ECEN 5803 – Mastering Embedded Systems Architecture – Project 1

Report on
Evaluation of KL25Z128M4 board
For the Sierra Instrumentation's
Vortex Flowmeter

Table of Contents

1. Executive Summary

This report outlines the various tests that were performed on the KL25Z128M4 to test its feasibility for this project. Based on our analysis and rigorous tests, we conclude that this microcontroller is an ideal candidate for the Vortex Flowmeter by Sierra Instrumentation. The microcontroller has several features that make it the most suitable for the project. The board has many interfaces to offer which allows connectivity with external devices. The chip in itself has a temperature sensor built into it which is used to determine various factors to calculate the flowrate. The software support for this device is extensive and covers all the features offered by this microcontroller in order to accelerate the development process.

The evaluation board is well priced at $15 which matches the budget for development and also providing great performance for the price. The temperature range for this board is also very high which proves that it is built for industrial environment. The on-board processor is an ARM Cortex M0+ which has a high range of optimized Thumb2 instruction set which greatly improves the throughput of the system while keeping the development process easy and fast. The low power capabilities of this board are suitable for environment in which servicing is difficult. The board outputs PWM signals to signal the control station of the flow and other quantities. All the test requirements outlined by Sierra Instruments were performed and the board seem to have passed all the requirements.

Although this product is far from development, the current firmware along with more hardware enhancements and software optimizations will make this product ideal to drive the flow meter in the industry, concluded from the justifications throughout this report.

After careful speculation of the requirements, the KL25Z128M4 is given a **GO** for the application of Vortex Flowmeter.

2. Problem Statement and Objectives

To design an Embedded Systems as per the requirements, presented by Sierra Instrumentation, for Vortex Meter using Cortex-M0+ based microcontroller KL25Z. In this project, the evaluation of FRDM KL25Z board is performed for the requirements presented by Sierra Instrumentation.

3. Approach and Methodology for Evaluation

Below is the block diagram for the proposed system design.

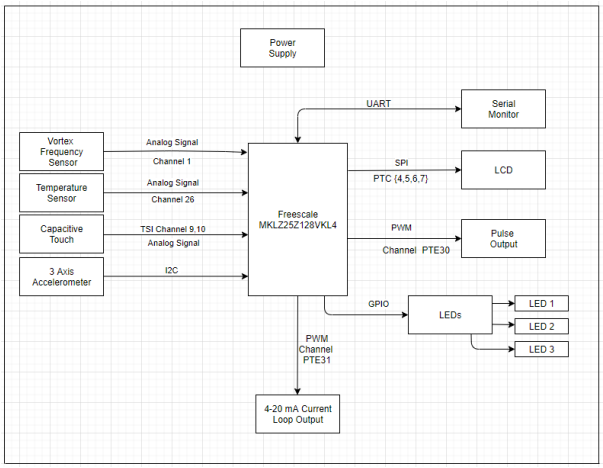


Figure 1 Block diagram for the proposed system

3.1 Hardware Evaluation – System Inputs

The proposed solution required 3 kinds of analog inputs to calculate the flow rate for the system.

3.1.1 Temperature Sensor

In this design, on-chip temperature sensor (AN3031) is used to determine the temperature reading. It uses a 16-bit ADC (channel 26), which gives the reference voltage that can be converted into the temperature reading (in Celsius) using below equations.

$$v_{temp} = \text{ADC\_Reading(channel\_2)} * 0.00005035;$$

$$\text{if}(v_{temp}) \geq 0.716? T = \left(25 - \frac{v_{temp} - 0.716}{0.001646}\right) : T = \left(25 - \frac{v_{temp} - 0.716}{0.001749}\right)$$

3.1.2 Touch Sensor

The touch sensor interface capability is provided by the kinetics lines of chips by NXP. The pins on the chip acts like a capacitive touch sensor, and they the processor can get the readings out of them using PSI registers directly. The TSI sensors are highly configurable and provide a wide range of touch sensing capabilities. The TSI capabilities of the pin can be multiplexed with other GPIO capabilities which means that the pins do not just act like a TSI sensor. Two Touch Sense Input (TSI) signals, TSI0_CH9 and TSI0_CH10, are connected to capacitive electrodes

configured as a touch slider. Freescale's Touch Sense Software (TSS) provides a software library for implementing the capacitive touch slider.

3.1.3 Vortex Frequency Sensor

A 16-bit ADC is present on KL25Z128M4, that gives up to 12-bit mode operation with the conversion rate of 20 KSPS. Using the timer interrupts in KDS, the conversion rate can be scaled down to 10 KSPS by using a delay function.

3.1.4 3-axis Accelerometer

A Freescale MMA8451Q low-power, 3-axis accelerometer is interfaced through I^2C bus and two GPIO signals. The default I^2C address is 0x1D.

3.2 Hardware Evaluation – System Outputs

3.2.1 Serial Monitor – UART

KL25Z128M4 has 2 standard and one low-power UART modules to be configured as serial port and display data on a monitor.

3.2.2 LEDs

KL25Z128M4 has 3 PWM-capable signals connected on PTB18 (Red Cathode), PTB19 (Green Cathode), and PTD1 (Blue Cathode).

3.2.3 LCD

A SPI LCD is interfaced with the KL25Z128M4 in order to display the flow rate.

3.2.4 Pulse Output for totalizer

KL25Z128M4 has two 2-channel timers/PWM modules, edge-aligned PWM mode and motor control functions which satisfies the requirement.

3.2.5 4-20 mA Current Loop

PWM helps to implement a current-loop transmitter. One 6-channel PWM and two 2-channel PWM modules are provided in the board that can be configured to meet the given requirements. KL25Z128M4 has a 12-bit DAC support which can be configured to implement a 4-20 current loop.

4. Module Test Results

Here are the test results for the various modules of test the functionalities of Vortex Flowmeter. The description and the output for these modules is listed as below:

4.1 Module 1: Finding Square root of an Integer

In this section, bisection method was used to determine the square root of a 32-bit integer. The program needs to be written in Assembly equivalent of a C code. There is a call to this assembly function from the main C program Usually this mixing of assembly and C code happens to make the execution faster. This method is in such a way that is gives an integer output for numbers which perfect square root doesn't exists. For example, integer 9 has a perfect square root which is 3 but for integer 11 it will return 3 which truncated equivalent of square root of 11. Below is the screenshot for finding square root of integer 121 (0x79).

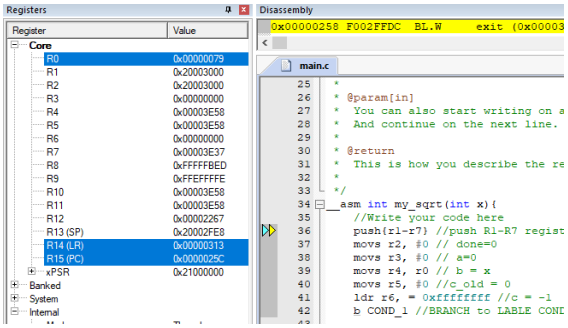


Figure 2 Input to the function is 121 or 0x79

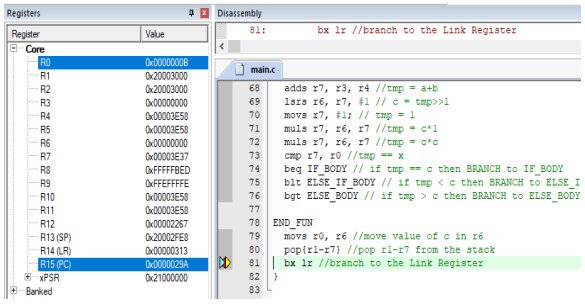


Figure 3 Output of the function is given as 11 or (0x0B)

4.2 Module 2: Feel the vibrations

In this module, the on-chip accelerometer (MMA) and touch sensors are configured to work with KL25Z128M4. Based on the value of x, y, and z axis component of accelerometer reading the color of LED changed. The brightness of the LED is controlled by the capacitive touch sensor.

4.3 Module 3: Serial Port Debug Monitor

In this module, a monitor is setup to display the readings of flowrate, temperature and frequency. TeraTerm was used to display the data which was being passed using UART Serial port. 4 modes namely Normal, Quiet, Debug and Quit were setup to display

relevant information about the execution of the program.

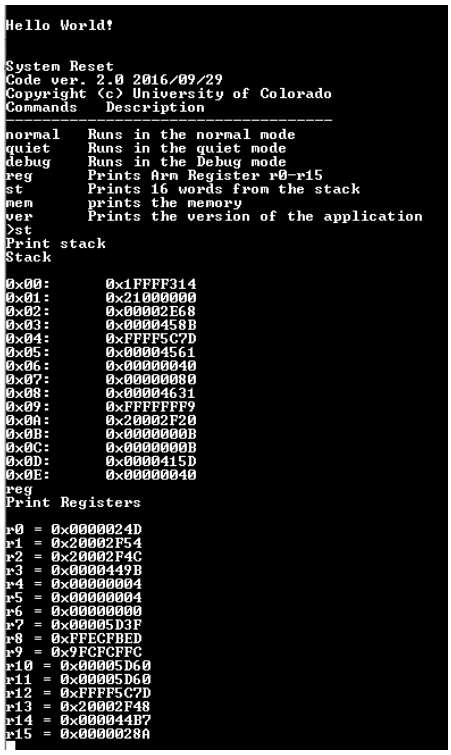


Figure 4 UART Debug Monitor

4.4 Module 4: Bare Metal Flowmeter Simulation

In this module, various peripherals like ADC, SPI and PWM is interfaced with the KL25Z128M4 to sense the data required to calculate the flowrate. An algorithm is developed and simulated to determine the frequency of the flow. An LCD is also interfaced with microcontroller to display the flowrate measurement. A PWM signal and a 4-20mA current loop are also generated as an output of the system.

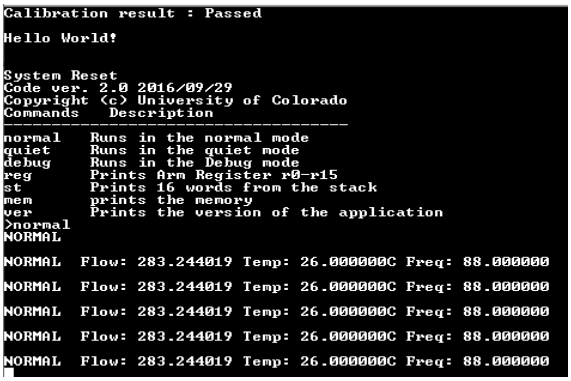


Figure 5 Flowrate, frequency and Temperature displayed on UART monitor display

The power requirements () of this implementation are in the accepted range (less than 100mW).

The implementation of this product with the mentioned components and requirements will cost approximately $169.77 which is well within the budget provided ($200). Please refer to the **BOM.xlsx** file for individual component pricing.

5. List of Project Deliverables

Module wise code files, simulation, block diagram and test results are compressed to ZIP file and attached with this report. Doxygen report files are also provided with the respective module directory.

6. Recommendations

The required parameters while designing the Vortex Flowmeter for Sierra Instrumentation are Speed, Cost and power efficiency. KL25Z128M4 performs well on all the parameters. After the evaluation and testing, it can be concluded that the KL25Z128M4 can be used as a microcontroller for Sierra Vortex Flowmeter. Peripheral programming, open-source software like mbed compiler and freeware Keil uvision are available and a lot of documentation is available for the same, debugging on KL25Z128M4 is quite easy. This makes the development of this product feasible in time and budget. Hence, the recommendation of the use of KL25Z128M4 in the Sierra Instrumentation Flowmeter product has been given a **GO**.

7. Appendix: References

[1] Requestforserives.pdf
[2] FRDM-KL25Z User's Manual (Rev 2)
[3] FRDM-KL25Z Pinouts (Rev 1.0)
[4] KL25P80M48SF0datasheet
[5] ARM mbed Tutorials

[6] 4_20currentLoop_dms-an20.pdf
[7] AN3031, Temperature Sensor for the S08 Microcontroller Family - Application Notes (nxp.com)

8. Appendix: Frequently asked questions

8.1 Module 1

8.1.1 Test your code with these inputs: 2, 4, 22, and 121. Record the results.

Answer: The output for the above inputs will be 1, 2, 4, 11 respectively.

8.1.2 Estimate the number of CPU cycles used for this calculation.

Answer: 56 CPU cycles (for input number 5).

8.1.3 Auto-generate documentation using doxygen.

Answer: The doxygen generated files are present in the Module-1/docs directory in the attached zip file.

8.2 Module 2

8.2.1 Estimate the processor load in % of CPU cycles.

Answer: 62.9%

8.2.2 Auto-generate documentation using Doxygen. Provide either an HTML directory or PDF file documenting your codebase.

Answer: The doxygen generated files are present in the Module-2/docs directory in the attached zip file.

8.2.3 Make a short (less than 1 minute) video showing your board working with the accelerometer input.

Answer: Test results can be found under Module-2/test results directory.

8.3 Module 3

8.3.1 What is the count shown in timer0 if you let it run for 30 seconds? Explain why it is this.
Answer: timer0_count = 3786

8.3.2 How much time does the code spend in the main loop versus in Interrupt Service Routines?

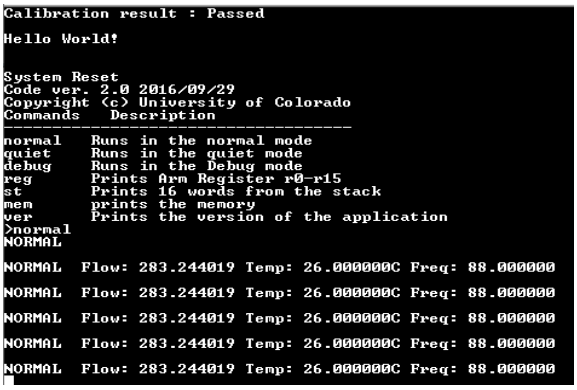
Answer: main loop: 50 micro sec, ISR: 14 micro sec

8.3.3 Test each of the commands in the Debug Monitor and record the results. Explain anything you see that you did not expect. Are you able to display all the registers?

Answer: Each command is expected. Yes, all the GPIO registers are being displayed.

8.3.4 What is the new command you added to the debug menu, and what does it do? Capture a screenshot of the new monitor window.

Answer: Added **reg**, **st** and **mem** commands in the UART Debug monitor. Here is the screenshot for the same.



8.3.5 A GPIO pin is driven high at the beginning of the Timer ISR, and low at the end. What purpose could this serve?

Answer: The idea behind doing that it to time the ISR, hook up that pin to the DSO and check for High and Low. The time between high and low will be equivalent to execution of instructions between them and hence we can capture this time on the DSO to calculate how much time did the interrupt take.

8.3.6 Estimate the % of CPU cycles used for the main background process, assuming a 100 milli-second operating cycle.

Answer: 52%

8.3.7 What is your DMIPS estimate for the MKL25Z128VLK4 MCU?

Answer: 58 DMIPS

8.4 Module 4

8.4.1 Write a frequency detection algorithm to determine the vortex frequency from the raw ADC quasi-sine wave samples. Also create code to determine the volumetric flow from the frequency. Once you have completed the code to calculate the volumetric flow, request the simulated ADC data file from your instructor. Each project team data file will be different.

Answer: Capture a reference value, and check for the transitions across this point in the sample data. Record the time difference between these transitions and add this time difference. Capture the number of times this transition occurs, to

calculate the frequency divide the total number of transitions by total time.

8.4.2 Record the reported values of frequency and flow from your monitor program.

Answer:

```
Calibration result : Passed

Hello World!


System Reset
Code ver. 2.0 2016/09/29
Copyright (c) University of Colorado
Commands    Description
-----------------------------------
normal    Runs in the normal mode
quiet     Runs in the quiet mode
debug     Runs in the Debug mode
reg       Prints Arm Register r0-r15
st        Prints 16 words from the stack
mem       prints the memory
ver       Prints the version of the application
>normal
NORMAL

NORMAL  Flow: 283.244019 Temp: 26.000000C Freq: 88.000000

NORMAL  Flow: 283.244019 Temp: 26.000000C Freq: 88.000000

NORMAL  Flow: 283.244019 Temp: 26.000000C Freq: 88.000000

NORMAL  Flow: 283.244019 Temp: 26.000000C Freq: 88.000000

NORMAL  Flow: 283.244019 Temp: 26.000000C Freq: 88.000000
```

8.4.3 Estimate the % of CPU cycles used for this process, assuming a 100 milli-second operating cycle.

Answer: 73%

8.4.4 Calculate the power consumption for your complete system (including proposed hardware additions) when in full run mode, and again in low power mode. Include detailed timing assumptions used in each mode.

Answer: The power consumed during normal mode is 19.20*5 = 96mW

8.4.5 Auto-generate documentation using Doxygen. Provide either an HTML directory or PDF file documenting your codebase.

Answer: The doxygen generated files are present in the Module-4/docs directory in the attached zip file.

9. Appendix: Project Staffing

Ayush Gupta
Graduate Student
University of Colorado Boulder
Email: aygu7370@colorado.edu
Phone: +1 (721)-761-3028

Sankalp Agrawal
Graduate Student
University of Colorado Boulder
Email: saag2511@colorado.edu
Phone: +1 (721)-761-3030