

CY3209-ExpressEVK Kit Guide

Doc. # 001-16798 Rev. **

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

© Cypress Semiconductor Corporation, 2007. 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.

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

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's 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.

Contents

1. Introduction **7**

1.1 Purpose 7
1.2 Getting Started 7
1.3 Design Files 7
1.4 Adding Drivers and Valuators to a Design 7
1.5 MiniProg Installation 9
1.6 Programming Instructions for CY3209 PSoC Express Projects 10
1.7 CY3209-ExpressEVK Board 10
1.8 Conventions 11
1.9 Document Revision History 11

2. First Example **13**

2.1 Description 13
2.2 Board Quadrant 13
2.3 PSoC Device 13
2.4 Finished Project Location 13
2.5 Start a New Project 13
2.6 Add an Input Driver 13
2.7 Add an Output Driver 14
2.8 Define Output Behavior 14
2.9 Simulate the Design 15
2.10 Build the Project 15
2.11 Program the PSoC Device 16
2.12 Test Your Project on the Demonstration Board 16

3. I2C Master/Slave Lab **17**

3.1 I2C Master/Slave Lab – Slave 17
3.1.1 Description 17
3.1.2 Board Quadrant 17
3.1.3 PSoC Device 17
3.1.4 Finished Project Location 17
3.1.5 Start a New Project 17
3.1.6 Add an Input Driver 18
3.1.7 Add an Interface Valuator 18
3.1.8 Add an Output Driver 19
3.1.9 Define Output Behavior 19
3.1.10 Add an Interface Driver 20
3.1.11 Verify the Project Register Map 20
3.1.12 Build the Project 20
3.1.13 Program the PSoC Device 21
3.2 I2C Master/Slave Lab – Master 22
3.2.1 Description 22

3.2.2 Board Quadrant 22
3.2.3 PSoC Device 22
3.2.4 Finished Project Location 22
3.2.5 Start a New Project 22
3.2.6 Add an Input Driver 22
3.2.7 Add Two Output Drivers 23
3.2.8 Define Output Behavior 24
3.2.8.1 FourDigLED 24
3.2.8.2 LevelControl 24
3.2.9 Build the Project 24
3.2.10 Program the PSoC Device 26
3.2.11 Test Your Project on the Demonstration Board 26

4. USB-UART Lab 27

4.1 Description 27
4.2 Board Quadrant 27
4.3 PSoC Device 27
4.4 Finished Project Location 27
4.5 Start a New Project 27
4.6 Add an Input Driver 28
4.7 Add an Interface Valuator 28
4.8 Add an Output Driver 29
4.9 Define Output Behavior 29
4.10 Add an Interface Driver 30
4.11 Verify the Project Register Map 30
4.12 Build the Project 30
4.13 Program the PSoC Device 32
4.14 Test Your Project on the Demonstration Board 32

5. CapSense Lab 35

5.1 Description 35
5.2 Board Quadrant 35
5.3 PSoC Device 35
5.4 Finished Project Location 35
5.5 Program I2C-USB Bridge Firmware 35
5.6 Start a New Project 36
5.7 Add an Interface Driver 36
5.8 Add an Output Driver 36
5.9 Add an Input Driver 37
5.10 Define Output Behavior 38
5.11 Duplicate the Button and LED 38
5.12 Build the Project 40
5.13 Program the PSoC Device 41
5.14 Test Your Project on the Demonstration Board 42
5.15 Tuning Your Project 42

6. LCD Lab 45

6.1 Description 45
6.2 Board Quadrant 45
6.3 PSoC Device 45
6.4 Finished Project Location 45
6.5 Start a New Project 45

6.6 Add an Input Driver 45
6.7 Add an Output Driver 46
6.8 Define Output Behavior 46
6.9 Build the Project 46
6.10 Program the PSoC Device 48
6.11 Test Your Project on the Demonstration Board 48

7. Cypress WirelessUSB™ Master/Slave Lab 1 49

7.1 WirelessUSB Master/Slave Lab – Slave 49
7.1.1 Description 49
7.1.2 Board Quadrant 49
7.1.3 PSoC Device 49
7.1.4 Finished Project Location 49
7.1.5 Start a New Project 49
7.1.6 Add an Input Driver 50
7.1.7 Add an Interface Valuator 50
7.1.8 Add an Output Driver 51
7.1.9 Define Output Behavior 51
7.1.10 Add an Interface Driver 52
7.1.11 Verify the Project Register Map 52
7.1.12 Build the Project 53
7.1.13 Program the PSoC Device 54
7.2 WirelessUSB Master/Slave Lab – Master 55
7.2.1 Description 55
7.2.2 Board Quadrant 55
7.2.3 PSoC Device 55
7.2.4 Finished Project Location 55
7.2.5 Start a New Project 55
7.2.6 Add an Input Driver 55
7.2.7 Add Three Output Drivers 56
7.2.8 Define Output Behavior 57
7.2.9 Build the Project 58
7.2.10 Program the PSoC Device 60
7.2.11 Test Your Project on the Demonstration Board 60

8. Cypress WirelessUSB™ Master/Slave Lab 2 61

8.1 WirelessUSB Master/Slave Lab – Slave 61
8.1.1 Description 61
8.1.2 Board Quadrant 61
8.1.3 PSoC Device 61
8.1.4 Finished Project Location 61
8.1.5 Start a New Project 61
8.1.6 Add an Input Driver 62
8.1.7 Add Two Output Drivers 62
8.1.8 Add an Interface Driver 63
8.1.9 Add Two Interface Valuators 63
8.1.10 Define Output Behavior 64
8.1.11 Verify the Project Register Map 64
8.1.12 Build the Project 65
8.1.13 Program the PSoC Device 66
8.1.14 Test Your Slave Device on the Demonstration Board 67
8.2 WirelessUSB Master/Slave Lab – Master 68

8.2.1 Description 68
8.2.2 Board Quadrant 68
8.2.3 PSoC Device 68
8.2.4 Finished Project Location 68
8.2.5 Start a New Project 68
8.2.6 Add Three Input Drivers 68
8.2.7 Add Five Output Drivers 69
8.2.8 Define Output Behavior 71
8.2.9 Build the Project 73
8.2.10 Program the PSoC Device 74
8.2.11 Test Your WirelessUSB System 75

1. Introduction

1.1 Purpose

This guide provides instructions for example projects for the CY3209-ExpressEVK. The projects illustrate many features of PSoC Express 3.

1.2 Getting Started

You will need:

- PSoC Express 3
- PSoC Programmer
- Express Pack 3

1.3 Design Files

The example projects in this guide are included on the kit CD. These projects have already gone through the build process and can be programmed immediately. Follow the paths listed in the Finished Project Location section at the beginning of each chapter to open the project files on the kit CD.

1.4 Adding Drivers and Valuators to a Design

It is very easy to add drivers and valuators to a design in PSoC Express 3. Throughout this guide, when a driver or valuator needs to be added to a design, a small table displays the required info. See Table 1-1 for an example. The table is broken into three sections:

- Device/Valuator: Shows the actual name of the driver or valuator.
- Location: Shows the path to the device in the Driver Catalog.
- Properties: Lists all properties associated with a device.

Table 1-1. Example of an Add Driver or Valuator Table

Driver/Valuator	Banked Output	
Location	Outputs Tab ↳ Digital Output ↳ Banked Output	
Properties	**Field**	**User Input**
	Name	XLeds
	Number of output pins	4
	DriveMode	Strong (default)

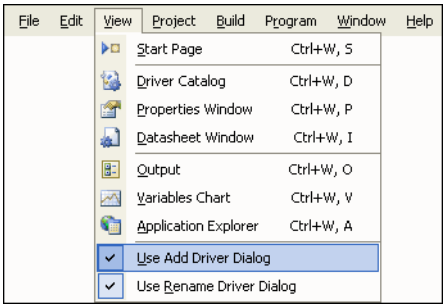
Note: To add a driver to a design, you must open a new or existing. Adding new drivers does not work in the Start Page in PSoC Express.

To add drivers and valuators to your design:

1. Ensure the Driver Catalog pane is visible. If it is not visible, select **View** → **Driver Catalog**.
2. Each driver and valuator has certain properties associated with it. For the purposes of this tutorial, ensure **View** → **Use Add Driver Dialog** is checked.

   Note: Some of the projects in this guide will not work unless the Add Driver Dialog option is selected.

   Figure 1-1. Use Add Driver Dialog Menu Option is Selected



3. In PSoC Express, there are three ways to add drivers and valuators to a design. In the Driver Catalog pane, navigate to the driver or valuator you want to add, and either drag-and-drop the item to your design (Figure 1-2), double-click the item, or right-click on the item and select **Add to Design** (Figure 1-3).

   Figure 1-2. Adding a Driver Via Drag and Drop

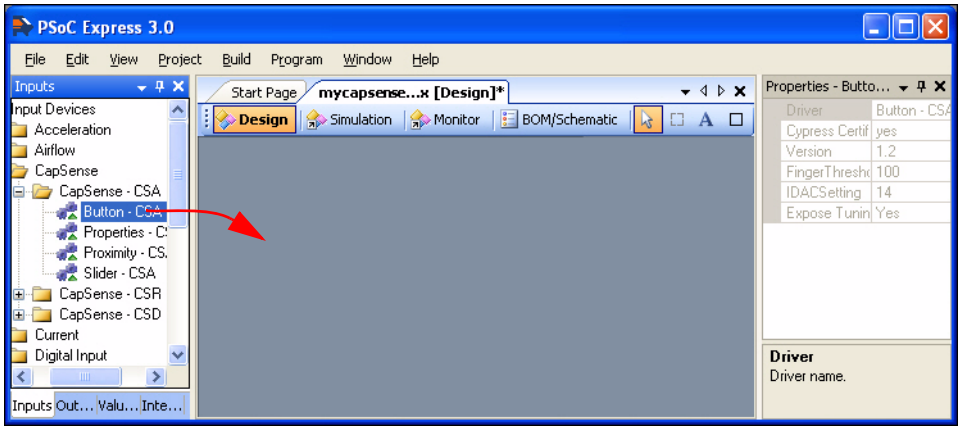
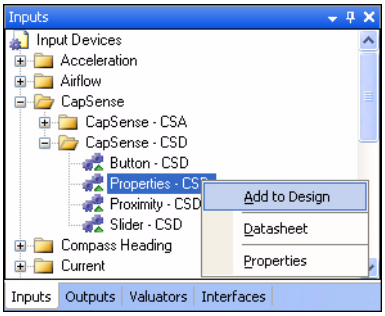
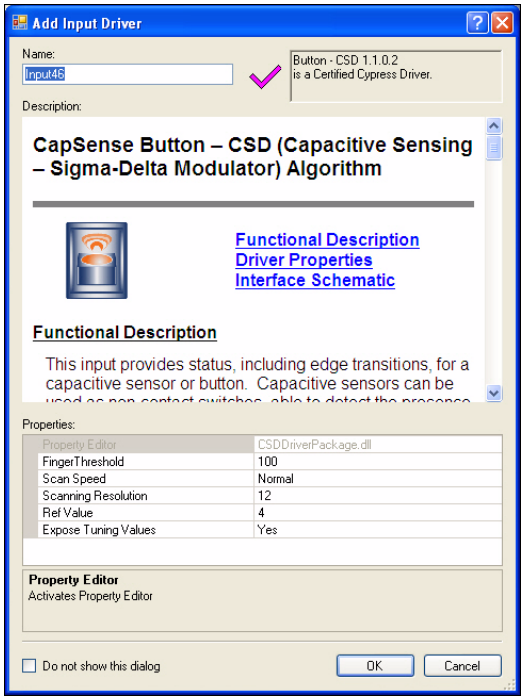


   Figure 1-3. Adding a Driver Via the Contextual Menu



4. When drivers and valuators are added to a design, the Add Driver window opens (Figure 1-4). Name the item and enter the properties required for the design.

   Figure 1-4. Add Driver Window



5. Click **OK**.

1.5 MiniProg Installation

1. Ensure all content on the CY3209 CD is installed.
2. The first time you plug the MiniProg into your PC via the USB cable, the Windows Found New Hardware Wizard opens.
3. Select the **No, not this time** option button, and click **Next**.
4. Select the **Install from a list or specific location (Advanced)** option button and click **Next**.
5. Select the **Don't search. I will choose the driver to install** option button, and click **Next**.
6. Select the **USB Mini Programmer (1.08.0120.0)** driver and click **Next**.
7. When prompted that the USB Mini Programmer (1.08.0120.0) software has not passed Windows Logo Testing, click **Continue Anyway**.
8. When Windows has completed the installation, click **Finish**.

1.6 Programming Instructions for CY3209 PSoC Express Projects

These instructions assume that the board is initially unpowered.

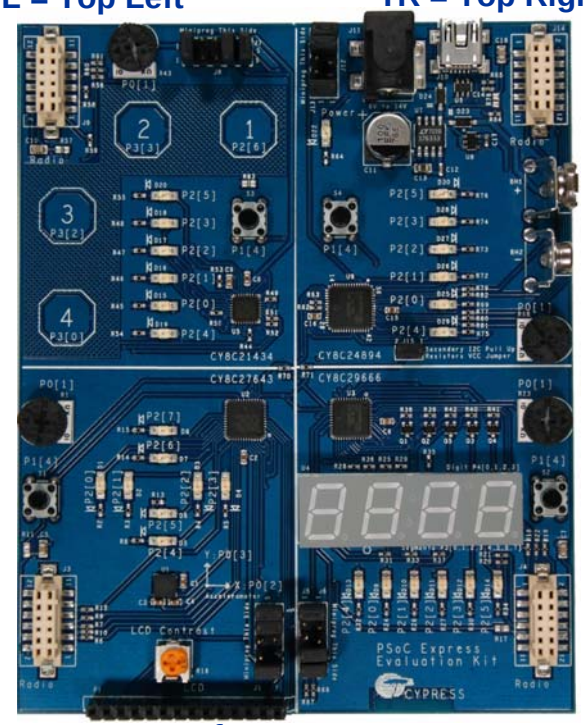
1. Connect the USB cable to the MiniProg and to the PC.
2. Connect the MiniProg to the programming header specified by the lab instructions.
3. Select **Program** → **PSoC Programmer** from PSoC Express. The following three actions are automatically accomplished:
   a. PSoC Programmer starts.
   b. The executable firmware is loaded into PSoC Programmer.
   c. The PSoC device to be programmed is specified in PSoC Programmer.

Note: If the Actions column in the PSoC Programmer status box indicates "Firmware update required at ...", the MiniProg's firmware is out of date. Before attempting to program the target PSoC chip, go to Utilities → Upgrade Firmware to update the MiniProg's firmware.

4. In PSoC Programmer:
   a. Select the **MiniProg** from the Port dropdown list.
   b. Select the **Power Cycle** option button.
   c. Click the **Program** button in the toolbar.
   d. Wait until the Actions column of PSoC Programmer's status window indicates either Programming Succeeded or Programming Terminated. If the Programming Terminated message appears, troubleshoot the problem, and re-attempt to program the device.
   e. Exit PSoC Programmer.
5. Disconnect the MiniProg from the CY3209 board.

1.7 CY3209-ExpressEVK Board

TL = Top Left **TR = Top Right**



BL = Bottom Left **BR = Bottom RIght**

1.8 Conventions

These conventions are used throughout this guide.

Table 1-2. Documentation Conventions

Convention	Usage
`Courier New Size 12`	Displays file locations and source code: `C:\ …cd\icc\`.
Italics	Displays file names and reference documentation: *sourcefile.hex*
[**bracketed, bold**]	Displays keyboard commands in procedures: [**Enter**] or [**Ctrl**] [**C**]
Bold → With → Arrows	Represents menu paths, user entered text: **File → New Project → Clone**
Bold	Displays commands and selections, and icon names in procedures: Click the **Debugger** icon, and then click **Next**.
Note:	Displays functionality unique to PSoC Designer or the PSoC device.
WARNING:	Displays cautions that are important to the subject.

1.9 Document Revision History

Document Title: CY3209-ExpressEVK Kit Guide **Document Number: 001-16144**				
Revision	**ECN#**	**Issue Date**	**Origin of Change**	**Description of Change**
**	1421984	June 11, 200	SFVTMP3	New guide for CY3209-ExpressEVK

2. First Example

2.1 Description

The example implements a pushbutton used to control a decaying LED.

2.2 Board Quadrant

Top Left

2.3 PSoC Device

CY8C21434

2.4 Finished Project Location

`\PSoCExpressExamples\CY3209FirstEx\CY3209FirstEx.app` on the kit CD.

2.5 Start a New Project

1. Select **File** → **New Project**.
2. Name the project **CY3209FirstEx**.
3. If needed, click **Browse** to save the project in a different location.
4. Click **OK**.

2.6 Add an Input Driver

Table 2-1. Internal Pulldown N_O Driver

Driver/Valuator	Pushbutton - (Normally Open, Int. Pull-Down)	
Location	Inputs Tab ↳ Tactile ↳ Button ↳ Normally Open ↳ Internal Pulldown N_O	
Properties	**Field**	**User Input**
	Name	Button

2.7 Add an Output Driver

Table 2-2. With Decay Driver

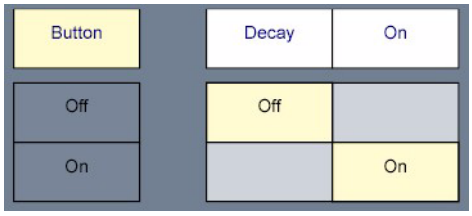
Driver/Valuator	Single LED - With Decay		
Location	Outputs Tab ↳ Display ↳ LED ↳ Single Color ↳ With Decay		
Properties	**Field**	**User Input**	
	Name	LED	
	Initial Value	Decay	(Default)
	Decay Time	1000	(Default)
	DriveFrequency	1000	(Default)
	Current Mode	Sourcing	(Default)

Note: Optional changes to the driver properties include changing the Decay Time and DriveFrequency.

2.8 Define Output Behavior

1. Right-click on the **LED** driver icon and select **Transfer Function**.
2. Select **TableLookup** in the list of transfer functions, and click **OK**.
3. Select **Button** in the Select Input(s) list.
4. Click **Next**.
5. Drag the Button Input values to the following LED state columns:
   - Off → Decay
   - On → On

Figure 2-1. CY3209FirstEx Table Lookup

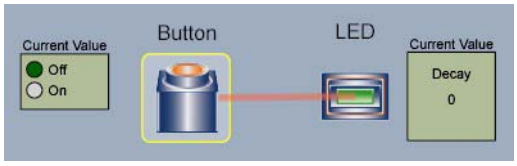


6. Click **OK**.

2.9 Simulate the Design

1. Click the **Simulation** option at the top of the design desktop. Rectangular widgets appear beside their respective driver icons.
2. Click the Button driver **On** and **Off** option buttons and observe the status of the LED output driver.

Figure 2-2. CY3209FirstEx Simulation



2.10 Build the Project

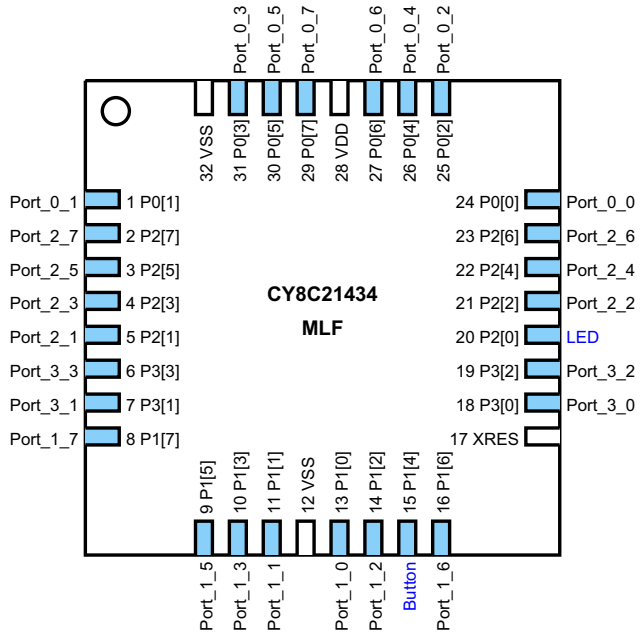
1. Select **Build** → **Generate/Build 'CY3209FirstEx' Project**.
2. In the Available Device Configurations pane, select **CY8C21030\CY8C21434, 32-Pin**.
3. In the Device Configurations Properties pane, set the following parameters:
   - Supply Voltage → 3.3V
   - Sample Rate → 64 Hz
   - Flash Interface → Disable
4. Ensure the **Assign pins automatically** check box is **not** selected.
5. Click **Next**.
6. In the User Pin Assignment window, click **Unassign All Pins**.
7. Assign drivers to pins by dragging and dropping the blue rectangles from the Unassigned Drivers list to the pins listed in the following table.

Table 2-3. CY3209FirstEx Pinouts

Function	Pin
Button	P1[4]
LED	P2[0]

Note: The pin label notation P1[2] indicates Port 1, Pin 2.

Figure 2-3. CY3209FirstEx Pinouts

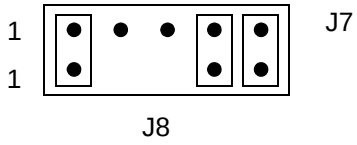


8. Click **Next**. PSoC Express generates and compiles the project source code.

2.11 Program the PSoC Device

1. Disconnect any power sources from the demonstration board.
2. Remove the jumpers between J7 and J8.
3. Program the part via J7 as described in the Programming Instructions for CY3209 PSoC Express Projects section on page 10.
4. Install jumpers on the following pin combinations:

Figure 2-4. CY3209FirstEx Jumper Settings



2.12 Test Your Project on the Demonstration Board

1. Power the board with a 9V battery (provided with the kit) or an optional 6–14V DC wall transformer.
2. In the top-left quadrant, press the pushbutton on the board to light the LED labeled P2[0].
3. Release the button and watch the LED slowly fade.

3. I2C Master/Slave Lab

This lab creates two projects: an I2C slave and an I2C master. The I2C master device reads a voltage value from the slave and displays the value on a four digit LED display. The master also writes a control value to the slave device to set an LED to Off, On, or Blinking depending on the voltage level read from the slave.

3.1 I2C Master/Slave Lab – Slave

3.1.1 Description

This project is the slave device for the I2C Master/Slave lab. The project has a voltage input and an LED that supports three states: Off, On and Blinking. The LED state is controlled by the I2C Master project through an Interface Valuator. The project becomes an I2C slave by adding an I2C Slave Interface driver.

Note: This project will not work unless the Add Driver Dialog option is selected. To use the Add Driver Dialog, ensure **View** → **Use Add Driver Dialog** is checked.

3.1.2 Board Quadrant

Bottom Left

3.1.3 PSoC Device

CY8C27643

3.1.4 Finished Project Location

`\PSoCExpressExamples\CY3209I2CSlaveLab\CY3209I2CSlaveLab.app` on the kit CD.

3.1.5 Start a New Project

1. Select **File** → **New Project**.
2. Name the project **CY3209I2CSlaveLab**.
3. If needed, click **Browse** to save the project in a different location.
4. Click **OK**.

3.1.6 Add an Input Driver

Table 3-1. 0 to 2600 mV Driver

Driver/Valuator	Unscaled Voltage - (0-2600mV)	
Location	Inputs Tab ↳ Voltage Input ↳ DC ↳ 0 to 2600 mV	
Properties	**Field**	**User Input**
	Name	Volts

Note: The potentiometers on the CY3209 board can generate up to 3300 mV. The voltage input driver used in this lab measures up to 2600 mV.

3.1.7 Add an Interface Valuator

Note: These steps must be performed in the Add Valuator dialog box that appears when the Discreate Interface Valuator is initially placed in the design.

Table 3-2. Discrete Interface Valuator

Driver/Valuator	Discrete	
Location	Valuators Tab ↳ Interface (Communication) Valuators ↳ Discrete	
Properties	**Field**	**User Input**
	Name	Command
	Default Value	0x0 (Default)
	States	(See Adding States Below)

Adding States:

a. Click in the **States** field. Select the ellipsis button to add values.

b. In the Add Values window, add the following names and expressions in the following table:

Note: After entering a name and expression, click **Add** to add the name and expression to the bottom of the list, or click **Insert** to insert the name and expression above the last selected name and expression in the list. The names and expressions must appear in the Add Values window in the order shown in the table:

Table 3-3. Command Interface Valuator States Expressions

Name	Expression
Low	0
Medium	1
High	2

c. Select the **High** check box to make it the default state.

d. Click **OK** to close the Add Values dialog box.

3.1.8 Add an Output Driver

Table 3-4. On/Off with Blink Driver

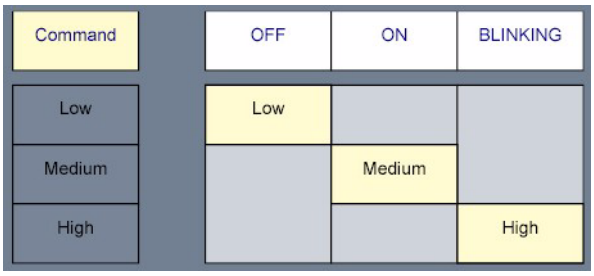
Driver/Valuator	Single LED - On-Off With Blink	
Location	Outputs Tab ↳ Display ↳ LED ↳ Single Color ↳ On/Off with blink	
Properties	**Field**	**User Input**
	Name	LED
	Initial Value	OFF (Default)
	BlinkRate	1 (Default)
	Current Mode	Sourcing (Default)

Note: An optional change to the driver properties is to change the BlinkRate.

3.1.9 Define Output Behavior

1. Right-click on the **LED** driver icon, and select **Transfer Function**.
2. Select **TableLookup** from the list of transfer functions and click **OK**.
3. Select **Command** in the Select Input(s) list.
4. Click **Next**.
5. Drag the Command Input values to the following LED state columns:
   - Low → OFF
   - Medium → ON
   - High → BLINKING

   Figure 3-1. CY3209I2CSlaveLab LED Output States



6. Click **OK**.

3.1.10 Add an Interface Driver

Table 3-5. Slave Driver

Driver/Valuator	I2C Slave Interface	
Location	Interfaces Tab ↳ Communication ↳ I2C ↳ Slave	
Properties	**Field**	**User Input**
	Name	I2Cs
	I2C_Address	5

3.1.11 Verify the Project Register Map

1. Select **Project → Assign Register Map for 'CY3209I2CSlaveLab' Project**.
2. Verify that the Read/Write section shows:
   - Command

   and that the Read-only section shows:
   - LED
   - Volts
3. If the values are not in the order shown, click on one of the values and use the Up/Down arrow buttons to move the value up or down until the values are in the order shown above.
4. Click **OK**.

3.1.12 Build the Project

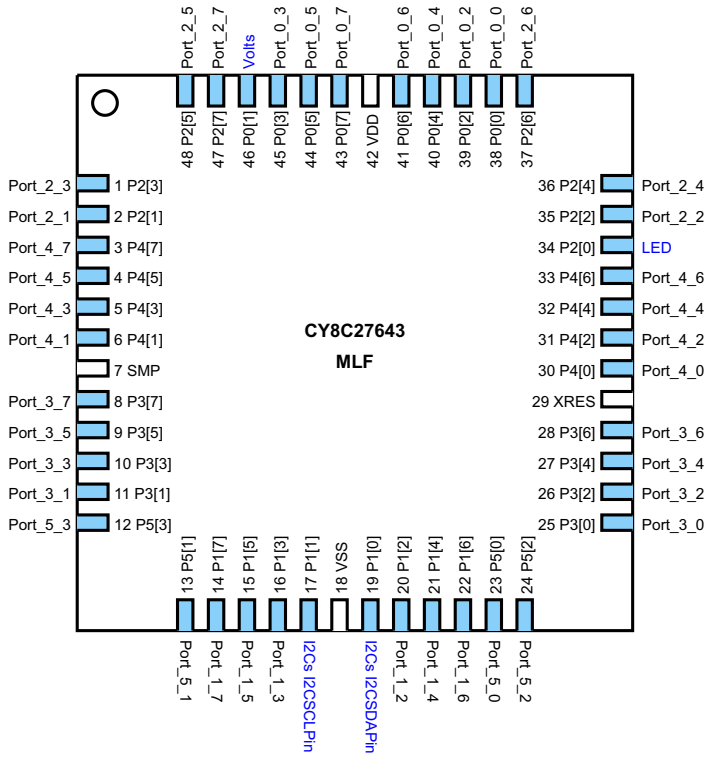
1. Select **Build → Generate\Build 'CY3209I2CSlaveLab' Project**.
2. In the Available Device Configurations pane, select **CY8C27000 → CY8C27643, 48-Pin MLF**.
3. In the Device Configurations Properties pane, set the following parameters:
   - Supply Voltage → 3.3V
   - Sample Rate → 64 Hz
4. Ensure the **Assign pins automatically** check box is **not** selected.
5. Click **Next**.
6. In the User Pin Assignment window, click **Unassign All Pins**.
7. Assign drivers to pins by dragging and dropping the blue rectangles from the Unassigned Drivers list to the pins listed in the following table.

Table 3-6. CY3209I2CSlaveLab Pinouts

Function	Pin
I2Cs I2CSCLPin	P1[1]
I2Cs I2CSDAPin	P1[0]
LED	P2[0]
Volts	P0[1]

Note: The pin label notation P1[2] indicates Port 1, Pin 2.

Figure 3-2. CY3209I2CSlaveLab Pinouts

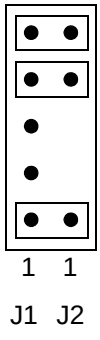


8. Click **Next**. PSoC Express generates and compiles the project source code.

3.1.13 Program the PSoC Device

1. Disconnect any power sources from the demonstration board.
2. Remove the jumpers between J1 and J2.
3. Program the part via J1 as described in the Programming Instructions for CY3209 PSoC Express Projects section on page 10.
4. Install jumpers on the following pin combinations:

Figure 3-3. CY3209I2CSlaveLab Jumper Settings



3.2 I2C Master/Slave Lab – Master

3.2.1 Description

This project is the master device in the I2C Master/Slave lab. The project reads a voltage value from the slave device over an I2C bus. The voltage value displays on the four digit LED display. A control value of 0, 1, or 2 is written to the slave device based on the voltage level. This value instructs the slave to set its LED to Off, On, or Blinking.

3.2.2 Board Quadrant

Bottom Right

3.2.3 PSoC Device

CY8C29666

3.2.4 Finished Project Location

`\PSoCExpressExamples\CY3209I2CMasterLab\CY3209I2CMasterLab.app` on the kit CD**.**

3.2.5 Start a New Project

1. Select **File** → **New Project**.
2. Name the project **CY3209I2CMasterLab**.
3. If needed, change the project location using **Browse** to move to a different folder.
4. Click **OK**.

3.2.6 Add an Input Driver

Table 3-7. External I2C Slave-Monitor Driver

Driver/Valuator	External I2C Slave Device - Monitor	
Location	Inputs Tab ↳ Remote Devices ↳ External I2C Slave-Monitor	
Properties	**Field**	**User Input**
	Name	VoltsMonitor
	I2C_Address	5
	I2C_SubAddress	2
	Variable_Type	INT (Default)

Note: The I2C_SubAddress property value is determined by the layout of the I2C slave project Interface Register Map.

3.2.7 Add Two Output Drivers

Table 3-8. Output Driver 1: Common Cathode Driver

Driver/Valuator	Four Digit LED, Common Cathode	
Location	Outputs Tab ↳ Display ↳ LED ↳ 7-Segment ↳ Multi Digit ↳ Four Digit with Decimal Point ↳ Common Cathode	
Properties	**Field**	**User Input**
	Name	FourDigLED
	Initial Value	0 (Default)
	Position of decimal point	NONE (Default)

Table 3-9. Output Driver 2: External I2C Slave-Control Driver

Driver/Valuator	External I2C Slave Device - Control	
Location	Outputs Tab ↳ Remote Devices ↳ External I2C Slave-Control	
Properties	**Field**	**User Input**
	Name	LevelControl
	Initial Value	0 (Default)
	I2C_Address	5
	I2C_SubAddress	0 (Default)
	VariableType	BYTE
	WriteMode	OnChange (Default)
	WriteInitalValue	No (Default)

Note: The I2C_SubAddress property value is determined by the layout of the I2C slave project Interface Register Map.

3.2.8 Define Output Behavior

3.2.8.1 FourDigLED

1. Right-click on the FourDigLED driver icon and select **Transfer Function**.
2. Select **StatusEncoder** from the list of transfer functions and click **OK**.
3. Enter rules for the StatusEncoder into the If/Then fields according to the following table.

Table 3-10. FourDigLED StatusEncoder If/Then Settings

If	Then
1	VoltsMonitor

4. Click **OK**.

3.2.8.2 LevelControl

1. Right-click on the LevelControl driver icon and select **Transfer Function**.
2. Select **PriorityEncoder** from the list of transfer functions and click **OK**.
3. Enter rules for the PriorityEncoder according to Table 3-11. To add a row to the PriorityEncoder, click the down arrow button as you complete each if/then statement.

Table 3-11. Level Control PriorityEncoder If/Then Settings

If/Else If	Then
VoltsMonitor < 1000	0
VoltsMonitor < 2000	1
1	2

4. Click **OK**.

3.2.9 Build the Project

1. Select **Build → Generate/Build 'CY3209I2CMasterLab' Project**.
2. In the Available Device Configurations pane, select **CY8C29000\CY8C29666, 48-Pin MLF**.
3. In the Device Configurations Properties pane, set the following parameters:
   - Supply Voltage → 3.3V
   - Sample Rate → Free Run
4. Ensure the **Assign pins automatically** check box is **not** selected.
5. Click **Next**.
6. In the User Pin Assignment window, click **Unassign All Pins**.

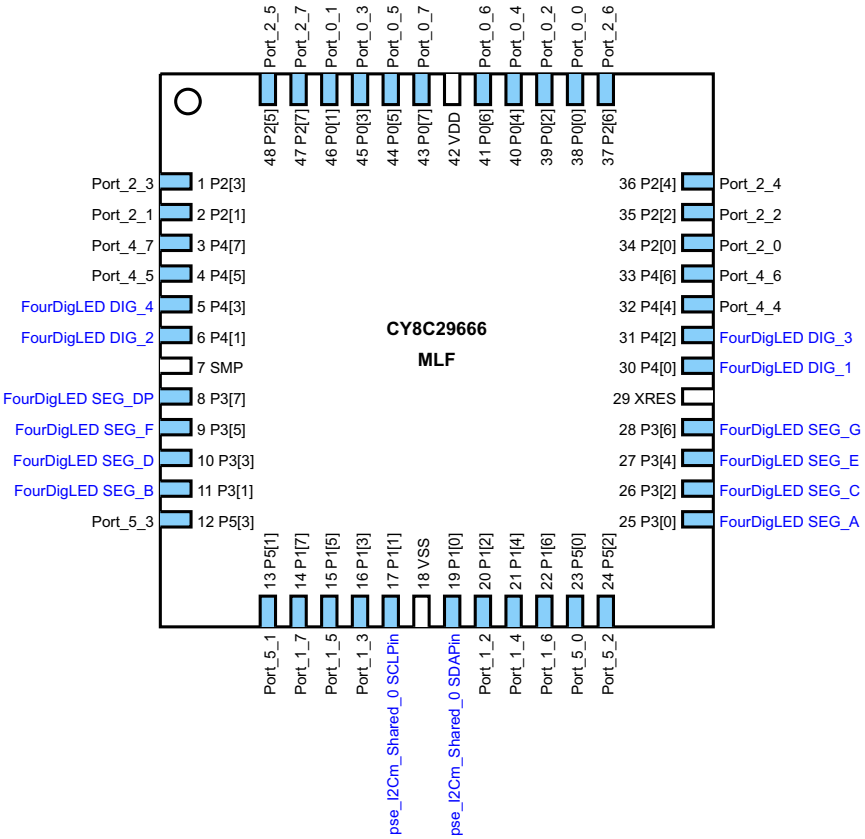
7. Assign drivers to pins by dragging and dropping the blue rectangles from the Unassigned Drivers list to the pins listed in the following table.

Table 3-12. CY3209I2CMasterLab User Pin Assignments

Function	Pin
FourDigLED SEG_A	P3[0]
FourDigLED SEG_B	P3[1]
FourDigLED SEG_C	P3[2]
FourDigLED SEG_D	P3[3]
FourDigLED SEG_E	P3[4]
FourDigLED SEG_F	P3[5]
FourDigLED SEG_G	P3[6]
FourDigLED SEG_DP	P3[7]
FourDigLED DIG_1	P4[0]
FourDigLED DIG_2	P4[1]
FourDigLED DIG_3	P4[2]
FourDigLED DIG_4	P4[3]
pse_I2Cm_Shared_0 SDAPin	P1[0]
pse_I2Cm_Shared_0 SCLPin	P1[1]

Note: The pin label notation P1[2] indicates Port 1, Pin 2.

Figure 3-4. CY3209I2CMasterLab Pinouts

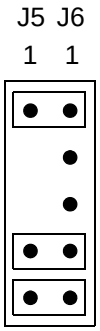


8. Click **Next**. PSoC Express generates and compiles the project source code.

3.2.10 Program the PSoC Device

1. Disconnect any power sources from the demonstration board.
2. Remove the jumpers between J5 and J6.
3. Program the part via J6 as described in Programming Instructions for CY3209 PSoC Express Projects section on page 10.
4. Install jumpers on the following pin combinations:

Figure 3-5. CY3209I2CMasterLab Jumper Settings



3.2.11 Test Your Project on the Demonstration Board

1. Remove any jumpers on J7, J8, J12, and J13.
2. Ensure the slave project has been programmed into the bottom left quadrant of the demonstration board and that the jumpers specified in Build the Project on page 20 have been installed. Power the board with a 9V battery (provided with the kit) or an optional 6–14V DC wall transformer.
3. Adjust the potentiometer on the slave (bottom left quadrant). Observe the master's multi-digit LED display and the slave's LED.
4. Observe the following relationship between the multi-digit LED display on the master and the single LED on the slave:
   - 0-999 → LED is Off
   - 1000-1999 → LED is On
   - >2000 → LED is Blinking

4. USB-UART Lab

4.1 Description

This project uses the USB-UART driver to demonstrate the ability to monitor and control a PSoC device with a PC using a virtual COM port on a USB hardware connection. The project has a voltage input and an LED that supports three states: Off, On, and Blinking. The LED state is controlled by an Interface Valuator.

Note: This project will not work unless the Add Driver Dialog option is selected. To use the Add Driver Dialog, ensure **View** → **Use Add Driver Dialog** is checked.

4.2 Board Quadrant

Top Right

4.3 PSoC Device

CY8C24894

4.4 Finished Project Location

`\PSoCExpressExamples\CY3209USBUARTLab\CY3209USBUARTLab.app` on the kit CD.

4.5 Start a New Project

1. Select **File** → **New Project**.
2. Name the project **CY3209USBUARTLab**.
3. If needed, click **Browse** to save the project in a different location.
4. Click **OK**.

4.6 Add an Input Driver

Table 4-1. 0 to 2600 mV Driver

Driver/Valuator	Unscaled Voltage - (0-2600mV)	
Location	Inputs Tab ↳ Voltage Input ↳ DC ↳ 0 to 2600 mV	
Properties	**Field**	**User Input**
	Name	Volts

Note: The potentiometers on the CY3209 board can generate up to 3300 mV. The voltage input driver used in this lab measures up to 2600 mV.

4.7 Add an Interface Valuator

Note: These steps must be performed in the Add Valuator dialog box that appears when the Discreate Interface Valuator is initially placed in the design.

Table 4-2. Discrete Interface Valuator

Driver/Valuator	Discrete	
Location	Valuators Tab ↳ Interface (Communication) Valuators ↳ Discrete	
Properties	**Field**	**User Input**
	Name	Command
	Default Value	0x0 (Default)
	States	(See Adding States Below)

Adding States:

a. Click in the **States** field. Select the ellipsis button to add values.

b. In the Add Values window, add the following names and expressions in the following table:

Note: After entering a name and expression, click **Add** to add the name and expression to the bottom of the list, or click **Insert** to insert the name and expression above the last selected name and expression in the list. The names and expressions must appear in the Add Values window in the order shown in the table:

Table 4-3. Command Interface Valuator States Expressions

Name	Expression
Low	0
Medium	1
High	2

c. Select the **High** check box to make it the default state.

d. Click **OK** to close the Add Values dialog box.

4.8 Add an Output Driver

Table 4-4. On/Off with Blink Driver

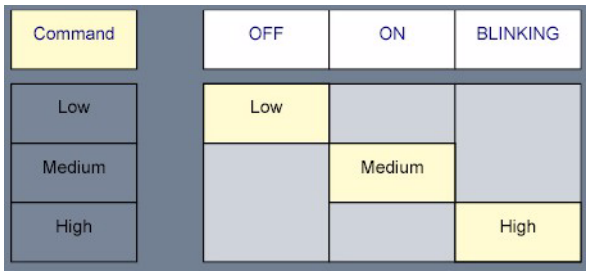
Driver/Valuator	Single LED - On-Off With Blink		
Location	Outputs Tab ↳ Display ↳ LED ↳ Single Color ↳ On/Off with blink		
Properties	**Field**	**User Input**	
	Name	LED	
	Initial Value	OFF	(Default)
	BlinkRate	1	(Default)
	Current Mode	Sourcing	(Default)

Note: An optional change to the driver properties is to change the BlinkRate.

4.9 Define Output Behavior

1. Right-click on the **LED** driver icon, and select **Transfer Function**.
2. Select **TableLookup** from the list of transfer functions and click **OK**.
3. Select **Command** in the Select Input(s) list.
4. Click **Next**.
5. Drag the Command Input values to the following LED state columns:
   - Low → OFF
   - Medium → ON
   - High → BLINKING

   Figure 4-1. CY3209USBUARTLab Output States



6. Click **OK**.

4.10 Add an Interface Driver

Table 4-5. USB-UART Driver

Driver/Valuator	USB-UART		
Location	Interfaces Tab ↳ Communication ↳ USB ↳ USB-UART		
Properties	**Field**	**User Input**	
	Name	USBUART	
	Vendor ID	1204	(Default)
	Product ID	57354	(Default)
	Vendor String	CypressCorp	(Default)
	Serial Number String	USBUART	(Default)
	Device Power	Self Powered	
	Max Power	500	

4.11 Verify the Project Register Map

1. Select **Project → Assign Register Map for 'CY3209I2CSlaveLab' Project**.
2. Verify that the Read/Write section shows:
   - Command

   and that the Read-only section shows:
   - LED
   - Volts
3. If the values are not in the order shown, click on one of the values and use the Up/Down arrow buttons to move the value up or down until the values are in the order shown above.
4. Click **OK**.

4.12 Build the Project

1. Select **Build → Generate/Build 'CY3209USBUARTLab' Project**.
2. In the Available Device Configuration pane, select the PSoC device: **CY8C24090 → CY8C24894, 56-Pin MLF**.
3. In the Device Configurations Properties pane, set the following parameters:
   - Supply Voltage → 3.3V
   - Sample Rate to → 64 Hz
4. Ensure the **Assign pins automatically** check box is **not** selected.
5. Click **Next**.
6. In the User Pin Assignment window, click **Unassign All Pins**.

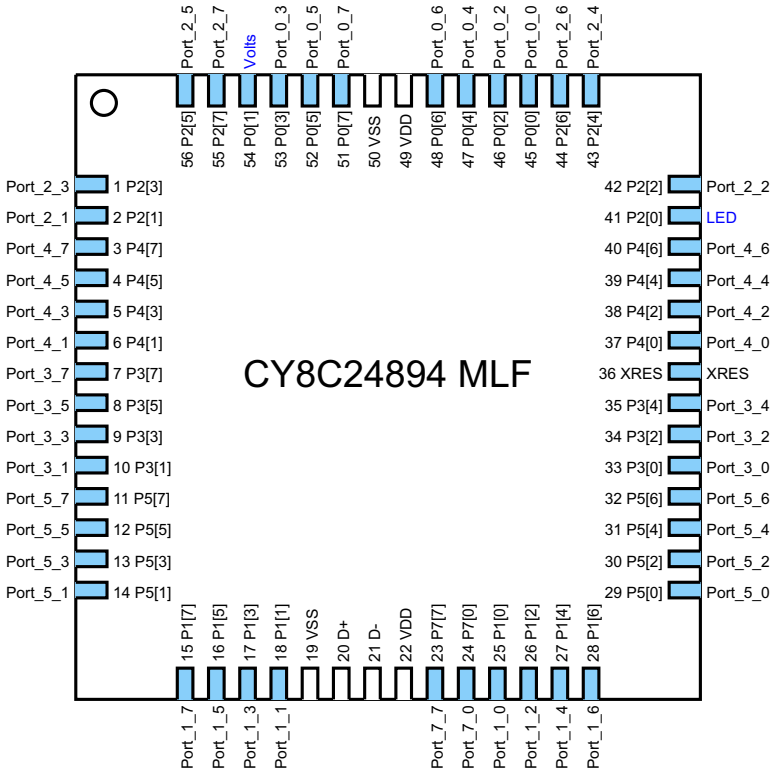
7. Assign drivers to pins by dragging and dropping the blue rectangles from the Unassigned Drivers list to the pins listed in the following table.

Table 4-6. CY3209USBUARTLab Pinouts

Function	Pin
LED	P2[0]
Volts	P0[1]

Note: The pin label notation P1[2] indicates Port 1, Pin 2. USB connections are on dedicated pins and are not assignable.

Figure 4-2. CY3209USBUARTLab Pinouts

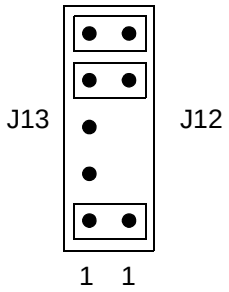


8. Click **Next**. PSoC Express generates and compiles the source code.

4.13 Program the PSoC Device

1. Disconnect any power sources from the demonstration board.
2. Remove the jumpers between J12 and J13.
3. Program the part via J13 as described in Programming Instructions for CY3209 PSoC Express Projects section on page 10.
4. Install jumpers on the following pin combinations:

Figure 4-3. CY3209USBUARTLab Jumper Settings



4.14 Test Your Project on the Demonstration Board

1. Power the board with a 9V battery (provided with the kit) or an optional 6–14V DC wall transformer.
2. Plug the USB cable into the mini USB connector in the top right quadrant of the board.
3. The Microsoft Windows New Hardware Wizard automatically starts. In the New Hardware Wizard dialog box:
   a. Click the **No, not this time** option button.
   b. Click **Next**.
   c. Click the **Install from a list or specific location (Advanced)** option button.
   d. Click **Next**.
   e. Click the **Search for the best driver in these locations** option button.
   f. Do not select the **Search removable media (floppy, CD-ROM...)** check box.
   g. Click the **Include this location in the search** check box.
   h. Click **Browse**.
   i. Navigate to where you saved the USBUART Lab project.
   j. Select the folder: `CY3209USBUARTLab\CY3209USBUARTLab\CY3209USBUARTLab\lib`.
   k. Click **OK**.
   l. Click **Next**.
   m. Click **Continue Anyway** in the Hardware Installation warning dialog.
   n. Click **Finish**.
   o. Open the Windows Device Manager.
   p. Expand the Ports (COM&LPT) branch of the device tree. Record the COM number assigned to USBUART.

4. HyperTerminal Configuration

   Note: These instructions are for a Windows XP installation, which includes HyperTerminal. For other Microsoft Windows versions, the location of HyperTerminal may differ. For Windows installations that do not include Hyperterminal, a similar serial terminal program may be used. In this case, the user must interpret these instructions as needed to set the appropriate configurations.

   a. Open HyperTerminal by going to **Start** → **All Programs** → **Accessories** → **Communications** and selecting **HyperTerminal** (Windows XP only; the location may be different in other versions of Microsoft Windows).
   b. Create a name for a terminal configuration file, select an icon for the connection, and click **OK**.
   c. Select the COM port number recorded in Step 3.p. on page 32.
   d. Click **OK**.
   e. In the Flow Control list, select **None**.
   f. Click **OK**.
   g. In the Hyperterminal window, select **File** → **Properties**.
   h. Select the **Settings** tab.
   i. Click **ASCII Setup**.
   j. Select the **Echo typed characters locally** check box.
   k. Ensure that no other check boxes are selected.
   l. Click **OK** to close the ASCII Setup dialog box.
   m. Click **OK** to close the Properties window.

5. Communicate with the USB-UART

   a. USB-UART Command Set

      - **WR** <register address> [byte_1] … [byte_n] - write to USB device
      - **RD** <count> - read from USB device
      - **CRD** <count> - continuous read from USB device

      Notes:

      - Command values must be in UPPER CASE.
      - Items in boldface must be typed verbatim.
      - Items not in boldface must be replaced with appropriate values.
      - Parameters enclosed in angle brackets <> are required for a command to successfully execute.
      - Parameters enclosed in square brackets [ ] are optional.
      - All parameters are separated from the initial command and the other parameters by spaces.
      - Data is formatted as 2 character hexadecimal values (i.e. 00, 1A).
      - All command lines are terminated by a linefeed or carriage return/linefeed combination to be executed.

- The WR command's register address parameter sets the read/write offset pointer. All byte values following the register address parameter in a WR command are written starting at the location specified by the register address parameter. Subsequent RD or CRD commands will start reading from the last register offset pointer location set by a WR command.
- The register address value used in the WR command is dependent on the Interface Register Map of your project. This information is provided in Table 2 of the DataSheet project report.

b. Write a Value to the PSoC to Control the LED
   i. Enter **WR 00 02** and press [**Enter**]. This action writes a value of 2 to register offset 0. The LED blinks.
   ii. Enter **WR 00 00** press [**Enter**]. This action writes a value of 0 to register offset 0. The LED switches off.
   iii. Enter **WR 00 01** press [**Enter**]. This action writes a value of 1 to register offset 0. The LED switches on.

c. Read the Volts Value from the PSoC
   i. Enter **WR 02** and press [**Enter**]. This action sets the register offset pointer to point to the Volts value (register offset 2).
   ii. Enter **RD 02** and press [**Enter**]. This action reads two bytes from the PSoC. The current value of the Volts inputs displays. The value displays as two HEX digits, that is, 1000 mV displays as 03 E8.

d. Continuously Read Values from the PSoC
   i. Enter **WR 00** and press [**Enter**]. This action sets the register offset pointer to 0.
   ii. Enter **CRD 04** and press [**Enter**]. This action initiates a continuous reading of four bytes from the PSoC. Four HEX values are displayed (each value displays as two HEX digits). The data values from left to right are:
      - Command value (last value written to the PSoC)
      - Current value of the LED outputs (this matches the command value)
      - High order byte of the Volts 16-bit value
      - Low order byte of the Volts 16-bit value
   iii. Adjust the potentiometer in the top right quadrant. The Volt values in the HyperTerminal display vary.
   iv. Press [**Enter**] to terminate the data stream from the PSoC.

5. CapSense Lab

5.1 Description

This project creates four capacitance sensing buttons that individually control four LEDs.

For detailed information on CapSense projects, see the PSoC Express CapSense guide by going to **Help** → **Documentation**, and selecting **CapSenseGuide**.

5.2 Board Quadrant

Top Left

5.3 PSoC Device

CY8C21434

5.4 Finished Project Location

`\PSoCExpressExamples\CY3209CapSenseLabCSD\CY3209CapSenseLabCSD.app` on the kit CD.

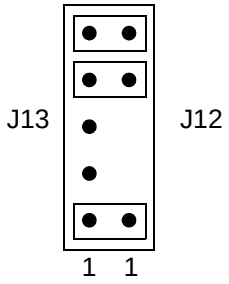
5.5 Program I2C-USB Bridge Firmware

This enables PC communication with the PSoC device.

1. Disconnect the demonstration board from its power supply.
2. Remove the jumpers between J12 and J13 (top right quadrant).
3. Connect the MiniProg to the PC using a USB cable.
4. Connect the MiniProg to J13.
5. Launch PSoC Programmer from the **Windows Start Menu** → **All Programs** → **Cypress MicroSystems** → **PSoC Programming**.
6. In PSoC Programmer:
   a. Select the **MiniProg** from the Port list.
   b. Select **24x94** in the Device Family list.
   c. Select **CY8C24894-24LFXI** in the Device list.
   d. Select the **Power Cycle** option button.
   e. Click the **File Load** button in the toolbar. Browse the CY3209 kit CD to find and select *I2C-USB Bridge.hex*.
   f. Click the **Program** button in the toolbar.
   g. Wait until the Actions column of PSoC Programmer's status window indicates that either Programming Succeeded or Programming Terminated. If the Programming Terminated message appears, troubleshoot the problem, and re-attempt to program the device.
   h. Exit PSoC Programmer.

7. Disconnect the MiniProg from the CY3209 board.
8. Install jumpers on the following pin combinations:

Figure 5-1. CY8C24894-24LFXI Jumper Settings



5.6 Start a New Project

1. Select **File** → **New Project**.
2. Name the project **CY3209CapSenseLabCSD**.
3. If needed, click **Browse** to save the project in a different location.
4. Click **OK**.

5.7 Add an Interface Driver

Table 5-1. Slave Driver

Driver/Valuator	I2C Slave Interface	
Location	Interfaces Tab ↳ Communication ↳ I2C ↳ Slave	
Properties	**Field**	**User Input**
	Name	I2C
	I2C_Address	4 (Default)

5.8 Add an Output Driver

Table 5-2. On/Off Driver

Driver/Valuator	Single LED - On-Off	
Location	Outputs Tab ↳ Display ↳ LED ↳ Single Color ↳ On/Off	
Properties	**Field**	**User Input**
	Name	LED1
	Initial Value	OFF (Default)
	Current Mode	Sourcing (Default)

5.9 Add an Input Driver

Note: PSoC Express automatically adds a CapSense Properties driver the first time you add a CapSense button, slider, or proximity detector to your design. When you click OK to close the Add Input Driver window for the button, slider or proximity sensor, a new Add Input Driver window opens for the Properties driver. Enter the required fields (listed below) just like you would for any other driver.

Table 5-3. Button - CSD Driver

Driver/Valuator	CapSense Button – CSD (Capacitive Sensing – Sigma-Delta Modulator) Algorithm **Note**: When you click OK to close the Add Input Driver window, a new Add Input Driver window for the Properties driver opens.		
Location	Inputs Tab ↳ CapSense ↳ CapSense - CSD ↳ Button - CSD ↳ Properties - CSD (*added automatically*)		
Properties	**Field**	**User Input**	
	Name	Button1	
	Property Editor	CSDDriverPack-age.dll	(Default)
	Finger Threshold	100	(Default)
	Scan Speed	Normal	(Default)
	Scanning Resolution	12	(Default)
	Ref Value	4	(Default)
	Expose Tuning Values	Yes	(Default)

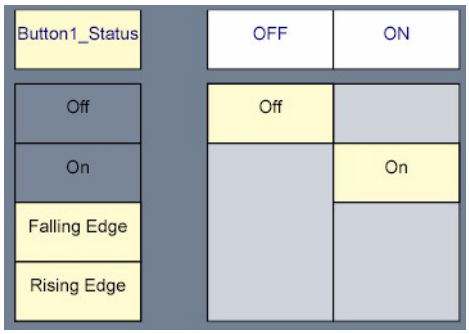
Table 5-4. Properties - CSD Driver

Driver/Valuator	CapSense Properties – CSD (Capacitive Sensing using a Sigma-Delta Modulator) Algorithm		
Location	Automatically added the first time a CapSense button, slider, or proximity detector is added to your design		
Properties	**Field**	**User Input**	
	Name	CSProperties	
	NoiseThreshold	40	(Default)
	NegativeNoiseThreshold	20	(Default)
	BaselineUpdateThreshold	200	(Default)
	Hysteresis	10	(Default)
	Debounce	3	(Default)
	LowBaselineReset	50	(Default)
	Sensors Autoreset	Disabled	

5.10 Define Output Behavior

1. Right-click on the LED1 driver, and select **Transfer Function**.
2. Select **TableLookup** from the list of transfer functions and click **OK**.
3. Select **Button1_Status** from the Select Input(s) list.
4. Click **Next**.
5. Drag the Button1_Status values to the following LED state columns:
   - Off → OFF
   - On → ON

   Figure 5-2. LED Output States

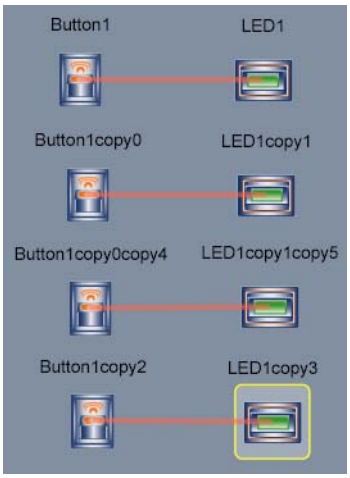


6. Click **OK**.

5.11 Duplicate the Button and LED

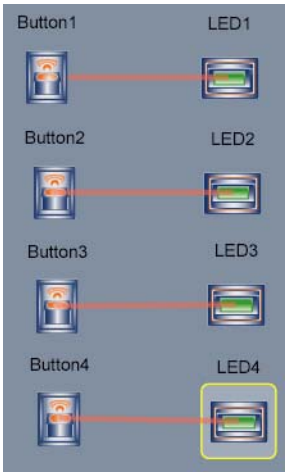
1. Click on the **Group Selection** tool on the top right of the tool bar (small dotted box).
2. Click on the **Button1** icon.
3. Click on the **LED1** icon.
4. Right-click on one of the selected driver icons and select **Duplicate Objects**. There are now two buttons and two LEDs.
5. Right-click on one of the selected driver icons and select **Duplicate Objects.** You now have four buttons and four LEDs.
6. Click in an open area on the desktop to deselect the driver icons.
7. Drag the button and LED icons on the design desktop to make a neat arrangement of button/LED pairs. Figure 5-3 is a good example.

Figure 5-3. Button and LED Arrangement Example



8. Click on the **Group Selection** tool on the top right tool bar (small dotted box).
9. Click on each of the duplicated button driver icons.
10. Right-click on one of the selected button driver icons and select **Rename Objects**.
11. Name the drivers **Button2**, **Button3**, and **Button4**.
12. Click **OK**.
13. Using the **Group Selection** tool, select the three new LED driver icons.
14. Right-click on one of the selected button driver icons and select **Rename Objects**.
15. Name the drivers **LED2**, **LED3**, and **LED4**, ensuring that LED2 is the driver connected to Button2, LED3 is the driver connected to Button3, and LED4 is the driver connected to Button4.

Figure 5-4. Buttons and LEDs Named and Connected Example



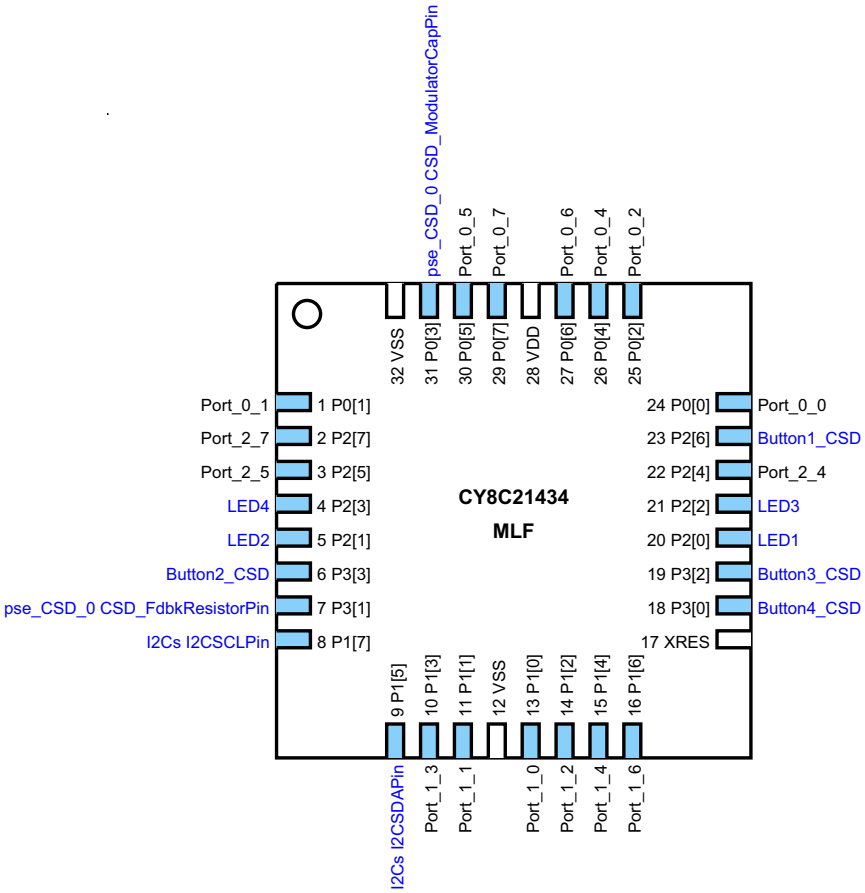
5.12 Build the Project

1. Select **Build → Generate/Build 'CY3209CapSenseLabCSD' Project**.
2. In the Available Device Configurations pane, select **CY8C21030 → CY8C21434, 32-Pin**.
3. In the Device Configurations Properties pane, set the following parameters:
   - Supply Voltage → 3.3V
   - Sample Rate → 64 Hz
4. Ensure the **Assign pins automatically** check box is **not** selected.
5. Click **Next**.
6. In the User Pin Assignment window, click **Unassign All Pins**.
7. Assign drivers to pins by dragging and dropping the blue rectangles from the Unassigned Drivers list to the pins listed in the following table.

Table 5-5. CY3209CapSenseLabCSD Pinouts

Function	Pin
Button1_CSD	P2[6]
Button2_CSD	P3[3]
Button3_CSD	P3[2]
Button4_CSD	P3[0]
pse_CSD_0 CSD_FdbkResistorPin	P3[1]
pse_CSD_0 CSD_ModulatorCapPin	P0[3]
I2Cs I2CSCLPin	P1[7]
I2Cs I2CSDAPin	P1[5]
LED1	P2[0]
LED2	P2[1]
LED3	P2[2]
LED4	P2[3]

Figure 5-5. CY3209CapSenseLabCSD Pinouts

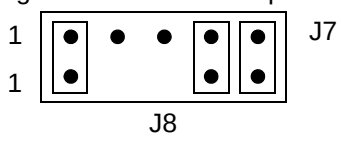


8. Click **Next**. PSoC Express generates and compiles the project source code.

5.13 Program the PSoC Device

1. Disconnect any power sources from the demonstration board, and connect the MiniProg to your PC using a USB cable.
2. Remove the jumpers between J7 and J8.
3. Program the part via J7 as described in the Programming Instructions for CY3209 PSoC Express Projects section on page 10.
4. Install jumpers on the following pin combinations:

Figure 5-6. CY3209CapSenseLabCSD Jumper Settings



5.14 Test Your Project on the Demonstration Board

1. Power the board with a 9V battery (provided with the kit) or an optional 6–14V DC wall transformer.
2. Place your finger on each of the CapSense buttons on the demonstration board. As each button is touched, its associated LED lights up.

5.15 Tuning Your Project

The goal of tuning a capacitive sensing application is to get the sensitivity of the buttons adjusted so that they accurately detect finger presses. This involves determining which raw counts coming from the sensor are actual finger presses or some other stimuli that changes the raw count. You need to achieve a signal to noise ratio of 5:1 or better to reliably determine a range for valid finger presses while rejecting transient noise caused by environmental and other factors. For information about this topic, see application note AN2403, *Capacitance Sensing - Signal-to-Noise Ratio Requirement for CapSense Applications.*

1. Click the **Monitor** button 

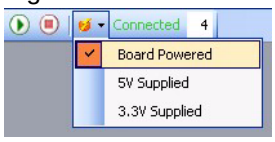
.
2. Connect the CapSense test board to your computer using the USB-I2C bridge and a USB cable.

   The Monitor toolbar will change from Not Connected 

, to Connected 

.
3. Select **Board Powered** from the Power Selection menu.

Figure 5-7. Power Selection Menu

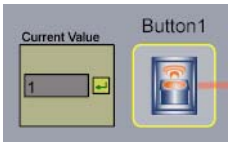


4. Click the **Play** 
, button. The Monitor toolbar will change from Connected 

, to Running 

.
5. Touch Button1 on you CapSense test board. Note how the Current Value for Button1 changes as you touch the button several times. Experiment by touching the other buttons and the slider.

Figure 5-8. Monitoring Button Values



6. Right-click **Button1** and select **Show Tuner** (Figure 5-9) to open the CSD Button Tuner Window (Figure 5-10).

Figure 5-9. Show Tuner Menu

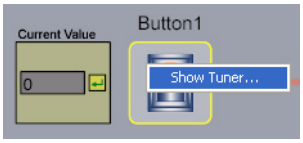
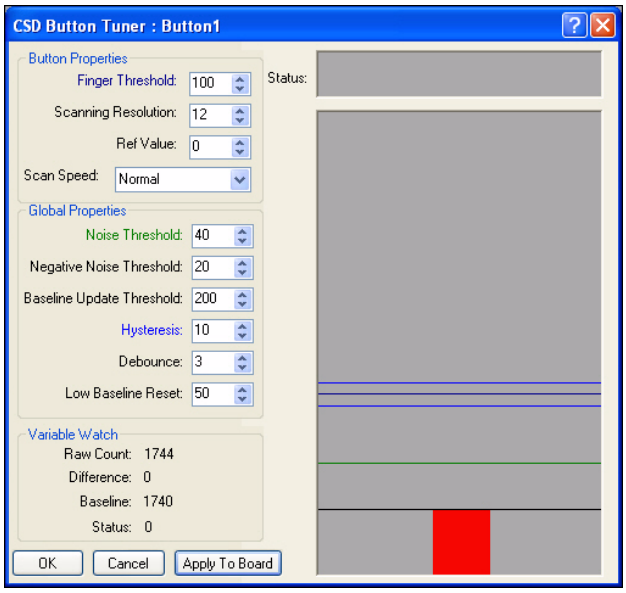
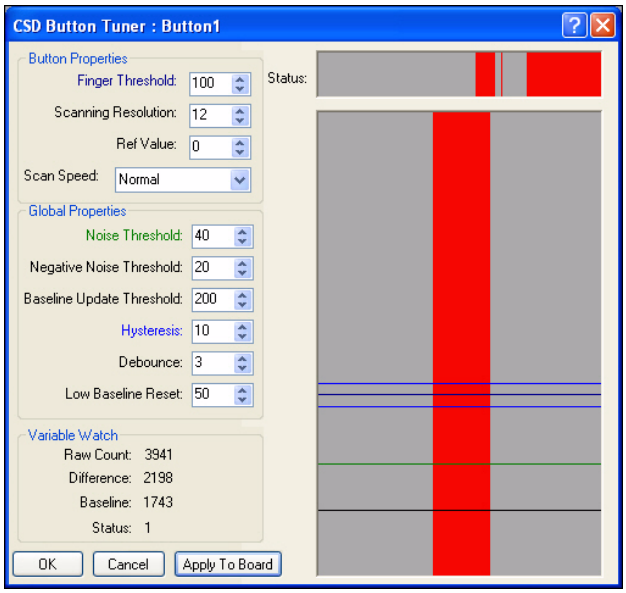


Figure 5-10. CSD Button Tuner Window



7. Touch the button on the board that corresponds to driver tuner displayed. The button is more sensitive than necessary.

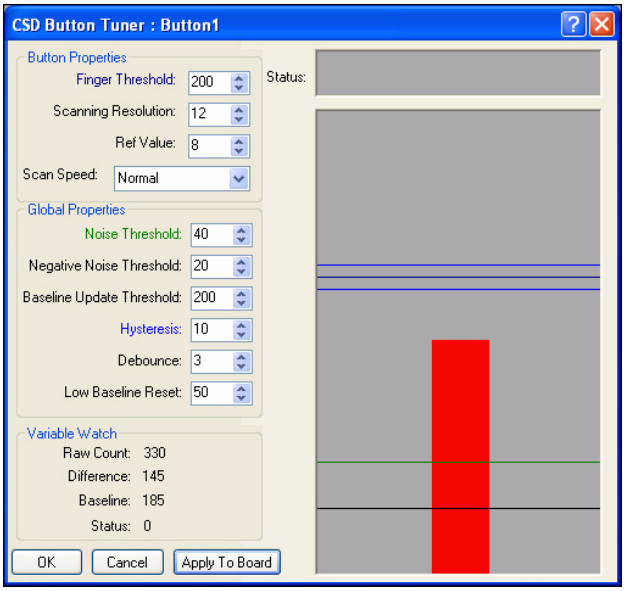
Figure 5-11. Finger Response to a Button Press



8. Increase the Finger Threshold to **200** and set the Ref Value to **8**. Changing the Finger Threshold changes where the button transitions from on to off. An increased Ref Value level will decrease sensor sensitivity, but increase the influence of the shielding electrode. See the driver datasheet for more information about this setting.
9. Click **Apply to Board** to write the changed parameters to Flash on the PSoC device.

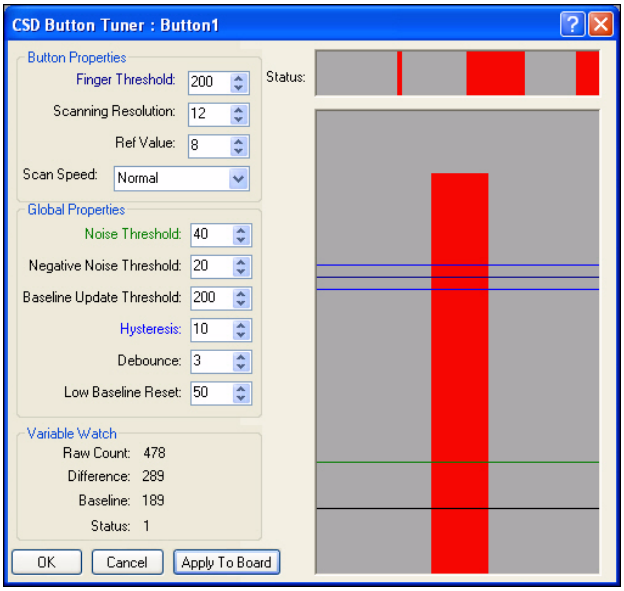
10. To see exactly where a button triggers, move your finger slowly on to the button. Do this to observe that the Finger Threshold is higher than it was previously.

Figure 5-12. Below the New Finger Threshold



11. Place a finger fully on the button to observe that it is now less sensitive.

Figure 5-13. Less Sensitive to the Touch



12. Click **OK** to save the values from the tuner window to the properties in your project, or click **Cancel** to discard your changes, and retain the original properties.

Congratulations! You are now done with the tutorial. Close the tuner window and stop the board monitor. Anytime that you are not actively using the board monitor you should stop it. Monitoring a design consumes a lot of processor cycles on your PC.

6. LCD Lab

6.1 Description

This project displays a voltage input as a numeric value and as a bar graph on an LCD display. Although, the potentiometer used to generate the voltage is wired to supply up to 3300 mV to the PSoC input pin, the firmware in this lab indicates the input voltage up to 2600 mV.

6.2 Board Quadrant

Bottom Left

6.3 PSoC Device

CY8C27643

6.4 Finished Project Location

`\PSoCExpressExamples\CY3209LCDLab\CY3209LCDLab.app` on the kit CD.

6.5 Start a New Project

1. Select **File** → **New Project**.
2. Name the project **CY3209LCDLab**.
3. If needed, click **Browse** to save the project in a different location.
4. Click **OK**.

6.6 Add an Input Driver

Table 6-1. 0 to 2600 mV Driver

Driver/Valuator	Unscaled Voltage - (0-2600mV)	
Location	Inputs Tab ↳ Voltage Input ↳ DC ↳ 0 to 2600 mV	
Properties	**Field**	**User Input**
	Name	Volts

Note: The potentiometers on the CY3209 board can generate up to 3300 mV. The voltage input driver used in this lab measures up to 2600 mV.

6.7 Add an Output Driver

Table 6-2. LCD Horizontal Bar Graph with Value Driver

Driver/Valuator	LCD Horizontal Bar Graph with Value		
Location	Outputs Tab ↳ Display ↳ LCD ↳ LCD Horizontal Bar Graph with Value		
Properties	**Field**	**User Input**	
	Name	BarGraph	
	Initial Value	0	
	Row Location	Rows 0-1	(Default)
	Min Value	0	(Default)
	Max Value	2600	
	Units	mV	

6.8 Define Output Behavior

1. Right-click on the **BarGraph** driver icon and select **Transfer Function**.
2. Select **PriorityEncoder** from the list of transfer functions and click **OK**.
3. Enter the following rule for the PriorityEncoder:

Table 6-3. BarGraph PriorityEncoder If/Then Settings

If/Else If	Then
1	Volts

4. Click **OK**.

6.9 Build the Project

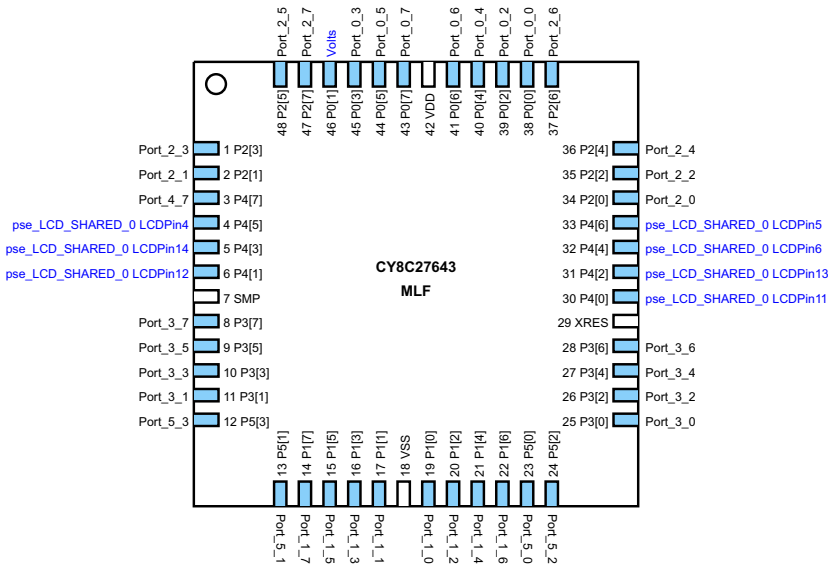
1. Select **Build → Build/Generate 'CY3209LCDLab' Project**.
2. In the Available Device Configurations pane, select **CY8C27000 → CY8C27643, 48-Pin MLF**.
3. In the Device Configurations Properties pane, set the following driver properties:
   - Supply Voltage → 3.3V
   - Sample Rate → 64 Hz
4. Ensure the **Assign pins automatically** check box is **not** selected.
5. Click **Next**.
6. In the User Pin Assignment window, click **Unassign All Pins**.

7. Assign drivers to pins by dragging and dropping the blue rectangles from the Unassigned Drivers list to the pins listed in the following table.

Table 6-4. CY3209LCDLab Pinouts

Function	Pin
pse_LCD_SHARED_0 LCDPin11	P4[0]
pse_LCD_SHARED_0 LCDPin12	P4[1]
pse_LCD_SHARED_0 LCDPin13	P4[2]
pse_LCD_SHARED_0 LCDPin14	P4[3]
pse_LCD_SHARED_0 LCDPin6	P4[4]
pse_LCD_SHARED_0 LCDPin4	P4[5]
pse_LCD_SHARED_0 LCDPin5	P4[6]
Volts	P0[1]

Figure 6-1. CY3209LCDLab Pinouts

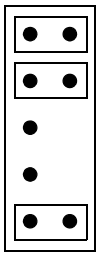


8. Click **Next**. PSoC Express generates and compiles the project source code.

6.10 Program the PSoC Device

1. Disconnect any power sources from the demonstration board.
2. Remove the jumpers between J1 and J2.
3. Program the part via J1 as described in the Programming Instructions for CY3209 PSoC Express Projects section on page 10.
4. Install jumpers on the following pin combinations:

Figure 6-2. CY3209LCDLab Jumper Settings



6.11 Test Your Project on the Demonstration Board

1. Install the LCD on P1 (bottom left quadrant). Align Pin 1 of the LCD with P1 Pin 1 (left side of P1).
2. When the LCD is used for the first time, it might be necessary to adjust the contrast for the LCD using the orange LCD Contrast potentiometer (R16). This potentiometer is located above the LCD connector within the bottom left quadrant. The potentiometer can be adjusted with an phillips screw driver. Turning the potentiometer clockwise increases the contrast and turning the potentiometer counter-clockwise decreases the contrast.
3. Power the board with a 9V battery (provided with the kit) or an optional 6–14V DC wall transformer.
4. Adjust the potentiometer in the bottom left quadrant. The voltage appears as a numeric value, and the bar graph sizes itself based on the ratio of the current voltage to 2600 mV.

7. Cypress WirelessUSB™ Master/Slave Lab 1

This lab creates two projects: a WirelessUSB slave and a WirelessUSB master. The WirelessUSB master device reads a voltage value from the slave and displays the value on a four digit LED display. The master also writes a control value to the slave device to set an LED to Off, On, or Blinking depending on the voltage level read from the slave.

7.1 WirelessUSB Master/Slave Lab – Slave

7.1.1 Description

This project is the slave device for the WirelessUSB Master/Slave lab. The project has a voltage input and an LED that supports three states: Off, On and Blinking. The LED state is controlled by the WirelessUSB master project through an Interface Valuator. The project becomes a WirelessUSB slave by adding a WirelessUSB Slave Interface driver.

Note: This project will not work unless the Add Driver Dialog option is selected. To use the Add Driver Dialog, ensure **View** → **Use Add Driver Dialog** is checked.

7.1.2 Board Quadrant

Bottom Left

7.1.3 PSoC Device

CY8C27643

7.1.4 Finished Project Location

`\PSoCExpressExamples\CY3209WUSBSlaveLab\CY3209WUSBSlaveLab.app` on the kit CD.

7.1.5 Start a New Project

1. Select **File** → **New Project**.
2. Name the project **CY3209WUSBSlaveLab**.
3. If needed, change the project location using **Browse** to move to a different folder.
4. Click **OK**.

7.1.6 Add an Input Driver

Table 7-1. 0 to 2600 mV Driver

Driver/Valuator	Unscaled Voltage - (0-2600mV)	
Location	Inputs Tab ↳ Voltage Input ↳ DC ↳ 0 to 2600 mV	
Properties	**Field**	**User Input**
	Name	Volts

Note: The potentiometers on the CY3209 board can generate up to 3300 mV. The voltage input driver used in this lab measures up to 2600 mV.

7.1.7 Add an Interface Valuator

Note: These steps must be performed in the Add Valuator dialog box that appears when the Dis-create Interface Valuator is initially placed in the design.

Table 7-2. Discrete Interface Valuator

Driver/Valuator	Discrete	
Location	Valuators Tab ↳ Interface (Communication) Valuators ↳ Discrete	
Properties	**Field**	**User Input**
	Name	Command
	Default Value	0x0 (Default)
	States	(See Adding States Below)

Adding States:

a. Click in the **States** field. Select the ellipsis button to add values.

b. In the Add Values window, add the following names and expressions in the following table:

Note: After entering a name and expression, click **Add** to add the name and expression to the bottom of the list, or click **Insert** to insert the name and expression above the last selected name and expression in the list. The names and expressions must appear in the Add Values window in the order shown in the table:

Table 7-3. Command Interface Valuator States Expressions

Name	Expression
Low	0
Medium	1
High	2

← Select the box next to High to set it as the default.

c. Select the **High** check box to make it the default state.

d. Click **OK** to close the Add Valuator dialog box.

7.1.8 Add an Output Driver

Table 7-4. On/Off with Blink Driver

Driver/Valuator	Single LED - On-Off With Blink	
Location	Outputs Tab ↳ Display ↳ LED ↳ Single Color ↳ On/Off with blink	
Properties	**Field**	**User Input**
	Name	LED
	Initial Value	OFF (Default)
	BlinkRate	1 (Default)
	Current Mode	Sourcing (Default)

Note: An optional change to the driver properties is to change the BlinkRate.

7.1.9 Define Output Behavior

1. Right-click on the **LED** driver icon, and select **Transfer Function**.
2. Select **TableLookup** from the list of transfer functions and click **OK**.
3. Select **Command** in the Select Input(s) list.
4. Click **Next**.
5. Drag the Command Input values to the following LED state columns:
   - Low → OFF
   - Medium → ON
   - High → BLINKING
6. Click **OK**.

7.1.10 Add an Interface Driver

Table 7-5. Slave Driver

Driver/Valuator	WirelessUSB™ LP 1:1 Single Packet Slave with Integrated Button Bind (Artaflex AWP24S)	
Location	Interfaces Tab ↳ Communication ↳ WirelessUSB LP 1-1 Single Packet Slave ↳ Integrated Button Bind ↳ AWP24S	
Properties	**Field**	**User Input**
	Name	Wireless
	Bind Button Resistor Configuration	Internal Pulldown, Normally Open
	Device Type	0 (Default)
	Writable Area Size	1
	Readable Area Size	2
	Readable Area Offset	2
	Update Interval	200 (Default)
	Receive Timeout	40 (Default)
	Bind Mode PA Setting	-24 dBm (Default)
	Data Mode PA Setting	0 dBm (Default)
	Preamble Size	1 (Default)

7.1.11 Verify the Project Register Map

1. Select **Project** → **Assign Register Map for 'CY3209WUSBSlaveLab' Project**.
2. Verify that the Read/Write section shows:
   - Command

   and that the Read-only section shows:
   - LED
   - Volts
   - Wireless_Bind
3. If the values are not in the order shown, click on one of the values and use the Up/Down arrow buttons to move the value up or down until the values are in the order shown above.
4. Click **OK**.

7.1.12 Build the Project

1. Select **Build → Generate\Build 'CY3209WUSBSlaveLab' Project**.
2. In the Available Device Configurations pane, select **CY8C27000 → CY8C27643, 48-Pin MLF**.
3. In the Device Configurations Properties pane, set the following parameters:
   - Supply Voltage → 3.3V
   - Sample Rate → Free Run
4. Ensure the **Assign pins automatically** check box is **not** selected.
5. Click **Next**.
6. In the User Pin Assignment window, click **Unassign All Pins**.
7. Assign drivers to pins in the order listed by dragging and dropping the blue rectangles from the Unassigned Drivers list to the pins listed in the following table.

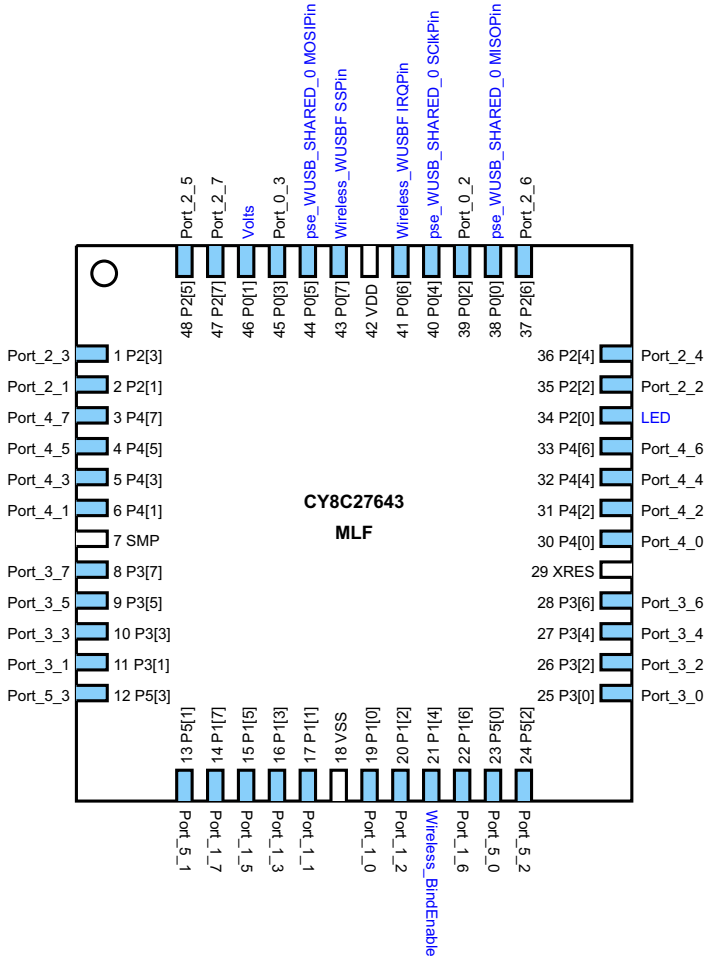
WARNING: Assign pins in the order listed. When the Wireless_WUSBF IRQPin driver is assigned (number 4 in the Assignment Order), PSoC Express automatically assigns the rest of the drivers. Manually reassign the last three drivers to the correct pins.

Table 7-6. CY3209WUSBSlaveLab Pinouts

Assignment Order (Starts at 1)	Unassigned Drivers Index (Starts at 0)	Function	Pin	Notes
1	0	LED	P2[0]	
2	1	Volts	P0[1]	
3	5	Wireless_BindEnable	P1[4]	
4	6	Wireless_WUSBF IRQPin	P0[6]	←The last four drivers are assigned automatically when the Wireless_WUSBFMASTER IRQPin driver is assigned. You must manually reassign the last three pse_WUSB drivers.
5	7	Wireless_WUSBF SSPin	P0[7]	
6	2	pse_WUSB_SHARED_0 MISOPin	Move From P1[0] to P0[0]	
7	4	pse_WUSB_SHARED_0 SClkPin	Move From P1[2] to P0[4]	
8	3	pse_WUSB_SHARED_0 MOSIPin	Move From P1[1] to P0[5]	

Note: The pin label notation P1[2] indicates Port 1, Pin 2.

Figure 7-1. CY3209WUSBSlaveLab Pinouts

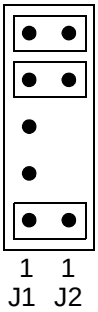


8. Click **Next**. PSoC Express generates and compiles the project source code.

7.1.13 Program the PSoC Device

1. Disconnect any power sources from the demonstration board.
2. Remove the jumpers between J1 and J2.
3. Program the part via J1 as described in the Programming Instructions for CY3209 PSoC Express Projects section on page 10.
4. Install jumpers on the following pin combinations:

Figure 7-2. CY3209WUSBSlaveLab Jumper Settings



7.2 WirelessUSB Master/Slave Lab – Master

7.2.1 Description

This project is the master device in the WirelessUSB Master/Slave lab. The project reads a voltage value from the slave device over a WirelessUSB connection. The voltage value displays on the four digit LED display. A control value of 0, 1, or 2 is written to the slave device based on the voltage level. This value instructs the slave to set its LED to Off, On, or Blinking.

7.2.2 Board Quadrant

Bottom Right

7.2.3 PSoC Device

CY8C29666

7.2.4 Finished Project Location

`\PSoCExpressExamples\CY3209WUSBMasterLab\CY3209WUSBMasterLab.app` on the kit CD.

7.2.5 Start a New Project

1. Select **File** → **New Project**.
2. Name the project **CY3209WUSBMasterLab**.
3. If needed, change the project location using **Browse** to move to a different folder.
4. Click **OK**.

7.2.6 Add an Input Driver

Table 7-7. WirelessUSB LP 1:1 Single Packet Slave Monitor Driver

Driver/Valuator	WirelessUSB LP 1:1 Single Packet Slave Monitor	
Location	Inputs Tab ↳ Remote Devices ↳ WirelessUSB ↳ WirelessUSB LP 1:1 Single Packet Slave Monitor	
Properties	**Field**	**User Input**
	Name	VoltsMonitor
	Variable Type	INT
	Offset	2

Note: The Offset property value is determined by the layout of the WirelessUSB slave project Interface Register Map.

7.2.7 Add Three Output Drivers

Table 7-8. Output Driver 1: Common Cathode Driver

Driver/Valuator	Four Digit LED, Common Cathode	
Location	Outputs Tab ↳ Display ↳ LED ↳ 7-Segment ↳ Multi Digit ↳ Four Digit with Decimal Point ↳ Common Cathode	
Properties	**Field**	**User Input**
	Name	FourDigLED
	Initial Value	0 (Default)
	Position of decimal point	NONE (Default)

Table 7-9. WirelessUSB LP 1:1 Single Packet Slave Control Driver

Driver/Valuator	WirelessUSB LP 1:1 Single Packet Slave Control	
Location	Outputs Tab ↳ Remote Devices ↳ WirelessUSB ↳ WirelessUSB LP 1:1 Single Packet Slave Control	
Properties	**Field**	**User Input**
	Name	LevelControl
	Initial Value	0 (Default)
	Variable Type	BYTE (Default)
	Offset	0 (Default)

Note: The Offset property value is determined by the layout of the WirelessUSB slave project Interface Register Map.

Table 7-10. Output Driver 3: AWP24S Driver

Driver/Valuator	WirelessUSB™ LP 1:1 Single Packet Master with Integrated Button Bind (Artaflex AWP24S)		
Location	Outputs Tab ↳ WirelessUSB LP 1-1 Single Packet Master ↳ Integrated Button Bind ↳ AWP24S		
Properties	**Field**	**User Input**	
	Name	Wireless	
	Bind Button Resistor Configuration	Internal Pulldown, Normally Open	
	Slave Device Type	0	(Default)
	Slave Writable Area Size	1	
	Slave Readable Area Size	2	
	Slave Readable Area Offset	2	
	Bind Mode PA Setting	-24 dBm	(Default)
	Data Mode PA Setting	0 dBm	(Default)
	Preamble Size	1	(Default)

7.2.8 Define Output Behavior

1. Right-click on the FourDigLED driver icon and select **Transfer Function**.
2. Select **StatusEncoder** from the list of transfer functions and click **OK**.
3. Enter rules for the StatusEncoder into the If/Then fields according to the following table.

Table 7-11. FourDigLED StatusEncoder If/Then Settings

If	Then
1	VoltsMonitor

4. Click **OK**.
5. Right-click on the LevelControl driver icon and select **Transfer Function**.
6. Select **PriorityEncoder** from the list of transfer functions and click **OK**.
7. Enter rules for the PriorityEncoder according to Table 7-12. To add a row to the PriorityEncoder, click the down arrow button as you complete each if/then statement.

Table 7-12. Level Control PriorityEncoder If/Then Settings

If/Else If	Then
VoltsMonitor < 1000	0
VoltsMonitor < 2000	1
1	2

8. Click **OK**.

7.2.9 Build the Project

1. Select **Build → Generate/Build 'CY3209WUSBMasterLab' Project**.
2. In the Available Device Configurations pane, select **CY8C29000 → CY8C29666, 48-Pin MLF**.
3. In the Device Configurations Properties pane, set the following parameters:
   - Supply Voltage → 3.3V
   - Sample Rate → Free Run
4. Ensure the **Assign pins automatically** check box is **not** selected.
5. Click **Next**.
6. In the User Pin Assignment window, click **Unassign All Pins**.
7. Assign drivers to pins in the order listed by dragging and dropping the blue rectangles from the Unassigned Drivers list to the pins listed in the following table.

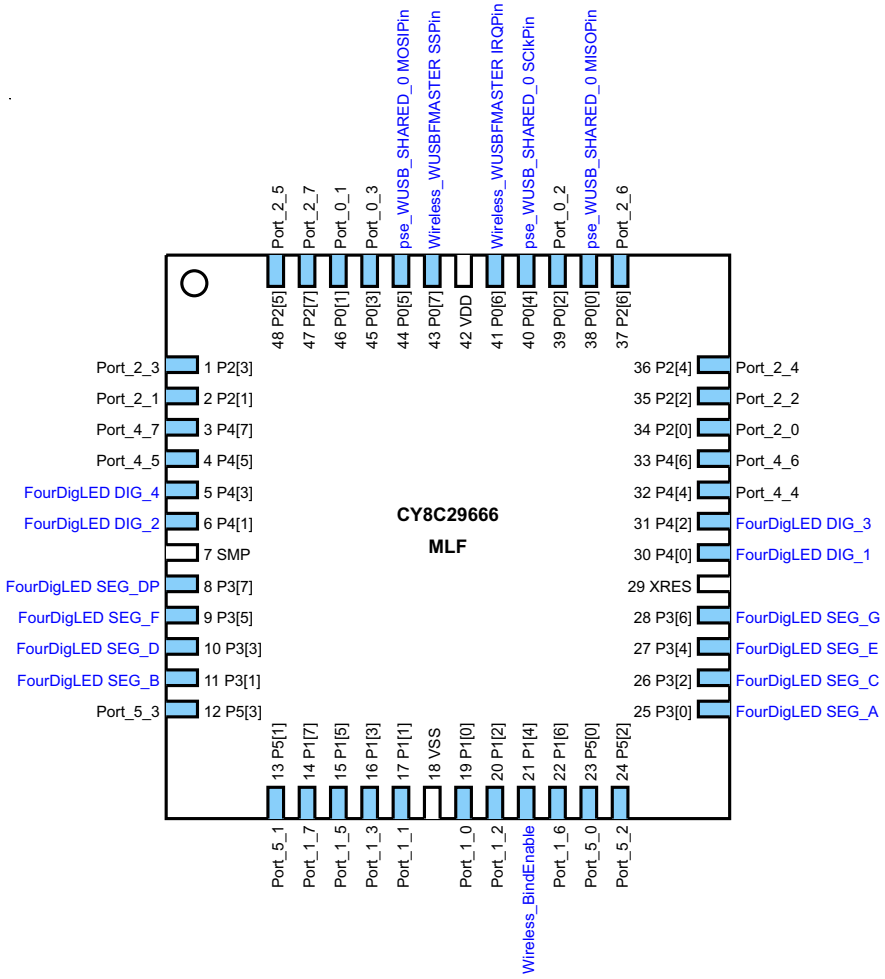
WARNING: Assign pins in the order listed. When the Wireless_WUSBF IRQPin driver is assigned (number 14 in the Assignment Order), PSoC Express automatically assigns the rest of the drivers. Manually reassign the last three pse_WUSB_SHARED_0 drivers to the correct pins.

Table 7-13. CY3209WUSBMasterLab User Pin Assignments

Assignment Order (Starts at 1)	Unassigned Drivers Index (Starts at 0)	Function	Pin	Notes
1	0	FourDigLED DIG_1	P4[0]	←FourDigLED DIGs 2 through 4 are assigned automatically when FourDigLED DIG_1 is assigned.
2	1	FourDigLED DIG_2	P4[1]	
3	2	FourDigLED DIG_3	P4[2]	
4	3	FourDigLED DIG_4	P4[3]	
5	4	FourDigLED SEG_A	P3[0]	←FourDigLED SEGs B through G are assigned automatically when FourDigLED SEG_A is assigned.
6	5	FourDigLED SEG_B	P3[1]	
7	6	FourDigLED SEG_C	P3[2]	
8	7	FourDigLED SEG_D	P3[3]	
9	8	FourDigLED SEG_DP	P3[7]	
10	9	FourDigLED SEG_E	P3[4]	
11	10	FourDigLED SEG_F	P3[5]	
12	11	FourDigLED SEG_G	P3[6]	
13	15	Wireless_BindEnable	P1[4]	
14	16	Wireless_WUSBFMASTER IRQPin	P0[6]	←The last four drivers are assigned automatically when the Wireless_ WUSBFMASTER IRQPin driver is assigned. You must manually reassign the last three pse_WUSB drivers.
15	17	Wireless_WUSBFMASTER SSPin	P0[7]	
16	12	pse_WUSB_SHARED_0 MISOPin	Move From P1[0] to P0[0]	
17	13	pse_WUSB_SHARED_0 SClkPin	Move From P1[2] to P0[4]	
18	14	pse_WUSB_SHARED_0 MOSIPin	Move From P1[1] to P0[5]	

Note: The pin label notation P1[2] indicates Port 1, Pin 2.

Figure 7-3. CY3209WUSBMasterLab Pinouts

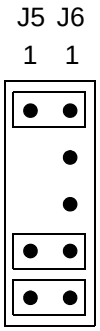


8. Click **Next**. PSoC Express generates and compiles the project source code.

7.2.10 Program the PSoC Device

1. Disconnect any power sources from the demonstration board.
2. Remove the jumpers between J5 and J6.
3. Program the part via J6 as described in Programming Instructions for CY3209 PSoC Express Projects section on page 10.
4. Install jumpers on the following pin combinations:

Figure 7-4. CY3209WUSBMasterLab Jumper Settings



7.2.11 Test Your Project on the Demonstration Board

1. Ensure the slave project has been programmed into the bottom left quadrant of the demonstration board and that the jumpers specified in Program the PSoC Device on page 54 have been installed.
2. Install radio modules on J3 and J4 of the demonstration board. Align Pin 1 of each radio module with Pin 1 of its respective board connector.
3. Power the board with a 9V battery (provided with the kit) or an optional 6–14V DC wall transformer.
4. Within 20 seconds of each other, press and release the pushbuttons on the bottom left and bottom right quadrants of the demonstration board. This accomplishes a binding process in which the slave obtains the master's ID and stores it in non-volatile memory. The master's ID provides information the slave needs to successfully exchange application data with the master.
5. Adjust the potentiometer on the slave (bottom left quadrant). Observe the master’s multi-digit LED display and the slave’s LED.
6. Observe the following relationship between the multi-digit LED display on the master and the single LED on the slave:
   - 0-999 → LED is Off
   - 1000-1999 → LED is On
   - >2000 → LED is Blinking

8. Cypress WirelessUSB™ Master/Slave Lab 2

This lab implements a wireless tilt sensor. The lab creates two projects: a WirelessUSB slave and a WirelessUSB master. The slave device transmits information from a dual axis accelerometer to the master device. The master device uses this information to display the XY tilt on the 8 LEDs in the bottom left quadrant of the master demonstration board. The LED pattern is also transmitted back to the slave device for display on the slave demonstration board.

This lab requires two CY3209 kits.

8.1 WirelessUSB Master/Slave Lab – Slave

8.1.1 Description

This project is the slave device for the WirelessUSB Master/Slave lab. The project includes a dual axis accelerometer input and a pair of 4-bit banked logic output drivers that control 8 LEDs. The accelerometer information is transmitted to a WirelessUSB master using a WirelessUSB interface driver. The slave's interface driver also captures the LED pattern sent back by the master into interface valuators. These values are then used to light the slave LEDs.

Note: This project will not work unless the Add Driver Dialog option is selected. To use the Add Driver Dialog, ensure **View** → **Use Add Driver Dialog** is checked.

8.1.2 Board Quadrant

Bottom Left

8.1.3 PSoC Device

CY8C27643

8.1.4 Finished Project Location

`\PSoCExpressExamples\CY3209WUSBTiltSlave\CY3209WUSBTiltSlave.app` on the kit CD.

8.1.5 Start a New Project

1. Select **File** → **New Project**.
2. Name the project **CY3209WUSBTiltSlave**.
3. If needed, change the project location using **Browse** to move to a different folder.
4. Click **OK**.

8.1.6 Add an Input Driver

Table 8-1. ADXL322 (+/-2g) Driver

Driver/Valuator	ADXL322 Dual Axis Accelerometer (±2g)	
Location	Inputs Tab ↳ Acceleration ↳ Dual Axis XY ↳ ADXL322 (+/-2g)	
Properties	**Field**	**User Input**
	Name	AccXY

8.1.7 Add Two Output Drivers

Table 8-2. Output Driver 1: Banked Output Driver

Driver/Valuator	Banked Output	
Location	Outputs Tab ↳ Digital Output ↳ Banked Output	
Properties	**Field**	**User Input**
	Name	XLeds
	Number of output pins	4
	DriveMode	Strong (default)

Table 8-3. Output Driver 2: Banked Output Driver

Driver/Valuator	Banked Output	
Location	Outputs Tab ↳ Digital Output ↳ Banked Output	
Properties	**Field**	**User Input**
	Name	YLeds
	Number of output pins	4
	DriveMode	Strong (default)

8.1.8 Add an Interface Driver

Table 8-4. AWP24S Driver

Driver/Valuator	WirelessUSB LP 1:1 Single Packet Slave (Artaflex AWP24S, Simple Configuration)	
Location	Interfaces Tab ↳ Communication ↳ WirelessUSB LP 1:1 Single Packet Slave ↳ Simple Configuration ↳ AWP24S	
Properties	**Field**	**User Input**
	Name	Wireless
	Device Type ID	0 (default)
	Update Interval	200 (default)
	Transmit Base Offset	2
	Bytes to Transmit	4
	Bytes to Receive	2

Note: In general usage, the values for the Transmit Base Offset, Bytes To Transmit, and Bytes To Receive properties are determined by the project Interface Register Map. In this lab, the values have been pre-calculated to ensure successful project execution.

8.1.9 Add Two Interface Valuators

Table 8-5. Interface Valuator 1: Discrete Interface Valuator

Driver/Valuator	Discrete	
Location	Valuators Tab ↳ Interface (Communication) Valuators ↳ Discrete	
Properties	**Field**	**User Input**
	Name	CommandX
	Default Value	0xF
	States	(Leave Blank)

Table 8-6. Interface Valuator 2: Discrete Interface Valuator

Driver/Valuator	Discrete	
Location	Valuators Tab ↳ Interface (Communication) Valuators ↳ Discrete	
Properties	**Field**	**User Input**
	Name	CommandY
	Default Value	0xF
	States	(Leave Blank)

8.1.10 Define Output Behavior

1. Right-click on the **XLeds** driver icon, and select **Transfer Function**.
2. Select **PriorityEncoder** from the list of transfer functions and click **OK**.
3. Enter the following rule for the PriorityEncoder:

   Table 8-7. XLeds PriorityEncoder If/Then Settings

If/Else If	Then
1	CommandX

4. Click **OK**.
5. Right-click on the **YLeds** driver icon and select **Transfer Function**.
6. Select **PriorityEncoder** from the list of transfer functions and click **OK**.
7. Enter the following rule for the PriorityEncoder:

   Table 8-8. YLeds PriorityEncoder If/Then Settings

If/Else If	Then
1	CommandY

8. Click **OK**.

8.1.11 Verify the Project Register Map

1. Select **Project → Assign Register Map for 'CY3209WUSBTiltSlave' Project**.
2. Verify that the Read\Write section shows:
   - CommandX
   - CommandY

   and that the Read-only section shows:
   - AccXY_X
   - AccXY_Y
   - XLeds_packed_val
   - YLeds_packed_val
3. If the values are not in the order shown, click on one of the values and use the Up/Down arrow buttons to move the value up or down until the values are in the order shown above.

8.1.12 Build the Project

1. Select **Build** → **Generate\Build 'CY3209WUSBTiltSlave' Project**.
2. In the Available Device Configurations pane, select **CY8C27000** → **CY8C27643, 48-Pin MLF**.
3. In the Device Configurations Properties pane, set the following parameters:
   - Supply Voltage → 3.3V
   - Sample Rate → Free Run
4. Ensure the **Assign pins automatically** check box is **not** selected.
5. Click **Next**.
6. In the User Pin Assignment window, click **Unassign All Pins**.
7. Assign drivers to pins in the order listed by dragging and dropping the blue rectangles from the Unassigned Drivers list to the pins listed in the following table.

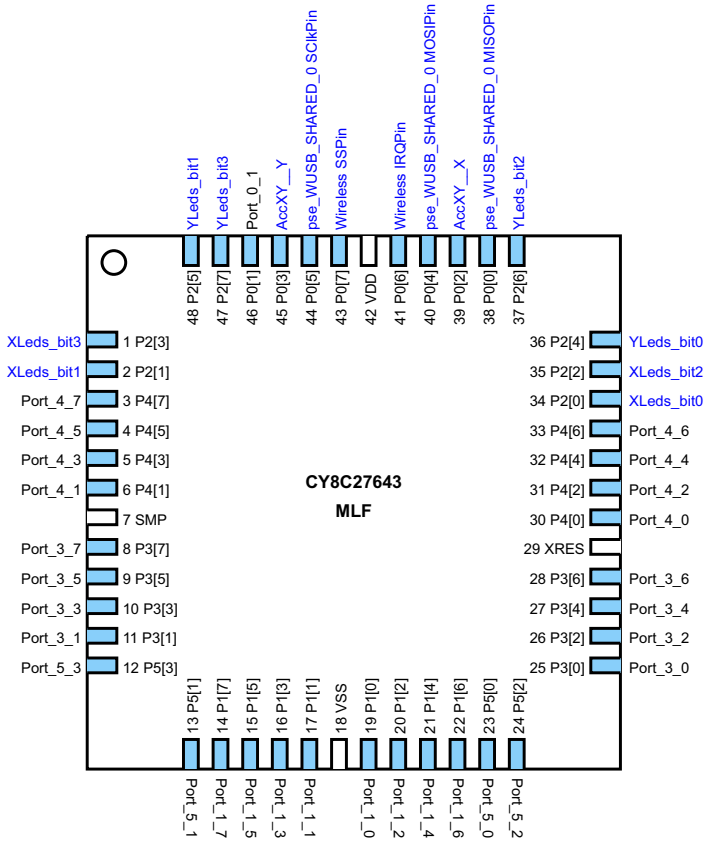
WARNING: Assign pins in the order listed. When the Wireless_WUSBF IRQPin driver is assigned (number 11 in the Assignment Order), PSoC Express automatically assigns the rest of the drivers. Manually assign the last three pse_WUSB_SHARED_0 drivers to the correct pins.

Table 8-9. CY3209WUSBTiltSlave Pinouts

Assignment Order (Starts at 1)	Unassigned Drivers Index (Starts at 0)	Function	Pin	Notes
1	0	AccXY__X	P0[2]	
2	1	AccXY__Y	P0[3]	
3	7	XLeds_bit0	P2[0]	
4	8	XLeds_bit1	P2[1]	
5	9	XLeds_bit2	P2[2]	
6	10	XLeds_bit3	P2[3]	
7	11	YLeds_bit0	P2[4]	
8	12	YLeds_bit1	P2[5]	
9	13	YLeds_bit2	P2[6]	
10	14	YLeds_bit3	P2[7]	
11	5	Wireless IRQPin	P0[6]	←The last four drivers are assigned automatically when the Wireless_ WUSBFMASTER IRQPin driver is assigned. You must manually reassign the three pse_WUSB drivers
12	6	Wireless SSPin	P0[7]	
13	2	pse_WUSB_SHARED_0 MISOPin	Move From P1[0] to P0[0]	
14	4	pse_WUSB_SHARED_0 SClkPin	Move From P1[2] to P0[4]	
15	3	pse_WUSB_SHARED_0 MOSIPin	Move From P1[1] to P0[5]	

Note: The pin label notation P1[2] indicates Port 1, Pin 2.

Figure 8-1. CY3209WUSBTiltSlave Pinouts

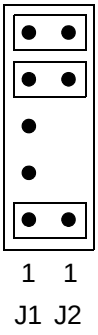


8. Click **Next**. PSoC Express generates and compiles the project source code.

8.1.13 Program the PSoC Device

1. Disconnect any power sources from the demonstration board, and connect the MiniProg to your PC using a USB cable.
2. Remove the jumpers between J1 and J2.
3. Program the part via J1 as described in Programming Instructions for CY3209 PSoC Express Projects section on page 10.
4. Install jumpers on the following pin combinations:

Figure 8-2. CY3209WUSBTiltSlave Jumper Settings



8.1.14 Test Your Slave Device on the Demonstration Board

1. Install the radio module on J3 of the demonstration board. Align Pin 1 of the radio module with J3, Pin 1 on the demonstration board.
2. Power the board with a 9V battery (provided with the kit) or an optional 6–14V DC wall transformer.
3. On power-up, the slave attempts to bind with a master. After approximately 20 to 25 seconds, the cross pattern of LEDs lights. With no WirelessUSB master device to bind with, the slave project times-out and displays the default LED pattern (all LEDs on).

WARNING If multiple people are performing the master lab at the same time as this lab in the same location, the results may be unpredictable.

8.2 WirelessUSB Master/Slave Lab – Master

8.2.1 Description

This project is the master device for the WirelessUSB Master/Slave lab. The project uses a WirelessUSB LP 1:1 Single Packet Master (Artaflex AWP24S, Simple Configuration) driver to provide a wireless connection to the slave. WirelessUSB LP 1:1 Single Packet Slave Monitor drivers access the accelerometer data from the slave device. The XY tilt of the slave board displays on the local LEDs based on the received acceleration values. WirelessUSB LP 1:1 Single Packet Slave Control drivers are used to send the LED display pattern values to the slave.

8.2.2 Board Quadrant

Bottom Left

8.2.3 PSoC Device

CY8C27643

8.2.4 Finished Project Location

`\PSoCExpressExamples\CY3209WUSBTiltMaster\CY3209WUSBTiltMaster.app` on the kit CD.

8.2.5 Start a New Project

1. Select **File** → **New Project**.
2. In the Name box, enter **CY3209WUSBTiltMaster**.
3. If needed, click **Browse** to save the project in a different location.
4. Click **OK**.

8.2.6 Add Three Input Drivers

Table 8-10. Input Driver 1: Internal Pulldown N_O Driver

Driver/Valuator	Pushbutton - (Normally Open, Int. Pull-Down)	
Location	Inputs Tab ↳ Tactile ↳ Button ↳ Normally Open ↳ Internal Pulldown N_O	
Properties	**Field**	**User Input**
	Name	BindReq

Table 8-11. Input Driver 2: WirelessUSB LP 1:1 Single Packet Slave Monitor Driver

Driver/Valuator	WirelessUSB LP 1:1 Single Packet Slave Monitor	
Location	Inputs Tab ↳ Remote Devices ↳ WirelessUSB ↳ WirelessUSB LP 1:1 Single Packet Slave Monitor	
Properties	**Field**	**User Input**
	Name	InAccXY_X
	Variable Type	INT
	Offset	2

Note: The Offset property value is determined by the layout of the WirelessUSB slave project's Interface Register Map. This map is found in the project data sheet generated when the slave project was built.

Table 8-12. Input Driver 3: WirelessUSB LP 1:1 Single Packet Slave Monitor Driver

Driver/Valuator	WirelessUSB LP 1:1 Single Packet Slave Monitor	
Location	Inputs Tab ↳ Remote Devices ↳ WirelessUSB ↳ WirelessUSB LP 1:1 Single Packet Slave Monitor	
Properties	**Field**	**User Input**
	Name	InAccXY_Y
	Variable Type	INT
	Offset	4

Note: The Offset property value is determined by the layout of the WirelessUSB slave project Interface Register Map.

8.2.7 Add Five Output Drivers

Table 8-13. Output Driver 1: WirelessUSB LP 1:1 Single Packet Slave Control Driver

Driver/Valuator	WirelessUSB LP 1:1 Single Packet Slave Control Driver	
Location	Outputs Tab ↳ Remote Devices ↳ WirelessUSB ↳ WirelessUSB LP 1:1 Single Packet Slave Control	
Properties	**Field**	**User Input**
	Name	OutXLeds
	Initial Value	15
	Variable Type	BYTE (Default)
	Offset	0 (Default)

Note: The Offset property value is determined by the layout of the WirelessUSB slave project Interface Register Map.

Table 8-14. Output Driver 2: WirelessUSB LP 1:1 Single Packet Slave Control Driver

Driver/Valuator	WirelessUSB LP 1:1 Single Packet Slave Control	
Location	Outputs Tab ↳ Remote Devices ↳ WirelessUSB ↳ WirelessUSB LP 1:1 Single Packet Slave Control	
Properties	**Field**	**User Input**
	Name	OutYLeds
	Initial Value	15
	Variable Type	BYTE (Default)
	Offset	1

Note: The Offset property value is determined by the layout of the WirelessUSB slave project Interface Register Map.

Table 8-15. Output Driver 3: AWP24S

Driver/Valuator	WirelessUSB LP 1:1 Single Packet Master (Artaflex AWP24S, Simple Configuration)	
Location	Outputs Tab ↳ WirelessUSB LP 1-1 Single Packet Master ↳ Simple Configuration ↳ AWP24S	
Properties	**Field**	**User Input**
	Name	Wireless
	Initial Value	DO_NOT_BIND (Default)
	Slave Device Type ID	0 (Default)
	Receive Base Offset	2
	Bytes to Receive	4
	Bytes To Transmit	2

Note: The Receive Base Offset, Bytes to Receive, and Bytes To Transmit property values align with the corresponding property values (Transmit Base Offset, Bytes to Transmit, and Bytes to Receive) of the WirelessUSB Interface driver used in the slave project.

Table 8-16. Output Driver 4: Banked Output Driver

Driver/Valuator	Banked Output	
Location	Outputs Tab ↳ Digital Output ↳ Banked Output	
Properties	**Field**	**User Input**
	Name	XLeds
	Number of output pins	4
	DriveMode	Strong (Default)

Table 8-17. Output Driver 5: Banked Output Driver

Driver/Valuator	Banked Output	
Location	Outputs Tab ↳ Digital Output ↳ Banked Output	
Properties	**Field**	**User Input**
	Name	YLeds
	Number of output pins	4
	DriveMode	Strong (Default)

8.2.8 Define Output Behavior

1. Define the XLeds output behavior:
   a. Right-click on the **XLeds** driver icon, and select **Transfer Function**.
   b. Select **PriorityEncoder** from the list of transfer functions and click **OK**.
   c. Enter rules for the PriorityEncoder according to Table 8-18. To add a row to the PriorityEncoder, click the down arrow button as you complete each if/then statement:

   Table 8-18. XLeds If/Then Settings

If/Else If	Then
InAccXY_X<-800	1
InAccXY_X<-300	2
InAccXY_X<300	0
InAccXY_X<800	4
1	8

   d. Click **OK**.
2. Define the YLeds output behavior:
   a. Right-click on the **YLeds** driver icon, and select **Transfer Function**.
   b. Select **PriorityEncoder** from the list of transfer functions and click **OK**.
   c. Enter rules for the PriorityEncoder according to Table 8-19. To add a row to the PriorityEncoder, click the down arrow button as you complete each if/then statement:

   Table 8-19. YLeds If/Then Settings

If/Else If	Then
InAccXY_Y<-800	1
InAccXY_Y<-300	2
InAccXY_Y<300	0
InAccXY_Y<800	4
1	8

   d. Click **OK**.

3. Define the Wireless output behavior:
   a. Right-click on the **Wireless** driver icon, and select **Transfer Function**.
   b. Select **TableLookup** from the list of transfer functions and click **OK**.
   c. Select **BindReq** as the input and click **Next**.
   d. Drag the BindReq values to the following output state columns:
      - Off → DO_NOT_BIND
      - On → BIND
   e. Click **OK**.
4. Define the OutXLeds output behavior:
   a. Right-click on the **OutXLeds** driver icon, and select **Transfer Function**.
   b. Select **PriorityEncoder** from the list of transfer functions and click **OK**.
   c. Enter the following rule for the PriorityEncoder:

      Table 8-20. OutXLeds If/Then Settings

If/Else If	Then
1	XLeds_packed_val

   d. Click **OK**.
5. Define the OutYLeds output behavior:
   a. Right-click on the **OutYLeds** driver icon, and select **Transfer Function**.
   b. Select **PriorityEncoder** from the list of transfer functions and click **OK**.
   c. Enter the following rule for the PriorityEncoder:

      Table 8-21. OutYLeds If/Then Settings

If/Else If	Then
1	YLeds_packed_val

   d. Click **OK**.

8.2.9 Build the Project

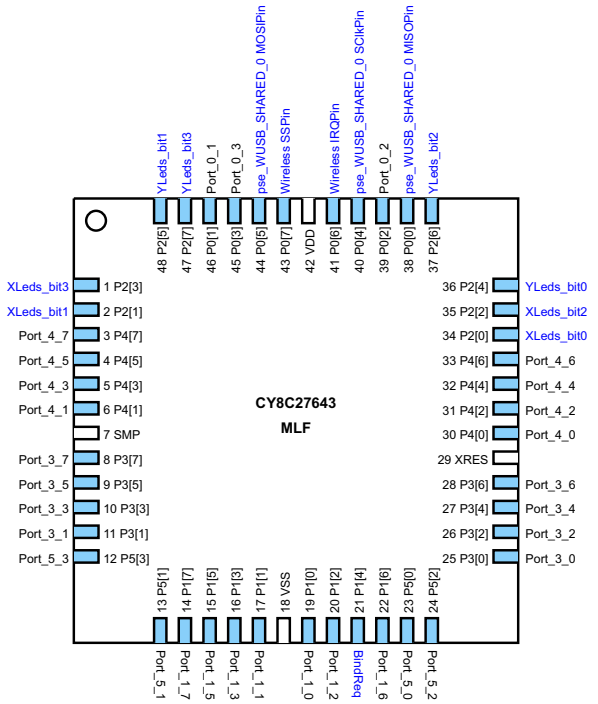
1. Select **Build** → **Generate/Build 'CY3209WUSBTiltMaster' Project**.
2. In the Available Device Configurations pane, select **CY8C27000** → **CY8C27643, 48-Pin MLF**
3. In the Device Configurations Properties pane, set the following driver properties:
   - Supply Voltage → 3.3V
   - Sample Rate → Free Run
4. Ensure the **Assign pins automatically** check box is **not** selected.
5. Click **Next**.
6. In the User Pin Assignment window, click **Unassign All Pins**.
7. Assign drivers to pins in the order listed by dragging and dropping the blue rectangles from the Unassigned Drivers list to the pins listed in the following table.

WARNING: **Assign pins in the order listed**. When the Wireless_WUSBF IRQPin driver is assigned (number 10 in the Assignment Order), PSoC Express automatically assigns the rest of the drivers. Manually assign the three pse_WUSB_SHARED_0 drivers to the correct pins.

Table 8-22. CY3209WUSBTiltMaster Pinouts

Assignment Order (Starts at 1)	Unassigned Drivers Index (Starts at 0)	Function	Pin	Notes
1	0	BindReq	P1[4]	
2	6	XLeds_bit0	P2[0]	
3	7	XLeds_bit1	P2[1]	
4	8	XLeds_bit2	P2[2]	
5	9	XLeds_bit3	P2[3]	
6	10	YLeds_bit0	P2[4]	
7	11	YLeds_bit1	P2[5]	
8	12	YLeds_bit2	P2[6]	
9	13	YLeds_bit3	P2[7]	
10	4	Wireless IRQPin	P0[6]	←The last four drivers are assigned automatically when the Wireless_WUSBFMASTER IRQPin driver is assigned. You must manually reassign the last three pse_WUSB drivers.
11	5	Wireless SSPin	P0[7]	
12	1	pse_WUSB_SHARED_0 MISOPin	Move From P1[0] to P0[0]	
13	3	pse_WUSB_SHARED_0 SClkPin	Move From P1[2] to P0[4]	
14	2	pse_WUSB_SHARED_0 MOSIPin	Move From P1[1] to P0[5]	

Figure 8-3. CY3209WUSBTiltMaster Pinouts



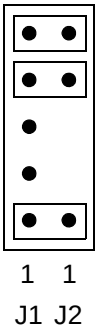
8. Click **Next**. PSoC Express generates and compiles the project source code.

8.2.10 Program the PSoC Device

Note: The master project must be programmed on a different demonstration board from the one used for the slave project.

1. Disconnect any power sources from the demonstration board.
2. Remove the jumpers between J1 and J2.
3. Program the part via J1 as described in the Programming Instructions for CY3209 PSoC Express Projects section on page 10.
4. Install jumpers on the following pin combinations:

Figure 8-4. CY3209WUSBTiltMaster Jumper Settings



8.2.11 Test Your WirelessUSB System

1. Install radio modules on J3 of both the master and slave demonstration boards. Align Pin 1 of the radio module with J3, Pin 1 on the demonstration board.
2. Power the master board with a 9V battery (provided with the kit) or an optional 6–14V DC wall-transformer.
3. Power the slave board with a 9V battery (provided with the kit) or an optional 6–14V DC wall-transformer.
4. Within 20 seconds of powering the slave board, press and release the pushbutton on the bottom left quadrant of the master board. This accomplishes one-time binding process in which the slave obtains the master's ID and stores it in non-volatile memory. The master's ID provides information the slave needs to successfully exchange application data with the master.
5. Tilt the WirelessUSB Slave project demonstration board. The “cross” pattern LEDs on both boards should indicate the direction of tilt. Tilting the WirelessUSB Master project demonstration board should have no affect on the LEDs.
6. Remove power from one board. Observe that the LED pattern stops updating on the board that remains powered.
7. Apply power to the unpowered board. Observe that the cross pattern starts updating on both boards.

Note: Switching the radios between boards after binding requires the slave to be reprogrammed and the binding process to be executed again.

WARNING: If multiple people are performing this lab in the same location, the results may be unpredictable.