PES Project 4 Readme

TEAM MEMBERS : 1) SANKALP PUND (sapu9724@colorado.edu)
2) SAKET PENURKAR (sape5303@colorado.edu)

This repository contains the files as mentioned below:

1) Repo Content.: .settings, board, CMSIS, CMSIS Driver, debug, doc, drivers, release, source, startup, ucunit-master, Utilities, .cproject, .project, PES_5813_Project_4 PE Debug.launch, PES_5813_Project_4.mex, I2C_LOGIC_Analyser Image, Output files.

2) Problem 1: In I2C read function we had used int32_t data type variable which used to store the calculated temperature value as well as unsigned int for function return type. But use of same lead to problem while handling negative temperature, instead of (-1C) it used to show (255C). To address the same issue we used the function return type as int and int8_t as variable data type, Thus getting correct negative temperature values.

3) Problem 2: While implementing the table driven state machine, in order to reuse the code in both state machines we used same functions giving transition from different states. Event driven state machine worked just fine but the table driven machine did not. To address this issue we had to make separate functions and event enum for table driven state machine.

4) Problem 3: During Systick handler implementation for 15sec timeout functionality, we tried using multiple prescaling values in place of 16 but the maximum timeout it could possibly give was only 5 secs. To address this issue we changed the clock frequency from clock configuration from 48MHz to 10.99 MHz using prescalor value in clock tree. Thus, getting the 15sec timeout period.

5) Problem 4: While setting the flag for 15 sec timeout in Systick_handler and not clearing it again lead to issue of continuous code execution in that state. This problem is addressed by clearing the flag as soon as required state is achieved.

6) Problem 5: While handling disconnected state, it was observed that whenever sensor was disconnected from board, (I2C0->S & I2C_S_IICIF_MASK) gets cleared. And checking its corresponding flag in state machine for handling disconnected state. But it never used to transfer to disconnected state. The issue was that in single state we were checking 3 different events such as completed, alert and disconnected and storing their corresponding next state in current state variable and used to return current state at the end of function. This scenario used get fail for disconnected state since it used to store corresponding next state and check for other event as well and if my next function is of I2C read then it would get stuck in my while statement and nothing would happen after that. To address this issue, we decided that as soon as disconnected event occurs we separately return the next state then and there itself so that it won't go for next function check and disconnected state would occur.

4) Installation/execution notes:

A) FRDM-KL25Z MCU: Connect FRDM-KL25Z to host machine through USB. Go to Windows tab and select show view and then Others and select terminal. Now choose terminal as serial terminal, serial port, baud rate 115200, data size 8, parity none, stop bit 1 and encoding default which will set the terminal for output through UART. Select the project and build in debug mode. Flash the code into the target KL25Z and check the terminal to see the status outputs.

B) Slave and register address values can be changed through #define functionality in I2C.h header file.

C) 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)

D) If you want to enable logging, write the #define value (1) to LOG_ENABLE macro in main.h

E) 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 eunum stated in the Function_Name_m.