



CY3214

## PSoCEVALUSB Kit Guide

Doc No. 001-67030 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, 2011-2012. 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™ is a trademark 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



<b>1. Introduction</b>	<b>5</b>
1.1 Kit Contents .....	5
1.2 Additional Learning Resources .....	6
1.2.1 Reference Documents .....	6
1.3 Document History .....	6
1.4 Documentation Conventions .....	6
<b>2. Getting Started</b>	<b>9</b>
2.1 Kit Installation .....	9
2.2 PSoC Designer .....	12
2.3 PSoC Programmer .....	14
2.4 SuiteUSB 3.4 .....	14
<b>3. Kit Operation</b>	<b>15</b>
3.1 Introduction .....	15
3.2 Square Mouse Example .....	15
3.2.1 Programming Specifications and Connections .....	16
<b>4. Hardware</b>	<b>19</b>
4.1 System Block Diagram .....	19
4.2 Functional Description .....	22
4.2.1 Character LCD Module .....	22
4.2.2 CapSense® Buttons (P3[0], P3[7]), and Sliders (Port5) .....	23
4.2.3 Legacy DB9 Connector (J1) .....	24
4.2.4 Power Supply Connector (J13) .....	25
4.2.5 LEDs (LED1–4) .....	26
4.2.6 Buzzer (LS1) .....	26
4.2.7 ISSP Connector (J15) .....	27
4.2.8 Joystick (JS1) .....	27
4.2.9 Reset Button (S3) .....	28
4.2.10 ICE-Cube Connector (J14) .....	28
4.2.11 USB Connector (P1) .....	29
4.2.12 Potentiometers (VR1 and VR2) .....	29
4.2.13 Buttons (S1 and S2) .....	30
<b>5. Code Examples</b>	<b>31</b>
5.1 My First Code Example .....	31
5.1.1 Project Objective .....	31
5.1.2 Flowchart .....	32
5.1.3 Creating My First PSoC 1 Project .....	33
5.1.4 Verify Output .....	48
5.2 Buttons and Lights Example .....	51

5.2.1	Project Description.....	51
5.2.2	Hardware Connections .....	51
5.2.3	Firmware Architecture.....	53
5.2.4	Verify Output .....	54
5.3	Temperature Sensing Keyboard.....	55
5.3.1	Project Description.....	55
5.3.2	Hardware Connection .....	55
5.3.3	Firmware Architecture.....	57
5.3.4	Verify Output .....	58
5.4	Process Monitoring.....	59
5.4.1	Project Description.....	59
5.4.2	Hardware Connection .....	60
5.4.3	Firmware Architecture.....	61
5.4.4	Verify Output .....	62
5.5	Async_PingExample.....	63
5.5.1	Project Description.....	63
5.5.2	Hardware Connections .....	63
5.5.3	Driver Installation .....	64
5.5.4	Firmware Architecture.....	67
5.5.5	Verify Output .....	67
5.6	Bulk_PingExample .....	68
5.6.1	Project Description.....	68
5.6.2	Hardware Connections .....	68
5.6.3	Driver Installation .....	69
5.6.4	Firmware Architecture.....	70
5.6.5	Verify Output .....	71
5.7	ISoc_Example .....	71
5.7.1	Project Description.....	71
5.7.2	Hardware Connections .....	71
5.7.3	Driver Installation .....	72
5.7.4	Firmware Architecture.....	73
5.7.5	Verify Output .....	73
5.8	USBUART .....	74
5.8.1	Project Description.....	74
5.8.2	Hardware Connections .....	75
5.8.3	Driver Installation .....	76
5.8.4	Firmware Architecture.....	77
5.8.5	Verify Output .....	78
5.9	Joystick.....	80
5.9.1	Description .....	80
5.9.2	Hardware Connection .....	80
5.9.3	Firmware Architecture.....	82
5.9.4	Verify Output .....	83
<b>A.</b>	<b>Appendix</b>	<b>85</b>
A.1	Schematic.....	85
A.2	Board Layout .....	86
A.2.1	CY3214 -PSoCEVALUSB Top Copper.....	86
A.2.2	CY3214-PSoCEVALUSB Top Overlay .....	87
A.2.3	CY3214 -PSoCEVALUSB Bottom Copper.....	88
A.3	Bill of Materials (BOM).....	89

# 1. Introduction



Thank you for your interest in the CY3214-PSoCEVALUSB Kit. This kit includes a development board for the CY8C24894-24LTXI PSoC<sup>®</sup> device. Note that the board has the CY8C24094 device, which is an on-chip debug part for the CY8C24x94 family. You can design your own projects with PSoC Designer<sup>™</sup> or by altering the code examples provided with this kit. It is a combination of technologies that create opportunities for a new range of low-cost, mixed-signal USB products. This architecture enables creating customized peripheral configurations that match the requirements of individual applications.

The CY3214-PSoCEVALUSB Kit is based on the PSoC 1 family of devices. PSoC is a Programmable System-on-Chip<sup>™</sup> platform for 8-bit applications. It combines precision analog and digital logic with a high-performance CPU. With PSoC, you can create the exact combination of peripherals and integrated proprietary IP to meet the needs of your applications.

Chapter 2 describes the installation and configuration of the CY3214-PSoCEVALUSB Kit, PSoC Designer and PSoC Programmer. Chapter 3 describes the kit operation, with the help of the 'Square Mouse' project, which comes as a demonstration with the CY3214-PSoCEVALUSB kit. It also explains the programming connections of a PSoC 1 device with the PSoC MiniProg and power supply. Chapter 4 describes the hardware operations. Chapter 5 provides the description of the execution of simple projects. The Appendix section provides the schematics and bill of materials (BOM).

## 1.1 Kit Contents

The CY3214-PSoCEVALUSB Kit includes:

- PSoCEVALUSB board
- LCD module
- MiniProg Programmer
- USB A to Mini B cable
- CY3214-PSoCEVALUSB Kit CD
  - PSoC Designer installation file
  - PSoC Programmer installation file
  - Code examples
  - Hardware files
  - Kit guide
  - Quick start guide
  - Release notes
- Jumper wire pack
- CY8C24894-24LTXI sample silicon

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

## 1.2 Additional Learning Resources

Visit [www.cypress.com](http://www.cypress.com) for additional learning resources in the form of data sheets, technical reference manuals, and application notes.

### 1.2.1 Reference Documents

- Schematic diagram for the CY3214 PSoCEVALUSB board:  
<http://www.cypress.com/go/CY3214-PSoCEvalUSB>
- CY8C24894-24LTXI data sheet:  
<http://www.cypress.com/?mpn=CY8C24894-24LTXI>
- PSoC MiniProg1 details:  
<http://www.cypress.com/go/CY3217-MiniProg1>
- ICE-Cube details:  
<http://www.cypress.com/go/CY3215-DK>
- 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>

## 1.3 Document History

Revision	PDF Creation Date	Origin of Change	Description of Change
**	01/31/2011	RKPM	Initial version of kit guide
*A	04/27/2011	SASH	Added My First Code Example section. Content updates throughout the document
*B	05/03/2012	RKPM	Updated screenshots in the Kit Installation section. Updated Figure 5-13. Updated installer file names.

## 1.4 Documentation Conventions

Table 1-1. Document Conventions for Guides

Convention	Usage
Courier New	Displays file locations, user entered text, and source code: <code>C:\...cd\icc\</code>
<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
<b>Bold</b>	Displays commands, menu paths, and icon names in procedures: Click the <b>File</b> icon and then click <b>Open</b> .
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 CY3214-PSoCEVALUSB 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/CY3214-PSoCEvalUSB>. Three different types of installers are available for download.

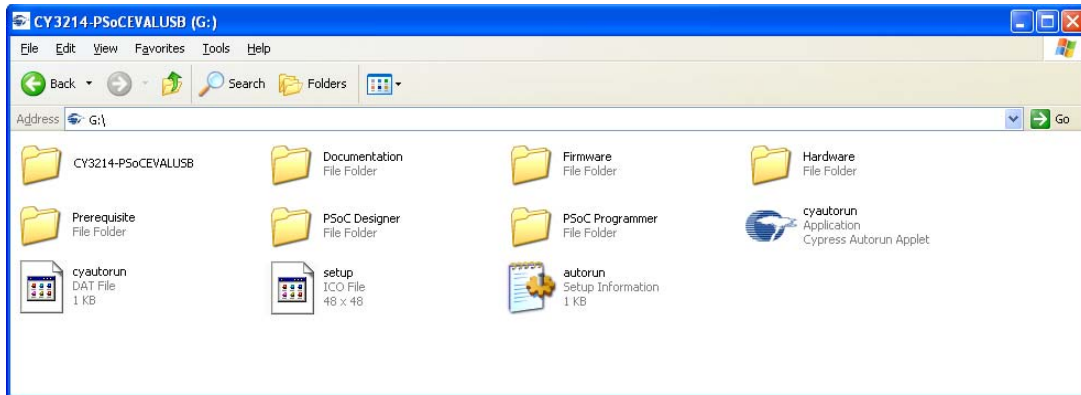
- a. CY3214-PSoCEVALUSB Kit ISO: This file (ISO image) is an archive file of the optical disc provided with the kit. You can use this to create an installer CD or extract information using WinRAR or similar tools.
  - b. CY3214-PSoCEVALUSB Kit Setup: This executable file installs the contents of the kit CD, which includes PSoC Programmer, PSoC Designer, kit code examples, kit hardware files, and user documents.
  - c. CY3214-PSoCEVALUSB Kit Only (without prerequisites): This executable file installs only the kit contents, which includes kit code examples, hardware files, and user documents.
2. Click **Install the CY3214-PSoCEVALUSB** to start the installation, as shown in Figure 2-1.

Figure 2-1. Kit Installer Startup Screen



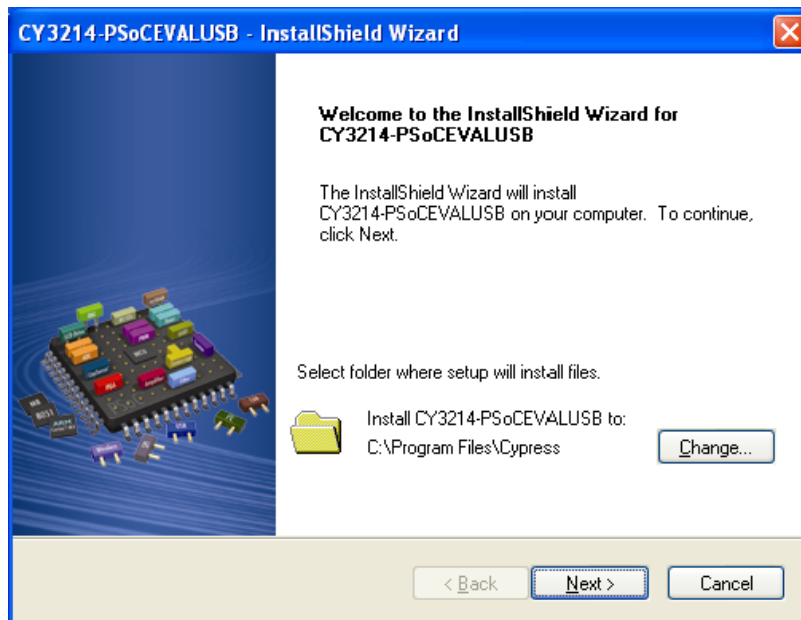
**Note** If auto-run does not execute, double-click **cyautorun.exe** file on the root directory of the CD/DVD, as shown in [Figure 2-2](#).

Figure 2-2. Root directory of the CD/DVD



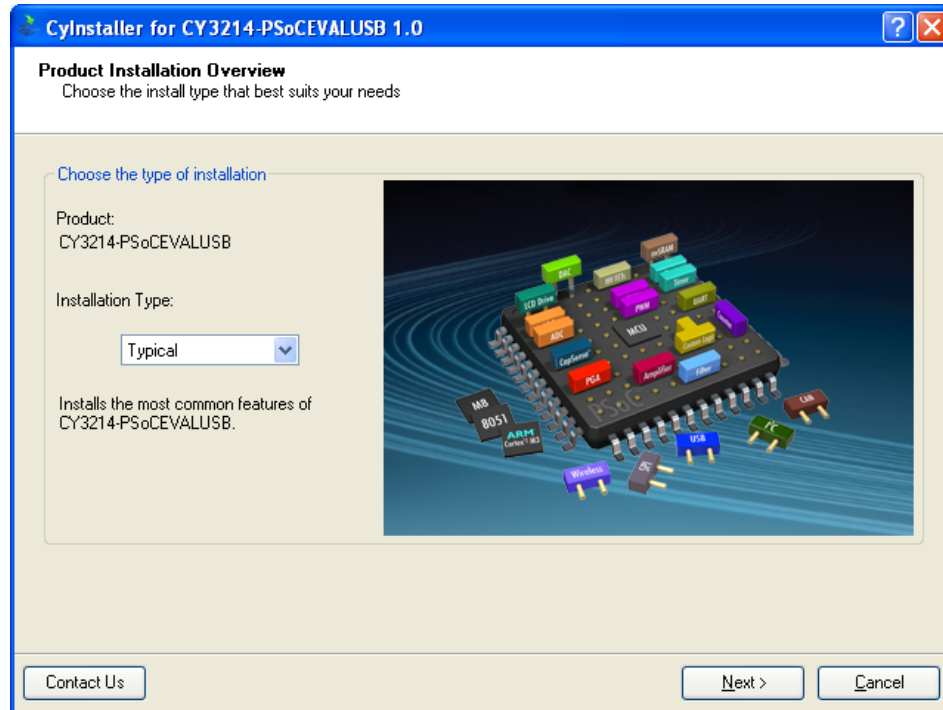
3. On the startup screen, click **Next** to start the installer.
4. The **InstallShield Wizard** screen is displayed, with the default location for setup. You can change the location using **Change**, as shown in [Figure 2-3](#).
5. Click **Next** to launch the kit installer.

Figure 2-3. InstallShield Wizard



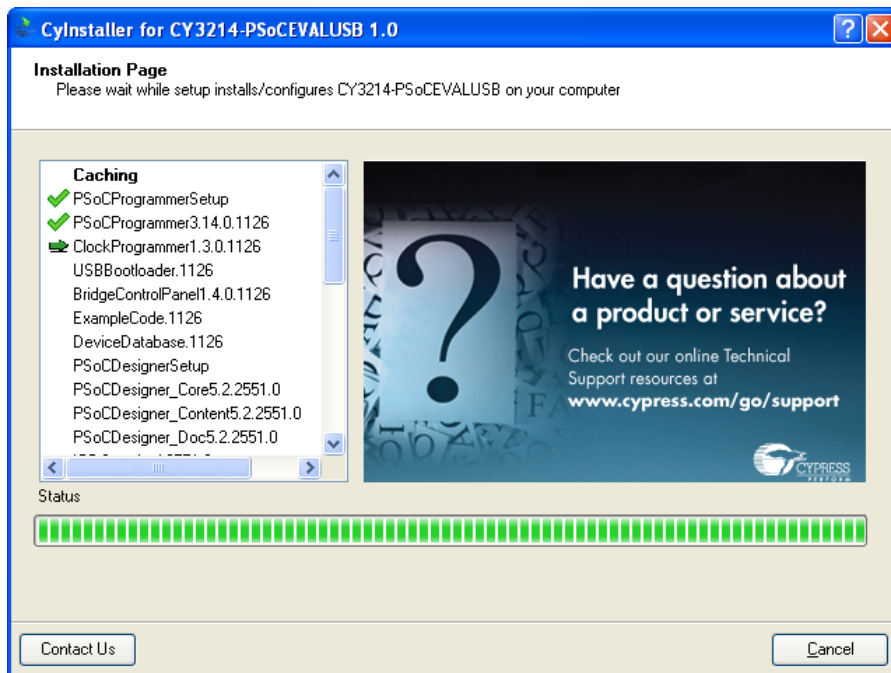
6. On the **Product Installation Overview** screen, select the installation type that best suits your requirement. The drop-down menu has three options: **Typical**, **Complete**, and **Custom**; see [Figure 2-4](#).
7. Click **Next** to continue the installation.

Figure 2-4. Installation Type Options



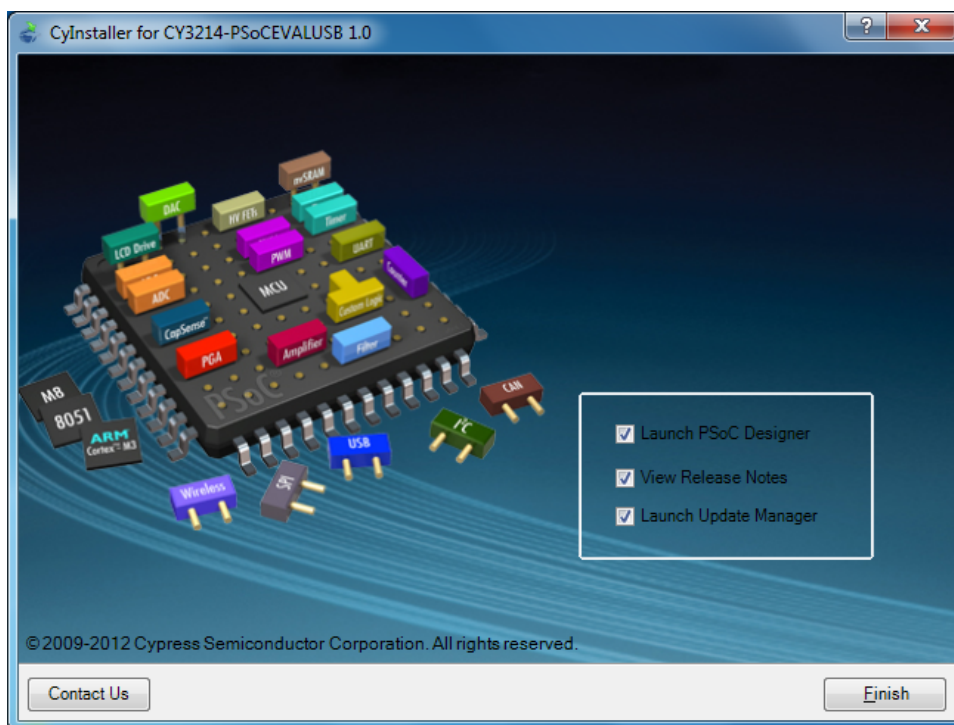
8. When the installation begins, a list of all packages appears on the **Installation Page**.
9. A green check mark appears adjacent to every package that is cached and installed. Wait until all the packages are downloaded and installed successfully.

Figure 2-5. Installation Page



10. Click **Finish** to complete the installation of the CY3214-PSoCEVALUSB Kit, as shown in [Figure 2-6](#).

Figure 2-6. Installation Completion Page



**Note** Advanced users can skip to [Code Examples chapter on page 31](#).

## 2.2 PSoC Designer

PSoC Designer is the revolutionary easy-to-use Integrated Development Environment (IDE) that helps to customize PSoC to meet your specific application requirements. PSoC Designer accelerates system bring-up and time-to-market.

1. Click **Start > All Programs > Cypress > PSoC Designer <version> > PSoC Designer <version>**.
2. To create a new project, click **File > New Project**; click **File > Project Workspace** to open an existing project.

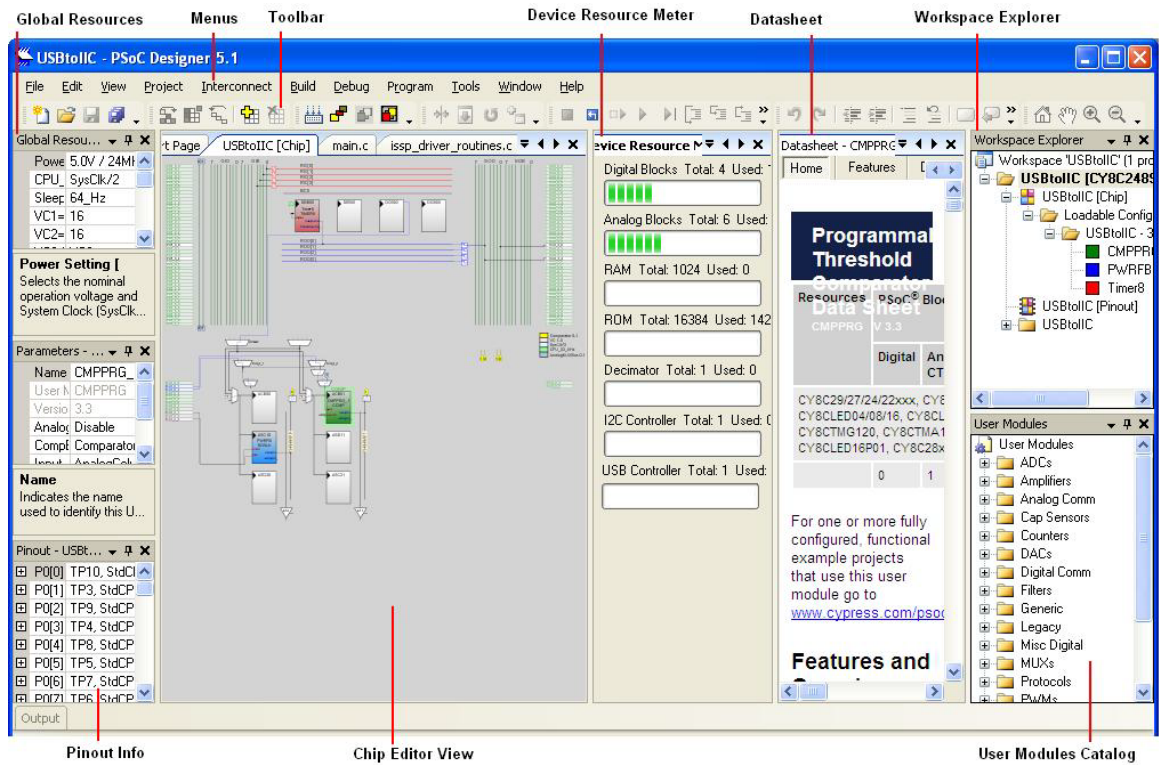
See [Additional Learning Resources on page 6](#) for links to PSoC Designer training.

The PSoC Designer quick start guide is available at: <http://www.cypress.com/?rID=47954>.

For more details on PSoC Designer, go to:

```
<Install_Directory>:\Cypress\PSoC Designer\<version>\Documentation
```

Figure 2-7. PSoC Designer Interconnect View





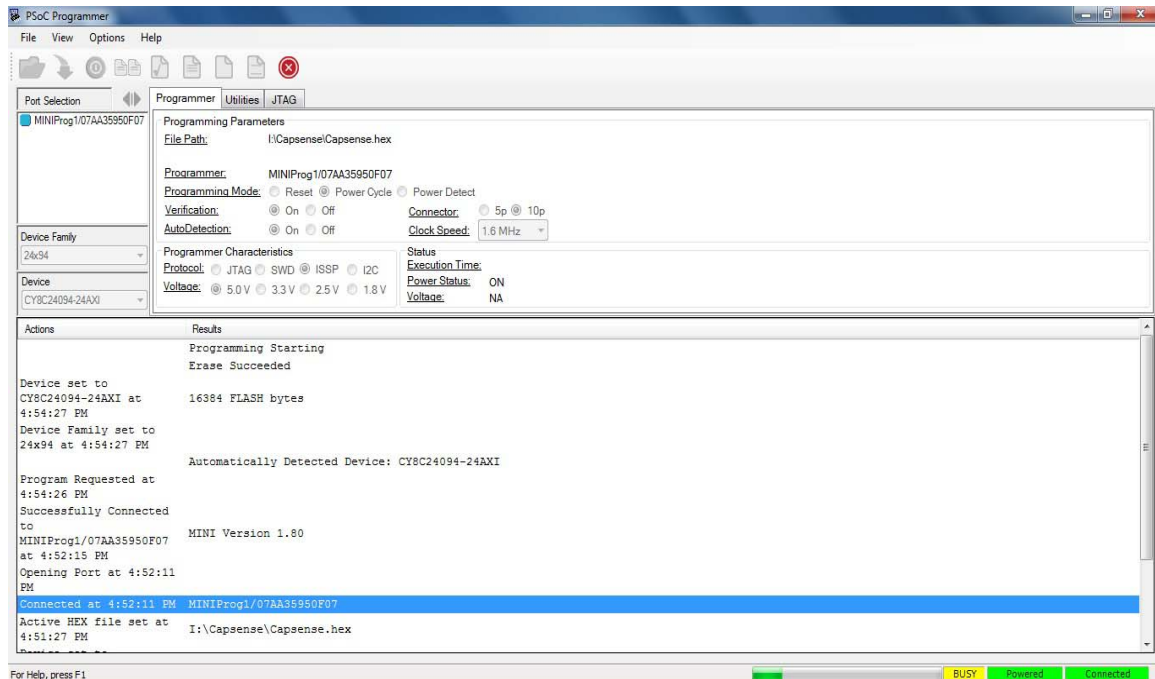
## 2.3 PSoC Programmer

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

To successfully program the device, follow these steps:

1. Use the **Power** button to power up the CY3214-PS0CEVALUSB kit before loading the hex file on to the kit.

Figure 2-8. PSoC Programmer



2. Click the **File Load** button to load the respective hex file.
3. Click the **Program** button to program the hex file to the chip.
4. When programming is successful, **Programming Succeeded** appears in the Actions pane.
5. Close PSoC Programmer.

**Note** For more details on PSoC Programmer, go to:

<Install\_Directory>:\Cypress\Programmer\<version>\Documents

## 2.4 SuiteUSB 3.4

The Cypress SuiteUSB 3.4 is a generic USB driver environment that you can freely use to develop host applications that interact with a USB device.

CyConsole is used to communicate with USB devices that match the CyUSB.sys device driver. The console supports hot-plugging of USB devices. When a device that matches the driver is plugged into the PC, it automatically appears in the device list near the top of the CyConsole window.

SuiteUSB 3.4 software can be installed from the link <http://www.cypress.com/?rID=34870>.

## 3. Kit Operation



### 3.1 Introduction

The CY3214-PSoCEVALUSB kit examples helps you to exchange data between a PC application and real world I/O. The kit includes a full-speed USB User Module with programmable I/O user modules to provide instant connection.

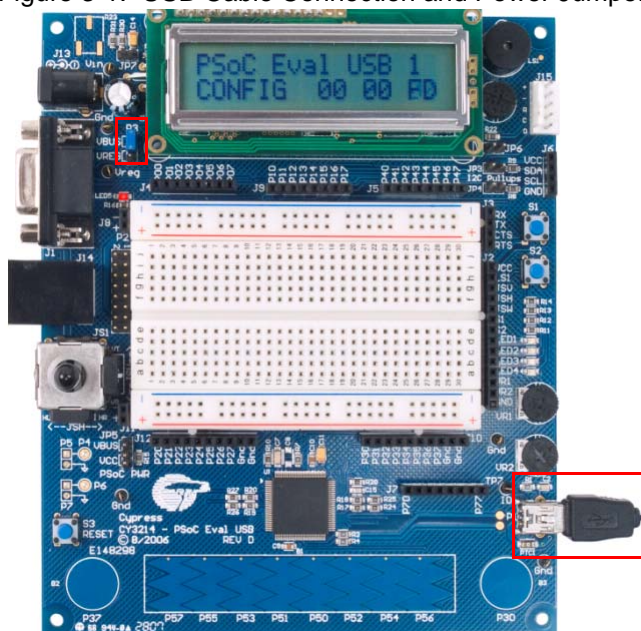
### 3.2 Square Mouse Example

The Square Mouse example is programmed to the CY3214-PSoCEVALUSB board as a default project for demonstration purposes. If the PSoC on the board has been reprogrammed or erased, follow the steps in the section, [Programming Specifications and Connections on page 16](#) to program the board with the square mouse hex file provided in the kit CD (<Install\_Directory>:\Cypress\CY3214-PSoCEVALUSB\<version>\Firmware\squaremouse2\_reset.hex).

To execute the program, follow these steps:

1. Connect the USB2 cable to the Mini USB type B connector on P1, as shown in [Figure 3-1](#). Ensure that the power selection jumper (shunt) is placed on the first two pins (designated as VBUS).
2. The message "PSoC Eval USB 1" is displayed on the LCD.
3. Check the movement of the mouse pointer on the screen; note that it moves in a square fashion.

Figure 3-1. USB Cable Connection and Power Jumper Setting



### 3.2.1 Programming Specifications and Connections

When the MiniProg is connected, you can use PSoC Programmer to program the CY3214-PSoCEVALUSB kit. Plug in the USB cable into the MiniProg before attaching it to the ISSSP 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

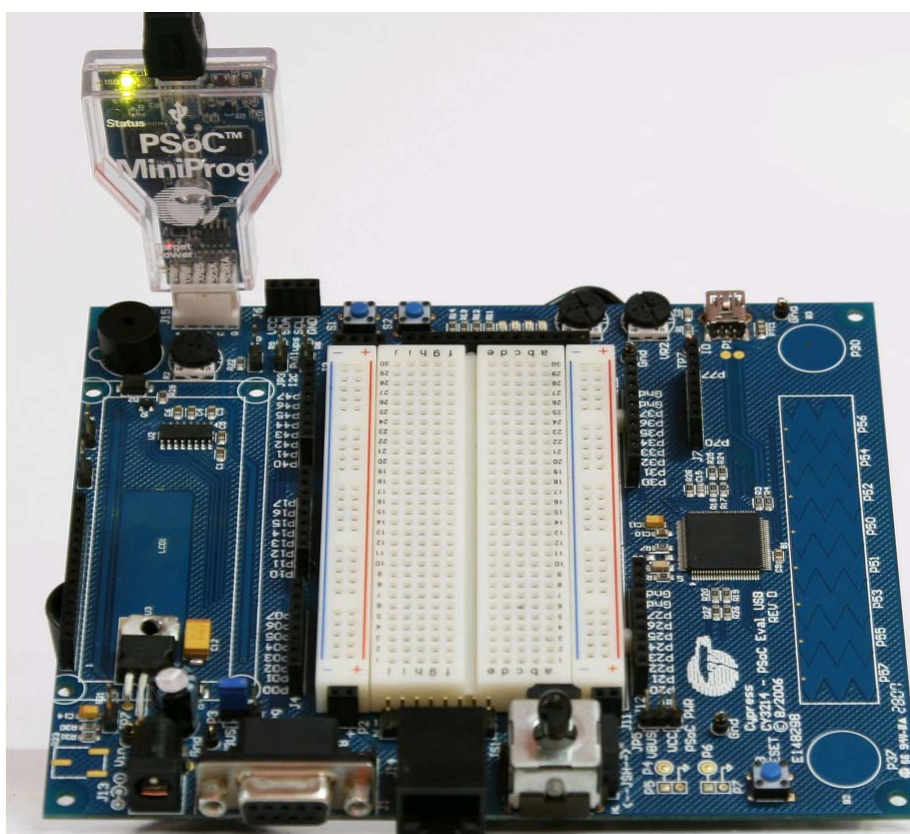
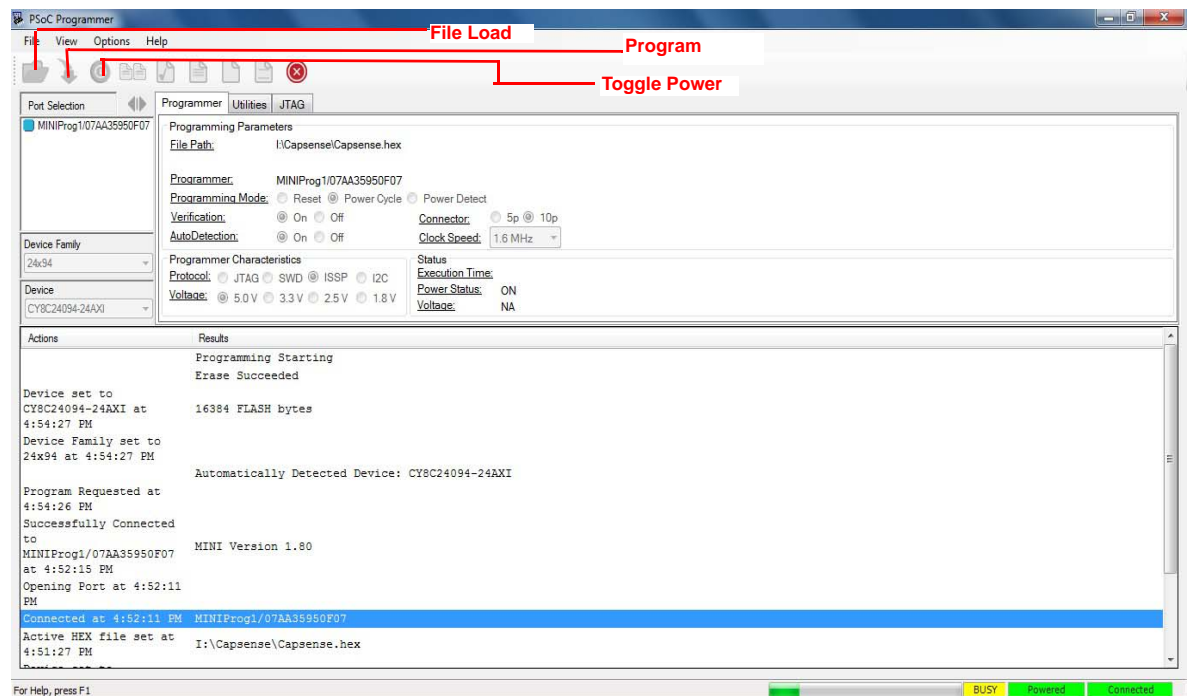




Figure 3-3. PSoC Programmer Screen



Follow these steps to program using MiniProg1:

1. Connect the USB cable to the PC and MiniProg1
2. Plug in the MiniProg1 to the ISSP header on the CY3214-PSoCEVALUSB board
3. When USB is connected to the MiniProg1, the LED (green) glows in the MiniProg1
4. Open PSoC Programmer
5. Click the **Load File** button and browse to the hex file location. 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. Click **Program** or press **[F5]** to initiate programming
8. The green LED on the MiniProg1 blinks to indicate the progress of programming
9. After successful programming, the red LED on MiniProg1 is powered off
10. Select the **Toggle Power** button in PSoC Programmer to power the board and verify output



## 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 chapter on page 31](#).

### 4.1 System Block Diagram

The CY3214-PSoCEVALUSB Kit has the following sections:

- Character LCD module
- Buzzer (LS1)
- Contrast potentiometer (R2)
- Buttons (S1 and S2)
- ISSP connector(J15)
- LEDs (LED 1–4)
- Potentiometer (VR1 and VR2)
- USB connector (P1)
- CapSense buttons (P3[0], P3[7]), and sliders (Port5)
- Power supply connector (J13)
- Joystick (JS1)
- Reset button (S3)
- Legacy DB9 connector (J1)
- ICE-Cube connector (J14)

Figure 4-1. CY3214-PSoCEVALUSB Board

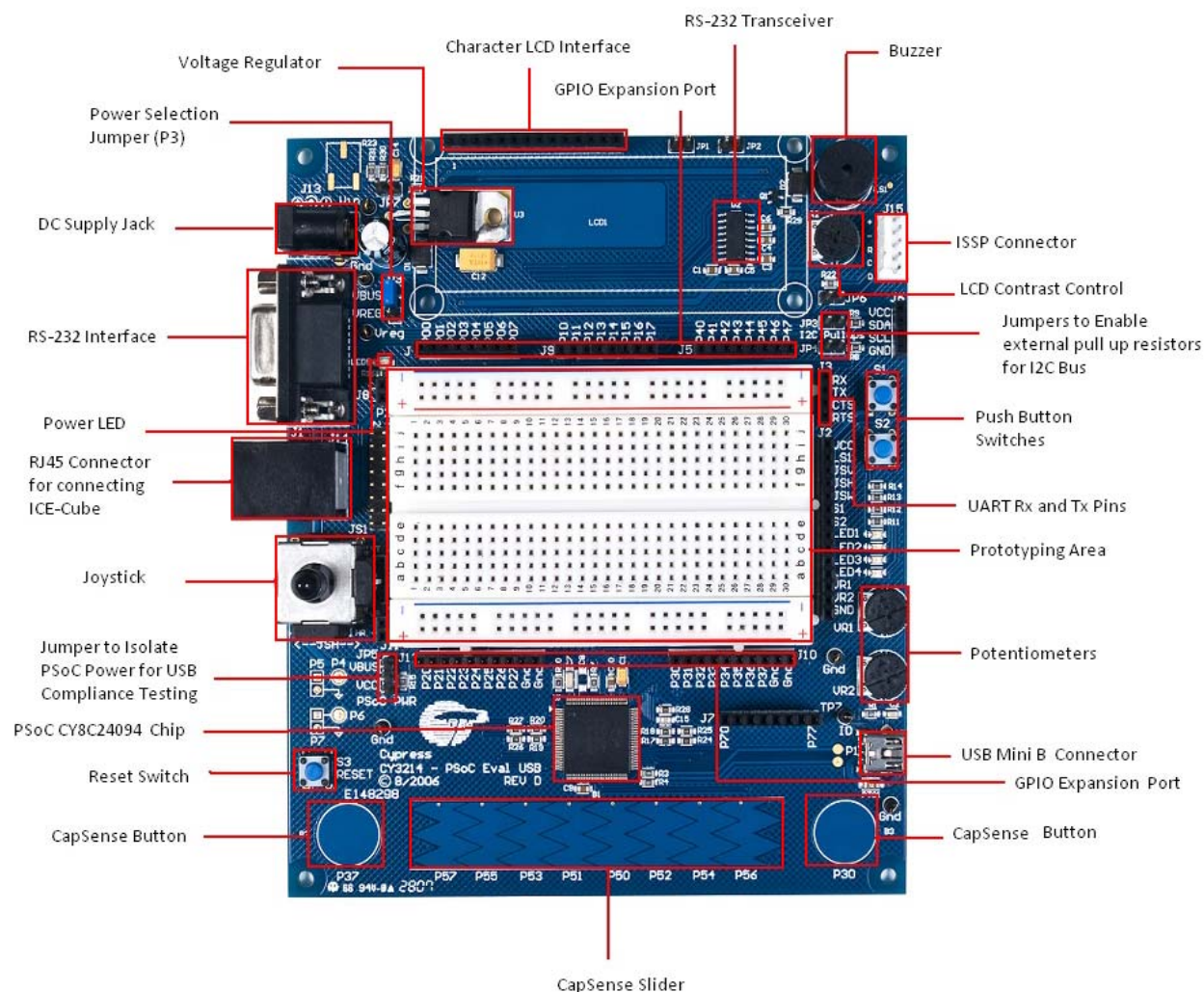


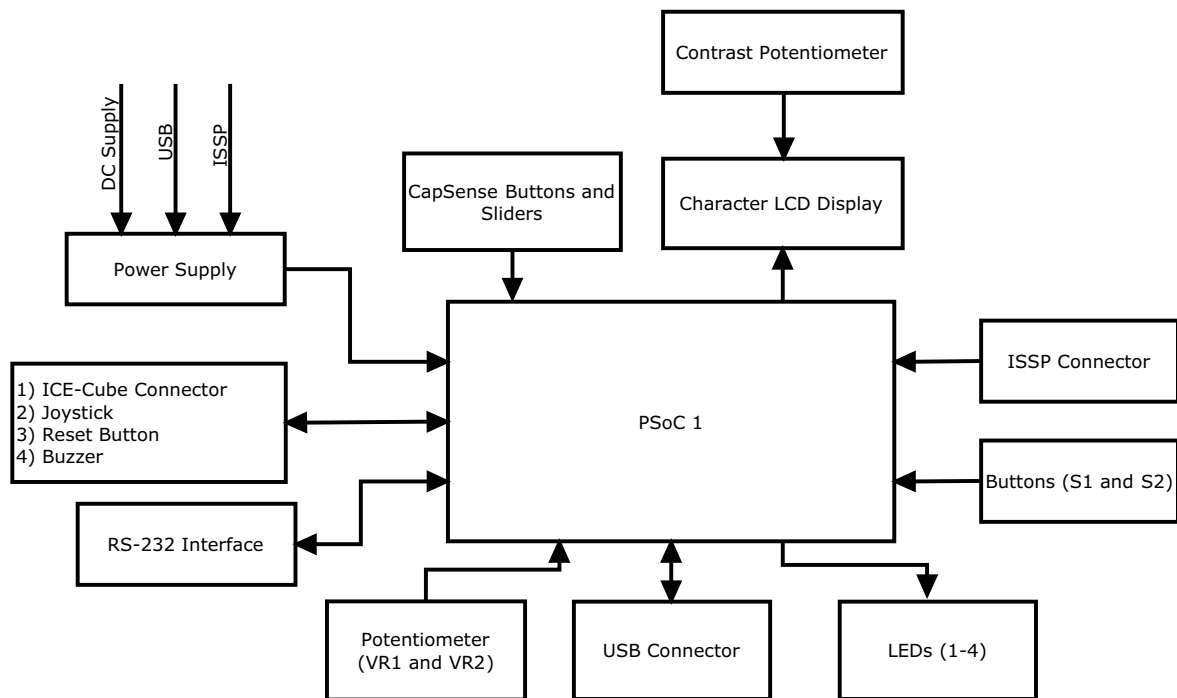
Table 4-1. Functional Pin Table

Pin No.	Pin Name	Description
1	P2[3]	Direct switched capacitor block input
2	P2[1]	Direct switched capacitor block input
3	P4[7]	GPIO
4	P4[5]	GPIO
5	P4[3]	GPIO
6	P4[1]	GPIO
7	P3[7]	GPIO
8	P3[5]	GPIO
9	P3[3]	GPIO
10	P3[1]	GPIO
11	P5[7]	GPIO
12	P5[5]	GPIO
13	P5[3]	GPIO
14	P5[1]	GPIO

Table 4-1. Functional Pin Table

Pin No.	Pin Name	Description
15	P1[7]	I <sup>2</sup> C serial clock (SCL)
16	P1[5]	I <sup>2</sup> C serial data (SCA)
17	P1[3]	GPIO
18	P1[1]	I <sup>2</sup> C serial clock (SCL), ISSP SCLK[2]
19	Vss	Ground connection
20	D+	GPIO
21	D-	GPIO
22	Vdd	Supply voltage
23	P7[7]	GPIO
24	P7[0]	GPIO
25	P1[0]	I <sup>2</sup> C serial data (SDA), ISSP SDATA[2]
26	P1[2]	GPIO
27	P1[4]	Optional external clock input (EXTCLK)
28	P1[6]	GPIO
29	P5[0]	GPIO
30	P5[2]	GPIO
31	P5[4]	GPIO
32	P5[6]	GPIO
33	P3[0]	GPIO
34	P3[2]	GPIO
35	P3[4]	GPIO
36	P3[6]	GPIO
37	P4[0]	GPIO
38	P4[2]	GPIO
39	P4[4]	GPIO
40	P4[6]	GPIO
41	P2[0]	Direct switched capacitor block input
42	P2[2]	Direct switched capacitor block input
43	P2[4]	External analog ground (AGND) input
44	P2[6]	External voltage reference (VREF) input
45	P0[0]	Analog column mux input
46	P0[2]	Analog column mux input
47	P0[4]	Analog column mux input VREF
48	P0[6]	Analog column mux input
49	Vdd	Supply voltage
50	Vss	Ground connection
51	P0[7]	Analog column mux input
52	P0[5]	Analog column mux input and column output
53	P0[3]	Analog column mux input and column output
54	P0[1]	Analog column mux input
55	P2[7]	GPIO
56	P2[5]	GPIO

Figure 4-2. Hardware Block Diagram

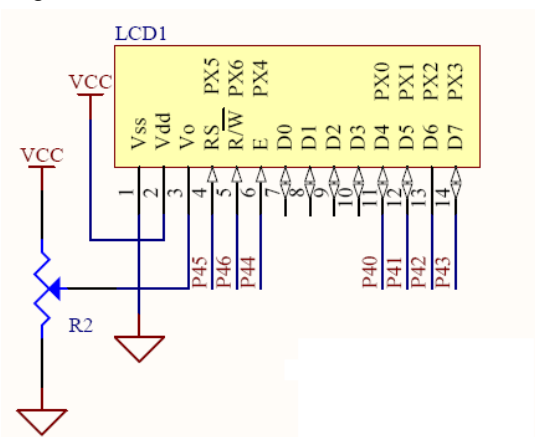


## 4.2 Functional Description

### 4.2.1 Character LCD Module

The kit has a character LCD module, which goes into the character LCD header, P8. The LCD runs on a 5-V supply and can function regardless of the voltage on which PSoc is powered. There is a zero-ohm resistor setting available on the LCD section making it possible to convert it to a 3.3-V LCD. The variable resistor R2 is a contrast potentiometer. It is used to control the LCD contrast.

Figure 4-3. LCD Module



## 4.2.2 CapSense® Buttons (P3[0], P3[7]), and Sliders (Port5)

There are two CapSense buttons and eight element CapSense sliders on the CY3214-PSoCEVALUSB kit. CapSense Sigma Delta uses an external modulation capacitor (CMOD), the copper sensor pad on the CY3214 PSoCEVALUSB board, and bleed resistor (Rb) in addition to the sensor capacitor (Cx).

**Bleed Resistor (Rb):** The CSD uses the switched capacitor circuitry to convert the sensor capacitance into a voltage, which is compared to a reference voltage. When the capacitor voltage reaches the reference voltage, the comparator triggers a bleed resistor discharging the capacitor. After the capacitor voltage discharges below the reference voltage, the bleed resistor is left floating to allow the capacitor to continue charging. The comparator output becomes a bit-stream as it toggles the bleed resistor and manipulates its input voltage. This bit stream is ANDed with a pulse width modulator (PWM) to provide consistent stream framing. The number of counts in each frame is analyzed to determine if the capacitive sensor is activated.

**Modulation Capacitor (CMOD):** The capacitor CMOD acts to attenuate high-frequency noise. The pin assignment for CapSense buttons used in the CapSense code example (see [My First Code Example on page 31](#)) are as follows:

B2 – P3[7]

B3 – P3[0]

The slider segments pin assignment used in [My First Code Example on page 31](#) are as follows:

Slider 0 – P5[7]

Slider 1 – P5[5]

Slider 2 – P5[3]

Slider 3 – P5[1]

Slider 4 – P5[0]

Slider 5 – P5[2]

Slider 6 – P5[4]

Slider 7 – P5[6]

Figure 4-4. CapSense Sliders

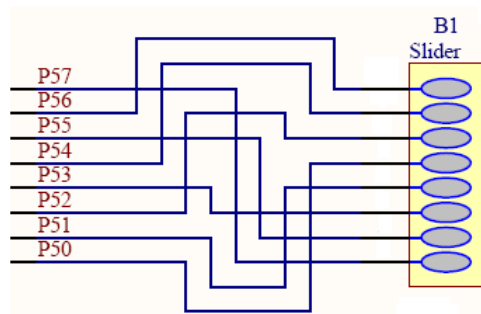
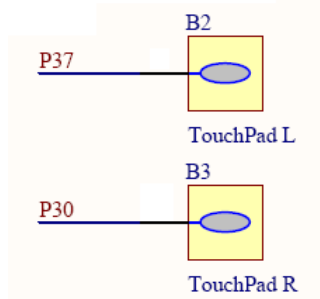


Figure 4-5. CapSense Buttons



### 4.2.3 Legacy DB9 Connector (J1)

This connector is used for serial interface. Connect the serial cable to the DB9 connector and the serial port on the PC. The supply voltage is 3.3 V to 5 V; the output voltage  $V_{out}$  (high) is  $V_{cc}-0.6$  V and  $V_{out}$  (low) is 0.4 V.

Figure 4-6. DB9 Connector

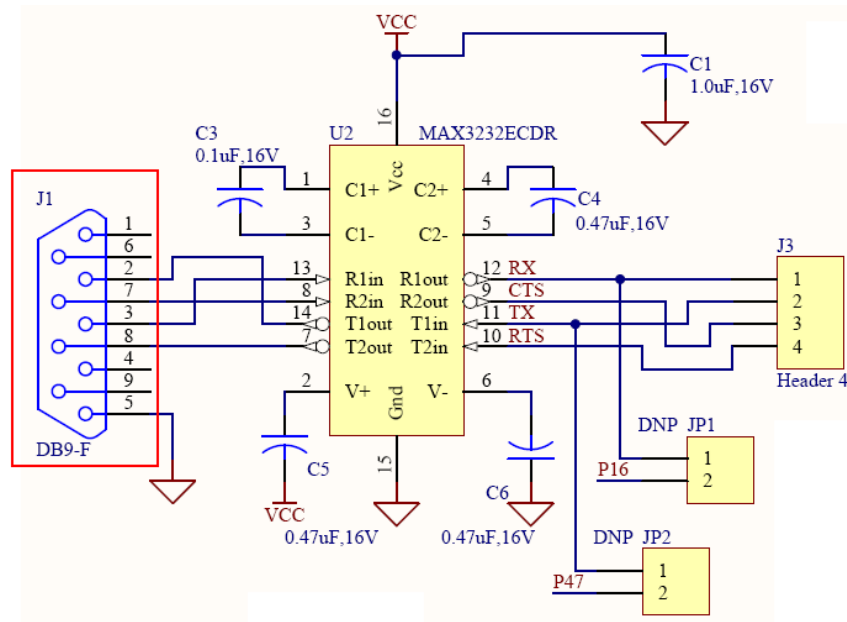


Table 4-2. IC Description

Pin No.	Pin Name	Description
1	C1+	Connected to +ve of capacitor C3
2	V+	Vcc supply
3	C1-	Connected to -ve of capacitor C3
4	C2+	Connected to +ve of capacitor C4
5	C2-	Connected to -ve of capacitor C4
6	V-	Vcc ground
7	T2 out	Request to send (OUT)



Table 4-2. IC Description

Pin No.	Pin Name	Description
8	R2in	Clear to send (IN)
9	R2out	Clear to send (OUT)
10	T2in	Request to send (IN)
11	T1in	Transmitted data (IN)
12	R1out	Receive data (OUT)
13	R1in	Receive data (IN)
14	T1out	Transmitted data (OUT)
15	Gnd	Ground
16	Vcc	Vcc supply

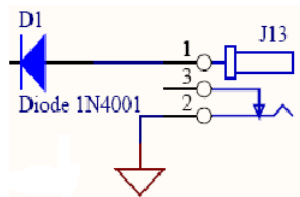
Table 4-3. DB9 Connector Description

Pin No	Pin Name	Description
1	DCD	Carrier detect
2	RD	Receive data
3	TD	Transmit data
4	DTR	Data terminal ready
5	SGND	System ground
6	DSR	Data set ready
7	RTS	Request to send
8	CTS	Clear to send
9	RI	Ring indicator

#### 4.2.4 Power Supply Connector (J13)

You can power the board from the USB cable or from an external 9 V to 12 V power supply. D1 is used as a protecting diode. The protecting diodes are necessary if the load discharges are slower than the filter capacitor after the rectification process.

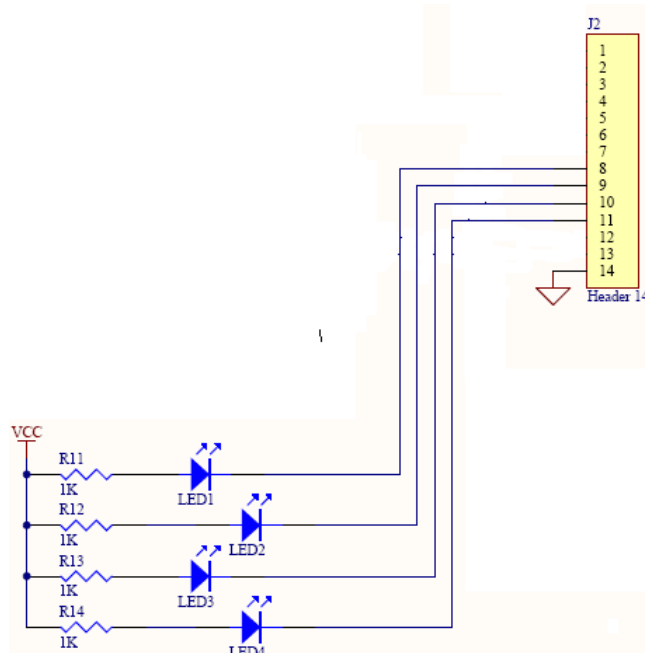
Figure 4-7. Power Supply Connector



#### 4.2.5 LEDs (LED1-4)

Four LEDs are accessible through headers LED1-4. LEDs are illuminated by connecting the signal to Vss (active low, set drive to strong, drive low). These LEDs are active low.

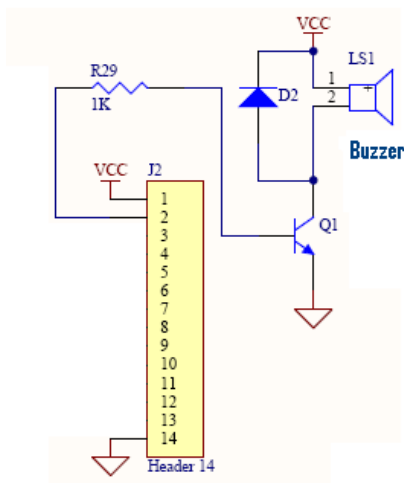
Figure 4-8. LEDs



#### 4.2.6 Buzzer (LS1)

The buzzer is accessible through LS1.

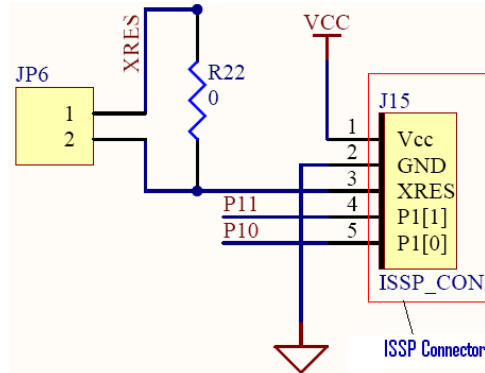
Figure 4-9. Buzzer



#### 4.2.7 ISSP Connector (J15)

This is used to program the device using the MiniProg device and the USB cable. To load a hex file to the PSoC device chip, plug the MiniProg device to the jumper connector (J15).

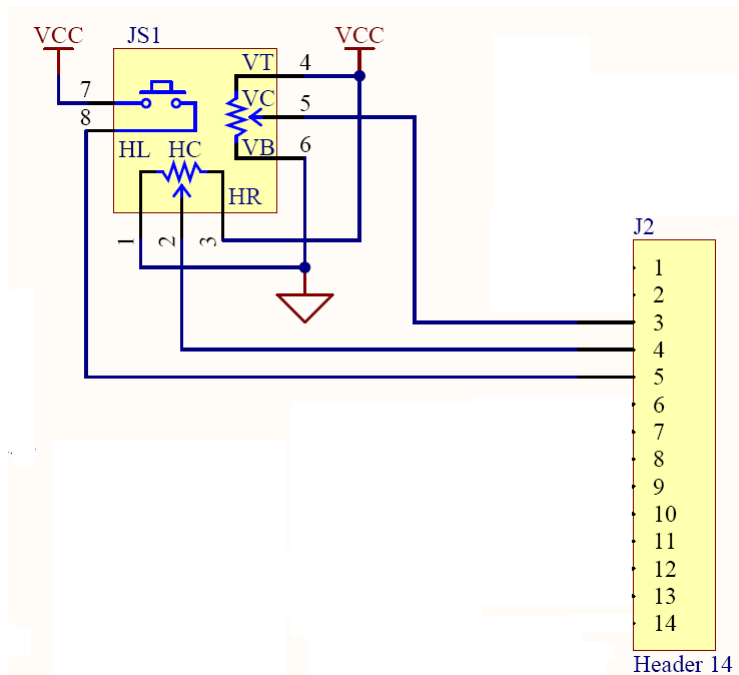
Figure 4-10. ISSP Connector



#### 4.2.8 Joystick (JS1)

A joystick is used to create a game. JSV (vertical pot value), JSH (horizontal pot value), and JSW (switch) are accessible through the header on the board.

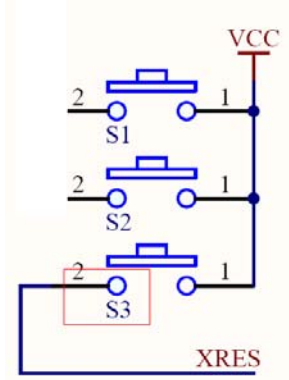
Figure 4-11. Joystick



#### 4.2.9 Reset Button (S3)

The reset button is used to reset the CY3214-PSoCEVALUSB kit.

Figure 4-12. Reset Button



#### 4.2.10 ICE-Cube Connector (J14)

The ICE-Cube (available in the CY3215-DK Kit) provides significant debugging functionality that requires full two-way communication between the ICE and your computer. This software interface allows to run, halt, and single-step the processor and view the content of specific memory locations. It also allows to set complex event points. Event points can start and stop the trace memory on the ICE, as well as break the program execution. The ICE-Cube supports large memory model PSoC devices (with more than 256 bytes of RAM) and is backward compatible with previous PSoC devices (CY8C27x43 and CY8C24x23A).

Figure 4-13. ICE-Cube Connector

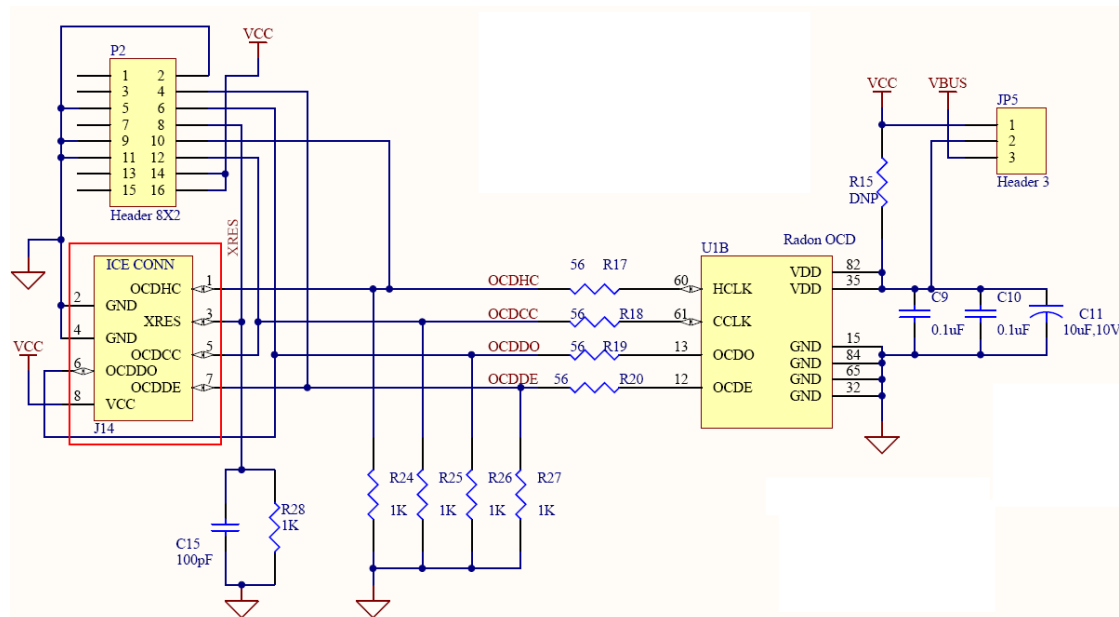


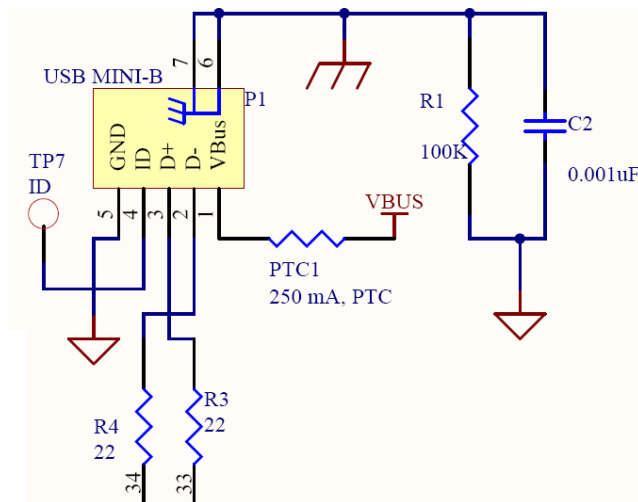
Table 4-4. Pin Description

Pin No	Pin Name	Pin Description
1	OCDHC	OCD high-speed clock output
2	GND	Ground
3	XRES	Reset signal (required only for Reset programming mode)
4	GND	Ground
5	OCDCC	OCD CPU clock output
6	OCDDO	OCD odd data output
7	OCDDI	OCD even data I/O.
8	VCC	Supply voltage

#### 4.2.11 USB Connector (P1)

The mini USB type B connector is compatible with USB 2.0 connector specifications. The RC filter is used to isolate the VBUS from the on-board voltage domain.

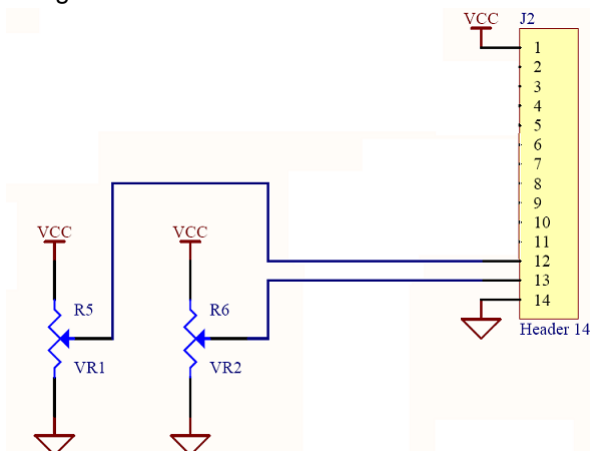
Figure 4-14. USB Connector



#### 4.2.12 Potentiometers (VR1 and VR2)

A potentiometer is a manually adjustable resistor. The potentiometer used here is of 10 K $\Omega$  variable resistors. They are tied to Vss and Vcc (set drive to analog high-Z).

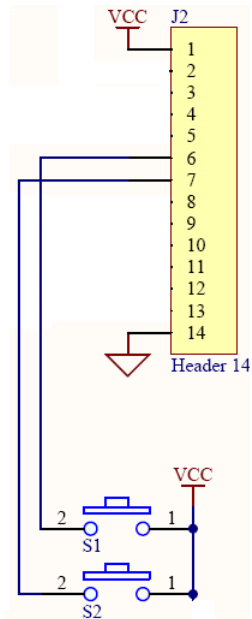
Figure 4-15. Potentiometers



### 4.2.13 Buttons (S1 and S2)

These are SPST (single pole, single throw) switches.

Figure 4-16. Buttons



## 5. Code Examples



All code examples for the CY3214-PSoCEVALUSB kit are available at the following location:  
<Install\_Directory>:\Cypress\CY3214-PSoCEVALUSB\<version>\Firmware.

### 5.1 My First Code Example

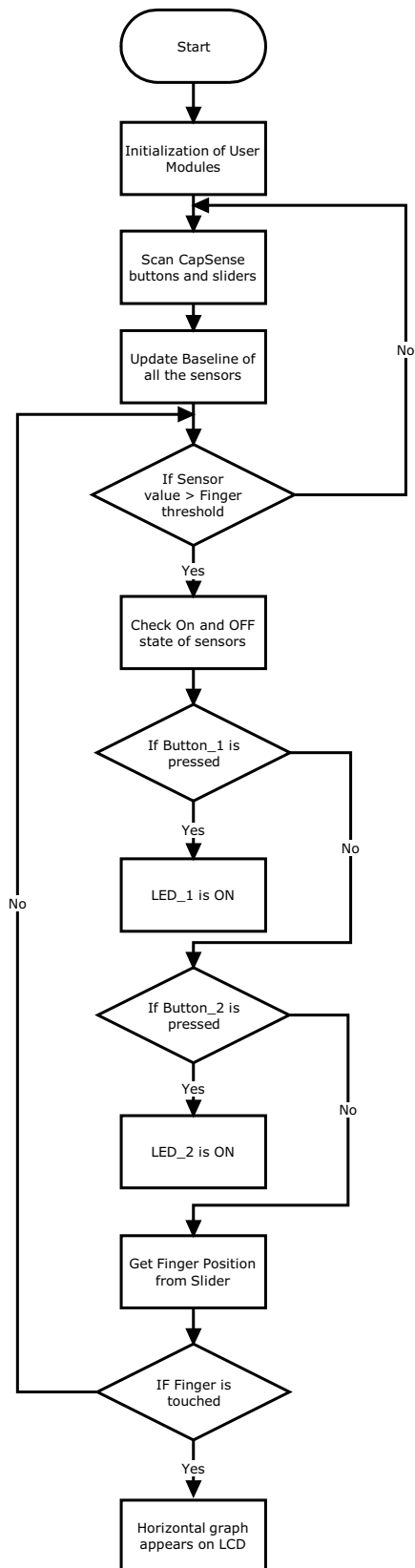
#### 5.1.1 Project Objective

This example demonstrates how to operate the CapSense buttons and sliders. When you configure CapSense buttons and sliders in the CSD wizard, the corresponding LEDs check the functionality of CapSense buttons provided on the board.

The following user modules are used in the project:

- **CSD:** The CSD (Capacitive sensing using a Sigma-Delta Modulator) provides CapSense functionality using the switched capacitor technique with a sigma-delta modulator to convert the sensing switched capacitor current to digital code.
- **LCD:** The LCD module is used to display a horizontal bar graph when the CapSense slider is touched.
- **LED (1–2):** LEDs 1–2 glows when CapSense buttons are touched.

## 5.1.2 Flowchart

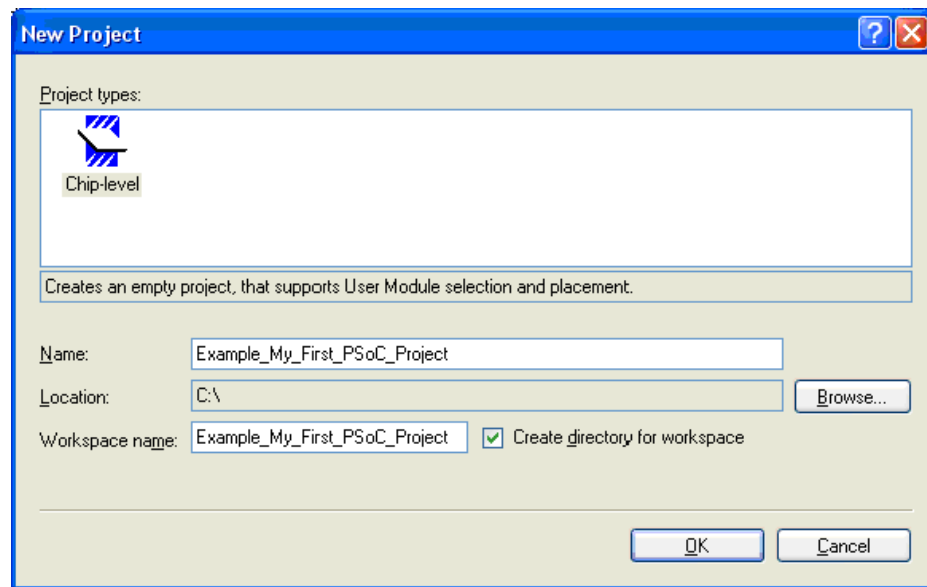




### 5.1.3 Creating My First PSoC 1 Project

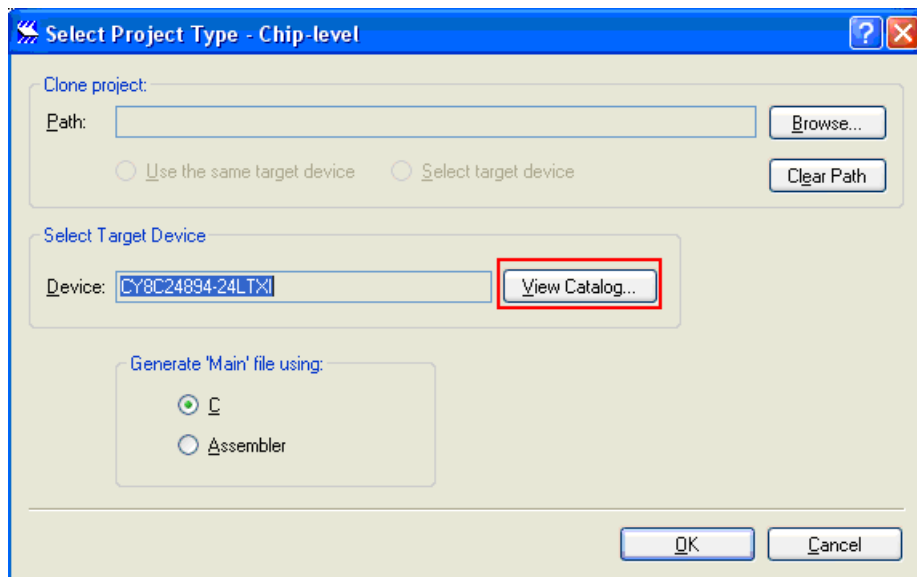
1. Open PSoC Designer.
2. To create a new project, click **File > New Project**.
3. In the New Project window, select the **Chip-level** icon. Name the project **Example\_My\_First\_PSoC\_Project**; see [Figure 5-1](#).
4. Click **Browse** and navigate to the directory in which the project is being created.

Figure 5-1. New Project Window



5. Click **OK**. The Select Project Type window opens.
6. In this window, under **Select Target Device**, click **View Catalog**, as shown in [Figure 5-2](#).

Figure 5-2. Select Project Type Window



7. The Device Catalog window opens. Click on the **All Devices** tab.

8. For this project, click **CY8C24894-24LTXI** and then click **Select**. See [Figure 5-3](#).

Figure 5-3. Device Catalog Window

Device Catalog - Chip-level

	Part Number	Analog Blocks	Digital Blocks	Flash	RAM	IO Count	Supply Voltage	SMP	USB Inter
	Click here to Remove All Filters	all	all	all	all	all	all	all	all
WUSB	CY8C23433-24PVXI	4	4	8K	256	26	3.0 to 5.25	N/A	M
	CY8C23533-24LQXI	4	4	8K	256	24	3.0 to 5.25	N/A	M
	CY8C24123A-24PXI	6	4	4K	256	6	2.4 to 5.25	N/A	M
	CY8C24123A-24SXI	6	4	4K	256	6	2.4 to 5.25	N/A	M
USB	CY8C24223A-12PVXE	6	4	4K	256	16	4.75 to 5.25	N/A	M
	CY8C24223A-24PVXI	6	4	4K	256	16	2.4 to 5.25	Yes	M
	CY8C24223A-24PXI	6	4	4K	256	16	2.4 to 5.25	Yes	M
PSoC	CY8C24223A-24SXI	6	4	4K	256	16	2.4 to 5.25	Yes	M
	CY8C24423A-12PVXE	6	4	4K	256	24	4.75 to 5.25	N/A	M
	CY8C24423A-24LFXI	6	4	4K	256	24	2.4 to 5.25	Yes	M
Automotive	CY8C24423A-24PVXA	6	4	4K	256	24	3.0 to 5.25	N/A	M
	CY8C24423A-24PVXI	6	4	4K	256	24	2.4 to 5.25	Yes	M
	CY8C24423A-24PXI	6	4	4K	256	24	2.4 to 5.25	Yes	M
	CY8C24423A-24SXI	6	4	4K	256	24	2.4 to 5.25	Yes	M
Lighting	CY8C24533-24PVXI	4	4	8K	256	25	3.0 to 5.25	N/A	M
	CY8C24633-24PVXI	4	4	8K	256	26	3.0 to 5.25	N/A	M
	CY8C24794-24LTXI	6	4	16K	1K	50	3.0 to 5.25	N/A	Full-
	CY8C24894-24LTXA	6	4	16K	1K	49	3.0 to 5.25	N/A	M
	<b>CY8C24894-24LTXI</b>	<b>6</b>	<b>4</b>	<b>16K</b>	<b>1K</b>	<b>49</b>	<b>3.0 to 5.25</b>	<b>N/A</b>	<b>Full-</b>
	CY8C24994-24BVXI	6	4	16K	1K	56	3.0 to 5.25	N/A	Full-
	CY8C24994-24LTXI	6	4	16K	1K	56	3.0 to 5.25	N/A	Full-
All Devices	CY8C27143-24PXI	12	8	16K	256	6	3.0 to 5.25	Yes	M
	CY8C27243-24PVXI	12	8	16K	256	16	3.0 to 5.25	Yes	M

Show Part Image **Select** Close

9. Under **Generate 'Main' File Using:**, select **C** and click **OK**.

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

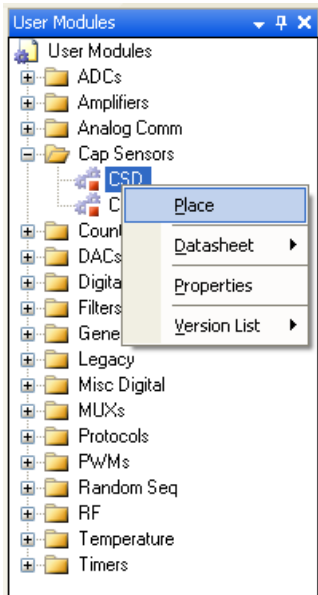
The screenshot displays the Proteus software interface for a microcontroller project. The top window, titled 'example\_m...object [Chip]', shows a pin configuration for a microcontroller. The pins are labeled 0-7, with GIO, GIE, and GOE. The middle section shows a block diagram of the microcontroller with four DCB00-DCB03 blocks, four RO0[0]-RO0[3] blocks, and a B00 block. The bottom section shows a detailed circuit diagram with various components like ACB00, ASC10, ASD20, ACB01, ASD11, ASD21, and two comparators (Comparator 0 and Comparator 1). A legend on the right identifies components like Comparator 0-1, VC 1-3, Cyclic2, CPU\_32\_KHz, and AnalogMUXBus 0-1.

11. In the User Modules window, expand the Cap Sensors folder.

The screenshot shows the 'User Modules' tree. The 'Cap Sensors' module is highlighted with a red rectangle. Below it, the 'CSD' and 'CSDADC' modules are visible.

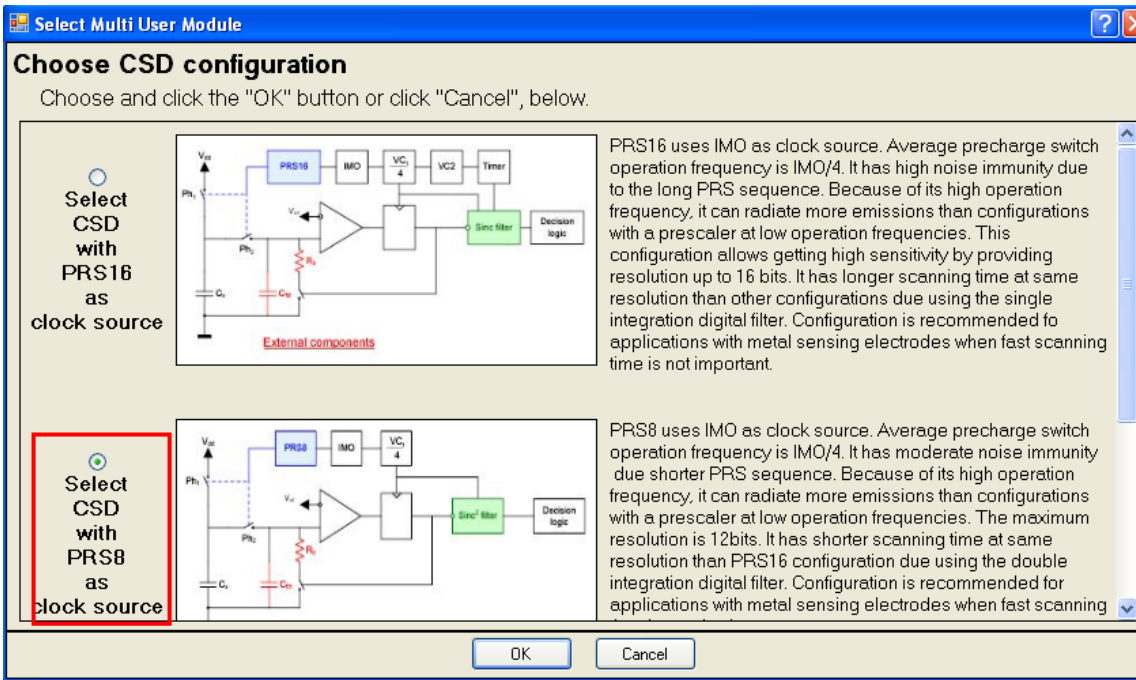
12. In the Cap Sensors folder, right-click on **CSD** to place the module.

Figure 5-6. CSD Module



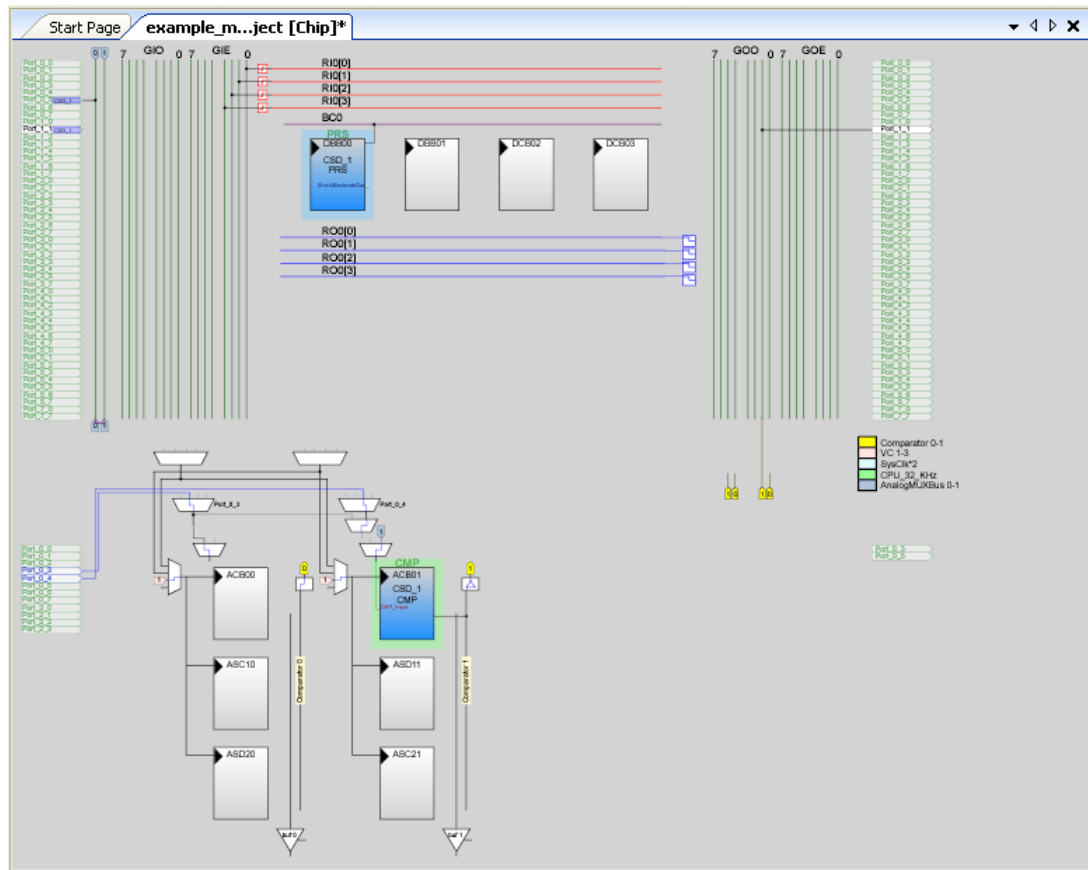
13. The splash screen "CSD Configuration" appears; select **CSD** with PRS8 as clock source.

Figure 5-7. CSD Configuration



14. The CSD module is placed in the analog and digital blocks, as shown in the following figure.

Figure 5-8. Chip View with CSD Module



15. The default parameter window of CSD\_1 is shown in Figure 5-9.

Figure 5-9. Default CSD Parameter Window

Parameters - CSD_1	
Name	CSD_1
User Module	CSD
Version	1.50
FingerThreshold	40
NoiseThreshold	20
BaselineUpdateThreshold	200
Sensors Autoreset	Enabled
Hysteresis	10
Debounce	3
NegativeNoiseThreshold	20
LowBaselineReset	50
Scanning Speed	Normal
Resolution	12
Ref Value	2
PRS Polynomial	Short
ShieldElectrodeOut	None

16. Configure the CSD Module parameter window.

Figure 5-10. CSD Parameter Window

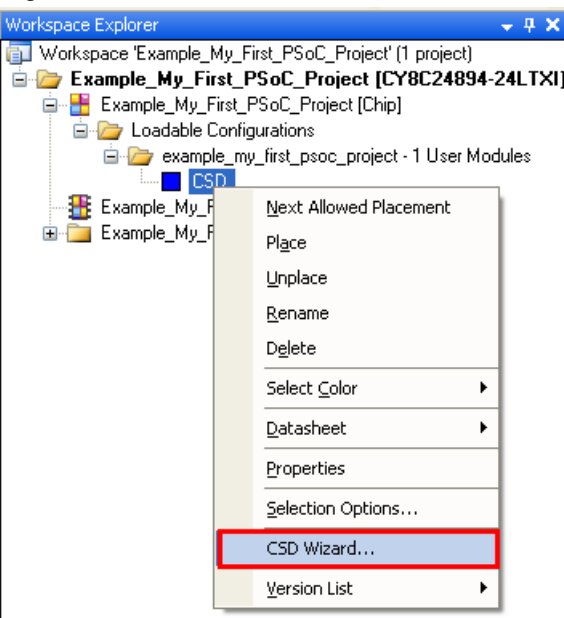
Parameters - CSD	
Name	CSD
User Module	CSD
Version	1.50
FingerThreshold	100
NoiseThreshold	40
BaselineUpdateThreshold	100
Sensors Autoreset	Disabled
Hysteresis	10
Debounce	3
NegativeNoiseThreshold	20
LowBaselineReset	50
Scanning Speed	Normal
Resolution	12
Ref Value	4
PRS Polynomial	Long
ShieldElectrodeOut	None

**Name**

Indicates the name used to identify this User Module instance

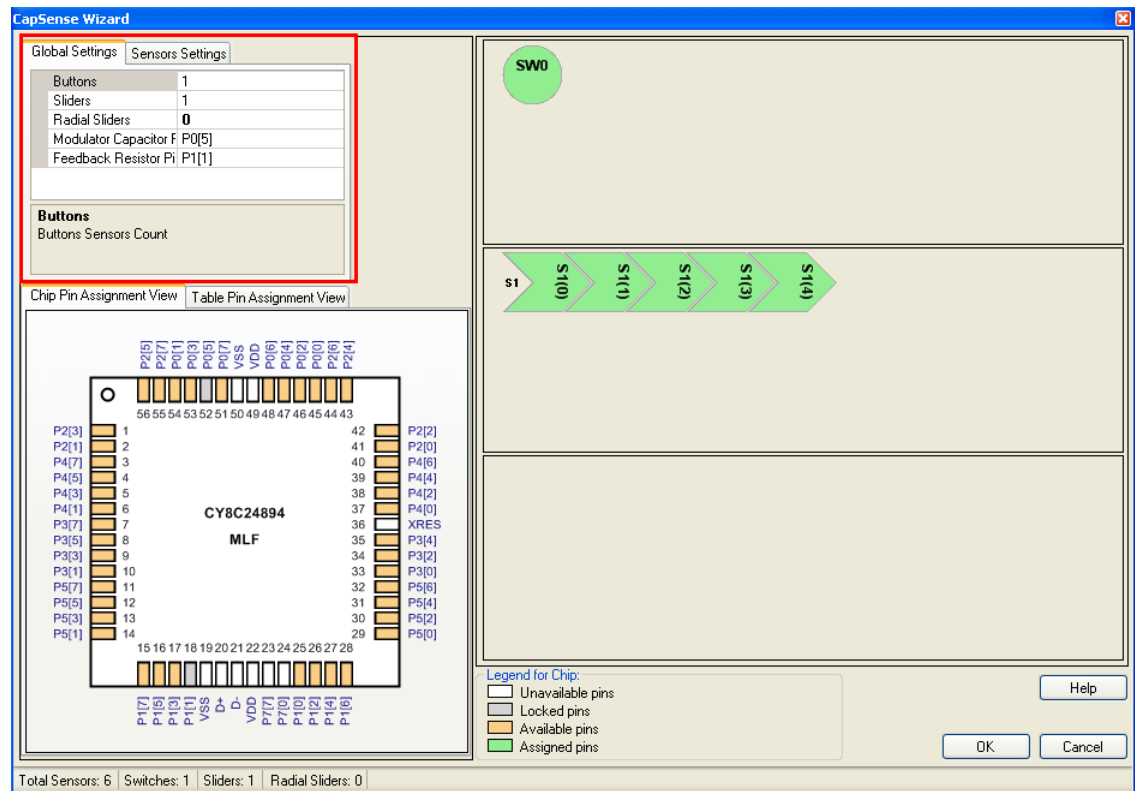
17. In the Workspace Explorer, select **CSD** and right-click to select the CSD wizard.

Figure 5-11. Select CSD Wizard



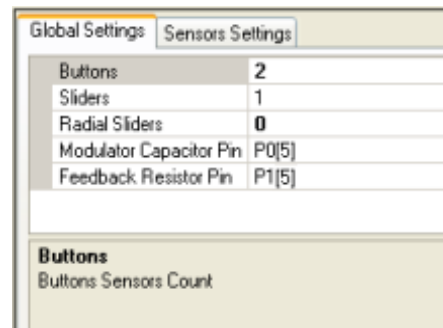
18. The default CapSense Wizard window appears. Configure the global settings to select the number of CapSense buttons and lights required for the project.

Figure 5-12. Default View of CSD Wizard



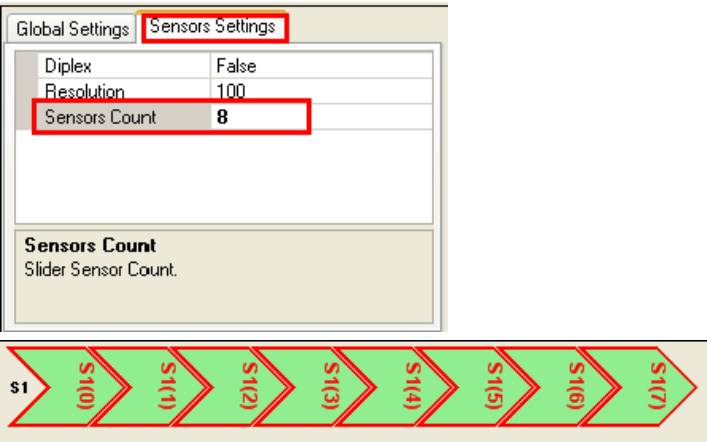
19. Configure the Global Settings window of CSD Wizard, as shown in the following figure.

Figure 5-13. Global Settings Window



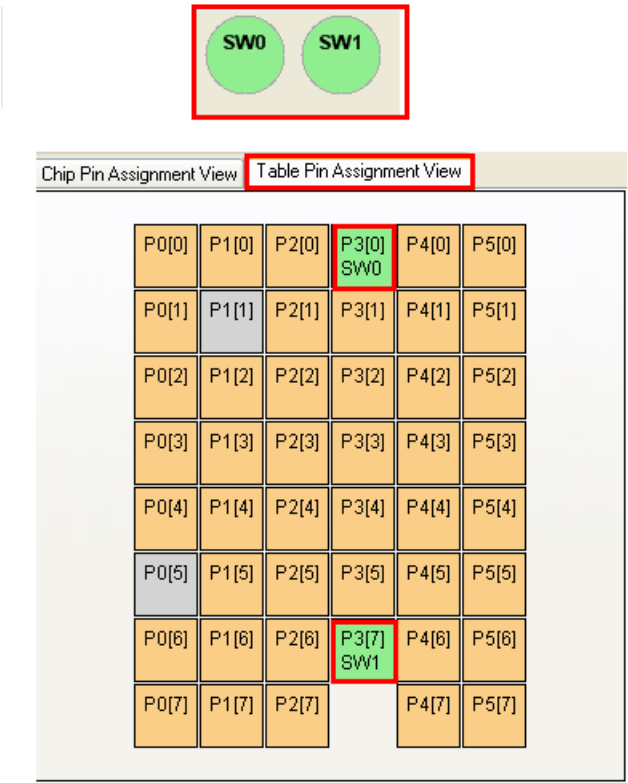
20. Similarly, click the slider once and configure the sensor settings, as shown in [Figure 5-14](#).

Figure 5-14. Sensor Settings Window



21. Select each of the CapSense switches SW0 and SW1; drag it to **Table Pin Assignment View** to select the ports for CapSense buttons

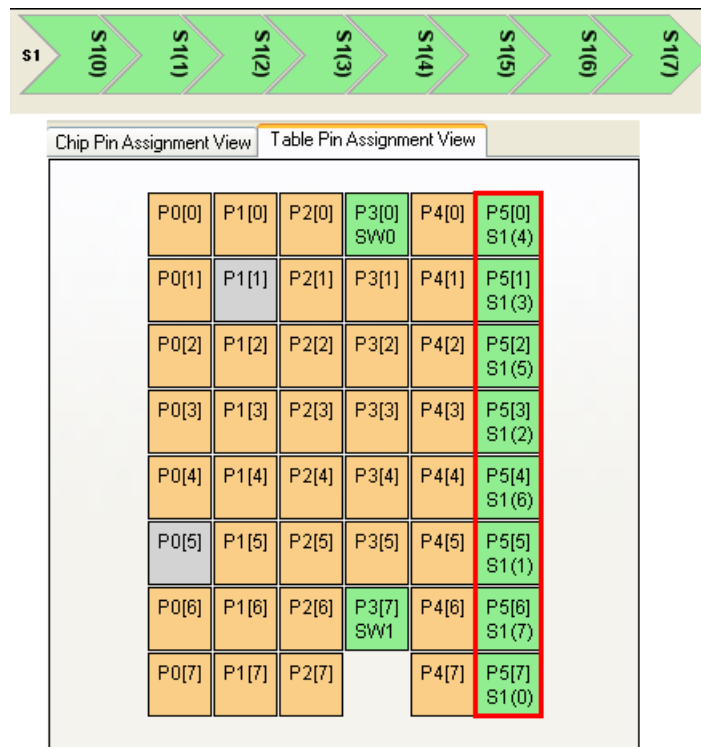
Figure 5-15. Assign CapSense Switch to Port



22. Configure all CapSense sliders to ports, as shown in the following figure.

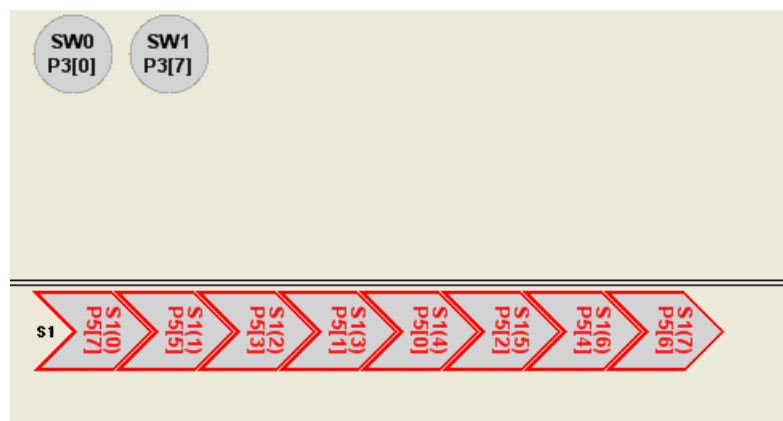


Figure 5-16. Assign CapSense Slider to Port



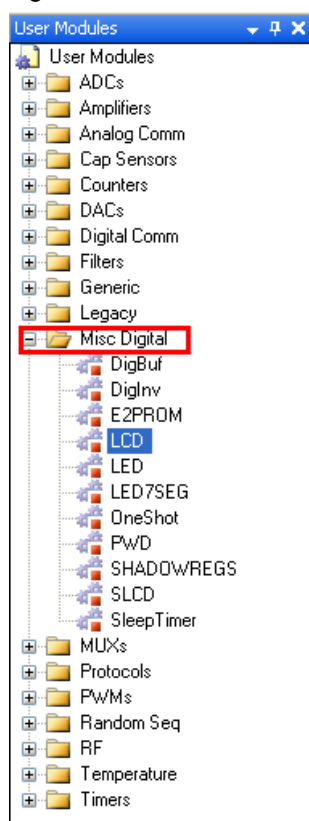
23. The CapSense buttons and sliders appear as follows.

Figure 5-17. Assigned CapSense Buttons and Slider View



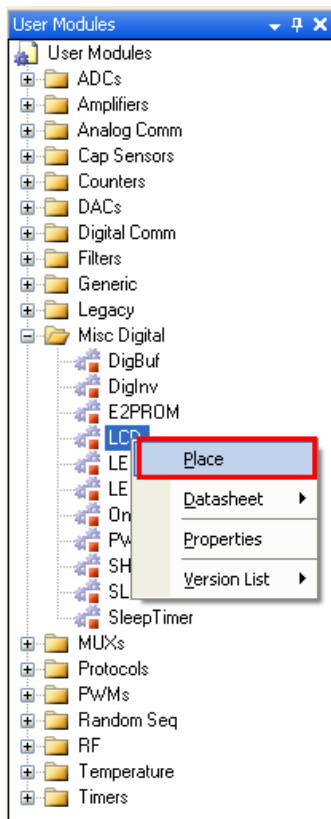
24. In the User Module Window, select **Misc Digital Folder** to select the LCD module.

Figure 5-18. User Module Window



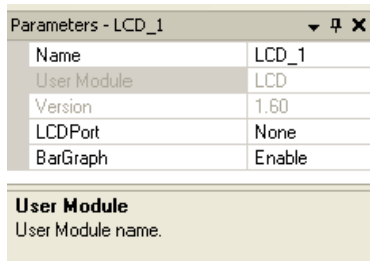
25. Place the module by right-clicking on the **LCD** and select **Place**.

Figure 5-19. Select LCD Module



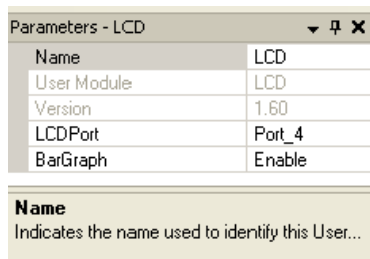
26. Default LCD\_1 parameters window are shown in the following figure.

Figure 5-20. Default LCD\_1 Parameter Window



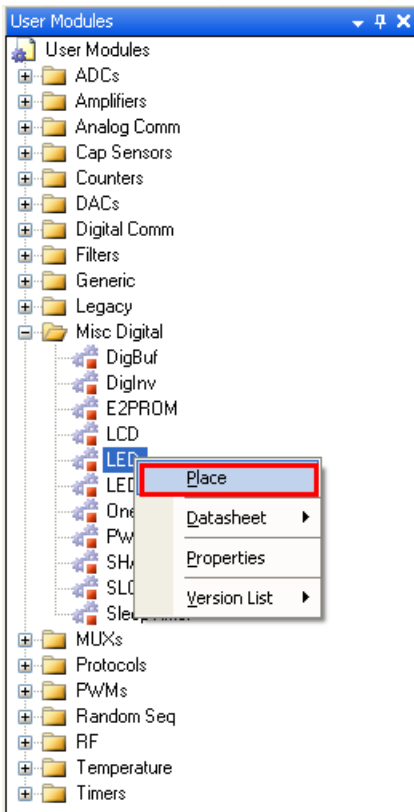
27. Rename LCD\_1 as **LCD** and configure the parameters to build the project successfully.

Figure 5-21. LCD Parameter Window



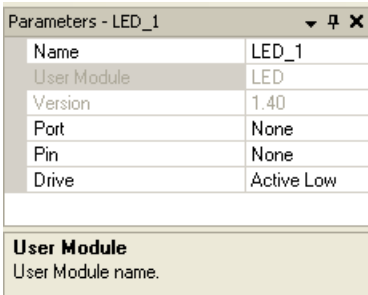
28. In the User Module window, go to Misc Digital folder and select **LED**.

Figure 5-22. Select LED Module



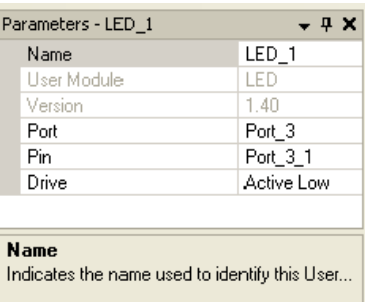
29. Default LED\_1 parameter window appears, as shown in the following figure.

Figure 5-23. Default LED\_1 Parameter Window



30. Configure LED\_1 parameter to successfully build the project.

Figure 5-24. LED\_1 Parameter Window



31. Similarly add another LED module and rename it to build the project.

Figure 5-25. LED\_2 Parameter Window

Parameters - LED_2	
Name	LED_2
User Module	LED
Version	1.40
Port	None
Pin	None
Drive	Active Low
<b>User Module</b> User Module name.	

Parameters - LED_2	
Name	LED_2
User Module	LED
Version	1.40
Port	Port_3
Pin	Port_3_2
Drive	Active Low
<b>Name</b> Indicates the name used to identify this User...	

32. Finally, LCD and LED modules are placed in Chip view window.

Figure 5-26. LED and LCD Modules



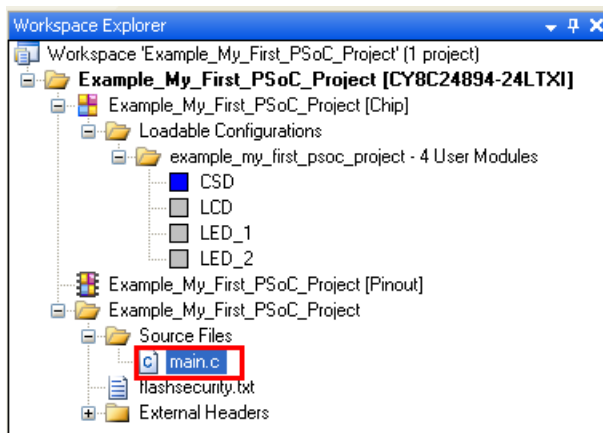
Figure 5-27. Global Resources Window

Global Resources - example_my_first_psoc_project	
Power Setting [ Vcc / SysClk freq ]	5.0V / 24MHz
CPU_Clock	SysClk/8
Sleep_Timer	512_Hz
VC1= SysClk/N	16
VC2= VC1/N	16
VC3 Source	VC2
VC3 Divider	256
SysClk Source	Internal
SysClk*2 Disable	No
Analog Power	SC On/Ref Low
Ref Mux	[Vdd/2]+/-BandGap
AGndBypass	Disable
Op-Amp Bias	Low
A_Buff_Power	Low
Trip Voltage [LVD]	4.81V
LVD ThrottleBack	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 cl...

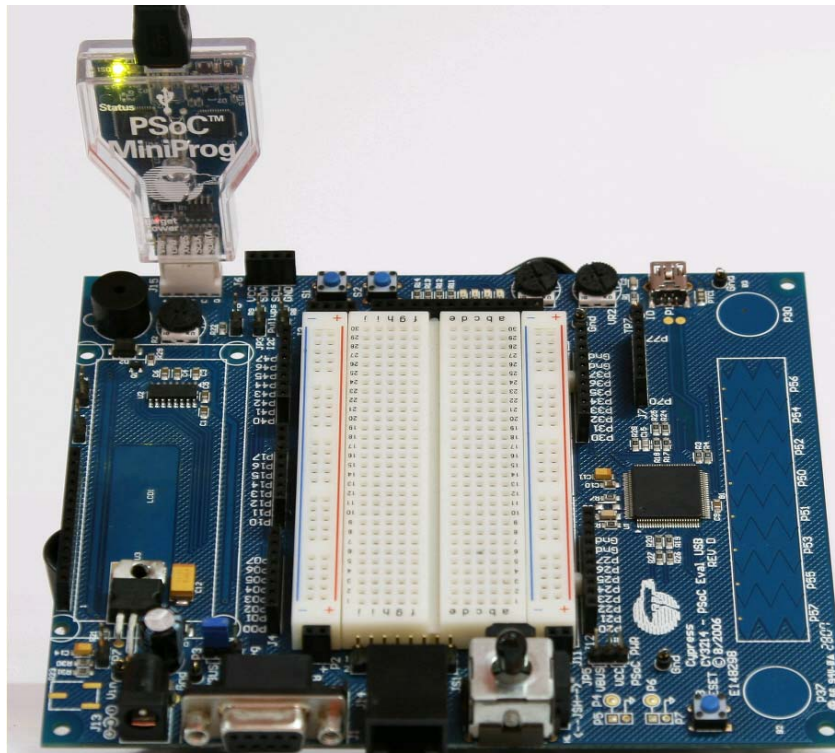
34. Open the existing *main.c* file in Workspace Explorer. Replace the existing *main.c* with the contents of *CapSense.c* file, which is attached to this document.

Figure 5-28. Workspace Explorer



35. Save the project.
36. Click **Build > Generate/Build 'Example\_My\_First\_PSoC\_Project'**.
37. When the project is built successfully, connect the CY3214-PSoCEVALUSB kit to PC through a MiniProg1 Connector (J15) on the board.

Figure 5-29. Connect MiniProg1 to Board



38. The board can be programmed either through PSoC Designer IDE itself or by launching PSoC Programmer.

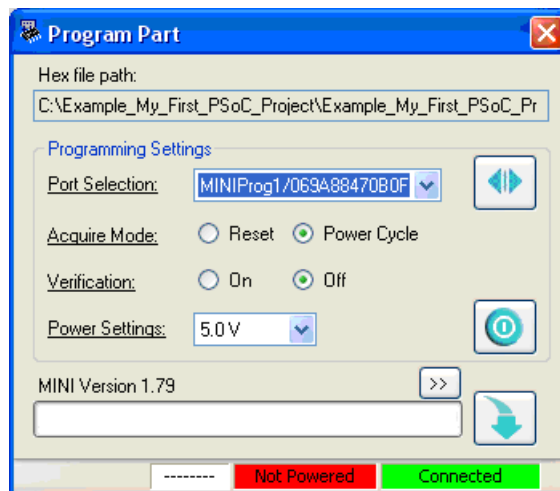
39. To program the board using PSoC Programmer, see [PSoC Programmer on page 14](#).

40. To program the board through PSoC Designer, follow these steps.

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

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

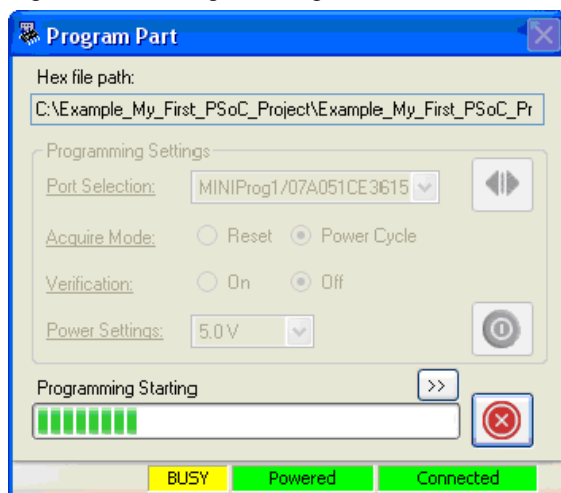
Figure 5-30. Program Part Window



- b. In the Program Part window, configure the following settings:

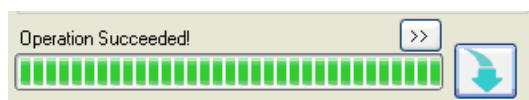
- Port Selection: select **MiniProg1/<MiniProg Number>** with the status **Connected**
  - Acquire Mode: **Power Cycle**
  - Verification: **Off**
  - Power Settings: **5.0 V**
- c. Click on the **Program** button to start programming the board.
  - d. The programming starts and the status is displayed on the progress bar.

Figure 5-31. Programming Status



- e. When the programming is done successfully, the 'Operation Succeeded!' Message is shown.

Figure 5-32. 'Operation Succeeded!' Message



#### 5.1.4 Verify Output

1. Make the following connections on the board using the single strand jumper wires provided.
  - a. Connect 2-K $\Omega$  resistor between Port P0.5 (P05 on J4) and P1.5 (P15 on J9).
  - b. Connect LED1 to Port P3.1 (P31 on J10) and LED2 to P3.2(P32 on J10).



Figure 5-33. USB Connection to Board

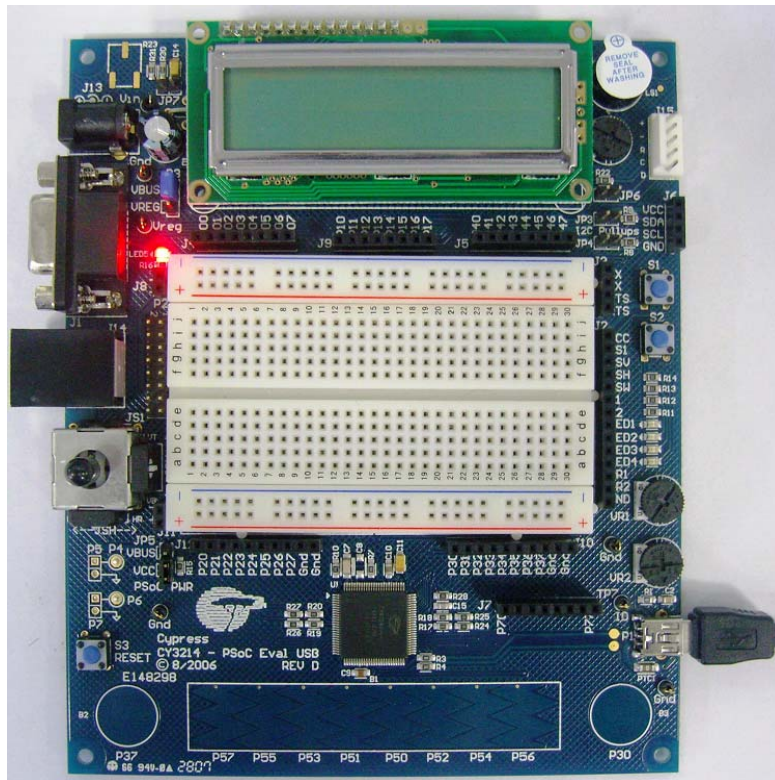
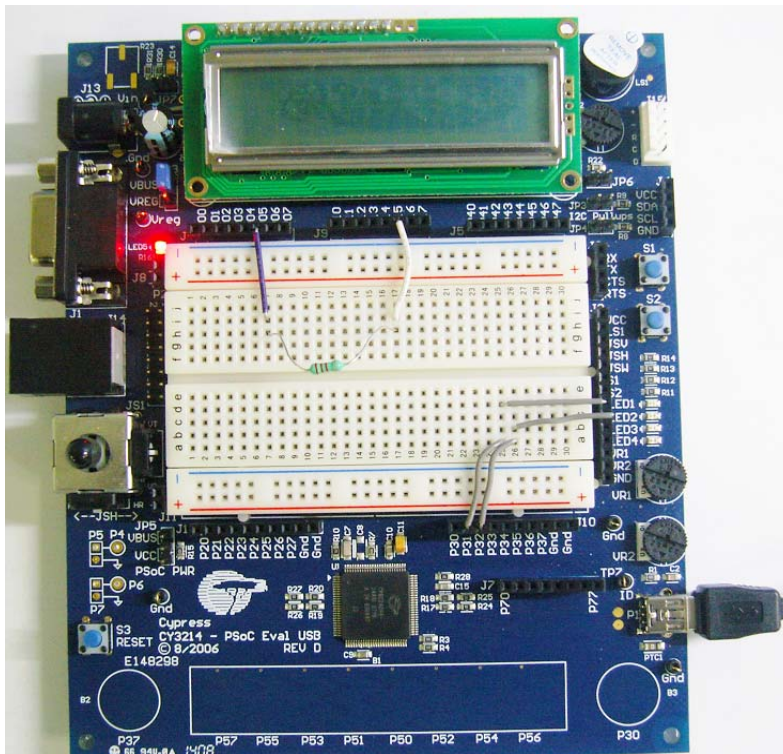
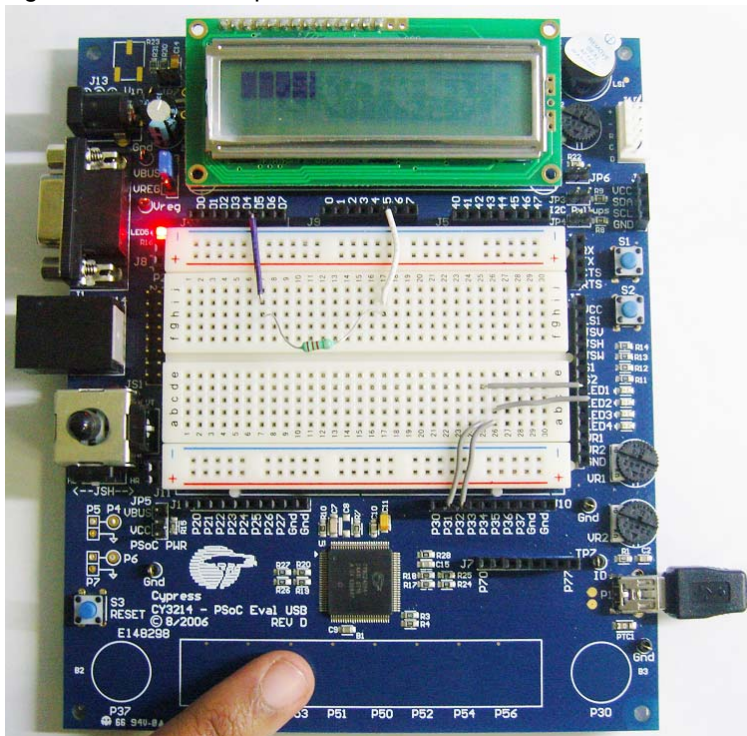


Figure 5-34. Port Selection and Resistor Connection



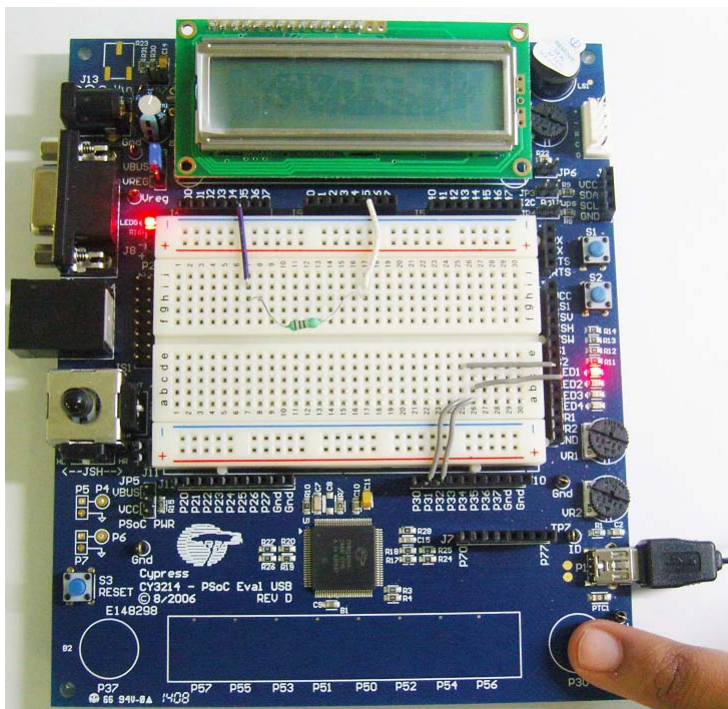
2. When the CapSense slider is touched, the horizontal bar-graph is displayed on the LCD.

Figure 5-35. Bar-Graph on LCD



3. LED1 glows when CapSense button P30 is touched.

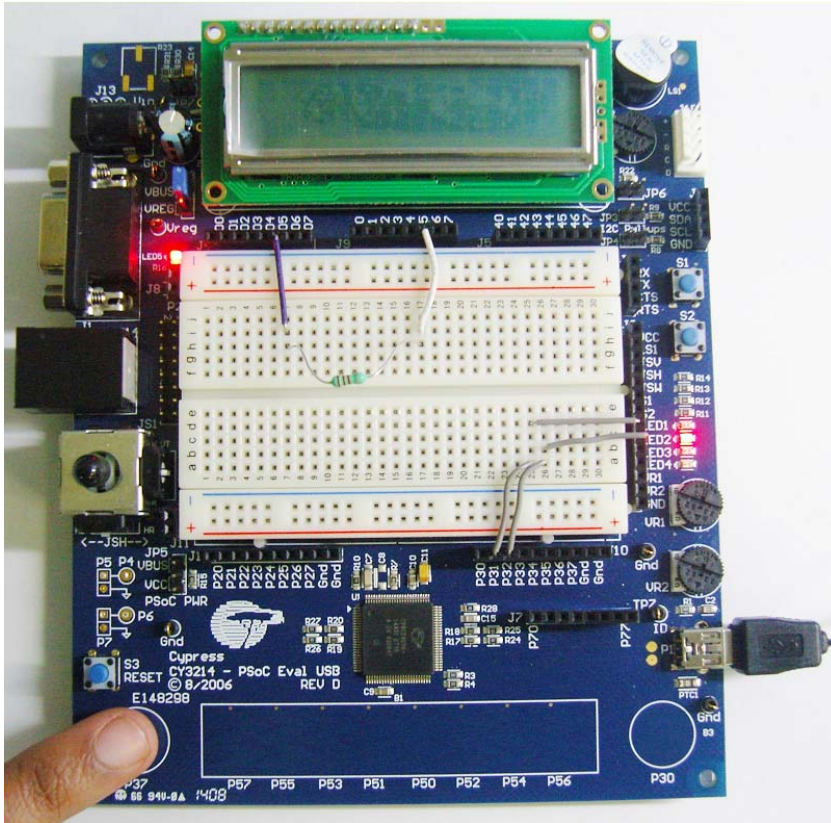
Figure 5-36. LED1 Glows for Button P30





4. LED2 glows when CapSense button P37 is touched.

Figure 5-37. LED2 Glows for Button P37



5. Save and close the project.

## 5.2 Buttons and Lights Example

### 5.2.1 Project Description

This code example demonstrates the USB function using the CY3214-PSocEVALUSB Kit. In this project, the Buttons and Lights PC Host GUI is used to control the LEDs on board. The GUI also has four soft buttons and four LEDs. When there is a button press event on the board, the board sends out the buttons report to the GUI. The GUI updates the local LEDs and sends out a Lights report back to the board. The LEDs on the board are updated based on this Lights report. The GUI also provides the LED output as a logical function of soft buttons and buttons on board. The LED output is a function of 'AND' or 'OR' of soft buttons and real buttons. By default, the LED output is that of real buttons. The following module is used in the project:

- **USB:** This module handles requests from the USB host (PC) to communicate with the device. It handles data packet transfer between the host and device connected to host through USB.

### 5.2.2 Hardware Connections

- Connect LEDs 1-4 to P3.2 to P3.5 (P32 to P35 on connector J10).
- Port pins P2.2 to P2.5 (P22 to P25 on connector J12) are configured as active-low buttons in firmware. To simulate a button press connect the required port pin to ground (terminal marked '-') on connector J11.

Figure 5-38. Hardware Connection

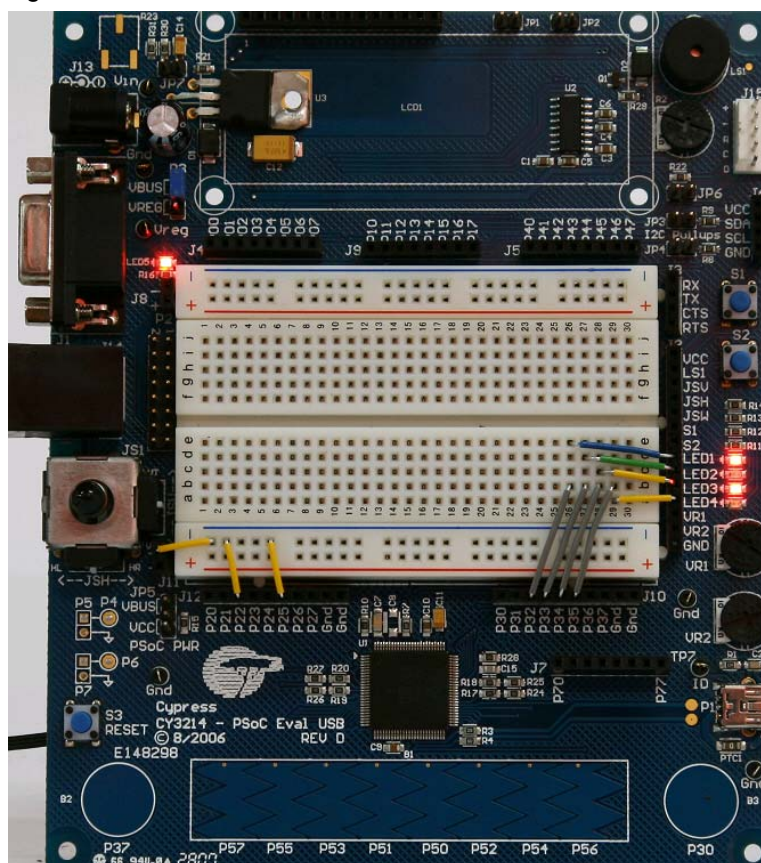
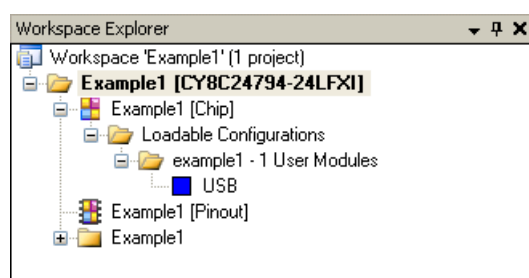


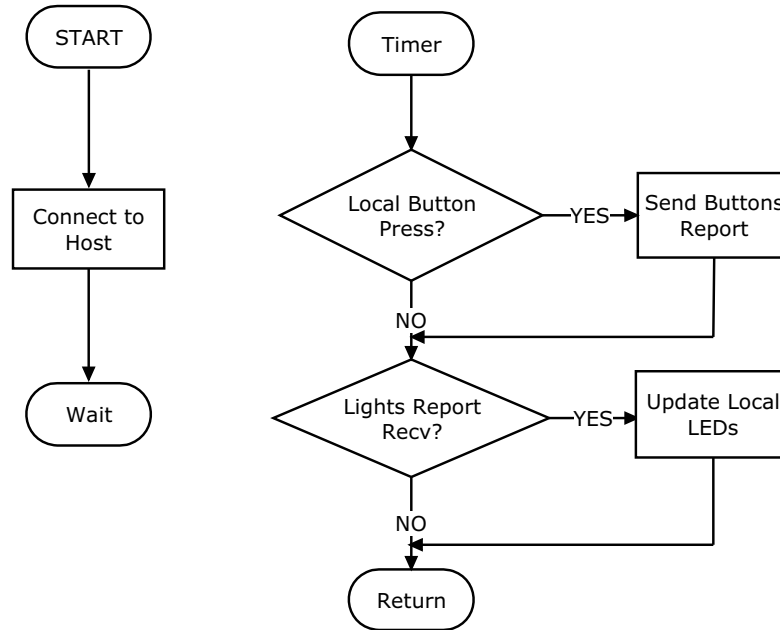
Figure 5-39. Workspace Explorer of Buttons and Lights



## 5.2.3 Firmware Architecture

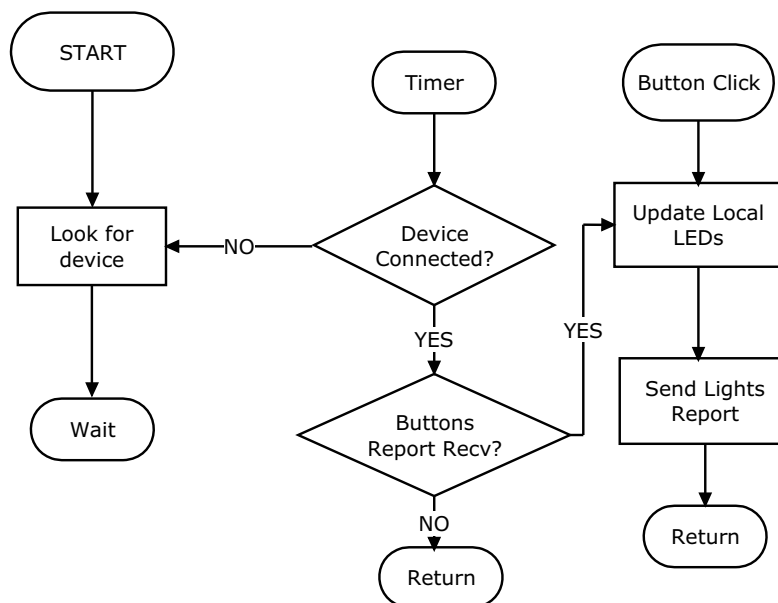
### 5.2.3.1 Device Program Firmware Architecture

Figure 5-40. Flowchart



### 5.2.3.2 Host Program Firmware Architecture

Figure 5-41. Flowchart



## 5.2.4 Verify Output

1. Load the Example1.hex from <Install\_Directory>:\Cypress\CY3214-PSoCEVAL-USB\<version>\Firmware\ButtonsAndLights\PSoC Project\Example1, as explained in [Programming Specifications and Connections on page 16](#).
2. On successful programming, disconnect the MiniProg1 and connect the board to PC.
3. Open the Buttons and Lights GUI (Example1.exe) from <Install\_Directory>:\Cypress\CY3214-PSoCEVALUSB\<version>\Firmware\ButtonsAndLights\PC Software\Example1.
4. Press any of the four soft buttons on the GUI; the corresponding LEDs in the GUI glow (green) and similar LEDs glow in the kit (red).

Figure 5-42. (a) GUI Display (b) Kit Display

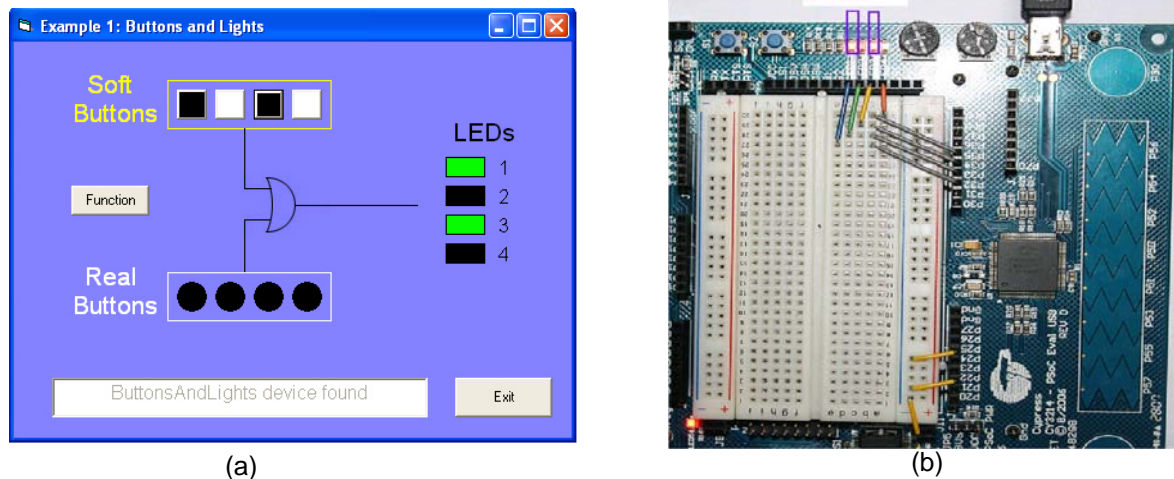
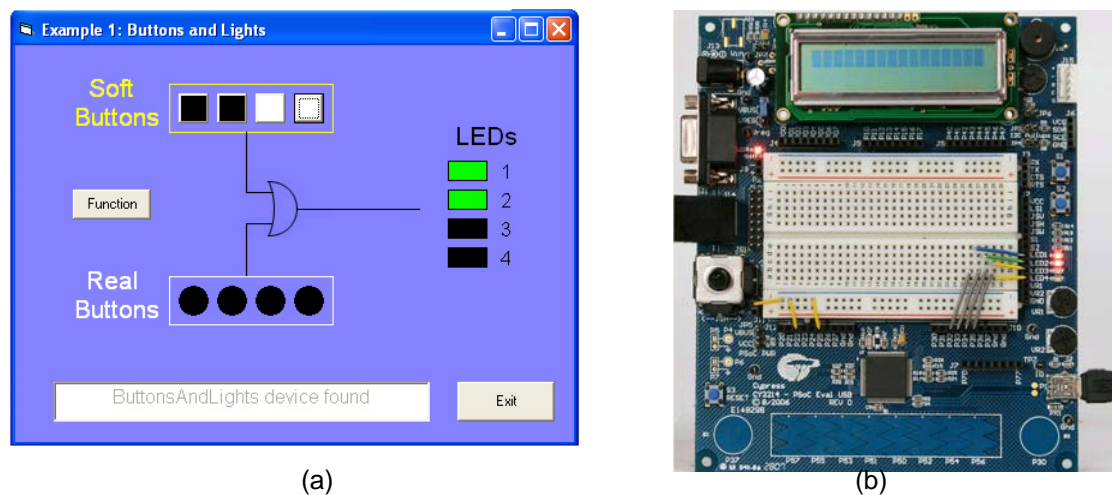


Figure 5-43. (a) GUI Display (b) Kit Display





## 5.3 Temperature Sensing Keyboard

### 5.3.1 Project Description

This project example operates similar to a keyboard, although it does not look like a standard keyboard. The example is a temperature sensor using potentiometer (VR1 and VR2) as surrogates for real temperature sensors. When plugged in using USB, the project enumerates as a keyboard and starts printing out temperature values.

The following modules are used in the project:

- **ADCIN:** It is a differential input ADC from where the samples are received and used as the reference values to calculate the real temperature.
- **LCD:** It is used to display the values of the real temperatures.
- **PGA:** It is used as an amplifier where it amplifies the signal received from ADCINC.
- **USB:** This module is used to handle the request from the USB host (PC) to communicate with the device. It handles all the data packet transfer between the host and device connected to host through the USB.

### 5.3.2 Hardware Connection

1. Connect the jumper wire between S1 on J2 and P3.0 (P30 on J10).
2. Connect the jumper wire between S2 on J2 and P3.1 (P31 on J10).
3. Connect the jumper wire between VR1 on J2 and P3.6 (P36 on J10).
4. Connect the jumper wire between VR2 on J2 and P3.7 (P37 on J10).
5. Connect the jumper wire between P2.7 (P27 on J12) and LS1 on J2.

Figure 5-44. Hardware Connection

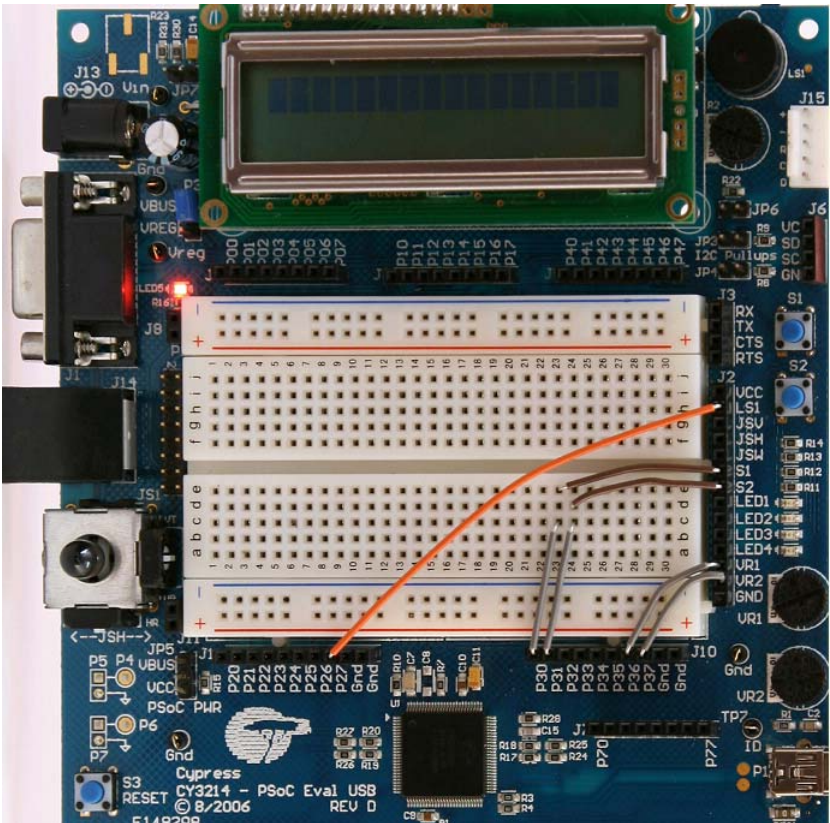
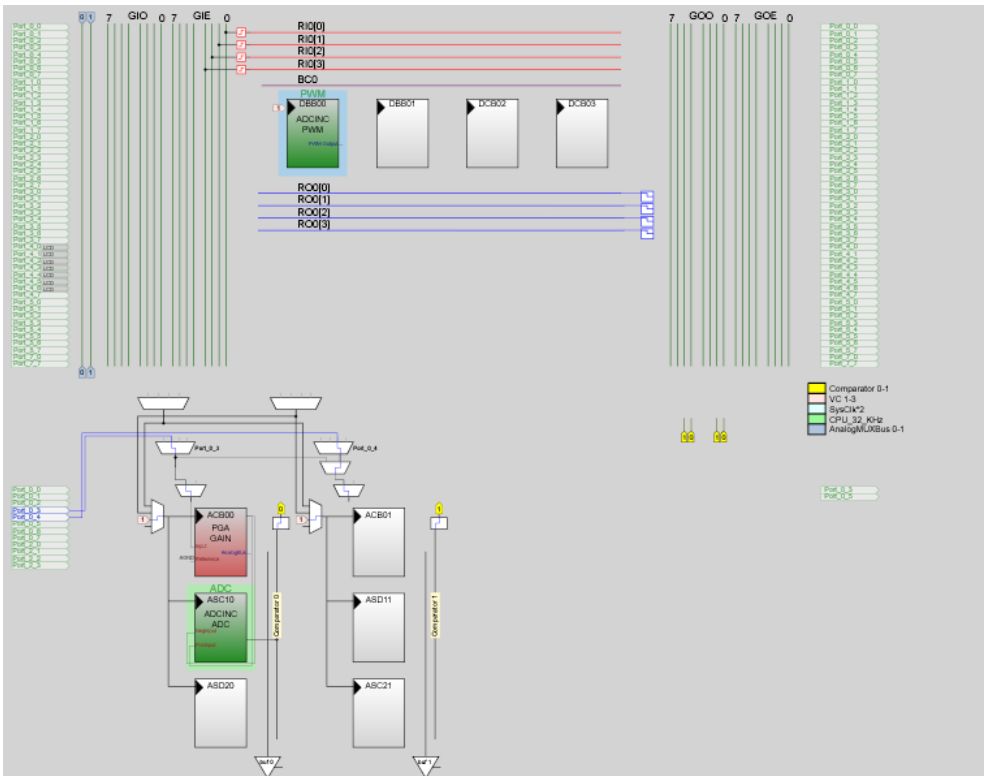


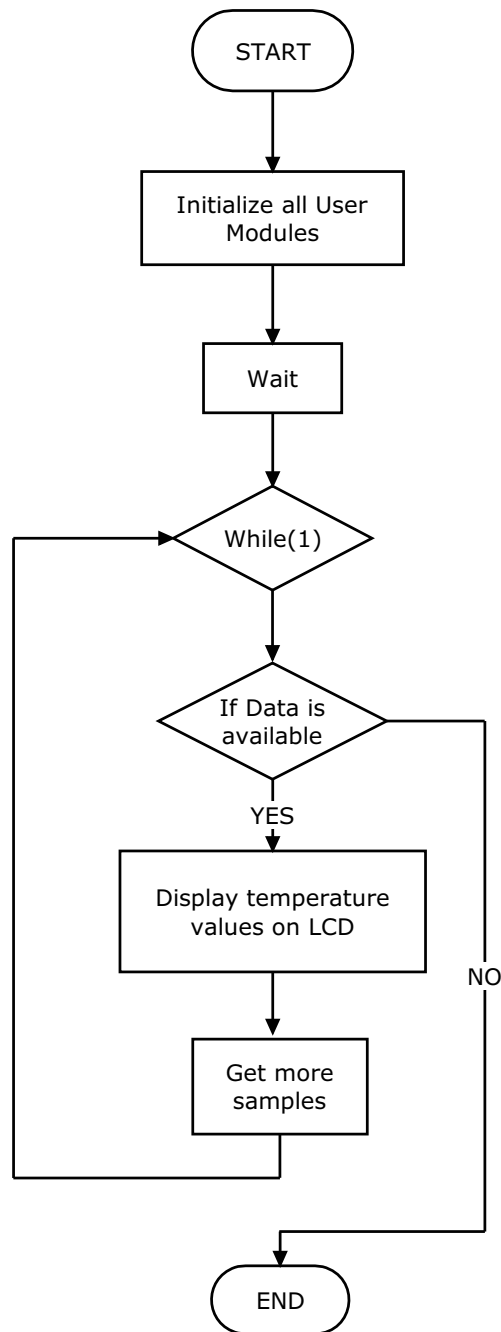
Figure 5-45. Device Configuration of Temperature Sensing Keyboard





### 5.3.3 Firmware Architecture

Figure 5-46. Flowchart



### 5.3.4 Verify Output

1. Load the Example2.hex from <Install\_Directory>:\Cypress\CY3214-PSoCEVAL-USB\<version>\Firmware\Temperature Sensing Keyboard\PSoC Project\Example2\, as explained in [Programming Specifications and Connections on page 16](#).
2. Open either a Microsoft Excel 2003 or 2007 application or a Windows Notepad application on the computer and plug in the board using USB.
3. Press **S1** or **S2** to see the temperature values directly in the editor or spreadsheet; the corresponding temperature values are displayed on the LCD.
4. Press the **CAPS\_LOCK** key on the keyboard to toggle the buzzer on and off. Vary the potentiometer (VR1 and VR2) and press **S1** and **S2** again.

Figure 5-47. Output

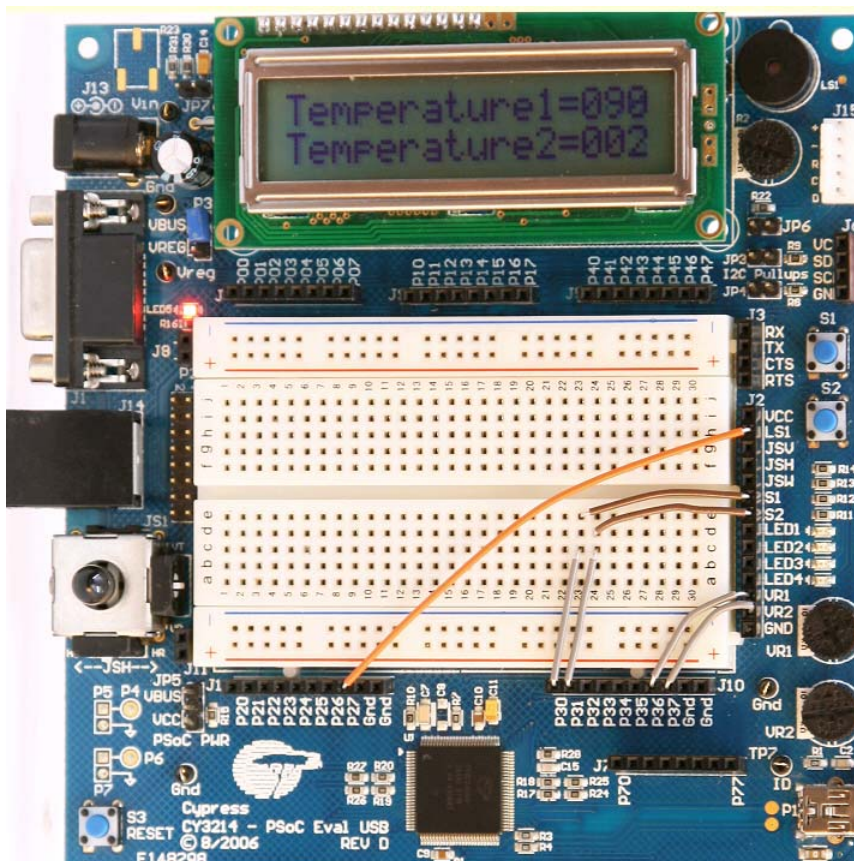
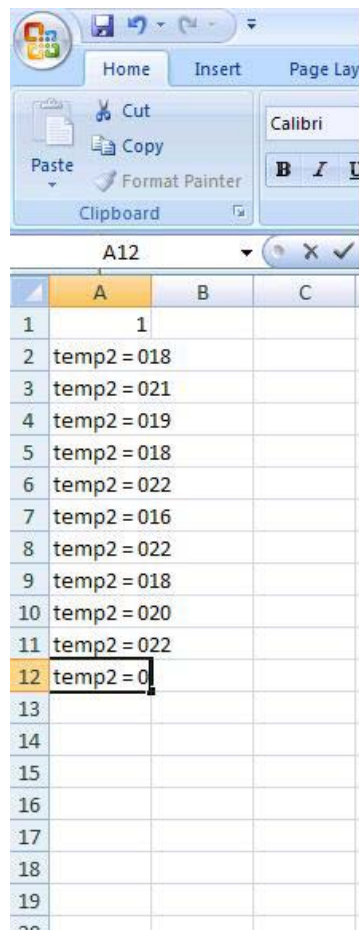


Figure 5-48. Temperature Sensor Output on Excel spreadsheet



## 5.4 Process Monitoring

### 5.4.1 Project Description

The process monitoring system code demonstrates how to collect the analog data at a rate determined by the host. The host then displays the data in a graph and is able to save and reload the data sets.

The following modules are used in the project

- **ADCIN:** It is a differential input ADC from where the samples are received and used as the reference values to calculate the real temperature.
- **LCD:** It is used to display the values of the real temperatures.
- **PGA:** It is used as an amplifier where it amplifies the signal received from ADCINC.
- **USB:** This module is used to handle the request from the USB host (PC) to communicate with the device. It handles all the data packet transfer between the host and device connected to the host through the USB.

## 5.4.2 Hardware Connection

Connect the jumper wire between P0.3 (P03 on J4) and VR1 on J2.

Figure 5-49. Hardware Connection

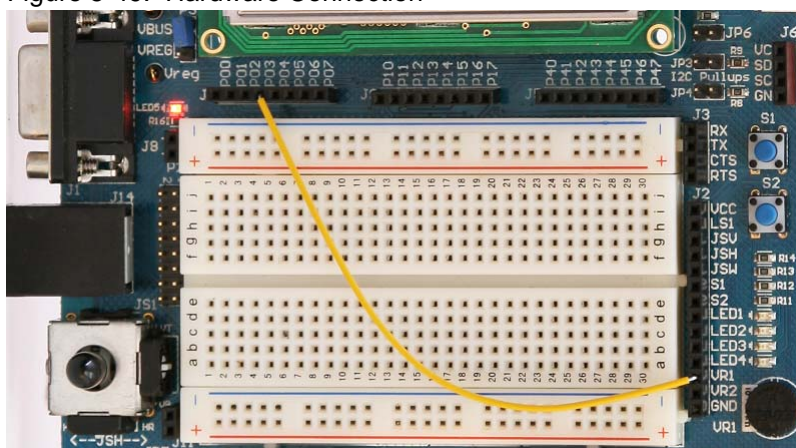
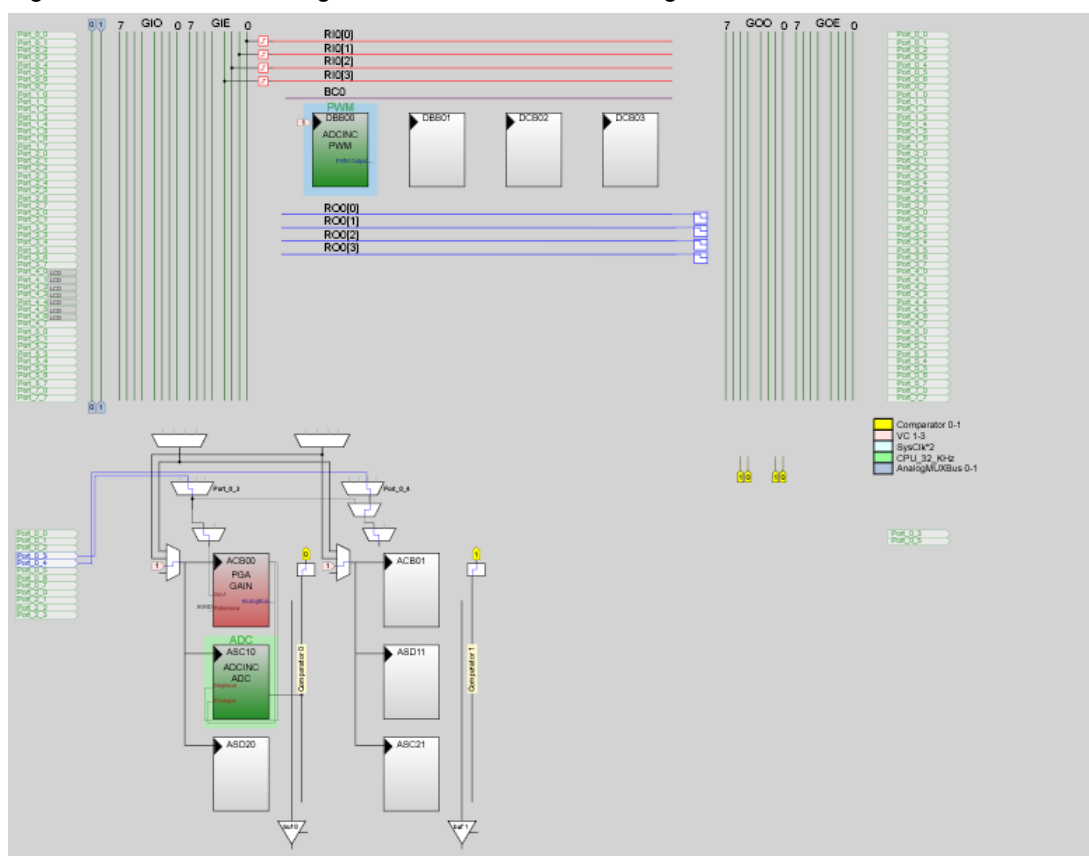
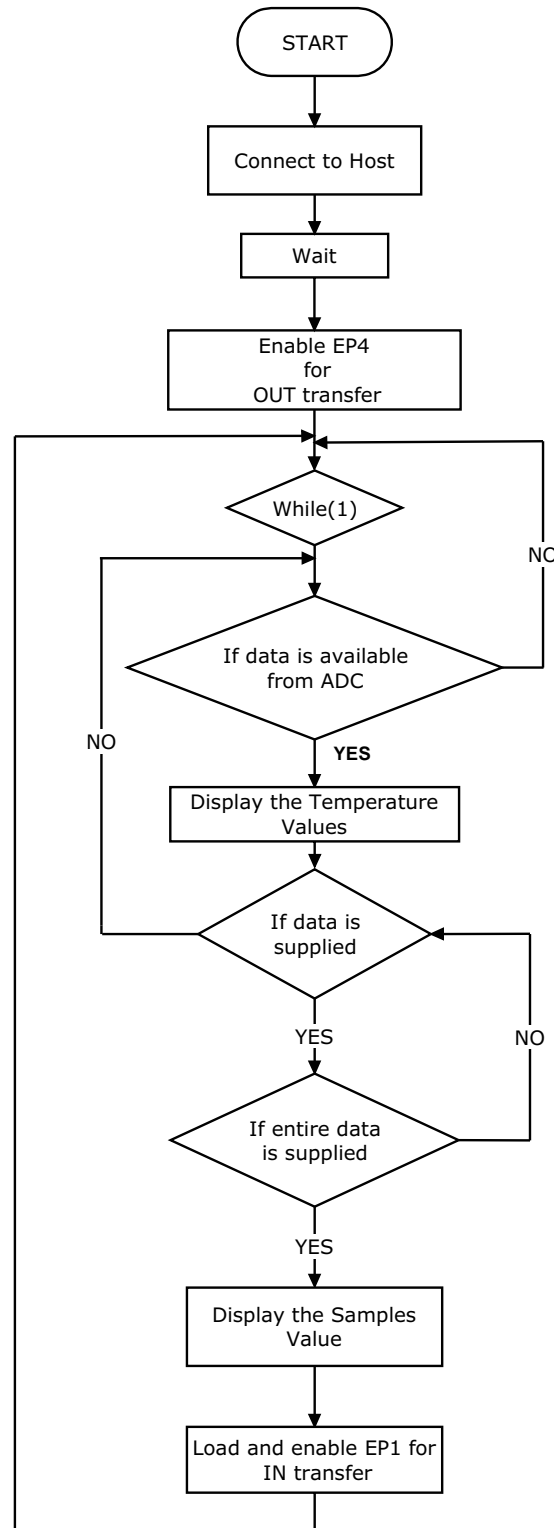


Figure 5-50. Device Configuration of Process Monitoring



### 5.4.3 Firmware Architecture

Figure 5-51. Flowchart





#### 5.4.4 Verify Output

1. Program the board with Example3.hex from <Install\_Directory>:\Cypress\CY3214-PSoCEVALUSB\<version>\Firmware\Process Monitoring\PSoC Project\Example3\, as explained in [Programming Specifications and Connections on page 16](#).
2. On successful programming, disconnect the MiniProg1 and connect the board to the PC.
3. Wait until the HID device is installed (first time only) and run *Example3.exe*.  
The LCD display shows the temperature value but indicates it is waiting for input from Host (PC).
4. Open the Process Monitoring GUI (Example3.exe) from <Install\_Directory>:\Cypress\CY3214-PSoCEVALUSB\<version>\Firmware\Process Monitoring\PC Software\Example3.
5. On the GUI, click the **GetSamples** button. The PC waits until 256 samples are received from the board and then displays a graph of Temperature Vs Time.
6. Vary the VR1 on the board and click the **GetSamples** button again to see the difference in the graph. This difference can be best viewed when Continuous Sampling is selected.
7. If continuous sampling is required, click the **Continuous Samples** button. The PC displays the graph after every 256 samples received.

**Note** If the Sample Rate slider is kept at a minimum, the time taken to display graph also increases as the graph is displayed only after receiving 256 samples.

Figure 5-52. Output

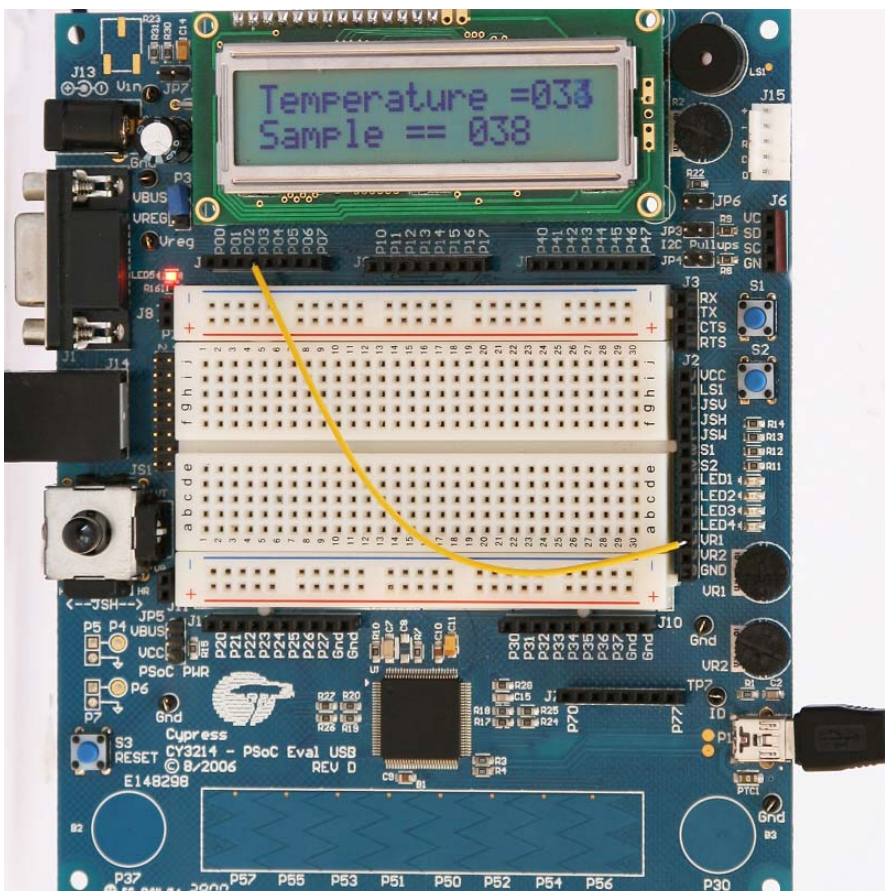
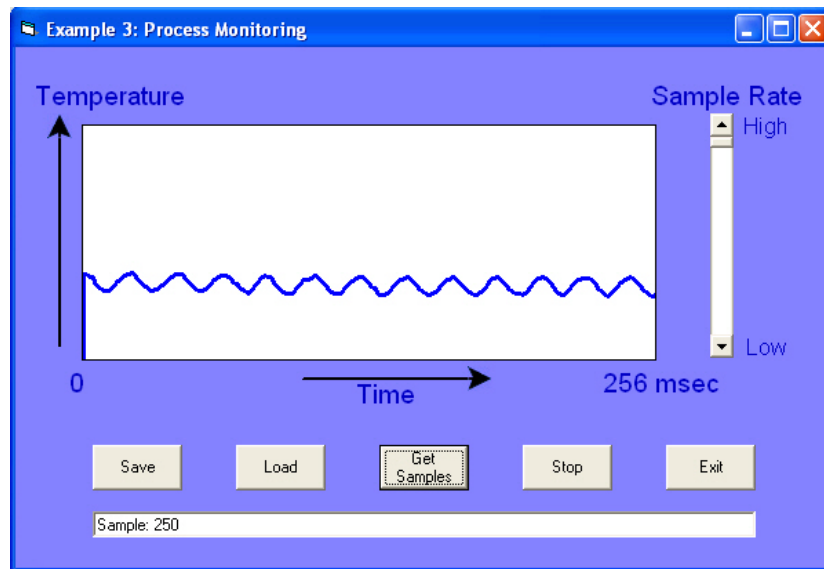


Figure 5-53. GUI Display



## 5.5 Async\_PingExample

### 5.5.1 Project Description

The project demonstrates asynchronous data transfer by sending a series of large USB write (OUT) and read (IN) requests by using USB functionality, which is available with CY8C24894 device.

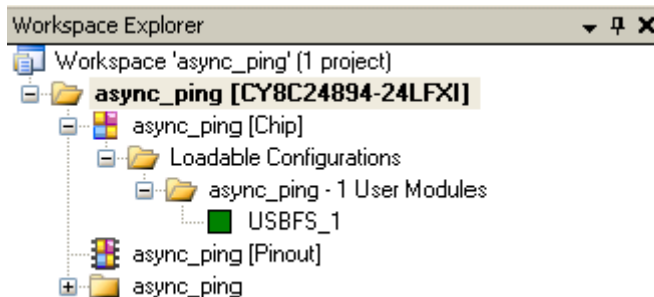
The following user module is used in the project:

- **USBFS\_1:** This module is used to handle the request from the USB host (PC) to communicate with the device. It handles all the data packet transfer between the host and device connected to host through the USB.

### 5.5.2 Hardware Connections

There are no hardware connections required.

Figure 5-54. Device Configuration of Async\_PingExample



### 5.5.3 Driver Installation

The driver installation steps are for Windows XP and not Windows 7. Program the CY3214-PSoCEVALUSB board with the *Async\_ping.hex*, located at:

<Install\_Directory>:\Cypress\CY3214-PSoCEVALUSB\<version>\Firmware\Async\_PingExample\PSoC Project\async\_ping and connect the board to the PC using a USB cable. When connecting the board for the first time, follow these steps to install the USB driver for this project.

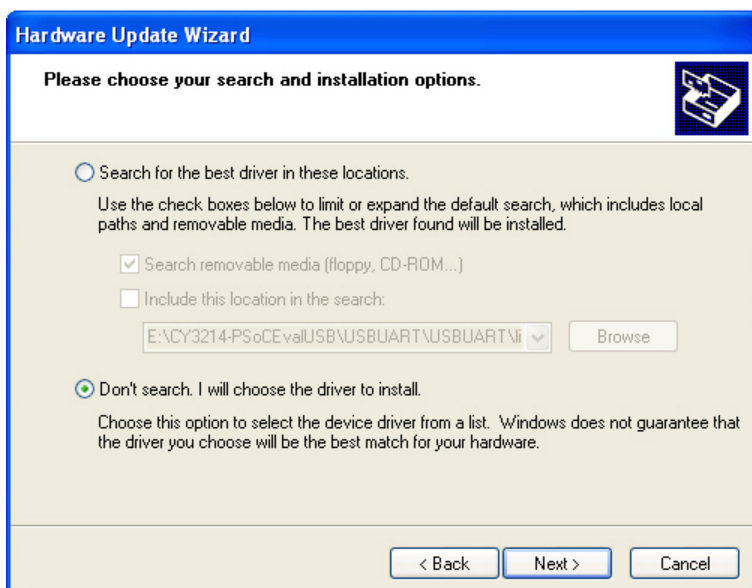
1. Select **Install from a list of specific location (Advanced)** and click the **Next** button.

Figure 5-55. Driver Installation Welcome Page



2. Select **Don't search. I will choose the driver to install** and click **Next** to continue.

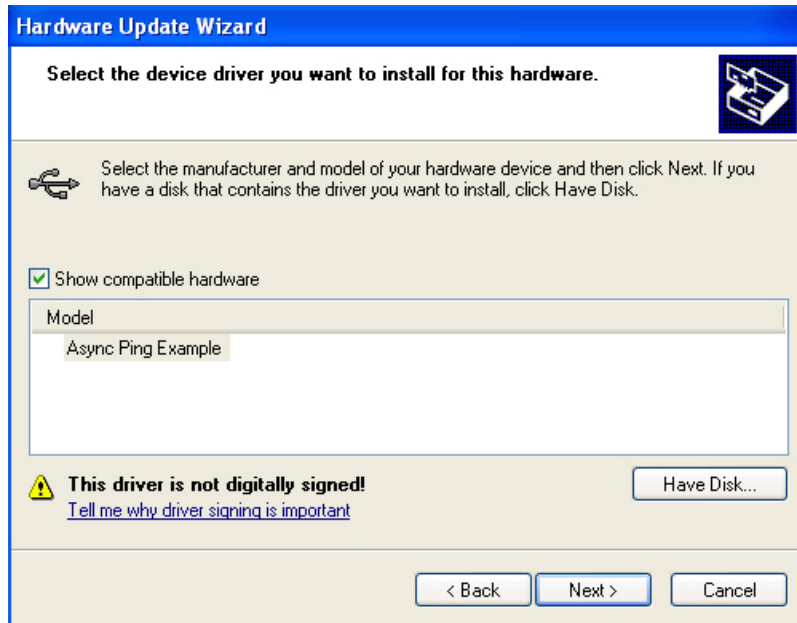
Figure 5-56. Installation Options Page





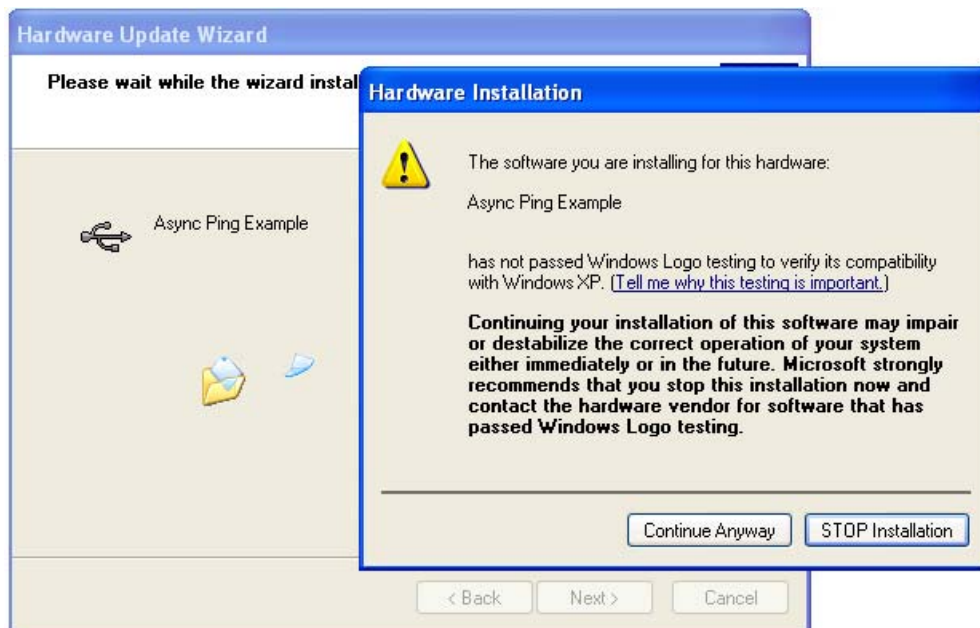
3. Select **Have Disk** to select the appropriate **INF** file. Find the Async\_PingExample drivers setup in the following path: <Install\_Directory>:\Cypress\CY3214-PSoCEVALUSB\<version>\Firmware\Async\_PingExample\Driver\_Setup

Figure 5-57. Device Driver Path



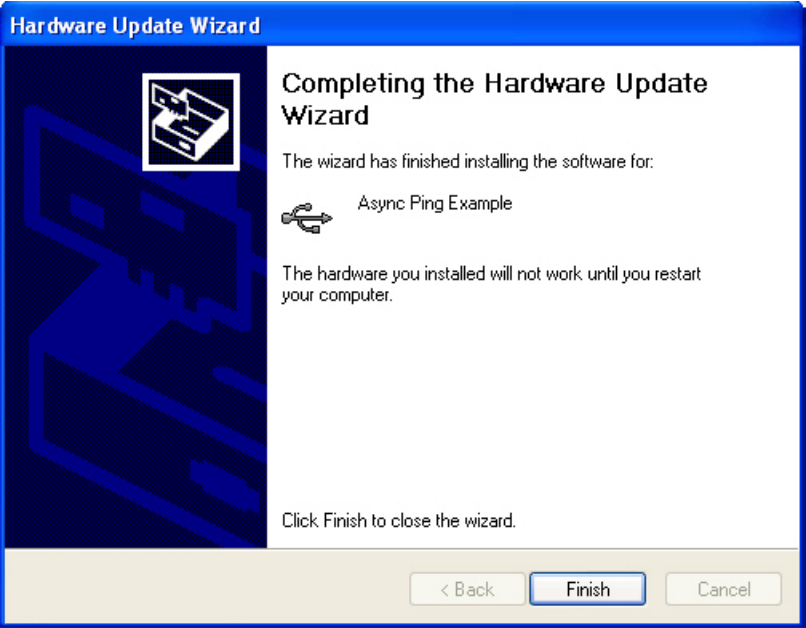
4. Click on **Continue Anyway**.

Figure 5-58. Installation Page



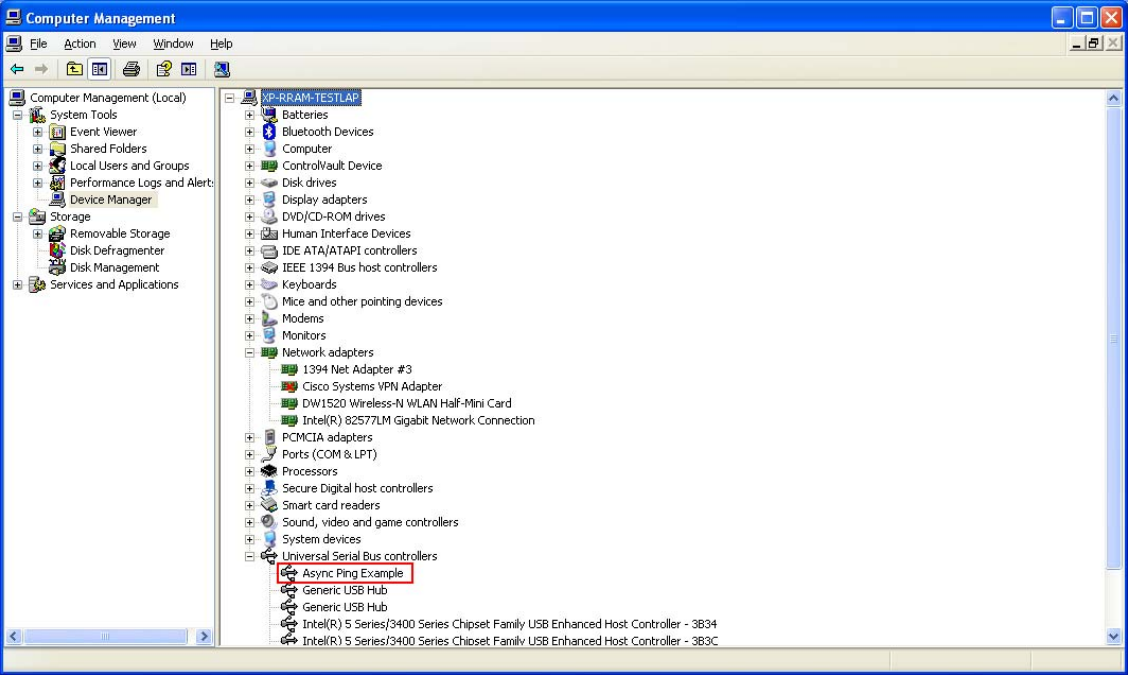
5. Click **Finish** to complete the driver installation.

Figure 5-59. Installation Complete Page



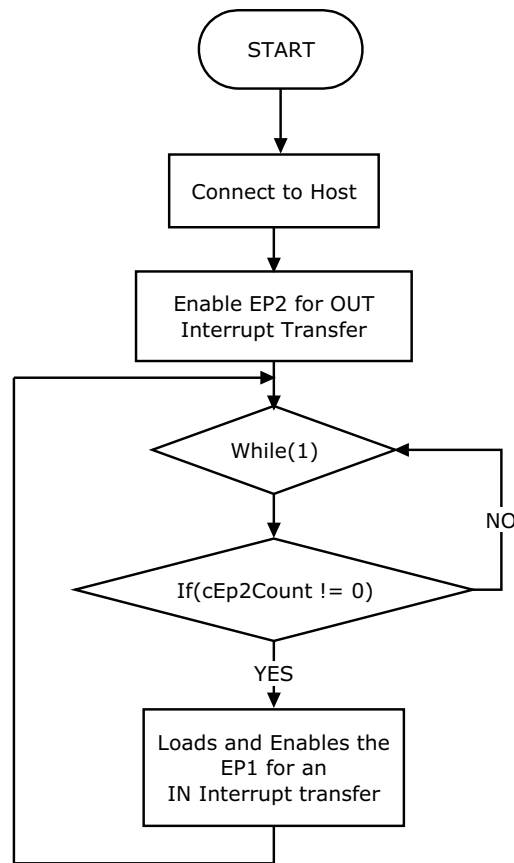
6. After successful driver installation, confirm in the following path: **Start > right-click My Computer > Manage > Device Manager > Universal Serial Bus Controller.**

Figure 5-60. Detect Async\_Ping Example in Universal Serial Bus Controller



## 5.5.4 Firmware Architecture

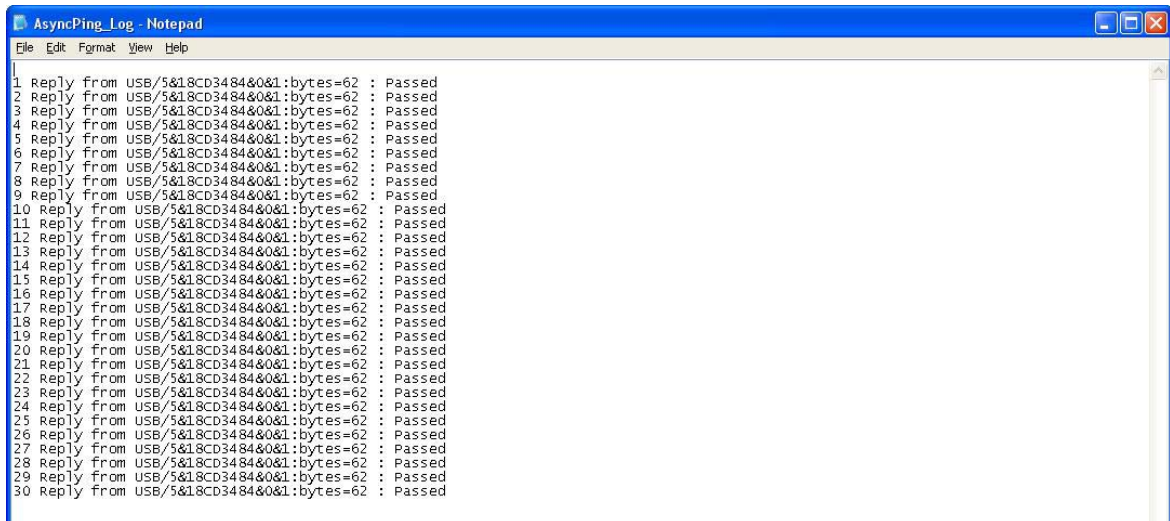
Figure 5-61. Flowchart



## 5.5.5 Verify Output

1. Program the board with *async\_ping.hex* from <Install\_Directory>:\Cypress\CY3214-PSoCEVALUSB\<version>\Firmware\Async\_PingExample\PSoC Project\async\_ping, as explained in [Programming Specifications and Connections on page 16](#).
2. On successful programming, disconnect the MiniProg1 and connect the board to the PC.
3. Run *Async\_Ping\_Host.exe* from <Install\_Directory>:\Cypress\CY3214-PSoCEVALUSB\<version>\Firmware\Async\_PingExample\PC\_Host\_App
4. A DOS-based host program starts and sends out the data transfer requests, reads and compares the received data and logs the data transfer results.
5. The log file *AsyncPing\_Log.txt* is saved in the application path. This log file gives a detailed description of each data transfer that occurred

Figure 5-62. Output



```

1 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
2 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
3 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
4 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
5 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
6 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
7 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
8 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
9 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
10 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
11 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
12 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
13 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
14 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
15 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
16 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
17 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
18 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
19 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
20 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
21 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
22 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
23 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
24 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
25 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
26 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
27 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
28 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
29 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed
30 Reply from USB/5&18CD3484&0&1:bytes=62 : Passed

```

## 5.6 Bulk\_PingExample

### 5.6.1 Project Description

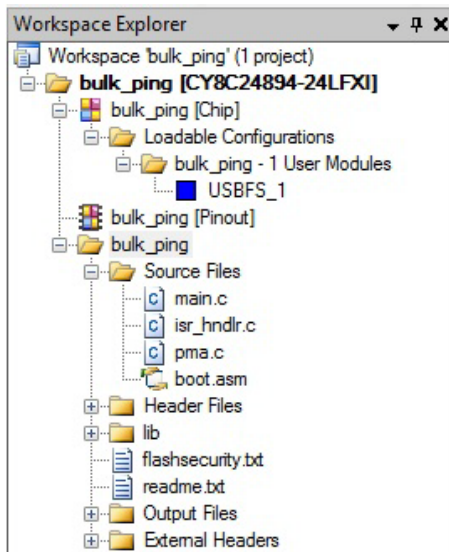
The project demonstrates bulk data transfer by sending a series of end point-sized USB write (out) and read (in) requests. The host effectively sends blocking writes and reads while the PSoC firmware reads the OUT packets and fills the IN data from it. The following user module is used in the project:

- **USBFS\_1**: This module is used to handle the request from the USB host (PC) to communicate with the device. It handles all the data packet transfer between the host and device connected to host through USB.

### 5.6.2 Hardware Connections

No hardware connections are required.

Figure 5-63. Device Configuration of Bulk\_PingExample



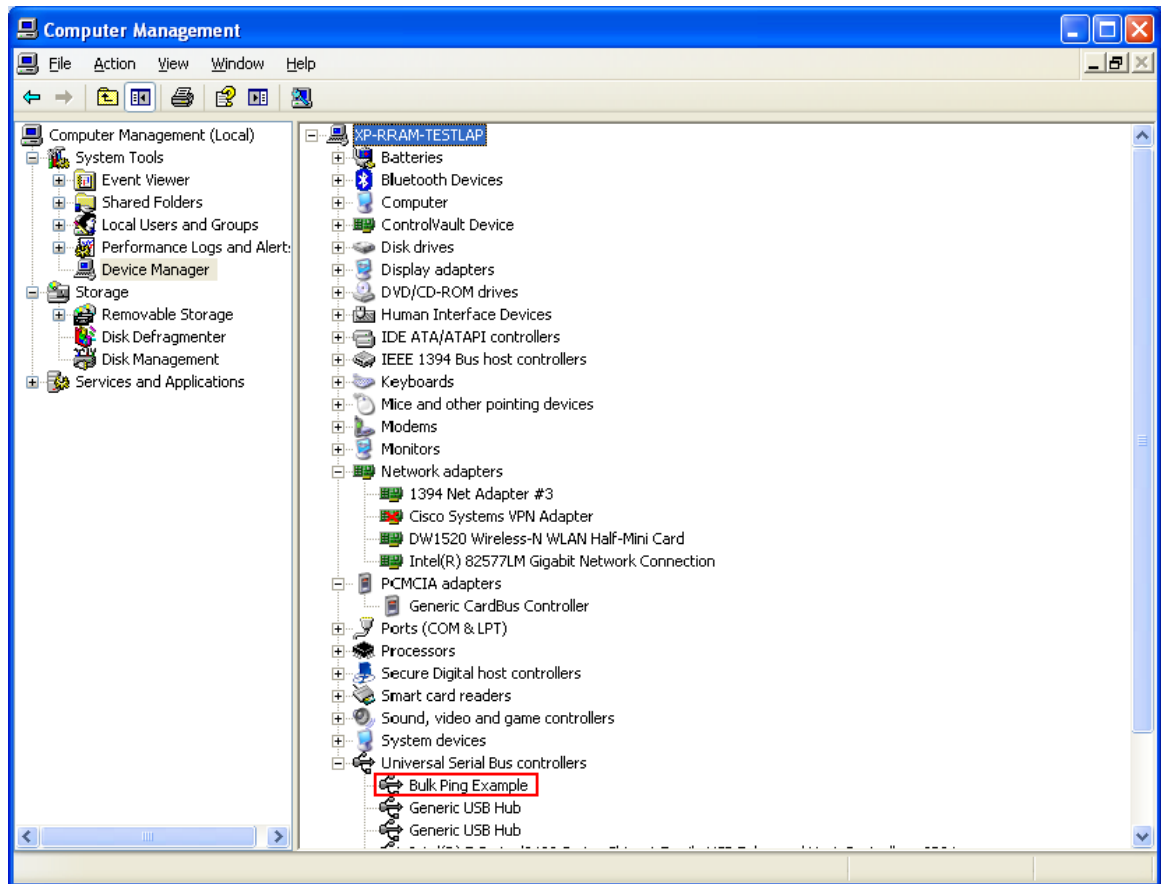
### 5.6.3 Driver Installation

Program the CY3214-PSoCEVALUSB board with the *Bulk\_ping.hex* located at `<Install_Directory>:\Cypress\CY3214-PSoCEVALUSB\<version>\Firmware\Bulk_PingExample\PSoC Project\bulk_ping` and connect the board to the PC using a USB cable. When connecting the board for the first time, follow steps 1 to 5 in the section [Driver Installation on page 64](#).

At step 3, browse and locate *BulkPing.inf* from `<Install_Directory>:\Cypress\CY3214-PSoCEVALUSB\<version>\Firmware\Bulk_PingExample\Driver_Setup`.

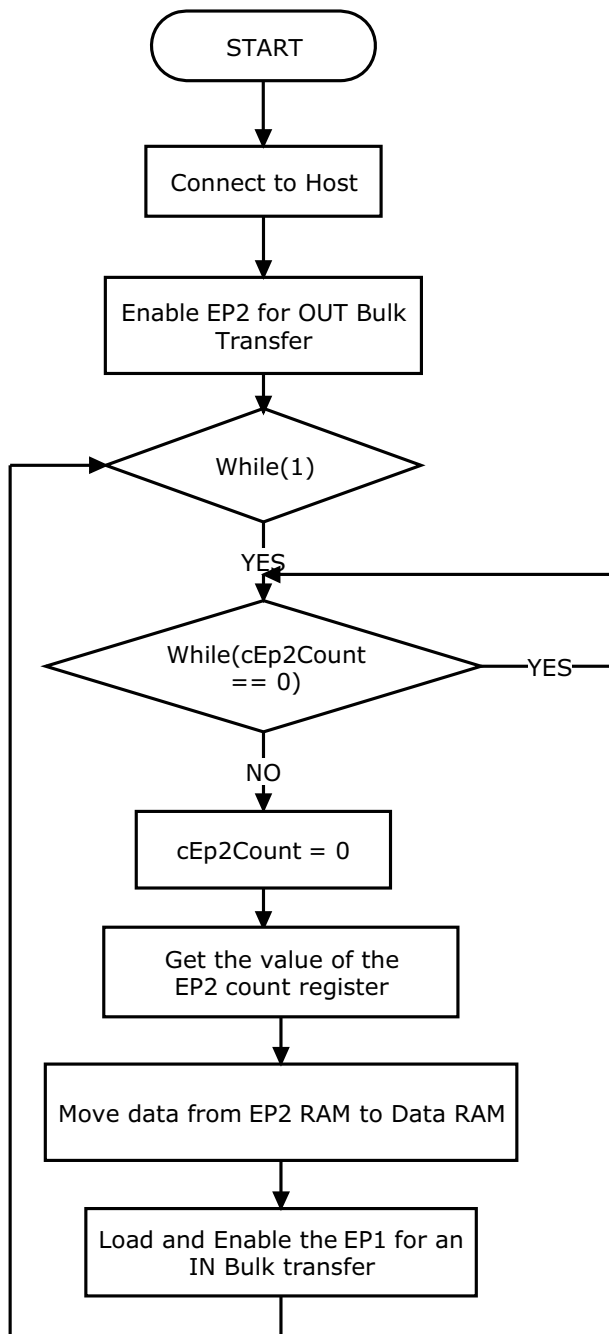
After successful driver installation, confirm in the following path: **Start > right-click My Computer > Manage > Device Manager > Universal Serial Bus Controller**.

Figure 5-64. Device Management



## 5.6.4 Firmware Architecture

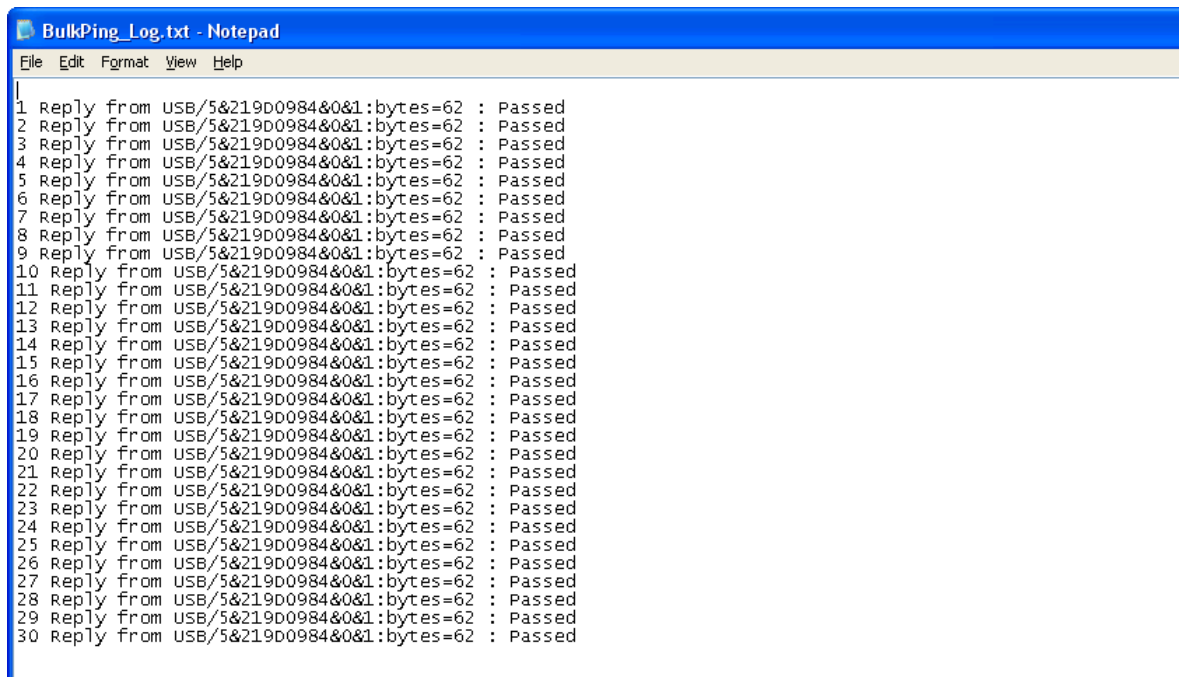
Figure 5-65. Flowchart



## 5.6.5 Verify Output

1. Program the board with *bulk\_ping.hex* from <Install\_Directory>:\Cypress\CY3214-PSoCEVALUSB\<version>\Firmware\Bulk\_PingExample\PSoC Project\bulk\_ping.
2. On successful programming, disconnect the MiniProg1 and connect the board to the PC.
3. Run *Bulk\_Ping\_Host.exe* from <Install\_Directory>:\Cypress\CY3214-PSoCEVAL-USB\<version>\Firmware\Bulk\_PingExample\PC\_Host\_App.
4. A DOS-based Host program starts and sends out the data transfer requests, reads and compares the received data, and logs the data transfer results.
5. The log file BulkPing\_Log.txt is saved in the application path at <Install\_Directory>:\Cypress\CY3214-PSoCEVALUSB\<version>\Firmware\Bulk\_PingExample\PC\_Host\_App. This log file gives a detailed description of each data transfer.

Figure 5-66. Output



```

BulkPing_Log.txt - Notepad
File Edit Format View Help

1 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
2 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
3 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
4 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
5 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
6 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
7 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
8 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
9 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
10 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
11 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
12 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
13 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
14 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
15 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
16 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
17 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
18 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
19 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
20 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
21 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
22 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
23 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
24 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
25 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
26 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
27 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
28 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
29 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
30 Reply from USB/5&219D0984&0&1:bytes=62 : Passed
  
```

## 5.7 ISoc\_Example

### 5.7.1 Project Description

This example demonstrates how to use an 'alternate' interface that performs IN transactions with isochronous communication. The application reads isochronous data from the PSoC device and measures throughput. The following user module is used in the project

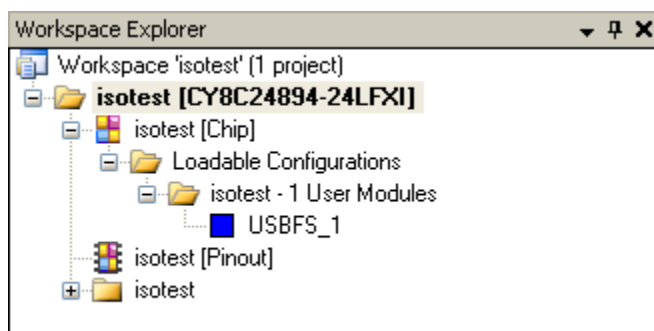
1. **USBFS\_1**: This module is used to handle the request from the USB host (PC) to communicate with the device. It handles all the data packet transfer between the host and device connected to host via USB.

### 5.7.2 Hardware Connections

No hardware connections are required.



Figure 5-67. Device Configuration of ISoc\_Example



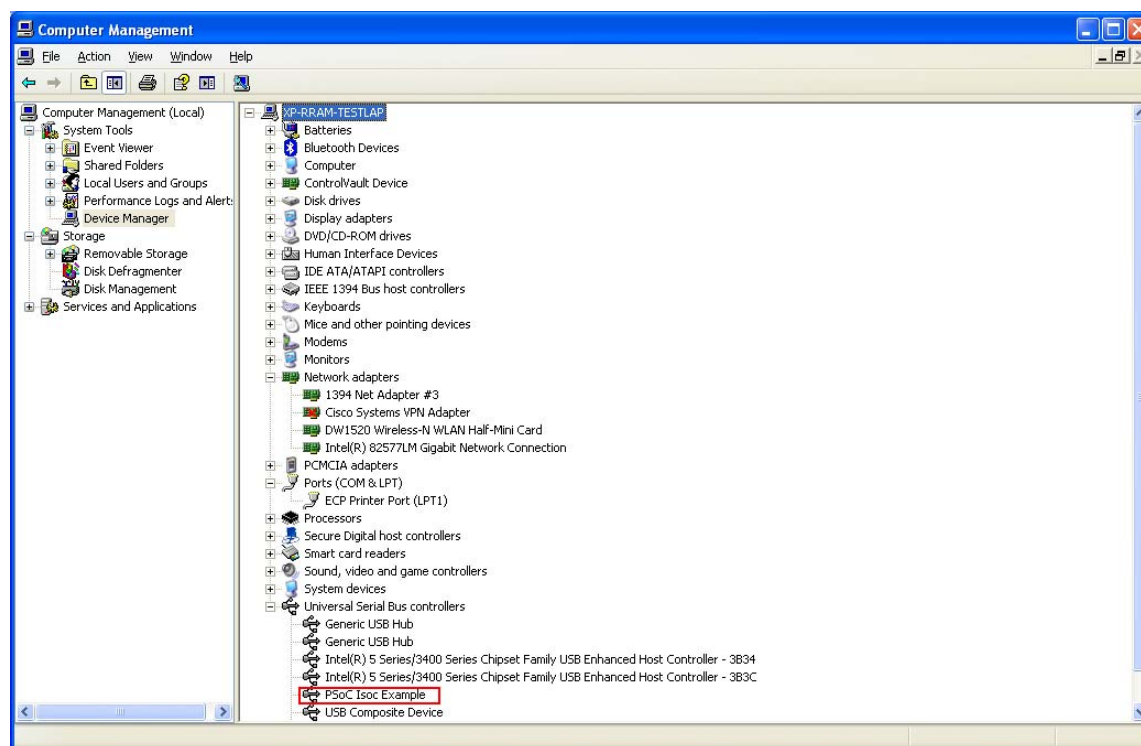
### 5.7.3 Driver Installation

Program the CY3214-PSoCEVALUSB board with the *Bulk\_ping.hex* located at `<Install_Directory>:\Cypress\CY3214-PSoCEVALUSB<version>\Firmware\ISoc_Example\PSoC Project\isotest` and connect the board to the PC using a USB cable. When connecting the board for the first time, follow steps 1 to 5 in the section [Driver Installation on page 64](#).

At step 3, browse and locate *ISoc.inf* from `<Install_Directory>:\Cypress\CY3214-PSoCEVALUSB<version>\Firmware\ISoc_Example\Driver_Setup`.

After successful driver installation, confirm in the following path: **Start > right-click My Computer > Manage > Device Manager > Universal Serial Bus Controller**.

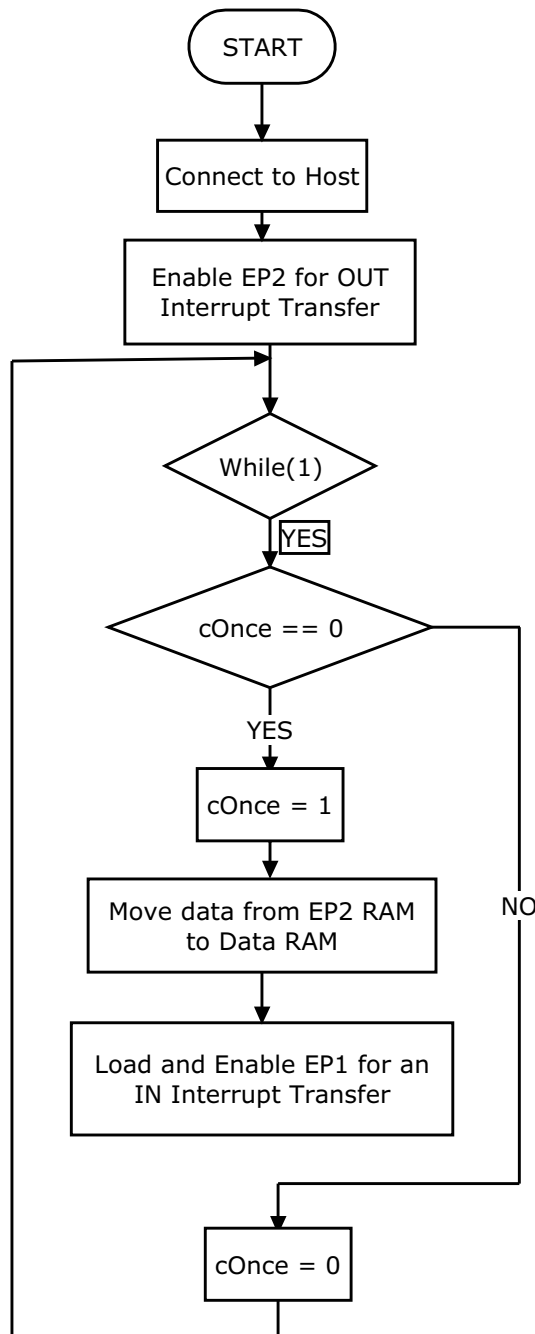
Figure 5-68. Device Management





## 5.7.4 Firmware Architecture

Figure 5-69. Flowchart

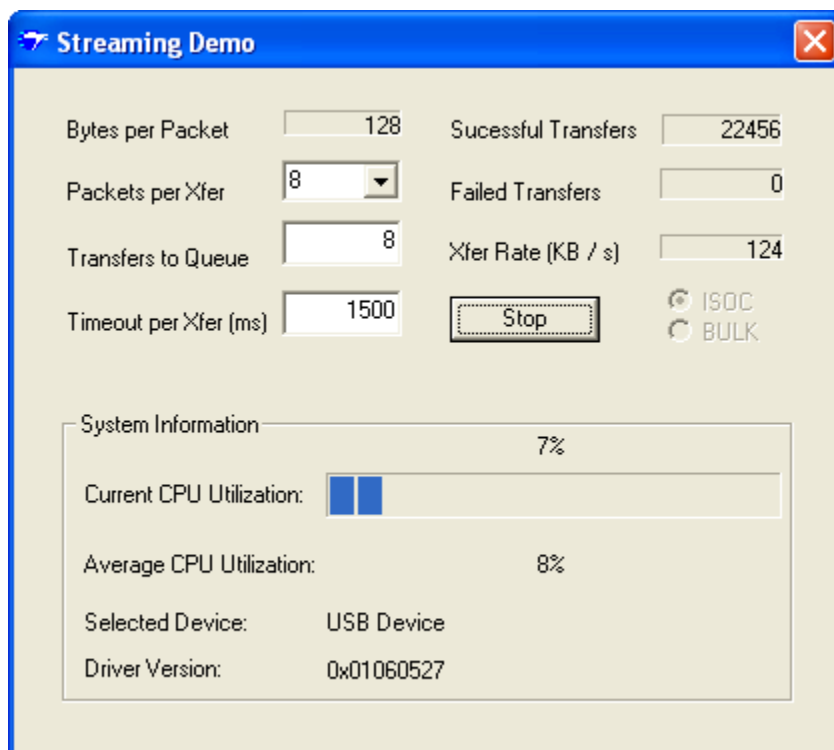


## 5.7.5 Verify Output

1. Program the board with *isotest.hex* from <Install\_Directory>\Cypress\CY3214-PSoc-EVALUSB\<version>\Firmware\ISoc\_Example\PSoc Project\isotest.
2. On successful programming, disconnect the MiniProg1 and connect the board to the PC.

3. Run *Streamer.exe* located at <Install\_Directory>:\Cypress\CY3214-PSocEVALUSB\  
<version>\Firmware\ISoc\_Example\PC\_Host\_App.
4. Click on **Start**.
5. Observe the button window **Successful Transfers**.

Figure 5-70. Output



## 5.8 USBUART

### 5.8.1 Project Description

The USBUART device uses a USB interface to emulate a COM port. UART like high level functions are available on the PSoC device side. The primary advantage of using USB as a RS-232 connection is that it makes debugging of PC applications easier. The following user module is used in the project:

- **USBUART\_1:** This module is used to handle the request from the USB host (PC) to communicate with the device. It handles the data packet transfer between the host and device connected to host through USB.

Figure 5-71. Basic USBUART Block Diagram

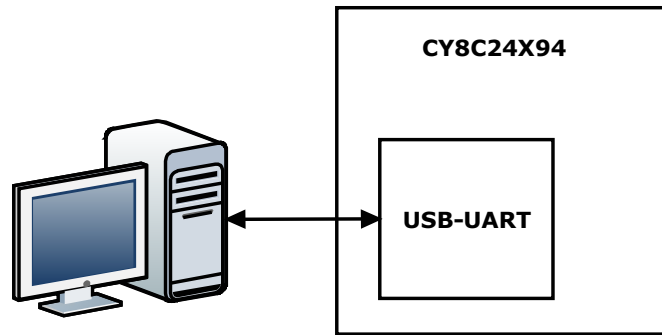
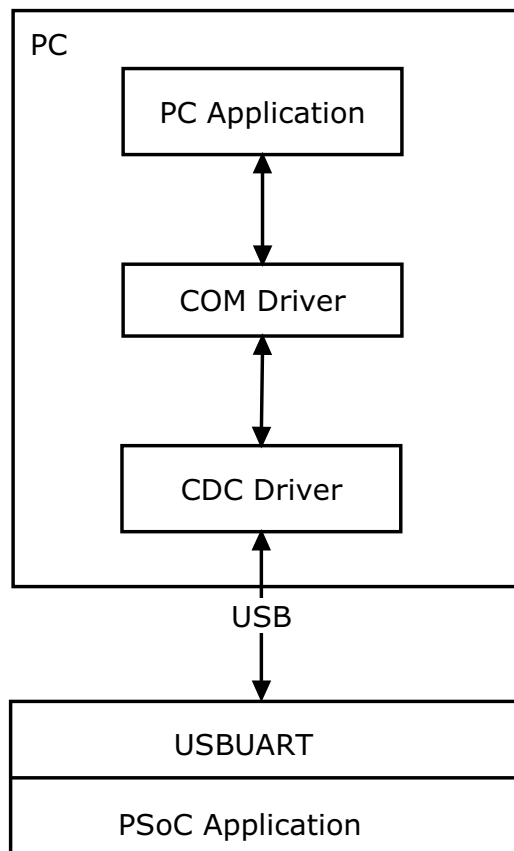


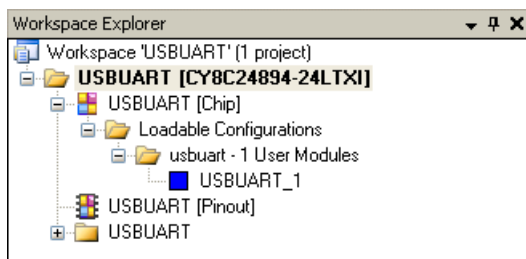
Figure 5-72. Basic Diagram



### 5.8.2 Hardware Connections

No hardware connections are required.

Figure 5-73. USBUART Device Configuration



### 5.8.3 Driver Installation

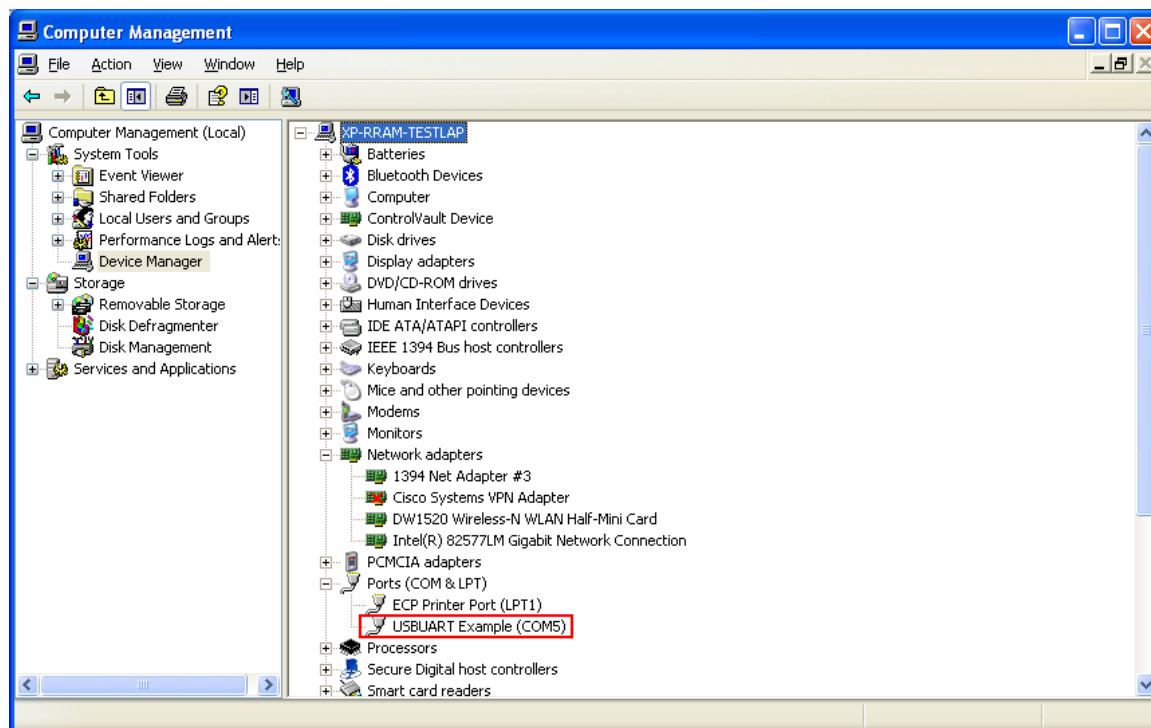
For driver installation, follow steps 1 to 6 in the section [Driver Installation on page 64](#) (sample project: Async\_PingExample).

At step 3, browse and locate the INF file from <Install\_Directory>:\Cypress\CY3214-PSo-CEVALUSB\<version>\Firmware\USBUART\USBUART\USBUART\lib.

After the successful installation of the drivers, check it in the following path:

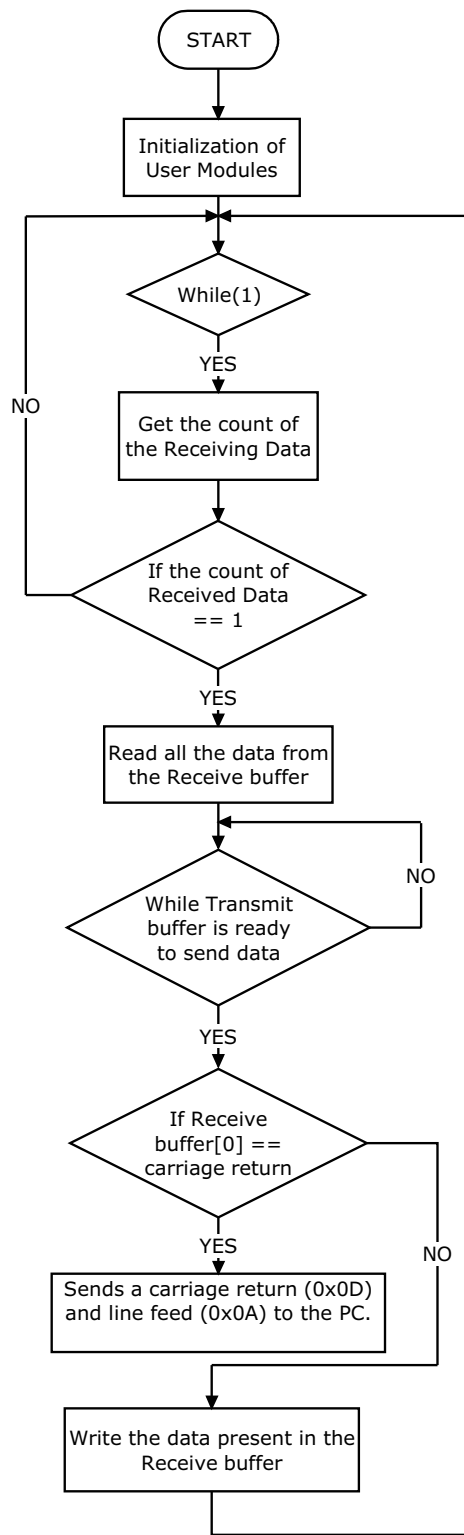
**Start > right-click on My Computer > Manage > Device Manager > Ports (COM & LPT)**

Figure 5-74. Detecting USBUART in Ports (COM &amp; LPT)



## 5.8.4 Firmware Architecture

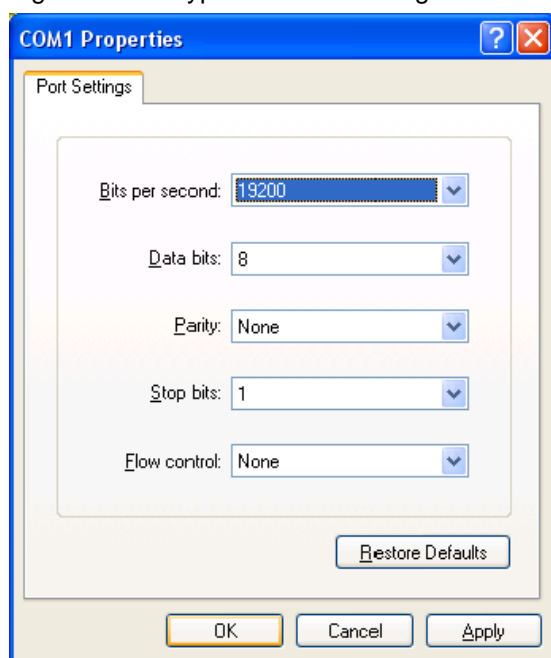
Figure 5-75. Flowchart



### 5.8.5 Verify Output

1. Program the board with *USBUART.hex* from <Install\_Directory>:\Cypress\CY3214-PSoCEVALUSB\<version>\Firmware\USBUART\USBUART, as explained in [Programming Specifications and Connections on page 16](#).
2. On successful programming, disconnect the MiniProg1 and connect a USB cable from the board to a free USB port on PC.
3. When connecting for the first time, wait for the device to be installed.
4. Ignore the Digital Signature warning when prompted; click on **Continue Anyway**.
5. Check the Device Manager for the comm port number allotted for USBUART device.
6. Open HyperTerminal and connect to the required comm port.
7. Set the comm port settings as follows: 19200-N-8-1. FlowControl: None.

Figure 5-76. HyperTerminal Settings



An echoed message displays on the HyperTerminal.

#### Notes

- The local echoing of transmit data in HyperTerminal is disabled by default. (See **File > Properties > Settings > ASCII Setup**)
- If enabled, two instances of every character transmitted are seen in the HyperTerminal: once due to local echoing and the other is the received data

Figure 5-77. ASCII Setup

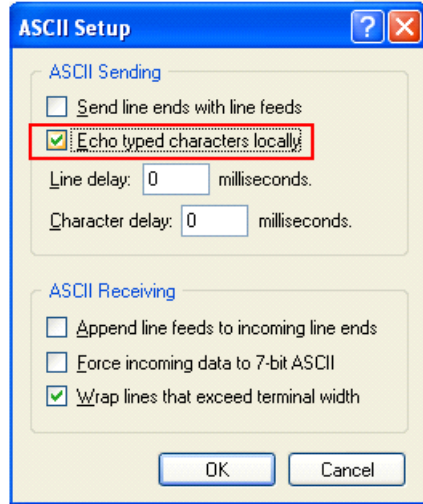
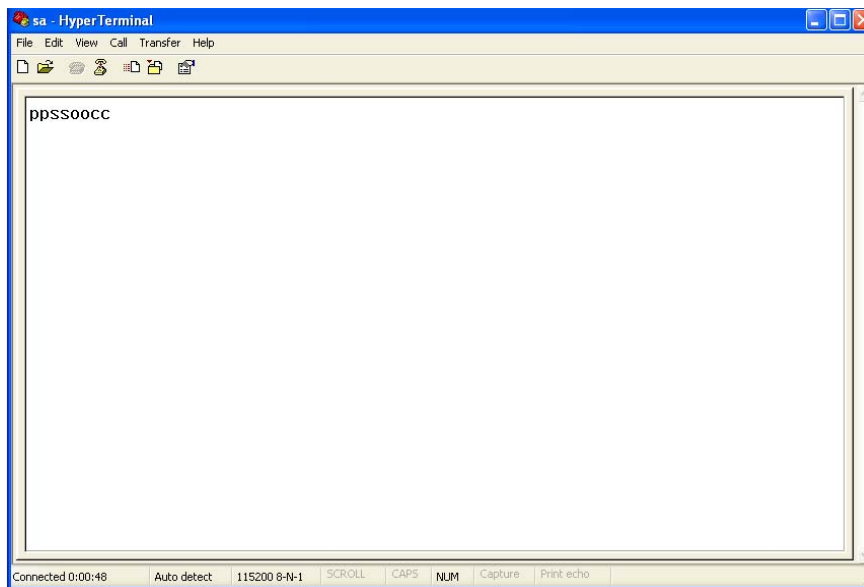


Figure 5-78. Messages Displayed on HyperTerminal



## 5.9 Joystick

### 5.9.1 Description

This project is used to control the mouse cursor movement using the joystick onboard. The joystick movements are digitized using an ADC and the change in ADC values are used to decipher the direction of movement. The USBFS module is used to enumerate as a mouse device and the mouse is controlled based on the ADC value and direction of movement. The following user modules are used in the project:

- **DUALADC:** It is used to get the samples that will be used as reference values for horizontal and vertical movement of the joystick. DUALADC is used to sample two input voltages simultaneously. Both the horizontal and vertical movements of the joystick are monitored simultaneously to detect any minimal change in position of joystick.
- **LCD:** It is used to display the message.
- **PGA:** Two PGAs are used at unity gain to supply two inputs to DUALADC
- **USBFS:** This module is used to transmit the data to control the mouse pointer based on the joystick movement.

### 5.9.2 Hardware Connection

- Connect P0.3 (P03 on J4) to JSV(J2).
- Connect P0.4 (P04 on J4) to JSH(J2).

Figure 5-79. Hardware Connection

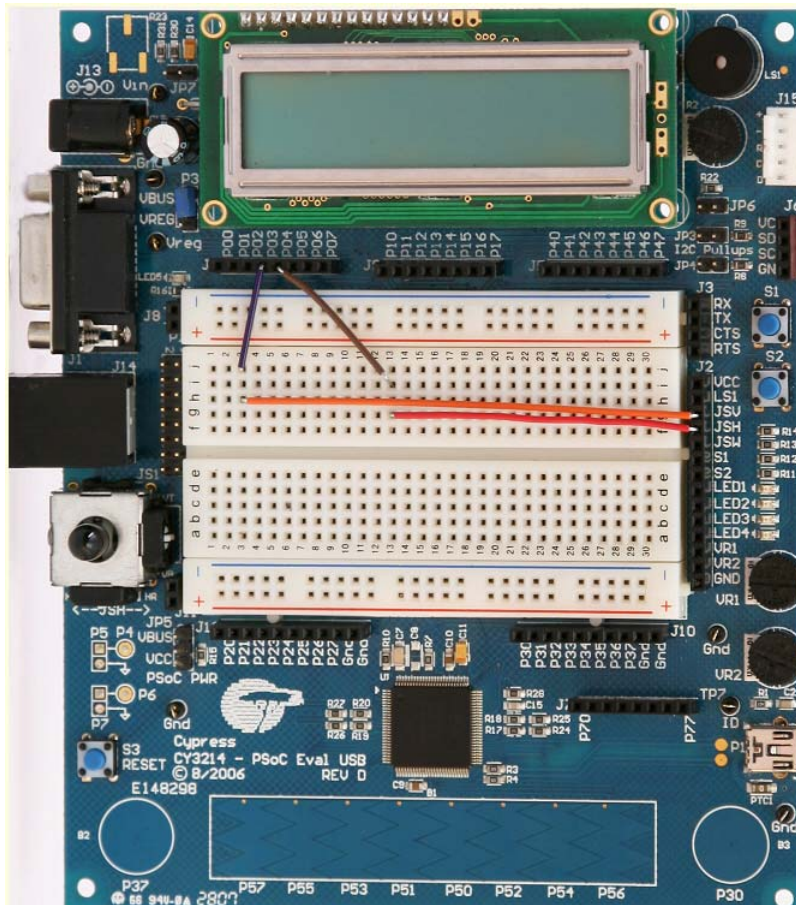
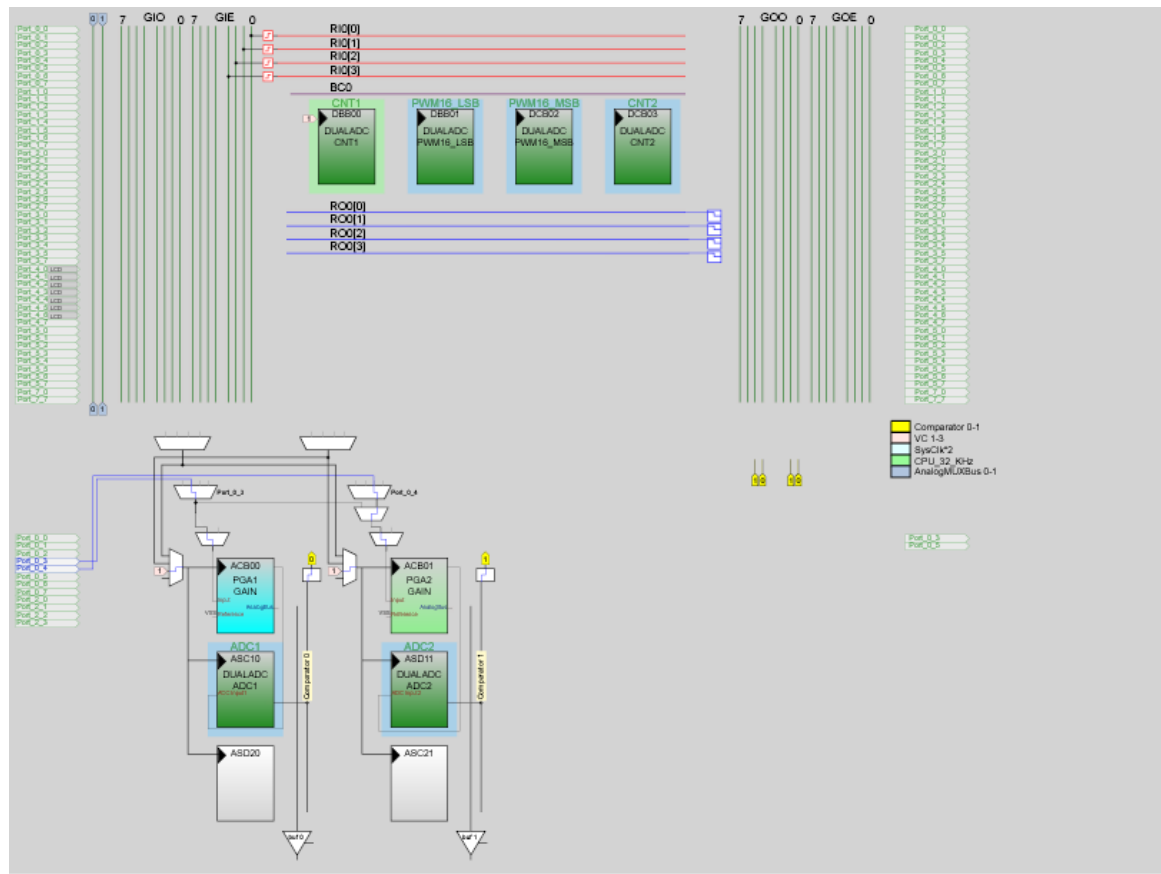


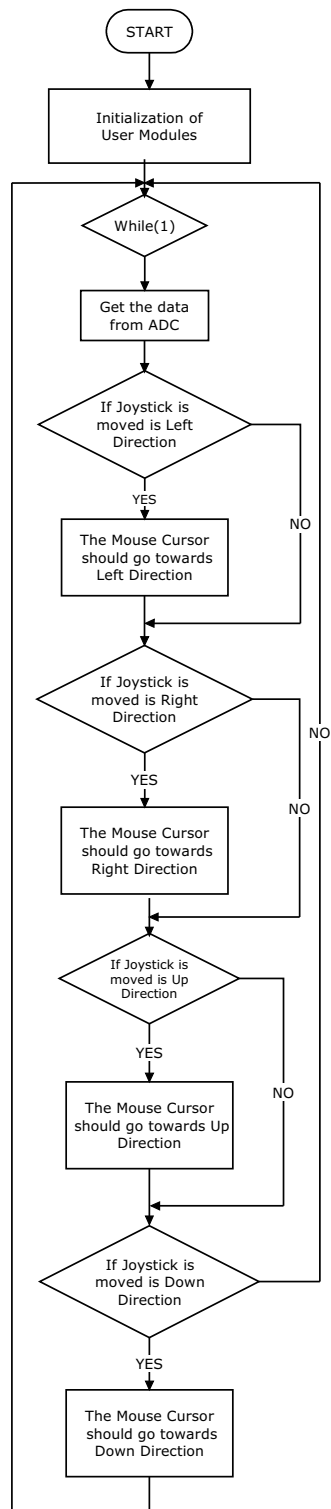


Figure 5-80. Device Configuration of Joystick



### 5.9.3 Firmware Architecture

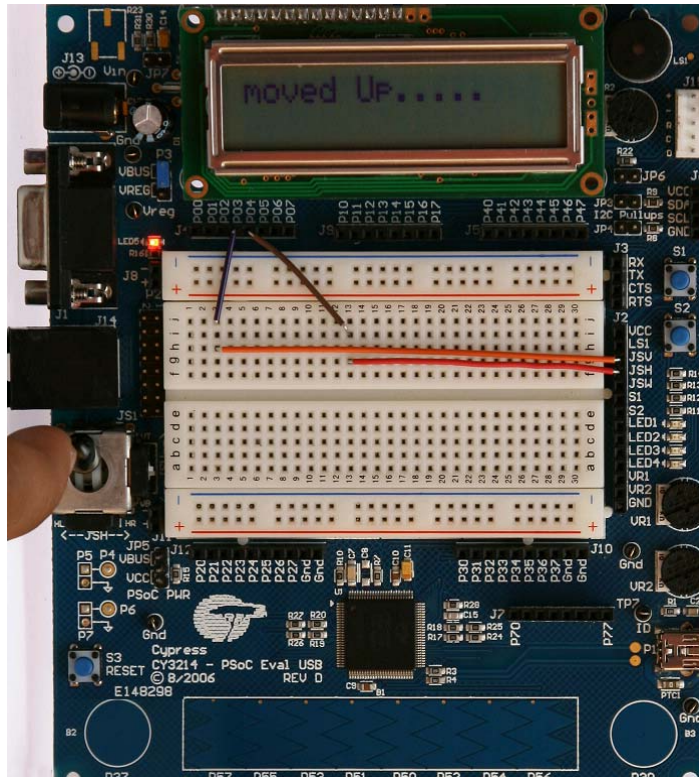
Figure 5-81. Flowchart



## 5.9.4 Verify Output

1. Program the board with *Joystick.hex* from <Install\_Directory>:\Cypress\CY3214-PSocEVALUSB\<version>\Firmware\Joystick.
2. On successful programming, disconnect the MiniProg1 and connect the board to the PC.
3. When the joystick on the kit is moved vertically up, vertically down, horizontally left, horizontally right, and also diagonally in all directions, observe the corresponding movement of the mouse pointer on your monitor screen.

Figure 5-82. Message on LCD according to Joystick Movement

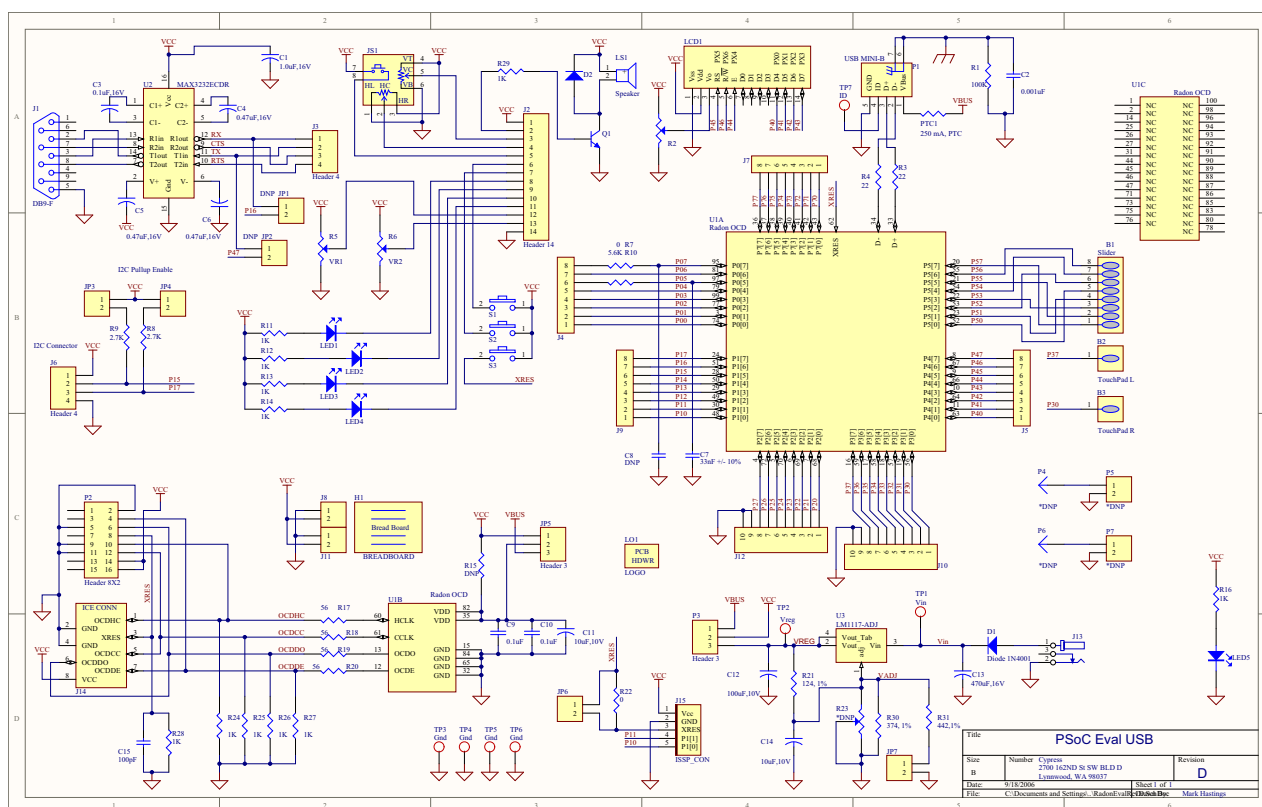




## A. Appendix

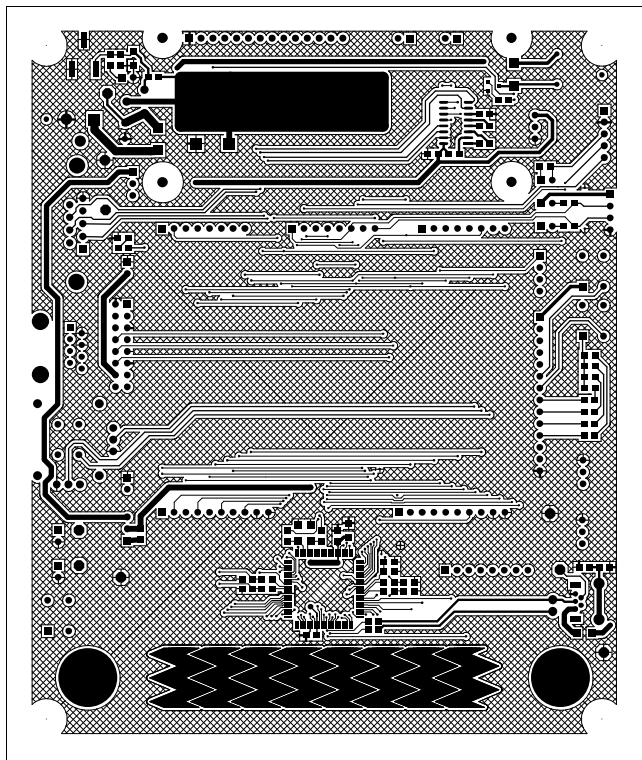


## A.1 Schematic

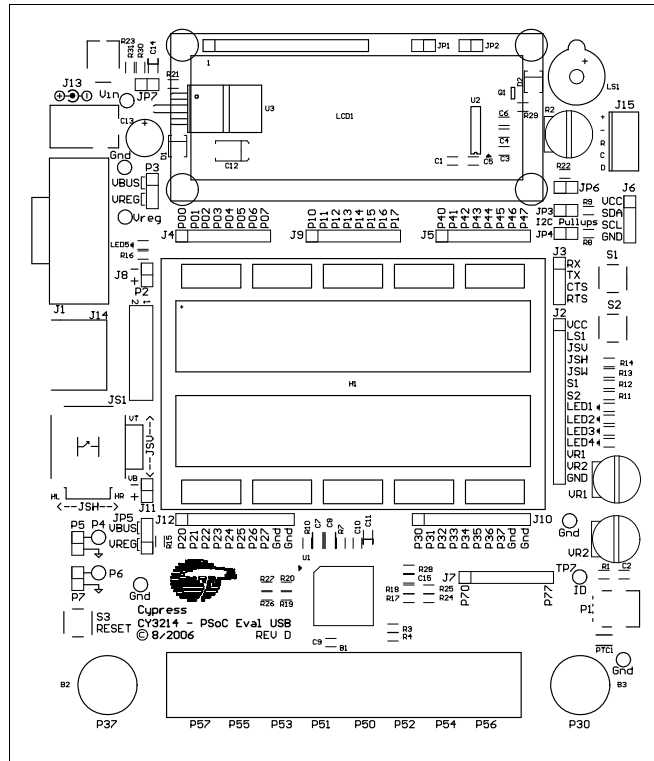


## A.2 Board Layout

### A.2.1 CY3214 -PSoCEVALUSB Top Copper

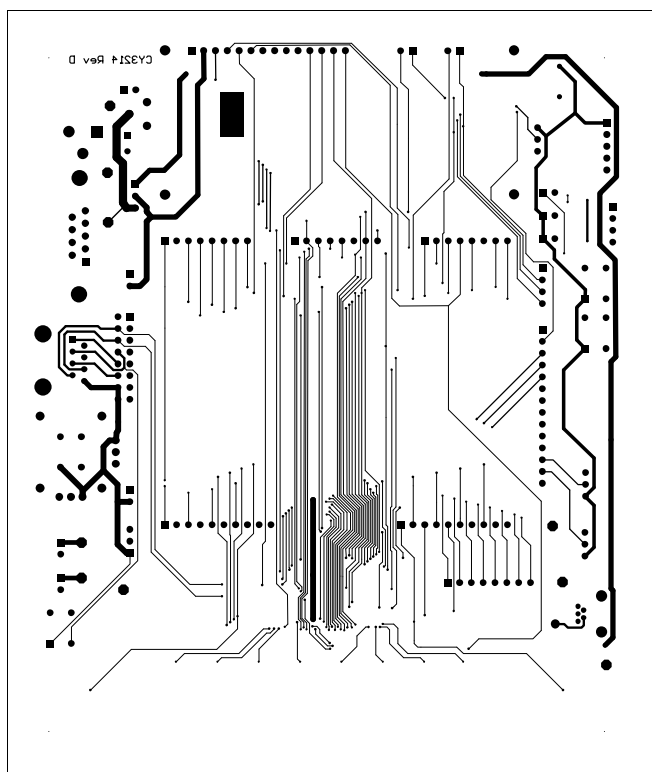


## A.2.2 CY3214-PSoCEVALUSB Top Overlay





### A.2.3 CY3214 -PSoCEVALUSB Bottom Copper



## A.3 Bill of Materials (BOM)

Description	Designator	Footprint	Qty.	Value	Vender	VPN
CY3214-PSoC EvalUSB PCB			1			121R-36800
Capacitor	C1	CC2012-0805	1	1.0uF,16V	Digikey	PCC1849TR-ND
Capacitor	C2	CC2012-0805	1	0.001uF	Digikey	399-1148-2-ND
Capacitor	C3, C9, C10	CC2012-0805	3	0.1uF	Digikey	PCC1864TR-ND
Capacitor	C4, C5, C6	CC2012-0806	3	0.47uF,16V	Digikey	PCC1847TR-ND
Capacitor	C11, C14	EIA3216-18	2	10uF,10V	Digikey	495-2201-1-ND
Capacitor	C12	EIA7343-31	1	100uF,10V	Digikey	495-1529-1-ND
Capacitor	C13	RB3.5X8	1	470uF,16V	Digikey	493-1286-ND
Capacitor	C15	CC2012-0805	1	100pF	Digikey	399-1121-2-ND
1 Amp General Purpose Rectifier	D1, D2	SMB Diode	2	1A,50V	Digikey	S1AB-FDICT-ND
3M solderless breadboard super strip	H1	BBoard3.25x2.125	1		Jameco	WBU-301-R
Female DB-9	J1	DB9F	1	DB9-F	Digikey	A32115-ND
Header, 14-Pin	J2	HDR1X14	1	14	Digikey	929850-01-14-ND
Header, 4-Pin	J3, J6	HDR1X4	2	4	Digikey	929850-01-04-ND
8-Pin Header, Female	J4, J5, J7, J9	HDR8X1	4	8	Digikey	929850-01-08-ND
Header, 2-Pin, Female	J8, J11	HDR2x1	2	2	Digikey	929850-01-02-ND
Header, 10-Pin	J10, J12	HDR1X10	2	10	Digikey	929850-01-10-ND
Power connector	J13	PWRCON	1		Digikey	CP-102A-ND
ICE Connection	J14	RJ45	1		Digikey	A31465-ND
ISSP Conn	J15	ISPCON	1		Digikey	WM4203-ND
Header, 2-Pin, Male	JP1, JP2, JP3, JP4, JP5, JP6, JP7	HDR2x1	7	2	Digikey	S1011E-36-ND
Mini Joystick	JS1	CTS_MiniStick_252	1		Digkey	252A103B60NB-ND
Standard 14 pin LCD interface	LCD1	LCDMODULE-T14	1		Digikey	929850-01-14-ND
Red LED	LED1, LED2, LED3, LED4, LED5	LED0805	5	Red	Digikey	67-1552-2-ND
Loudspeaker	LS1	CEM-1206S	1		Digikey	102-1155-ND
USB type B connector	P1	USB_MINI-B_TMP	1		Digikey	WM17115-ND
Header, 8-Pin, Dual row	P2	HDR2X8REV	1		Digikey	WM6816-ND
Header, 3-Pin	P3	HDR1X3	1	3	Digikey	A19330-ND
250 mA PTC	PTC1	1206	1	250 mA, PTC	Digikey	F2110CT-ND
NPN General Purpose Amplifier	Q1	SOT23-3	1		Digikey	FMMT491ATR-ND
Resistor, SMT	R1	R2012-0805	1	100K	Digikey	P100KACT-ND
Potentiometer	R2, R5, R6	POT-3352T	3		Digikey	3352T-1-103LF-ND
Resistor, SMT	R3, R4	R2012-0805	2	22	Digikey	P22ACT-ND
Resistor, SMT	R7, R15, R22	R2012-0805	3	0	Digikey	P0.0ACT-ND
Resistor, SMT	R8, R9	R2012-0805	2	2.7K	Digikey	P2.7KACT-ND
Resistor, SMT	R11, R12, R13, R14, R16, R24, R25, R26, R27, R28, R29	R2012-0805	11	1K	Digikey	P1.0KACT-ND
Resistor, SMT	R17, R18, R19, R20	R2012-0805	4	56	Digikey	P56ACT-ND
Resistor, SMT	R21	R2012-0805	1	124, 1%	Digikey	P124CTR-ND
Resistor, SMT	R30	R2012-0805	1	374, 1%	Digikey	P374CTR-ND
Resistor, SMT	R31	R2012-0805	1	442,1%	Digikey	P442CTR-ND
Switich, SPST	S1, S2, S3	SW_SPST6.5/4.5	3		Digikey	SW403-ND

Description	Designator	Footprint	Qty.	Value	Vender	VPN
Simple Test point	TP1, TP2, TP3, TP4, TP5, TP6, TP7	TP64	7		Digikey	5006K-ND
Radon OCD Part	U1	100TQFP	1	RADON	Cypress	CY8C24094-24AXI
RS-232 transceiver (0.47uF Caps)	U2	SO-16	1		Digikey	296-19851-2-ND
Voltage Regulator	U3	TO-220	1	ADJ	Digikey	497-3456-5-ND
Capacitor	C7	1206	1	0.010uF +/- 10%	Digikey	PCC2292CT-ND
Resistor, SMT	R10	R2012-0805	1	5.6K	Digikey	P5.6KACT-ND
<b>SPECIAL JUMPER INSTALLATION:</b>						
Jumper pins 2 and 3 "VREG" on P3			1	2-pos 0.1"	Digikey	STC02SYAN
<b>DO NOT POPULATE:</b>						
Potentiometer	R23	POT3361P	1	500	Digikey	3361P-1-501GLFCT-ND
Capacitor	C8	1206	0	0.033uF ± 10%	Digikey	478-1548-1-ND
Plug	P4, P6	PIN1	0	1	na	na
Header, 2-Pin	P5, P7	HDR2x1	0	2	na	na

Компания «Океан Электроники» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Поставка оригинальных импортных электронных компонентов напрямую с производств Америки, Европы и Азии, а так же с крупнейших складов мира;
- Широкая линейка поставок активных и пассивных импортных электронных компонентов (более 30 млн. наименований);
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Помощь Конструкторского Отдела и консультации квалифицированных инженеров;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Поставка электронных компонентов под контролем ВП;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- При необходимости вся продукция военного и аэрокосмического назначения проходит испытания и сертификацию в лаборатории (по согласованию с заказчиком);
- Поставка специализированных компонентов военного и аэрокосмического уровня качества (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Actel, Aeroflex, Peregrine, VPT, Syfer, Eurofarad, Texas Instruments, MS Kennedy, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits, General Dynamics и др.);

Компания «Океан Электроники» является официальным дистрибьютором и эксклюзивным представителем в России одного из крупнейших производителей разъемов военного и аэрокосмического назначения «JONHON», а так же официальным дистрибьютором и эксклюзивным представителем в России производителя высокотехнологичных и надежных решений для передачи СВЧ сигналов «FORSTAR».



## JONHON

«JONHON» (основан в 1970 г.)

Разъемы специального, военного и аэрокосмического назначения:

(Применяются в военной, авиационной, аэрокосмической, морской, железнодорожной, горно- и нефтедобывающей отраслях промышленности)

«FORSTAR» (основан в 1998 г.)

ВЧ соединители, коаксиальные кабели,  
кабельные сборки и микроволновые компоненты:

(Применяются в телекоммуникациях гражданского и специального назначения, в средствах связи, РЛС, а так же военной, авиационной и аэрокосмической отраслях промышленности).



Телефон: 8 (812) 309-75-97 (многоканальный)

Факс: 8 (812) 320-03-32

Электронная почта: [ocean@oceanchips.ru](mailto:ocean@oceanchips.ru)

Web: <http://oceanchips.ru/>

Адрес: 198099, г. Санкт-Петербург, ул. Калинина, д. 2, корп. 4, лит. А