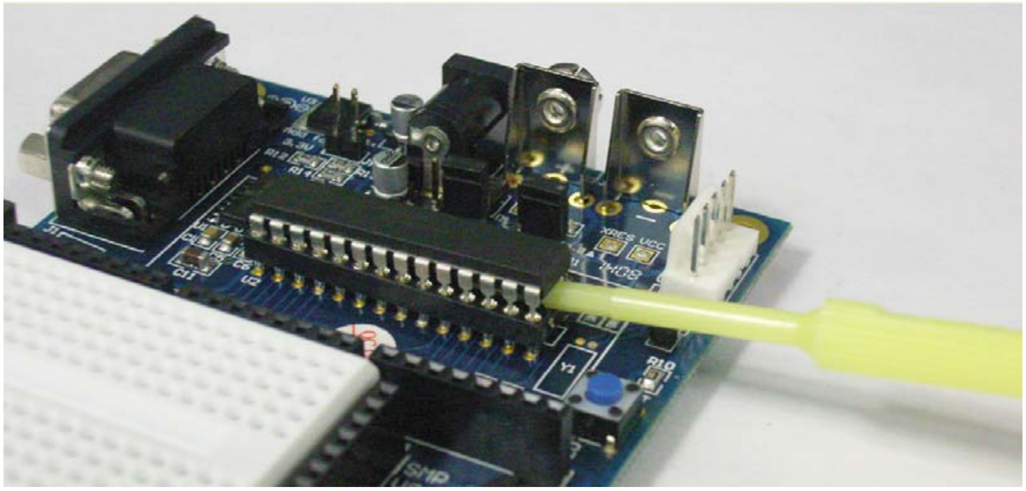
Item	Qty	Reference	Value	Description	Manufacturer	Manufacturer Part No.
1	1		rev *B	PDC-9286 PSoC EVAL1 Board	Cypress	PDC-9286
2	3	C1, C3, C6	0.1uF,16V	Capacitor	YAGEO (PHYCOMP)	CC0805ZRY5V9BB104
3	3	C2, C4, C5	0.47uF,16V	Capacitor	MULTICOMP	MCCA000287
4	1	C7	22uF,10V	Capacitor	NICHICON	UWX1A220MCL1GB
5	1	C8	10uF,16V	Capacitor	NICHICON	UWT1C100MCL1GB
6	1	C11	10uF,16V	Capacitor	TAIYO YUDEN	EMK316BJ106KL-T
7	1	D1	50V, 1A	Schottky Diode	DIODES INC.	RS1A-13-F
8	1	J1	DB9-F	Female DB-9	TE Connectivity//Amp	5747844-2
9	1	J2	-	3M solderless breadboard super strip	3M	923273-I
10	1	J5	11	Header, 11-Pin	3M	929850-01-36-RA
11	3	J6, J7, J8	8	8-Pin Header, Female	3M	929850-01-36-RA
12	1	J9	14	14-Pin header, Female	3M	929850-01-36-RA
13	1	J10	-	Power connector	CUI Inc	PJ-102A
14	1	J11	5	ISSP connector	MOLEX	22-23-2051
15		J12	-	9-Volt battery connector	KEYSTONE	593
16	1	J12	-	9-Volt battery connector	KEYSTONE	594
17	1	J13	4	Header, 4-Pin	3M	929850-01-36-RA
18	5	LED1, LED2, LED3, LED4, LED5	Red	Red LED	LUMEX	SML-LXT0805IW-TR
19	5	R1, R2, R3, R4, R5	1K	Resistor, SMT	PANASONIC	ERJ6GEYJ102V
20	1	R6	10K	Potentiometer	Panasonic	EVN-D8AA03B14
21	1	R7	10K	Potentiometer	BOURNS	3352T-1-103LF
22	2	R8, R9	0	Resistor, SMT	PANASONIC	ERJ6GEY0R00V
23	1	R10	330	Resistor, SMT	PANASONIC	ERJ6GEYJ331V
24	1	R11	0	Resistor, SMT	PANASONIC	ERJ8GEY0R00V
25	1	R12	249	Resistor, SMT	PANASONIC	ERJ6ENF2490V
26	1	R13	750	Resistor, SMT	PANASONIC	ERJ6ENF7500V
27	1	R14	887	Resistor, SMT	PANASONIC	ERJ-6ENF8870V
28	2	S1, S2	-	Swtich, SPST	OMRON ELECTRONIC COMPONENTS	B3F-1022
29	3	TP1, TP5, TP6	-	Test point	KEYSTONE	5006
30	1	U1	3V	3V RS-232 tranceiver (1.0uF Caps)	TEXAS INSTRUMENTS	MAX3232IDR
31	1	U2	28 pin	28 pin DIP socket, machine pin	MILL MAX	110-99-328-41-001000
32	1	U3	3.3/5V	Adjustable Voltage Regulator	TEXAS INSTRUMENTS	LM317MKTPR
33	1	JP3	2	Header, 2-Pin, Male	Sullins Connector Solu-tions	PBC36SAAN
34	4	N/A	BUMPER	BUMPER CLEAR .375X.15"DOME	Richco Plastic	RBS-12
35	2	JP1, JP2	2	Header, 2-Pin, Male	TE Connectivity//Amp	0-0142270-3
NO LOAD Components						
36	1	C9	12pF	Capacitor	PANASONIC	ECJ-2VC1H120J
37	1	C10	100pF	Capacitor	PANASONIC	ECJ-2VC1H101J
38	1	J14	-	Ground conn.	N/A	
39	3	TP2, TP3, TP4	-	Simple Test point	Keystone Electronics	5006
40	1	Y1	32kHz	Crystal	ECS Inc	ECS-3X8

A.4 Replace CY8C29466-24PXI with CY8C27443-24PXI

The CY3210-PSoCEVAL1 kit includes two PSoC parts, CY8C29466-24PXI and CY8C27443-24PXI. The CY8C29466-24PXI part is placed on the IC socket of the board when shipped. To replace the default part with CY8C27443-24PXI, follow these instructions.

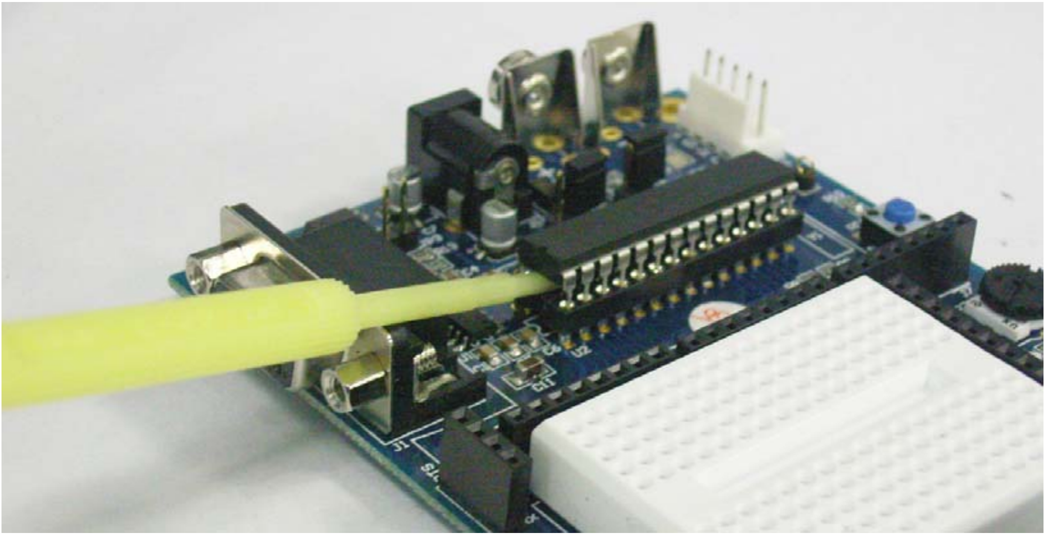
Use an IC extractor – place the ends of the extractor between the IC and the socket. Gently pull the IC out evenly, away from the board. If an IC extractor is not available, using a nonmetallic tool, lift one side of the IC, as shown in Figure A-4. Do not lift the IC completely because this can damage the pins on the opposite side.

Figure A-4. Lift IC - One Side



Repeat on the other side to lift the silicon from the socket, as shown in Figure A-5.

Figure A-5. Lift IC - Other Side



Place the CY8C27443-24PXI silicon on the IC socket. Ensure that the silicon notch is aligned with the notch on the IC socket; see Figure A-6. Press down gently to finish the replacement.

Figure A-6. Place New Chip