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.

CE97088 - PSoC® 4 ADC to PWM DMA Transfer

Objective

This code example demonstrates how to set up a PSoC® 4 direct memory access (DMA) channel to transfer data from the ADC peripheral to the PWM.

Overview

This code example demonstrates a peripheral-to-peripheral data transfer using PSoC 4 DMA. The DMA transfers the ADC result data to a PWM compare register. The change in ADC value creates a change in PWM duty. The transfer is triggered on each ADC end of conversion (EOC). This example is a use case of the “Single data element per trigger” mode.

Requirements

Tool: PSoC Creator™ 3.2.

Associated Parts: PSoC 4200M and PSoC 4100M

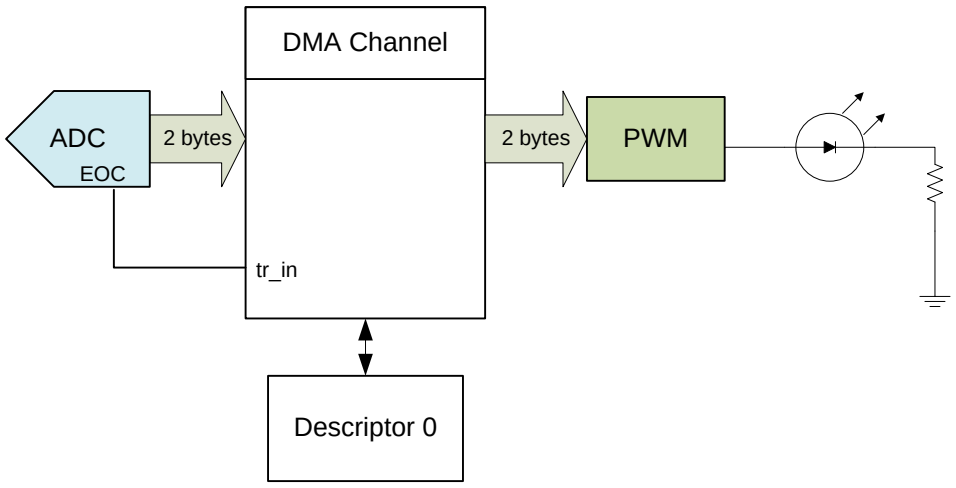
Related Hardware: CY8CKIT-044

Design

Every DMA channel in PSoC 4 has two descriptor structures. The descriptor comprises information regarding the source and destination address, the modes of transfer, and other specifics related to a transfer. You can choose to use one or both of the descriptors in the channel. Refer to the DMA Component datasheet for more details on the use of descriptors.

In this code example, a single DMA channel is configured with a single descriptor, which has the source address as the ADC result register and the destination as the PWM compare register. The trigger (tr_in) for the DMA is the ADC EOC. The transfer mode is set as “Single data element per trigger,” which results in the ADC result being transferred to the PWM compare register every time the EOC is asserted. Figure 1 illustrates the concept of this code example.

Figure 1. ADC-to-PWM DMA Transfer

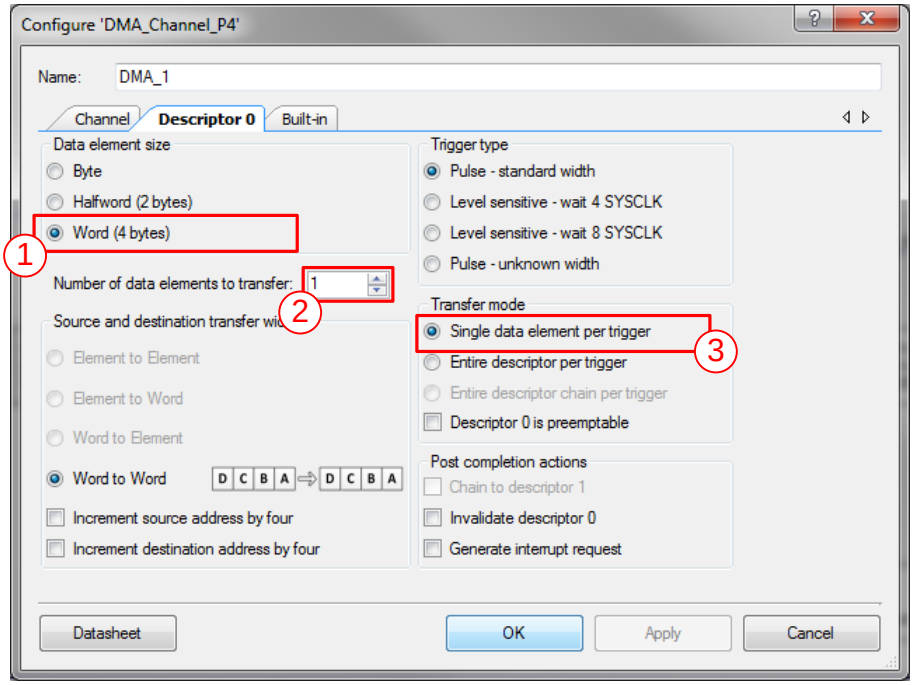


The SAR ADC is set up to run at 218 sps and with a resolution of 12 bits. The PWM is implemented using the TCPWM and is clocked at 12 MHz. The period of the PWM is selected to match the resolution of the SAR ADC. The sample rate of the SAR ADC is selected as very slow, compared to the PWM frequency.

Figure 2 shows the DMA Component configuration.

1. The ADC result and the PWM compare register are both addressed as 32-bit registers in PSoC 4. Hence the **Data element size** in the DMA configuration is set as 4 bytes. Though the relevant bits in the registers are only the lower 16 bits, the transfer needs to be 32 bits since 16-bit accesses from the peripheral registers are not supported.
2. The descriptor is configured to transfer one data element before being considered complete. The next EOC retriggers the DMA channel and the descriptor for another single-element transfer.
3. The **Transfer mode** is selected as a “Single data element per trigger.” This means that one data element (a 32-bit word) is transferred from the ADC to the PWM compare register each time the trigger (ADC EOC) is asserted.

Figure 2. DMA Configuration



The only code required for the DMA is to call the Start routine, available in the Component application programming interface (API). The start function has source and destination addresses as parameters for the transfer.

```
DMA_1_Start((void*)ADC_SAR_Seq_SAR_CHAN0_RESULT_PTR,(void*)PWM_COMP_CAP_PTR);
```

The register names for the ADC result and PWM compare registers are located in the “Register” section of the corresponding Component datasheets. The Registers Technical Reference Manual (TRM) includes the register names and addresses.

The source and destination addresses can also be set using the API functions DMA_SetSrcAddress() and DMA_SetDstAddress().

Design Considerations

No additional code is required after the DMA initialization, when the CPU is free to do other tasks. The DMA can function in the PSoC 4 Sleep mode, so you can put the device to sleep after initialization to reduce system power consumption.

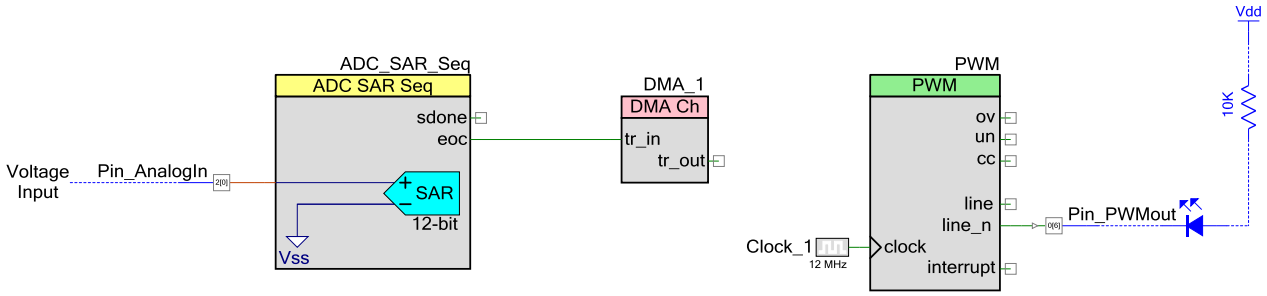
Note When the voltage on the ADC is close to zero, there is a chance that the ADC result will go negative due to a negative offset. This can create an unexpectedly large PWM duty cycle.

Hardware Setup

The hardware setup explained in this section is for use with the CY8CKIT-044. Figure 3 shows the project schematic.

- Input setup: Connect a voltage between Vdd and GND to P2[0], for example, using a potentiometer or signal generator.
- Output setup: The PWM output is already connected to pin P0[6], which connects to the Red LED on CY8CKIT-044.

Figure 3. Project Schematic



Operation

Changing the input voltage will change the brightness of the Red LED on the CY8CKIT-044.

Note When the voltage on the ADC is close to zero, there is a chance that the ADC result will go negative due to a negative offset. This will cause the LED to step to very high brightness when the input voltage is very close to zero.

Components

Table 1 lists the PSoC Creator Components used in this example, as well as the hardware resources used by each. The DMA Component settings are shown in Figure 2.

Table 1. PSoC Creator Components

Component or User Module	Hardware Resources
ADC_SAR_SEQ	SAR ADC
DMA	1 DMA channel
PWM	TCPWM

Related Documents

Table 2 lists the relevant application notes, code examples, knowledge base articles, device datasheets, and Component datasheets.

Table 2. Related Documents

Application Notes	
AN79953 – Getting Started with PSoC 4	Introduces you to PSoC 4, an ARM® Cortex™-M0 MCU based programmable system-on-chip. It helps you explore the architecture and PSoC Creator development tools.
Code Examples	
CE95351 – Fixed Function PWM with PSoC 4	
CE95275Sequencing SAR ADC and Die Temperature Sensor with PSoC 4	
CE95272 – SAR ADC in Differential Mode using Pre-Amplifier with PSoC 4	
PSoC Creator Component Datasheets	
PSoC 4 Sequencing Successive Approximation ADC (ADC_SAR_Seq)	
PSoC 4 Timer Counter Pulse Width Modulator (TCPWM)	
Device Documentation	
PSoC 4 Datasheets	
PSoC 4 Technical Reference Manuals	
Development Kit (DVK) Documentation	
CY8CKIT-044 – PSoC 4 M-Series Pioneer Kit	

For questions or suggestions on this code example please contact qvs@cypress.com.

Document History

Document Title: CE97088 - PSoC® 4 ADC to PWM DMA Transfer

Document Number: 001-97088

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	4866747	QVS	08/13/2015	New code example.
*A	5742438	AESATMP9	05/19/2017	Updated logo and copyright.

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

©Cypress Semiconductor Corporation, 2015-2017. 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, 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.