Subsystem Design

DMA Ping Pong With ADC

Description

The DMA Ping Pong with ADC example demonstrates how to use the DMA to transfer ADC data between two different buffers, also known as a DMA "Ping Pong". A DMA Ping Pong is commonly used to transfer data to one buffer while the CPU is working with the other buffer. As shown in Figure 1, the blue path shows the DMA transfers data to Buffer 1 and the CPU gets data from Buffer 2. When the paths switch, the DMA transfers data to Buffer 2 and the CPU gets data from Buffer 1. The benefit to this technique is faster total application runtime because the CPU is free to operate on a section of data at all times. In this example, the ADC is configured in single conversion mode and the DMA and CPU will switch between buffers after each conversion. Download the code for the DMA Ping Pong example.

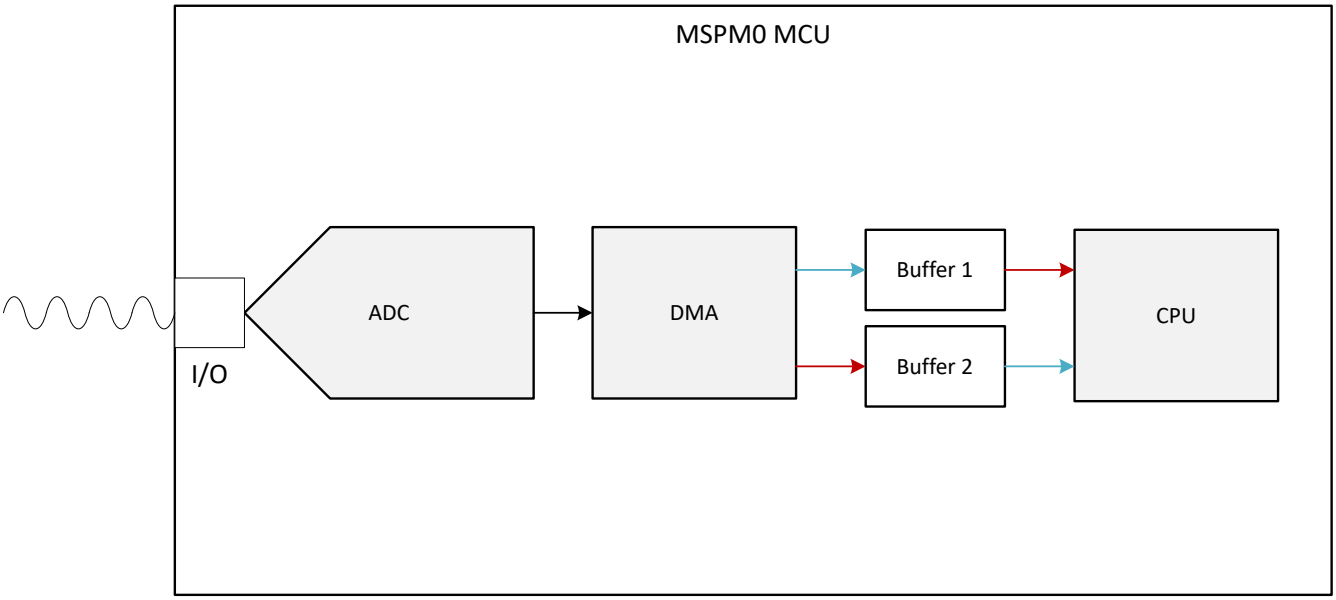

Figure 1. Subsystem Functional Block Diagram

Required Peripherals

This application requires the integrated ADC and DMA. The internal VREF is an additional option for the ADC reference, if a different reference value is required.

Table 1.

Sub-block Functionality	Peripheral Use	Notes
Analog Signal Capture	ADC	Called ADC12_0_INST in code
Moving memory	DMA	Full featured DMA channel is required to utilize the PREIRQ functionality. The example can be altered to work without the PREIRQ.

Compatible Devices

Based on the requirements in Table 1, the compatible devices are listed in Table 2. The corresponding EVM can be used for quick evaluation.

Table 2.

Compatible Devices	EVM
MSPM0Lx	LP-MSPM0L1306
MSPM0Gx	LP-MSPM0G3507

Design Steps

1. Determine the configuration for the ADC including reference source, reference value, resolution, and sampling rate based on the given analog input and design requirements.
2. Generate 2 array buffers to store the ADC data and set the buffer size and DMA transfer size the same so the DMA fills the whole buffer.
3. Configure the ADC in SysConfig based on the project requirements discovered in Step 1.
4. Configure the DMA in SysConfig in the ADC section.
5. Write Application Code to dynamically change the destination address of the DMA to alternate between buffers. See the Figure 2 for an overview or view the code directly.

Design Considerations

1. Maximum Sampling Speed: The sampling speed of the ADC is based on input signal frequency, analog front end, filters, or any other design parameters that affect sampling.
2. ADC Reference: Choose the reference to align with the expected maximum input to utilize the full scale range of the ADC.
3. Clock Settings: The clock source determines the total time for the conversion. The clock divider in tandem with the SCOMP setting determines the total sampling time. SysConfig sets the appropriate SCOMP depending on the sampling time setting.

Software Flowchart

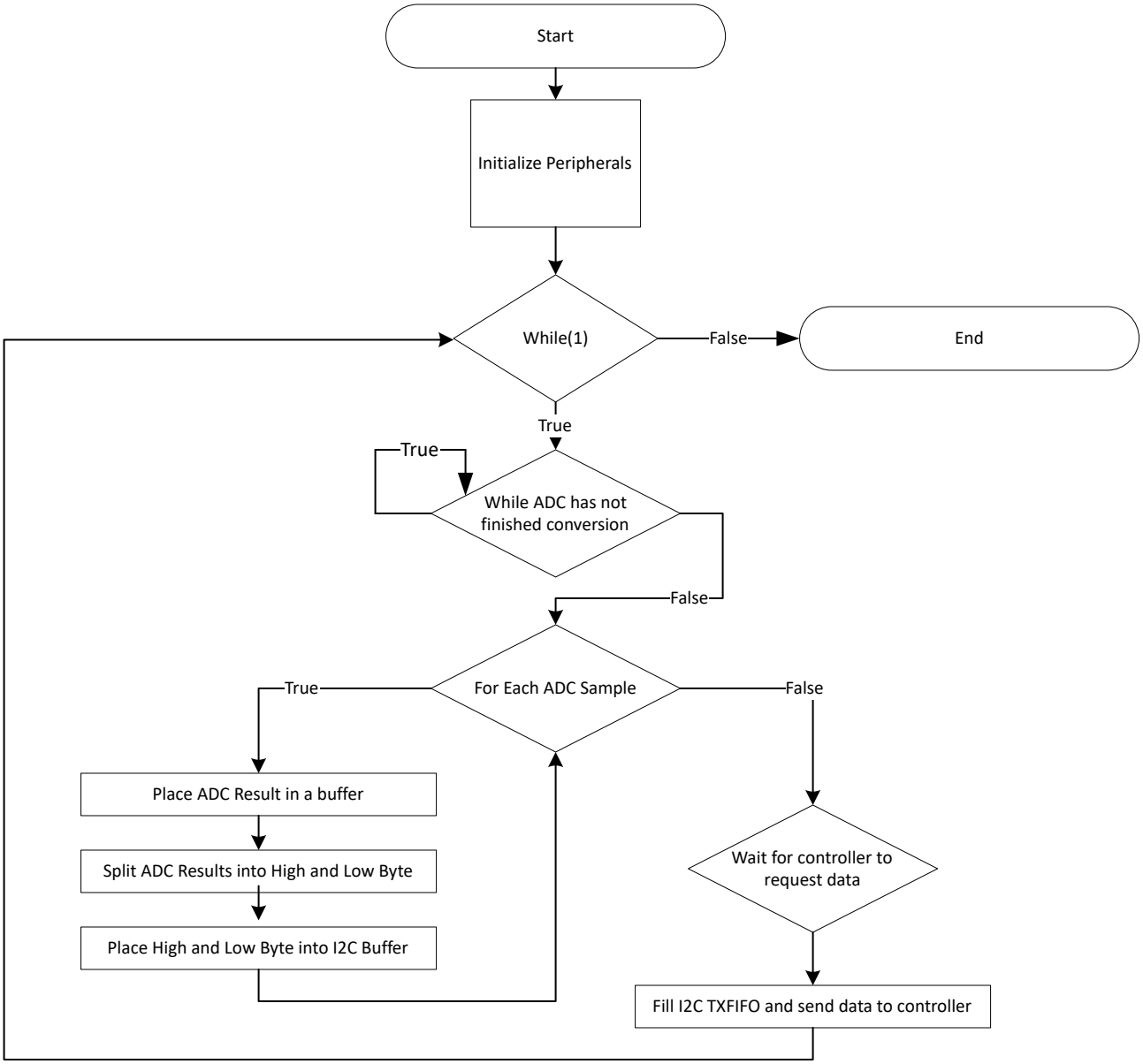

Figure 2. Application Software Flowchart

Design Results

Below are the results of the code executing. Figure 3 shows the results of the first buffer after the ADC readings complete. After this point the code will swap the DMA destination to the second buffer and the CPU will now be free to act on the first buffer.

Expression	Type	Value	Address
gADCSamplesPing	unsigned short[64]	[2702,0,2702,2698,2701...]	0x20000000
gADCSamplesPong	unsigned short[64]	[0,0,0,0,0...]	0x20000080

Figure 3. First Buffer Filled by the DMA

Figure 4 shows the results of the second buffer after the ADC readings complete. The code swaps the DMA destination back to the first buffer and now the CPU can execute on the second buffer.

Expression	Type	Value	Address
gADCSamplesPing	unsigned short[64]	[2702,0,2702,2698,2701...]	0x20000000
gADCSamplesPong	unsigned short[64]	[2698,2699,2700,2699,2699...]	0x20000080

Figure 4. Second Buffer Filled by the DMA

Additional Resources

- Download the MSPM0 SDK
- Learn more about SysConfig
- MSPM0L Launchpad
- MSPM0G Launchpad
- MSPM0 ADC Academy
- MSPM0 DMA Academy

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265