

Please note that Cypress is an Infineon Technologies Company.

The document following this cover page is marked as "Cypress" document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content

The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers

Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.



CY3215-DK

PSoC[®] 1 In-Circuit Emulator Development Kit Guide

Doc. # 001-66514 Rev. *E

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



Copyrights

© Cypress Semiconductor Corporation, 2011-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PUR-POSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities. including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.

Contents



1.1 Kit Contents 1.2 Additional Learning Resources 1.3 Documentation Conventions 2. Getting Started 2.1 Kit Installation 2.2 PSoC Designer 2.3 PSoC Programmer	
1.3 Documentation Conventions 2. Getting Started 2.1 Kit Installation 2.2 PSoC Designer 2.3 PSoC Programmer	4
1.3 Documentation Conventions 2. Getting Started 2.1 Kit Installation 2.2 PSoC Designer 2.3 PSoC Programmer	5
 2.1 Kit Installation 2.2 PSoC Designer 2.3 PSoC Programmer 	
 2.2 PSoC Designer 2.3 PSoC Programmer 	8
 2.2 PSoC Designer 2.3 PSoC Programmer 	8
2.3 PSoC Programmer	
3. Using ICE-Cube	
	14
3.1 Connecting the ICE-Cube	.14
3.1.1 Connect using a USB Port	
3.1.2 Connect using a Flex Cable	.15
3.1.3 Connect using a Backward Compatibility Adapter	.17
3.1.4 Software Configuration	
3.1.5 Debug a Project	.19
3.1.5.1 Break Points	.21
3.1.5.2 CPU and Register Views	.21
3.1.5.3 Watch Variables	.22
3.1.5.4 Trace	.22
3.1.5.5 Locals	.23
3.2 PSoC Programmer	.24
4. Code Examples	26
4.1 My First Code Example	.26
4.1.1 Project Objective	
4.1.2 Creating My First PSoC 1 Project	
4.1.3 Verify Output	

Contents





Thank you for your interest in the CY3215-DK PSoC[®] 1 In-Circuit Emulation (ICE) Development Kit (DVK) guide. You can use this kit with PSoC Designer[™] or PSoC Programmer. The CY3215-DK provides debugging functionality for PSoC 1 and manages all the emulation communication between the debugger software running on the computer (PSoC Designer) and the target PSoC chip.

PSoC Designer is PSoC 1's GUI-based integrated design environment (IDE), which includes a powerful set of tools for developing code, prototyping, and debugging. This document gives an overview of the hardware configurations that can be used to set up a debugging system for PSoC 1 using the CY3215-DK kit. The Code Examples on page 26 demonstrate the use of the critical software elements of the debugging system.

The CY3215-DK kit supports the following PSoC 1 families.

- CY8C20x34
- CY8C20xx6A
- CY8C21x23
- CY8C21x34
- CY8C22xxx/CY8C21x45
- CY8C23x33
- CY8C24x23A
- CY8C24x94
- CY8C27x43
- CY8C28xxx
- CY8C29x66
- CY8C95xx

1.1 Kit Contents

The CY3215-DK kit includes the following:

- ICE-Cube in-circuit emulator
- ISSP cable
- USB 2.0 cable
- Blue Cat-5e cable
- MiniEval programming board
- Two units of 28-pin DIP samples (CY8C29466-24PXI)
- ZIF socket
- CY3250 flex cable
- Backward compatibility adapter
- 29000-28 PDIP kit



- 12-V 1-A adapter
- CY3215-DK kit CD
 - PSoC Designer installation file
 - PSoC Programmer installation file
 - □ Bridge Control Panel installation file (packaged along with PSoC Programmer)
 - □ Kit guide
 - Quick start guide
 - Release notes

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

1.2 Additional Learning Resources

Visit http://www.cypress.com for additional learning resources in the form of datasheets, technical reference manual, and application notes. For more information, go to:

- PSoC Designer functionality and releases: http://www.cypress.com/go/psocdesigner
- PSoC Programmer, supported hardware and COM layer: http://www.cypress.com/go/psocprogrammer
- PSoC Designer-related trainings: http://www.cypress.com/go/psocdesignertraining
- CY3250-29xxxQFN ICE Pod Schematic: http://www.cypress.com/go/CY3250-29xxxQFN
- CY3250-FLEXCABLE Mechanical Layout Drawing: http://www.cypress.com/go/CY3250-29xxxQFN
- CY3250-29xxxQFN ICE Pod Mechanical Layout Drawing: http://www.cypress.com/go/CY3250-29xxxQFN
- AN73212 Debugging with PSoC 1: http://www.cypress.com/?rID=57555

1.3 Documentation Conventions

Table 1-1. Document Conventions for Guides

Convention	Usage
Courier New	Displays file locations, user entered text, and source code: C:\cd\icc\
Italics	Displays file names and reference documentation: Read about the <i>sourcefile.hex</i> file in the <i>PSoC Designer User Guide</i> .
[Bracketed, Bold]	Displays keyboard commands in procedures: [Enter] or [Ctrl] [C]
File >> Open	Represents menu paths: File >> Open >> New Project
Bold	Displays commands, menu paths, and icon names in procedures: Click the File icon and then click Open .
Times New Roman	Displays an equation: 2+2=4
Text in gray boxes	Describes cautions or unique functionality of the product.



Introduction

Introduction







This chapter describes how to install the CY3215-DK PSoC 1 In-Circuit Emulator Development Kit.

2.1 Kit Installation

To install the CY3215-DK kit, follow these steps:

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

Note You can also download the latest kit installer from http://www.cypress.com/go/CY3215-DK. Three different types of installers are available for download.

- a. CY3215-DK_ISO: This file (ISO image) is an archive file of the optical disc provided with the kit. You can use this to create an installer CD or extract information using WinRar or similar tools.
- b. CY3215-DK_Setup.exe: This executable file installs the contents of the kit CD, which includes PSoC Programmer, PSoC Designer, and user documents.
- c. CY3215-DK_SetupOnlyPackage.exe: This executable file installs only the kit contents, which includes kit user documents.
- 2. Click Install CY3215-DK to start the installation, as shown in Figure 2-1.

Figure 2-1. Kit Installer Startup Screen





Note If auto-run does not execute, double-click the *cyautorun.exe* file on the root directory of the CD, as shown in Figure 2-2. To access the root directory, click **Start > My Computer > CY3215-DK <drive:>**.

Figure 2-2. CD Root Directory

🗣 CY3215-DK (F:)				
Eile Edit ⊻iew Favorites Tools	Help			
🚱 Back 🔹 🕥 - 🏂 🔎 Se	arch 😥 Folders 🛄 🕶			
Address 🗇 F:\				💌 🄁 Go
Сүз215-DК	Documentation	Prerequisite	PSoC Designer	
PSoC Programmer	Reference	autorun Setup Information 1 KB	Cyautorun DAT File 1 KB	
Cypress Autorun Applet Cypress Semiconductor	setup 48 × 48 ICO File			

- 3. In the **InstallShield Wizard**, choose the folder location to install the setup files. You can change the location of the folder for the setup files using **Change**, as shown in Figure 2-3.
- 4. Click **Next** to launch the kit installer.

Figure 2-3. InstallShield Wizard

CY3215-DK - InstallShield	Wizard	\mathbf{X}
	Welcome to the InstallShield Wizard for CY3215-DK The InstallShield Wizard will install CY3215-DK on your computer. To continue, click Next.	
	Select folder where setup will install files. Install CY3215-DK to: C:\Program Files\CypressChange]
	< <u>B</u> ack <u>N</u> ext > Cancel	

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

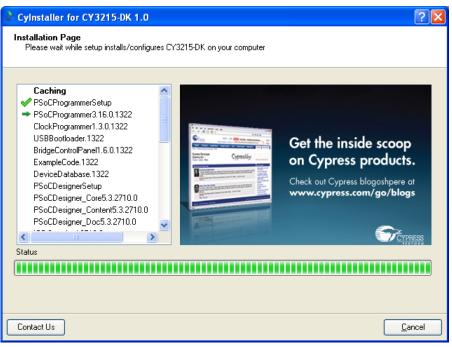


Figure 2-4. Installation Type Options

👶 CyInstaller for CY3215-DK 1.0	? 🗙
Product Installation Overview Choose the install type that best suits your needs	
Choose the type of installation Product: CY3215-DK Installation Type: Installs the most common features of CY3215-DK.	
Contact Us	ncel

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

Figure 2-5. Installation Page





9. Click Finish to complete the kit installation, as shown in Figure 2-6.

Figure 2-6. Installation Complete Page

🕹 Cyinstaller for CY3215-DK 1.0	? 🗙
© 2009 Cypress Semiconductor Corporation All rights reserved	
Contact Us	Einish

After software installation, verify that you have all hardware and drivers set up for the CY3215-DK kit by connecting the kit to your PC via its USB interface. Because this is the first time you have connected this board to this PC, initial drivers are installed. If you encounter a prompt related to Windows Logo Testing, click the **Continue Anyway** button to allow the driver installation to finish. Follow the instructions to complete the installation process.

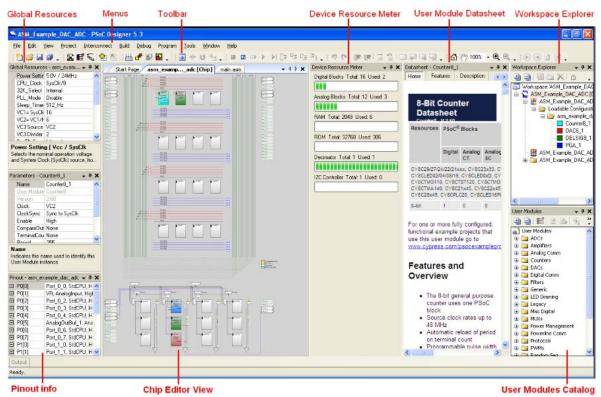


2.2 **PSoC Designer**

PSoC Designer is the revolutionary integrated design environment (IDE) that helps to customize PSoC 1 to meet specific application requirements. PSoC Designer software accelerates system bring-up and time-to-market.

- 1. To open PSoC Designer, click Start > All Programs > Cypress > PSoC Designer <version> > PSoC Designer <version>.
- 2. To create a new project in PSoC Designer, click File > New Project.
- 3. To open an existing project in PSoC Designer, click File > Open.

Figure 2-7. PSoC Designer Interconnect View



To experiment with the code examples, go to Code Examples on page 26.

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

See Additional Learning Resources on page 5 for links to PSoC Designer training.

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



2.3 **PSoC Programmer**

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

To successfully program the device, follow these steps:

1. Select the ICE-Cube connectivity in **Port Selection**, as shown in Figure 2-8.

Figure 2-8. PSoC Programmer Window

SoC Programmer	Power	
File View Options Help	File Load Program	
Port Selection	Programmer Utilities JTAG	
USB/0735C749	Programming Parameters File Path; File Is Not Present Programmer; USB/0735C749 Programming Mode; Reset ● Power Cycle ○ Power Detect Verification; ● On ○ Off Connector; ○ 5p ● 10p	
	Verification: O On O ff Connector: 5p ■ 10p AutoDetection: O On O ff Clock Speed: 1.6 MHz	
Device Family 60100	Programmer Characteristics Status Protocol: JTAG SWD ISSP 2C Voltage: ● 5.0 V 3.3 V 2.5 V 1.8 V	
CY7C60113-PVXC 🔽	Voltage: NA	
Actions	Results	^
Successfully Connect to USB/0735C749 at 2:00:17 PM Opening Port at 2:00 PM	IceCube	
Device set to CY7C60113-PVXC at 2:00:15 PM	8192 FLASH bytes	
Device Family set to 60100 at 2:00:15 PM Active HEX file set		
ACTIVE MEX IIIE SET	at	<u> </u>
For Help, press F1	Cor	inected 💦 💥

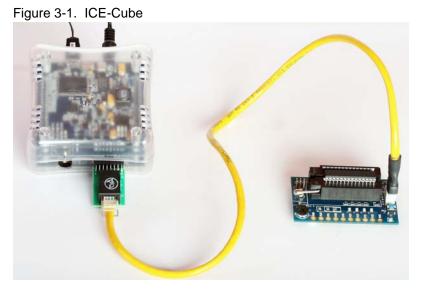
- 2. Click the File Load button to load the hex file.
- 3. Select the device and device family from the list.
- 4. Click the Program button to program the selected PSoC 1 device.
- 5. Close PSoC Programmer.

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



3.1 Connecting the ICE-Cube

PSoC Designer supports the ICE-Cube. This new in-circuit emulator replaces the ICE-4000 and the USB adapter for seamless USB connection, debugging, and programming.



3.1.1 Connect using a USB Port

The ICE-Cube connects to any computer using a standard USB 2.0 cable, included in Cypress development kits. To connect the ICE-Cube to your computer, plug the USB cable into your computer and attach the other end to the ICE-Cube.

The ICE-Cube is a plug-and-play device and it should be recognized automatically by any computer with PSoC Designer and PSoC Programmer installed. If a USB connection problem occurs, refer to Microsoft Windows Help for troubleshooting Windows connectivity issues.

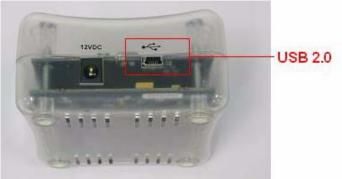


Figure 3-2. USB Port in ICE-Cube

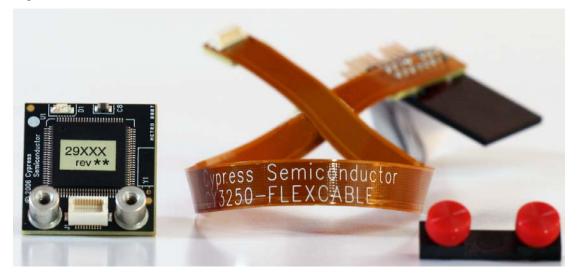


3.1.2 Connect using a Flex Cable

A flex cable is used to connect the ICE-Cube to the pod main board.

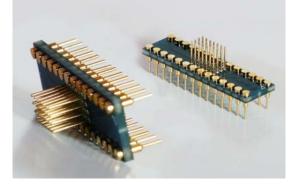
 Pod: Pods connect to target circuits via foot. As each pod contains a fully functional PSoC bondout device, pods may be used instead of devices for test purposes. Simply plug a pod into the circuit without connecting it to an ICE base station. The pod power LED lights up when it is powered and operational.

Figure 3-3. Flex Cable and 29000-28 DIP



Foot: A foot is used to connect the pod to the MiniEval board. Each foot has a pinout that models a PSoC, for example, a 28-pin DIP. A foot that emulates surface-mount components must be soldered to target circuits. The main board of the pod can then be attached or removed, as desired.

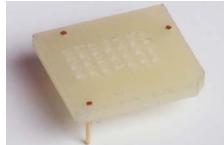
Figure 3-4. 28-Pin DIP Feet



 Plastic Mask: The plastic mask is used to orient the foot. Plastic masks are provided to expose only the pins that connect to the foot.



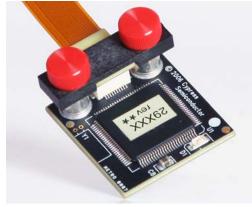
Figure 3-5. Plastic Mask



Before connecting the flex cable to the ICE-Cube, assemble the three pieces (pod, plastic mask, and foot), as shown in Figure 3-6.

- 1. Select a foot. The foot should match the pinout of the PSoC 1 device used in the target circuit.
- 2. Next, select the mask that matches the desired foot.
- 3. Insert the mask into the bottom of the pod, aligning the chamfered corners of the mask to the pin-1 triangle on the pod.
- 4. Insert the foot through the plastic mask. Use the alignment triangles to orient the foot to the pod.
- 5. Plug the pod into the MiniEval board via ZIF socket.

Figure 3-6. Assembled Pod with Flex Cable



Connect the other end of the flex cable to the ICE-Cube pod connector, as shown in Figure 3-7.
 Figure 3-7. Flex Pod Connected to ICE-Cube Pod



Pin No.	Pin Name	Pin Description
1	POD EXTRA3	Future use
2	GND	Ground
3	_	-
4	POD_OCDDE	POD_OCD even data I/O
5	GND	Ground
6	POD_OCDD)	POD_OCD odd data output
7	POD EXTRA1	Future use
8	POD_XRES	Reset signal (required only for Reset programming mode)
9	GND	Ground
10	POD_OCDHC	POD_OCD high speed clock output
11	GND	Ground
12	POD_OCDCC	POD_OCD CPU clock output
13	POD EXTRA4	Future use
14	PODVCC	Supply voltage
15	_	-
16	PODVCC	Supply voltage

Table 3-1.	Pin Description of Pod Connector

3.1.3 Connect using a Backward Compatibility Adapter

A backward compatibility adapter can be used to debug and program the PSoC 1 chips.

Figure 3-8. Backward Compatibility Adapter

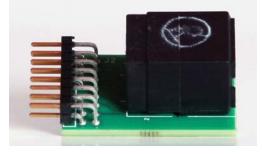
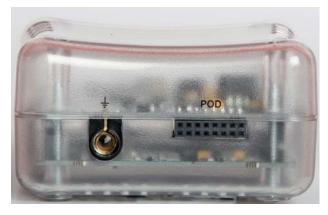


Figure 3-9. Connector in ICE-Cube

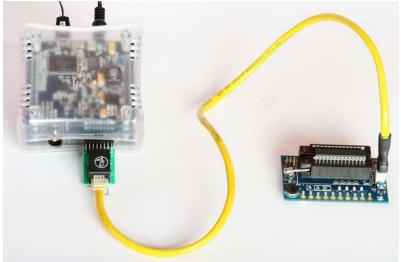




To program the device, follow these steps:

- 1. Using a USB cable, plug the ICE-Cube into the installed application. Make sure the ICE-Cube is powered on.
- 2. Connect the ISSP cable to the ICE-Cube through the backward compatibility adapter.
- 3. Place the 5-pin end of the ISSP cable on the 5-pin header(J2) of the MiniEval board.
- 4. Place the CY8C29466 chip on the MiniEval board via ZIF socket.
- 5. Launch PSoC Programmer.
- 6. Load the hex file using the **File Load** button.
- 7. Program it successfully.

Figure 3-10. Backward Compatibility Adapter Connected to ICE-Cube Pod



3.1.4 Software Configuration

After the physical connection is made, you are ready to configure the internal connection from the computer to the ICE. The ICE enables communication and debugging between PSoC Designer and the pod. To connect the ICE from within PSoC Designer, perform the following steps:

- 1. Confirm that the flex cable and pod are attached to the ICE-Cube.
- 2. Confirm that the ICE is powered from the power adapter.
- 3. Confirm that the USB cable is connected from the ICE-Cube to the PC.
- 4. Create a project using PSoC Designer. For more information, see My First Code Example on page 26.
- 5. To access the debugger subsystem, go to **Project > Settings > Debugger** and click the **Debugger** icon.
- Select the correct port from the drop-down window. The target board may either be powered by ICE-Cube or an external source. Select 5V as Pod Supply Voltage if the board is powered by ICE-Cube.
- 7. Click OK.



Figure 3-11. Debugger Project Setting

Settings	? <mark>- × -</mark>
Example_My_First_PSoC_Project Build Compiler Linker Chip Editor	Select ICE Device: ICE 0, Cube (SN: 0642C003)

8. On successful connection, you receive a notification in the Output tab of the Status window; a green indicator displays **Connected** in the lower-right corner of the program.

Figure 3-12. PSoC Designer Connected with ICE-Cube

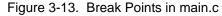
								-		
A: 00	X: 00	SP: 00	PC: 0000	F: 00	0.0 KHz	ICE 0	Connected	Ln 33	Col 2	INS

3.1.5 Debug a Project

To successfully debug a project, follow these steps:

- 1. Connect the ICE-Cube to the PSoC 1 development board.
- Click Start > All Programs > Cypress > PSoC Designer <version> > PSoC Designer <version>.
- Create a new project in PSoC Designer by clicking File > New Project.
 Note To open an existing project, click File > Open.
- 4. To connect the ICE-Cube and PSoC 1 development board, go to **PSoC Designer > Debug**.
- 5. Click Connect/Disconnect or press F9.
- 6. Right-click a line in the project from where the debugging process should start. The **Insert/Delete Breakpoint** option appears.





🕌 Exa	mple_My	r_First_PSoC_Project (Debugging) - PSoC Designer 5.3
1 🛍 🖻	j 🔒 🕯	▶
File	Edit V	iew Project Interconnect Build Debug Program Iools Window Help
	Start Pa	
User Modules	75	
od	76	
s	77	
_		* ClockSync = Sync to SysClk
	79	* Invertenable = Normal
	80	*
	81	·*************************************
	82	· · · · · · · · · · · · · · · · · · ·
		<pre>#include <m8c.h> // part specific constants and macros</m8c.h></pre>
	84	<pre>#include "PSoCAPI.h" // PSoC API definitions for all User Modules</pre>
	85	
	86	unsigned int i; // Varible used for delay
	87	
	88	void main(void)
	89	
0	90	<pre>PWM8_1_Start(); // Turn on the PWM to blink LED on P0.7</pre>
	91	<pre>LED_1_Start(); // Enable Software controlled LED</pre>
	92	
	93	
	94	// The following loop controls the software LED conneted to P1.7
	95	while(1)
	96	
	97 :	
0	98	LED_1_Invert(); //Switch the state of Software LED, if on turn it off,
	99	//if off turn it on
	100	- } //End of while(1)
	101	
		L}//End of main
	103	
14		II.
Ou	tput	

- 7. To view memory, registers, and watch variables at a particular location, go to **Debug > Windows**.
- 8. To start the debugging process, go to **Debug > Go** or press F5.

Use one of the following options for the debugging process:

- a. Debug > Step Over (or press F8): Steps over next statement
- b. Debug > Step Into (or press F9): Steps into next statement
- c. Debug > Step Out (or press Shift + F11): Steps out of current function
- d. Debug > Step ASM (or press Shift + F10): If the current line is C code, the line is located in the first file and that line is executed



₩.	Examp	ole_My	/_First_	PSoC_P	roject	t (Debug	ging) - PS	oC De	signer 5.3										
1) 🞽		· .	S 6	£.	雷衝		⊡≽	▶ ▶I [́⊒ '	te de ser	- 🖽 🖻	1 🗊 🗸	•	🚽 🗗 😼	9	🎽 🛊 👘	Ξ.	😫 💷 🖓 ಢ 🤿	- 🖧 १९ 100% - €
<u>E</u> il	e <u>E</u> d	lit <u>V</u>	iew	<u>P</u> roject	Inte	erconnec	t <u>B</u> uild	Det	ug P <u>r</u> ogra	m <u>T</u> ools	<u>W</u> indow	<u>H</u> elp	_						
Use	S	tart Pa	ge	example_	_my	project [[Chip] n	1i 🗔	<u>W</u> indows			•		Memory		Ctrl+Alt+M			↓ 4
User Modules		75	*		P۱	ulseW	lidth		Output file	;		+	i7x	Registers		Ctrl+Alt+G			
dules		76	*				етуре	1.111	Execute Pro	gram		F4		Watch		Ctrl+Alt+W			
		77 78	*			nterr lockS	uptTy		Go			F5		Locals		Ctrl+Alt+L			
		79	*				Enabl		Run To Cu	sor	Ct	rl+F5		Break Points		Ctrl+Alt+B			
		80	*				2110002		Halt		Shi	ft+F5	2	Events		Ctrl+Alt+E			
		81	**	****	k nik nik n	****	****		Reset		Ctrl+Shi	ft+F5	~	Trace		Ctrl+Alt+T	k rn	******	/
		82						[]	Step Over			F8		USB Endpoint	Data	Ctrl+Alt+U			
		83				<m8c.< td=""><td></td><td></td><td>Step Into</td><td></td><td></td><td>F11</td><td></td><td>cs and ma</td><td></td><td></td><td>-</td><td></td><td></td></m8c.<>			Step Into			F11		cs and ma			-		
		84 85	#1	nciud	te .	"PSoC	API.h		Step Out		Shift	+F11	ıs	for all (Jser	Modules	5		
		86	un	signe	ed :	int i	;		Step ASM			+F10	lay						
		87							Connect / I	× .	51111	F9	-						
		88	vo	id ma	ain	(void	l)	*	Download t		c1 -	ft+F9							
	_	89	3 {		_				Refresh M8		snr	rt+F9							
	•	90 91			_	_	rt();							<mark>LED on P(</mark> d LED	0.7				
		91		LEI	<u></u>	_Star	ι();		<u>T</u> race Mod			•	Te	a LED					
		93						-	Get next tra										
		94		11	The	e fol	lowin	d 🕑	Get all trace	data			re	LED conne	eted	to P1.7	7		
		95		whi	ile	(1)		÷	Configure	JSB Data Tr	acking								
		96	Т	{	_							-	Ξ.					1 0000 1	
		97 (98					.=0;1< Inver											and CPU cl f on turn i	
	-	99			101	<u></u>	THAT	5()	,		'if off				DICWO	are heb,	, 1.	I ON CUIN I	c orr,
		100	_	} /	//E1	nd of	whil	e (1)										
		101																	
			-}/	/End	of	main	L												
		103																	

Figure 3-14. Debug Options

3.1.5.1 Break Points

The break point feature allows you to stop program execution at predetermined address locations. When a break point is encountered, the program stops at the address of the break point, without executing the address code. The program is restarted using the available menu or icon options.

To set break points, first open the file to debug. Right-click the mouse at specific points and select **Insert Break Point**. You can view and remove active break points in the Break Points window. To open the Break Points window, select **Debug > Windows > Break Points**.

Figure 3-15. Break Points Window

Break Points	
X 🕺 🏹 📷	
main.c, line 90	
main.c, line 98	
main.c, line 95	
Output	
	A: 04 X: 63 SP: 02 PC: 0410 F: C4 12

3.1.5.2 CPU and Register Views

During debugging, you can read and write in five areas: CPU registers, bank registers 0, bank registers 1, RAM, and flash. The CPU registers are shown in their own window (**Debug > Windows > Registers**) and in the notification area at the bottom of PSoC Designer. The other four areas can be viewed in the Memory Window (**Debug > Windows > Memory**). Select one of the four memory areas from the **Address Space** box.



CPU Registers: This window allows you to examine and change the contents of the CPU registers. Data is entered in hexadecimal notation. CPU register values can be viewed across the bottom of PSoC Designer.

Figure 3-16. CPU Register in Memory Window



RAM: RAM locations can be modified by clicking the data at the specific location and typing in the new value. Data is entered in hexadecimal notation.

Flash: The flash window displays the data stored in flash. This is the program memory; it is readonly.

Bank Registers 0 and 1: You can scroll through the register bank to view the values in the register bank. Type a new value into the Offset to scroll directly to that offset. Click next to a value and type a new value for the register.

Figure 3-17. Memory Window

Memory									
Address:	0x0000)	•	Colu	umn:	s 8		 Views 	
RAM	FLAS	н							
0018	00 00	00	00	00 0	0 0	0 0)		
0020	00 00	00	00	00 0	0 0	0 0)		
0028	00 00	00	00	00 0	0 0	0 0)		
0030	00 00	00	00	00 0	0 0	0 0)		
0038	00 00	00	00	00 0	0 0	0 0)		
Output									
									A: 04 X: 63 SP: 02 PC: 0410 F: C4 12.00 N

3.1.5.3 Watch Variables

To set watch variables, right-click a variable in a source file and select **Add Watch**. You can also select **Global Variables**. Right-click **Add**, **Delete**, or **Properties** in the Watch/Global Name window to add, delete, or modify values.

Figure 3-18. Watch Variables Window

lame	Value		Location			Туре	
🚰 i	0		RAM 0x0005			unsig	ned int
Dutput							
		A: 04 X: 63	SP: 02 PC: 0410	F: C4	12.00 MHz	ICE 0	Connected

3.1.5.4 Trace

The Trace window is displayed when **Debug > Windows > Trace** is chosen. It displays a continuous, configurable listing of project symbols and operations from the last breakpoint. The trace shows symbolic, rather than address data, to enhance readability.



Each time the program executes, the trace buffer is cleared. When the trace buffer becomes full, it continues to operate and overwrite old data.

Figure 3-19. Trace Window

1 PC / SYMBOL INSTRUCTION 1 PC / SYMBOL INSTRUCTION 2 0138 MOV REG[E2h], 00h 3 0136 AND F, EFh 4 0133 OR REG[E7h], 80h 5 0130 MOV REG[E0h], 02h 6 012E OR F, 10h 7 012B MOV REG[E0h], 00h 8 0129 AND F, EFh 9 0127 OR F, 10h 10 0126 POP A 11 00DA JZ 0126h 12 00D9 INC A 13 0D7 MOV REG[D5h], A 14 00D6 ROMX 15 00D5 PUSH A 16 00D2 MOV REG[E3h], 00h	Star	rt Page example_myproject [Chip] ma	in.c boot.asm Trace.txt
2 0138 MOV REG[E2h], 00h 3 0136 AND F, EFh 4 0133 OR REG[E7h], 80h 5 0130 MOV REG[E0h], 02h 6 012E OR F, 10h 7 012B MOV REG[E0h], 00h 8 0129 AND F, EFh 9 0127 OR F, 10h 10 0126 POP A 11 00DA JZ 0126h 12 00D9 INC A 13 00D7 MOV REG[D5h], A 14 00D6 ROMX 15 00D5 PUSH A	1	PC / SYMBOL	INSTRUCTION
3 0136 AND F, EFh 4 0133 OR REG[E7h], 80h 5 0130 MOV REG[E0h], 02h 6 012E OR F, 10h 7 012B MOV REG[E0h], 00h 8 0129 AND F, EFh 9 0127 OR F, 10h 10 0126 POP A 11 00DA JZ 0126h 12 00D9 INC A 13 00D7 MOV REG[D5h], A 14 00D6 ROMX 15 00D5 PUSH A	1	PC / SYMBOL	INSTRUCTION
4 0133 OR REG[E7h], 80h 5 0130 MOV REG[E0h], 02h 6 012E OR F, 10h 7 012B MOV REG[E0h], 00h 8 0129 AND F, EFh 9 0127 OR F, 10h 10 0126 POP A 11 00DA JZ 0126h 12 00D9 INC A 13 00D7 MOV REG[D5h], A 14 00D6 ROMX 15 00D5 PUSH A	2	0138	MOV REG[E2h], 00h
5 0130 MOV REG[E0h], 02h 6 012E OR F, 10h 7 012B MOV REG[E0h], 00h 8 0129 AND F, EFh 9 0127 OR F, 10h 10 0126 POP A 11 00DA JZ 0126h 12 00D9 INC A 13 00D7 MOV REG[D5h], A 14 00D6 ROMX 15 00D5 PUSH A	3	0136	AND F, EFh
6 012E OR F, 10h 7 012B MOV REG[E0h], 00h 8 0129 AND F, EFh 9 0127 OR F, 10h 10 0126 POP A 11 00DA JZ 0126h 12 00D9 INC A 13 00D7 MOV REG[D5h], A 14 00D6 ROMX 15 00D5 PUSH A	4	0133	OR REG[E7h], 80h
7 012B MOV REG[E0h], 00h 8 0129 AND F, EFh 9 0127 OR F, 10h 10 0126 POP A 11 00DA JZ 0126h 12 00D9 INC A 13 00D7 MOV REG[D5h], A 14 00D6 ROMX 15 00D5 PUSH A	5	0130	MOV REG[E0h], 02h
8 0129 AND F, EFh 9 0127 OR F, 10h 10 0126 POP A 11 00DA JZ 0126h 12 00D9 INC A 13 00D7 MOV REG[D5h], A 14 00D6 ROMX 15 00D5 PUSH A	6	012E	OR F, 10h
9 0127 OR F, 10h 10 0126 POP A 11 00DA JZ 0126h 12 00D9 INC A 13 00D7 MOV REG[D5h], A 14 00D6 ROMX 15 00D5 PUSH A	7	012B	MOV REG[E0h], 00h
10 0126 POP A 11 00DA JZ 0126h 12 00D9 INC A 13 00D7 MOV REG[D5h], A 14 00D6 ROMX 15 00D5 PUSH A	8	0129	AND F, EFh
11 00DA JZ 0126h 12 00D9 INC A 13 00D7 MOV REG[D5h], A 14 00D6 ROMX 15 00D5 PUSH A	9	0127	OR F, 10h
12 00D9 INC A 13 00D7 MOV REG[D5h], A 14 00D6 ROMX 15 00D5 PUSH A	10	0126	POP A
13 00D7 MOV REG[D5h], A 14 00D6 ROMX 15 00D5 PUSH A	11	00DA	JZ 0126h
14 00D6 ROMX 15 00D5 PUSH A	12	00D9	INC A
15 00D5 PUSH A	13	00D7	MOV REG[D5h], A
	14	00D6	ROMX
16 00D2 MOV REG[E3h], 00h	15	00D5	PUSH A
	16	00D2	MOV REG[E3h], 00h
17 00D0 ADC A, 00h	17	00D0	ADC A, 00h
18 00CF INC X			
19 0105 JMP 00CFh			
20 0103 JNZ 00EFh			
21 0101 DEC [0003h]			
22 0100 POP A	22		
23 00F9 JNZ 0100h	23		
24 00F6 TST [0004h], FFh			
25 00F4 MVI [0004h], A			
26 00F3 ROMX			
27 00F2 PUSH A			
28 00F0 ADC A, 00h			
29 OOEF INC X			
		0102	THY OOREN

A: 04 X: 63 SP: 02 PC: 0410 F: C4 12.00 MHz

3.1.5.5 Locals

A separate window is available for local variables. Whenever execution halts, the local variables are updated to the current value.

Value			Loc	ation		Туре	
8298			RA	M 0x0007		unsig	ned int
3			RA	M 0x0005		unsig	gned int
	8298 3	8298 3	8298 3		8298 RAM 0x0007 3 RAM 0x0005		8298 RAM 0x0007 unsig

WARNING: The time taken to execute a system supervisory call (SSC) such as FlashRead/Write, Table Read, and Erase Block is significantly more while emulating a code through ICE-Cube when compared to the time taken when running the code on chip.



3.2 **PSoC Programmer**

PSoC Programmer is used as a standalone application to program PSoC devices. It can be launched within PSoC Designer or accessed from the desktop as a standalone program.

Figure 3-21. PSoC Programmer Interface

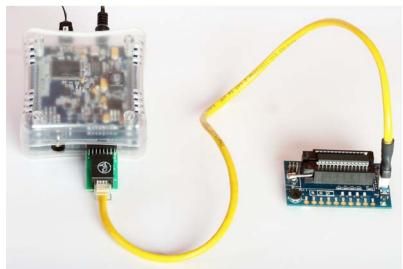
PSoC Programmer	
File View Options Help	
🖆 · 衤 🎯 🕒	
Port Selection	rogrammer Utilities JTAG
MINIProg1/03AFA04E150B	Programming Parameters
	File Path: C\perforce\apps\PSoC1Kits\CY3215-DK\project\VTExample_My_First_PSoC_ProjectExample_My_First_PSoC_ProjectExample_My_First_PSoC_Project
	Programmer. MINIProg1/03AFA04E150B
	Programming Mode: O Reset O Power Cycle O Power Detect
	Verification: On Off Connector: Connect
Device Family	<u>AutoDetection:</u>
	Programmer Characteristics Status Status Status Status
Destau	Protection of TAG O SWD @ 1501 @ 12C
CY8C29466-24PXI v	Voltage:
Actions	Results
Program Finished at 3:22:46 PM	
3:22:46 PM	Programming Succeeded
	Doing Checksum
	Doing Protect
	Verify Succeeded
	Verify Starting
	Programming Succeeded
	Programming Starting
	Erase Succeeded
Device set to CY8C29466-24PXI at 3:21:59 PM	32768 FLASH bytes
Device Family set to 29x66 at 3:21:59 PM	
	Automatically Detected Device: CY8C29466-24FXI
Program Requested at 3:21:59 PM	
Successfully Connecte	d
to	MINI Version 1.80
MINIProg1/03AFA04E150	•
For Help, press F1	PASS INDEPOWERED Connected

To program a target board using PSoC Programmer via ICE-Cube pod connector, follow these steps:

- 1. Set up all hardware, including the device to be programmed.
- 2. Disconnect power from the target board.
- 3. Launch PSoC Programmer:
 - a. From the desktop, click Windows Start > All Programs > Cypress > PSoC Programmer.
 b. From within PSoC Designer, click Program > Program Part.
- 4. Click File Load to select a file for programming.
- 5. Select the port used to connect the programmer.
- 6. Select the device family and device used to generate the hex file.
- 7. Select Reset under programming mode.
- 8. Apply power to the target board.
- 9. Click **Program** to start device programming.
- 10. The action window reports the status and success of programming.











4.1 My First Code Example

4.1.1 Project Objective

This project is used to demonstrate blinking an LED at a varying duty cycle using a hardware pulse width modulator (PWM). Another LED is caused to blink using a software delay. The clock dividers VC1, VC2, and VC3 are used to divide the 24-MHz system clock by 16, 16, and 256, respectively. The resulting 366-Hz clock is used as the input to an 8-bit PWM. This in turn produces an LED blink period of 1.4 Hz. The project also demonstrates how an LED can toggle on/off with a delay of approximately 1 second.

The following user modules are used in this project:

- PWM An 8-bit PWM is used to generate a 366-Hz signal. An LED is connected to the PWM output. This LED blinks at 1.4 Hz.
- LED This module is used to toggle an LED on/off.

4.1.2 Creating My First PSoC 1 Project

- 1. Open PSoC Designer 5.3.
- 2. To create a new project, click File > New Project. The New Project window opens.
- 3. In this window, select the **Chip-level** icon. Name the project **Example_My_First_PSoC_Project**, as shown in Figure 4-1.
- 4. Click **Browse** and navigate to the directory in which the project should be created.

Figure 4-1. New Project Window

New Project	? <mark>- x-</mark>
Project types:	
<u>N</u> ame:	Example_My_First_PSoC_Project
Location:	C:\
Workspace:	Create new Workspace
Workspace Name:	Example_My_First_PSoC_Project Vertex directory for workspace
Project Creation:	New Project
<u>T</u> arget Device:	CY8C24894-24LTXI Device Catalog
Generate 'Main' file using:	C
Creates an empty C - base This project type supports	I Chip-level project for CY8C24894-24LTXI device. Jser Module selection and placement.

5. In the project creation drop down menu select new project, as shown in Figure 4-2



6. In this window, under Select Target Device, click Device Catalog.

Figure 4-2. Select Project Type Window

Project types: Chip-level Name: Example_My_First_PSoC_Project Location: C:\ Workspace: Create new Workspace Workspace Name: Example_My_First_PSoC_Project Workspace Name: Example_My_First_PSoC_Project Project Creation: New Project Target Device: CY8C24894-24LTXI Generate 'Mgin' file using: C Creates an empty C - based Chip-level project for CY8C24894-24LTXI device. This project type supports User Module selection and placement.	New Project	? 💌
Name: Example_My_First_PSoC_Project Location: C:\ Workspace: Create new Workspace Workspace Name: Example_My_First_PSoC_Project Workspace Name: Example_My_First_PSoC_Project Project Creation: New Project Iarget Device: CY8C24894-24LTXI Generate 'Majn' file using: C Creates an empty C - based Chip-level project for CY8C24894-24LTXI device.	Project types:	
Location: C:\ Workspace: Create new Workspace Workspace Name: Example_My_First_PSoC_Project Workspace Name: Example_My_First_PSoC_Project Project Creation: New Project Iarget Device: CY8C24894-24LTXI Generate 'Main' file using: C Creates an empty C - based Chip-level project for CY8C24894-24LTXI device.		
Workspace: Create new Workspace Workspace Name: Example_My_First_PSoC_Project Workspace Name: Example_My_First_PSoC_Project Project Creation: New Project Iarget Device: CY8C24894-24LTXI Generate 'Majn' file using: C Creates an empty C - based Chip-level project for CY8C24894-24LTXI device.	<u>N</u> ame:	Example_My_First_PSoC_Project
Workspace Name: Example_My_First_PSoC_Project Image: Create directory for workspace Project Creation: New Project Image: Create directory for workspace Iarget Device: CY8C24894-24LTXI Device Catalog Generate 'Mgin' file using: C Image: Create an empty C - based Chip-level project for CY8C24894-24LTXI device.	Location:	C:\
Project Creation: New Project Target Device: CY8C24894-24LTXI Generate 'Mgin' file using: C Creates an empty C - based Chip-level project for CY8C24894-24LTXI device.	Workspace:	Create new Workspace 👻
Iarget Device: CY8C24894-24LTXI Device Catalog Generate 'Mgin' file using: C Creates an empty C - based Chip-level project for CY8C24894-24LTXI device.	Workspace Name:	Example_My_First_PSoC_Project Create directory for workspace
Generate 'Mgin' file using: C Creates an empty C - based Chip-level project for CY8C24894-24LTXI device.	Project Creation:	New Project
Creates an empty C - based Chip-level project for CY8C24894-24LTXI device.	Target Device:	CY8C24894-24LTXI Device Catalog
Creates an empty C - based Chip-level project for CY8C24894-24LTXI device. This project type supports User Module selection and placement	Generate "Main" file using:	C -
	Creates an empty C - base This project type supports	d Chip-level project for CY8C24894-24LTXI device. User Module selection and placement.
QK Cancel		

- 7. The Device Catalog window opens, then scroll down to the CY8C29466, CY8C29566,... section.
- 8. For this project, click CY8C29466-24PXI and then click Select; see Figure 4-3.

Figure 4-3. Device Catalog Window

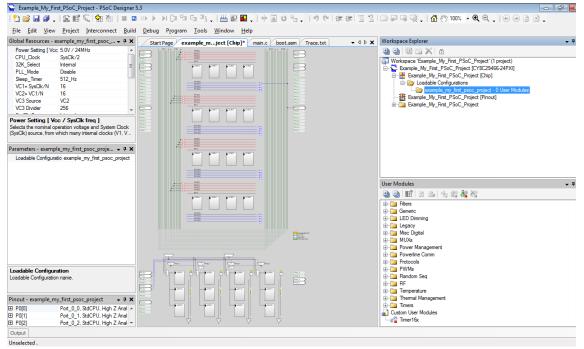
De	vic	e Catalog - Chip-level	-	-	-								
)ev	vice Type: All Devices 👻	B	Co <u>m</u> pare	Devices	17	Rese	t Fi	ind				
(Compare	Part Number	Pin Count	Package Type	Analog Blocks	Digital Blocks	CapSense	Flash	RAM	IO Count	Supply Voltage	SMP	USB Interface
	-	Filters:	T	T	T	T	T	T	T	T	T	T	T
		CY8C28513-24AXI	44	TQFP	0 + *4	12	Yes	16K	1K	40	3.0 to 5.25	Yes	N/A
		CY8C28623-24LTXI	48	QFN	6	12	No	16K	1K	44	3.0 to 5.25	Yes	N/A
		CY8C28433-24PVXI	28	SSOP	6 + *4	12	Yes	16K	1K	24	3.0 to 5.25	Yes	N/A
E		CY8C28533-24AXI	44	TQFP	6 + *4	12	Yes	16K	1K	40	3.0 to 5.25	Yes	N/A
E		CY8C28243-24PVXI	20	SSOP	12	12	No	16K	1K	16	3.0 to 5.25	Yes	N/A
E		CY8C28643-24LTXI	48	QFN	12	12	No	16K	1K	44	3.0 to 5.25	Yes	N/A
E		CY8C28445-24PVXI	28	SSOP	12 + *4	12	Yes	16K	1K	24	3.0 to 5.25	Yes	N/A
E		CY8C28545-24AXI	44	TQFP	12 + *4	12	Yes	16K	1K	40	3.0 to 5.25	Yes	N/A
		CY8C28645-24LTXI	48	QFN	12 + *4	12	Yes	16K	1K	44	3.0 to 5.25	Yes	N/A
E		CY8C28452-24PVXI	28	SSOP	12 + *4	8	Yes	16K	1K	24	3.0 to 5.25	Yes	N/A
		CY8C29466-24PXI	28	PDIP	12	16	No	32K	2K	24	3.0 to 5.25	Yes	N/A
E		CY8C29466-24PVXI	28	SSOP	12	16	No	32K	2K	24	3.0 to 5.25	Yes	N/A

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

10.By default, the project opens in chip view, as shown in Figure 4-4.



Figure 4-4. Default View



11. In the User Modules window, expand the **PWMs** folder. In this folder, right-click on **PWM8** and select **Place**. The user module (UM) is placed in the first available digital block.

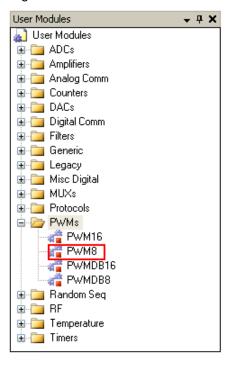


Figure 4-5. User Modules Window

12. Configure the PWM8_1 properties, as shown in the following figure.

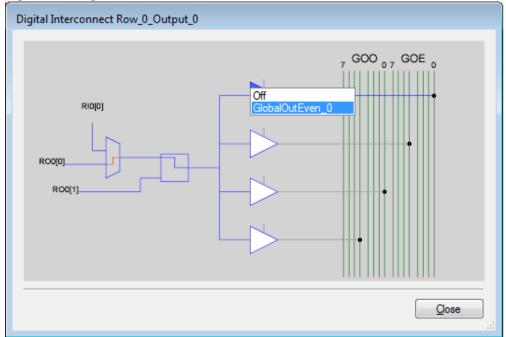


Name	PWM8_1							
User Module	PWM8							
Version	2.60							
Clock	VC3							
Enable	High							
CompareOut	Row_0_Output_0							
TerminalCountOut	None							
Period	100							
PulseWidth	50							
CompareType	Less Than Or Equal							
InterruptType	Terminal Count							
ClockSync	Sync to SysClk							
InvertEnable	Normal							
ame								
	ed to identify this User Module in.							

Figure 4-6.	PWM8 User Module Properties
-------------	-----------------------------

13.Click on the LUT node on Row_0_Output_0 to open the Digital Interconnect window.14.In this window, configure Row_0_Output_0_Drive_0 to connect to GlobalOutEven_0.

Figure 4-7. Digital Interconnect Window

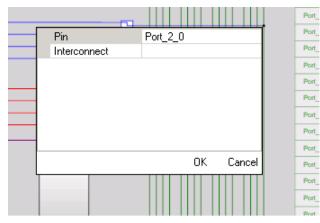


15.Click Close.

16.Click GlobalOutEven_0. In the window that appears, configure the pin for Port_2_0.

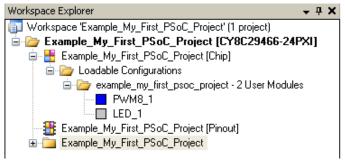


Figure 4-8.	Configur Pin for Port_	_2_(0
-------------	------------------------	------	---



- 17. Click OK to continue.
- 18. In the User Modules window, expand the Misc Digital folder. In this folder, right-click LED and select Place; this adds the UM to the project. This UM does not use digital or analog blocks. It appears in Workspace Explorer > Example_My_First_PSoC_Project[CY8C29466-24PXI] > Example_My_First_PSoC_Project[Chip] > Loadable Configurations > example_my_first_psoc_project 2 User Modules.

Figure 4-9. Workspace Explorer



19. Configure the LED properties, as shown in the following figure.

arameters - LED_1 🚽 🗸 🛪					
Name	LED_1				
User Module	LED				
Version	1.40				
Port	Port_2				
Pin	Port_2_1				
Drive	Active High				

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

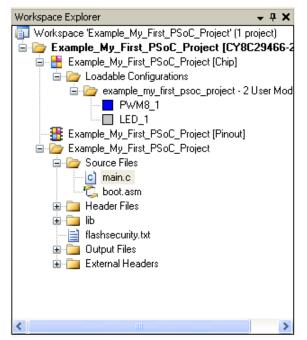


Glo	Global Resources - example_my_first_psoc_p 👻 📮 🗙					
	Power Setting [Vcc	5.0V / 24MHz				
	CPU_Clock	SysClk/2				
	32K_Select	Internal				
	PLL_Mode	Disable				
	Sleep_Timer	512_Hz				
	VC1= SysClk/N	16				
	VC2= VC1/N	16				
	VC3 Source	VC2				
	VC3 Divider	256				
	SysClk Source	Internal				
	SysClk*2 Disable	Yes				
	Analog Power	SC Off/Ref Low				
	Ref Mux	(Vdd/2)+/-(Vdd/2)				
	AGndBypass	Disable				
	Op-Amp Bias	Low				
	A_Buff_Power	Low				
	SwitchModePump	OFF				
	Trip Voltage [LVD (S	4.81V (5.00V)				
	LVDThrottleBack	Disable				
	Watchdog Enable	Disable				
Ref Mux Selects the range and accuracy of various analog references. This sets the analog ground and peak-to-p						

Figure 4-11.	Global	Resources	Window

21. Open the existing *main.c* file in Workspace Explorer. Replace the existing *main.c* content with the content of the *My_First_Example_Project_Main.c* file, which is available as an attachment to this PDF document.

Figure 4-12. Workspace Explorer Window



22. Save the project.

23. Click Build > Generate/Build 'Example_My_First_PSoC_Project' .



- 24. Connect the ICE-Cube to the PC. Connect the ICE-Cube to the MiniEval board, as explained in Connecting the ICE-Cube on page 14.
- 25. In PSoC Designer, select **Debug > Connect/Disconnect** to connect the ICE to PSoC Designer.
 - Figure 4-13. Connecting ICE to PSoC Designer

5	Example_I	My_Fi	rst_PSoC_Project - PSoC Designer S	i.3				
1) 📂 🔒	Ø.,	. 265 (186) - 5)) [I 🗄 🤤	= 🖹 🗸 🛗 🗗	-	* 🗉 U 💁 🚬 🔊 🤉
<u>E</u> il	e <u>E</u> dit	<u>V</u> iew	<u>P</u> roject <u>I</u> nterconnect <u>B</u> uild	Det	oug P <u>r</u> ogram	<u>T</u> ools <u>W</u> indow	<u>H</u> elp	
Use	Start	Page	example_myproject [Chip]		<u>W</u> indows		•	
er Mo	:	1 🖂 /	/******	1	Output files		•	*****
User Modules		2	* File Name: main.c		Execute Program	1	F4	1
		3 1 4 1	<pre>version: 1.1</pre>		Go		F5	
		5	version: 1.1	⊳I	Run To Cursor	Ctrl	+ F5	
			Description:		Halt	Shift	+ F5	
		7	• This project is u	s 📻	Reset	Ctrl+Shift	+F5	, an LED at a va
		3 3	' using a hardware		Step Over		F8	
		9		5	Step Into		F11	
	1		* Note: * Any unusual or n		Step Out	Shift+		be noted here.
	1		* wise, this secti		Step ASM	Shift+		, be noted nere.
	1	3 1		:=				
	1	4 1	Vowner:	**	Connect / Discor	nnect	F9	
	1	5 1	* KUK.		Download to em	ulator Shift	+ F9	
	1	6 1	k -	U	Refresh M8C Vie	W5		
	1	7 /	* Related Document:		Trace Mode		•	1
	1		k	4	Get next trace da	ta		
	1		Hardware Dependen	9	Get all trace data	1		
	2		* CY3215-DK - PSoC		Configure USB D	ata Tracking		lopment.
	2		Code Tested With:		comgate 000 D	ata macking m		1

26.On successful connection, the status bar at the bottom of the PSoC Designer IDE shows the status as 'Connected'.

Figure 4-14. ICE-Cube Connected

								-		
A: 00	X: 00	SP: 00	PC: 0000	F: 00	0.0 KHz	ICE 0	Connected	Ln 33	Col 2	INS

27. If there is an error, ensure that the Debugger settings in **Project > Settings** match the following figure.



Figure 4-15. Debugger Settings

Marchael Project Settings	? X
Example_My_First_PSoC_Project Build Compiler Unker Debugger Chip Editor	Select ICE Device: ICE 0, Cube (SN: 0642C003)
	QK Cancel

Note The pod may be powered by the ICE-Cube or through external power. If external power is used, ensure that the power is on for the ICE to get connected.

28.Click **Debug > Execute Program** or the keyboard shortcut **F4**. The project starts compiling; it is downloaded to the pod and starts running.

4.1.3 Verify Output

Verify the output as follows:

- 1. Two LEDs connected to P2[0] and P2[1] start blinking.
- Open main.c file, right-click any line in the while(1) loop in the code and select Insert/Delete Breakpoint to include a breakpoint at that point. The execution of the code stops at the point where breakpoint is inserted (LED controlled by software (LED connected to P2[1]) stops blinking). The line is highlighted in yellow when the code execution stops.
- At this point, you may either continue running the code by selecting Debug > Run (keyboard shortcut F5) or step through the code by selecting either Debug > Step Over (keyboard shortcut F8) or Debug > Step Into (keyboard shortcut F11).

Note The Step Over command executes statement one by one and does not enter into the function calls whereas the Step Into command enters the function calls as well.

Use the debug windows (see Debug a Project on page 19) to explore the other debugging features offered in PSoC designer. For more details on debugging, debug menu options, and debug strategies, see **Help > Help Topics > Debugger**.

Revision History



Document Revision History

Documen	Document Title: CY3215-DK PSoC® 1 In-Circuit Emulator Development Kit Guide								
Document Number: 001-66514									
Revision	ECN#	Issue Date	Origin of Change	Description of Change					
**	3128877	01/27/2011	RKPM	Initial version of kit guide					
*A	3256318	05/11/2011	RKPM	Added Code Examples chapter. Updated images in section 2.1. Content updates throughout the document.					
*B	3508496	01/30/2012	PAVA	Added information on the devices that CY3215-DK supports in the Intro- duction chapter. Updated Figure 2-2.					
*C	3915589	02/27/2013	KUK	Updated images in chapters 2, 3, and 4. Minor content updates throughout the document.					
*D	4241128	01/09/2014	PAVA	No content updates; sunset review					
*E	5629678	02/13/2017	AARA	Updated logo and copyrights/disclaimer. Moved revision history to the end of the document.					