ECE 445L Lab 8

PCB Layout 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](#_Toc965531379)

[Team Size 1](#_Toc2051058422)

[Goals 2](#_Toc769711026)

[Review 2](#_Toc1259863722)

[Starter Files 2](#_Toc1214983682)

[Required Hardware 2](#_Toc2065973084)

[Lab Overview 2](#_Toc1593293404)

[Preparation 2](#_Toc390183934)

[Procedure 3](#_Toc1449142582)

[Lab Checkout 6](#_Toc672487843)

[TA and Professor Review 7](#_Toc1152153183)

[Lab Report 7](#_Toc390147558)

[Deliverable 1&2 7](#_Toc363687118)

[Deliverable 3 7](#_Toc1471761394)

[Deliverable 4 7](#_Toc988793925)

[Deliverable 5 7](#_Toc1714898529)

[Deliverable 6 (10pts Extra Credit) 7](#_Toc9505488)

[Deliverable 7 (10pts Extra Credit) 7](#_Toc2039968146)

[Hint (PCB Resources) 8](#_Toc1353419684)

# Team Size

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

# Goals

* Layout a PCB design for your embedded system.
* Integrate embedded subsystems together.

# 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

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

# 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 8, you will finalize the design of your embedded system. Taking the schematic developed from Lab 7, you will convert your electrical design into a PCB design that will be fabricated for your final project submission in Lab 11. Ideally, you and your team will go through multiple cycles of the design process to identify bugs, refine the design, and reduce project risk. However, given the limited time of this class you will only have one pass. One way to reduce your design risk is by performing integration tests that verify that your design will work as expected for the final prototype.

# Preparation

1. At this stage, you should have finalized your system design, a Bill of Materials (BoM), and a schematic that has been demonstrated to work standalone via unit tests. Your schematic in your KiCAD project should pass ERC and have a footprint assigned to every component.
2. Take your finalized schematic and import it into the PCB editor.
3. **Place the components on the PCB** and arrange them such that it meets your teams’ requirements, including:
   1. **Expected board outline:** what is the size and shape of your PCB and how does it mount to your mechanical enclosure and/or systems?
   2. **Expected user inputs:** where is the user expected to interface with any sensors or actuators?
   3. **Design for test:** are the test points or high-risk circuitry easily reachable and debuggable?
   4. **Design for manufacturing:** are the components oriented in positions that make it easy to solder and assemble?
4. **Receive approval** from your TA that your parts placement is satisfactory.

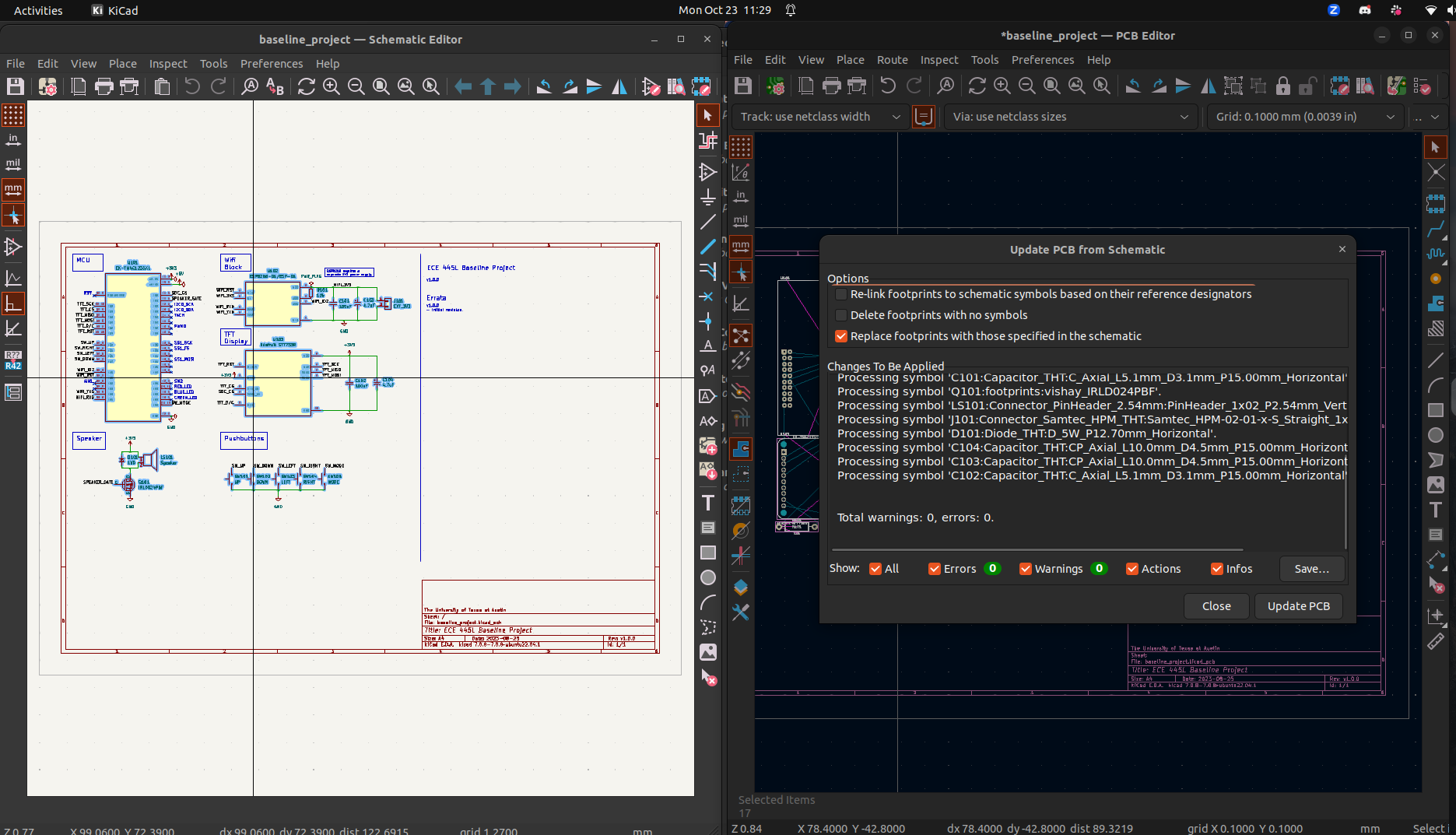


Figure 8.1. Importing Footprints to PCB Editor.

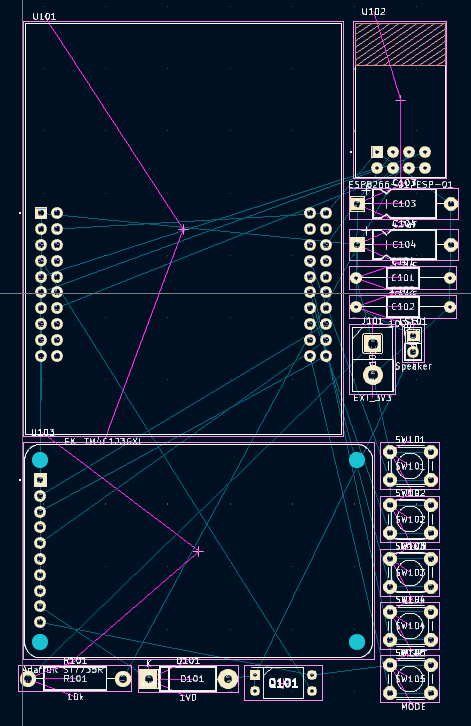


Figure 8.2. Imported Footprints in PCB Editor.

# Procedure

1. **Perform PCB layout** using KiCAD. Make sure that the design passes DRC. Ensure that the following silkscreen is added to the PCB:
   1. Team names, UTX\_FALL\_2023, TA’s name, project name.
   2. Test point and debug labels.
   3. Assembly labels (e.g. dot for first pin of IC).
2. **Install the JLCPCB Fabrication Toolkit plugin** from the KiCAD project menu and use the plugin in the PCB editor. When generated, it should open a browser window to a folder with the production files.

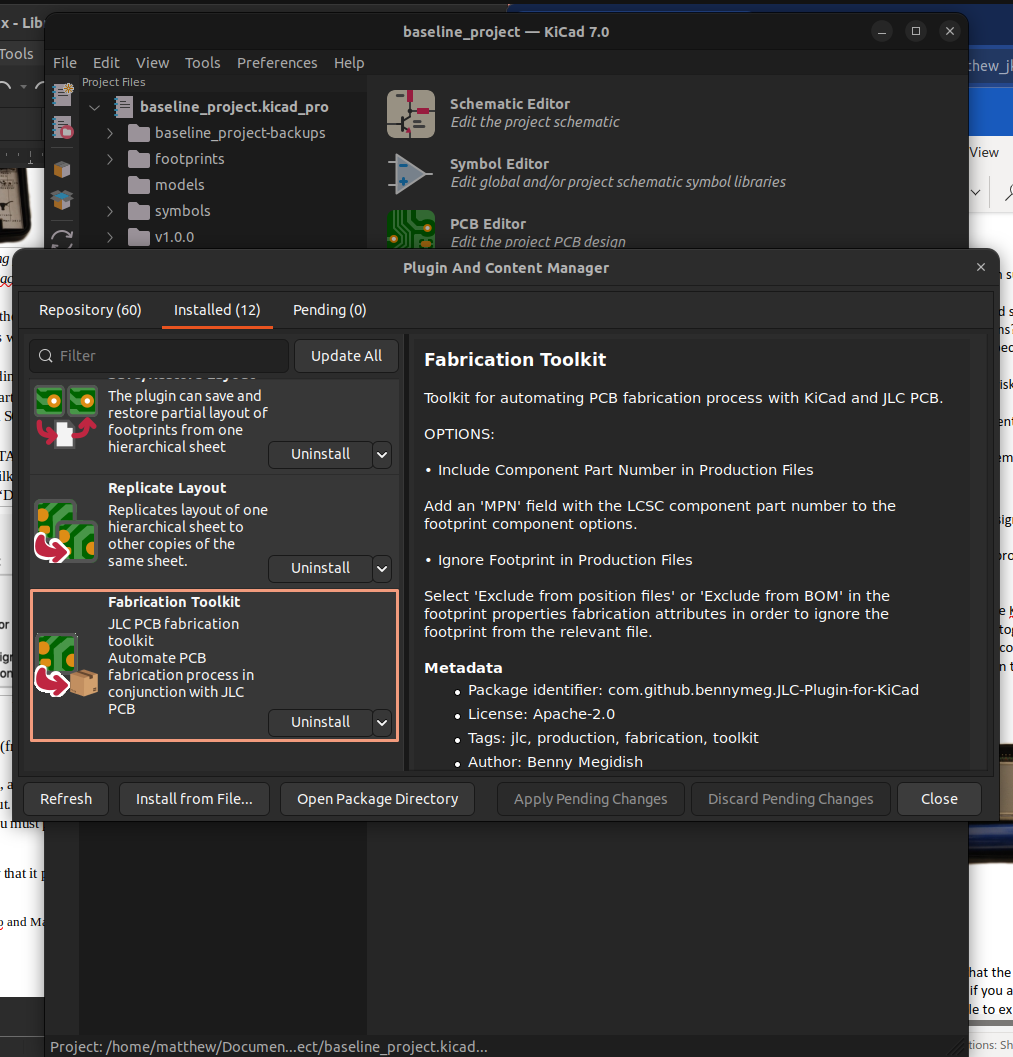


Figure 8.3. JLCPCB Fabrication Toolkit Plugin.

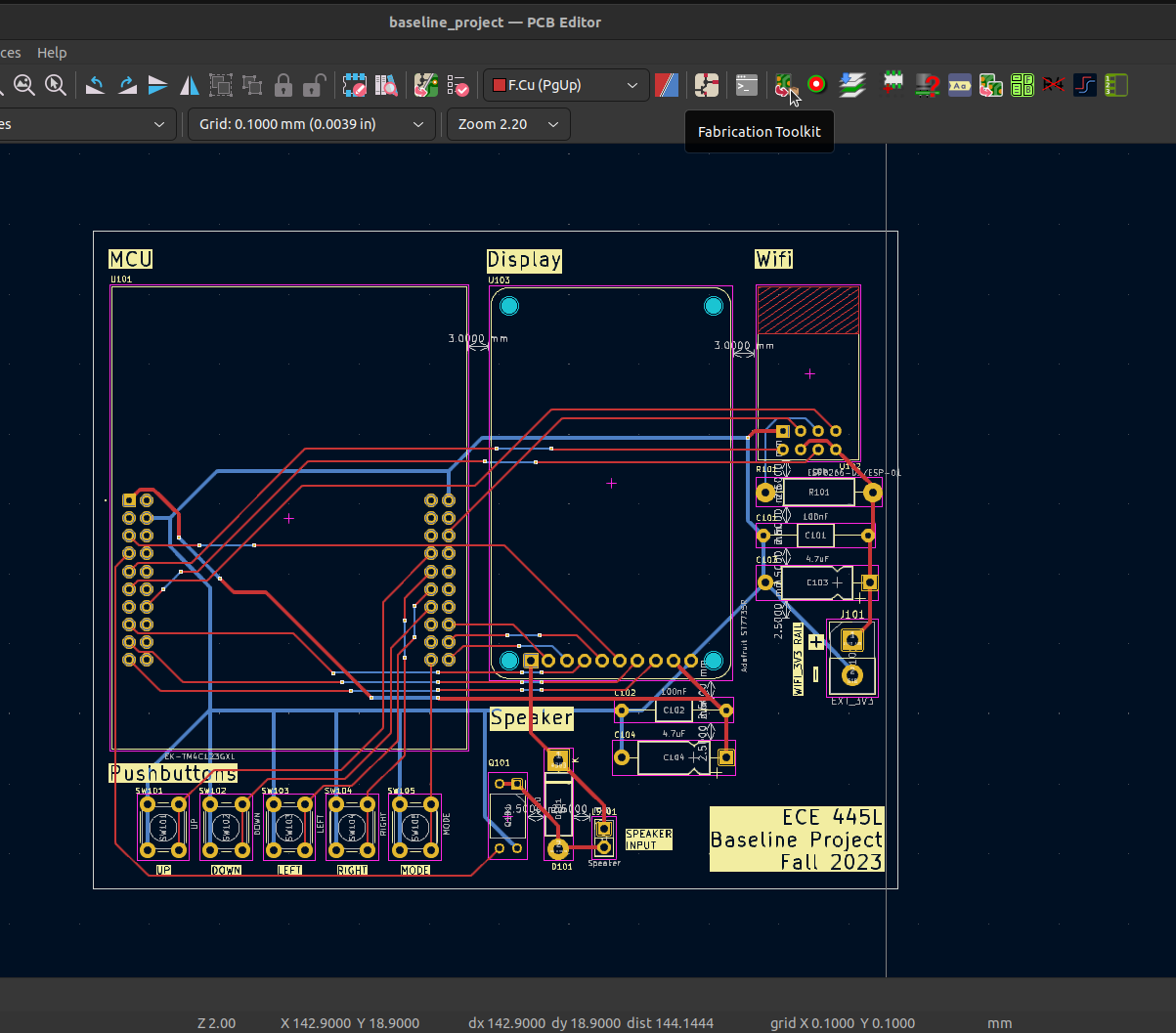


Figure 8.4. JLCPCB Fabrication Toolkit Plugin (PCB Editor).

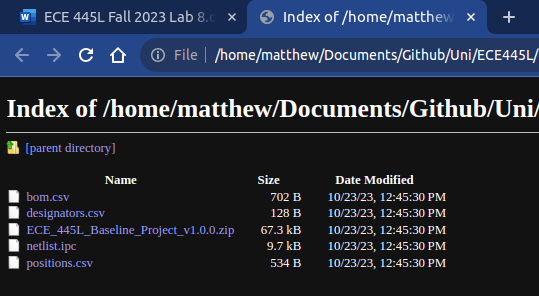


Figure 8.5. JLCPCB Fabrication Outputs.

1. **Go to the JLCPCB site** and provide the zip file when asked to upload gerbers. Verify that the board loads properly in the viewer and the dimensions and costs are not unexpected.

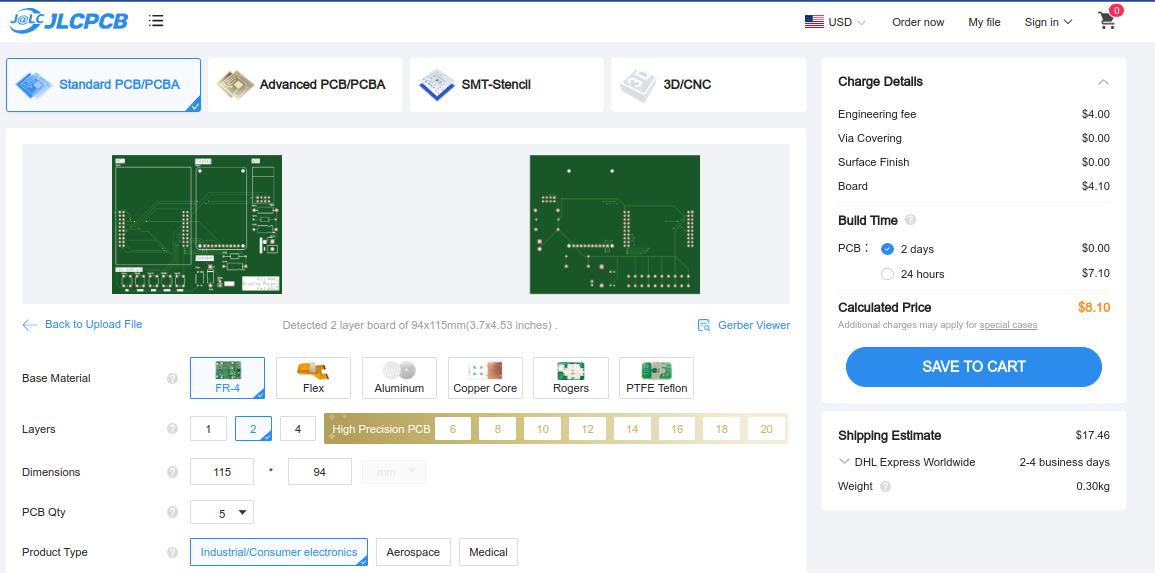


Figure 8.6. JLCPCB Upload.

1. **Print out the PCB** on paper and glue the two pieces together. Punch and drill holes to place components on the simulated “PCB”. Verify that all components fit in the spaces allocated for them. Presumably you have had somebody to design the enclosure by now.

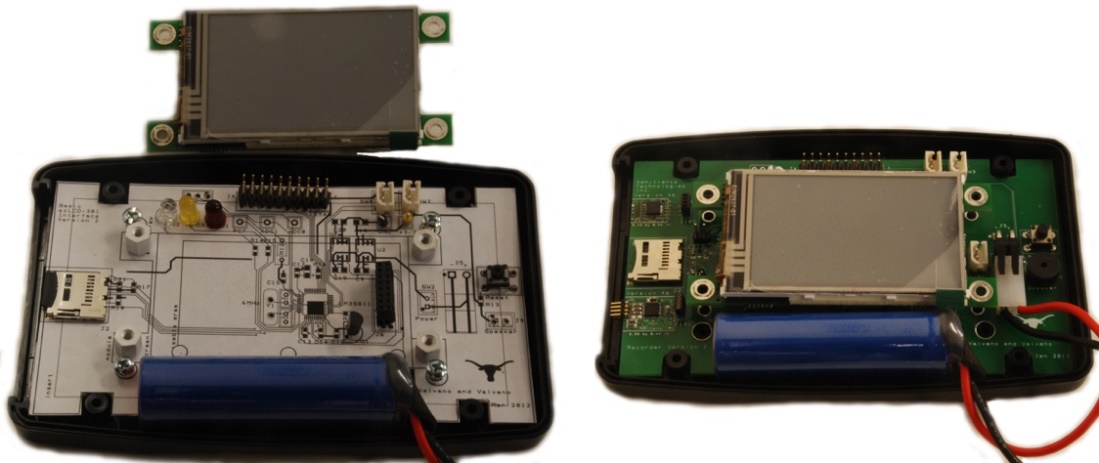


Figure 8.7. Simulated PCB.

1. **Write integration tests** to confirm that your subcircuits work together without breaking functionality. E.g. for Lab 4, a unit test may verify if your UART communication works between TM4C and ESP; an integration test may verify if a button press results in the correct UART message being sent from TM4C to ESP.

# Lab Checkout

1. Open your project in KiCAD and demonstrate that the layout is complete and passes DRC.
2. Explain how the system will be powered. Even if you are using the TM4C LaunchPad, you still are required to have a regulator. You should be able to explain how the regulator works.
3. Demonstrate that your breadboard circuit of the system is functional and there are no major design flaws that may show up in the PCB.
4. Explain your testing procedure for when the PCB arrives. This needs to be a list of 4 or 5 steps, detailing the bottom-up construction and test procedures. Each step will include hardware to be built, software to be written, debugging tools to be used, and expected results to be collected.
5. Show any connectors, I/O devices, or other devices that will connect to the PCB. Explain how they will be powered and communicate with your system.
6. Show that the simulated PCB has correct sizing of the components and that it fits within the enclosure.

# TA and Professor Review

*We preemptively note that students shall submit their preliminary PCB to GitHub.*

*TA & Professor Review (9:00AM, March 29th)*

*Final Submission (10:00AM, April 2nd)*

# Lab Report

## 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

Describe the tests that you added in lab 8 to verify your modules and their integration.

## Deliverable 6 (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. This extra credit can be completed before lab 8 and will be applied to lab 11.

## Deliverable 7 (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.)

This extra credit can be completed before lab 11 and will be applied to lab 11.

# Hint (PCB Resources)

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)