ECE 445L Lab 7

Prototype and Low-Level Software for 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 © 2021.

# Table of Contents

[Table of Contents 1](#_Toc160492848)

[Team Size 3](#_Toc160492849)

[Goals 3](#_Toc160492850)

[Review 3](#_Toc160492851)

[Starter Files 3](#_Toc160492852)

[Required Hardware 3](#_Toc160492853)

[Teamwork evaluations 3](#_Toc160492854)

[Lab Overview 3](#_Toc160492855)

[Project Constraints 5](#_Toc160492856)

[Competition Constraints 5](#_Toc160492857)

[Optional Components 6](#_Toc160492858)

[Pre-preparation 7](#_Toc160492859)

[Requirements Document 7](#_Toc160492860)

[Preliminary BOM 7](#_Toc160492861)

[Datasheets 7](#_Toc160492862)

[KiCad Project 7](#_Toc160492863)

[Deliverables 7](#_Toc160492864)

[Preparation 8](#_Toc160492865)

[Deliverables 8](#_Toc160492866)

[Procedure 9](#_Toc160492867)

[Lab Checkout 9](#_Toc160492868)

[TA and Professor Review 9](#_Toc160492869)

[Deliverable 1&2 10](#_Toc160492870)

[Deliverable 3 10](#_Toc160492871)

[Deliverable 4 10](#_Toc160492872)

[Deliverable 5 10](#_Toc160492873)

[Deliverable 6 (5pts Extra Credit) 10](#_Toc160492874)

[Deliverable 7 (10pts Extra Credit) 10](#_Toc160492875)

[Deliverable 8 (10pts Extra Credit) 10](#_Toc160492876)

[Hints 11](#_Toc160492877)

[Other Stuff 12](#_Toc160492878)

# Team Size

The team size for Labs 7, 8 and 11 is **4**.

# Goals

* Create a proposal and requirements document for your embedded system,
* Design the embedded system at the schematic level,
* Implement low-level software for each component of the system,
* Collect components and implement a prototype of your embedded system on a breadboard,
* Develop unit testing skills to evaluate your subsystems,
* And demonstrate that the project is feasible.

# Review

* Data sheets for your microcontroller.
* Data sheets for your hardware components.

# Starter Files

* Starter project:
  + Final project sw and hw template provided in the GH Classroom repo.

# Required Hardware

We require you to collect and evaluate all hardware (chips, sensors, and actuators) needed for the final project.

# Teamwork evaluations

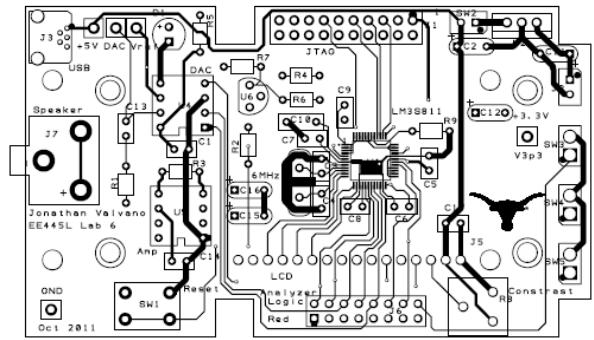
Each student will get a significant evaluation of their teamwork performance (5% of the total ECE445L grade). There should be four or more major components to the project, so we expect each member of the team to be responsible for at least one major component. The TA will show you your teamwork score after Lab 8, and you will have a second teamwork performance grade for Labs 9, 10 and 11. The grading rubric considers:

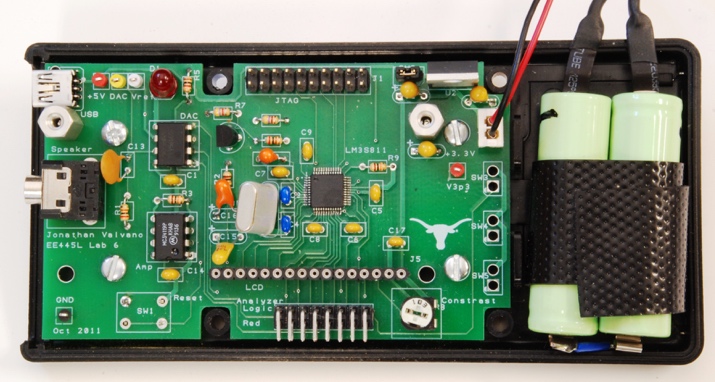
* Completion and understanding of your assigned responsibilities(s) for the project.
* The number of meaningful commits you made to your teams GitHub repo.
* The effort you contributed to the project.
* Your ability to resolve conflicts.
* Your participation in your teams’ communications. You are encouraged to show your TA the communication during Labs 7 and 8.

# Lab Overview

In Lab 7, you will begin the process of designing your own embedded system. The design of the system must satisfy certain requirements. Rather than simply redesigning one of the previous labs, this embedded system must do something new and useful. There are some ideas posted on Valvano’s [page](https://users.ece.utexas.edu/~valvano/EE445L/projectideas.htm), but you have flexibility to define exactly what it is to do. If you look at [Sparkfun.com](https://www.sparkfun.com/) you will see lots of ideas of I/O devices that you could attach to the system. The scope of the project is a microcontroller-based embedded system demonstrating the educational objectives of this class.

There are separate constraints for the project and the final design competition. You can still get a 100 on the final project without conforming to the design competition constraints, however you cannot compete officially in the competition if you do not satisfy all constraints.





Figures 7.1 and 7.2: Example mock-up and eventual Lab 11 system.

## Project Constraints

1. Each group shall produce a PCB design (.kicad\_pro .kicad\_sch and .kicad\_pcb) for the system.
2. The system should perform something useful.
3. The system shall include at least two inputs, two outputs, and two interrupt service routines.
4. The system must contain four or more identifiable major subcomponents, where each team member must be responsible for at least one major subcomponent.
5. This system shall mount the LaunchPad or TM4C123 processor, and other chips to the PCB.
6. The final system must fit in an appropriate enclosure.
7. TAs will judge if the project is sufficiently complicated.
8. Each team can order a two-layer PCB up to 30 square inches (e.g., 5 by 6 in) from JLCPCB.
9. If you use an ESP8266 from the checkout desk, do NOT solder it to the PCB.
10. Each PCB must conform to [JLCPCB's design capabilities,](https://jlcpcb.com/capabilities/Capabilities)
11. Parts that are not provided must be purchased by the group.
12. You can use two motors and their corresponding wheels from the Lab 10 supplies, but they must be returned.
13. You should NOT use a ground pour (this will introduce difficulties with debugging and fixing your board if there are any errors).

## Competition Constraints

There will be a "science fair"-like public demonstration for Lab 11. Students with the best design will be presented with special awards. The judging will be performed by the other ECE445L students by viewing demos in person.

Competition restrictions:

1. The design shall use only the TM4C123 as the primary microcontroller.
2. I/O components such as LCD displays, switches, sensors, LEDs, speakers, keypads, and microphones can be off the PCB.
3. All other electronics (resistors, capacitors, diodes, ICs, etc.) shall be on the PCB.
4. The team shall spend no more than $60 on extra components not provided for other labs.

Details on the $60 budget:

1. There are two system costs you will calculate:
   1. Cost of all components (regardless of where they were obtained)
   2. Cost of all components that factor into the $60 limitation.

The 1b) metric will be used when determining whether the project is eligible for winning the competition. For example, you happen to already own a component that is unlikely for every group to also already own; the 1b) cost will be the price for another group to purchase the same component. For example, the 1b) cost will be 0.

1. Parts that do NOT count toward the 1b) $60 limitation:
   1. Parts obtained from the ECE lab checkout counter or the [Lab7BOM.xlsx](https://github.com/ECE445L/ECE445L-Final-Lab/blob/cbc8974bb119772c4ba4bbe3bc5a70f1535af41d/resources/bom/Lab7BOM.xlsx)
      1. Passive components do not have to be returned
      2. Solid state components may have to be returned (please ask the checkout counter)
   2. Free samples
   3. Components that is likely for every group to also already own (e.g., speaker, laptop, cellphone, wire, video monitor, or mouse)
   4. The PCB (If ordered by Professor Valvano or Professor McDermott)

## Optional Components

1. ESP8266:
   1. If you desire Wi-Fi capabilities
   2. Keep in mind that the ESP8266 requires on average 80mA (can go much higher)



Figure 7.3: ESP8266 version 1

1. Audio amp:
   1. For single channel you can use an audio amp like the MC34119 or the TPA731
   2. For stereo sound, use two separate audio channels and two speakers
   3. You can order free samples from a chip vendor like [Analog Devices](https://www.analog.com/)
2. Accelerometers and other sensors:
   1. There are lots of devices available as free samples from chip vendors, but be aware that they may be very difficult to solder
   2. The easiest way add sensors is to purchase a module (chip and breakout board) from a hobby store like [SparkFun](https://www.sparkfun.com/), [Adafruit](https://www.adafruit.com/), or [Pololu](https://www.pololu.com/)
3. Enclosure:
   1. You may purchase an enclosure and count it toward your $60 budget.
   2. You may get a big and ugly enclosure from Valvano/McDermott (come check it out)
   3. You may build your enclosure separately (the enclosure will be judged on functionality and not beauty). You can use resources from [Texas InventionWorks (TIW)](https://linktr.ee/texasinventionworks) to create your box (resources used from TIW will not count toward your $60 budget). Here are some resources TIW offers:
      1. 3D printers (requires training)
      2. laser cutters (requires training)

# Pre-preparation

## Requirements Document

Update the template Requirements.docx file in the deliverables folder to reflect your design. The goal is to create a preliminary, one-page **requirements document** for the system. We expect the document to change throughout the project, so keep it up to date as you progress through the design, implementation, and testing phases.

## Preliminary BOM

Update the template BOM.xlsx file in the deliverables folder to reflect your design. List the parts that the system will use in a preliminary Bill of Materials. List the locations that you will acquire the parts from:

* ECE lab checkout desk
* Professor's cabinet
* Places like [SparkFun](https://www.sparkfun.com/), [Adafruit](https://www.adafruit.com/), or [Pololu](https://www.pololu.com/)

Determine the expected cost for the system and prepare to justify the cost to the TA.

## Datasheets

While compiling the BOM, gather the datasheets of the relevant parts. Note their communication protocols and interfaces, as well as projected current consumption. Collect these datasheets into the deliverables folder **OR** record the links to the datasheets in Datasheets.xlsx.

## KiCad Project

Create or repurpose a KiCad project in the hardware folder. As you add components to the BOM, consider if you will need to create or find footprints suitable for use.

## Deliverables

1. Initial version of the requirements document
2. List of parts that the system will use (include the locations that the parts will be acquired from)
3. Datasheets of all parts that the system will use
4. KiCad project files

You will be judged on the clarity of your project. You should be able to explain what you intend to do and how at an abstract level. This will be like a marketing requirements document (MRD) presentation to your TA. An MRD explains what a product does from a user perspective (how can we sell it).

1. Objectives: why are we doing this project? What is the purpose?
2. Interactions with Existing Systems: How will it fit in?
3. Terminology: Define terms used in the document.
4. Functionality: What will the system do precisely?
5. Usability: Describe the interfaces. Be quantitative if possible.
6. Outcomes: What are the deliverables? How do we know when it is done?

# Preparation

1. **Finalize the BoM** and collect as many of the components as you can. The bill of materials should include all components used within the system, such as: capacitors, cables, connectors, LEDs, ICs, resistors, and your enclosure.
2. **Create a system diagram** to illustrate how your hardware will interact with hypothetical low-level drivers. From this system diagram, write out the preliminary header files for all software modules that you will use.
3. **Design the schematic** (.kicad\_sch) for the system using [KiCad](https://www.kicad.org/). You must follow these rules:
   1. All components must have labels (U1, R1, C1, J1, etc) on the schematic and board.
   2. Each IC should have a bypass cap placed as close to the component as possible (look at your component's datasheet to find the recommended size of each bypass capacitor).
   3. For resistors, specify impedance (10k ohms), wattage (1/4 watt), and tolerance (5%).
   4. For capacitors specify capacitance (100uF, tolerance (20%), and material (ceramic, tantalum, electrolytic, etc).
4. If the operator needs to use the reset button, bring the reset pin out to a user-reachable negative logic switch
5. **Design the power circuit** for your embedded system.Determine if you are using a battery, and how the system can be turned on and off. You **must** use a regulator on the PCB. The typical configuration has a battery or USB plug into the PCB, and off-board power switch, and then the LM2937-3.3 regulator for the 3.3V supply.

## Deliverables

1. Bill of materials
2. System diagram and software module header files
3. Completed KiCAD schematic file that passes ERC
4. Explanation of how the system will be powered

By preparation day, you should have a very clear idea about your project. You should be able to describe the lower-level interface of the system. This will be like a product requirements document (PRD) presentation to your TA. A PRD explains how a product will be developed from an engineering perspective (how do we build it).

1. Process: How will the project be developed?
2. Roles and Responsibilities: Who will do what? Who are the clients?
3. Scope: List the phases and what will be delivered in each phase.
4. Prototypes: How will intermediate progress be demonstrated?
5. Performance: Define the measures and describe how they will be determined.

# Procedure

1. **Build the hardware** wired on a breadboard or protoboard and attach the hardware to a Launchpad. Because you are doing low-level testing, it is appropriate to use a separate protoboard and a separate Launchpad to test each module. We recommend using the I/O pins you’re planning to use in the schematic, although you should prepare to have alternative I/O pin initializations if the final layout works out better if you use a different peripheral pinout.
2. **Implement the low-level software** working with the hardware. The low-level modules need not be integrated together. The system integration will be a Lab 11 task. The goal is to perform enough low-level testing to be confident the wiring of the design is proper.
3. **Write unit tests** to confirm your subcomponents work and to profile its performance. We suggest creating projects like those found in sw/examples to test the low-level software drivers, and that the unit test tests all the functionality of your driver to make sure the driver behavior is as expected.
   1. The following is an example testing plan for a DAC used to create audio.
      1. Write the SSI/DAC software driver.
      2. Build the DAC circuit.
      3. Write interrupting software that outputs a sinewave.
      4. Verify the DAC output is a sine wave using the scope and spectrum analyzer.
      5. Build the speaker amplifier circuit.
      6. Verify the speaker outputs are sine waves using the scope and spectrum analyzer.
      7. Connect the speaker and verify the waveforms are still correct.

Note that because this is unit testing, we ask that you minimize the integration of other drivers and keep the tests as simple as possible. For example, the above testing plan would be a lot harder to debug if you’re also playing music from a music driver and not just a sine wave because there are more things to test.

1. **Verify that the unit tests meet** any performance or functionality tests as defined in the requirements document.

# Lab Checkout

For lab demo, we ask for the following:

1. Show the hardware wired on a protoboard/breadboard and attached to the Launchpad.

2. Demo low-level software working with each subcomponent of the hardware.

# TA and Professor Review

*We preemptively note that students shall submit their preliminary PCB schematics and layouts to Canvas under a known assignment name by 9 AM, Friday, October 27th for TAs and professors to review for any issues and to provide final feedback. This is a lab 8 deliverable. The final PCB schematics and layouts shall be due to Canvas under a known assignment name by 10 AM, Tuesday, October 31st for ordering and submission.*

## Deliverable 1&2

Create a KiCad Schematic and PCB for your project.

## Deliverable 3

Create a high-level system design diagram, this diagram should show how the different modules created in lab 7 interact with each other and the rest of the system.

## Deliverable 4

Record in your report the total cost estimation generated from the BoM. Additionally, record the total cost of only the components which count towards competition eligibility.

## Deliverable 5

Estimate the current and power consumption of each subcomponent in active and idle states, including microcontroller and I/O devices. Record this estimate in the lab report.

Assuming your prior estimate is correct, determine the total current and power consumption from the power source. Record this estimate in your report. **Note:** a system may draw more power than the sum of its components due to the regulators used to create the given voltage rails.

## Deliverable 6 (5pts Extra Credit)

Have at least 1 member of your team complete 3-D printing or laser cutting training in the Texas Invention Works. There are sign-up sheets for training on the 0th floor of the Texas Invention Works. Provide some proof that you got training in your report.b

## Deliverable 7 (10pts Extra Credit)

Use a TM4C123 chip in your design in addition to the launchpad. You will need to add the component to the PCB like in Lab 6. You will need to ensure that it is Launchpad redundant (e.g. if the chip fails, the LaunchPad can be plugged in and used). One way to make it redundant is to add headers to the PCB where jumper can be run from the launchpad to the PCB.

## Deliverable 8 (10pts Extra Credit)

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:

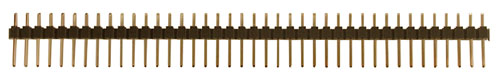
* How you performed this measurement
* How/if the measured behavior meets your system’s requirements
* And what sources of errors may affect your measurement

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 all ISRs
* 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.)

# Hints

1. You must collect ALL parts (except battery, connector, and cables) while you are doing Lab 7 so you can be sure they function properly.
2. There are two types of LEDs you can get from the professor’s office. Low current red/yellow/green HLMP-D150 LEDs can be connected directly to a microcontroller output using just a 680 ohm or 1 kohm resistor. The other colors and sizes that we have require 10 mA and will need an interface (like a 2N2222 and a 100ohm resistor.) You should test the LED/resistor circuit on a breadboard to make sure the brightness is acceptable.
3. Lab 8 hint: My advice is to do a little bit of Lab 7, and then have someone check it. DO NOT DO THE COMPLETE DESIGN SCH/PCB THEN GET IT CHECKED. To have Lab 7 checked, you can contact your TA, or email the SCH and PCB files to your professor. We will evaluate your SCH files for gross design errors in the I/O interface PCB files for style (line width, mitered corners).
4. The datasheets for the components used in this cabinet tab of the BOM can be found on the datasheets page http://users.ece.utexas.edu/~valvano/Datasheets/.
5. Sparkfun and Adafruit have many low-cost displays. The Nokia 5110 is a very low-cost display, and you will find lots of starter code for it. We have a starter code for the Kentec display. See http://users.ece.utexas.edu/~valvano/arm/#HumanInterfaces.
6. Lab 8 hint: Everyone’s embedded system must be placed in a box, so you should think about the box during Lab 7. Starting to think about squeezing all the components into the little box once you get to Lab 11 will be difficult. Placing components in the proper place on the PCB during Lab 8 will greatly simplify the box-building process. See Figures 7.1 and 7.7.
7. Lab 8 hint: KiCad has a feature that allows for ground planes (copper pour). WE DO NOT ALLOW you to use this feature. Ground planes are useful for high frequency and/or low noise systems. The ground plane makes it much harder to visually see what wire connects to what pin, it makes it much harder to cut/add traces in Lab 11 to fix mistakes, and it makes it harder to create good solder joints without using a high-temperature soldering iron.
8. Lab 8 hint: One common mistake new PCB layout designers make is placing two wires too close to each other. Subsequently, during fabrication, these two wires may become shorted because of the tolerance of the manufacturing process. A general rule of thumb is that you should allow enough space between two wires to fit the smallest allowable trace between them. For this PCB manufacturer, separate all traces by at least 0.007 inch.
9. Lab 8 hint: Male-male header pins. This photo shows a straight header with 0.1in separation. This can be broken into any number of pins and used for connectors or mode selection.



2-pin jumper, [SJ-1], $0.10



## Other Stuff

A computation music game, do a web search for Otomata:

* http://earslap.com/projectslab/otomata/?q=5s6x3m2g402z4o6k4q8k480z6g512x3p1z4t7k44
* https://www.youtube.com/watch?v=lHCdHh1eSi0

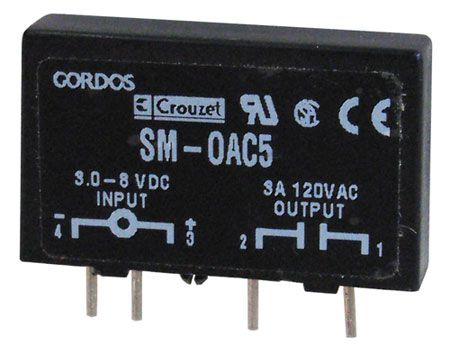
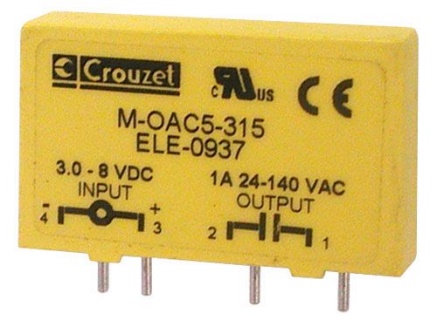
Web site references:

1. Enclosure manufacturers
   1. http://www.OKW.CO.UK/ OKW
   2. http://www.TEKOENCLOSURES.COM/ TECO
   3. http://www.PACTECENCLOSURES.COM/ PACTEC

Most box manufacturers will not ship samples to students.

1. General purpose suppliers
   1. http://www.SPARKFUN.COM Spark Fun - Transducers, Buttons, Displays, etc. Lots of cool stuff.
   2. http://www.allelectronics.com All Electronics - All sorts of random stuff
2. LCDs
   1. http://www.newhavendisplay.com/ New Haven Displays - LCD Manufacturer. Has a bunch of displays around $10
   2. http://www.varitronix.com/ Varitronix - LCD Manufacturer.
   3. http://www.crystalfontz.com/ Crystalfontz - good quality and price LCD's
   4. http://www.sparkfun.com/products/710 $20 Sparkfun LCD-00710 64 by 128 LCD
   5. http://www.circuit-ed.com/128x64-BLWH-TOUCHSCREEN-GLCD-P146.aspx $26 64by128 LCD touch screen
   6. http://www.sparkfun.com/commerce/product\_info.php?products\_id=8977 Touch Screen
3. Batteries
   1. http://www.powerstream.com/ PowerStream – Batteries
   2. http://www.batteryjunction.com/ Battery Junction - Li-Ion packs
4. Industrial suppliers
   1. http://www.digikey.com/ Digikey - Useful parametric search. Lots of standard components. This is the easiest way to find multiple manufacturers of something common like connectors
   2. http://www.mouser.com/ Mouser - Pretty much any standard chip you want is available here. Ships from Dallas, so usually faster than Digikey, but sometimes a bit pricier.
   3. http://www.avnet.com/ Avnet - Some higher-end stuff and hard to get chips can be found here.
   4. http://www.newark.com/ Newark - Similar to Avnet

http://www.ladyada.net/library/procure/samples.html Getting Samples - How to get free samples.



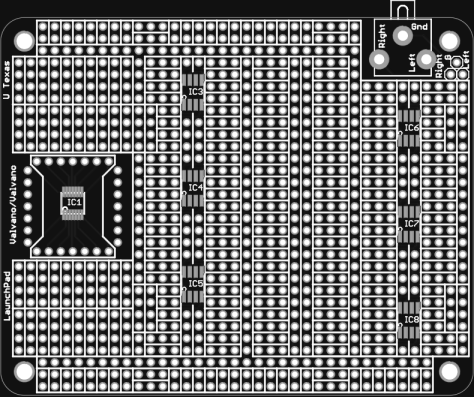
CAT# SRLY-503 www.Allelectronics.com

http://www.crouzet-ssr.com/english/products/download/M\_output\_modules.pdf

18-bit color, 128\*160, 1.8" TFT LCD display, Sitronix ST7735R, www.Adafruit.com part number 358, $19.96



If a project is so much more complicated than the typical ECE445L project that they need two boards, please see your professor. One option for creating more than one device is to make one with the PCB layout and make the second one by soldering parts onto an ECE319K project board.



ECE319K Project Board.

Lab 8 hint: If you want to learn much more about PCB design (details beyond what is needed for ECE445L, but awesome if you wish to create PCBs professionally), Matthew Yu created two playlists with many hours of educational material.

https://www.youtube.com/playlist?list=PLqUBXn7oPxmxXZYo1X\_91ucJfMEAJdrzt (PCB Design)

https://www.youtube.com/playlist?list=PLqUBXn7oPxmzVwcAnNQxI12CLg\_SvGdOF (PCB Backlog)