

Lab Exercise #3

Objectives

- Collect data from multiple A/D channels
- Display collected data on a virtual terminal to verify correct data capture
- Determine how long it takes to sample and convert 9 A/D channels

References

- [1] NXP Kinetis KL25 processor sub-family data sheet (pdf) on class web page.
- [2] NXP Kinetis KL25 processor sub-family reference manual (pdf) on class web page.
- Textbook, particularly pages 164-173

Lab 3 Problem Statement

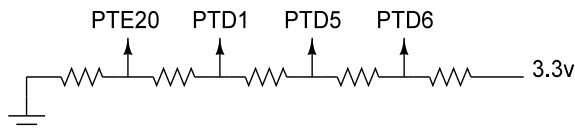
Design, implement, and verify software (firmware) that will sample 9 analog signals and save a 12-bit value for each in a uint32_t integer array. Future application routines will use those values as data for navigation routines. Create a main program to demonstrate operation of your firmware and facilitate its verification by sending data via UART2 to a monitoring terminal.

Create a function that when called will sequence through the 9 analog signals converting each and saving the resulting data. The desired analog signal must be selected, the ADC started, and when conversion is complete the data saved. Use polling to determine when conversion is complete. This sequence is to repeat 9 times.

Design Flow

Keeping in mind the “walk before you run” concept when developing new capabilities, I suggest that the following sequence be followed.

- Write an `init_ADC` routine to initialize the A/D subsystem
- Write a `scan_ADC` routine that will sequence through four analog inputs: PTE20, PTD1, PTD5, PTD6 capturing values in an array and returning them to a calling program. Poll the ADC each time to know when a conversion is complete. Install a debug signal in `scan_ADC` by setting PTD7 high at the beginning of `scan_ADC` and setting it low again after the last value is read from the ADC. PTD7 can then be monitored with the oscilloscope to determine how long it takes to select and scan all the specified channels
- In `main.c` call `init_ADC` and then in the `while(1)` loop call `scan_ADC` followed by sending the four values via UART2 to a terminal where values can be viewed. Place a delay in the loop so values are displayed at a reasonably rate on the terminal.
- Test operation by converting 4 analog voltages created with a voltage divider circuit shown here:



Five resistors, each 100 ohms or about that.

- e) Once the `init_ADC` and `scan_ADC` routines are working with 4 channels add the additional 5 channels. The values returned for the 5 channels will not be defined since their respective pins will not be connected to a known voltage source. Suggestion: have 2 “good” channels converted first, then the 5 that are unconnected, followed by 2 more “good” channels where good means connected to a known voltage.

To Turn In

- In the “comment header” of your `main.c` file report success, failure, or other observations
- Submit your `main.c` file to a D2L drop box
- Zip up your complete lab 3 uVision project and submit to the D2L drop box