**PES Project 6 Readme**

**TEAM MEMBERS :** 1) SANKALP PUND (sapu9724@colorado.edu)

2) SAKET PENURKAR ([sape5303@colorado.edu](mailto:sape5303@colorado.edu))

This repository contains the files as mentioned below:

**Repo Content.:**

* .settings
* Board
* CMSIS
* Debug
* Doc
* Drivers
* Freertos
* Source
* Startup
* Utilities
* .cproject
* .project
* PES\_5813\_Project\_6\_Final\_PE Debug.launch
* PES\_5813\_Project\_6.mex
* Scope Image Program 1
* Scope Image Program 2
* Readme.md

**Observations:**

* **Problem 1**: While using an instance of a queue in other files using an extern, it was transferring garbage values from ADC buffer queue to DSP buffer. The reason was, whenever we used extern to use ADC Buffer queue instance in another file it would create new instance and it would not get the memory access leading to hard fault as well and used to store some garbage values if checked in debug mode. The same we tried solving by creating an instance in one header file and including header file wherever instance was required.
* **Problem 2**: While creating tasks, function vTaskdelay() used throw a hard fault as well as it used to execute only one task out of three other tasks. The issue was with the SysTick\_Handler() ISR of systick timer which used to conflict with xPortSysTickHandler() of freeRTOS defined in port.c the issue was resolved by removing SysTick\_Handler().
* **Problem 3**: While DMA Transfer after resolving the queue instance issue, it was again transferring the different values from ADC Buffer to DSP buffer. The issue was the data type of DSP buffer was uint32\_t and data we were storing was uint16\_t. We changed the data type of DSP buffer to uint16\_t and issue got resolved.
* **Problem 4**: While toggling Green LED during semaphore give in a DAC task, the task used to execute only four times. This is due to the fact that the task was giving away the semaphore after the LED was toggled, this caused the delay and thus it was executing only four times. To resolve this issue the LED was toggled after giving the semaphore.
* **Problem 5:** To get the timestamps there was no systick\_handler, thus we were not able to get the timestamps in log statements. Issue was resolved creating a software timer task and using its callback function to get current number of ticks.

**Installation/Execution Notes:**

* FRDM-KL25Z MCU: Connect FRDM-KL25Z to host machine through USB. For testing purpose,

mcu-expresso serial terminal is used, Program 1 and Program 2 Mode were tested at baud rate of 115200 at serial terminal.

* If you want to run Program 1 code, #define PROGRAM\_1 to (1) in main.h file otherwise change it to (0).
* If you want to run Program 2 code, #define PROGRAM\_2 to (1) in main.h file otherwise change it to (0).
* If you want to run the code in the test mode, for running the test scripts, #define TEST\_MODE (1) in the main.h otherwise change it to (0)
* If you want to enable logging, write the #define value (1) to LOG\_ENABLE macro in main.h
* For logging functionalities, please follow the respective enums in the logger.h file. For Log level setting, use the enums in Logging\_level\_m. For functions’ enum, use the enum stated in the Function\_Name\_m.