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.

Peripheral Driver Library (PDL) Release Notes

Version 3.1.3

Release Date: September 24, 2020

Thank you for your interest in Peripheral Driver Library (PDL) version 3.1.3. This document lists the content of the release package.

Overview

PDL v3.x simplifies software development of the PSoC® 6 family of devices using PSoC® Creator™. This release provides the following features:

- PSoC 6 digital and analog peripheral drivers, which enable rapid peripheral software development in PSoC Creator and third-party IDEs
- The ARM Cortex® Microcontroller Software Interface Standard (CMSIS) core access header files directly from the CMSIS 5.3.0 release
- CMSIS-compliant device header files, startup code (platform initialization), and device configuration header files
- Fully configurable Device Firmware Update (DFU, previously Bootloader) SDK ready to be integrated into any application
- FreeRTOS source code integrated with the PDL
- Bluetooth Low Energy (BLE) middleware and stack library
- Secure Image reference design, which demonstrates Trusted Execution Environment (TEE) execution on the CM0+ core
- PDL Application Programming Interface (API) Reference
- PDL User Guide

Note PDL v3.X does not support FM microcontroller devices – use PDL v2.1 to develop firmware for FM0+ and FM4 MCU devices. Use PDL v2.0 for FM3 MCU devices. All supported versions of the PDL are available through the PDL product page.

- **Note** While PSoC Creator remains fully supported and shall be updated periodically, we encourage all users to move to ModusToolbox® for all future designs of PSoC 6 and future families of Cypress programmable devices. ModusToolbox supports the new CY8C6xx8, CY8C6xx9, and CY8C6xxA series, has new drivers for these series (such as SD Host and DMAC), and has new middleware for all the families such as CapSense® CSD ADC, and BLE. For more details, visit the ModusToolbox landing page at http://www.cypress.com/products/modustoolbox-software-environment.

New Features

This release includes the following new and updated drivers and middleware. Refer to the driver documentation sections of the PDL API Reference for more details.

Updated Middleware and Software

- BLE 2.60. Bug fixing. For more detail, refer to section “Changelog of PSoC 6 BLE Middleware API Reference”.
- SYSCLK 2.0. Updated the ECO trimming values calculation algorithm in the Cy_SysClk_EcoConfigure implementation.

Limitations and Known Issues

Limitation	Description
USB Configurator	As in previous releases, supported only on the Windows x64 platform. Perform the development on the x64 platform. It also requires the Microsoft Visual C++ Redistributable Packages for Visual Studio 2017. The installer provides the option to install that package.
RAM and Flash memory consumption increase	As in previous releases, this version uses a register access layer and register access macros instead of direct register access. This makes the register access device independent. To support migration between devices without the need to recompile the PDL, the driver logic that handles the differences in silicon IP blocks is performed at runtime instead of compile time.
Reserved system resources	As in previous releases, the PDL reserves certain system resources for internal use. These resources include IPC resources, like the first 16 IPC semaphores and interrupt lines to the Cortex M0+ CPU. Using any of these resources in your design will lead to unexpected behavior. Refer to the PDL API Reference for details.
SVD file schema	The PDL provides SVD files that use schema version 1.3. Some IDEs use a previous schema version. As a result, the IDE debugger does not display device registers. This issue disappears as third parties update their IDEs.
SYSPM 4.0 is not backwards compatible	Refer to the SYSPM driver documentation within the PDL API Reference for the design impact, backward compatibility aspects, and the migration guide.
Cache coherency	PSoC 6 does not support cache coherency. As a result, when a row of flash that executes instructions is written/updated, the updated information is not reflected in the cache. The cache should be invalidated in the firmware during such instances. This is applicable for both CM4 and CM0+ cache. In other words, the appropriate (CM0+ and/or CM4) cache should be invalidated.
Third-party IDE integration	The project templates for IAR and MDK are still provided and reflect the changes done in the PDL organization. The project templates for GCC and iSYSTEM are discontinued. For GCC, use ModusToolbox software, which has complete support of the MAKE command line flow for various compilers
IPC and Crypto drivers no longer use configuration files.	Older code examples and customer projects may have them. Move all needed configuration code from configuration files to the application source / header files.

Known Issue	Workaround

The function malloc() does not return an error when the allocation size is bigger than the heap size.	Correct operation of malloc and related functions depends on the working implementation of the 'sbrk' function. Newlib-nano (default C runtime library used by the GNU Arm Embedded toolchain) provides weak 'sbrk' implementation that doesn't check for heap and stack collisions during excessive memory allocations. To ensure the heap always remains within the range defined by __HeapBase and __HeapLimit linker symbols, provide a strong override for the 'sbrk' function:
	
	`#include <errno.h>`
	`#include <malloc.h>`
	
	`void *_sbrk_r(struct _reent *reent, ptrdiff_t diff)`
	`{`
	`  (void) reent;`
	
	`  extern uint32_t __HeapBase;`
	`  extern uint32_t __HeapLimit;`
	
	`  static uint8_t *_cur_brk = (uint8_t *)&__HeapBase;`
	`  static uint8_t *_heap_end = (uint8_t *)&__HeapLimit;`
	
	`  uint8_t *_old_brk = _cur_brk;`
	`  if ((_cur_brk + diff) > _heap_end)`
	`  {`
	`  return (void *)-1;`
	`  }`
	`  _cur_brk += diff;`
	`  return _old_brk;`
	`}`

Preproduction PSoC 6 devices do not support changing core voltage if the protection context (PC) does not equal 0.	See SYSPM driver API Reference for details. This also impacts dependent drivers and middleware: FLASH, BLE, Emulated EEPROM, and DFU.
The Watchdog Timer (WDT) lock state is not retained during system Deep Sleep power mode. After wakeup from system Deep Sleep, the WDT is locked.	To change the configuration of WDT, Clk LF, ILO, or PILO, unlock the WDT (call Cy_WDT_Unlock()) after a wakeup from system Deep Sleep.

The `Cy_SysClk_EcoEnable` and `Cy_SysClk_EcoGetStatus` functions return `success` status when ECO is disconnected.	Check the ECO_OK bitfield after the return success status:
	`cy_en_sysclk_status_t status = Cy_SysClk_EcoEnable(3000);`
	`if(status == CY_SYSCLK_SUCCESS)`
	`{`
	`  if(0UL == (SRSS_CLK_ECO_STATUS & SRSS_CLK_ECO_STATUS_ECO_OK_Msk))`
	`  {`
	`  /* There is something wrong with ECO */`
	`  status = CY_SYSCLK_TIMEOUT;`
	`  }`
	`}`

Known Issue	Workaround
Configuration of PDL build options in PSoC Creator is reset for every updated middleware.	Open the Peripheral Driver Library tab from the Project→Build Settings… menu, note the configuration for PDL 3.0.4, and select the missing checkboxes for the PDL 3.1 manually. The affected components are: Crypto, DFU SDK, emWin, Retarget I/O, and FreeRTOS.
ARM MDK linker returns error during build of DFU SDK with custom DFU linker script.	Remove `__asm void cy_DFU_mdkAsmDummy(void); the` function prototype from the dfu_mdk_common.h header file.
The USB Device ignores LPM requests after wake up from Deep Sleep.	Call USBFS driver Cy_USBFS_Dev_Drv_Lpm_SetResponse() after calling Cy_USBFS_Dev_Drv_Resume() to restore response to the LPM packets.Call USBFS driver Cy_USBFS_Dev_Drv_Lpm_SetResponse() after calling Cy_USBFS_Dev_Drv_Resume() to restore response to the LPM packets.
The USB Device modes with DMA do not work after wake up from Deep Sleep, due to incorrect restore of the ARB_CFG register.The USB Device modes with DMA do not work after wake up from Deep Sleep, due to incorrect restore of the ARB_CFG register.	Save the ARB_CFG values before entering Deep Sleep and restore after calling Cy_USBFS_Dev_Drv_Resume.
Em_EEPROM Read function may become stuck in the infinite loop.	See KBA 227502.
Em_EEPROM Read function returns incorrect data when restored from a Redundant Copy.	See KBA 227530.
The USB Configurator supports import from the HID Descriptor Tool. Current version 2.4 contains an error related to strings.	Before importing, check values of String Index, String Minimum and String Maximum items per spec HID 1.11 and fix these items manually in the file generated by the HID Descriptor tool.
The CY_BLE_EVT_STACK_BUSY_STATUS event may not return a CY_BLE_STACK_STATE_FREE state, if the application initiates an active connection (Peripheral/Central GAP role) along with Scan activity (GAP Observer) with a high duty cycle (scan window value is close to scan interval). Such behavior is observed only when BLE is operated in Dual CPU mode.	Increase the scan interval and reduce the scan window values to have a ratio of at least 1:2.
In some very rare cases, waking the system from Deep Sleep could potentially cause a hard fault. This is due to the system requesting early access to flash/sflash (before they are ready to be accessed).	See KBA 229335.

Device Support

The PDL includes:

- Device-specific header files that define all peripheral registers and bits in the device
- CMSIS-compliant startup code to initialize the system after a device reset and transfer the code execution to `main()`
- Linker files for each supported device and toolchain
- SVD files with a detailed description of peripherals, registers, fields, and bit values

This release supports devices in the PSoC 6 MCU architecture such as PSoC 62 and PSoC 63. It does not support CY8C6xx8, CY8C6xx9, CY8C6xxA. Use the Eclipse IDE for ModusToolbox, which contains the PDL and middleware within the PSoC 6 Software Library.

Peripheral Drivers

The PDL provides a high-level API to configure, initialize, and use a peripheral driver. The drivers are designed for peripheral IP blocks. They work on any PSoC 6 device that instantiates that IP block.

Driver	Description	API Functionality
CRYPTO	Cryptographic Operations	Perform cryptographic operations on user-designated data
CTB	Continuous Time Block	Configure and access the analog CTB
CTDAC	Continuous-Time DAC	Generate a 12-bit DAC output voltage from the reference
DMA	Direct Memory Access	Perform direct memory transfers
EFUSE	Electronic Fuses	Read the customer-accessible electronic fuses
FLASH	Flash Memory	Manage flash memory operations
GPIO	General Purpose I/O Ports	Configure and access device input/output pins
I2S	Inter IC Sound	Manage digital audio streaming to external I2S devices
IPC	Inter Process Communication	Manage data transfer between CPUs or processes in a device
LPCOMP	Low power comparator	Fast detection of voltage changes in both normal and ultra-low power operation
LVD	Low voltage detection	Monitor whether the VDDD voltage level is above the configurable threshold
MCWDT	Multi-counter watchdog timer	Manage counters to create a free-running timer or periodic interrupts
PDM_PCM	PDM to PCM converter	Convert one-bit digital audio streaming data to PCM data
PROFILE	Energy Profiler	Measure relative energy consumption of monitored operations
PROT	Memory Protection	Manage the MPU, Shared MPU (SMPU), and Peripheral Protection Unit (PPU)
RTC	Real Time Clock	Manage calendar dates and clock time
SAR ADC	SAR ADC Subsystem	Manage a fast 12-bit multichannel SAR ADC with sample rate of 1 Msps
SCB	Serial Communication Block	Manage serial communication as I2C, SPI, or UART
SMIF	Serial Memory Interface	Manage a SPI-based interface to external memory devices
SYSANALOG	System Analog Reference	Generate highly accurate reference voltages and currents for the analog subsystem
SYSCLK	System Clock	Manage system and peripheral clocks
SYSINT	System Interrupt	Manage interrupts and exceptions, in conjunction with the CMSIS core NVIC API
SYSLIB	System Library	Utility functions to handle delays, register read/write, asserts, software reset, silicon unique ID, and more
SYSPM	System Power Modes	Manage power modes and get power mode status
SYSTICK	Systick Timer	Manage a 24-bit down-counter timer
TCPWM	Timer Counter PWM & Quadrature Decoder	Manage a 16- or 32-bit periodic counter, PWM, or Quadrature decoder
TRIGMUX	Trigger Multiplexer	Manage the multiplexing of trigger outputs to specific trigger inputs across multiple peripherals

Driver	Description	API Functionality
WDT	Watchdog Timer	Manage a watchdog timer

Middleware and Other Software

The PDL includes the following middleware components:

Middleware	Description
Bluetooth Low Energy (BLE)	Cypress Bluetooth Low Energy (BLE) stack, along with a comprehensive set of APIs to configure the BLE stack and the underlying hardware. The BLE middleware also provides a general interface between the BLE application and the BLE stack module.
Emulated EEPROM	Cypress Emulated EEPROM middleware provides an API that allows creating an Emulated EEPROM in flash, with the ability to do wear leveling and restore corrupted data from a redundant copy.
Embedded graphic library (emWin)	emWin is an embedded graphic library and graphical user interface (GUI) framework designed to provide an efficient, processor- and LCD controller-independent GUI for any application that operates with a graphical display. It is compatible with single-task and multitask environments. Developed by SEGGER Microcontroller, Cypress has licensed the high-performance emWin library from SEGGER and offers it for free to customers.
USB Device	The USB Device middleware provides a full-speed USB 2.0 Chapter 9 specification compliant device framework. It uses the USBFS driver from PDL to interface with the hardware. The middleware provides support for Audio, CDC, and HID classes. Also, it allows implementation of other class support. The standalone USB Device Configuration tool is shipped in PDL to make it easy to construct the USB Device descriptors.
DFU SDK	The Device Firmware Update (DFU, previously Bootloader) SDK allows you to design flexible bootloading applications with varying levels of complexity.
RTOS	FreeRTOS v10.0.1 source code is integrated with the PDL.
Secure Image	The Secure Image is a project template intended to run only on the CM0+ secure processor in a multi-processor system. It sets up hardware and software protection for the system, validates the user application, and jumps to its starting point. The template provides a basic secure system sufficient for most applications. You may modify or replace the template to match the requirements of a specific user system.

Utilities

The utilities directory contains source files you can use to redirect standard I/O to user defined target hardware.

Tools

The tools directory contains applications you can use to configure a software component, or to perform post build processes for all supported toolchains.

Tool	Description
CyMCUElfTool	The build process for PSoC 6 devices uses the CyMCUElfTool. The tool post-processes linked ELF images to add data necessary for the boot process, perform security checking, and merge images for multiple cores into a complete image for an entire application. It also supports the PSoC 6 Bootloader SDK.
SMIF Configuration Tool	The SMIF Configuration Tool consists of two applications: SMIF Configuration GUI Tool and the command line SMIF Generation Tool. This tool allows you to generate input structures for the SMIF driver memslot API.
USB Configurator	The USB Configurator is also part of the ModusToolbox software. Use this tool to configure the USB Device descriptors. The USB Configurator generates header and source files that store the USB Device descriptors and other information used by USB Device middleware.

Supported Toolchains

- PSoC Creator 4.4
- IAR Embedded Workbench for ARM 8.32
- Keil Embedded Development Tools for ARM 5.25

Release Contents

The PDL is organized into several folders. The following table shows the PDL folder structure.

Path\Folder	Description
bootloader	Bootloader SDK
cmsis	CMSIS core access headers and DSP library
devices	Device header files, startup code, linker file, flash loader implementation, and the CMSIS SVD file for each device series
doc	PDL and other documentation
drivers	Driver source code and headers
middleware	Firmware development stacks, such as BLE
rtos	RTOS source code supported by the PDL, such as FreeRTOS
security	Basic secure system project template
tools	User-level applications; for example, to configure a software component or to perform post-build processing
utilities	Various utility files, such as standard I/O support

Documentation

PDL User Guide and API Reference are in the `\doc` folder of the PDL installation directory.

Technical Support

Visit https://community.cypress.com/community/psoc-6 for help and support.

Cypress Semiconductor
An Infineon Technologies Company
198 Champion Ct.
San Jose, CA 95134-1709

www.cypress.com
www.infineon.com