Table of Contents

1) Executive Summary

The scope of this document concerns the analysis done for Sierra's Model 240V In-line Volumetric Vortex Flowmeter .This report evaluates the FRDM MKL25Z on various criteria such as hardware requirements, software requirements, budget and other power requirements. The evaluation of FRDM KL25Z was carried out using a few inputs: 1000 samples of simulated flow data to find the frequency of flow, a temperature sensor was used in conjunction with the frequency data to find the overall Flow. The flow and frequency data was then output via PWM to a pulse output and a 4-20 mA output. The data was sent to a Serial monitor as well. An LCD interface was also requested for data display.

The approach taken was to develop a monitor program with UART usage. Then ADC was used to get temperature data and also noise to add to the sample inputs. This sample data is then run through an algorithm that uses histerises. The algorithm tests whether the last input was positive or negative and whether the input is a certain threshold value above or below the center of the PWM. It then counts the "zeros" and divides out the time sampled multiplied by two. The frequency is then taken and used with the temperature to calculate flow. The flow and frequency are then output to the various modes of expression. The frequency found from the test data was 400 Hz and the Flow was found to be 12911 GPM.

The big problem with all of this was the storage of the code in the limited code space. The limited code space caused major problems. The inclusion of the sample data took up a whole 8 kb of code data and thus made everything else sacrifice. The code barely fits ____. The project code going into the code space is too much. The individual code segments work though with the added simulation samples the code is too large for the KL25z. The final recommendation for this project is NoGO due to the future possibilities for the project. The KL25z just doesn't have the storage for the code for this device.

2) Problem Statement and Objectives

Evaluate the hardware and software capabilities of the Freescale Kinetis K25LZ MCU by using the FRDM-K25LV platform to perform a series of tests and also by design calculations.

Draw a system block diagram showing the inputs, processor, and outputs for this product.

Create an algorithmic block diagram in Simulink and simulate your system. Show simulation results in the report using simulated ADC data provided to you.

Evaluate system performance by completion of the 4 assigned modules. See the Project 1 Guide for details. If a better processor choice is available, please recommend it.

Evaluate hardware suitability by suggesting on paper possible I/O interfaces including ADC input, +24 volt Pulse Out drivers, 4/20 circuit driver, LED drivers, LCD interface and calculating required I/O current drive and voltage levels for the processor I/O.Create a BOM of major parts required in a circuit solution.

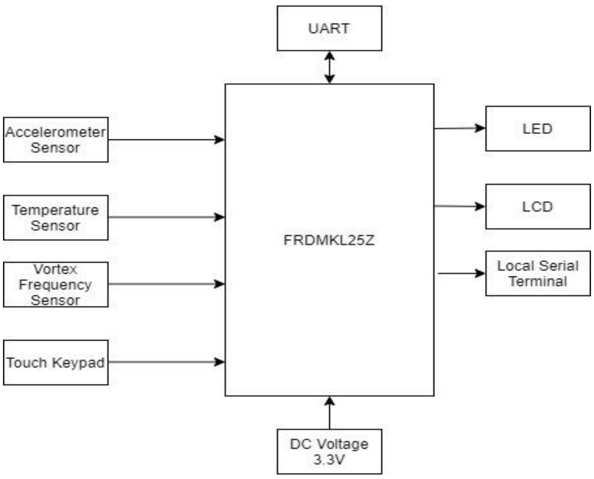
Measure simulated flow using your prototype. You will be given captured data to use in the flow calculations. Estimate power consumption for your system as this occurs.

Produce a Technical Report showing results of the proof of concept prototype.

Provide a conclusion regarding the viability of the concept and recommendations for further development.

3) Approach and methodology of evaluation

a) Block diagram or Semi-Schematic diagram



b) Hardware evaluation :

The hardware evaluation can be done based on whether the required hardware was available for the successful implementation of vortex flowmeter .
Based on the requirements the Freedom board was able to satisfy all the requirements for input and output for implementation of vortex flowmeter.

INPUT REQUIREMENTS:

i) Temperature Sensor : Freedom board KL25Z has an on board ADC which has a temperature sensor, whose output is connected to one of the ADC channels. We were able to use this Temperature sensor through ADC as an input for our flow meter.

ii) Vortex frequency Sensor: Freedom board comes with ADC which can be used along with timers to produce a frequency.

iii) Touch Keypad: Freedom board already consists of a capacitive touch that can be configured and used as a touch keypad for our project.

iv) Accelerometer: Freedom board contains a MMA8451Q low-power, three-axis accelerometer that is interfaced through an I2C bus and two GPIO signals.

OUTPUT REQUIREMENTS:-

i) LEDs :- The KL25Z board has RGB LEDS that can be configured through the PWM channels and used .

ii) Local Serial Terminal:- The KL25Z supports UART and can be configured to output to serial monitor.

iii)4-20 current loop (PWM or DAC output): The freedom board consist of 6 PWM channels that can be configured to serve our requirement. The board also consists of an onboard DAC that can be configured.

iv) Average Power Consumption:- The average power consumption for the board is below 100W and can be used for our project.

c) Additional evaluations

Further evaluation is done to check the processor's capability .A benchmark test was done and a result of 4.6 DIMS was obtained which satisfies the requirements.

Components	Price
FRDM KL25Z	$16.96
Temperature sensor	$1.25
RED LED	$0.24
AMC1602AR-B-B6WTDW-SPI	$12.69
4-20 circuit driver	$1
Touch keypad MPR121	$11.09
Touch keypad driver	$7.95
connectors	$60

Further the bill of material satisfies the cost requirements it will cost $111.18 which is in budget of $200.

4) Module test results

MODULE 1 :-

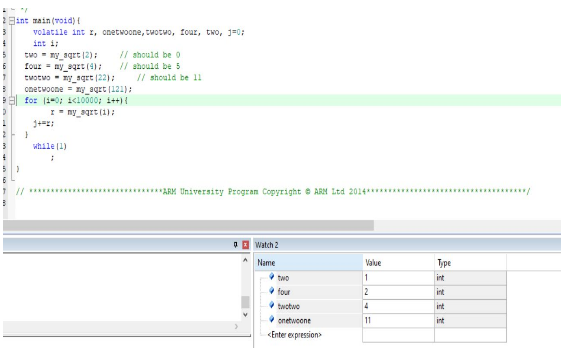


Fig 4.1: Test code for inputs 2 , 4, 22, 121

<u>Estimate the number of CPU cycles used for this calculation:</u>

Number	Cycles
2	125
4	106
121	128
22	145

MODULE 2

Estimate the processor load in % of CPU cycles :
The CPU cycles required for the entire program is 2680 cycles and it takes 1876 cycles in main so the processor load in percentage is 70%

Video:
https://drive.google.com/file/d/1VbzSK4zqoN0u97-U43fjUWkCiAPhuOi8/view?usp=sharing

MODULE 3

1. What is the count shown in timer0 if you let it run for 30 seconds? Explain why it is this.

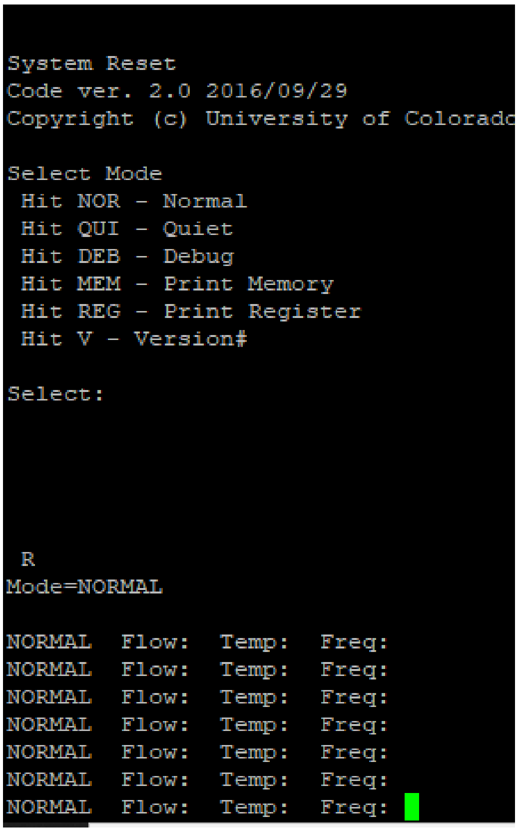
-> The count in timer0 after 30 seconds is 0x93E0. This is because the timer0 loops every 6.5ish seconds because that is the tolerance for 16 bits. The System timer at 30 seconds equals 0x493E0 which is equivalent to 300,000 timer interrupts which happen at 100us intervals so this is 10000 per second. The timer0 resets as it reaches its limits and at this same time the System timer's 17th bit gets incremented. Thus timer0 counts in intervals of 6.5ish seconds and this leads to the value 0x93E0.

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

-> The time spent in the Main loop versus the interrupt was calculated by taking the time at the start of the while loop and at then end and adding up the time spent in the while loop over time the ratio of system time to main loop time is 0x55C3 to 0x8213F or 21955 to 532733 meaning 1:24.26.

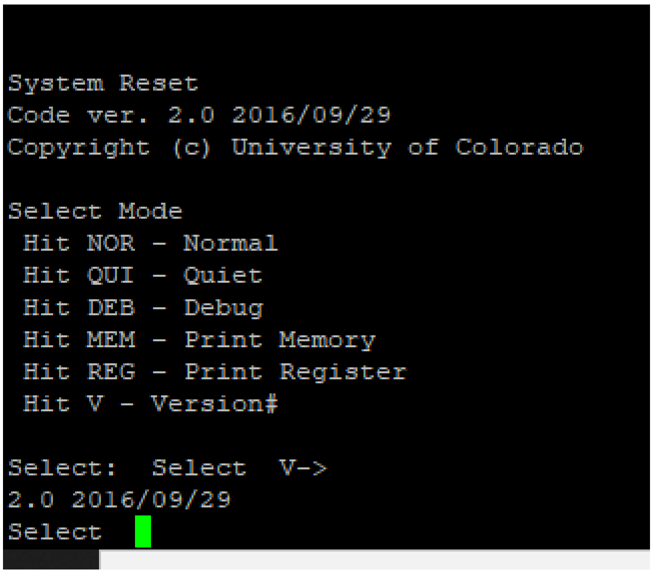
3. Test each of the commands in the Debug Monitor and record the results. Explain anything you see that you did not expect. Are you able to display all the registers?

We saw everything that we expected and all registers were correctly displayed.



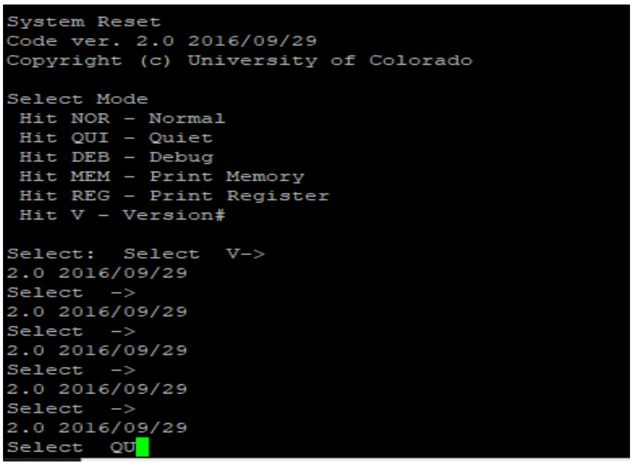
```
System Reset
Code ver. 2.0 2016/09/29
Copyright (c) University of Colorado

Select Mode
 Hit NOR - Normal
 Hit QUI - Quiet
 Hit DEB - Debug
 Hit MEM - Print Memory
 Hit REG - Print Register
 Hit V - Version#

Select:

 R
Mode=NORMAL

NORMAL  Flow:  Temp:  Freq:
NORMAL  Flow:  Temp:  Freq:
NORMAL  Flow:  Temp:  Freq:
NORMAL  Flow:  Temp:  Freq:
NORMAL  Flow:  Temp:  Freq:
NORMAL  Flow:  Temp:  Freq:
NORMAL  Flow:  Temp:  Freq:
```

Fig 4.2: NOR: Normal mode



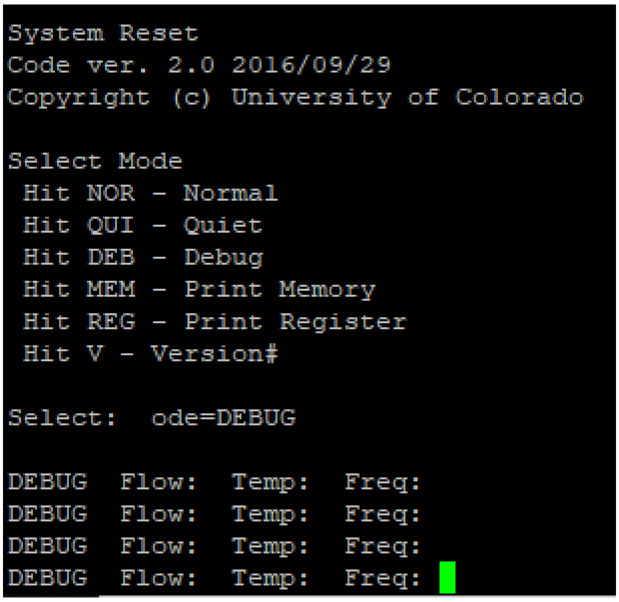
```
System Reset
Code ver. 2.0 2016/09/29
Copyright (c) University of Colorado

Select Mode
 Hit NOR - Normal
 Hit QUI - Quiet
 Hit DEB - Debug
 Hit MEM - Print Memory
 Hit REG - Print Register
 Hit V - Version#

Select:  Select  V->
2.0 2016/09/29
Select
```

Fig 4.3: V: Version mode displays version



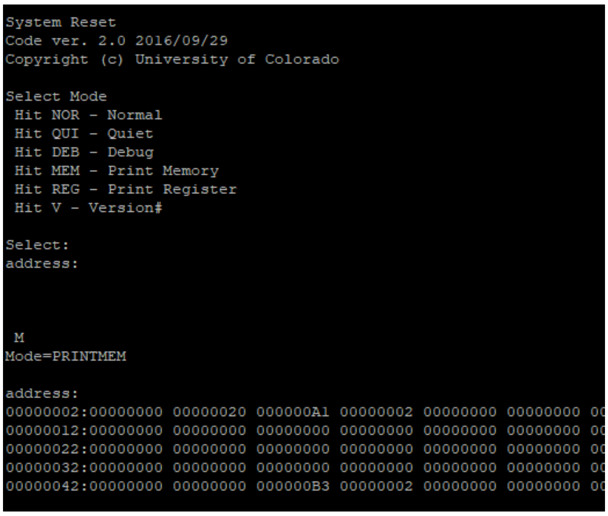
```
System Reset
Code ver. 2.0 2016/09/29
Copyright (c) University of Colorado

Select Mode
 Hit NOR - Normal
 Hit QUI - Quiet
 Hit DEB - Debug
 Hit MEM - Print Memory
 Hit REG - Print Register
 Hit V - Version#

Select:  Select  V->
2.0 2016/09/29
Select  ->
2.0 2016/09/29
Select  ->
2.0 2016/09/29
Select  ->
2.0 2016/09/29
Select  ->
2.0 2016/09/29
Select  QU
```

Fig 4.4: QUI: Quiet mode



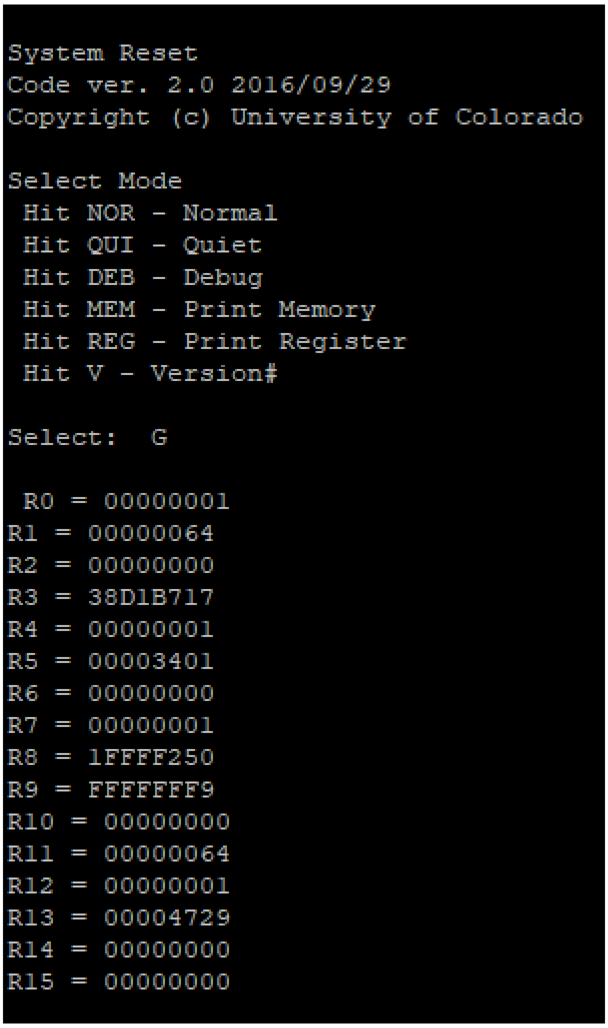
```
System Reset
Code ver. 2.0 2016/09/29
Copyright (c) University of Colorado

Select Mode
 Hit NOR - Normal
 Hit QUI - Quiet
 Hit DEB - Debug
 Hit MEM - Print Memory
 Hit REG - Print Register
 Hit V - Version#

Select:  ode=DEBUG

DEBUG  Flow:  Temp:  Freq:
DEBUG  Flow:  Temp:  Freq:
DEBUG  Flow:  Temp:  Freq:
DEBUG  Flow:  Temp:  Freq:
```

Fig 4.5: DEB: DEBUG mode



```
System Reset
Code ver. 2.0 2016/09/29
Copyright (c) University of Colorado

Select Mode
 Hit NOR - Normal
 Hit QUI - Quiet
 Hit DEB - Debug
 Hit MEM - Print Memory
 Hit REG - Print Register
 Hit V - Version#

Select:
address:


 M
Mode=PRINTMEM

address:
00000002:00000000 00000020 000000A1 00000002 00000000 00000000 00
00000012:00000000 00000000 00000000 00000000 00000000 00000000 00
00000022:00000000 00000000 00000000 00000000 00000000 00000000 00
00000032:00000000 00000000 00000000 00000000 00000000 00000000 00
00000042:00000000 00000000 000000B3 00000002 00000000 00000000 00
```

Fig 4.4: MEM: Print memory



```
System Reset
Code ver. 2.0 2016/09/29
Copyright (c) University of Colorado

Select Mode
 Hit NOR - Normal
 Hit QUI - Quiet
 Hit DEB - Debug
 Hit MEM - Print Memory
 Hit REG - Print Register
 Hit V - Version#

Select:  G

 R0 = 00000001
R1 = 00000064
R2 = 00000000
R3 = 38D1B717
R4 = 00000001
R5 = 00003401
R6 = 00000000
R7 = 00000001
R8 = 1FFFF250
R9 = FFFFFFF9
R10 = 00000000
R11 = 00000064
R12 = 00000001
R13 = 00004729
R14 = 00000000
R15 = 00000000
```

Fig 4.4: REG: Register read mode

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.

-> The new command added was MEM :- To print a memory section and REG : to print the register values from R0-R15.

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

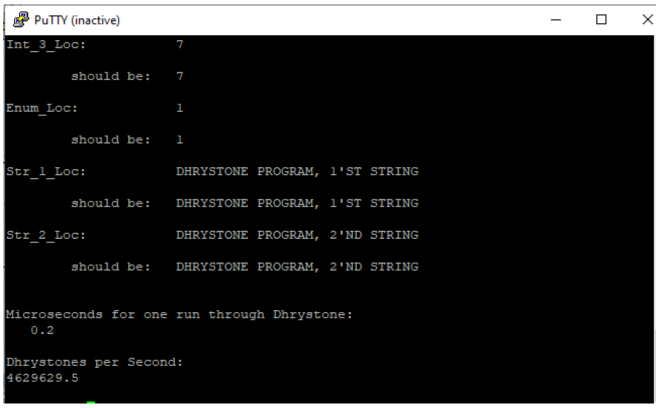
-> This can be used as a visual for when the ISR is active. This could also be used sometimes with an energy profiler to indicate ISR and differentiate it from other energy readings.

6. Estimate the % of CPU cycles used for the main background process, assuming a 100 millisecond operating cycle.

->The CPU cycles for the total cycles for the process is 3,761, and main takes 2,256 cycles so it takes 60% CPU cycles in main.

7. What is your DMIPS estimate for the MKL25Z128VLK4 MCU?

-> DMIPS = 4.6296295



```
PuTTY (inactive)
Int_3_Loc:          7
        should be:  7
Enum_Loc:           1
        should be:  1
Str_1_Loc:          DHRYSTONE PROGRAM, 1'ST STRING
        should be:  DHRYSTONE PROGRAM, 1'ST STRING
Str_2_Loc:          DHRYSTONE PROGRAM, 2'ND STRING
        should be:  DHRYSTONE PROGRAM, 2'ND STRING

Microseconds for one run through Dhrystone:
   0.2

Dhrystones per Second:
4629629.5
```

Fig 4.5: Drystone

MODULE 4

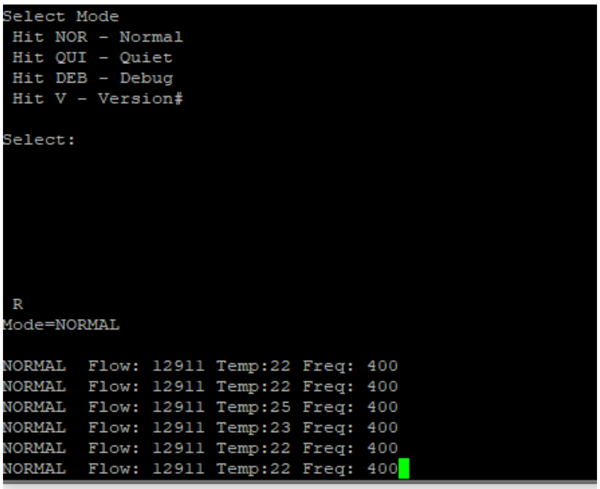


Fig 4.6: Normal mode

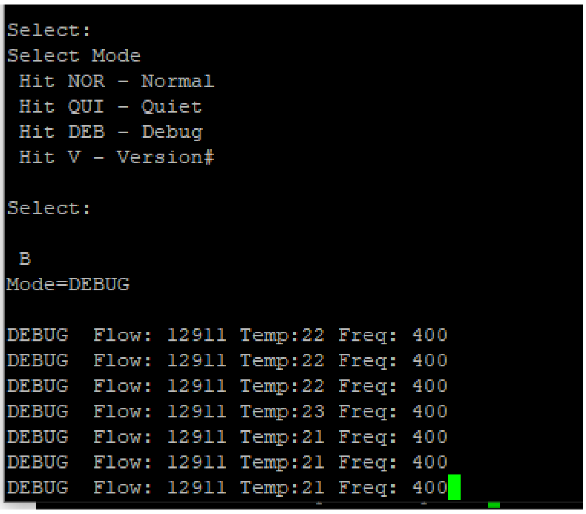


Fig 4.4: DEBUG mode

1. What is the frequency estimate from your provided sample ADC data?
->The frequency values we got were 400 Hz

2. What is the calculated flow you see from your input?
->The flow values we got were around 12911 GPM

3. What is the range of temperatures you measured with your embedded system?
->We measured temperatures around 25 degrees Celcius we did not see a major variation.(22-26)

4. How much time does the code spend in the main loop versus in Interrupt Service Routines?
-> The CPU spends 300036 usec in main whereas 1214 useconds in interrupt service.

5. Estimate the % of CPU cycles used for the main foreground process, assuming a 100 millisecond operating cycle.
-> It takes 300,360 cycles for the entire process and 2,988,220 cycles in the foreground process so it takes 99% CPU cycles in main.

6. Calculate the power consumption for your complete system (including proposed hardware additions) when in full run mode, and again in low power mode.
-> Full power is estimated at 873mW and low power at 2.5 mW
396 mW board
2.5 mW in sleep
4-20 circuit takes up 330mW
LEDs approx 140 mW
7.5mW for LCD

5) Recommendations: GO/NoGO :

The FRDM board does satisfy the hardware requirements , is in budget and also satisfies the power consumption criteria . But if we consider the software requirements and other considerations for memory then this product recommendation is a NoGO due to the limited code space. Any future updates to this product would be very cumbersome. Due to the limited code space there is a limited amount that you can accomplish. Many items had to be cut or limited in the current project design to make it fit into the required space.

6) References

1. FRDM-KL25Z User's Manual (Rev 2)
2. FRDM-KL25Z Pinouts (Rev 1.0)
3. KL25P80M48SF0datasheet
4. Project 1 Guide
5. Request for Services
6. Getting Started with ARM using mbed
7. Freescale Kinetis KL25Z Datasheet

7) Project Staffing

1. **Maitreyee Rao**
   Graduate Student
   University of Colorado Boulder
   maitreyee.rao@colorado.edu

2. **Kanin Mcguire**
   Undergraduate Student
   University of Colorado Boulder
   kanin.mcguire@colorado.edu