**Lab 1 Graphics, LCD, ADC, Timer and Interpreter**

|  |  |
| --- | --- |
| **Goals** | • Introduction to Tiva TM4C123 LaunchPad, |
|  | • Implementing a task to display the joystick on the BoosterPack LCD |
|  | • Be able to understand the BoosterPack API |
| **Starter files** | **on Blackboard** |

**Data sheets (look for LM4F120/TM4C123 Reference material on EE319K site**

**• http://users.ece.utexas.edu/~valvano/Volume1/**

**Credited:** Valvano

# Background

The overall goal of the class will be to develop a real-time operating system. In this lab, however, you will familiarize yourself with the LaunchPad board, μVision development system and the TM4C123 ARM Cortex-M4 microcontroller. Most of the fundamental concepts in this lab should be review. Therefore, you will use this lab to explore the details of the development environment.

Look ahead to the next couple of labs. How you design this lab will simplify how you use these programs in subsequent labs. Do the lab in order. Do the preparation before coming to the first day of lab, do each step of the procedure before checking out. Read the entire lab assignment before starting the procedure, so you can gather the right data while you're doing the lab instead of at the end. Write the report the same day you finish checkout. Everything will be fresh in your mind and your lab will still be working so you can take meaningful data.

An important design step occurs in writing the header file for a driver. It is in the header file that you define the interfaces between software components. As part of the preparation in addition to the header files you will include rough pseudo code with descriptions of their approach to what you plan to write in the C files. As part of the preparation, you should have a plan of how you will complete the lab. The TA can check the preparation at the start of lab. This way the TA has an opportunity to set you on the right track by looking at what you have thought of so far.

# Preparation (do this before you do the lab)

1. Look at the documentation in the BSP.c and BSP.h files, you will be using this for the first five labs in this course. You should be able to read the comments above each function and know what they do and how to use them, that way you can use them in the labs when asked.

# Procedure (do this as part of your lab period)

1. Implement TaskC\_Init, this should initialize the joystick and change any
2. Implement TaskC to save the joystick data to global variables, you will also need to define the global variables.
3. Go into Task4 and fill in the joystick mode to have it plot the three variables, there are pre-defined colors at the top like STICKXCOLOR, STICKYCOLOR, and STICKSCOLOR but you can use any color you would like.
4. Analyze the main function, task 0 is set to run at 10 [Hz], but we want it to run at 1,000 [Hz], use a loop to change the timing to make sure that it runs at 1,000 [Hz] without changing the frequency of the other tasks and add 1 [ms] of delay after each time Task0 runs. You can find a function to delay in the BSP.c and BSP.h files

# Deliverables

1. Each lab member should submit a copy of the code to the lab 1 assignment under assignments on blackboard. Preferred submittal is in the form of a zip/rar/7z file.

# Demonstrations (To the TA)

You should be able to show the TA the lab running on the board and be prepared to answer any questions about the lab.

# Hints

1. It is appropriate to copy-paste software from the example files. You must, however, clearly document which code is copied, which code is modified, and which code is original.

1. Even though you are allowed to copy-paste software, there should be no magic in this class. In other words, you are responsible for understanding all the details of how your system runs.