Release Notes SRN067

PSoC Designer Version 5.0 Service Pack 3a

Release Date: February 6, 2009

Thank you for your interest in PSoC Designer™ version 5.0 Service Pack 3a. These release notes list installation requirements and describe software updates and changes.

System Requirements and Recommendations

System Requirements	Minimum	Recommended
▪ Processor Speed	1 GHz	2 GHz Dual Core
▪ MB of RAM	256 MB	1 GB
▪ MB of Free Hard Drive Space	600 MB	1 GB
▪ Screen Resolution	1024x768	1280x1024
▪ CD-ROM Drive	✓	✓
▪ USB Port, preferably USB 2.0	✓	✓
▪ Windows® XP (SP2 or 3), or Vista	✓	✓
▪ Microsoft Internet Explorer 6.0 (See note in the installation section)	6.0 (SP1)	7.0
▪ .NET Framework	2.0	2.0 (SP1 or higher)
▪ Adobe Reader (For Viewing of .PDF Documentation)	6	9
▪ PSoC Programmer	3.00.0.92	3.00.0.93

Updates

Check http://www.cypress.com/psocdesigner for the latest downloads of software and documentation.

New for PSoC Designer 5.0 Service Pack 3a

PSoC Designer 5.0 allows you to create chip-level (PSoC Designer) projects, or system-level (Express) projects that can also be used as chip-level projects. Included in Service Pack 3a are many bug fixes and support for many new product families listed below.

Please note that in many cases, new product support may be in the beta test phase. This is called out below (beta) and indicates that production release support will follow in the near future.

New Product and Kit Support Included in PSoC Designer 5.0 Service Pack 3a

Service Pack 3 erroneously included two unsigned drivers that were intended for PSoC Programmer. Service Pack 3a removes these two drivers. There are no other differences between Service Pack 3 and Service Pack 3a.

The following is a list of all the new product support provided in Service 3. For specific new user module and product device model numbers, see the appropriate sections below.

- CY8CLED03Dxx and CY8CLED04Dxx Product Family (production)
- CY8C2xx45 family (beta)

Installation

If an ICE-Cube is connected to the installation machine it must be disconnected and then reconnected after installing PSoC Designer 5.0 SP3.

Note: PSoC Designer SP3SP3 is not presently compatible with the internet browser, Microsoft Internet Explorer 8 beta (any beta version). If Microsoft Internet Explorer 8 beta is installed on your system, it must be uninstalled and an earlier version of Microsoft Internet Explorer installed in its place. Refer to the Microsoft Internet Explorer 8 beta for instructions of how to perform this operation. When the final release of IE8 is available, this issue will be reexamined to see if running PD 5.0 is an ongoing problem, or not.

To Install:

1. Shut down any currently running instances of PSoC Designer.
2. If PSoC Designer 5.0 is currently installed, uninstall it. Click Start, click Control Panel, and then double click Add or Remove Programs.
3. Download and install PSoC Designer 5.0 SP3 by running the provided installer.
4. Download and install the latest PSoC Programmer 3.00 by running the provided installer.

   Note: During the installation, if you have not already installed the HI-TECH compiler v9.61 PL5, a dialog will present you with the option of upgrading to the new version. The patch is necessary to support new devices and features in PSoC Designer 5.0 SP3.

   To revert to an earlier version of the HI-TECH compiler after an update is installed, select the compiler version under **Tools → Options → Compiler** in PSoC Designer.

   If you choose to decline the compiler update, you must manually update the psoc.ini file to add device support before you can compile projects that use the new devices. The HI-TECH "psoc.ini" file is found in the HI-Tech installation folder. The default location of the psoc.ini is here:

   `C:\Program Files\HI-TECH Software\HCPSOC\PRO\9.61\dat\psoc.ini`

   The default location of the replacement "psoc.ini" file that adds support for the new devices is here:

   `C:\Program Files\Cypress\Common\CypressSemiBuildMgr\tools\psoc.ini`

Part Support

PSoC Designer 5.0 Service 3 provides support for the following new devices.

New CY8CLED03Dxx and CY8CLED04Dxx Parts

Part Name	Pin Package	Notes
CY8CLED03D01	48-pin QFN	16K Flash

Part Name	Pin Package	Notes
CY8CLED03D02	48-pin QFN	16K Flash
CY8CLED04D01	56-pin QFN	16K Flash
CY8CLED04D02	56-pin QFN	16K Flash

CY8C21x45/CY8C22x45 Product Family

Device	Description
CY8C21345-24SXI	28-SOIC
CY8C22345-24SXI	28-SOIC
CY8C22545-24AXI	44-TQFP

New User Modules

Included in PSoC Designer 5.0 SP3 are several new user modules. The service pack includes user modules for configuring the power handling (PWR) and navigation functions (NAV) of the OvationONS II devices, user modules for accessing specific hardware features of the CY8CLED0xDx product family, multiple capacitive sensing methods (CSD and CSA) customized to work with new device families, and others.

Name	Devices Supported	Description
CSD2X	CY8C2xx45	Dual Channel CSD support
CMPEn	CY8C2xx45	A low power comparator that enables inverse terminal input from I/O
SAR10	CY8C2xx45	10 bit successive approximation ADC
SPIMVL	CY8C2xx45	8 to 16-bit data length SPI
SPISVL	CY8C2xx45	8 to 16 bit data length SPI
PWMDB8L	CY8C2xx45	An enhanced version of the 8-bit deadband PWM. It uses a single digital block to support the deadband.
PWMDB16L	CY8C2xx45	A PWM that uses two digital blocks to support a 16-bit deadband.
SHIFTREG8	CY8C2xx45	An 8-bit stream input shift

Name	Devices Supported	Description
		register
RTC	CY8C2xx45	Support for a real time clock.
PRISM16HW	CY8CLED03/4D0x	16-bit Hardware PrISM (Precise Illumination Signal Modulation)
PWM16HW	CY8CLED03/4D0x	16-bit hardware pulse width modulator (PWM)
DMM16HW	CY8CLED03/4D0x	16-bit hardware density modulated PWM
DualDAC8HW	CY8CLED03/4D0x	Dual 8-bit hardware digital to analog converter (DAC)
CMPHW	CY8CLED03/4D0x	Hardware comparator
CURSENSEHW	CY8CLED03/4D0x	Hardware current sense amplifier
HYSTCTRL	CY8CLED03/4D0x	Hysteretic controller
FLBUCK	CY8CLED03/4D0x	Floating load buck regulator
BOOST	CY8CLED03/4D0x	Boost regulator

Known Limitations for PSoC Designer 5.0 Service 3

CY8C2xx45 Limitations

	Problem	Work Around
1.	The CY8C22345 and CY8C22545 devices have 1K bytes RAM. The psoc.ini file in the Hi-Tech compiler folder only defines 512 bytes.	No work around at this time
2.	The SAR10 functionality fails when the ADC clock is set to SYSCLK/2	Use another clock setting
3.	The PWMDB16L user module datasheet contains incorrect register information.	No work around at this time
4.	Incorrect resources specified in the CSD2x User Module datasheet	No work around at this time

	Problem	Work Around
5.	Using I2C-USB bridge with CSD2X could cause a “TimeOut”	Reset when “TimeOut”
6.	Base part CY8C21345 is missing from CSD and CSD2X datasheet.	The base part is not mentioned in the data sheet, but the user module supports the device.
7.	CY8C21x45 and CY8C22x45 base parts have mismatched Digital Blocks in "Device catalog" and "Device Resource Meter."	The number of digital blocks listed in the device catalog is incorrect.
8.	CSD2X Wizard GUI shown in the data sheet is an older style wizard and does not match the current wizard.	The data sheet is incorrect.
9.	The DTMFDialer User Module is not listed in the user module catalog for any CY8C21x45/CY8C22x45 base parts	The DTMFDialer will be available to the named parts in the full release.
10.	Discrepancies in the SPISVL User Module datasheet: User Module parameter window "InterruptType" parameter has two options: 'TX Reg Empty' and 'SPI Complete'. While in the user module datasheet the parameter is named 'InterruptMode'	The data sheet name is incorrect, but correctly describes the InterruptType parameter.
11.	The CSD2X User Module wizard allows more switches & sliders than the number of available pins.	You must be aware of the number of pins you have available for switches and sliders.
12.	A build error is generated with CY8C22345-24SXI Base Part for “I2C Operation for Boot” and “Full I2C API Support with Bootloader” User Module Error Message: MUL_X(lib/libpsoc.lib: lib/obj/bootldri2c_1flashblock.obj) MUL_Y(lib/libpsoc.lib: lib/obj/bootldri2c_1flashblock.obj) MUL_DH(lib/libpsoc.lib: lib/obj/bootldri2c_1flashblock.obj) MUL_DL(lib/libpsoc.lib: lib/obj/bootldri2c_1flashblock.obj) C:\PROGRA~1\Cypress\Common\CY3E64~1\tools\make: *** [output/TESt.hex] Error 1	No work around at this time

	Problem	Work Around
13.	Build error generated with CY8C22345-24SXI Base Part for “CSD With Clock Prescale” and “CSDwithout Clock Prescale” Error Message: : 0: (499) undefined symbol: (error) DAC_CR(lib/libpsoc.lib: lib/obj/csd.obj)	No work around at this time
14.	Mulishot_cnt variable not generated in a CY8C21x45/CY8C22x45 project with PWMDB16L User Module	No work around at this time
15.	For the below mentioned CY8C21x45/CY8C22x45 User Modules, the “Show PDF” option is disabled. 1.SPIMVL 2.PSISVL 3.SHIFTREG8 4.PWMDB 16L 5.PWMDB 8L 6.CMP En	Enable option manually for “Show PDF”

CY8CLED03/4D0x Limitations

	Problem	Work Around
16.	The context menu features are disabled for Power PSoC parts. Typically right clicking a placed User Module instance in the chip level view provides user with Datasheet, Properties options.	The required menu options may be accessed through the Workspace Explorer.
17.	The Buck and Boost User Module contain incomplete references to an Application Note that contains formulae and description of the Efficiency Calculator.	This Application Note will be available on the Cypress Website.

Known Problems for PSoC Designer 5.0 Service 3

General Problems

Problem	Work Around

	Problem	Work Around
1.	Slow behavior in monitor mode for High Brightness LED drivers. Performance of the communication interface with the target hardware and updates of the driver GUI widgets can be slow when in monitor mode for a project that includes a High Brightness LED driver. It can take from 1 to 2 seconds for keystrokes entered in a text box to be reflected in the GUI. Values entered in the text entry box to be sent to the target hardware can take from 3 to 5 seconds to be transmitted to the target hardware and be read back and displayed in the High Brightness LED driver monitor widget. [26693]	No work around at this time.

Problems with Chip-Level Projects

	Problem	Work Around
2.	When cloning a project you must be aware that some devices do not share user module capabilities. If you clone a project for one device to another device certain user modules may not appear. [27409]	Check what user modules are available for the cloned device.
3.	The I2CHW and EZI2Cs User Modules are unable to detect whether I^2C bus pins are consumed by the CSD User Module. Therefore, it is possible to configure the I2CHW or EZI2Cs User Module to use pins already consumed by the CSD User Module. [22470]	Place and configure the applicable I2CHW, or EZI2Cs User Module prior to placing the CSD User Module. The CSD User Module wizard will detect that the I^2C pins have been consumed.

Problems with System-Level Projects

	Problem	Work Around
4.	When the monitor is running for System-level projects, some of the GUI elements (such as menu items or dialog buttons) may not update properly or may be disabled. [27057]	If this occurs, stop the Monitor before initiating other activities.

	Problem	Work Around

5.	The literal code transfer function does not support multiple variable declarations on a single line of code. [27225]
	Example:
	`int a, b;`
		Declare one variable per line of code.
		Example:
		`int a;`
		`int b;`

6.	PSoC Designer 5.0 allows you to add additional analog input drivers after adding a CSDR CapSense driver. [27385]	When using a CSDR driver you cannot add an additional analog driver even though this is allowed by the program.
7.	A Discrete interface valuator cannot be an input for a TableLookup transfer function if the Discrete interface valuator's state values are not defined in the state list as an unbroken numeric sequence of constants starting at 0. [27342]	When defining state values, define the first state value as 0, the second state value as 1, and so on until the last state is defined. Ensure that each successive value in the list is greater than the previous value by 1.
8.	When a State Machine transfer function valuator's name contains an underscore character (_) and an input driver, output driver, or other valuator's name is identical to the portion of the State Machine valuator's name before the underscore (ex. State Machine valuator named abc_def and LED output driver named abc) , the System Level editor tool might show a transfer function relationship line between these two even though there is no relationship. The project will also fail to build. [27010]	Avoid using the underscore (_) character in driver and valuator names. Also avoid naming drivers or valuators with substrings of other driver and valuator names.
9.	When a System-level project created on a system with high display resolution is open on a system with lower display resolution some of the design elements (drivers, valuator) may appear outside of the viewport. [27356]	Use Zoom Out to bring all design elements into the viewport. Then relocate items as necessary. Or switch to higher display resolution.

Problems with Bootloader Projects

	Problem	Work Around
10.	In HI-Tech, if the last application block (Last_Application_Block) and checksum block (Application_Check_Sum) are not sequentially defined, then the generated bootload file (.txt) will not function correctly. For example if the last application block is 0xFD, and the checksum block is 0xFF, it will not generate a correct bootloadable file (.txt). [40611]	Specify the last application block and the application checksum block sequentially. For example set the Last_Application_Block to 0xFE and the Application_Check_Sum block to 0xFF. A fix for this issue is planned in the next service pack to PSoC Designer.

Problems with Compilation and Debug

	Problem	Work Around
11.	System-level projects do not allow compiler selection before the first "Generate/Build Project" invocation. [26969]	Set the desired compiler to be the default PSoC Designer 5.0 compiler using Tools → Options → Build → Compiler before creating a System-level project.
12.	Multiple instances of PSoC Designer and PSoC Programmer can be open at the same time, but only a single instance is allowed to access the ICE at a given time. An error message stating "ICE Unavailable" will result when instances of either application attempt to access the ICE if it is already connected. [23569]	Press the disconnect button in the application that is currently connected to the ICE or close it completely to free it for connection in one of the other open instances.
13.	PSoC Designer 5.0 may not recognize a second ICE-Cube. [26513]	Go to Project → Setting → Debugger where the new ICE-Cube is listed and select it for use.
14.	If you move a project from one location to another and you try to compile an individual file, the compile will fail because the compiler's path to the file is absolute. If you move the project from one location and then rebuild the entire project, the absolute file paths are regenerated and compilation will succeed. The bug only affects individual file compilation. [26059]	There are build files generated that include explicit paths. These files are rebuilt when the overall project is re-built. You should rebuild the entire project after copying the project before performing individual file compilation.

OvationONS™ II Known Problems

	Problem	Work Around

	Problem	Work Around
15.	I/O registers banks 2 and 3 may not be displayed during debug. [21468]	It may be necessary to add test code to read the relevant registers into a location that is readable for debug. Fix planned.

Considerations Using HI-TECH as the Default C Compiler with System-Level Designs

This section refers to a project that was either done originally in PSoC Express 3.0 or using PSoC Designer 5.0 in the system-level mode (formerly called Express) to generate the design.

- When recompiling an old project with the HI-TECH Lite C compiler, it is unlikely that you will get an error due to any change in compilers. This is because of the standardized and modular programming model of PSoC Express. However, if the base device runs out of available ROM space, a compile error does occur. To remedy this condition:
    - Choose a device with more resources
    - Upgrade to the Pro version of the HI-TECH C compiler
    - Revert to the ImageCraft C compiler
- Compiler settings are saved in individual PSoC projects. When you open an existing project that you built with the ImageCraft C compiler, the project is built with ImageCraft. To change this behavior, you must switch the compiler vendor to HI-TECH (**Build → Select Compiler Vendor** for *project*) for all previously built projects. There is a default compiler selection as well (**Build → Select Default Compiler Vendor**). The default compiler applies to all new projects, not previously built projects.
- The default compiler selection is applied to a project when it is created. After creating a new project, the compiler vendor cannot be changed until the project is built once. Changing the default compiler vendor with a project open does not alter the compiler setting for the open project.
- Because of the amount of code and floating point math required, ColorLock and ColorMix driver operations may exceed the timing of the 64Hz system loop timer, resulting in "Free Run" operation. When compiled with the HI-TECH Lite C Compiler, the behavior of the system may appear to be sluggish when compared to the same code compiled using the HI-TECH Pro C Compiler. The Lite compiler produces less optimized code than Pro, resulting in longer execution time.
- Under **Project → Settings → Build → Compiler** with HI-TECH C compiler selected as the default, there is a button that allows you to switch between HI-TECH Pro mode and HI-TECH Lite mode. The correct control code will be entered by PSoC Designer 5.0 automatically when the button is pressed.

Considerations Using HI-TECH as the Default C Compiler with Chip Level

This section refers to a project that was either created originally in PSoC Designer 4.x or using PSoC Designer 5.0 exclusively in the chip level mode to generate the design. The issues relate to PSoC Designer 5.0 with version 9.61 of the HI-TECH C compiler. For additional help with the HI-TECH compiler, see the docs folder in the HI-TECH installation directory.

- Opening projects created with PSoC Designer version 4.1 or earlier and building them with the HI-TECH compiler may result in the loss of any ImageCraft custom compiler/linker settings.
- The UART_szGetParam() function for the UART User Module currently does not function as expected using the HI-TECH C Compiler. The function is supposed to return a null pointer if there are no more parameters left. Under certain circumstances, when returning a null pointer through a fastcall16 function such as szGetParam(), the pointer is returned to C as a 16-bit pointer with the most significant bit set to 1. See HI-TECH Manual Section 3.3.11.2 Data Pointers. This causes a test against 0 to return as FALSE. The workaround is to include a terminator token in the command string sent to the UART and check for the existence of the token rather than using szGetParam() to terminate the processing loop.
- The HI-TECH C Compiler operates on 7-bit ASCII source code. Extended ASCII characters in source code cause errors. If you need to use extended characters, enter them as \x*FF* where *FF* is the hex value of the extended ASCII character.
- When passing a structure pointer to a fastcall16 function, you may receive a warning if any of the members of the structure are uninitialized, (350) unused member. This is true even if the target function writes values to those members.
- Reusable Local Labels using the HI-TECH C compiler may only contain 1 period. Extraneous periods result in a compile error. See HI-TECH Manual section 4.3.5.5 Symbolic Labels.

Documentation

User guides and key documents are located in the `\Documentation` subdirectory of the PSoC Designer installation directory. The default location is:

C:\Program Files\Cypress\PSoC Designer 5\Documentation

Also included in this documentation folder is a documentation guide which can assist you in understanding all the documentation that is included with PSoC Designer 5.0.

Supporting documents for PSoC Designer's public-domain functionality, using "Find in Files" text search (*grep.pdf*) and the build utility (*make.pdf* and *sed.pdf*), are located in:

…\Program Files\Cypress \PSoC Designer 5\Documentation\Supporting Documents

PSoC Training

We recommend that first time users download and take PSoC Designer *Module 1: Introduction to PSoC* for free by visiting www.cypress.com/psoctraining.

Silicon Errata

The most up-to-date versions of the silicon errata are available on the web site at http://www.cypress.com/psoc and navigating to **Errata Update → PSoC Mixed-Signal Array**.

For assistance go to http://www.cypress.com or contact our Applications Team at 425.787.4814.

Cypress Semiconductor
198 Champion Ct.
San Jose, CA 95134-1709 USA
Tel: 408.943.2600
Fax: 408.943.4730
Application Support Hotline: 425.787.4814
www.cypress.com

© Cypress Semiconductor Corporation, 2008. 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™, and PSoC Express™ 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.

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

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

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