Release Notes SRN081

PSoC® Designer™ Version 5.0 Service Pack 6

Release Date: October 30, 2009

Thank you for your interest in PSoC® Designer™ version 5.0, Service Pack (SP6). These release notes list installation requirements and describe software updates and changes. Major sections of this document include:

- System Requirements and Recommendations
- De-emphasis of System-Level Design (Express)
- Installation Notes
- New for PSoC Designer 5.0 Service Pack 6
- Known Problems and Limitations for PSoC Designer 5.0 Service Pack

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), or Vista	✓	✓
▪ Microsoft Internet Explorer (not IE8 beta)	6.0 (SP1)	7
▪ .NET Framework	2.0	2.0 (SP1 or higher)
▪ Adobe Reader (for viewing PDF Documentation)	6	9
▪ PSoC Programmer	3.10 (build 519)	3.10 (build 519+)

Updates

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

Compiler Improvements

Service Pack 6 (SP6) introduces several enhancements and bug fixes to ImageCraft, the free C compiler that ships with PSoC Designer. There are no code size limitations or time expirations with this production-quality compiler. This is a free compiler which is improved over the version that was previously for sale.

De-emphasis of System Level Design (Express)

Beginning with this service pack, we are de-emphasizing System-Level Design (PSoC Express) . This will not affect CapSense® Express™ users. With the next release of PSoC Designer, System Level Design will be completely removed. While no functionality has been removed from this release, we recommend using Chip-Level Design (PSoC Designer style). We will continue to make service pack 6 available for System-Level Design users, but we are not recommending it for production design (except for CapSense Express). There will be continued support for CapSense Express in a service pack of PSoC Designer 5.1. When further information is available, it will be posted at http://www.cypress.com/psocdesigner.

Installation Notes

Note: PSoC Designer 5.0 SP6 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.

To Install:

Shut down any currently running instances of PSoC Designer.

If PSoC Designer 5.0 is currently installed, uninstall it. Click Start, click Control Panel, and then double click Add or Remove Programs.

1. Download and install the latest PSoC Programmer 3.10 by running the provided installer.
2. Download and install PSoC Designer 5.0 SP6 by running the provided installer.

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

New for PSoC Designer 5.0 Service Pack 6

Included in Service Pack 6 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

Expanded Support for the CY8C20xx6 Family

Additional analog capability has been added for the 8K flash device in this family. Specifically, support for the ADC User Module, Comparator User Module, and a precision 32 kHz oscillator have been added. The newest CY8C20xx6 8K family adds 1.8V operation and best in-class current consumption with up to 36 CapSense® buttons

Expanded Support for the CYONS Family

In this service pack release, several new devices of the CYONS family are added and several unused devices are removed. There are also big improvements in the user modules. The NAV and PWR User Modules are moved to the Legacy UM folder while a new user module, LaserNav, replaces and improves the functionality of both. Older designs using NAV and PWR will continue to work.

In addition, Cypress has added the capability to calibrate the output voltage of the Boost Regulator for the OvationONS II family. Customers who use the calibration feature (by re-compiling the source code) will improve the nominal boost accuracy for CYONS2001, CYONS2101, and CYONS2110 devices.

New Package available for the CY8C21223

The CY8C21223 is now available in a 16-pin QFN (COL) package. The part number is CY8C21223-24LGXI.

Notable New Functionality

New Filters

4 new filter user modules (UMs) have been added to PSoC Designer Service Pack 6. The names of these modules are:

- BPF4 – Four pole switched capacitor band pass filter
- LPF4 - Four pole switched capacitor low pass filter user module
- ELPF2 - Switched capacitor two pole elliptical low pass filter user module
- ELPF4 - Switched capacitor four pole elliptical low pass filter user module

These user modules enable quick and easy 2/4 pole filter designs based on the switched capacitor blocks. In Service Pack 6, these new filter UMs are only available for the CY8C28xxx family of parts, but will be ported to new families in the future.

SmartSense™ CapSense® User Module (CSDAUTO) Moves to Production.

The CSDAUTO self-tuning user module for CapSense designs, first released in Service Pack 5, is now in production.

- True plug-and-play performance
- No manual tuning of multiple parameters
- Enables customers to get to market faster than ever!

The API is compatible with the previous version of the CSD (CapSense Sigma-Delta) User Module, so you can upgrade your existing CSD designs without modifying your user code. This user module brings the robust operation of CSD with the ease of use of automatic in-system determination of all CapSense parameters. Consult the user module data sheet for details and a Quick Start.

New CapSense Successive Approximation User Module (CSAMFS) (beta)

In Service Pack 6, a new user module, CSAMFS (CSA-Multi-Frequency Scanning), is available. This new user module implements high-level software routines to compensate for environmental and physical sensor variations including EMC. Consult the user module data sheet for details and a Quick Start.

Expanded Part Support and Functionality for the CyFi™ Star Network Protocol Stack (CYFISNP) UM

The CyFi Star Network Protocol Stack (CYFISNP) User Module is designed to address up to 250 general-purpose nodes; it provides reliable two-way radio frequency communication between the hub and node(s). This User Module is now supported in several new part families. There are also performance enhancements as well as pin selection and routing improvements.

Start Page Updates

The Start page has links to web locations where you may find Example Projects. As mentioned earlier, SP6 is the last release of PSoC Designer containing PSoC Express (aka System Level Design) functionality. This de-emphasis is reflected in the start page.

Project Cloning Improvements

Several enhancements and bug fixes have been made to project cloning in Service Pack 6. Users will get additional help when changing target devices. For instance, not all devices support all user modules. You will receive better information and guidance about these issues when you clone projects now.

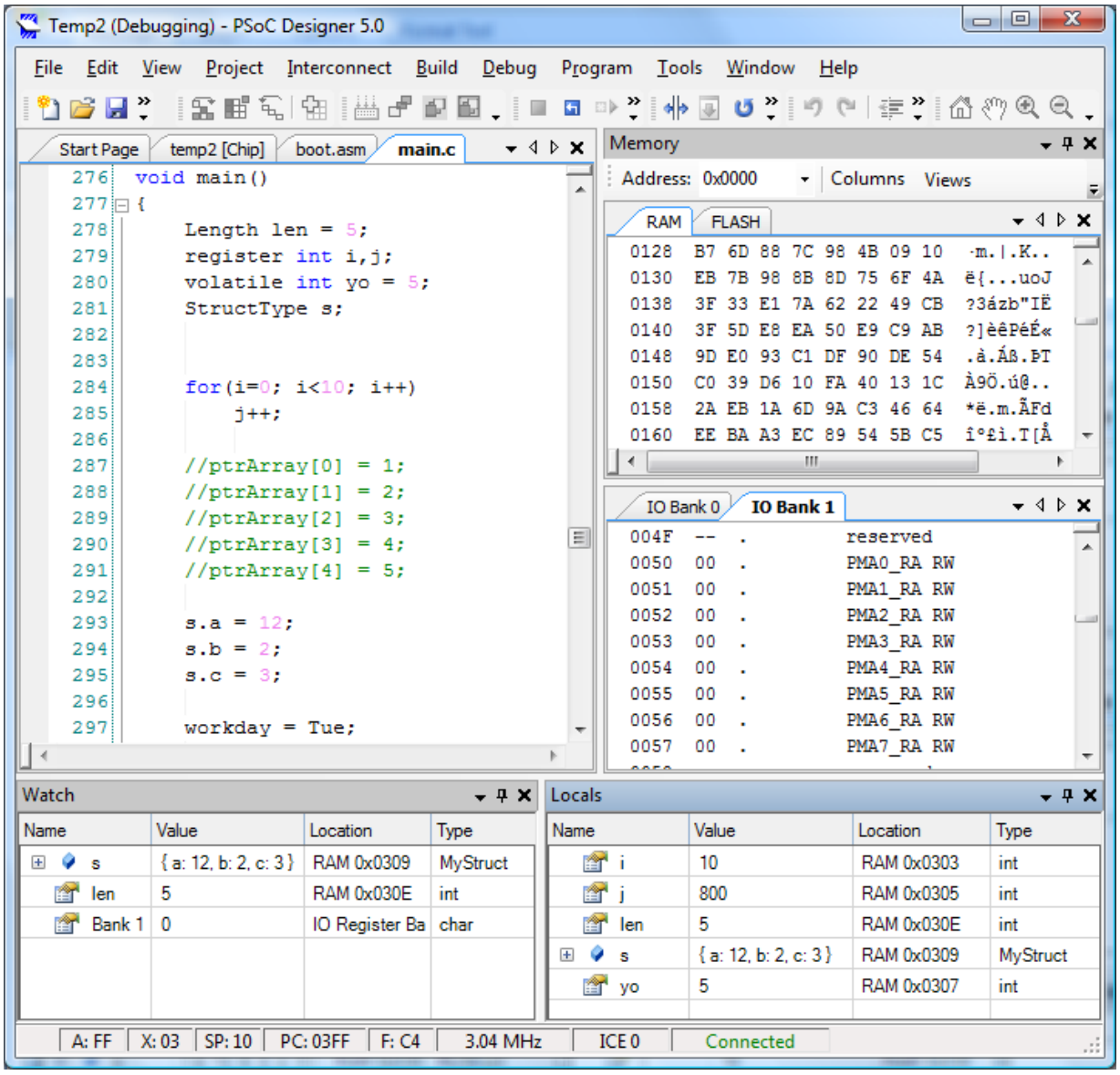
Project cloning is most often used to change PSoC Devices in the middle of a design, or to port older designs to newer devices. It saves a great deal of time over recreating the design with the new device, as many of the settings and connections can be preserved. See the on-line help for more info.

Tabbed Memory Window in Debugger

A more powerful user interface has been created for the Memory Window of the Debugger, with each memory bank displayed in a separate tab. You can configure the Memory Window to display multiple views of the same memory bank, re-name the views, and split the Memory Window horizontally or vertically. These features allow you to have several different views of memory visible at the same time.

Locals Watch Window

A second Watch Window has been added that displays all local variables. Whenever execution halts, the content of the Locals watch window is automatically updated to contain the current scope's local symbols. The screen-shot below shows the new Locals window and an example configuration of the new Tabbed Memory Window.



Legacy User Modules

New to PSoC Designer 5.0 Service Pack 6 is a user module folder named "Legacy." User modules in this folder are not recommended for new designs, but are still available to your existing designs that may use them. User modules are placed in the legacy folder when newer user modules are created that offer superior performance. The Legacy folder was added to help you make better decisions when selecting between seemingly equivalent user modules. For example, the NAV and PWR User Modules have been replaced by the better performing LaserNav user module, so NAV and PWR were moved to the Legacy folder to avoid confusion.

Beta Support for Windows 7

Significant testing was done running PSoC Designer on Windows 7. You may encounter small issues with installation, but the bulk of PSoC Designer functionality works fine on Windows 7.

Some users have reported occasional failures (error messages) when creating new projects under 32-bit Windows 7. The problems were transient. If you receive a message saying, “project name already exists” when it does not, close the window and try again.

Known Problems and Limitations for PSoC Designer 5.0 Service Pack 6

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.	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: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_LOCALMACHINE_LOCK
3.	In the Code Editor, if you choose a proportional font, especially if you have a mix of fixed-width and proportional fonts chosen for various code items, you may notice that the text moves as you select or enter text.	By definition, the width of letters varies in proportional fonts. Changes in spacing are unavoidable with proportional fonts. For best results, use only fixed-width fonts in the code editor. Fixed-width fonts are highlighted in bold in the Font list. Only Fixed Width fonts will be available in PD5.1.

	Problem	Work Around
4.	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.
5.	When System-level project is created using some of the Designs from Design Catalog, the driver icons are hidden or partially visible in the Express Editor.	Use Zoom Out/Pan toolbar buttons to bring all driver icons into the viewport.
6.	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
7.	The default configuration of the SW only bootloader alters the checksum block after the first reset after a device program or device bootload. For this reason the options described in the datasheet "Checksum on Checksum block" is incompatible with the SW only bootloader (BootLdrI2C).	Do not enable the Checksum on Checksum block feature if using the SW only Bootloader (BootLdrI2C).
8.	Cloned projects that contains bootloader user module does not generate the correct boot.tpl.	1. Right click on the bootloader and select “Boot Loader Tools”. 2. Select Get Files. 3. Rebuild.

9.	When selecting SysClk source as an external 24MHz clock (P1[4]), the device does not start up.	Set the SysClk source to internal. At the start of main.c, insert the following code:
		`int i;`
		`for (i=0; i<254; i++)`
		`{`
		`M8C_ClearWDT;`
		`}`
		`OSC_CR2 |= (OSC_CR2_EXTCLKEN | OSC_CR2_IMODIS | OSC_CR2_SYSCLKX2DIS);`
		See http://www.cypress.com/?id=4&rID=37908 for more information.

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

	Problem	Work Around
10.	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.
11.	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.
12.	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.
13.	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.

14.	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
15.	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.
16.	In designs using the LCD User Module, the following build error is received – "Make Const Behave as Standard C"	The LCD user module contains APIs that are incompatible with treating CONST as RAM. This will be fixed in PSoC Designer 5.1.
17.	Occasional problems creating new projects in 32-bit Windows 7. Sometimes when creating a project, you may receive a message that the project already exists.	Simply close the window and try again. It should be successful the second time. This problem has been reported only on systems with a fresh Windows 7 installation and only occasionally in these instances. This problem has not been reported on 64-bit systems.

Device-Specific Problems

	Problem	Work Around
18.	EzI2C user module should be changed to prevent SCL getting stuck to low. This applies to: CY8C20xx6, CY7C643/4/5xx, CY7C604xx, CY7C6053x, CYONSxxxx, CYONSFNxxxx, CY8CTST2xx, CY8CTMG2xx, CY8CTMA3xx	Writing to the SCR register during the interval between start and address byte complete will indirectly cause the SCL to get stuck. The following code should be removed in the EzI2C ISR to avoid this problem, `reg[EzI2Cs_SCR_REG],~EzI2Cs_SCR_STOP_STATUS ; clear stop bit`
19.	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.

	Problem	Work Around

20.	Problems with SPIS and SPIM CLK register setting with CY7C64315 and CY7C64316. P1_0 is the SPIS/SPIM SCLK pin for the CY7C64315/6. However, this pin is not selected for the clock by default. To activate the SCLK on P1_0, a bit in the IO_CFG register needs to be set to select P1_0 as the SPI CLK.	Set bit 2 of IO_CFG before starting SPIS/SPIM in these two devices. For example:
		`IO_CFG |= 0x04;`
		`SPIS_1_Start(SPIS_1_SPI_MODE_0 |`
		`SPIS_1_SPI_MSB_FIRST);`

21.	Cannot Place CSD UM after cloning to a CY8C24x94 part.	Exit and restart PSoC Designer.
22.	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.
23.	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.
24.	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.
25.	CY8C201xx: In a System Level project, after placing a 1 button or two button driver, you should not be able to place other drivers. However, some drivers are still allowed to be placed.	When using a 1 button or 2 button driver with a CY8C201xx project, don't use any other drivers in that project.
26.	In the interconnect view of CY8C28xxx, CY8C22x45, CY8C21x45, CY8CTxx300, and CY8CTxx301 devices, it is possible to drive global broadcasts with each other at the same time. For example, you can drive GIE[0] with GOE[0] (and vice versa) at the same time. Currently there is no DRC in the software to check for this design error.	This issue will be addressed in an upcoming release. Until then, please be aware of this problem.

	Problem	Work Around
27.	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.

28.	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

29.	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.
30.	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.
31.	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.

	Problem	Work Around

32.	When using shadow registers (e.g. LED user module) with the PLT user module, the PLT_1XTAL_STABILITY (P1[3]) and PLT_1TXSHUTDOWN (P2[7]) port data registers (PRT1DR and PRT2DR, respectively) will get accidentally modified.	When updating Port 1, make sure to update the shadow register with the present value of P1[3], as in the following code:
		`mov A, reg[PRT1DR]`
		`and A, 0x08`
		`or A,[Port_1_Data_SHADE]`
		`mov reg[PRT1DR],A`
		
		When updating Port 2, make sure to update the shadow register with the present value of P2[7], as in the following code:
		`mov A, reg[PRT2DR]`
		`and A, 0x80`
		`or A,[Port_2_Data_SHADE]`
		`mov reg[PRT2DR],A`
		
		Note: If modifying any of the code in the library files (e.g. LED.asm), the code will be overwritten when the application is regenerated, so it is recommended to back up the modified code and restore it after regenerating.

33.	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

34.	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.

35.	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
36.	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.

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, 2009. 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.