ECE 1181: Microcontrollers and Embedded Systems Lab
Fall 2024 – Professor Joe Camp

Lab 1: Getting Started (8/29 or 8/30)
(Due before the start of the next lab – 9/5 or 9/6)

First and foremost, welcome to the lab for Microcontrollers and Embedded Systems! In this laboratory course, you will be designing and implementing programs written in ARM assembly language. To build, run, debug, and demonstrate each lab, you will have your own Red Pitaya, which we will be exploring in this lab.

You will have a document like this for your use in each lab. The lab description (problem description), any given code or hints, and the requirements for completion are provided in the "Description", "Code Given", and "You have completed this lab when:" fields below respectively.

Each lab report should have the following:
1. Name, SMU ID, and Lab Session Number
2. In your own words, the objective of the lab assignment and description of code you use and write. If writing an algorithm, provide pseudocode. (25%)
3. The actual code and in-line comments (25%)
4. In your own words, answers to any questions on the lab description, results, screenshots, and verification (50%)

Reports must be submitted before the start of the next lab.

Description:

Getting started with a microcontroller such as the Red Pitaya can be overwhelming. To make this as seamless as possible, there are two resources that are available to you:

1. Canvas Document entitled "Headless_RedPitaya_Configuration.pdf" – This document walks through the steps necessary to interface with the Red Pitaya in a "headless" fashion, meaning that you do not need an extra monitor and keyboard, but rather you use your laptop or desktop to log into the Red Pitaya with either SSH (command line) or VNC (graphical user interface). The document will discuss these two interfacing approaches.

2. Raspberry Pi Assembly Language Programming (1st edition) by Stuart Smith. For this lab, we will follow Chapter 1 (Getting Started) very closely. In particular, you will be typing in the Hello World Program on page 17-18, building it, running it, and slightly modifying it according to the bullets on the bottom of page 25:
    a. *Change the string but remember to change the length loaded into **R2**.*

For this first bullet, change the string from “HelloWorld!” to “Hello ” followed by your first and last name and an exclamation point. Perform a print screen and include in your report.

b. *Change the return code loaded into **R0** before the second **SVC** call and see what happens.*

For the second bullet, what did you try and what result did it have?

Code Given (WARNING: I strongly recommend typing this into your Red Pitaya directly – The reason is that there are hidden symbols that may cause a high degree of frustration from copy/paste that would be spared by just typing it in. Also, you get a better feel for the code in doing so.):

```
@
@ Assembler program to print "Hello World!"
@ to stdout.
@ R0-R2 - parameters to linux function services
@ R7 - linux function number
.global         _start                  @ Provide program starting @ address to linker
@ Set up the parameters to print hello world
@ and then call Linux to do it.
_start:  mov R0, #1                     @ 1 = StdOut
                ldr R1, =helloworld     @ string to print
                mov R2, #13             @ length of our string
                mov R7, #4              @ linux write system call
                svc 0                   @ Call linux to print
@ Set up the parameters to exit the program
@ and then call Linux to do it.
.data
                mov R0, #0              @ Use 0 return code
                mov R7, #1              @ Service command code 1
                                        @ terminates this program
                svc 0                   @ Call linux to terminate
helloworld: .ascii "Hello World!\n"
```

You have completed this lab when:

Demonstrate to your lab instructor that you have completed the steps above.