CE219517 – KitProg2 Power Monitoring

Objective

This code example demonstrates how to create a bootloadable PSoC® 5LP (KitProg2) project to monitor the power consumed by the PSoC 6 MCU on CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit.

Overview

This code example is implemented as a custom application for KitProg2. The design utilizes an external high side current-sense amplifier to sense the current consumed by the target PSoC 6 MCU across a high-precision resistor. The amplified signal is sampled and digitized by PSoC 5LP's on-chip delta-sigma ADC. The measured signal is processed in the firmware and then sent to the PC by using a USB-UART interface and displayed on the SerialPlot serial data plotting tool.

Requirements

Tool: PSoC Creator™ 4.2

Programming Language: C (Arm® GCC 5.4.1)

Associated Parts: PSoC 5LP family of devices

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

Design

Figure 1. Power Monitoring Architecture Block Diagram

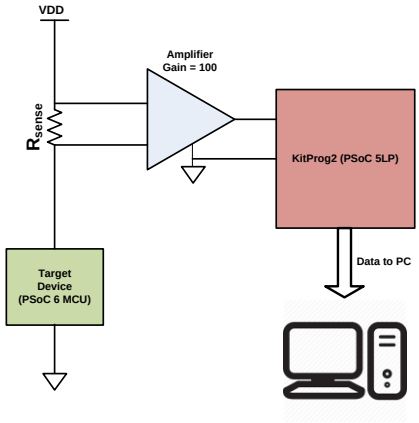


Figure 1 shows the power monitoring architecture. The load current is measured by converting the current to an equivalent voltage using a sense resistor (Rsense). The value of the sense resistor should be chosen such that the voltage drop across it is minimal so that the target device gets the required voltage for its operation. On the PSoC 6 BLE Pioneer Kit, an Rsense value of 200 mΩ has been used. For a current of 50 mA, it will drop only 10 mV across it. The sensed voltage is then amplified using an external high-side current-sense amplifier with a gain of 100.

The amplified signal is digitized using the delta-sigma ADC and is then processed in firmware. An adaptive filter is implemented in this design. The nature of the filter is adaptive in the sense that filtering is performed only if the current is less than a threshold current or the change in current is less than a threshold current change. The value of threshold current change is set such that it is above the system noise floor.

Figure 2 (a). and (b) shows the project schematic. Note that the schematic is developed using the KitProg2 custom application framework, which is available in the PSoC Programmer installation directory under *<Installation_Directory>\Programmer\Examples\Misc\KitProg2_Custom_App*. The power monitoring schematic is implemented in the 'Custom Application' page of the *TopDesign.cysch*. For more details, see the **Developing Applications for PSoC 5LP** section of KitProg2 user guide.

Figure 2 (a). KitProg2 Custom Application Framework

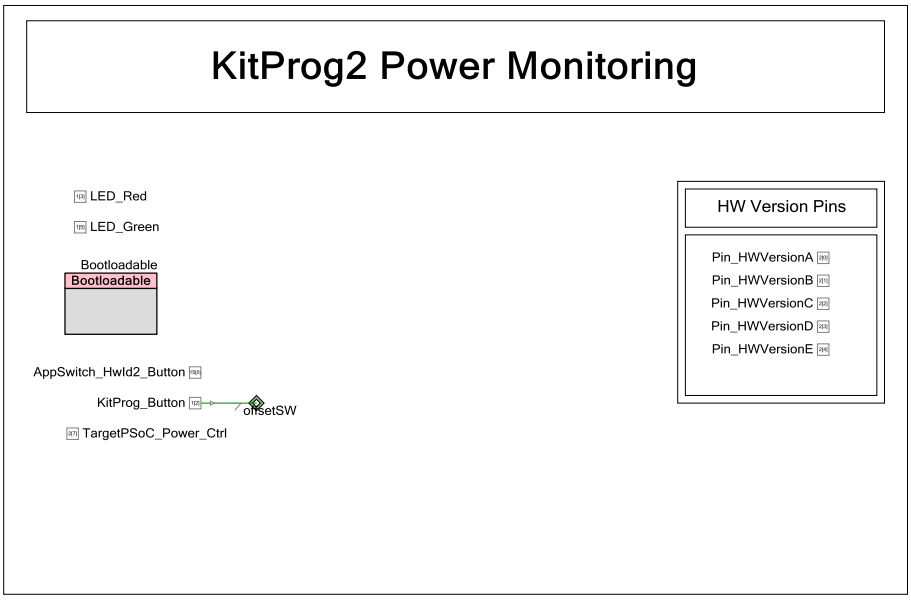
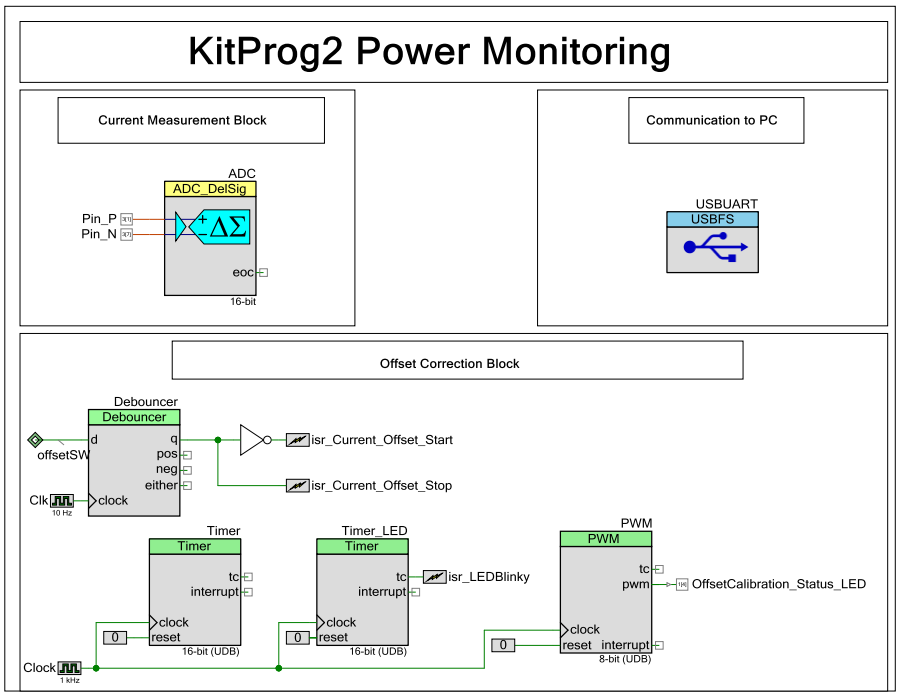
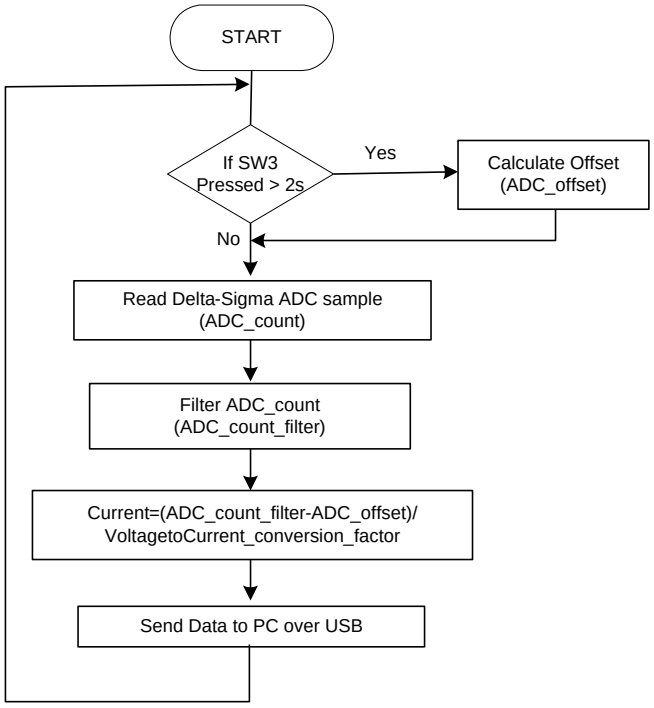


Figure 2 (b). KitProg2 Power Monitoring Schematic



Note: This design utilizes the custom application framework of KitProg2. If you need to develop your own application, use the KitProg2 custom application framework, which is available in the PSoC Programmer installation directory under <*Installation_Directory*>*Programmer\Examples\Misc\KitProg2_Custom_App*.

Figure 3. Flowchart for Power Monitoring Using PSoC 5LP



Hardware Setup

Use the PSoC 5 LP device on CY8CKIT-062-BLE to bootload and test the code example. Refer to the Operation section for instructions on how to bootload the project onto PSoC 5 LP. For more details refer to the KitProg2 user guide.

Software Setup

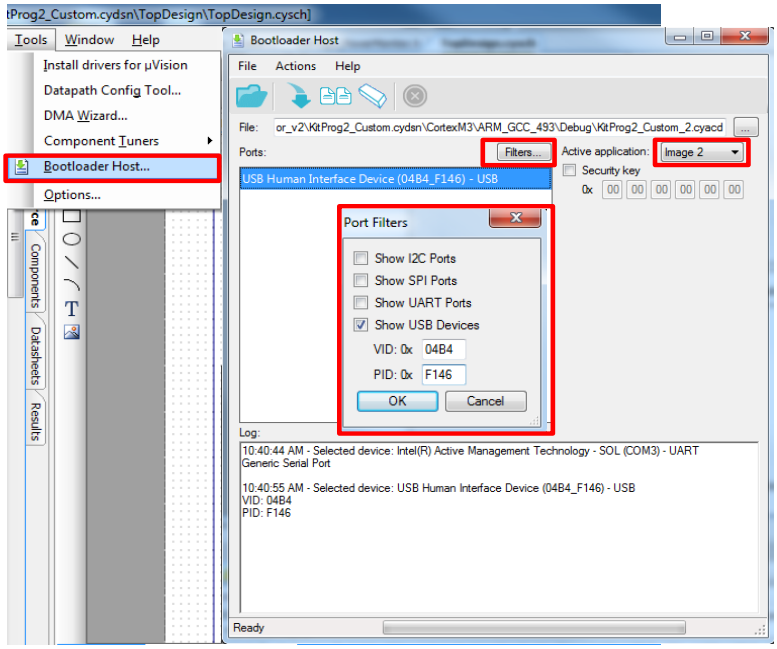
This code example uses the SerialPlot serial data plotting tool to plot the measured current. SerialPlot is an open source real-time plotting software. You can download SerialPlot from https://bitbucket.org/hyOzd/serialplot/downloads/.

Operation

Bootloading the project onto PSoC 5LP

1. Build the project in PSoC Creator by choosing **Build** > **Build Project** or pressing **[Shift] [F6]**.
2. To bootload the project onto the PSoC 5LP device, open the Bootloader Host tool, which is available in PSoC Creator. Choose **Tools** > **Bootloader Host** as shown in Figure 4.
3. Press and hold **SW3** while connecting the kit to the computer to enter the bootloader mode. The amber status LED will start blinking to indicate that PSoC 5LP is in bootloader mode.
4. In the Bootloader Host tool, click **Filters** and add a filter to identify the USB device. Ensure that **Show USB Devices** is enabled. Set VID as **04B4**, PID as **F146**, and click **OK**, as shown in Figure 4.
5. Set the **Active application** as Image 2 as shown in Figure 4. Selecting ‘Image 2’ means that the custom application will run by default after bootloading is complete.

Figure 4. Bootloader Host Tool in PSoC Creator



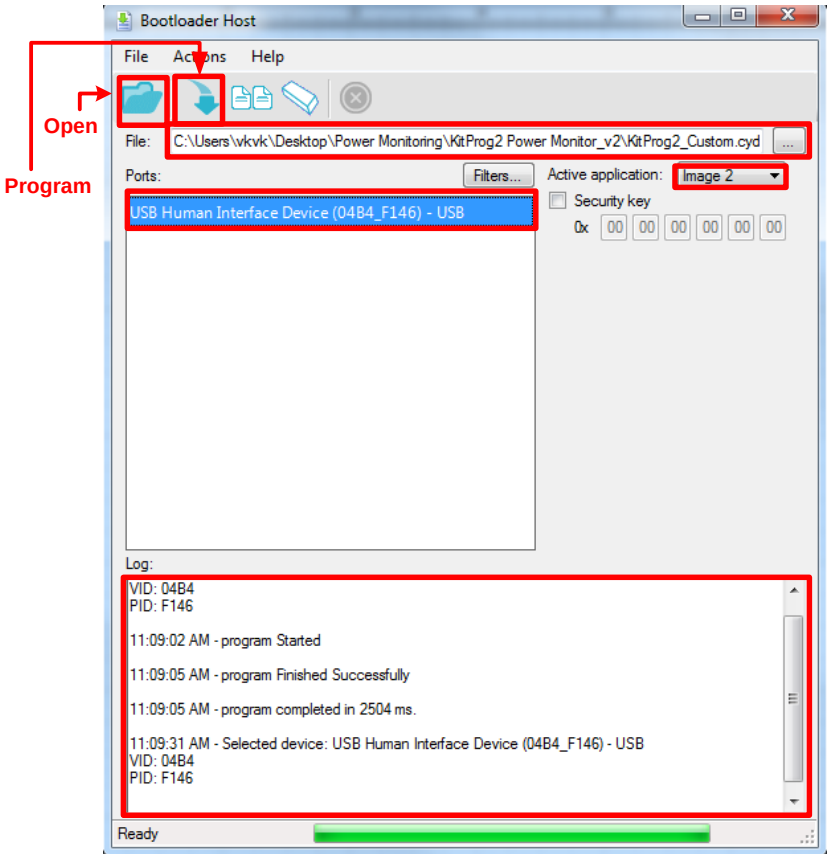
6. In the Bootloader Host tool, click **Open File** to browse to the location of bootloadable file (**.cycad*) as shown in Figure 5. The .cyacd file is available in the following path:

   <Project_Directory>\CE219517_KitProg2_Power_Monitoring.cydsn\CortexM3\ARM_GCC_541\Debug\ CE219517_KitProg2_Power_Monitoring_2.cyacd.

7. Select **USB Human Interface Device (04B4_F146) - USB** from the **Ports** list and click the **Program** button.

8. If the bootload is successful, the log displays “program Finished Successfully”; otherwise, it displays “Failed” and a reason for the failure.

Figure 5. Bootloading PSoC 5LP using Bootloader Host Tool



See the KitProg2 user guide for more details on bootloading applications onto PSoC 5LP.

Performing Power Measurement

1. Press the application selection button (**SW4**) to enter power monitoring mode (if KitProg2 is in programmer mode). When the kit is in power monitoring mode, **LED1** (Red) and **LED3** (Green) are turned ON.
2. Before starting measurement, you should perform offset correction. For this, remove jumper **J8** on the kit and then press and hold **SW3** for more than two seconds. By doing so, the program starts offset calculation, which is indicated by blinking of **LED2** (Amber). Note that offset correction needs to be performed whenever the kit is power cycled.
3. Once the offset is calculated, **LED2** stops blinking. Install the jumper **J8**. Now, the PSoC 5LP is ready for current measurement.
4. To use PSoC 5LP as a programmer once you are done with power measurement, press and release button (SW4) to switch back to programmer mode.

SerialPlot is used to display the measured current. Follow the steps below to read the Current value:

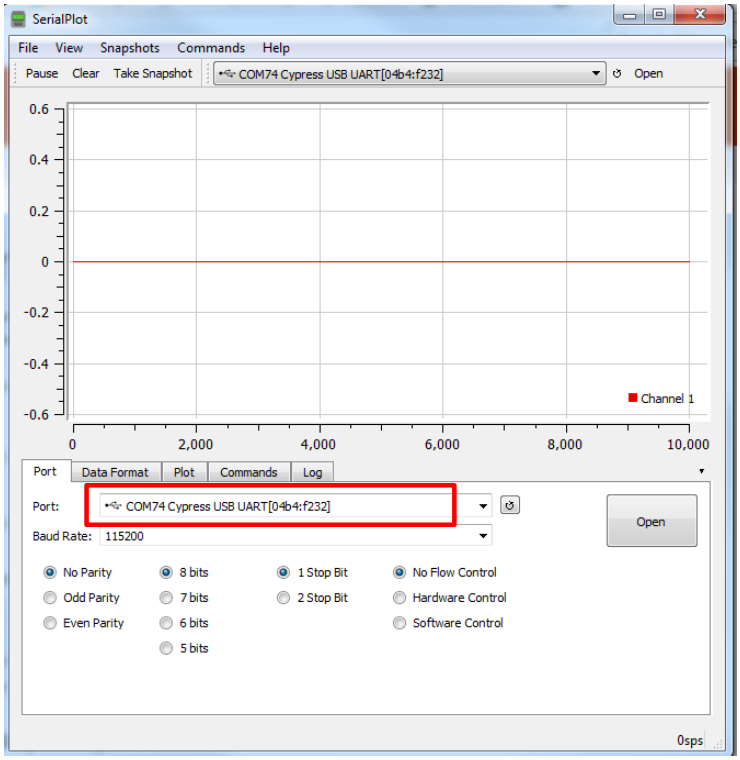
1. Open SerialPlot from **Start** > **All Programs** > **serialplot** > **SerialPlot**.
2. Select the Cypress USB-UART COM port as shown in Figure 6.

   Note: SerialPlot uses the “Cypress USB UART” COM port. If the SerialPlot does not detect the “Cypress USB UART” COM port automatically, you should manually install the driver. From the device manager, update the driver using the *USBUART_cdc.inf* file available in the project directory:

   <Project_Directory>\CE219517_KitProg2_Power_Monitoring.cydsn\Generated_Source\PSoC5

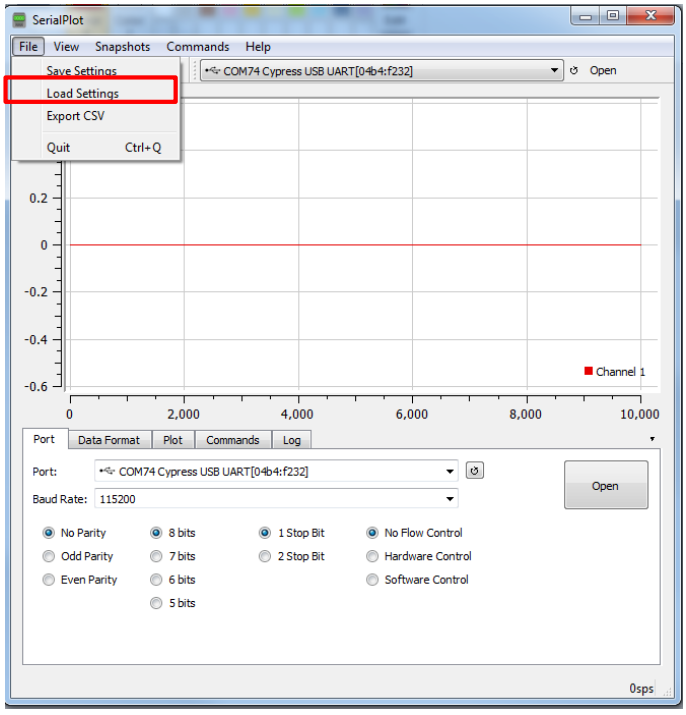
   However, this driver is not digitally signed. To use it, you should disable the driver signature enforcement on your PC.

Figure 6. Connecting to USB-UART COM Port of KitProg2



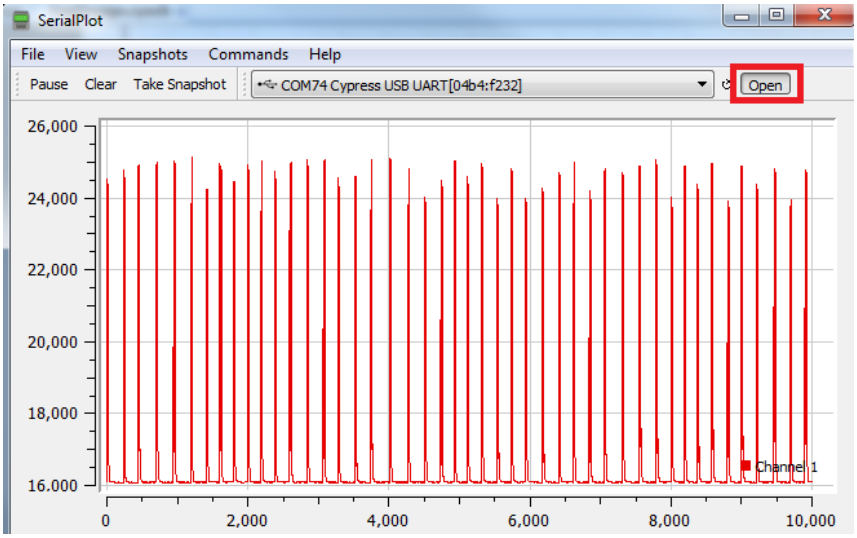
3. From the File tab, load the settings file *KitProg2_PowerMonitoring.ini* located at *<Project_Directory>\CE219517_KitProg2_Power_Monitoring.cydsn\SerialPlot_Configuration* as shown in Figure 7.

Figure 7. Loading the Setting File to Read the Measured Current



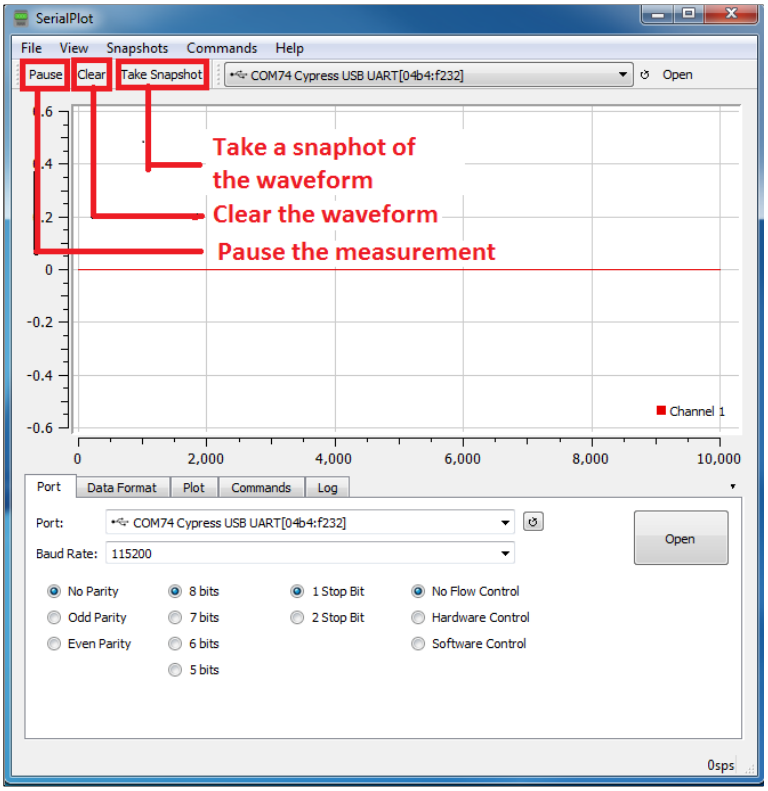
4. Click **Open** to start displaying the measured current. Measured current is displayed in units of µA. Figure 8 shows the current measured for a CE212736 - PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity - Find Me code example when the BLE is advertising.

Figure 8. Current Measured CE212736 as Displayed on SerialPlot



5. The SerialPlot tool provides options to pause the current measurement, clear the measured waveform and take a snapshot of the waveform. You can click and drag your mouse to zoom-in the graph and clicking the mouse anywhere on the graph gives you the value of the coordinates at that point. Figure 9 highlights these features.

Figure 9. Features of SerialPlot



Components

Table 1 lists the PSoC Creator Components used in this example in addition to the Components used in the KitProg2 Custom Application framework.

Table 1. List of PSoC Creator Components

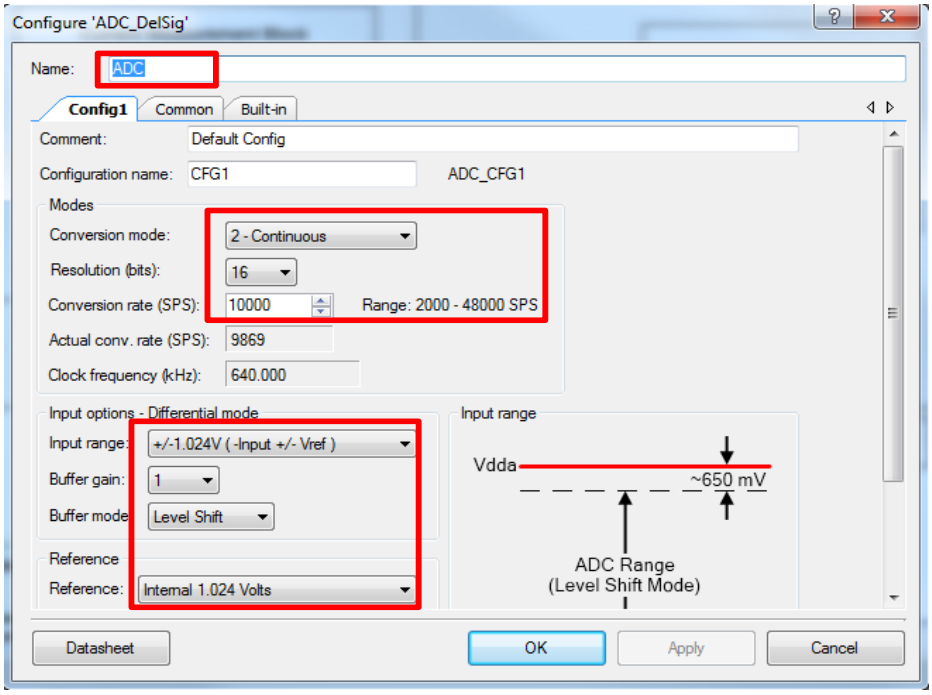
Component	Instance Name	Version	Hardware Resources
Delta Sigma ADC	ADC	3.30	Delta-Sigma Modulator
USBFS	USBUART	3.20	USB Block
Debouncer	Debouncer	1.0	UDB
Timer	Timer Timer_LED	2.80	UDB
PWM	PWM	3.30	UDB
Digital Input Pin	KitProg_Button	2.20	GPIO
Digital Output Pin	OffsetCalibration_Status_LED	2.20	GPIO
NOT (gate)	not_1	1.0	UDB
Interrupt	isr_Current_Offset_Start isr_Current_Offset_Stop isr_LEDBlinky	1.70	Interrupt Vectors
Clock	Clk Clock	2.20	Clock resource
Logic Low	cy_constant_1 cy_constant_2 cy_constant_3	1.0	Macrocell

Parameter Settings

Delta-Sigma ADC

Figure 10 shows the Delta-Sigma ADC configuration. The ADC is configured to have 16-bit resolution at a data rate of 10 ksps. The ADC is in differential mode with a differential input voltage range of +/-1.024 V.

Figure 10. Delta-Sigma ADC Configuration



USBFS

The USBFS Component is configured as a CDC interface. Figure 11 to Figure 14 show the configuration settings for the USBFS Component used in this project. For more details on configuring the USBFS Component as a USBUART interface, see the USBFS_UART code example provided with PSoC Creator.

Figure 11. USBUART Root Descriptor

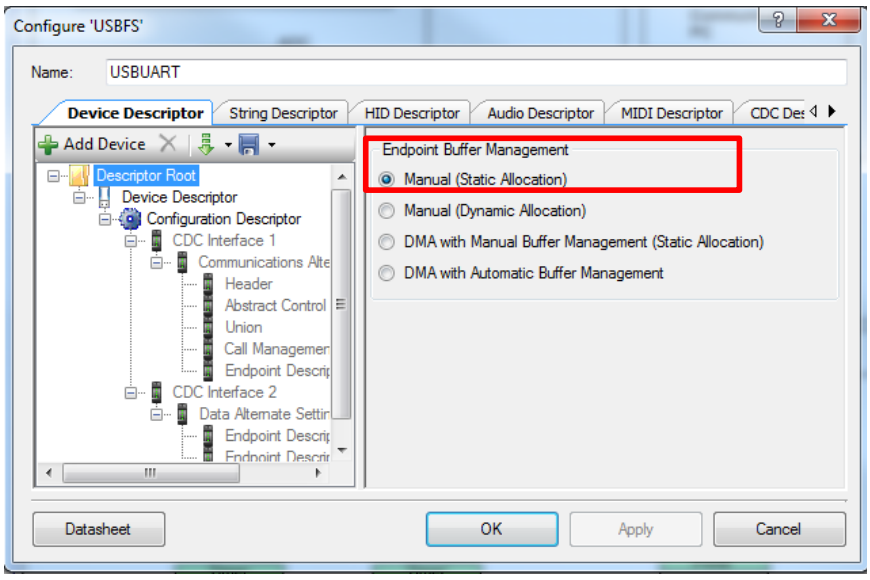


Figure 12. USBUART Device Descriptor

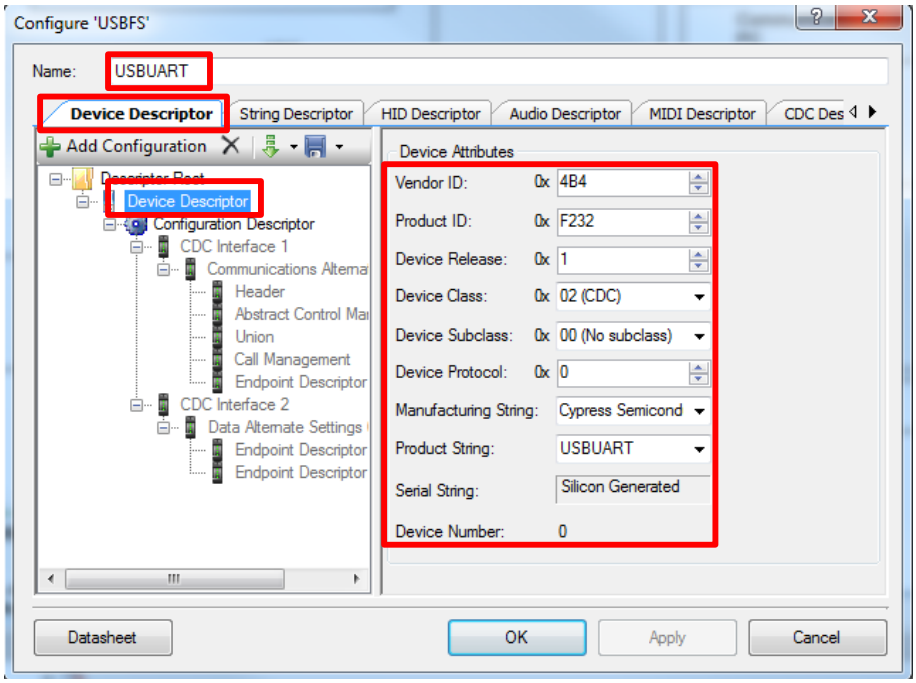


Figure 13. USBUART Configuration Descriptor

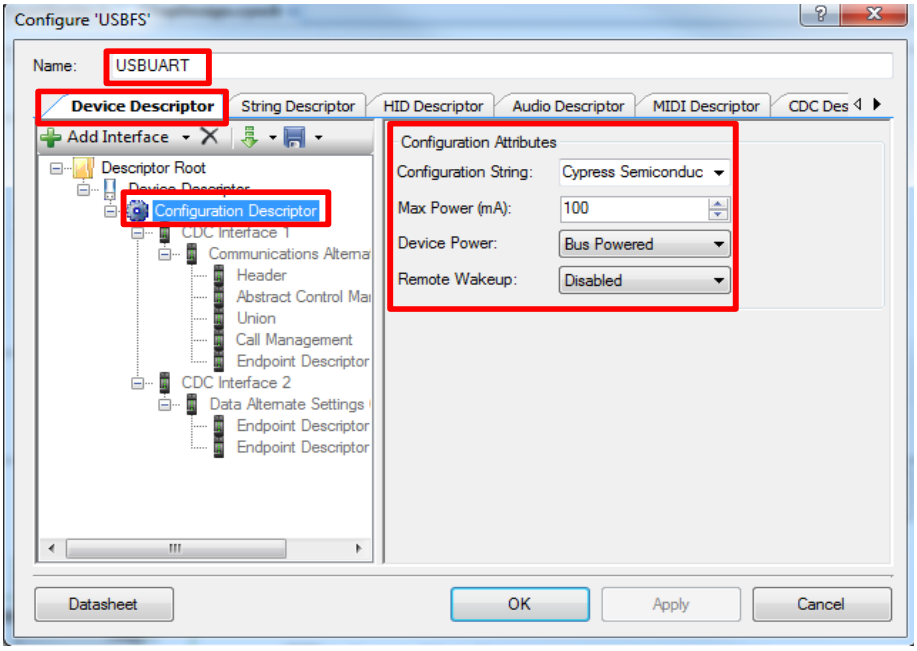


Figure 14. USBUART Interface Descriptor

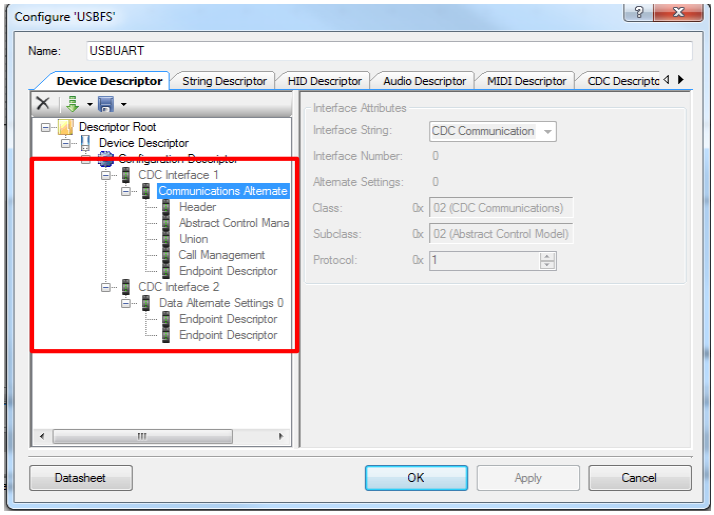


Table 2 shows the interface settings for the USBUART Component.

Table 2. USBUART Interface Setting

Interface	Endpoint	Direction	Transfer Type	Interval (ms)	Packet Size
CDC Interface 1	EP1	IN	Interrupt	10	8
CDC Interface 2	EP2	IN	Bulk	10	64
	EP3	OUT	Bulk	10	64

Offset Correction Block

The design implements offset correction of the current measurement as shown in Figure 2 (a). The design utilizes a Debouncer Component, one Digital Input Pin, one Digital Output Pin, two Timers, one PWM Component, three ISR Components, and a NOT gate.

To calculate the offset current, remove jumper **J8** on the PSoC 6 BLE Pioneer Kit. Pressing the mode selection button (**SW3** on PSoC 6 BLE Pioneer Kit) starts the Timer while releasing the button stops the Timer. When the button is released, if it had been held for more than two seconds the firmware begins offset calculation. The offset is calculated by measuring the average current when the target device (PSoC 6 MCU) is not drawing any current (by removing jumper **J8**). The offset calculation status is indicated by blinking an LED for two seconds with PWM and Timer Components. LED blinking is controlled using a PWM Component while a Timer Component is used to turn OFF the LED when offset correction is done.

Design-Wide Resources

Under **Pins** tab of **Design Wide Resources** assign the pins as shown in Figure 15.

Figure 15. Device Pin Assignment

Name	Port	Pin	Lock
\USBUART:Dm\	P15[7]	23	☑
\USBUART:Dp\	P15[6]	22	☑
AppSwitch_HwId2_Button	P15[0]	27	☑
KitProg_Button	P1[2]	13	☑
LED_Green	P1[5]	16	☑
LED_Red	P1[3]	14	☑
OffsetCalibration_Status_LED	P1[4]	15	☑
Pin_HWVersionA	P2[0]	62	☑
Pin_HWVersionB	P2[1]	63	☑
Pin_HWVersionC	P2[2]	64	☑
Pin_HWVersionD	P2[3]	65	☑
Pin_HWVersionE	P2[4]	66	☑
Pin_N	P3[7]	37	☑
Pin_P	P3[1]	30	☑
TargetPSoC_Power_Ctrl	P2[7]	2	☑

Related Documents

Application Notes	
AN77759 – Getting Started with PSoC 5LP	Describes the PSoC 5LP
PSoC Creator Component Datasheets	
PWM	Supports fixed block and UDB-based PWMs
Interrupt	The Interrupt component defines hardware triggered interrupts.
Pins	Supports connection of hardware resources to physical pins
Delta-Sigma ADC	Provides low power, low noise, precise analog to digital conversion
USBFS	Supports USB 2.0 Full Speed Communication
Debouncer	Eliminates unwanted oscillations on digital input lines
NOT (gate)	Logic gates provide basic Boolean operations
Timer	Supports basic timer function and advanced features such as interrupt generation
Digital Constant	Provides a convenient way to represent digital values in designs
Device Documentation	
PSoC 5LP Datasheets	PSoC 5LP Technical Reference Manuals
Development Kit (DVK) Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	

Document History

Document Title: CE219517 - KitProg2 Power Monitoring

Document Number: 002-19517

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5767816	VKVK	08/23/2017	Initial release
*A	6000997	VKVK	03/15/2018	Updated template and minor text changes. Updated project to PSoC Creator 4.2 Updated template

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 MCU

Cypress Developer Community

Community 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

© Cypress Semiconductor Corporation, 2017-2018. 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. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. 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.