Alpha Release Notes SRN087

PSoC® Designer™ Version 5.1 Beta 1 (Alpha)

Release Date: February 24, 2010

Thank you for your interest in PSoC® Designer™ version 5.1, Beta 1. These release notes list installation requirements and describe software updates and changes. Major sections of this document include:

System Requirements and Recommendations

Installation Notes

New for PSoC Designer 5.1 Beta 1

Known Problems and Limitations for PSoC Designer 5.1

System Requirements and Recommendations

System Requirements	Minimum	Recommended
▪ Processor Speed	1 GHz	2 GHz Dual Core
▪ MB of RAM	1 GB	2+ 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 higher), Vista, or Windows 7	✓	✓
▪ Microsoft Internet Explorer (not IE8 beta)	7	7+
▪ .NET Framework	2.0	2.0 (SP1 or higher)
▪ Adobe Reader (for viewing PDF Documentation)	6	9
▪ PSoC Programmer	3.11	3.11

Updates

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

Support for ImageCraft Pro Compiler

PD5.1 introduces native support for the ImageCraft Pro compiler, available for purchase at http://imagecraft.com (click on Cypress PSoC Compiler Tools).

Installation Notes

Note: PSoC Designer 5.1 is not compatible with the Internet browser, Microsoft Internet Explorer 8 *beta* (any beta version). If you are running IE8 beta, please upgrade to the released version or downgrade to a previous version.

PSoC Designer 5.1 will co-exist alongside PSoC Designer 5.0 SP6. You do not need to remove SP6 to install PSoC Designer 5.1.

PSoC Designer 5.1 uses a new Installation process. See the next section for more details.

Installation

The Installation process has changed radically from PSoC Designer 5.0. This was done to accommodate a Live Update capability and decrease download size. The installation process is a set of wizards that walk you through installing various components. You can install PSoC Designer and various prerequisites from the web. For those that have issues with installing behind a firewall or on a machine with no internet connection, an ISO file is provided on the web for creation of your own CD. There are slight differences in the process, based on the medium used to install the software.

In order to install PSoC Designer, you need to first install PSoC Programmer. The CDs provide the necessary prerequisites and the wizards to guide you through installing the appropriate software. The Web installation requires you to download and install the executables separately. The following sections contain more specific installation details.

Note Do NOT plug in your Miniprog3 until all software installation is complete AND the PSoC Designer application has been opened.

Web Installation

If you are downloading the software from the web, you should run the PSoC Programmer executable first, because it is a prerequisite for PSoC Designer. Then run the PSoC Designer executable.

1. Double-click the PSoC Programmer executable file to launch PSoC Programmer InstallShield Wizard.
2. Follow the prompts to install PSoC Programmer. The CyInstaller for PSoC Programmer opens and displays a series of steps to install PSoC Programmer and various drivers. When complete, close the installer.
3. Double-click the PSoC Designer executable file to launch the installer.
4. Follow the prompts to install PSoC Designer. The CyInstaller for PSoC Designer opens and displays a series of steps to install PSoC Designer, and it will perform pre-requisite checks.
5. Install the prerequisites as needed.
6. When complete, close the installer.

PSoC Designer CD Installation

The PSoC Designer CD contains PSoC Designer and PSoC Programmer, as well as various prerequisites.

1. Load the CD. The main installer program should run automatically. If not, double-click the cyautorun.exe file to launch it.
2. On the main installer, click the **Install Software for PSoC…** button to launch the PSoC Designer InstallShield Wizard.
3. Follow the prompts on the wizard. The first step will prompt to install PSoC Programmer.

   The CyInstaller for PSoC Programmer opens and displays a series of steps to install PSoC Programmer and various drivers.
4. Continue to follow the prompts until PSoC Programmer and the drivers are installed, then resume with the main installer program.

   The CyInstaller for PSoC Designer opens and displays a series of steps to install PSoC Designer. One of the steps displays the Required Software page.
5. Click the “Download” link for any software that is not installed as indicated (such as Keil, PDF, etc.). Run the installer for that program as needed.
6. Continue following the prompts to install PSoC Designer.
7. Once complete, close all the windows including the main installer program.

Note to HI-TECH Compiler Users

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

Software Update Instructions

As part of the installation process, the Cypress Update Manager utility will also be installed and located on the **Start** menu. You can use this utility to update all programs you have installed when updates for them become available.

Follow the instructions provided with the tool, as needed.

New for PSoC Designer 5.1 Beta 1

Included in Beta 1 are many bug fixes, new features, and support for several new parts. New products and new user modules are discussed in following sections. Please note that in many cases new product support may be in the beta release phase. It will be explicitly stated in cases where support is at the beta level.

New Device Support

CY7C63833

New enCoRe III parts

CY7C64215-56LTX**C**

CY7C64215-56LTX**I**

New enCoRe V parts

New Automotive and Automotive Extended parts

CY8C22345-24PVXA

CY8C21345-24PVXA

CY8C22345-12PVXE

CY8C21345-12PVXE

New parts in the CY8C20xx6 family

CY8C20446A-24LQXI

CY8C20746A-24FDXI (30 Ball WLSP)

CY8C20766A-24FDXI (30 Ball WLSP)

Notable New Functionality

Segment LCD Driver UM

Haptics UM

Support for Windows 7

Significant testing was done running PSoC Designer on Windows 7, and support has existed at a beta level since November 2009.

User Feedback Mechanism

Have a quick comment about PSoC Designer? Just go to the Help menu and send us a comment.

Problem Using External Crystal Oscillator Fixed

Problem: In previous releases, using the External Crystal Oscillator (ECO) of CY8C20xx6, CY7C643xx, and CY7C604xx devices was not directly supported by PSoC Designer.

Solution: This problem has been fixed with this release of PSoC Designer. The option to use the ECO can be enabled in the Global Resources table for these devices. However, a new boot.tpl file is required in order for this solution to take effect. Therefore, you may be prompted to update your boot.tpl file when opening projects that already exist for these devices.

Known Problems and Limitations for PSoC Designer 5.1 Beta 1

General Problems

	Problem	Work Around
1.	The I^2C user modules have operating frequency options of 50 kHz, 100 kHz, and 400 kHz. However, these options assume that SysClk is nominally 24 MHz. If SysClk is less than nominal 24 MHz (when using SLIMO mode, for example), these I^2C operating frequencies will vary with the SysClk frequency.	When SysClk is lower than 24 MHz (nominal), choose a higher frequency option for the user module than is actually desired on the I^2C bus.
2.	Blank Part Catalog Blank Interconnect View The Part Catalog or Interconnect view shows nothing. HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureControl\ FEATURE_LOCALMACHINE_LOCK	This issue may be a result of your local IT policy. It can be fixed by adding "psocdesigner.exe" and "cmx.exe" to the 'Allow List' in the following registry key:
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.	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.
4.	Cannot Open Moved/Renamed PD Projects. Error message "Invalid Project Structure. Name of the folder that contains project's '.cmx' file must match '.cmx' file name".	This may occur when the workspace folder is renamed or copied with a different name. Workspace copies should be made with the "Save Workspace As" feature available from the File menu, i.e. File -> Save Workspace As
5.	Cloned projects that contains bootloader user module does not generate the correct boot.tpl. [60161]	1. Right click on the bootloader and select “Boot Loader Tools”. 2. Select Get Files. 3. Rebuild.

Problems with Compilation and Debug (including ICE-Cube and MiniProg3)

	Problem	Work Around
6.	For projects that use the Hi-Tech compiler, the values of pointer variables shown in the Watch Window sometimes differ from what is reported in the memory window.	The value of the pointer shown in the Watch Window is the true value of the pointer variable. Use the value displayed in the Watch Window for debugging purposes.
7.	For projects that use the Hi-Tech compiler, you may not be able to set breakpoints on some source code lines.	The Hi-Tech compiler optimizes-out (removes from compiler output) statements that it deems of no consequence to the execution of the program. Set the breakpoint on a subsequent source code line.
8.	Projects that use the Hi-Tech compiler and include one assembly source file (*.asm) within another (using the 'include' statement) will confuse the Debugger.	Only use the assembly language 'include' instruction to embed assembly language include (*.inc) files.
9.	A field of a structure cannot be watched independently from the whole structure. That is, you cannot add just one field of a struct to the Watch Window.	If you want to watch the fields of a struct, add the struct instance itself to the Watch Window. Then, expand the node of the variable in the Watch Window to see the struct's fields.

10.	The latest ImageCraft compiler is reporting an error "operands of = have illegal types `pointer to const char' and `pointer to char'". This was not seen previously.	The new compiler is stricter than the previous compiler. The previous compiler would allow a statement such as:
		`const char *cptr;`
		`cptr = (char *) 0x3000;`
		The new compiler will generate an error and you must change this code to:
		`const char *cptr;`
		`cptr = (const char *) 0x3000;`
		Although the original code is legal C, the ImageCraft compiler generates an error because it uses the const specifier to indicate that the data will be stored in flash. This special meaning of const means that pointers to a const location are different from a normal pointer and therefore a standard pointer cannot be assigned to a const pointer.

	Problem	Work Around
11.	On smaller PSoC devices, the ImageCraft compiler may occasionally fail to build projects with messages like: `!E (1918): Cannot allocate space for paged area 'xxxxxxxxxxxx'` This is most often seen on devices with small amounts of RAM or in applications that use a lot of RAM. By default, the stack is allocated to the last page of RAM with an offset of 0x00, so if the code requires slightly more than is available on the other pages, the build will fail.	On smaller PSoC devices, you may want to set the stack page offset to a value other than '0' (the default). This gives you more code space, but a smaller stack. In PSoC Designer, select **Project → Settings → Compiler** and enter a value for the **Stack page offset**. For example, setting the offset to 0xC0 will allow adequate stack space and room for local variables as well.

Device-Specific Problems

	Problem	Work Around
12.	When creating designs with CY8C20xx6 devices using the CMP user module, the value displayed by the LPFSource variable in the Properties Window does not match the value displayed by the LPF variable.	The value in the user interface does not update properly. The actual value contained in the register is the proper value. The defect that causes the value not to update will be fixed in PSoC Designer 5.1.
13.	OvationONS™ II I/O registers banks 2 and 3 may not be displayed during debug.	It may be necessary to add test code to read the relevant registers into a location that is readable for debug.
14.	The analog column numbers of some CY8C28xxx devices in PSoC Designer do not match the column numbers given in the Technical Reference Manual for CY8C28xxx devices.	In these cases, the column numbers in PSoC Designer are incorrect. This should be accounted for when correlating data in the Technical Reference Manual (TRM) with what is seen in the Chip View of PSoC Designer.
15.	Debugger problems with CY8C63813 – does not work properly when MSTIMER or PITIMER12 user modules are set in certain conditions.	For the MSTIMER or PITIMER12 user modules to work properly, "CPU Clock / N" needs to be set to "SysClk/32" or faster.
16.	If you use an external crystal with a CY8C20xx6 or CY8CTxx20x device, you may get an error from the compiler: Code address xx already contains a value	Go to **Project → Settings → Build → Linker** and set the relocatable code start address a few bytes higher (for example, from 0x150 to 0x170). The external crystal adds some code to the project and this addition is not properly compensated. This issue will be fixed in a future release.

	Problem	Work Around

17.	If the decimator is used in the base configuration and a dynamic configuration overlay is loaded, the Decimator Control Register 2 (DEC_CR2, address 1,E7h) will be incorrectly modified. This will affect the operation of any user module in the base configuration that requires the decimator (e.g. ADC).	In PsoCConfig.asm, in the LoadConfig function of the overlay, comment out the line that modifies the DEC_CR2 register as follows:
		`; writing Type2Decimator_Control register`
		`; M8C_SetBank1`
		`; and reg[e7h], ~c8h`
		`M8C_SetBank1`
		Note: The *psocconfig.asm* code will be overwritten when the application is regenerated, so this code will need to get commented out again after regenerating

18.	The CURSENSEHW user module lists available gain values of 3, 4, 15, and 20. However, only the gain value of 20 is guaranteed by the device datasheet.	Only use the gain value of 20.
19.	For CY8CLED0xG01 devices, when setting the HYSTCTL user module “Gate Driver” property to External, the wire connection between the HYSTCTL and GD pin in the chip view is not shown. However, the register is properly configured and the gate driver will be enabled.	Ignore the missing wire connection in the chip view.
20.	PSoC Designer projects using the following PowerPSoC devices cannot connect to the CY8CLED04DOCD1 on-chip debugger: CY8CLED01D01 CY8CLED02D01 CY8CLED03D01 CY8CLED03D02	Create a project using the CY8CLED04D01 device (using only the resources that are present in the actual part you are interested in) and debug your application. When ready to program, clone the project to the desired device, then build and program.
21.	The PLT user module does not use shadow registers when it writes to pins P2[7] and P1[3]. Therefore, while PLT user module functions are executing, the port data register of any GPIOs that are configured as inputs on port 1 and 2 (PRT1DR and PRT2DR, respectively) might be modified if the input voltage to the pin is modified during this time.	If planning to read a GPIO pin on Port 1 or Port 2, shadow registers should be used. For more information on shadow registers, refer to the SHADOWREGS (in the Misc Digital folder) user module data sheet. When using shadow registers with the PLT user module, refer to the workaround above so that the PLT pins do not get accidentally modified.

	Problem	Work Around

22.	If the PLT user module is placed in a dynamic reconfiguration overlay, it will not fully restore the previous register settings when the PLT user module is stopped and the overlay is unloaded.	In PLT_1.asm, replace the PLT_Stop API code with the following:
		`_PLT_Stop:`
		`PLT_Stop:`
		`  RAM_PROLOGUE( RAM_USE_CLASS_4 )`
		`  lcall _PLT_IsTransmitterLoaded`
		`  cmp A, 0`
		`  jz .PLT_StopRX`
		`  lcall _FSK_StopTransmitter`
		`  lcall  _PLT_UnloadConfig_Transmitter`
		`  jmp .PLT_StopEnd`
		`.PLT_StopRX:`
		`  lcall _FSK_StopReceiver`
		`  lcall _PLT_UnloadConfig_Receiver`
		`.PLT_StopEnd:`
		`  M8C_SetBank0`
		`  and reg[PRT0GS], ~01h`
		`  and reg[AMX_IN], ~3fh`
		`  or reg[AMX_IN], 3bh`
		`  and reg[RDI1RI], ~ffh`
		`  or reg[RDI1RI], 55h`
		`  and reg[INT_MSK0], ~0x40`
		`  RAM_EPILOGUE( RAM_USE_CLASS_4 )`
		`  ret`
		Note: The PLT_1.asm code will be overwritten when the application is regenerated, so it is recommended to back up the modified code and restore it after regenerating, or create a new function (e.g. PLT_Stop2) in a separate file.

23.	In the PLT Modem Only user module option, the PLT_RXEnableInt API does not enable one of the receiver interrupts. Similarly, the PLT_RXDisableInt API does not disable one of the receiver interrupts.
	However, when PLT_Start is called, all receiver interrupts are correctly enabled. This problem only applies if planning to disable and re-enable the receiver interrupts.
		In PLT_1.asm, add the following line to PLT_RXEnableInt:
		`lcall  _PLT_Func_22_EnableInt`
		Also, add the following line to PLT_RXDisableInt:
		`lcall  _PLT_Func_22_DisableInt`
		Note: The PLT_1.asm code will be overwritten when the application is regenerated, so it is recommended to back up the modified code and restore it after regenerating.

	Problem	Work Around
24.	If generating the application with the PLT user module and the PLL Mode is set to “Disable”, a DRC warning will be incorrectly generated. The PLL Mode is allowed to be set to “Disable” as long as the external 24MHz oscillator will be used on P1[4]. Using the IMO is not recommended for PLT communication. However, the 32.768kHz external crystal must always be selected (32K_Select = “External”)	Ignore the DRC warning if not using the IMO.
25.	Other CDTs with the Release Notes tag, but whose workaround was not ready at Press Time.	44438, 45559, 47476, 48504, 52269, 60663, 60868, 61614, 61693, 62026, 62325, 64310, 64673, 65779, 66437, 66698, 67628, 67852, 68042, 69024, 69138, 69345
26.		

Problems with Compilation and Debug (including ICE-Cube and MiniProg3)

	Problem	Work Around
27.	For projects that use the Hi-Tech compiler, the values of pointer variables shown in the Watch Window sometimes differ from what is reported in the memory window.	The value of the pointer shown in the Watch Window is the true value of the pointer variable. Use the value displayed in the Watch Window for debugging purposes.
28.	For projects that use the Hi-Tech compiler, you may not be able to set breakpoints on some source code lines.	The Hi-Tech compiler optimizes-out (removes from compiler output) statements that it deems of no consequence to the execution of the program. Set the breakpoint on a subsequent source code line.
29.	Projects that use the Hi-Tech compiler and include one assembly source file (*.asm) within another (using the 'include' statement) will confuse the Debugger.	Only use the assembly language 'include' instruction to embed assembly language include (*.inc) files.
30.	A field of a structure cannot be watched independently from the whole structure. That is, you cannot add just one field of a struct to the Watch Window.	If you want to watch the fields of a struct, add the struct instance itself to the Watch Window. Then, expand the node of the variable in the Watch Window to see the struct's fields.

	Problem	Work Around

31.	The latest ImageCraft compiler is reporting an error "operands of = have illegal types `pointer to const char' and `pointer to char'". This was not seen previously.	The new compiler is stricter than the previous compiler. The previous compiler would allow a statement such as:
		`const char *cptr;`
		`cptr = (char *) 0x3000;`
		The new compiler will generate an error and you must change this code to:
		`const char *cptr;`
		`cptr = (const char *) 0x3000;`
		Although the original code is legal C, the ImageCraft compiler generates an error because it uses the const specifier to indicate that the data will be stored in flash. This special meaning of const means that pointers to a const location are different from a normal pointer and therefore a standard pointer cannot be assigned to a const pointer.

32.	On smaller PSoC devices, the ImageCraft compiler may occasionally fail to build projects with messages like: `!E (1918): Cannot allocate space for paged area 'xxxxxxxxxxxxx'` This is most often seen on devices with small amounts of RAM or in applications that use a lot of RAM. By default, the stack is allocated to the last page of RAM with an offset of 0x00, so if the code requires slightly more than is available on the other pages, the build will fail.	On smaller PSoC devices, you may want to set the stack page offset to a value other than '0' (the default). This gives you more code space, but a smaller stack. In PSoC Designer, select **Project → Settings → Compiler** and enter a value for the **Stack page offset**. For example, setting the offset to 0xC0 will allow adequate stack space and room for local variables as well.

Other Issues

	Problem	Work Around
33.	Other CDTs with the Release Notes tag, but whose workaround was not ready at Press Time.	44438, 45559, 47476, 48504, 52269, 60663, 60868, 61614, 61693, 62026, 62325, 64310, 64673, 65779, 66437, 66698, 67628, 67852, 68042, 69024, 69138, 69345

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

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**.

For assistance go to http://www.cypress.com/support 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, 2010. 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™, CapSense Express™, and PSoC Express™ are trademarks and PSoC® and CapSense® are registered trademarks 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.