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.

Team Size: 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 (also found in hw folder of your GH repository)

* **EE445L** library for Eagle, dated 12/29/2022 or later.

<https://www.dropbox.com/s/m0r4mr03hiykuq9/EE445L.lbr?dl=1>

Micro USB socket <https://www.dropbox.com/s/9yjg506if6fuea0/105017-0001.lbr?dl=1>

* Lab7 .SCH and .BRD starter files, (Spring 2023 Eagle version)

<https://www.dropbox.com/s/xc3v7ii0cdpisnf/EE445L_Lab_7_Starter.sch?dl=1>   
<https://www.dropbox.com/s/dq9vqqrtaxg6yja/EE445L_Lab_7_Starter.brd?dl=1>

* Lab7BOM.xls (Spring 2023 version)

<https://www.dropbox.com/s/8x11h7qlpift6cf/Lab7BOM.xlsx?dl=1>

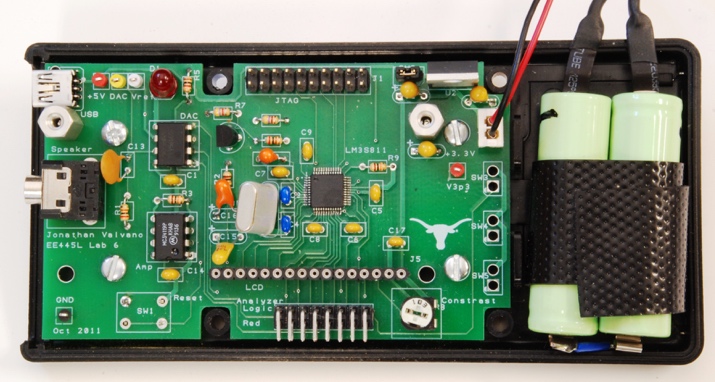
 

Figure 7.1. Example mock-up and eventual Lab 11 system.

## Background

In Lab 7, you will use the CAD program **EAGLE PCB** to design an embedded system. The design of the system must satisfy certain requirements. Rather than simply redesigning one of the previous labs like you would have done in Lab 6, this embedded system must do something useful. There are some ideas posted at <http://users.ece.utexas.edu/%7Evalvano/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 the development of a microcontroller-based embedded system demonstrating the educational objectives of this class.

Additionally, we will ask to you prototype your embedded system to investigate its feasibility as you get farther into the design process. This is to attempt to ensure that your final product works as intended without any hiccups before you order your PCB from the PCB manufacturer. As part of this prototyping process, you will learn how to develop unit and integration tests for an embedded system, among others. Here’s an explainer (among others on the internet) explaining the two types of testing: <https://www.headspin.io/blog/unit-integration-and-functional-testing-4-main-points-of-difference>.

In these tests, we’d like you to take the debugging and profiling techniques you’ve learned over the semester and apply them. If you encounter a bug during unit testing, you may need to use dumps, print statements, or debugger to fix it. To validate your performance requirements during integration tests, you may need to measure jitter and latency to observe how different submodules may affect each other.

## System Requirements

* MINIMUM COMPLEXITY: The embedded system must have at least two inputs, two outputs, and two interrupt service routines. The system must also contain four or more identifiable subcomponents, and each team member must be responsible for at least one major subcomponent. TAs will evaluate whether the project is sufficiently complex.
* OTHER MCUS: Other microcontrollers or boards may be used – as coprocessors only. Most of the system and software complexity should occur in the TM4C and ESP8266.
* NUMBER OF PCBS: Each team must produce at least one PCB design.
* BOARD SIZE: The maximum PCB size is 30 in2.
* COMPONENTS: All components must be through hole – no surface mount parts without approval from the instructor.
* ENCLOSURE: The final embedded system must fit into an appropriate enclosure (wood, metal, mu
* DESIGN SOFTWARE: The embedded system must include schematics and layout (.SCH, .BRD file) having been created with **EAGLE PCB**.
* BOARD ID: Teams must have an identifier on their PCB in the format **UTX-2023Sxxx.sch**, **UTX-2023Sxxx.brd** for review and ordering purposes. This must also be present on the PCB silkscreen itself, alongside student identifiers on both sides of the PCB. The **xxx** number will be assigned by the TA.
* DESIGN CHECKS: Each **PCB** must be done using the JLCPCB design rule checks (<https://jlcpcb.com/capabilities/Capabilities>). The starter files also have minimum trace and other distance metrics that cannot be changed.
* BUDGET: The maximum budget for being eligible for the final competition is **60$**. You may spend more for your project but the final system BoM at competition must cost at most **60$**.
* GOING BEYOND: doing any of the following means that teams **will be responsible for ordering and paying for their own PCBs**. We are also not obligated to check them. Do at your own risk!
  + You may use another design tool, such as KiCAD, if all team members agree and there is at least one person well-versed in the tool.
  + You may make use of advanced PCB manufacturing features, such as 4-layer PCBs, SMT assembly, etc.
  + You may make PCBs larger than the limit. But why would you possibly want to do this unless you are starving for space?
  + You may attempt to reorder PCBs and make new prototypes up until competition day (hard real time deadline!)
  + You may make more than one design or companion PCB in addition to the one specified earlier in point NUMBER OF PCBS.
* FINALLY: The embedded system should be useful, interesting, or fun! The team must also have fun when building it!

## Other Notes

* You can use the ESP8266 from the hardware kit. Use a 2 by 4 socket, so no adapter board needed.
* ECE lab checkout EER 1.824 has through-hole resistors (1/4 watt) and through-hole capacitors you can checkout, solder onto your PCB and never return. The professor has 1/8-watt resistors.
* **Lab7BOM.xls** has a specific list of parts that we will be able to give you.
* You must purchase, borrow, or get free samples for any additional parts that you require.
* You can use two motors from the Lab 10 supplies to build a robot (we have matching wheels), but the motors and wheels must be returned.
* We strongly, vehemently recommend that you should **NOT** use copper pours unless you are experienced at PCB design because it makes modifying or repairing the board exceptionally challenging if you were to make a design or layout error.

## Processor Selection

***Implementation option 1****.* We expect all students to design a complete system by attaching the TM4C123 Launchpad onto the PCB. For this option, use the Lab 7 starter files. You must include a power regulator on your PCB because the onboard Launchpad 5V to 3.3V regulator only supplies up to 400mA, much of which is consumed by the TM4C123GH6PMs on the Launchpad.

***Implementation option 2****.* You can implement a complete system using a TM4C123GH6PM chip soldered onto one PCB, like Lab 6. For this option, we will solder one chip onto one PCB using the oven. You will need to work closely with the professor and your TA during the design and layout. For this option, use the Lab6 starter files. To salvage your project and grade during Lab 11, we require you to also place the 40 pins for the Launchpad on the same PCB. This way if the TM4C123GH6PM microcontroller never boots up, you can unsolder the TM4C123GH6PM, and attach your EK-TM4C123GXL Launchpad board. You must include a power regulator on your PCB. We will have enough TM4C123GH6PM chips for each team to have one.

We recommend if you have any unused pins, that you route some of the pins away from the processor and connect them to vias or logic analyzer pins. This way you can convert unused pins to debugging pins or use them to fix mistakes that you have discovered during testing.

## Optional Components to Consider

***Other microcontrollers.*** Your PCB must have a TM4C123. However, in addition to the TM4C, your system may include microcontrollers. Some students add a Raspberry Pi to provide for quantum boost in processing power. If you use a microcontroller in addition to the TM4C, you can get full credit, but will not be eligible for the “best design” award.

***ESP8266 Wi-Fi.*** If you are adding the ESP8266 be aware the device requires currents generally greater than 70 mA (although the data sheet reports up to 200 mA). Therefore, select a +3.3V regulator that can supply this required current (i.e., do not use the LP2950, rather use the LM2937-3.3). For more information on the ESP8266 refer back to Lab 4.



Figure 7.2. ESP8266 version 1.

***Audio amp.*** For a single channel you can use an audio amp like the MC34119 or the TPA731. For stereo sound, use two separate audio channels and two speakers (4 wires).

***Accelerometers and other sensors****.* There are lots of devices available as free samples. However, be very careful because most of them are very difficult to solder because the pins are underneath the device. The best way to add acceleration is to get a module (chip soldered onto a board) from a hobby store like Sparkfun, Adafruit, or Pololu.

***Enclosure.*** There are many approaches for the enclosure. Please check with the Texas InventionWorks staff before cutting, drilling, or milling boxes. NOTE: You will need to be trained on the equipment you want to use.

Approaches:

1. We will give you a Hammond 1591ESBK for your Lab 7,8,11 project if you wish. The cost of this box does not count towards your $60 budget. This box is general purpose Acrylonitrile-Butadiene-Styrene, GP-ABS. This material will outgas chlorine if melted, so it cannot be cut with the laser cutter, but it can be safely cut with a saw or drilled with a drill. This box is 7.5" by 4.3" by 2.2" and the data sheet is <http://users.ece.utexas.edu/~valvano/EE345L/Labs/Fall2011/Hammond1591E.pdf>
2. You may purchase a box and count it towards your $60 budget.
3. You may build your own box. You may use any material, such as wood, plastic, metal, or cardboard. You will not be judged on the beauty of the enclosure, rather you will be judged on the functionality of the system. The cost of these materials does not count towards your $60 budget.
4. You may build your own box at the Texas InventionWorks. Any costs for raw material used to build the box do NOT count towards your $60 out of pocket limit. Note that students can successfully complete labs 7,8,11 without spending any out-of-pocket money. Design the box in a CAD program like Fusion 360, create 2-D pdf files for the 6 sides and cut the pieces out of acrylic/wood on the laser cutter.

## Final Design Competition

There will be a “Science Fair”-like public demonstration for Lab 11. We will present special awards to the team with the best design. The judging will be performed by the public during the demonstration.

Some students will put extra electronics off the PCB, because it doesn’t all fit on the PCB. If you do have off-board electronics, then you will need a connector or something to create the bridge. You can get good grades in Labs 7, 8 and 11 with off board electronics, but you will not be eligible to win “best design”. Your grade depends on if the required tasks are completed on time, and if your eventual project (I/O, microcontroller, and software) works.

To win “best design” you will need to meet the following restrictions:

* Uses only TM4C123 as the primary microcontroller.
* All electronics (resistors, capacitors, ICs) are on the PCB.
* LCD displays, switches, sensors, LEDs, speakers, keypads, and microphones can be off the PCB
* Your team of 4 spends less than $60 on extra components.
* This $60 extra expense can be negotiated with your TA.

### $60 Limit

There are two system costs you will add up. The first cost is the sum of all the components regardless of where the part was obtained. This would be the cost of manufacturing the system. The second cost is the sum of all the parts you purchased to put into the device. Components from ECE lab checkout or from Lab7BOM.xls do not count against your maximum of $60 additional components. Free samples, the PCB, batteries, and components checked out of the ECE lab checkout desk do not count towards your $60 limit but should be listed with fair market price in Procedure e. Passive components (resistors, capacitors, male-male header pins, test-points) you get from the ECE lab checkout desk do not have to be returned. Solid state parts from checkout may have to be returned, so please ask at the checkout counter. If you wish to use a part you already own, or a part given/lent to you by another, then you must find the part in stock and report the cost as part of your $60 limit.

## Pre-preparation

Write a one-page **Requirements Document** for the system. Refer to Labs 3 and 5 for general information about requirements documents. We expect this document to change as you develop your system, so keep it up to date as you progress through the design, implementation, and testing phases. However, remember that the requirements direct the design – not the opposite! Please use this outline:

**1. Overview**

**1.1. Objectives:** Why are we doing this project? What is the purpose?

**1.2. Roles and Responsibilities:** Who will do what? Who are the clients?

**1.3. Interactions with Existing Systems:** Include this if you are connecting to another board.

**2. Function Description**

**2.1. Functionality:** What will the system do precisely?

**2.4. Performance:** Define the metrics used to evaluate the performance of your system and describe how they will be measured.

**2.5. Usability:** Describe the interfaces. Be quantitative if possible.

**3. Deliverables**

**3.1. Reports:** Simply state the reports for Labs 7 and 11 will be written at a future date.

**3.2. Outcomes:** Simply copy/paste the Lab 7 and Lab 11 deliverables when they are complete.

We remind the teams that there must be at least four identifiable subcomponents (including but not limited to sensors, actuators, Wi-Fi, BLE, enclosure, power, user interface) in the system, and each student is responsible for at least one subcomponent. TAs will judge if the project is sufficiently complex and may ask the students to revise their project idea and redo the requirements document. This subcomponent breakdown is required as part of the requirements document.

Alongside the requirements document, students should **present a preliminary parts list and datasheet collection** and be able to explain how the parts interact with each other in the idealized embedded system. After approval, students should immediately begin ordering/sampling any required parts for them to arrive in a timely manner.

## Parts (should arrive before Lab 7 Demo)

1. Parts you get from ECE lab checkout (1/4-W resistors, capacitors, male-male headers, and test points)
2. Parts you plan to buy from places like Sparkfun, Robotshop, Adafruit, Mouser, Digikey, or Pololu.
3. Parts you plan to get from the professor’s cabinet.

## Preparation (do this before your lab period)

1. **Create a bill of materials** of all parts using the **Lab7BOM.xls** template. Collect all ICs, sensors, displays LEDs, resistors, and capacitors. Basically, collect everything except batteries, connectors, and cables, and enclosure (these are part of Lab 8).
2. **Draw a data flow graph** showing hardware and software modules.
3. **Create a software call graph** showing how software modules are interconnected.
4. **Draw the circuit diagram** (SCH file) using Eagle. Refer to the suggestions and guidelines described in Lab 6. Use file names as given by TA. Make sure you use starter files for the current semester. Add features such as test points and labels that will make it easy to test the hardware. You must also ensure you are following these rules:
   1. No net names that begin with **N0**;
   2. All components need labels (e.g., U1 R1 C1 J1 etc.), shown both on the board and the circuit diagram.
   3. Each IC should have a bypass capacitor, placed on the PCB as close to the chip as possible.
   4. For resistors, specify wattage (1/4 watt) and tolerance (5% carbon).
   5. For capacitors, specify type as 20% Z5U ceramic, or 10% tantalum.

This schematic should pass ERC by the time the TA checks you out.

1. **Write the header files** for all software modules.
2. **Determine how the system will be powered**, and how the power can be turned on and off. You must use a regulator on the PCB. The typical configuration has battery or USB plug into the PCB, an off-board power switch, and then the LM2937-3.3 regulator for 3.3V supply. In this configuration, the TM4C123 gets its power from the LM2937-3.3, and the 2-pin jumper on the Launchpad is removed.
3. Does the operator need to hit the reset button? If so, bring the reset pin out to a user-reachable negative logic switch.

## Procedure (do during the lab period)

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 may be appropriate to use a separate protoboard and 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 integration will be a Lab 11 task. The goal is to perform enough low-level testing to be confident 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.

## Valvano Review (see syllabus for due dates)

1. Email both UTX-2023Sxxx.sch UTX-2023Sxxx.brd files (even though the BRD file is unedited) to [valvano@mail.utexas.edu](mailto:valvano@mail.utexas.edu), and come to an office hour for a quick review. This review can be done over zoom. All members need not be present. Perform this review as soon as most of the SCH file is complete and none of the PCB file has been started. There will be expanded office hours the week of 3/7. Valvano will email you back the SCH and BRD files with his comments.
2. After your Valvano review, update your **Lab7BOM** excel sheet as needed, and make a copy of it, delete all rows except where column L specifies **EER5.822 (Professor)**. Put ALL team members in the first row. Put your UTX-2023Sxxx part number on row 2. Delete all columns except (A) Quantity, (D) Description, and (L) Where to get it.

Email this edited excel sheet to [valvano@mail.utexas.edu](mailto:valvano@mail.utexas.edu). Valvano will put parts in a bag and leave the bag with your printed excel sheet in lab.

## Demo (see syllabus for due dates)

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.

## Deliverables (exact components of the lab report)

1. A summary of the lab and what the student expects to learn from it.
2. An updated requirement document.
3. A summary of the software and hardware that was created to complete the lab (keep the full software on GH, paste a screenshot/printout of HW in the report).
4. Measurements taken by the student to verify the design, in particular:
   1. Current and power consumption of each subcomponent in active and idle states, including microcontroller and I/O devices. You measured the supply current during Labs 3 and 5.
   2. An estimation of the total current and power consumption from the power source. Note that a system may draw more power than the sum of its components if lower voltage sinks source current from an LDO regulator.
5. A cost estimation generated from the BoM.
6. Analysis and discussion, particularly:
   1. How did you debug the system? Was it intrusive?
   2. What’s the difference between unit testing, integration testing, and functional testing?
7. Any extra credit components.

## Lab 7 grading

1. **(5) Attendance**, graded by (1) presence (in person or otherwise) at lab prep AND demo days, as well as (2) equal contribution to lab completion.
2. **(10) Lab pre-prep**, graded by initial requirements document, and preparation of parts list and datasheets. Students shall need to be able to explain to the TA the project pitch, and how they intend to accomplish it. This will be like an MRD (Marketing Requirements Document) presentation to your TA. MRD is what it does from a user perspective (how can we sell it). You must be able to discuss the following parts of your requirements document:

1.1. Objectives: Why are we doing this project? What is the purpose?

1.4. Interactions with Existing Systems: How will it fit in?

1.5. Terminology: Define terms used in the document.

2.1. Functionality: What will the system do precisely?

2.5. Usability: Describe the interfaces. Be quantitative if possible.

3.3. Outcomes: What are the deliverables? How do we know when it is done?

1. **(10) Lab prep**, graded by the design of the system. By the prep-day you should have a very clear idea about your project. So, you should be able to describe the lower-level interface of your system. You can go through your schematic and BOM while you explain it to the TA. While you do this you would also like to point out your hardware and software design boundaries. This will be like a PRD (Product Requirements Document) presentation to your TA. PRD is how it will be developed from an engineering perspective (how do we build it). Additionally, you should be able to discuss the following parts of your requirements document:

1.2. Process: How will the project be developed?

1.3. Roles and Responsibilities: Who will do what? Who are the clients?

2.2. Scope: List the phases and what will be delivered in each phase.

2.3. Prototypes: How will intermediate progress be demonstrated?

2.4. Performance: Define the measures and describe how they will be determined.

1. **(30) Lab demo and Q/A**, graded by the implementation of the system. You must show off subcomponents of your project working independently on a breadboarded prototype. You must also demonstrate that the project is of sufficient complexity.
2. **(10) Professor review** of schematic, completion grade.
3. **(20) Software and/or hardware quality.** Software (or hardware quality) of the lab will be evaluated by focusing on the quality of the design and formatting of the code or hardware schematic/layout. This includes but is not limited to: consistent syntax and proper use of version control, a clear flow of logic or proper handling of edge cases and user inputs, schematic/layout that passes ERC/DRC, and understanding of placement of components or selection of components relative to each other (e.g., why XX capacitor and YY resistor for this low pass RC filter?).
4. **(15)** **Lab report**, only one member of the team must submit.

## Bonus (Can be obtained either during Lab 7 or 8)

1. **(5 points)** Have at least 1 member of your team complete 3-D printing or laser cutting training in the Texas InventionWorks. There are sign-up sheets for training on the 0 floor of the Texas InventionWorks.
2. **(10 points)** Use a TM4C123 chip in your design in addition to the launchpad. Follow the instructions in processor implementation option 2.
3. **(10 points)** Validate at least two performance metrics of your system from the following list:
   * CPU utilization (thread profile) measured separately for each module.
   * Ping latency with Wi-Fi
   * Maximum execution time for ISRs
   * DAC or ADC sampling jitter
   * Signal to noise ratio (SNR)

You must also provide a short write up (~1 paragraph, 1-2 sentences ea.) for each of the metrics validated describing:

* + How you performed this measurement,
  + Does the measurements meet your system’s requirements,
  + And what sources of errors may affect your measurement.

## 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  or 1 k resistor. The other colors and sizes that we have require 10 mA and will need an interface (like a 2N2222 and a 100  resistor.) You should test the LED/resistor circuit on a breadboard to make sure the brightness is acceptable.
3. 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. 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. Eagle 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. 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. 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>

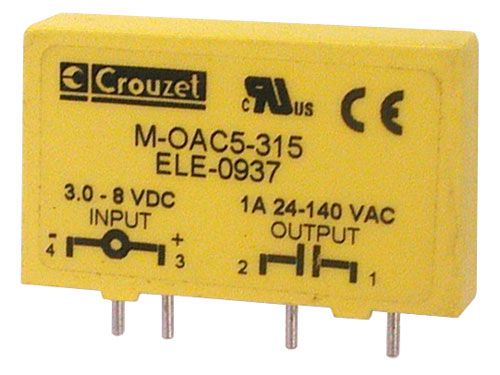
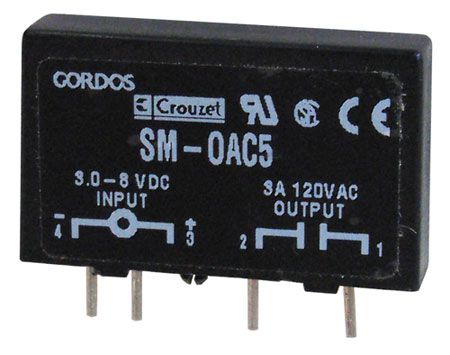
Web site references:

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.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](http://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.

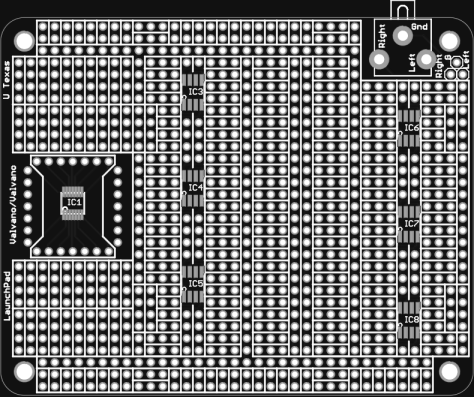


Figure 7.6. ECE319K Project Board.

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)