Peripheral Driver Library (PDL) Release Notes

Version 3.0.3

Release Date: June 6, 2018

Thank you for your interest in Cypress Peripheral Driver Library (PDL) version 3.0.3. This document lists the content of the release package.

Overview

Cypress provides PDL v3.x, which simplifies software development of PSoC 6 family of devices. This SDK release provides the following features:

- PSoC 6 digital and analog peripheral drivers, which enable rapid peripheral software development in Cypress PSoC Creator and third-party IDEs
- The ARM Cortex Microcontroller Software Interface Standard (CMSIS) core access header files, and CMSIS Digital Signal Processing (DSP) code directly from the CMSIS 5.0.1 release
- CMSIS compliant device header files, startup code (platform initialization) and device configuration header files
- Fully configurable Bootloader SDK that can readily be integrated for any application need
- FreeRTOS source code integrated with the PDL
- Bluetooth Low Energy (BLE) middleware and stack library
- Secure Image reference design, which demonstrates a Trusted Execution Environment (TEE) execution on the CM0+ core
- PDL Application Programming Interface (API) Reference Manual
- PDL User Guide

Note: PDL 3.0.3 does not support the FM microcontroller portfolios. Use PDL v2.1 to develop firmware for the FM0+ and FM4 MCU portfolios. Use PDL v2.0 for FM3 MCU portfolio support. All supported versions of the PDL are available through the PDL product page.

If you have technical questions, visit www.cypress.com/support for help or contact information.

New Features

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

Updated Drivers

- FLASH 3.11
- IPC 1.20
- SYSCLK 1.11
- SYSPM 2.21

Updated Middleware

- BLE 2.10
- Bootloader SDK 2.20

Updated Security

- Cypress Secure Image 1.10

Updated RTOS

- FreeRTOS 10.0.1

Design Impact and Errata

System Reserved Resources

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. Please refer to the PDL API Reference Guide for details.

SVD file limitation with some IDEs

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.

Changing core voltage limitation

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 Bootloader SDK.

RAM size for single core devices

RAM size for single core devices is increased by 0x80 in linker scripts for each supported compiler.

BLE stack errata

- Link Disconnected due to collision of Connection Update and PHY Update Procedures with Samsung S8 Phone.
- Memory allocation failure is observed during BLE Stack initialization, when DLE Rx Payload Size is less than Tx Payload Size.
- Connection timeout observed during second connection when PHY update and Connection Update procedures happens together in device (operated in slave role).

Device Support

The PDL includes:

- Device-specific header files that provide a complete definition of all peripheral registers and bits in the device
- CMSIS-compliant startup code to initialize the system after 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 all devices in the PSoC6 MCU architecture.

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; therefore, they work on all PSoC 6 devices that instantiate 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 date 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
WDT	Watchdog Timer	Manage a watchdog timer

Bootloader SDK

The bootloader SDK allows you to design flexible bootloading applications with varying levels of complexity.

Middleware

The PDL includes the following middleware components:

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.

RTOS

FreeRTOS v10.0.1 source code is integrated with the PDL.

Security

The PDL provides a basic secure system that consists of the following components:

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 the user-level applications you can use to configure a particular software component, or to do the post build process for all supported toolchains.

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.

Supported Toolchains

- PSoC Creator 4.2
- IAR Embedded Workbench for ARM 8.11
- Keil Embedded Development Tools for ARM 5.23
- GCC ARM Embedded 5.4-2016-q2-update
- iSYSTEM winIDEA 9.12

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
examples	Code examples organized by the supported starter kit
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 Manual are in the `\doc` subdirectory of the PDL installation directory.

Technical Support

For assistance, go to http://www.cypress.com/go/support.

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, 2018. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

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

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