Project 3

ADC, DMA and DSP

ECEN 5813, Fall 2018

Prof. Kevin Gross

Revised 2018-12-07

# Introduction

In this project you will set up a channel of the KL25Z onboard analog-to-digital converter (ADC) for continuous sampling of an input signal. Sampled data will be moved from the ADC to memory using the direct memory access (DMA) controller. An application you write will examine the received data and perform analysis and print real-time information about the input signal out the UART.

# Guidelines

* Since this project uses the UART to output results, you may choose to base your project on your Project 1 port to KDS, Project 2 or start from scratch.
* Use the Kinetis board support package
* Follow ESE coding guidelines
* Have boundary checks on arguments and NULL pointer checks if required
* Use MACROS instead of hard coding other values. Macro names should begin with module name. For example, DMA\_xx, ADC\_xx.
* Work should be committed to the repository throughout development but at a minimum after completion of each part as outlined below.
* Students may work in groups. Groups may share information but each group must do their own work on the project.

# Part 1: Block diagram

In this project we will be receiving continuous data from the onboard ADC into memory. An interrupt triggers data processing by the application and results of the analysis are outputted to the UART. Draw a block diagram featuring these elements and indicate the flow of data through them.

# Part 2: ADC setup

Configure processor pins 13 and 14 for use with differential-mode channel 0 of the ADC. This signal is available at pins 1 and 3 of J10 on the FRDM development board. Configure the ADC to continuously sample channel 0 with 16-bit resolution. Sampling frequency is determined by the clock provided to the ADC, the programmed divide factor, channel and sampling mode. The default ADC clock is the 48 MHz bus clock. A sampling frequency of 8 kHz or higher should be adequate for this project.

An ADC needs reference voltages. On FRDM they are pre-connected to the 3.3v supply and ground. Signals applied should not exceed the 3.3v power supply or go negative with respect to the FRDM's ground. **Damage may result if these limits are not respected.** A 10 kOhm series resistor and attention to ESD will provide some protection to the processor. See [this article](https://www.digikey.com/en/articles/techzone/2012/apr/protecting-inputs-in-digital-electronics) for additional background.

Do not enable any interrupt or DMA requests at this point. All data will be retrieved through polling in this part of the project.

Write a simple application that prints raw sample values to the UART. Note that ADC data is in the 16 least-significant bits of the ADC data result register (ADCx\_Rn). Results should be displayed as a signed number with range -32768 to 32767. Monitor the ADC digital output as you apply an analog signal to the ADC inputs.

## Questions

1. What is the behavior of the system when no input signal is applied?

# Part 3: DMA setup

In this part you set up a buffer for incoming ADC data and configure DMA channel 0 to transfer a block of samples from the ADC to the buffer. A buffer size of 64 samples or more should be adequate.

Set up the DMA multiplexer to route the DMA request from the ADC to DMA channel 0. Configure the source address for the channel to be the ADC data register with no address update after each transfer. The destination address should be the beginning of the buffer with address increment and modulo update after each transfer.

The DMA controller supports independently configurable source and destination data sizes of 1, 2 or 4 bytes. Although the ADC data register presents 16-bit data, it may be read as a 32-bit word with the MS 16 bits set to 0. Read ahead and understand the requirements of the rest of the project before making a configuration decision here.

Trigger the DMA controller to transfer one buffer worth of data. Use the IDE to examine the data and verify that it is consistent with your results from part 2.

## Questions

1. What source and destination DMA read and write size did you choose and why?

# Part 4: Interrupt service

Configure the DMA controller to generate an interrupt when the buffer is filled. Acknowledge the interrupt and configure DMA to do another transfer from ADC to the same buffer. Repeat. Set a GPIO pin at the start of the ISR and clear it once DMA has been restarted. Monitor this signal with an oscilloscope to verify expected frequency, duration and consistency of the interrupt service. **Take a screenshot of your oscilloscope output and include it in your lab report.**

## Questions

1. What is the consequence of a lengthy or delayed interrupt service for the data retrieved by the DMA controller?

# Part 5: Application

Write an application examines each sample in the data buffer and performs analysis on the data. We will be doing a peak-level metering on the data.

Peak level is a measurement the absolute value of the highest sample value observed over an interval. To produce a continuously updated peak level reading, the reading is increased instantaneously when a sample with an absolute value higher than the current peak level reading is encountered. Otherwise, peak level is decreased using a first-order decay to zero.



Where the coefficient alpha is between 0 (instant decay) and 1 (infinite hold). The decay calculation can be performed once per buffer period.

The application should update the reading for each buffer of samples processed. Reading should be in absolute sample value. For our 16-bit converter, this will be a number in the range 0-32767. The application should provide continuous updates of this reading as text through the UART as frequently as the output baud rate allows.

## Questions

1. What effect does changing the buffer size have on system processing efficiency?
2. What other effect(s) does changing the buffer size have on the system?

## Extra credit

Perform a logarithmic calculation using a lookup table to report signal levels in [dBFS units](https://en.wikipedia.org/wiki/DBFS). Also report the linear (0-32767) version so that graders can verify your results. (10 points)

# What to turn in

1. Brief lab report with block diagram, oscilloscope screenshot and answers to questions posed here in the project description
2. Code dump report in txt or PDF format
3. Github repo link, should be tagged
4. KDS zipped project

# Grading rubric

1. Block Diagram: 10 points
2. Oscilloscope screenshot 10 points
3. Final application: 20 points
4. Lab questions: 10 points
5. Demo: 20 points
6. Extra credits(max): 10 points