ECE 445L Lab 10

Final Design and Evaluation of an Embedded System

This laboratory assignment accompanies the book, [*Embedded Systems: Real-Time Interfacing to ARM Cortex M Microcontrollers, ISBN-13: 978-1463590154*](https://www.amazon.com/Embedded-Systems-Real-Time-Interfacing-Microcontrollers/dp/1463590156), by Jonathan W. Valvano, copyright © 2024.

# Table of Contents

[Table of Contents 1](#_Toc202863665)

[Team Size 2](#_Toc202863666)

[Goals 2](#_Toc202863667)

[Starter Files 2](#_Toc202863668)

[Required Hardware 2](#_Toc202863669)

[Lab Overview 2](#_Toc202863670)

[Preparation 2](#_Toc202863671)

[Procedure 2](#_Toc202863672)

[Lab Checkout 4](#_Toc202863673)

[Lab Report 4](#_Toc202863674)

[Deliverable 1 4](#_Toc202863675)

[Deliverable 2 4](#_Toc202863676)

[Deliverable 3 4](#_Toc202863677)

[Deliverable 4 4](#_Toc202863678)

[Deliverable 5 4](#_Toc202863679)

[Hints 5](#_Toc202863680)

[Procedure for bring up a stand-alone TM4C123, including JTAG Connection 5](#_Toc202863681)

# Team Size

The team size for this lab is **4**.

# Goals

* Build, test, and validate an embedded system.

# Starter Files

* Starter project:
  + Use the hardware and software developed in Lab 7A and Lab7B.

# Required Hardware

There is no required hardware for this lab outside of the restrictions for the final competition provided further in the document.

# Lab Overview

You will complete the embedded system you began in Labs 7A and 7B. This lab includes hardware construction, software debugging, and system evaluation. There will be a “science fair” style public demonstration at the end of Lab 10. The TAs shall judge your project during your lab demo for qualification for the final competition, and the professor(s) shall present special awards to the teams with the best designs at the end of the competition. The best designs are selected by your fellow students and observers.

# Preparation

1. **Revise your requirements document** one final time to reflect the final version of your design. Emphasize in detail how your system will be evaluated, tested, and verified.
2. **Gather any remaining parts** needed to build your system, including any makeshift repairs.
3. **Update your Bill of Materials** (BoM).
4. **Design and implement the high-level main application** that demonstrates the final objective of your embedded system. If you want to change the project scope, please consult with the TA for approval.

# Procedure

The following is a procedure for assembling and debugging your system. Feel free to adjust this to what makes sense for your circuit layout.

**1. Verify that the Lab 7A breadboard system works**. Using integration tests or other tests, understand where corrections need to be made to your PCB preemptively to get it in the same working state as your breadboard prototype.

**2. Verify that there are no shorts** between the various power and ground lines. Ensure that:

Any power signal (e.g. 3V3, 5V0, VBATT, GND) does not short to any other signal.

Shorts can occur. So, before soldering, check for shorts caused by design or manufacturing

**3. Solder down the power regulation circuitry**, including:

Power input connectors.

On-off switch.

Regulators.

Regulator capacitors and supporting passives.

*Check for shorts from power to ground. Verify that when power is connected, the corresponding output voltages are generated.*

**4. Solder down the LaunchPad headers.** Verify that when plugged in that LaunchPad turns on. See additional instructions for assembling the SMD chip of the TM4C.  
*Check for shorts from power to ground. Verify that when power is connected, the corresponding output voltages are generated.*

**5. Solder down debug components**. Solder down all test points, logic analyzer headers, and things like power supply LEDs and heartbeat LEDs.

*Check for shorts from power to ground. Verify that when power is connected, the corresponding output voltages are generated.*

**6. Solder down each subcircuit.**

Solder down components in subcircuit in order of height – lowest ones first.

*Check for shorts from power to ground. Verify that when power is connected, the corresponding output voltages are generated.*

Verify subcircuit functionality by using your unit/integration tests.

Measure current consumption for this subcircuit.

At this point, everything should have been soldered down and verified individually.

**7. Debug your embedded system application.**

# Lab Checkout

You should be able to demonstrate the full functionality of your system as described in your requirements document. Present your BoM, project cost, and features for approval for the final competition. Be prepared to discuss any difficulties and challenges during the implementation, and any testing procedures or risk mitigation processes.

# Lab Report

## Deliverable 1

Final requirements document including the data flow graph and call graph.

## Deliverable 2

Describe the software tests that you added since Lab 7 to verify your modules and their integration.

## Deliverable 3

Describe the hardware tests that you performed since Lab 7 to verify your embedded system.

## Deliverable 4

Measurements of the current for various system states as appropriate for your design. E.g., sleep, idle, receiving, transmitting, and/or active.

## Deliverable 5

Validate the performance of your embedded system. For each aspect of the system you are verifying, you must create a 1-3 sentence paragraph outlining:

* How you performed this measurement
* How/if the measured behavior meets your system’s requirements
* And what sources of errors may affect your measurement

Validate at least two performance metrics of your system from the following list:

* CPU utilization (thread profile) measured separately for each module.
  + Maximum execution time for each ISR
  + Maximum time doing work in the main loop
* Ping latency (Wi-Fi, Bluetooth, LoRa, 433Mhz Radio, etc.)
* DAC or ADC sampling jitter
* Signal to noise ratio (SNR)
* Used vs total bandwidth of IC-to-IC communication protocols (UART, SPI, I2C, CAN, etc.)
* Feel free to add a performance metric you think is relevant to your project

# Hints

1. When you pick up your PCB boards, make sure you get your board.
2. You should have collected all the electrical parts during Lab 7A. Think about your system mechanically. Here in Lab 10 collect nuts/bolts/screws. Will it all fit together? Put it together before soldering. This will help to determine the proper order of construction.
3. Take the system with you to job interviews.

## Procedure for bring up a stand-alone TM4C123, including JTAG Connection

See the **Using TM4C12x Devices Over JTAG Interface** document in the resources folder of the GH classroom repo.

1. Before you start, verify there are no shorts between +3.3V power, ground, battery input, or 5V power line. Solder the TM4C123 first. Clean the TM4C123 pads with isopropyl alcohol. Align the TM4C123 on the board. Apply solder flux to the pin. Align the TM4C123 on the board again. Gently touch the solder iron then the solder to the pin. You use VERY little solder. Solder one pin and show it to a TA/professor or Mark Innmon. Align the TM4C123 on the board. This is critical. Solder a second pin and again have someone check it. Then solder the rest of the 64 pins. Clean solder-bridges with solder-removal braid.
2. Verify there are no shorts between +3.3V power, ground, battery input, or 5V power line. Check for shorts between adjacent pins on the processor. Use thin wire (28 or 30 gauge) to touch the ohmmeter probes to processor pins. If you can see excessive solder between pins, remove it with solder wick. If adjacent pins are connected but you see no excessive solder, very gently cut between processor pins with a sharp knife.
3. Solder crystal and all caps around the processor. Make sure crystal caps are 10 pF ceramic C0G. Place tantalums in polarized direction, and double check all values. For example, ceramic 104 means 10\*104pF= 0.1uF. Verify there are no shorts between +3.3V power, ground, battery input, or 5V power line.
4. Carefully mark on your board where JTAG pin 1 is. Solder the reset button, pull up, and JTAG connector. Verify there are no shorts between +3.3V power, ground, battery input, or 5V power line.
5. Solder the 1 MΩ resistor attached to Wake, also solder your 3.3V and ground test points. Verify there are no shorts between +3.3V power, ground, battery input, or 5V power line. Before proceeding to the next step, please attach the bench power supply to the 3.3V power rails by connecting to the 3.3V and ground test points you should have. Please set the bench supply current limit to 100mA to ensure safe operation and prevent damage. Make sure to test with a multi-meter to get the polarity right to not damage the processor. You should be able to observe the 16 MHz periodic wave on both crystal pins using an oscilloscope.
6. Before soldering the 3.3V regulator, please put together and test your regulator circuit(s) to verify you have connected the power rails correctly. Now proceed to soldering the regulator by attaching the caps regulator to the PCB from your breadboard (everything you need to create 3.3V power). Include 3.3V and ground test points. Essentially, in steps 3–6, you will need to solder all the parts included in the Lab 6 starter SCH except the LCD. As part of the procedure, you will solder the rest. Verify there are no shorts between +3.3V power, ground, battery input, or 5V power line.
7. Test the power and crystal. Connect voltmeter(s) to the regulated power line(s) 3.3V and 5V if present. Use a bench supply and limit the input current to 50 mA. Apply power to your TM4C123 board (remove power immediately if not 3.3V). Record the current required to run just the processor. When powered, you should be able to see 16 MHz oscillations on both sides of the crystal using an oscilloscope.
8. Test it with the Flash programmer
   1. <http://users.ece.utexas.edu/~valvano/EE345L/Labs/Fall2011/Ifyoumessuptheboard.pdf>
   2. Download, install, but do not run the Flash programmer yet.
   3. Connect the 5-pin JTAG cable from LaunchPad to your TM4C123 board (see “How to program the board”)
   4. Attach a voltmeter to 3.3V pin on your board
   5. Both the LaunchPad and your board must be independently and separately powered, with grounds connected. Apply power your LaunchPad board and apply power to your TM4C123 board (remove power immediately if not 3.3V)
   6. Run the Flash programmer as instructed in the pdf. If this flash procedure finishes without error, then it means the TM4C123 is operational. If the flash procedure fails then there is a mistake in power, reset, crystal, or JTAG. The two most common reasons to fail are 1) the TM4C123 has no 3.3 power; and 2) the JTAG cable is backwards.
9. Download a very simple TM4C123 program that has no I/O
   1. http://users.ece.utexas.edu/%7Evalvano/arm/SimpleProject\_4C123asm.zip
   2. Connect the 5-pin JTAG cable from LaunchPad to your TM4C123 board (see “How to program the board”)
   3. Attach a voltmeter to 3.3V pin on your board
   4. Power LaunchPad board, apply power to your board (remove power immediately if not 3.3V)
   5. Download the SimpleProject\_4C123asm project to your TM4C123
   6. Run debugger
10. Solder components in the order of height. The lowest ones come first. Check for shorts from power to ground before applying power again. If your system has both 3.3V and 5V power, please place a voltmeter on the 5V output and measure the current while testing it for the first time. Record the current required to run the entire PCB.