CE220120 – PSoC 6 MCU: Blocking Mode Flash Write

Objective

This example demonstrates how to write to the flash memory of a PSoC® 6 MCU device. In this example, the flash write API function blocks the caller until the write is completed.

Overview

This example writes a flash row using a Peripheral Driver Library (PDL) API function. The function blocks the caller until the write is completed.

Requirements

Tool: PSoC Creator™ 4.2 with PDL 3.0.1

Programming Language: C (ARM® GCC 5.4.1 and ARM MDK 5.22)

Associated Parts: All PSoC 6 MCU parts

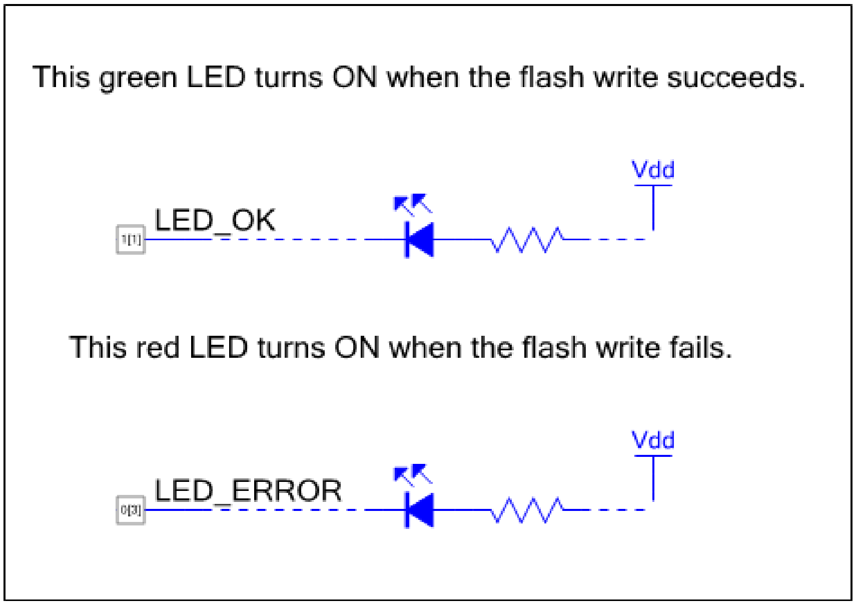
Related Hardware: CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit

Design

In this example, the ARM® Cortex®-M0+ core (CM0+) simply enables the Cortex-M4 core (CM4) and enters Deep Sleep. The CM4 core executes the implementation of this example.

The example uses a constant array with size equaling the size of one flash row. The main function places this array in the flash at an address such that it occupies one complete flash row, by calling the flash write API function. It then verifies the flash data by comparing the flash data with the written data. If the flash write is successful, the green LED turns ON. Otherwise the red LED turns ON. Figure 1 shows the PSoC Creator schematic for this code example.

Figure 1. TopDesign Schematic



This green LED turns ON when the flash write succeeds.

This red LED turns ON when the flash write fails.

Design Considerations

This code example is designed to run on CY8CKIT-062-BLE. To port the design to other PSoC 6 MCU family devices and kits, you must change the target device in the PSoC Creator Device Selector, and change the pin assignments in the *.cydwr* file.

This code example uses a green LED to indicate the successful flash write operation which does not operate at 1.8 V.

Hardware Setup

This example uses the CY8CKIT-062-BLE kit’s default configuration. Refer to the kit guide to ensure the kit is configured correctly.

Operation

1. Plug the CY8CKIT-062-BLE kit board into your computer’s USB port.
2. Build the project and program it into the PSoC 6 MCU device. Choose **Debug** >**Program**. For more information on device programming, see PSoC Creator Help. Flash for both CPUs is programmed in a single program operation.

   The green LED turns ON if the flash write is successful.
3. Set the value of the macro `MAKE_FLASH_WRITE_FAIL` in the *main_cm4.c* file to 1, program the device, and observe that the red LED turns ON to indicate that the flash write operation failed.

Components

Table 1 lists the PSoC Creator Components used in this example, as well as the hardware resources used by each.

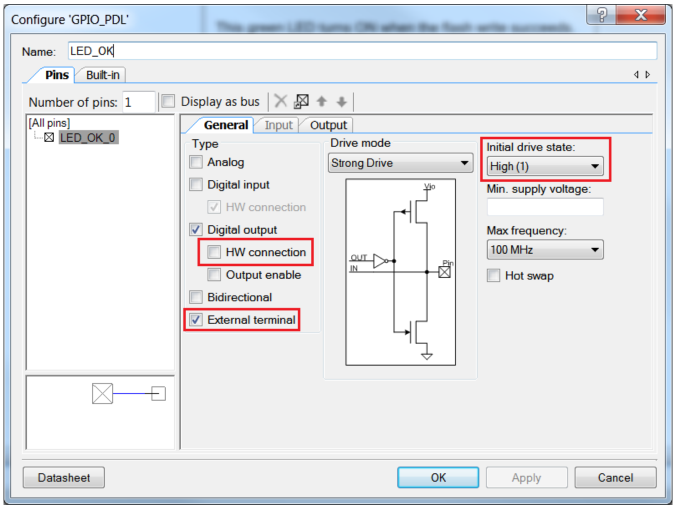
Table 1. PSoC Creator Components

Component	Instance Name	Hardware Resources
Digital Output Pin	LED_OK	1 GPIO pin
Digital Output Pin	LED_ERROR	1 GPIO pin

Parameter Settings

Figure 2 shows the Digital Output Pin Component configuration for the `LED_OK` instance with the changed settings highlighted. This setting also applies to the other instance of the Component, `LED_ERROR`. See the Pins Component datasheet for information on these settings. To access the datasheet, right-click the Component and click **Open Datasheet**.

Figure 2. Digital Output Pin Configuration



Configure 'GPIO_PDL'

Name: LED_OK

Number of pins: 1 □ Display as bus

[All pins]
☒ LED_OK_0

Type
□ Analog
□ Digital input
☑ HW connection
☑ Digital output
□ HW connection
□ Output enable
□ Bidirectional
☑ External terminal

Drive mode: Strong Drive

Initial drive state: High (1)

Min. supply voltage:

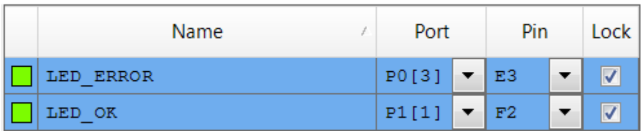
Max frequency: 100 MHz

□ Hot swap

Design-Wide Resources

Figure 3 shows the pin assignment for the project. These assignments are compatible with CY8CKIT-062-BLE.

Figure 3. Pin Assignments



	Name	Port	Pin	Lock
🟩	LED_ERROR	P0[3]	E3	☑
🟩	LED_OK	P1[1]	F2	☑

Related Documents

Application Notes	
AN210781 Getting Started with PSoC 6 MCU with BLE Connectivity	Describes PSoC 6 BLE devices and how to build your first PSoC Creator project
AN215656 PSoC 6 MCU Dual-Core CPU system Design	Describes the dual-core CPU architecture in PSoC 6, and shows how to build a simple dual-core design
AN219434 Importing PSoC Creator Code into an IDE for a PSoC 6 MCU Project	Describes how to import the code generated by PSoC Creator into your preferred IDE
PSoC Creator Component Datasheets	
Pins	Supports connection of hardware resources to physical pins
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual
Development Kit (DVK) Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	

Document History

Document Title: CE220120 - PSoC 6 MCU: Blocking Mode Flash Write

Document Number: 002-20120

Revision	ECN	Orig. of Change	Submission Date	Description of Change
*A	5859184	VAIR	08/21/2017	Initial public release

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

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, WICED, 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.