ECSE426 - Microprocessor Systems Fall 2011

Lab 2: 3D Tilt Angle Detection with Accelerometer

1 Objective

This exercise will introduce you to the iNErtial MOdule (iNEMO) development kit, Version 2, and to the sensor data processing. You should familiarize yourself with the hardware, schematics and the manual and carry out a mini-project with clear performance/precision goals.

You will be using the accelerometer part of the LSM303DLH sensor module that is connected to the processor by I^2C bus. With this accelerometer, you will implement the algorithm to calculate tilt angles presented in the Lectures.

2 The Problem

(1) Tilt angle measurement

For iNEMO board, you are required to design the system that calculates tilt angles in 3 dimensions with 2o accuracy. The system should be calculating angles in real time, at least 40 times in second, and should be resilient to the variations (noise) in the measurements above 8 Hz.

Please read the class notes and, optionally, the application notes by ST Microelectronics on how the tilt angles are measured from the projections of the gravity force. The notes also contain the explanation on how various imperfection sources (zero-*g* offset accuracy, temperature dependence etc.) get treated, but please note that the application note might not be for exactly the same device. Initial calibrations are often necessary when working with real-time sensor systems.

(2) Software library support

There is a set of functions acting as drivers for I^2C, LSM303DLH and all other peripherals, which is available with iNEMO. You can rely on it and in the tutorial, you will be shown how these drivers are to be used as well as the steps needed to adjust those drivers for your needs. As the original functions were written for IAR compiler, which is deviating at times from ARM standards, you will be provided with the initial version that works with Keil software.

You can identify the functions and the data structure for initiation of the accelerometer. Based on the specification of the problem, you need to provide the values for the fields of that data structure to select values such as: data rate, enable axes, define the scale of the accelerometer and the update mode of the device, as well as the potential filtering of the data. Please be aware of the interplay between your readouts and the data being updated, depending on the mode that might impact the performance, as well as the big/little endian selection importance for subsequent processing.

As communication to the accelerometer needs to be done via I^2C, proper initialization of I2C and sending/receiving of the packets needs to happen first, followed by the actual setting of the registers in LSM303DLH via I^2C and reading of the sampled data.

(3) Calculating and observing calculated data in real time

Since the sensor measurements change over time, you are expected to demonstrate the measurement results in real time, as per above specification, using the SWD debug interface.

3 Tasks

A. Please check with the lab technician counters for obtaining complete lab kits, consisting of one iNEMO V2 board (including the JTAG-to-SWD cable), one ULINK-ME programming attachment and two USB-to-miniUSB cables. As a preparation, attend the tutorials on Tuesday, Oct. 11th and, optionally, on Thursday, Oct. 13th for the details of the kit use with Keil software, including real-time debug features. You are encouraged to try out the demonstration software showcasing sensor collecting capabilities if iNEMO V2.

B. Try out and test the use of the modified driver file for the LSM303DLH sensor. In doing this, you should first ensure that the required I^2C communication is taking place. Since you will be given the code that has been modified from code base written for a non-standard compiler, you are encouraged to share experiences via discussion groups, including any possible code improvements. This practice is similar to the much of open-source development elsewhere and this kind of collaboration on the common code is encouraged.

C. Write a program that reads out 16-bit signed values from the accelerometer and performs trigonometric calculation needed to obtain tilt angles. Select the approach most suited for fixed-point implementation of trigonometric functions, be it lookup-table based, CORDIC or any other that you can implement. You will be judged by how efficient your program is in terms of the speed of execution of sampled data.

D. The system should be calibrated to operate on any inclined surface. You will need to write code that calibrates the accelerometers in a reference plane first. All tilt angles obtained should be relative to that reference plane. The TAs will go over basic calibration methods in the tutorials.

E. Set up the debug features such that you can demonstrate in real time the angle values obtained, as well as any other data of interest. Of course, for debugging purposes, you should be able to observe any intermediate variables obtained in the calculation. Try showcasing by built-in performance counters how well your program performs.

F. (Optional Task) An additional source of errors for accelerometer based 3-D tilt detection methods, is if acceleration occurs on the sensor itself, known as “G-forces” experienced by aircraft instrumentation. You can measure and account for these changes using the MEMS Gyroscope present on the iNEMO V2. Software drivers for DMA interfacing are included in the ST firmware files (present on the lab PCs). Correct implementation of solutions that account for changing acceleration on the module will earn bonus points.

4 Demonstration

You will need to demonstrate that your program is correct, and explain in detail how you guarantee the desired precision, and how you tested your solution. You will be expected to demonstrate a functional program and its measurements of tilt angles in 3 dimensions. The final demonstration will be on Oct.18th and Oct.20th.

5 Lab Notes

You do not need to hand in a report for this lab, but you do need to submit lab notes. These lab notes should consist of 5 sections: (i) function specifications and analysis of requirements, (ii) a high-level description (flow diagram, etc.) for software, (iii) methodology/implementation notes, (iv) performance testing method and results, and (v) discussion. Each of these sections should take approximately one page; you should be able to write the notes during lab hours (and they will be marked as though this is the case).
However, make sure you express yourself concisely and clearly, and keep the presentation professional.