16 Nov 2021

PES FINAL PROJECT PROPOSAL

AIM:

To explore FreeRTOS on FRDM-KL25Z.

BACKGROUND:

FreeRTOS is a Real time Operating System that allows us to schedule various tasks such that they can be executed within the given deadline. It also supports interrupts and provides features like mutexes and semaphores that allow resources to be shared between multiple tasks. It is popular among embedded applications.

PROJECT FUNCTIONALITY:

The functionality of the project is to read inclination value from a MMA8451Q and display the read X, Y and Z values on a 16x2 LCD. To achieve these features, two tasks will be defined in FreeRTOS, the first task would read the MMA8451Q accelerometer value, and the second task would display this value on a 16x2 LCD, the tasks will be scheduled using rate monotonic scheduling method which is basically priority-based preemption and I intend on giving the first task higher priority.

IMPLEMENTATION PLAN:

First step is to implement the MMA8451Q module individually. As part of this, the module would essentially have three major API's,

- Module Initialization
- Module calibration
- Data read

Second step is to implement the LCD individually, for this there would be a fair bit of abstraction since the module would be running on 8-bit MCU mode which makes use of GPIO pins to communicate between FRDM-KL25Z and the LCD display. This implies the LCD command codes and conversion of characters to the required data format must all be abstracted away from the user.

Once the above two modules are fully functional, they need to be integrated to display the accelerometer values on the LCD without the use of FreeRTOS. At this stage, I can be confident that the modules are up and running and all of what is left is to schedule them i.e., move from bare metal to FreeRTOS.

To start with FreeRTOS, firstly two very simple functions like blinking two LEDs at different time periods can be implemented this would help in getting comfortable with the syntaxes and function APIs.

Next, the LED tasks need to be replaced with the accelerometer task and LCD task. The functionality would be implemented such that the accelerometer values would be read every 500ms and the LCD would display these values every 600ms.

The technologies used in this project are

- FreeRTOS
- I2C
- GPIO
- SysTick (For testing)

RESOURCES:

To complete this project,

- I need to learn how to schedule tasks using FreeRTOS
- I need to interface LCD and MMA8451Q with KL25Z

The following are the links I found helpful.

https://mcuoneclipse.com/2012/09/29/tutorial-freedom-with-freertos-and-kinetis-l/

https://learningmicro.wordpress.com/interfacing-lcd-with-kl25z-freedom-board/

https://mcuoneclipse.com/2012/07/20/freertos-with-gcc-cortex-m0-and-kinetis-kl25z-freedom-board/

For accelerometer, I intend to use Alexander Dean’s code as reference.

HARDWARE:

This project requires one additional hardware component that is, LCD display and along with this I might require a potentiometer to adjust its contrast. I already have these components with me. I might need to test the LCD display before starting with the project. If it does not work, I would order one from the following link.

16 Nov 2021

https://www.sparkfun.com/products/255

TEST PLAN:

After every stage of the Implementation plan, the modules need to be manually tested. For the FreeRTOS stage alone, automated test suit would be written.

For the accelerometer, it first needs to be calibrated and then the measured readings need to be verified by visual inspection or with a mobile app.

Visual verification is required for LCD as well, where the output on the LCD can be verified with the actual characters that were sent to it.

After implementing on FreeRTOS, I need to verify if the tasks are being scheduled at the required period every time. For this, I can implement systick timer and check the time elapsed between each call to the task. To achieve this, I can use automated test suite that would verify if the time elapsed is the same every time the task is the scheduled.

ADDITIONAL FEATURES:

If time permits, I would be implementing the following features:

- Read the accelerometer values when an interrupt is triggered.
- Display both Instantaneous and Average position values on the LCD.
- Explore uCUnit test framework for unit testing

MILESTONES:

Week	Particulars
Week 1 (11/16/2021 – 11/23/2021)	Implementing MMA8451Q and its testing
Week 2 (11/24/2021 – 11/30/2021)	Implementing LCD and integrating with MMA8451Q
Week 3 (12/1/2021 – 12/7/2021)	FreeRTOS basic functionality implementation and integrating the above modules
Week 4 (12/8/2021- 12/13/2021)	Function implementation and thorough testing