

EEL3926

**Electrical and Computer Engineer Design
Lab Manual**

University of Central Florida



Table of Contents

PRELAB	1
SOLDERING RULES AND SAFETY	1
SOLDER TRAINING FOR JUNIOR DESIGN (PCB DESIGN AND FABRICATION)	2
GETTING AUTODESK FOR EDUCATIONAL USE	11
WEEK 1	12
LABORATORY 1: BILL OF MATERIALS (BOM).....	12
WEEK 2	16
LABORATORY 2: WEBENCH	16
WEEK 3	20
LABORATORY 3: REGULATOR PROTOTYPE	20
WEEK 4&5.....	31
RESOURCE: ULTRA LIBRARIAN AND FUSION360 LIBRARIES.....	31
LABORATORY 4: FUSION 360 PCB DESIGN	33
LABORATORY 5: PICK AND PLACE INSTRUCTIONS.....	50
WEEK 6	58
LABORATORY 6: DATASHEET AND RESEARCH	58
WEEK 7&8.....	62
LABORATORY 7: JUNIOR DESIGN PROJECT PCB.....	62
WEEK 9&10.....	67
LABORATORY 8: JUNIOR DESIGN PROJECT PROTOTYPE / SOFTWARE	67
WEEK 11-14	78
LABORATORY 9: PCB FABRICATION AND SOLDERING (REGULATOR BOARD)	78
LABORATORY 10: PCB FABRICATION AND SOLDERING (FINAL PROJECT BOARD).....	82

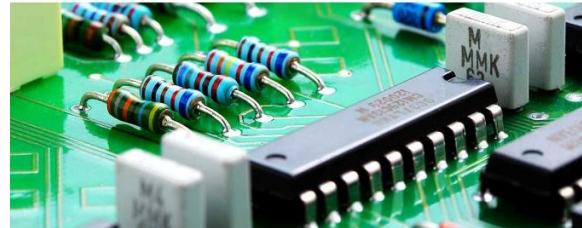
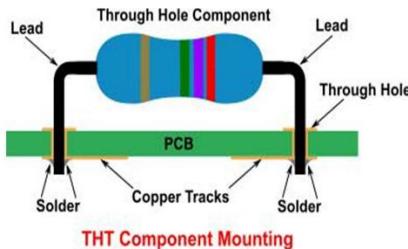
PRELAB

SOLDERING RULES AND SAFETY

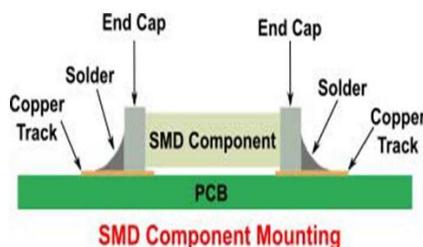
1. The soldering station temperature should be set to 340 to 370 degrees C.
2. When the soldering station is not being used it should be turned down or off.
3. Always clean the tip with the wool cleaner and the water sponge.
4. Also add clean solder to the tip when you begin and end soldering.
5. Use the amount of flux as required.
6. Turn down the air flow on the hot air gun so not to blow the SMD parts on the PCB.
7. Always use safety glasses (this is a requirement with no exceptions).
8. When soldering turns on the vent fans.
9. When handing solder paste use gloves.
10. Never touch the hot end of the solder pencil or hot air gun.
11. Use the ground strap when dealing with sensitive ESD parts.

SOLDER TRAINING FOR JUNIOR DESIGN (PCB DESIGN AND FABRICATION)

1. Design your electronic circuit.
2. Use a spice simulation tool to simulate your design.
3. Next start to lay out your PCB. Here you must decide on what type of parts to use: through holes or surface mount. SMD parts are more popular and have a wider selection.



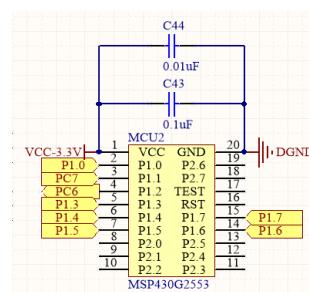
Through Hole Components



Surface Mount Devices (SMD)

<https://www.pcbpower.us/>

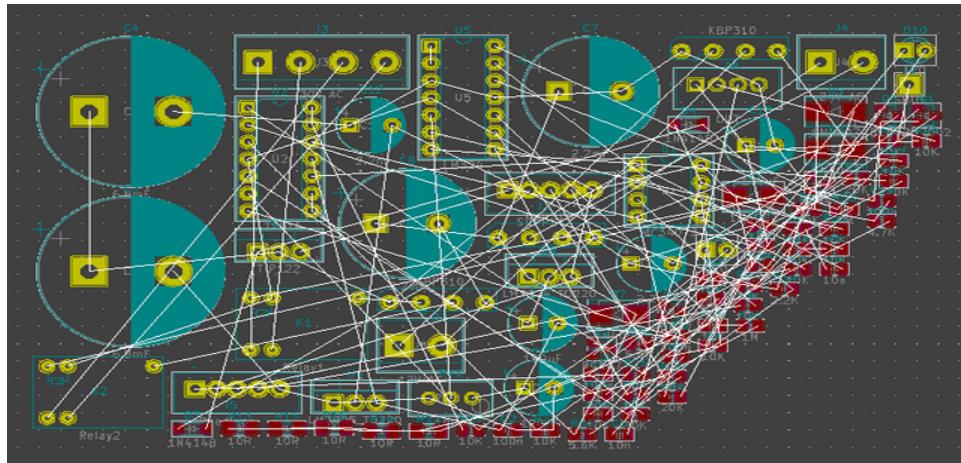
- a. Using the desired PCB design tool like Cadence, Eagle, PADS, Fusion 360, Altium etc. next you will draw your schematic. You will probably need to import your parts into the parts library of the PCB software.



From TI.com

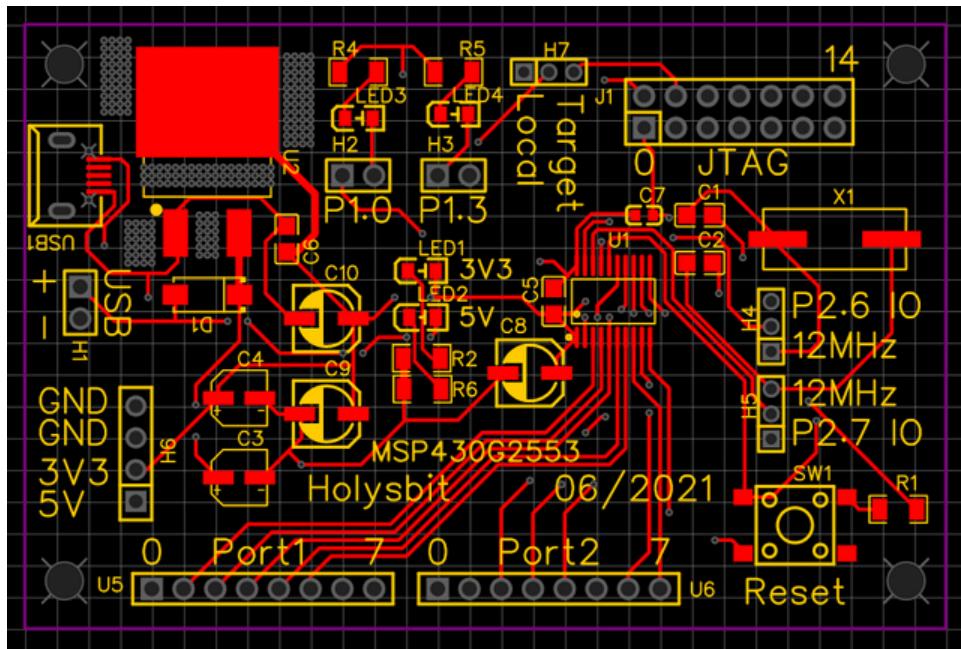
Note all the pin numbers, function of each pin and the part number are defined.

- b. Next, the PCB software is going to generate a PCB layout that looks like this called a “Rats Nest”.



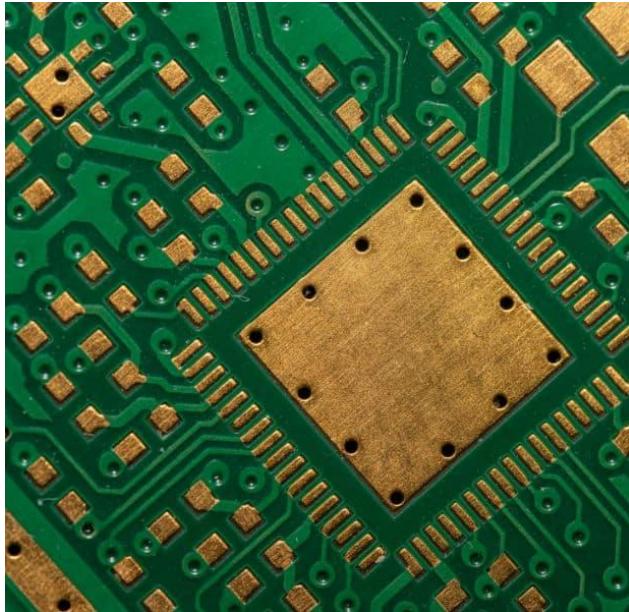
<https://www.reddit.com/>

- c. The PCB layout software must route the traces. Do not forget to use ground plains both top and bottom layers if required. The ground planes can be tied together using vias.

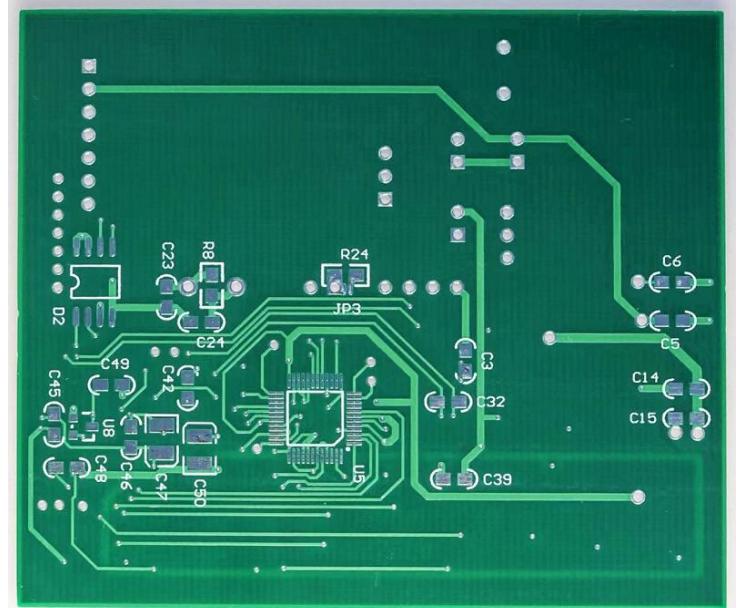


<https://www.google.com/>

4. The PCB needs to be fabricated. There are PCB manufacturers that can do this cheaply. At UCF we use PCBWAY and JCLPCB.

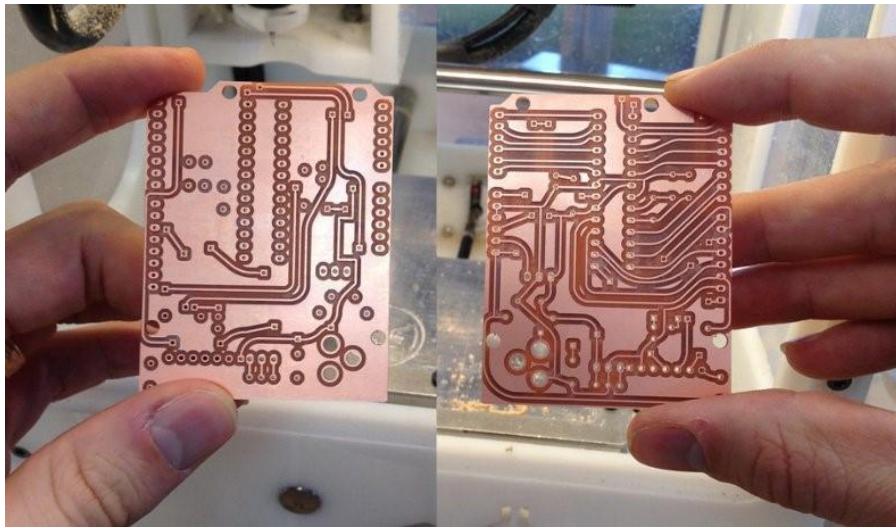


<https://www.pcbelec.com/>



From JCLPCB

- a. Alternatively, we can fabricate the board using a PCB milling Machine. UCF-ECE has one located in the back of senior design. The green mask covering the traces is referred to as a solder mask and the lettering is referred to as a silk screen layer.



<https://www.digikey.com/>

- b. Now that the blank PCB has arrived, we now must put the parts on the board by soldering the components to the PCB pads. The approach we take depends on if the parts are through hole or surface mount.

Through Hole (Classic soldering Method)

We use a soldering station that contains a soldering pencil to melt the solder wire.



Solder Station

Solder Wire

Solder wire comes in leaded or lead free (UCF only allows lead free).



Solder Pencil cleaner

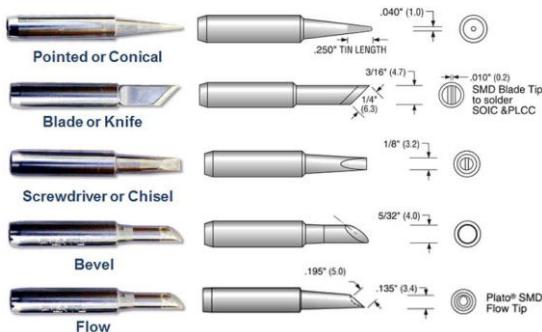
Solder Flux

Water Bottle

Solder flux is used to clean the parts and the PCB pad and to allow the solder to flow easier. The water bottle is used to clean the solder pencil tip.

Example of through hole soldering

“600°- 650°F (316°- 343°C) is a good place to start for lead-based solder and 650°- 700°F (343°- 371°C) for lead-free solder. Hold the tip against both the lead and contact point/pad for a few seconds. The idea is to bring both up to a soldering temperature at the same time.” From the reference below.



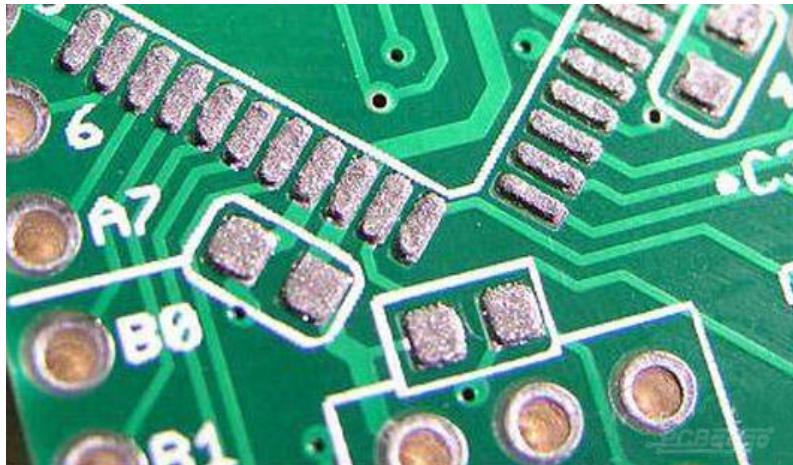
<https://www.techspray.com/>

“At UCF we do not use our soldering tips to bake a cake.” We need to turn the soldering station off when it is not in use.

Surface Mount Soldering

The process of soldering SMD parts is quite a bit different.

In this approach we use a process called reflow soldering. We start with a mixture of solder and solder flux called solder paste. We place this solder paste on the exposed pads of the PCB followed by placing the SMD parts on the board.

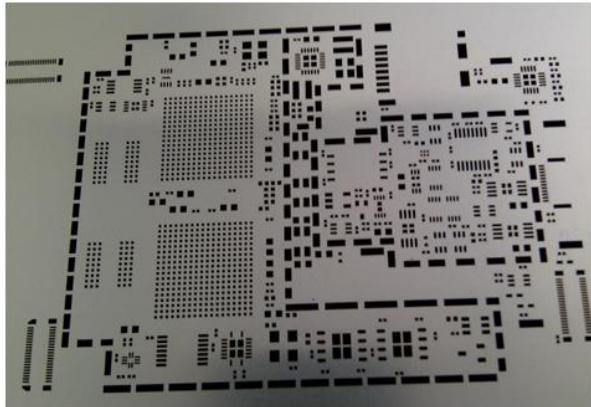


Solder paste is composed of solder balls floating in solder flux. Solder pastes come in either lead free or leaded and with different melting temperatures, 138 degrees C, 183 degrees C or 217 degrees C. In addition, it is available in different size solder balls, defined as T1 - T8 with T3 - T5 as the most popular choices.

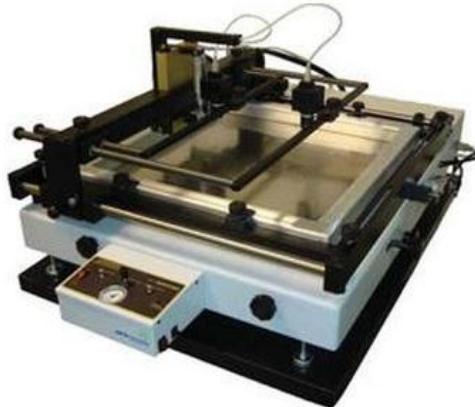
Type designation [IPC]	Particle size in μm (80% min. between)
Type 1	150-75
Type 2	75-45
Type 3	45-25
Type 4	38-20
Type 5	25-10
Type 6	15-5
Type 7	11-2
Type 8	8-2

<https://www.youtube.com/watch?v=uSb1fgCMUug>

There are several ways of placing solder paste on a PCB. One method is to order a solder paste stencil along with the PCB. A solder paste stencil is a stainless-steel plate that has holes in it where the pads are located on the PCB. The solder paste stencil is ordered at the same time as the PCB. The solder paste stencil can be ordered as unframed (the size of the PCB) or framed (about 11" by 17").



<https://www.optimizech.net/>



<https://wwwpcbunlimited.com/>

Solder paste is applied to the PCB using a solder paste stencil and a solder paste printer.

Another method of applying solder paste to the PCB is via a solder paste dispenser. This method works great for repair and rework.

Manual (by hand): UCF has several in Senior and Junior design lab.



Amazon.com



Amazon.com

Manual (Pneumatic): UCF has several in Senior and Junior design lab.

Automatic: <https://www.youtube.com/>

We are now at the stage of placing the components on the PCB. For automatic manufacturing processes components are placed on a PCB using a pick and place machine.



Voltera V-One Four-in-One Desktop PCB Printer

<https://www.youtube.com/>



Neoden YY1 (Just arrived and will be in the back of Senior Design)

<https://neodenusa.com/>

<https://www.youtube.com/>

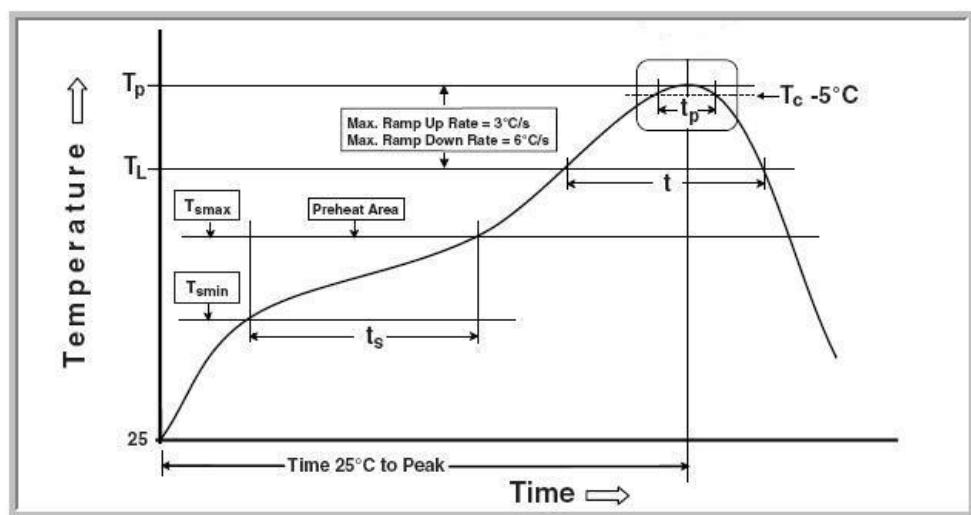
The manual method is to use a set of tweezers and carefully place the components on the PCB.

We finally come to the point of soldering the surface mount components to the PCB.

There are several options here. The first and most popular method is to use a reflow oven to heat the PCB to the melting point of the solder paste. This method is the most accurate in that most reflow ovens can be programmed to the solder profile which is the solder heating temperature vs time and as driven by the of solder paste used. UCF has two of these Neoden reflow ovens.



<https://neodenusa.com/>



<https://www.microsemi.com/>

Another method of soldering the components to the PCB includes using a hot air gun and an IR thermal plate. The thermal plate temperature is set just below the melting point of the solder paste and the hot air gun is used to heat the components the rest of the way melting the solder paste. This approach also works great for rework and repair. UCF has several of these systems in Junior and Senior design.

Rework / Repair



<https://xtronicusa.com/> ||| <https://www.youtube.com/>

Finally, a short discussion on PCB repair and rework of surface mount components. Rework of PCB is similar to soldering SMD components. The main difference to surface mount soldering fabrication is that there is usually an additional step of removing the defective component. UCF is waiting delivery on the Manncorp rework station (shown below).

Another version of an IR rework station (located in senior design). This system also allows the user to enter a solder profile.

After soldering the last step is to clean the PCB of the leftover solder flux. Isopropyl Alcohol (IPA) is used with a brush or a cotton swab to remove this excess flux.



<https://www.youtube.com/>

<https://www.youtube.com/>

GETTING AUTODESK FOR EDUCATIONAL USE

Please follow the steps below to have your Autodesk Education Version.

- 1- Type on Google “Autodesk student and educator access to products.”
- 2- Click the first link as shown in Fig. 1.

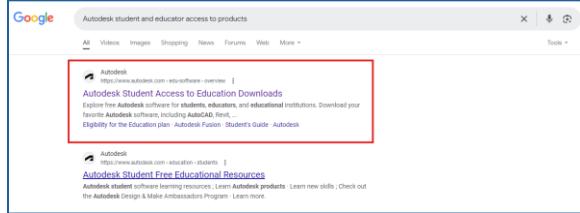


Figure 1: Education access page

- 3- You will be directed to the page as seen in Fig. 2.

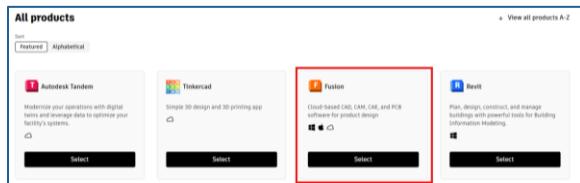


Figure 2: Confirm your eligibility page

- 4- Click “Select” on Fusion window and choose student subscription plan. Enter your information with your knight’s email account and password.

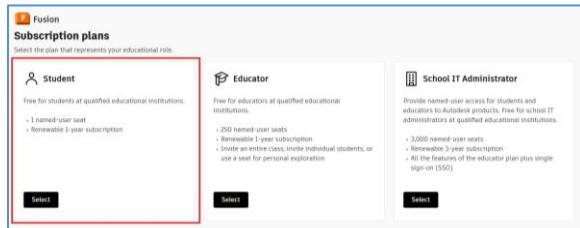


Figure 3: Confirmation of Educational use

5- After signing up, you will need to verify your account from your school email. You need to follow the rest of the step to complete your set up.

- 6- Sign in to Fusion 360 app. If it works, you will see on the top left “Educational License”. See Fig. 4. **Take a screenshot and submit this along with your first assignment.**



Figure 4: Confirmation of Educational use

- 7- If that doesn’t work, please click the link “How it works (3:08 min)” as seen in Fig. 6. Please watch that video and follow those steps.

- 8- If you still do not get the education access, you can click the “GET HELP” as you can see in Fig. 5. Then, on the next page below, you can ask your questions to the virtual assistant 24/7.

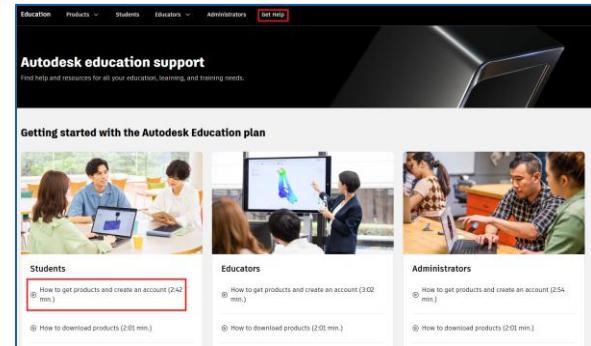
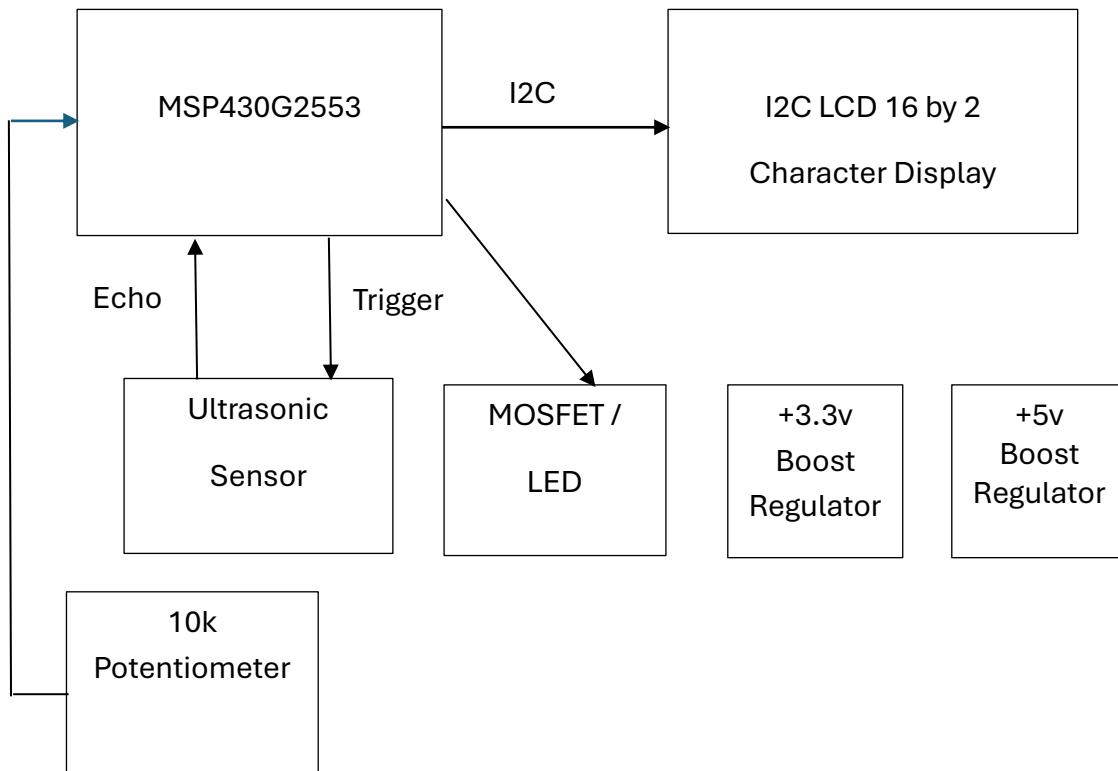


Figure 5: Get Help page

WEEK 1

LABORATORY 1: BILL OF MATERIALS (BOM)

- 1.0. Unclosed is a list of materials that will be required for the junior design semester project (shown below). Read through this list.



The web links given with this BOM is **only a suggestion**. You can use any vendor that you wish.

- 2.0. Create a master BOM list using EXCEL from this list. Generate the following columns: 1. The part name, 2. The part number, 3. the purchasing vendor (where to buy the part), 4. if it is in stock (yes / no), 5. the delivery date, and 6. the single unit or lot price. You don't need to include the optional items.
- 3.0. Generate the total cost for this project.
- 4.0. Upload to Webcourse your spreadsheet with the total price of the project and screenshot of **Autodesk licenses** by the due date.

BOM for the Junior Design Project

1. 1602 display I2C Version Not the Parallel version P1.6 and P1.7 MSP4302553 pins

<https://a.co/d/gNPPdYd>

Hitachi 16 by 02 datasheet

<https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>

Quantity: 1

2. Ultrasonic sensor through hole 5x

<https://a.co/d/ddxt4Ca>

Ultrasonic sensor Datasheet

<https://www.handsontec.com/dataspecs/HC-SR04-Ultrasonic.pdf>

Quantity: 1

3. Bs170 SMD

<https://www.digikey.com/short/tpj3jzqd>

Quantity: 1

4. Msp430g2553 TSSOP 20 pin package

<https://www.digikey.com/short/3ff3v8r5>

Quantity: 1

5. Breadboard

<https://a.co/d/85m3Hpm>

Quantity: 1

Note: This way you can take your project home.

6. Battery holder wire leads

<https://a.co/d/5Q5UIVj>

Quantity: 1

7. PCB on off switch through hole

<https://a.co/d/e83uAqI>

Quantity: 1

8. Momentary push button reset

<https://a.co/d/3Vgb6Ck>

Quantity: 1

9. 22 uF 0805 SMD chip capacitor

<https://www.digikey.com/short/hrbj42ph>

Quantity: 4

10. 4.7uh inductor

<https://www.digikey.com/short/38cnbnh0>

Quantity: 2

11. 5v regulator TPS61322ADBVR SMD

<https://www.digikey.com/short/v0c8mqnf>

5v regulator datasheet

<https://www.ti.com/lit/ds/symlink/tps61322.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1714654712983>

Quantity: 1

12. 3.3v regulator TPS613221ADBVR SMD

<https://www.digikey.com/short/77h58fmh>

3.3v regulator datasheet

<https://www.ti.com/lit/ds/symlink/tps61322.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1714654712983>

Quantity: 1

13. 10k potentiometer through hole

<https://a.co/d/i4HdfZ9>

Quantity: 1

14. 10k resistor SMD 1206

<https://www.digikey.com/short/04tq434q>

Quantity: 2

15. 0.047uf capacitor SMD 1206

<https://www.digikey.com/short/5z8dp8pz>

Quantity: 1

16. 1k SMD1206

<https://www.digikey.com/short/zqzj42h7>

Quantity: 2

17. 510-ohm resistor SMD 1206

<https://www.digikey.com/short/z327qt77>

Quantity: 1

Note: Voltage divider

18. LED any color 0805

<https://a.co/d/fo1PTaP>

Quantity: 2

Note: P1.0 and P1.3 MSP4302553 pins

19. Msp430g2553et development board

<https://www.digikey.com/short/4dpc37br>

Msp430g2553 Datasheet

<https://www.ti.com/lit/ds/symlink/msp430g2553.pdf?ts=1715326392152>

Msp430g2553 user guide

<https://www.ti.com/lit/ug/slau144k/slau144k.pdf?ts=1715320688266>

Quantity: 1

20. Male 2.54mm header

<https://a.co/d/iIQD9pw>

Quantity: 1

Note: 4 pin males for lcd (LCD 4 +5v, GND, SCL, SDA)

4 pin males for programming port (Prog= 5 +5v, GND, SMBCL, SMBIO)

21. 4 pin 2.54mm female header

<https://a.co/d/6XDyIvV>

Quantity: 1

Note: +3.3V regulator 4 pins

+5v regulator 4 pins

Sensor 4 +5v, GND, trigger, echo 4 pins

22. DuPont wire cable (optional)

<https://a.co/d/7hVDqEc>

Quantity: 1

23. AA batteries (4 count)

<https://a.co/d/4ozi2fS>

Quantity: 1

WEEK 2

LABORATORY 2: WEBENCH

- 1.0. Description** – This laboratory project is intended as an introduction to DC regulators using the Texas Instruments WEBENCH online tool.
- 2.0. Goal** – (<https://www.ti.com/tool/WEBENCH-CIRCUIT-DESIGNER>) The goal is to use the TI's WEBench-Circuit-Designer, to design an efficient power supply regulator circuit. We will be using TI's WEBench-power designer tools for DC-DC power designs. The goal of this laboratory exercise is to become familiar with regulator design parameters and to design two regulators (DC-DC power converters) that will be used with two AA batteries and output a regulated 3.3 volts and a regulated 5.0 volts (both needed for the junior design project). From the table below, for AA battery technologies, the input voltage varies for two AA batteries from 2.4 volts to 3.3 volts depending on the battery technology.

Chemistry	Description	Fully Charged	Fully Depleted
Alkaline	Most common non-rechargeable AA battery	1.65 volts	1.4 volts
NiMH	Most common rechargeable AA battery	1.45 volts	1.2 volts

Table 1: AA battery technologies.

The input to the regulator circuit should be in a range of 2.4 volts (from two depleted NiMH batteries in series) to 3.3 volts (from two fully charged alkaline batteries in series). The two regulator circuits should produce outputs of 3.3 volts and 5.0 volts. Since the input voltage is less than the output voltage these types of regulators are referred to as a **Boost Regulator**. Regulators where the input voltage is greater than the output voltage are called **Buck Regulators**. For the junior design project, this semester, we will need a 3.3-volt and a 5.0-volt regulator, each requiring less than 150 milliamperes. The MSP430G2553 will use the 3.3-volt regulator and the ultrasonic sensor, and the LCD display will use the 5.0-volt regulator.

3.0. Procedure –

- 3.1.** Design two efficient low-cost regulator (DC-DC conversion circuits) circuits using the TI WEBENCH Design Center. Go to <https://webench.ti.com/power-designer/> and select efficient, low-cost 3.3-volt and 5.0-volt regulator circuits. Include in your project these two regulators and why you chose them.

3.2.

- 3.2.1.** For the rest of this laboratory project, we will be using the TI - TPS613221ADBVR - 3.3-volt and the TPS613222ADBVR - 5.0-volt regulators. The datasheet for these regulators can be found at https://www.ti.com/lit/ds/symlink/tps61322.pdf?ts=1715929959602&ref_url=https%253A%252F%252Fwww.mouser.ch%252F. Include in your report the BOM, the recommended schematic and the recommended PCB (printed circuit board) layout. Also include the log scale efficiency plots and a table of the operating values in your laboratory report. Given the switching frequencies of today's switching regulators, it is important that the manufacturer's recommended PCB layout be followed otherwise the regulator may not work properly (for example it may not want to start or be unstable).

3.2.2. Fill in the following table below (Table 2) from the operating point table given in WEBench for $V_{in} = 2.4V$, $V_{out} = 3.3V$ and $I_{out} = 150mA$.

3.2.3. For a V_{in} of 2.4 volts, $V_{out} = 3.3 V$ and an output current $I_{out} = 150mA$, what is the efficiency? Calculate the output power P_{out} for the given V_{out} and I_{out} ($P = V \times I$).

$$P_{in} = P_{regulator} + P_{out} \quad (1)$$

and

$$\text{Efficiency} = \frac{P_{out}}{P_{in}} \text{ (As a fraction)} \quad (2)$$

or

$$P_{regulator} = \frac{P_{out} \times (1 - \text{Efficiency})}{\text{Efficiency}} \quad (3)$$

The power dissipated by the regulator is converted to heat resulting in the regulator increasing in temperature.

From Equations (1) – (3) and the efficiency value from the Operating Point Table, find P_{in} , I_{in} , P_{out} (for $V_{out} = 3.3V$ and $I_{out} = 150 mA$) and $P_{regulator}$, V_{in} is given in the Operating Point Table).

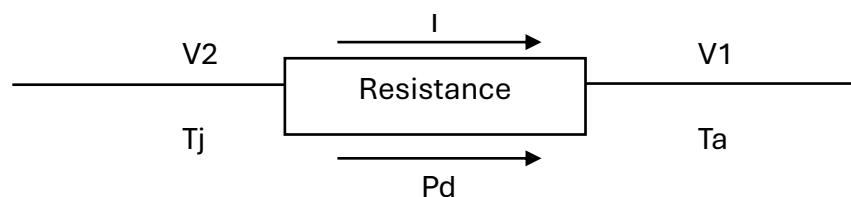
Output Voltage	Efficiency @ 150ma	BOM Cost (Total Parts Cost)	Footprint (PCB Area)	BOM Count (Parts Count)
5.0 VDC				
3.3 VDC				

Table 2: Enter the following information about your DC regulator circuit in the table above.

3.2.4. It is important, for power devices, that the junction temperature (T_j) of the integrated circuit (IC) is below the maximum allowable temperature ($T_j - max$). To calculate this junction temperature T_j , Thermal conductivity is used. The equations used to calculate the junction temperature follow ohms law. The equations used for thermal conductivity are similar to electrical Ohms law, where voltage is replaced by temperature (T), current is replaced by power dissipation (P_d) and resistance in ohms is placed by resistance in degrees C per Watt. The figure below gives a comparison between ohms law and “Thermal ohms law”.

$$\text{Ohms Law } V = I \cdot R \quad R = \text{ohms} = \text{Volts / Amps} \quad (4)$$

$$\text{Thermal Conductivity } T = P_d \cdot R \quad R = \text{Degrees C / Watt} \quad (5)$$



$$\text{Ohms Law } V_2 - V_1 = I \cdot R \quad R = \text{ohms} = \text{Volts / Amps} \quad (6)$$

$$\text{Thermal Conductivity } T_j - T_a = P_d \cdot R \quad R = \text{Degrees C / Watt} \quad (7)$$

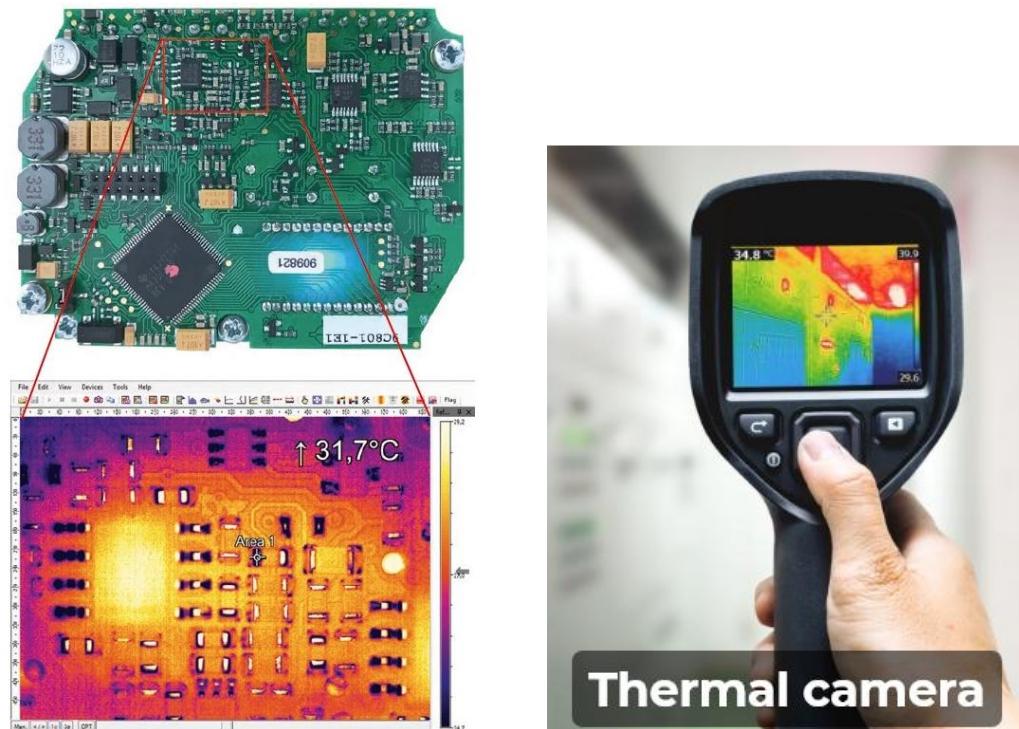
Or

$$\text{Thermal Conductivity } T_j = P_d \cdot R + T_a \quad R = \text{Degrees C / Watt} \quad (8)$$

Just like Ohms law, two thermal resistances in series add like two electrical resistors add in series and two thermal resistance paths in parallel are equivalent to two electrical resistors in parallel.

1. Let T_a be the ambient temperature (assumed to be 28 degrees C), let T_j be defined as the junction temperature, given the thermal resistance R_{ja} from the **operating values table**, using Equations (3) and (8) (with $P_d = P_{\text{regulator}}$), find the junction temperature T_j .
2. What is the maximum junction temperature that is allowed? This value can be found in the datasheet.
3. If the current output is increased to 2.0 amps and using the same thermal resistance and ambient temperature as above, what is the new junction temperature? Will the regulator work for this Junction temperature T_j ?

The best way to measure the thermal temperature of an electronic device is to use a thermal camera to view the temperature of the electronic components.



<https://www.azooptics.com/Article.aspx?ArticleID=1834>

<https://reolink.com/blog/thermal-vs-infrared-camera/>

(only for educational use)

- 3.2.5. Compare your results for I_{in} , P_{out} , Equations (1) - P_{in} , (3) - $P_{\text{regulator}}$ and (8) - T_j with the operating values for $I_{out} = 150 \text{ mA}$. Calculate P_{in} from operating values.

- 3.2.6.** If the inductor peak current hits the peak current limit I_{LIM} , the low-side MOSFET is turned off and stops the further increase of the inductor current. In this case the output voltage drops until power balance between the input side and output side is achieved. Record the values in the datasheet for the minimum, typical, and maximum values for this peak current limit.

3.3. Repeat Step 3.1 for the 5 – volt regulator.

4.0. Project Assignment - What is to be uploaded to Webcourses:

Complete the laboratory assignment by uploading to week 2 Webcourses assignment the WEBench design report: The report should contain 4 sections.

Section 1: Your initially chosen regulators from Step 3.1 and why you chose these regulators.

Section 2: Your 3.3-volt regulator based upon TPS613221ADBVR

- 2.1: Schematic from WEBench
- 2.2: PCB layout from WEBench
- 2.3: The Bill of Materials (BOM)
- 2.4: Operating Point Table from WEBench
- 2.5: Log Efficiency plot versus current
- 2.6: Table 2 from above
- 2.7: $P_{out}, P_{in}, P_{regulator}$
- 2.8: I_{in}
- 2.9: T_j for $I_{out} = 150 \text{ mA}$
- 2.10: Compare your results for steps 2.7 – 2.9 with the Operating Point Table
- 2.11: T_j maximum from the datasheet
- 2.12: T_j for $I_{out} = 2A$, Is the regulator too hot?
- 2.13: I_{LIM} (Peak switch current limit - Min, typ, and max) from the datasheet

Section 3: Your 5.0-volt regulator based upon TPS613222ADBVR

- 3.1: Schematic from WEBench
- 3.2: PCB layout from WEBench
- 3.3: The Bill of Materials (BOM)
- 3.4: Operating Point Table from WEBench
- 3.5: Log Efficiency plot versus current
- 3.6: Table 2 from above
- 3.7: $P_{out}, P_{in}, P_{regulator}$
- 3.8: I_{in}
- 3.9: T_j for $I_{out} = 150 \text{ mA}$
- 3.10: Compare your results for steps 3.7 – 3.9 with the Operating Point Table
- 3.11: T_j maximum from the datasheet
- 3.12: T_j for $I_{out} = 2A$, Is the regulator too hot?
- 3.13: I_{LIM} (Peak switch current limit - Min, typ, and max) from the datasheet

Section 4: A short summary on what you have learned.

5.0. File format allowed: PDF

6.0. Due Date: Tentative and available as Webcourses Assignment.

NOTE: Please read carefully all the steps for this laboratory assignment so not to miss any of the steps.

WEEK 3

LABORATORY 3: REGULATOR PROTOTYPE

- 1.0. Description** – This is the second part of the regulator laboratory where the students will become familiar with boost switching regulators.
- 2.0. Goal** – The goal is to use the two boost regulators chosen from the TI's WEBench-Circuit-Designer (<https://www.ti.com/tool/WEBENCH-CIRCUIT-DESIGNER>) selected in the WEBench part of this laboratory project. For this part of the laboratory project, we will be using the TPS613221ADBVR - 3.3-volt and the TPS613222ADBVR - 5.0-volt regulators. The data sheet for these regulators can be found at: https://www.ti.com/lit/ds/symlink/tps61322.pdf?ts=1715929959602&ref_url=https%253A%252F%252Fwww.mouser.ch%252F.
- 3.0. Background** – Regulators are specialized electronic circuits with the function of producing a constant voltage output independent of current and input voltage. Figure 1 shows the output voltage for an ideal voltage regulator as a function of output current.

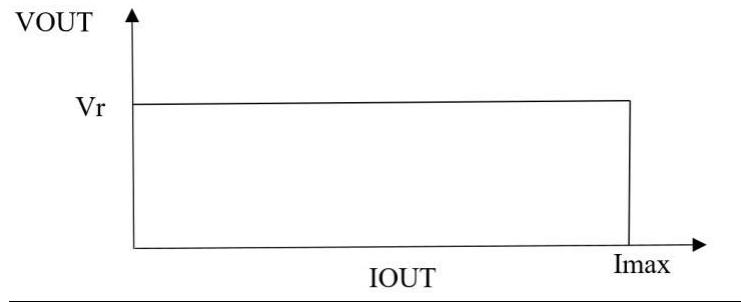


Figure 1: Plot of the output voltage versus output current for an ideal voltage regulator.

Regulators come in two major types: linear where the input voltage is greater than the output voltage and switching regulators. Figure 2 gives an example of a commercially available 3-terminal linear regulator.

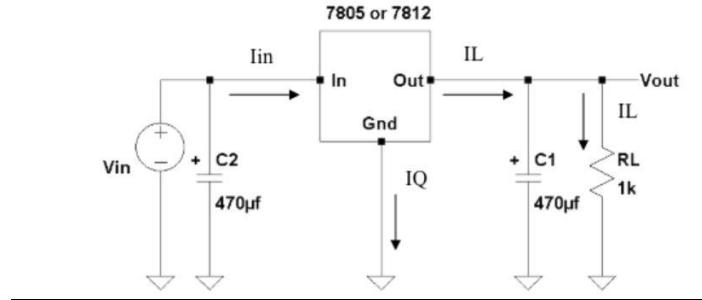


Figure 2: An example of a three terminal linear voltage regulator.

The major difference between a linear regulator and a switching regulator is that a linear regulator operates in the linear region of its circuit design where a switching regulator turns on and off an active electronic component like a MOSFET (acts like a switch) to perform the voltage conversion. Switching regulators are available in three major types, Boost, Buck, and Boost – Buck. For the case where the input voltage is less

than the output voltage these types of regulators are referred to as a **Boost Regulator**. Regulators where the input voltage is greater than the output voltage are called **Buck Regulators**.

So why use a switching regulator over a linear regulator? The efficiency of a regulator which is defined as:

$$\text{Efficiency} = \frac{P_{out}}{P_{in}} \times 100\% \quad (1)$$

and

$$P_{in} = P_{regulator} + P_{out} \quad (2)$$

The difference between the input power and the output power is the power absorbed by the regulator which becomes wasted power and is converted to heat. The efficiency of a linear regulator is:

$$\text{Efficiency} = \frac{V_{out}}{V_{in}} \times 100\% \quad (3)$$

For example, consider a linear regulator with an input of 10 volts and an output regulated voltage of 5 volts. This regulator circuit is only 50% efficient with 50% of the power being wasted in heat. Some of today's switching regulators can be as high as 97% efficient. This is one of the main reasons why switching regulators are used over linear regulators (also because the active electronics components are either full on or full off there is very little loss of power across these devices). Most switching regulators use both an inductor to store energy in a magnetic field and a capacitor to store energy in an electric field.

There are many parameters that are important to regulators and many of these parameters apply to both linear and switching regulators.

$V_{in(max)}$ – The maximum allowed input voltage that can be applied to regulators.

$V_{in(min)}$ - The minimum input voltage that can be applied and maintain regulation.

$I_{out(max)}$ - The maximum output current allowed before the regulator shuts off.

$I_{out(min)}$ - The minimum input current that is required to maintain regulation.

Percent regulation - This is the change in the regulated output voltage as a change in the output current.

Thermal resistance R_{ja} - for a given regulator power this determines how hot the regulator will become - (see the WEBench part of this laboratory assignment).

Maximum Junction Temperature T_j - (see the WEBench part of this laboratory assignment).

Inductor Current - Required to select the proper sized inductor for switching regulators.

Switching Frequency - This is the frequency the active electronic components turn on and off in a switching regulator.

Duty Cycle - This is the percentage of time the active electronic components in a switching regulator are on.

In EEE4309 laboratory assignments the student will study both the linear and the **buck** switching regulator. In this laboratory assignment for Junior design, the student will focus on the **boost** switching regulator. From the datasheet for the TI's TPS613221ADBVR and TPS613222ADBVR boost regulators, the schematic for these regulators is shown in Figure 3. Figure 4 gives the SOT23-5 pinout for these regulators.

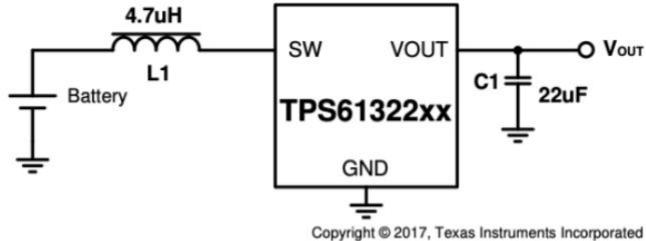


Figure 8-1. Typical Application Circuit without Schottky Diode

Table 8-1. Design Requirements

PARAMETERS	VALUES
Input voltage	0.9 V to 1.6 V
Output voltage	2.2 V
Output current	50 mA
Output voltage ripple	±10 mV

Figure 3: Schematic and Requirements of the TI - 3.3-volt and 5.0-volt boost regulators.

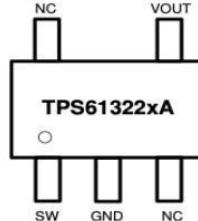


Figure 5-2. DBV Package 5-Pin SOT Top View

Figure 4: P_{out} for the TI's - 3.3-volt and 5.0-volt boost regulators.

Figure 5a gives a schematic of an inductor with an on off switch SW1. At $T = 0$ the switch closes, and the voltage appears across the inductor. Given that the current cannot change instantaneously in an inductor, the current slowly increases. Equation (4) gives the current in the inductor as a function of time. Which is nothing more than the integral of the instantaneous voltage appearing across the inductor ($V_L = U(t)$). Figure 5b shows the current through the 4.7 uH inductor turning the switch on and off at a rate of 1 millisecond with a 50% duty cycle. (this is a linear increase in the inductor current until the switch opens). When the switch opens the current tries to instantaneously go to zero. Given that the voltage across an inductor is related to the derivative of the current as a function of time, a voltage spike appears at the right side of the inductor (The voltage at the switch SW1). This voltage spike is shown in Figure 5c. Take note of the size of this voltage spike (about a million volts – due to the model being ideal $\frac{di}{dt} = \infty$).

$$V_L = \frac{L di}{dt} \quad \text{or} \quad I_L = \frac{1}{L} \int_0^t V_L(\tau) d\tau \quad (4)$$

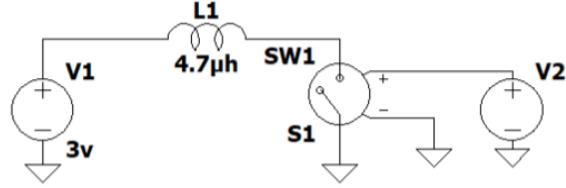


Figure 5a: Schematic of an inductor pulsed at 0.001 seconds.

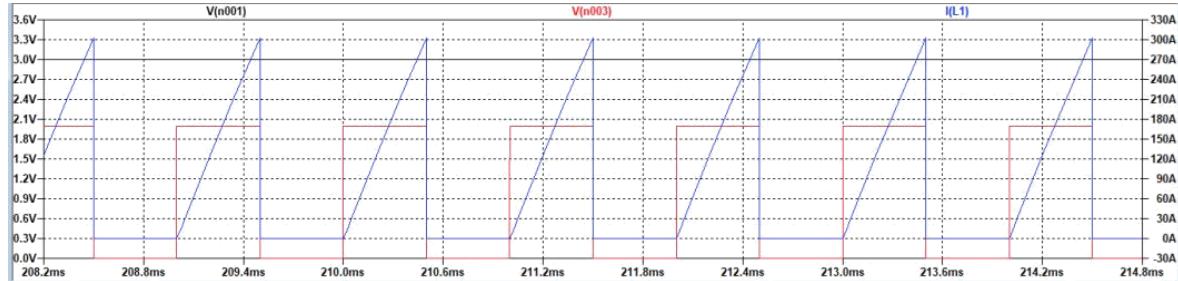


Figure 5b: The Current through the 4.7 uH inductor when SW1 is closed.

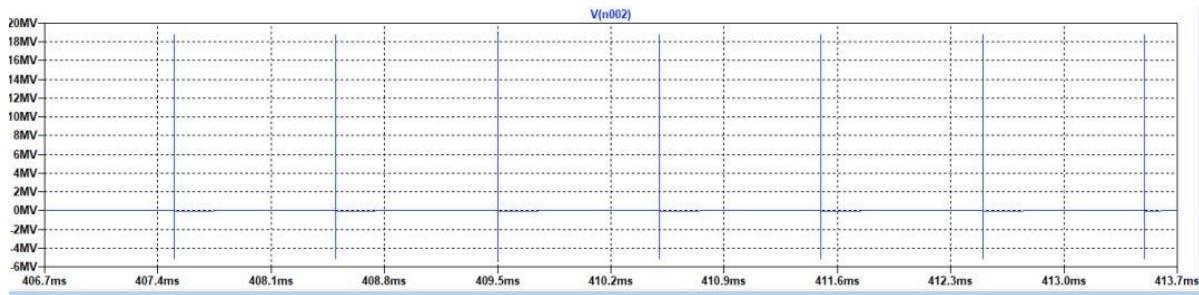


Figure 5c: The voltage when the switch (node at the right side of the inductor) when SW1 opens.

For a second example, let's add to the circuit of Figure 5a a second switch in the form of a diode. A diode acts like another switch when the Anode > Cathode the switch is closed (diode is on) and when the Anode < Cathode the switch is open (diode is off). When the diode is off the current in the inductor increases and we are storing energy in a magnetic field. When the diode is on, we are going to charge the capacitor C transferring the energy from the magnetic field to an electric field. Figure 6 shows the schematic of this circuit. Figure 7 shows the voltage of the capacitor C_1 as it charges during each 1 millisecond cycle.

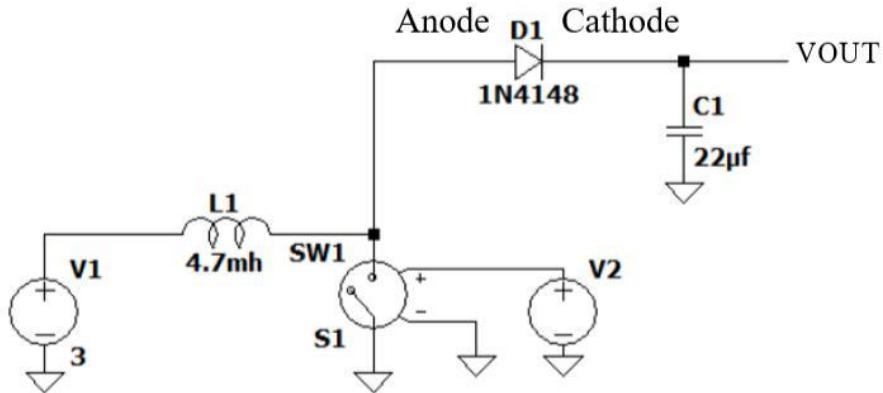


Figure 6: Schematic diagram showing the transfer of energy from the magnetic field to the electric field.

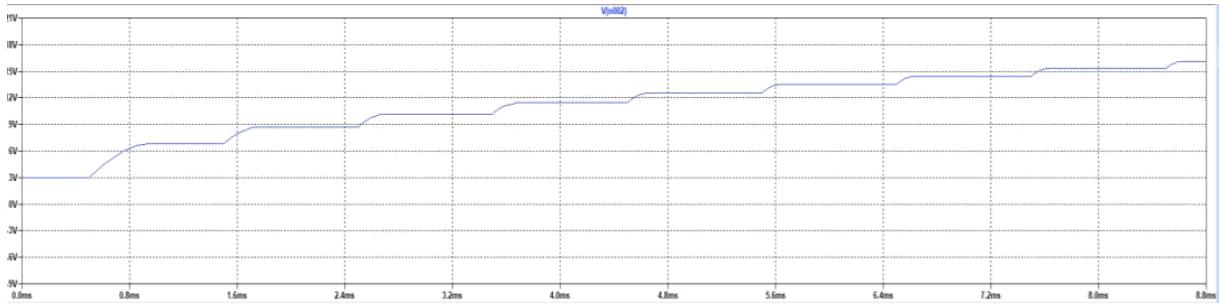


Figure 7: A plot showing the capacitor charging up during each cycle.

A load resistor of 1kohm has been added to the schematic and the simulation time has been increased to 1 second. Figure 8 shows that the output reaches a steady state in about 0.2 seconds with an output of 16.5 volts ($V_{in} < V_{out}$ boost power supply). The small AC signal that is part of this waveform is due to the capacitor charging and then discharging through the resistor when the diode is off. This AC signal is referred to as the voltage ripple. By reducing the switching period so there is less time for the capacitor to discharge through the resistor or increasing the capacitor so there is more energy storage this ripple voltage can be reduced as shown in Figure 9 (C_1 increased to $220mF$)

By decreasing the duty cycle, which results in decreasing the time in which switch SW1 is closed, storing less energy in the magnetic field we can reduce the output voltage as shown in Figure 10 to about 4.4 volts.

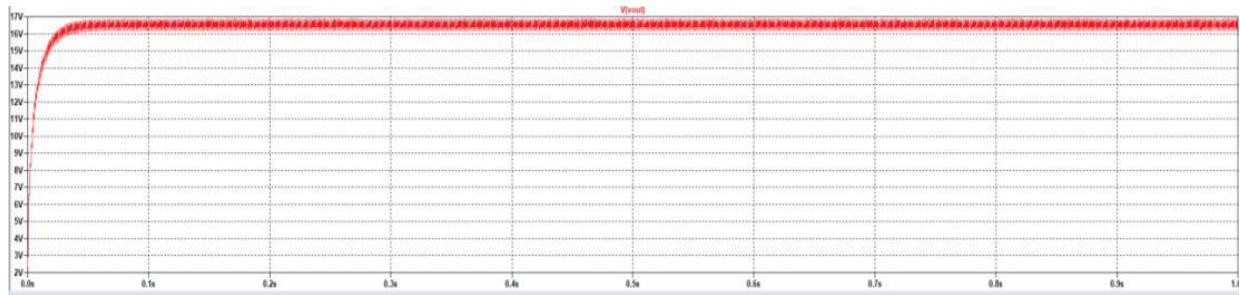


Figure 8: The output of the boost power supply showing $V_{out} = 16.5V > V_{in} = 3V$ (50% duty cycle).

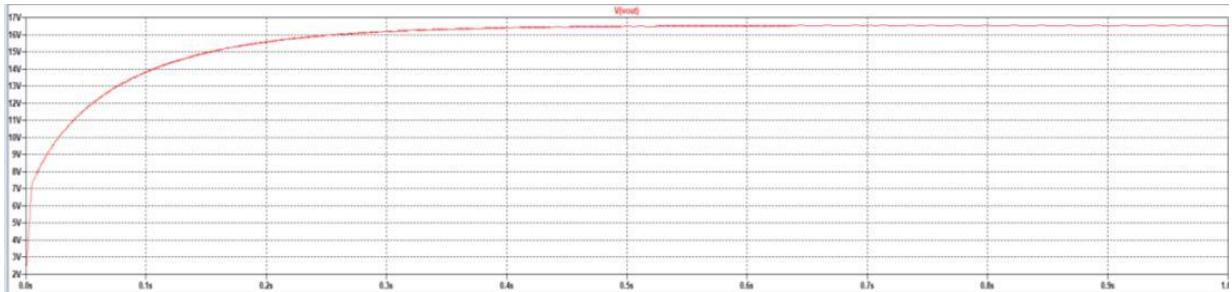


Figure 9: The output of the boost power supply showing $V_{out} = 16.5V > V_{in} = 3V C_1 = 220mF)$

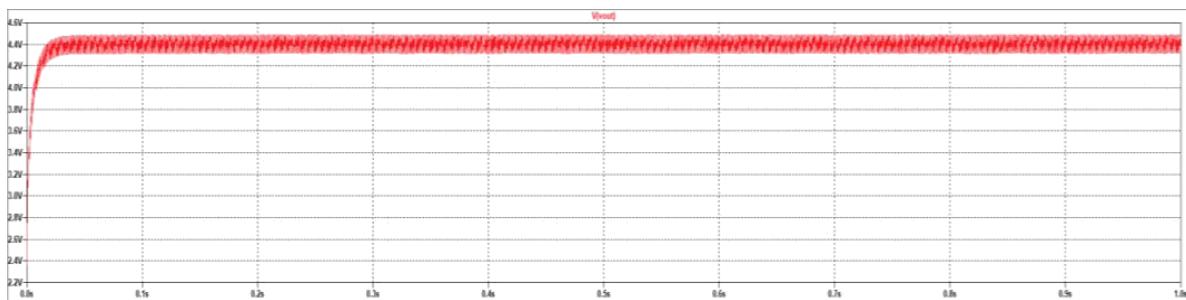


Figure 10: The output of the boost power supply showing $V_{out} = 4V > V_{in} = 3V$ (10% duty cycle).

The only item left to make a boost regulator instead of a boost power supply is a feedback system that changes the duty cycle of switch SW1 based on the measured output. As the voltage output increases this feedback system just simply needs to reduce the duty cycle for switch SW1 being closed (less time on). On the other hand, if the voltage in the output decreases the duty cycle is increased so that switch SW1 stays open longer.

4.0. Procedure (3.3 Volt):

Using the breakout board for the TI - TPS613221ADBVR 3.3volt regulator build the following circuit (Figure 11b). Use 4.7uH for the inductor and 22uF for the capacitor (input capacitor) supplied in the lab. For the output capacitor read the note below. Please take note of the direction and polarity of the regulator on the breakout board (Figures 4 and 11a) and the polarity of the input capacitor.

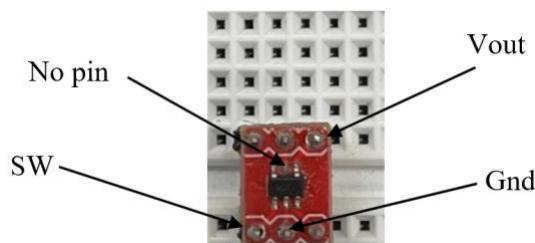


Figure 11a: The orientation of the 3.3V / 5.0V regulators with the breakout board.

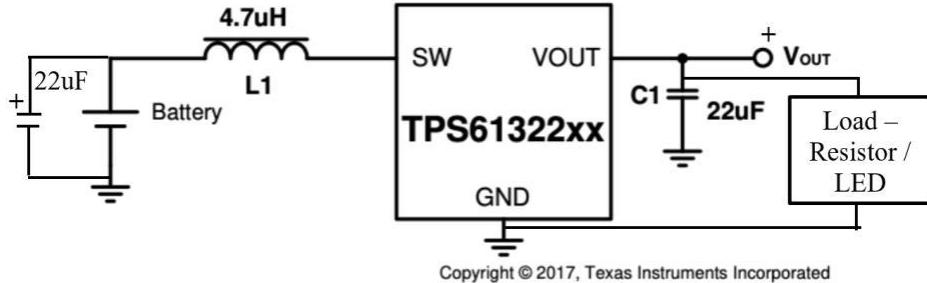


Figure 11b: The schematic diagram for the TI's - TPS61322XX boost regulators.

Note: The 22uF output capacitor is implemented with two 10uF ceramic capacitors. Switching regulators are very sensitive to the type of capacitors that are used. Non-ideal capacitors are composed of the capacitor in series with a resistor (leads) (Effective Series Resistance – ESR) and a series inductance (Effective Series Inductance – ESL). Electrolytic capacitors are known to have a large amount of series inductance and series resistance and make them unsuitable for uses in many switching regulators applications making the regulator control system unstable. Most switching regulators circuits are very sensitive to this series inductance (ESL) requiring the leads to be very short between the output pin of the regulator and the capacitor and well as the ground lead to the capacitor. Here is the recommended layout for the prototype regulator using the SOT23-5 breakout board shown in Figure 12.

As with the selection of the capacitor so is the selection of the inductor, and particularly the maximum inductor current. Like a capacitor which has a maximum operating voltage, inductors have a maximum operating current. From equation (1):

$$P_{in} = \frac{P_{out}}{\text{Efficiency}} \quad (5)$$

and ($P = V \times I$)

$$V_{in} \times I_{in} = \frac{V_{out} \times I_{out}}{\text{Efficiency}} \quad (6)$$

The input current I_{in} which is equal to the inductor current I_L and is given by,

$$I_{in} = \frac{V_{out} \times I_{out}}{V_{in} \times \text{Efficiency}} \quad (7)$$

The actual equation given in the datasheet as

$$I_{in} = I_{Lpeak} = \frac{V_{out} \times I_{out}}{V_{in} \times \text{Efficiency}} + \frac{I_{LH}}{2} \quad (8)$$

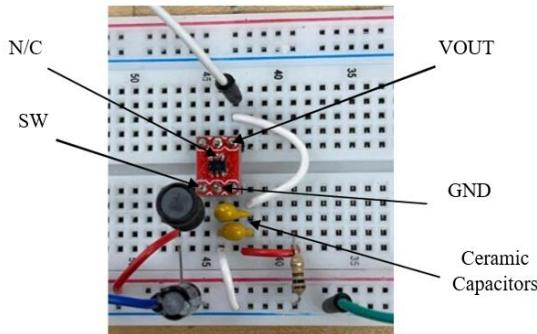


Figure 12: The regulator circuit breadboard showing the placement of the output ceramic capacitors.

- Have your breadboard layout verified with the laboratory TA before applying power to the board.
- The first case uses an LED from the available LED kit supplied with this laboratory assignment as the load for the regulator and a 100-ohm resistor for the LED bias resistor (also available in the laboratory). Measure the ohmic value for this resistor (Record this value). The current that flows through the LED - is given by Equation 9:

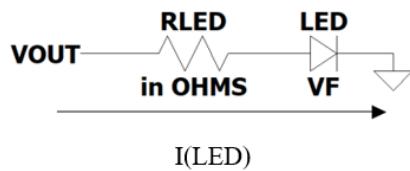


Figure 13: Biasing a LED with a resistor.

- You are going to replace the battery shown in Figure 11a with the laboratory supply. Apply 2.9 volts to the input of the regulator using the laboratory power supply, limit the maximum current to 0.5 amps and measure the output of the regulator V_{out} (Record this value) with the LED connected to the output as the load (Record this value).

$$I_{LED} = \frac{V_{out} - V_F}{R_{LED}} \quad (9)$$

Where V_F is the forward voltage across the LED.

- Measure the forward voltage across the LED. Using Equation 9, calculate the LED current (Record this value).
- Take an image of the regulator with the LED on (Save this image – cell phone camera is OK).

Another method of applying a load to a regulator is to use a piece of test equipment known as an electronic load. An electronic load provides an easy means of selecting the desired resistance or load current. Figure 14 shows an example of an electronic load used as a load for a 5.0-volt regulator with a current load set to 0.15 amps.



Figure 14: An example of an electronic load used as a load for a 5-volt regulator.

- f. Make sure that you shut off the input to the regulator. Add in parallel to the LED and its LED resistor a 100-ohm resistor using electronic load. Select the CR option and adjust to 100-ohm.
- g. Turn on the regulator and the electronic load and measure the regulator output V_{out} (Record this value), check if the value matched with the value shown by the electronic load.



Figure 15: Electronic load setting for a 3.3-volt regulator.

- h. Record the current value shown by the electronic load.
- i. Using the current measured from step h and the LED current, calculate the total current delivered by regulator and the power output from regulator ($P = V \times I$).

$$I_{out} = I_{led} + I_{res} \quad (10)$$

- j. Record the input current I_{in} and voltage V_{in} displayed on the laboratory power supply. Calculate the input power ($P = V \times I$).
- k. Calculate the efficiency from Equation 1 and compare results to that given in the WEBench part of the laboratory.

1. Using an oscilloscope measure the peak-to-peak ripple voltage at the output of the regulator (put the oscilloscope input in AC mode and change the probe's attenuation factor to 1:1) (Save this image – cell phone camera is OK). The attenuation factor settings are located under the probe menu under the channel menu.



Summary of Steps a – l

- a. Verify your breadboard with the laboratory TA.
- b. Measure the ohmic value of the LED resistor.
- c. Measure V_{out} .
- d. Measure V_F and calculate I_{LED} .
- e. Obtain an image with the LED On.
- f. Record the resistance of the electronic load.
- g. Measure V_{out} again.
- h. Record the current through the electronic load.
- i. Calculate I_{out} from Equation 10 and the power output P_{out} .
- j. Measure I_{in} and V_{in} calculate the input power P_{in} .
- k. Calculate efficiency and compare it to the WEBench part of the laboratory.
- l. Using an oscilloscope measure and recording the output ripple peak to peak voltage.

5.0. Procedure (5.0 Volt):

- a. Repeat procedure 4, this time using the 5-volt regulator. Change the two resistors that you use from 100 ohms to 200 ohms (R Load and the bias resistor for the LED). Calculate the new total output current from the 5-volt regulator.
- b. Using the thermal camera available in the laboratory take a thermal image of your 5.0V regulator circuit (Save this image – cell phone camera is OK).

6.0. Procedure:

Return all the parts to the Laboratory TA for reused in subsequent laboratory session.

7.0. Project Assignment: What is to be uploaded to webcourses:

A report containing the data collected in procedures 4 and 5. Include all calculations and equations used.

- a. The results of Procedure 4 steps b - l.

- b. The results of Procedure 5 steps a and b.
- c. Discuss the results you obtained in Procedures 4 and 5.
- d. A short summary of what you have learned.

8.0. Accepted File Format: PDF

9.0. Due Date: Tentative and available as Webcourses Assignment.

NOTE: Please read carefully all the steps for this laboratory assignment so do not miss any of the steps.

WEEK 4&5

RESOURCE: ULTRA LIBRARIAN AND FUSION360 LIBRARIES

- 1.0 Description –** This assignment introduces students to the process of using existing and adding new parts to Fusion’s libraries. In PCBs design, schematics and footprints may exist for common parts and is usually provided by the vendors. Many different PCB software packages exist that can be used to design PCB with each having different file compatibilities, and hence web-based libraries are more commonly used to attain script files for the appropriate PCB design package. Another approach is that libraries can also be created by users with custom parts. However, this process is beyond the scope of this assignment.
- 2.0 Goal –** In this exercise, students will use existing Fusion’s libraries for common parts such as resistors and capacitors. Students will then utilize the web-based tool called Ultra Librarian:

<https://www.ultralibrarian.com/>

The Ultra Librarian tool can be used to import more specific parts in the design of a PCB. Additionally, SnapEDA can also be used to import new libraries, please watch:

https://www.youtube.com/watch?v=1YPOZSJy_QY
<https://www.youtube.com/watch?v=U0eaQQYukCw>

for additional details.

- 3.0 Procedure –**
- 3.1** Starting by looking up (searching) and importing the 3.3 regulator integrated circuit TPS613221ADBVR used in the regulator laboratory assignment. Using a web browser to go to (see Figure 1):

<https://www.ultralibrarian.com/>

(You need to have an account to download the Ultra Librarian file, make sure to create an account before proceeding).

- 3.2** Select the appropriate part that the librarian found. A new webpage will open showing the expected schematic, footprint and 3D model (some applications can process, and display 3D assembled PCB), next select the “Download Now” option and a new window will open (Figure 2).
- 3.3** In the “Select Your CAD Format” area, select the “Autodesk” dropdown, then check the “Fusion360 PCB” option box. You may need to confirm that you are not a robot, then select the “Login to Download” button. You should now have a zipped folder called “ul_TPS613221ADBVR”, unzipped this folder at the saved location. In the folder there will be an Eagle folder with an *.lbr file. This is the part file needed by Fusion.

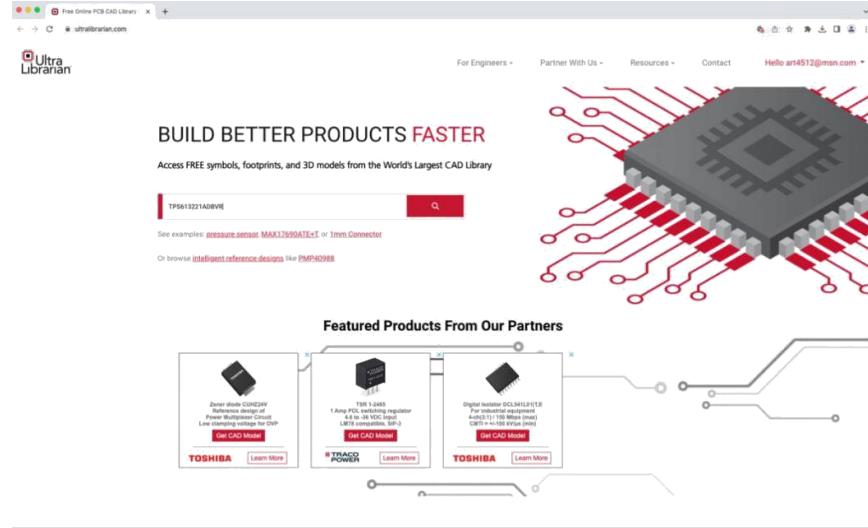


Figure 1: Ultra Librarian Webpage.

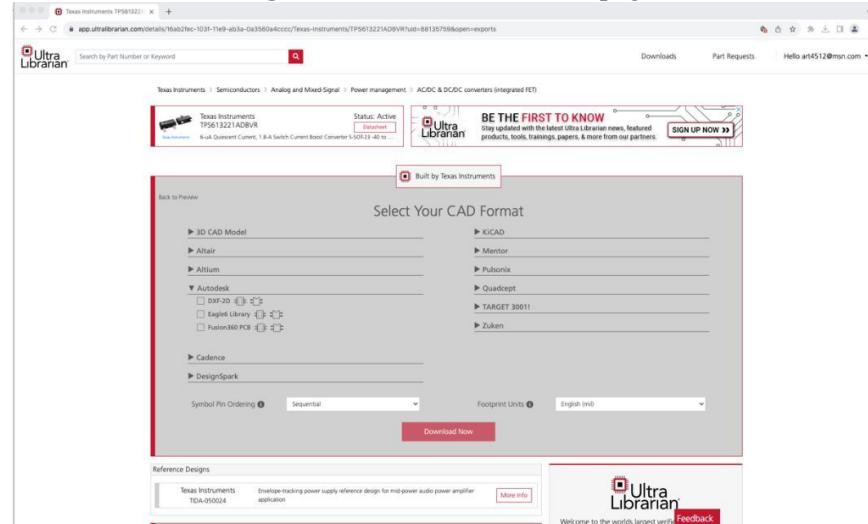


Figure 2: Ultra Librarian Webpage choosing Fusion360 as the output file format.

- 3.4 Go to ultra-librarian and add the rest of your PCB components (capacitor, inductors, and pinout headers) and complete your schematic based on Week 2 laboratory assignment (the regulator assignment).
- 3.5 Repeat with your 5V regulator design using TPS613222ADBVR library.
- 3.6 The steps presented here will be similar in designing your range finder PCB.
- 3.7 Please read and implement the Fusion360 PCB laboratory assignment to add the ultra-librarian parts to your project.

4.0 Project Assignment – See Fusion 360 Assignment.

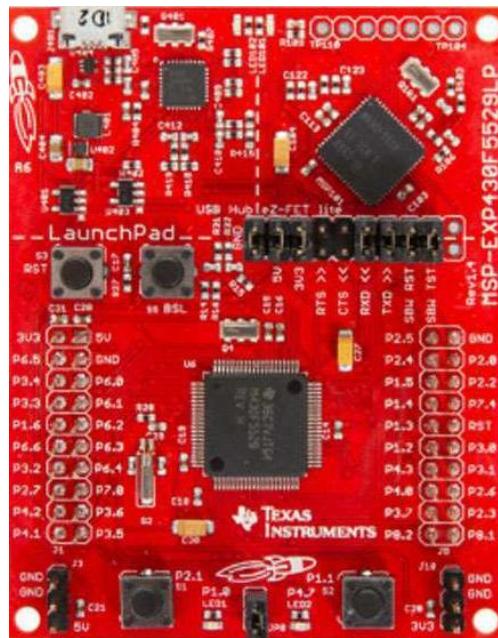
5.0 File Format Allow – *.lbr for components from the Ultra-Librarian.

NOTE: Please read carefully all the steps for this laboratory assignment so not to miss any of the steps.

LABORATORY 4: FUSION 360 PCB DESIGN

1.0: Description – This laboratory assignment introduces the student to the processes used in the fabrication of a printed circuit board (PCB). In addition to learning the PCB design process, the student will become familiar with the process of specifying electronic components.

2.0: Goal – The goal is to use the two boost regulators chosen from the TI's WEBench-Circuit-Designer, (<https://www.ti.com/tool/WEBENCH-CIRCUIT-DESIGNER>) selected in week 2 as the basis for the laboratory assignment (TPS613221ADBVR - 3.3-volt and the TPS613222ADBVR - 5.0-volt regulators, (<https://www.ti.com/lit/ds/symlink/tps61322.pdf?ts=1717528253526>)). The Students will then choose the additional electronic components and using ULTRA librarian download the schematic symbols and the physical footprints for these components. At the completion of this laboratory assignment, the student will be able to draw a schematic and a layout PCB using the FUSION 360 design software. The students will also be able to purchase their PCB from a PCB manufacturer. The image below gives an example of a typical PCB board generated using PCB design software.



Example of a high-density PCB layout MSP430g2553 launchpad G2ET
<https://www.digikey.com/short/rvp9p00z>

3.0: Procedure –

3.1: Start the FUSION 360 program. Login into Fusion to have access to the educational license. Figure 1 shows the startup window.

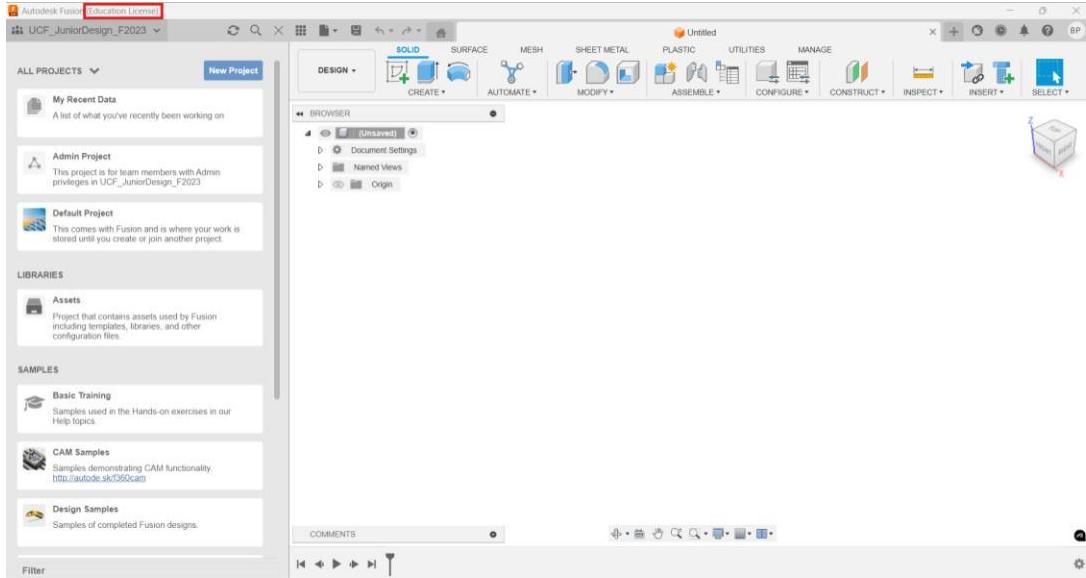


Figure 1: Fusion's startup screen.

3.2: Start a new project in the data panel (on the left of the screen) and name the project Junior_Reg. You can access the projects from Fusion's data panel window (double click) as shown in Figure 2.

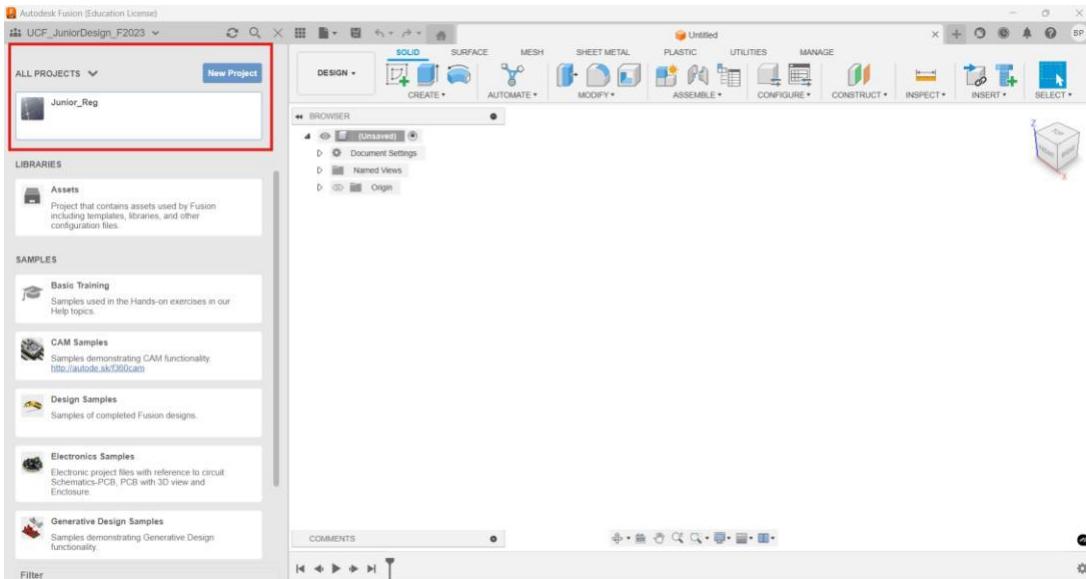


Figure 2: Fusion's control panel screen.

3.3: Open a new electronic design then new schematic as shown in Figure 3. Save this schematic with a new name not untitled.sch.

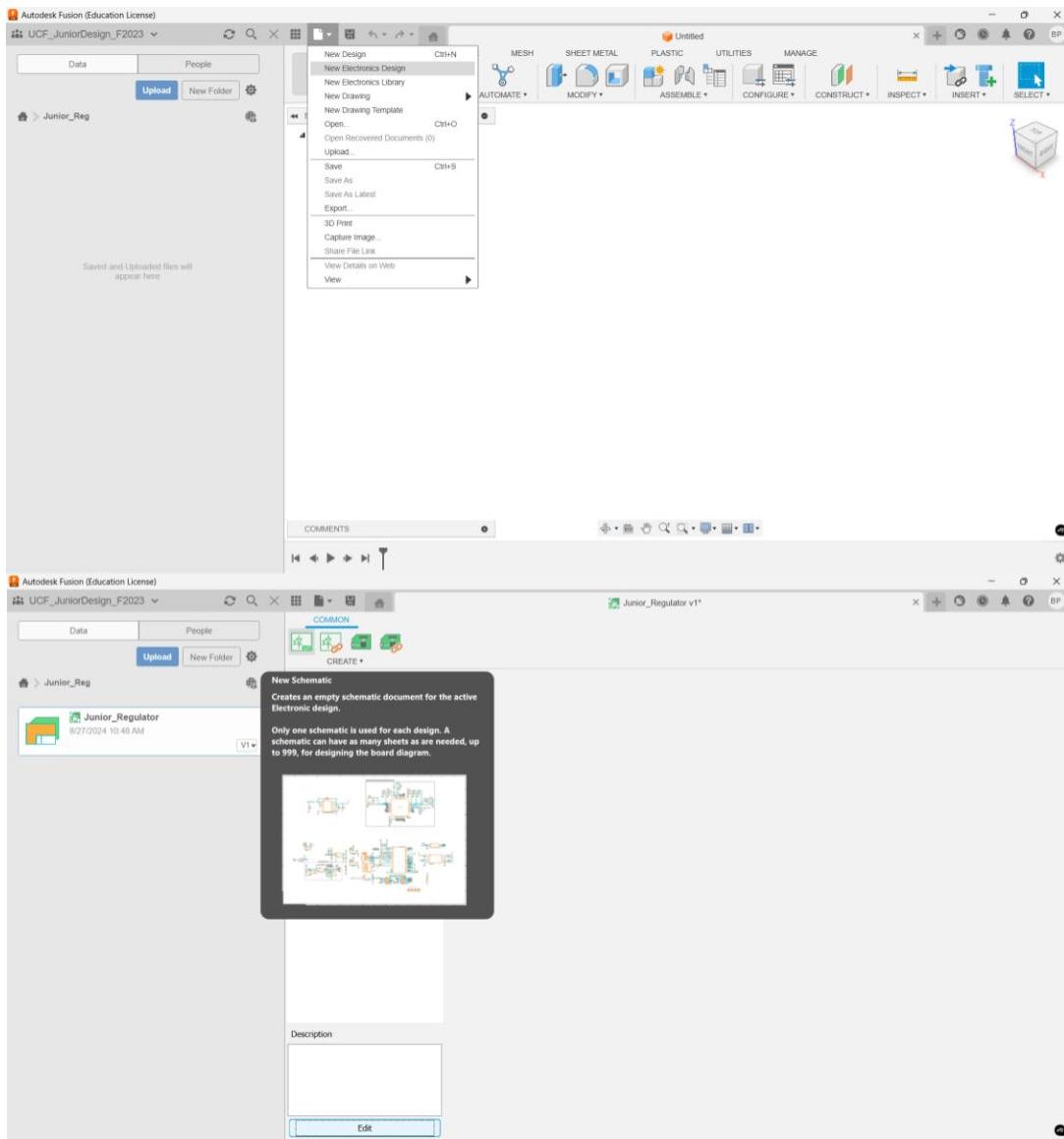


Figure 3: Fusion's new schematic.

3.4: Select the Switch icon in the top left of the menu-bar. This will create a new board layout coupled with the schematic. Save this board layout.

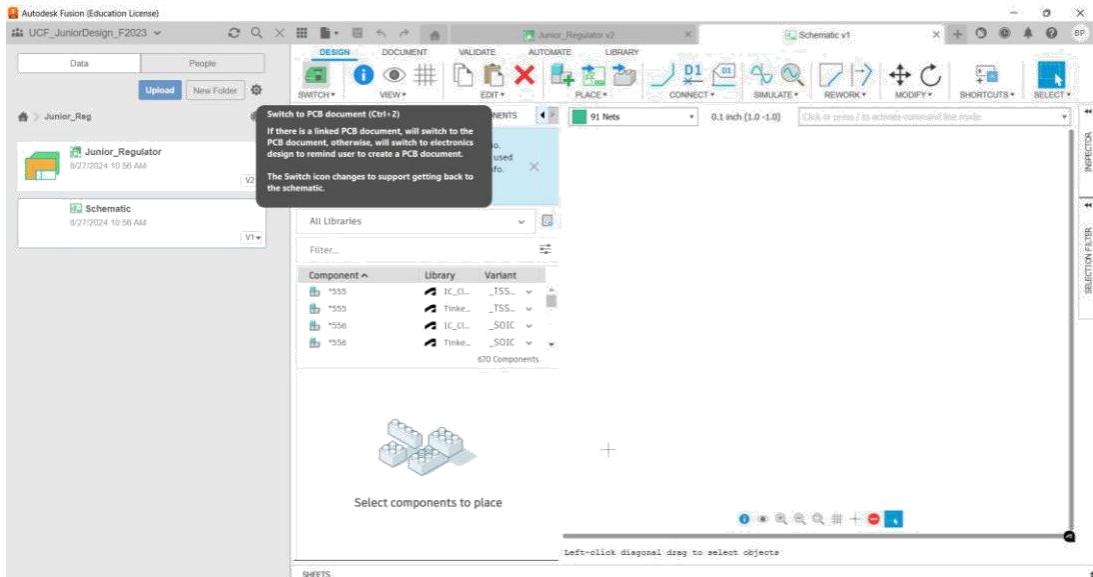


Figure 4: The Fusion's main menu.

You will now have three tabs: The project, the schematic tab, and the PCB tab. These tabs can be accessed through the data panel shown in the top menu to the right.

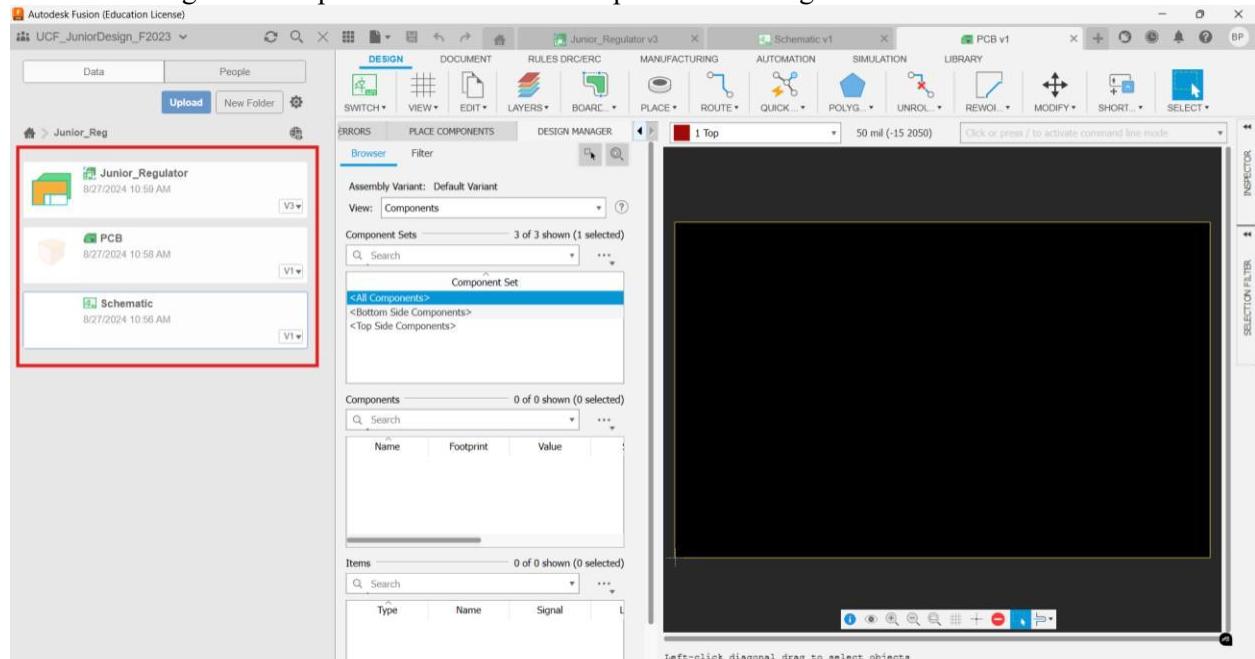


Figure 5: Fusion's board window

3.5: Next, we need to create a parts library so we can import the parts required for this laboratory assignment. Go back to the “Home” window and create a new Electronic Library. There are other ways of importing parts into fusion. One of the preferred methods is to put the parts for this project into a separate library making it much easier to maintain imported parts.

3.6: Save the library with a new name. You should now see your schematic, board, and library files in the control panel window as shown in Figure 6.

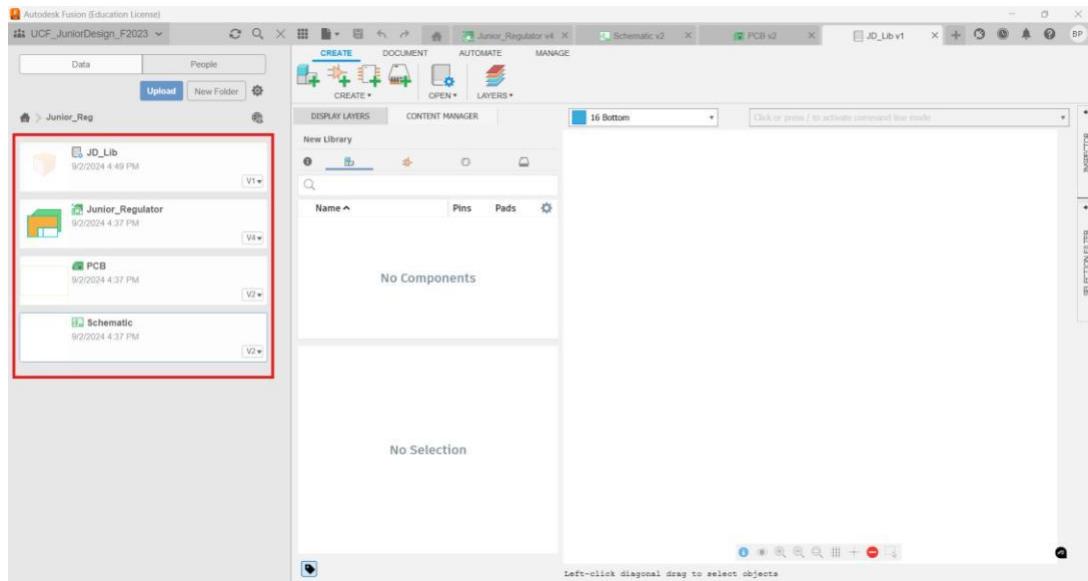


Figure 6: Fusion's Data Panel showing the project's schematic and board.

3.7: We now need to import our parts into our library. These are the parts we need to obtain from the ULTRA librarian.

Input and output capacitors 22uF ceramic:

CL21A226MQQNNNE

Inductor: 4.7 uH 650ma

LQM18PN4R7MFRL

3.3 Volt Regulator

TPS613221ADBVR

5.0 Volt Regulator (Can use the same library as 3.3V regulator)

TPS613222ADBVR

1K 1206 Resistor – used for the bias resistor for the LED:

([Find this on Webcourses](#))

RC1206FR-071KL

LED 0805:

([Find this on Webcourses](#))

150080BS75000

*4 Pin Male Header (0.1" pin spacing):

PINHD-1X04

We will also need a 4 pin 0.4 Pin header with a pin spacing of 0.1" so we can plug the regulator board into the junior design project board (range finder board). We will use the one under **Libraries Manager*** – **Tinkercad - Part: PINHD-1X04**.

3.8: Using the reference documentation go to ULTRA Librarian and download the *.lbr files for these parts (**except 4-pin header and LED**). Do not forget to pick the output file as the Fusion360 PCB under the Autodesk option. Also, you will need to unzip these files and access the *.lbr files.

3.9: Go to the data panel window and double click on your library to open it. Go to Open Library Manager . Then choose Import Library in the upper right tab  then Import from local disk and go to your downloaded *.lbr directory. When all parts are added to the library manager. Go back and add New Component  and choose Import..., then import your part from library manager. Make sure you save the library between each part installed into the library.

3.10: Now your library should contain all parts as shown.

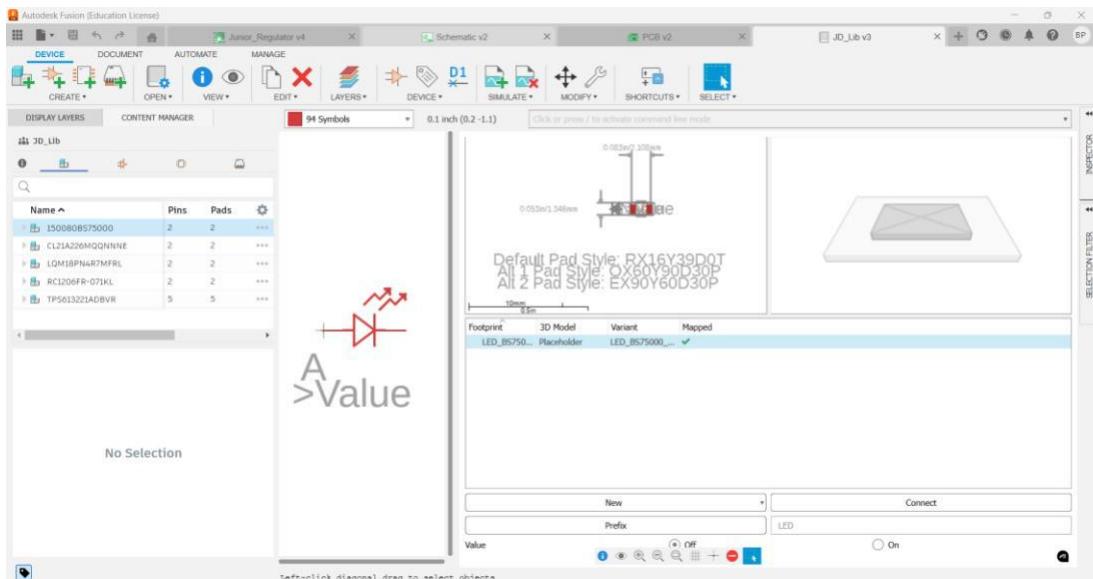


Figure 7: Project library showing parts installed in the library.

The regulator circuits that we will be implementing (for week 2 laboratory assignment) is the one for the 3.3volt Regulator TPS613221ADBVR and the 5.0 Volt Regulator TPS613222ADBVR (located in the datasheet) (Figures 8 and 9). The battery initially will be replaced with the Junior Design laboratory supply but will be used for the Junior Design semester project.

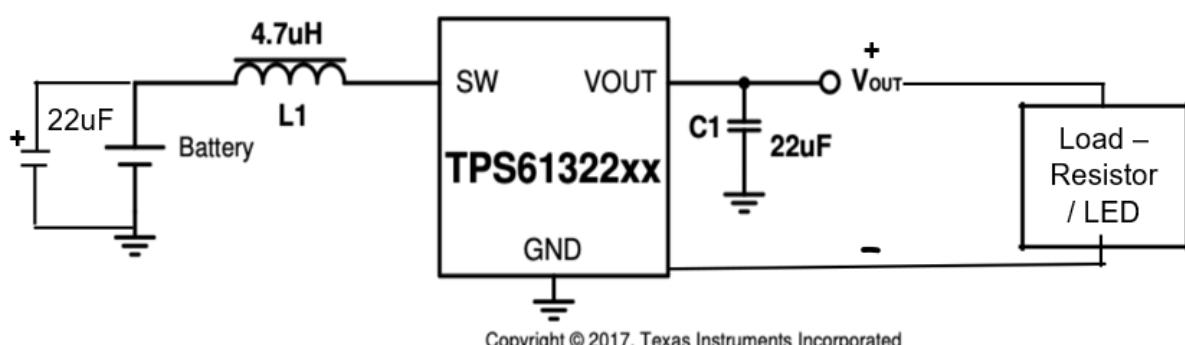


Figure 8: Schematic for the TI's - 3.3-volt and 5.0-volt boost regulators.

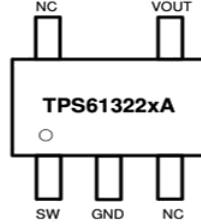


Figure 5-2. DBV Package 5-Pin SOT Top View

Figure 9: Pinout for the TI's - 3.3-volt and 5.0-volt boost regulators.

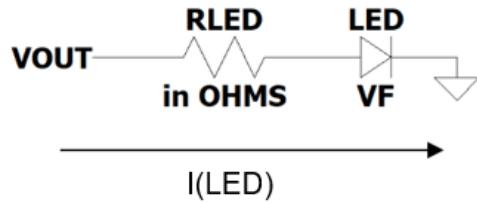


Figure 10: Biasing an LED with a resistor (1k for this project).

$$I(LED) = V_{out} - V_F / R_{LED} \quad (1)$$

3.11: We are now going to add these parts from the library to our schematic. Go to your schematic tab in Data Panel. Then open your library in the right panel.

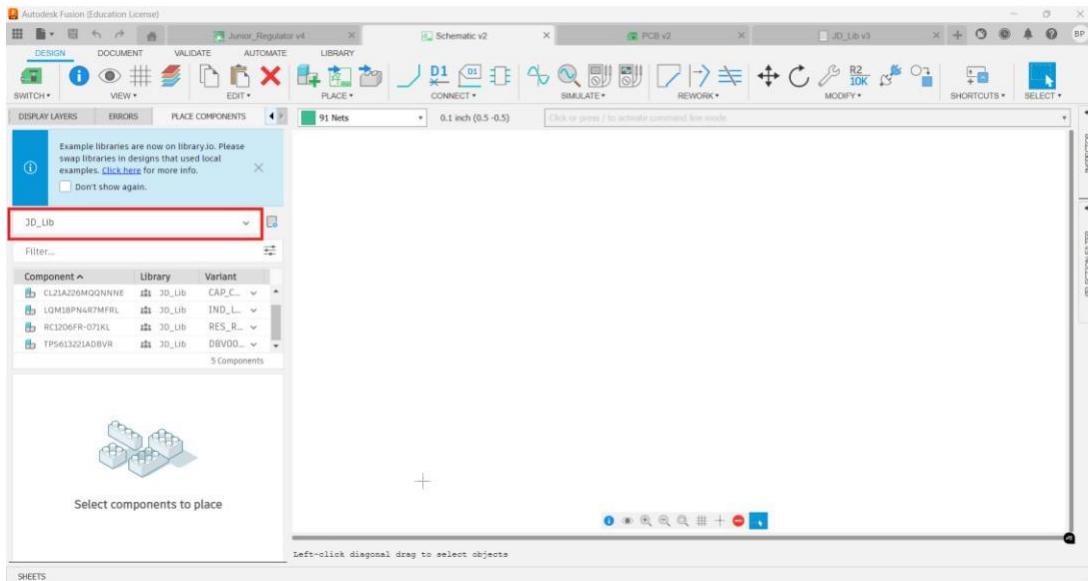


Figure 11: A highlighted part from the project library to be installed into the schematic.

3.12: Choose the first variant on each part: the regulator 3.3 Volt, two 22uF capacitors, 4.7uH inductor, 1k ohm resistor, and the LED. Then double click to add to your schematic. You will be guided to the schematic window to place the part. At this point in time choose any location by

clicking the mouse and then hit the esc key twice to stop adding additional version of this part. We will organize the parts shortly in the schematic. For the four-pin connector, we will use the one under **Tinkercad library - Part: PINHD-1X04**. Add this part to schematic and place it in the schematic. Figure 12 shows all six parts located in the schematic (Figure 12).

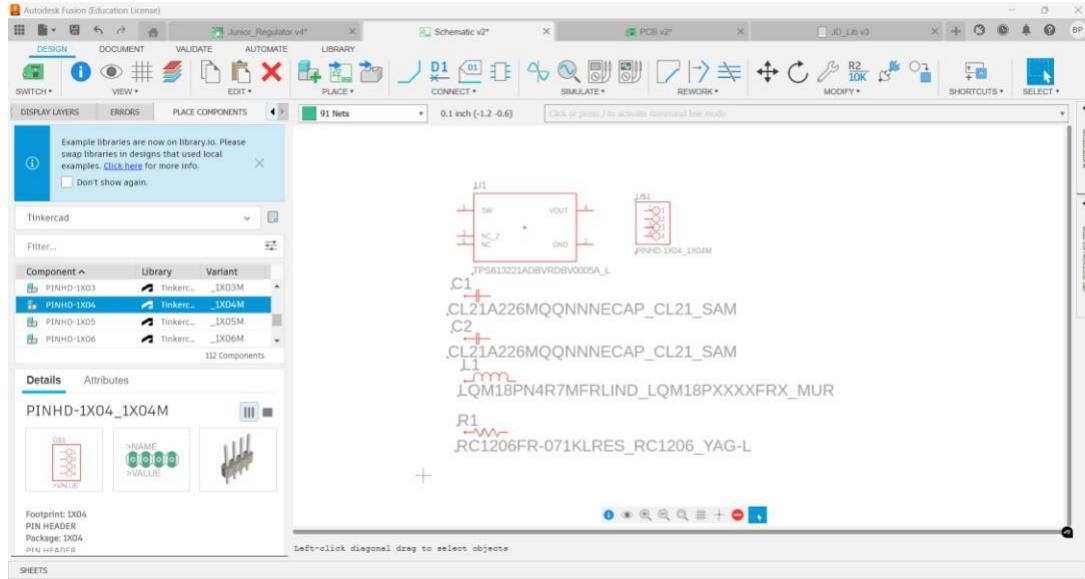


Figure 12: Parts from the project library installed onto the schematic.

3.13: We are now going to draw the schematic using the tools on the left. Each part has a small crosshair in the center that is used to access the part. On the top menu there are zoom in and zoom out icons so you can zoom in or out as required. Figure 13 gives some of the menu tools that you will need to draw your schematic. The trick in using Fusion is to select the tool first and then perform the desired operation. The first tool we will select is the move tool to move the parts to the desired location in the schematic. You will use the mouse to select the crosshair for the park and drag it to the desired location. We will also use the rotate command to orient the parts to the desired orientation. Save your schematic at this point so do not lose what you have completed so far. Do not worry about the parts labels and values we will correct them with the name, value, rotate and move tools.

Note: The Data Panel on the top menu lets us move between our three windows: Control Panel Window, the Schematic window, and the Board Window.

Note: Holding the alt key (option key on the mac) during move / dragging gives a finer movement resolution.



Figure 13: Schematic Menu / tools.

Note: More options can be found in drop down menu and other tabs.

3.14: Select the Value tool and modify the value of each part. Select a part so its value can be changed. A new menu will appear and select to change the part's value. The goal is to remove completely the long part name for each part except for the regulator. Replace each part's value with its actual value (for example 22 uF). The regulator value will be replaced with its part number of TPS61322 1 / 2 for the 3.3V and 5.0V regulators. Next, change to the rotate tool and rotate the part's value and name to the desired orientation. Use the move tool to move the value and name of each part to the desired location in the schematic by selecting value and crosshair (Figure 14).

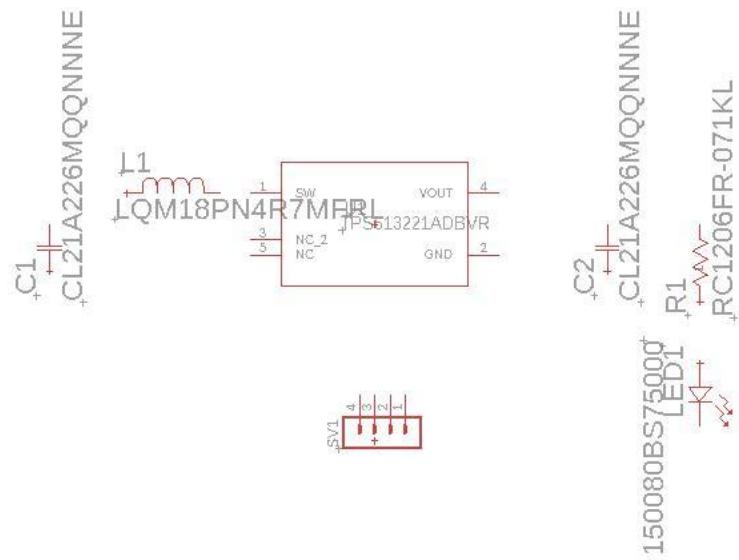


Figure 14: Parts in the schematic moved to the desired location.

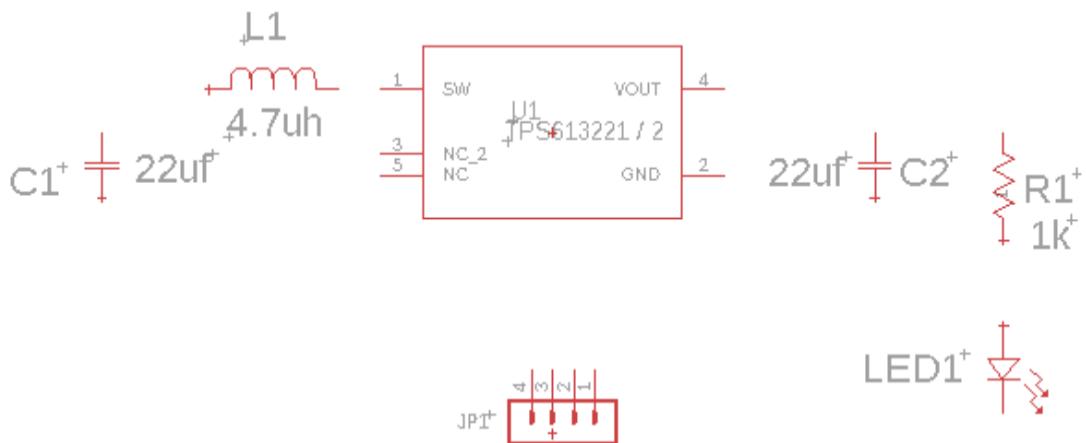


Figure 15: Parts in the schematic - names and values changed as desired.

3.15: Using the wire tool, we are now going to connect the parts as shown in Figure 16. Use the esc key when done placing a wire to move onto the next wire.

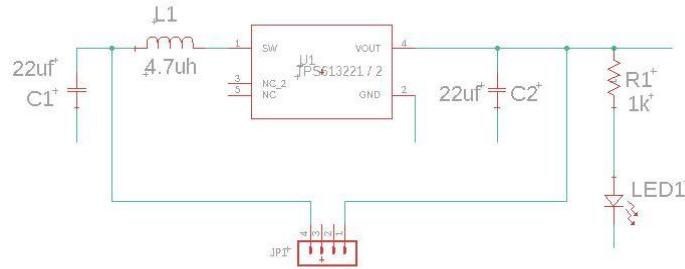


Figure 16: Various parts connected except for the ground connections.

3.16: Next, we are going to use the label tool to add a ground label to the unconnected parts that need to be connected to ground and to label Vin and Vout. Select the label tool and then click on the wire you wish to assign the label to. Use the name tool to rename this label to “GND”, Vin or Vout as appropriate. Add the number of labels as required. You will be prompted to merge the labels with a common node of GND. Answer yes. If desired, you can rotate these labels. Another approach to creating grounds is to use the add part tool and select the ground part (search for GND). Figure 17 shows both labels and the ground symbol used to indicate a ground connection (both methods are connected with the same net).

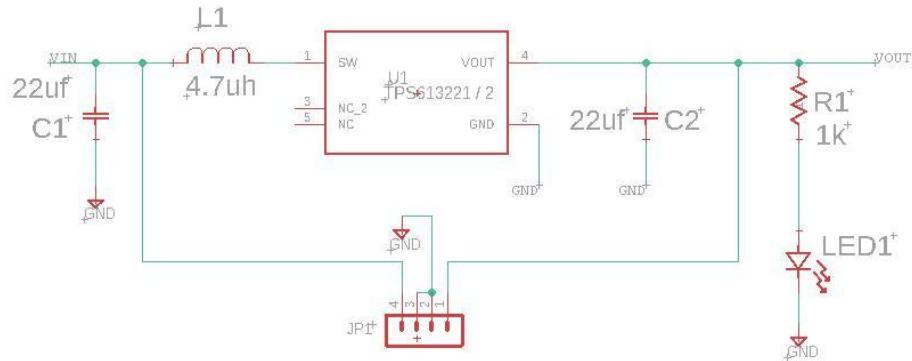


Figure 17: Ground, Vin, and Vout labels and ground symbol added to the final schematic

3.17: Using the Switch icon  to bring up the board window/PCB editor. Zoom out and notice that all our parts are located off board. The wires connecting the parts are known as “air wires” and they are not traces.

3.18: Using the move tool, we must place the parts onto the board. Figure 18 shows one part placed onto the board using the move tool with the other parts yet to be moved. Figure 19 shows all the parts placed and oriented on the board (rotate tool) on the board. The maze of interconnecting air wires is known as a ratsnest (still not traces yet).

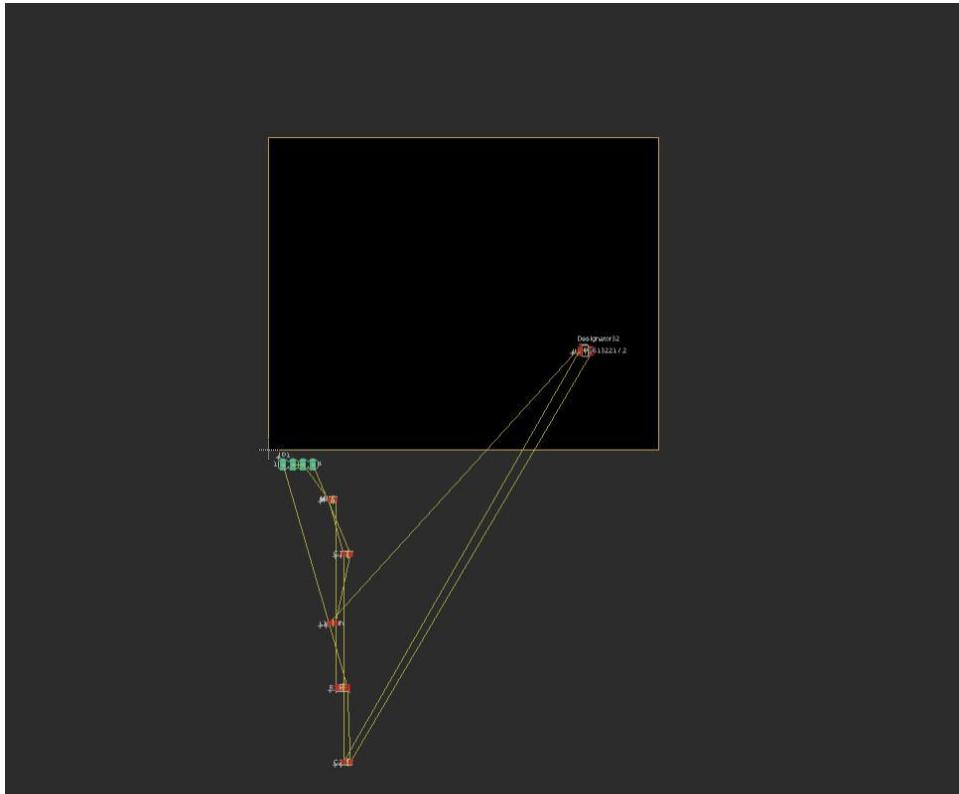


Figure 18: Board window showing a part on the board and other parts yet to be moved.

3.19: Figure 19 gives some of the tools available on the board menu. Using the route manual tool , we will route the traces for each component. Select the route manual tool and in its top menu change the width from **6 mils** default to **20 mils** and for the trace that connect to LED pads, the width should be **16 mils** due to the small pad's footprint. Pick an air wire in the schematic to create its trace. Do not use the auto router tool to route your traces. If you make a mistake, you can use the rip up tool to remove the trace. Figure 21 shows the result of routing the ground air wires with 20 mil traces.



Note: Holding the alt key (option on the mac) and dragging gives a finer movement resolution. More options can be found in drop down menu and other tabs.

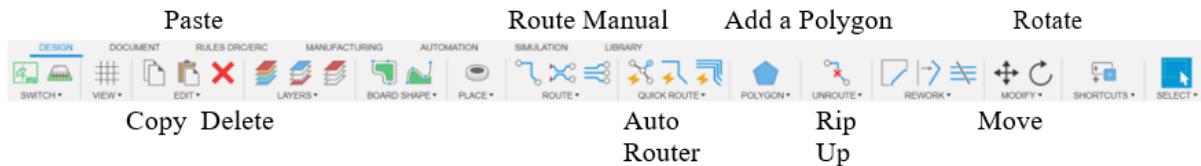


Figure 19: Board Menu / tools.

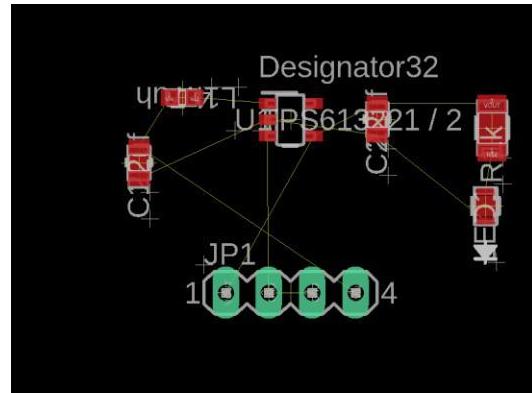


Figure 20: Board window showing the parts placed and rotated on the board.

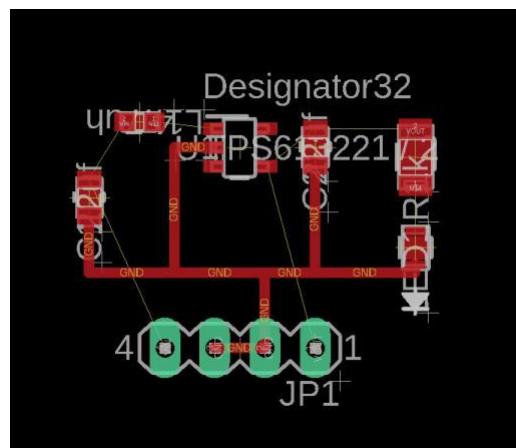


Figure 21: Board window showing the ground air wires manually routed with 20 mil traces.

3.20: At this time, save your board and schematic so not to lose your work.

3.21: Figure 22 shows the board manually traced. Some of the parts had to be moved slightly to make the manual tracing operating easier using the trace air wire tool. Your board should look similar to this board.

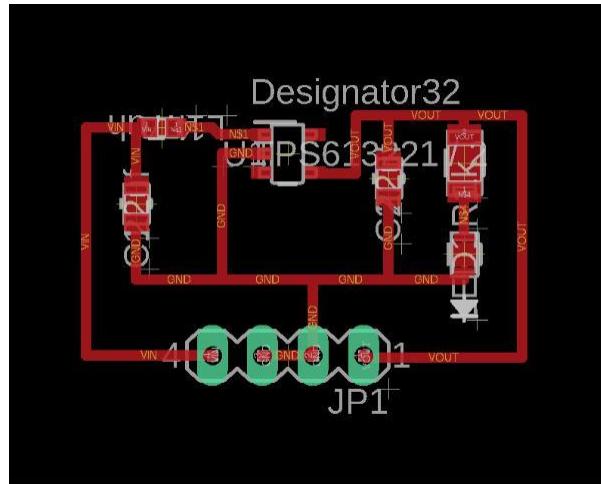


Figure 22: Board window showing the air wires manually routed.

3.22: Using the move and rotate tool you can now move the names and labels, so they are legible. Figure 23 shows the result of this operation (click on the cross hair for each name and label).

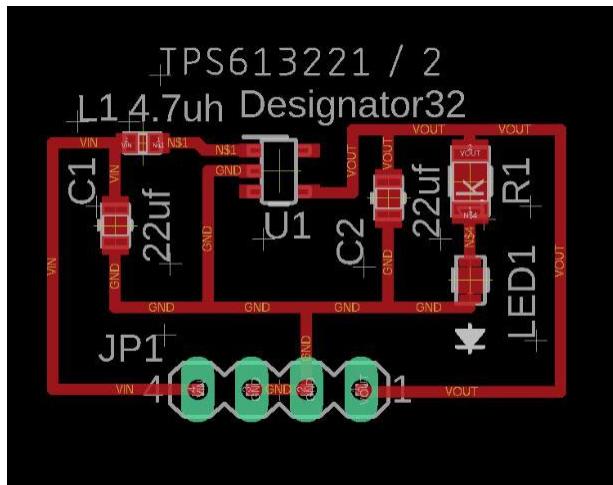


Figure 23: Board window showing final board layout.

3.23: Now that we have the board laid out, we can resize the board as required. Zoom out until the border around the board is present as shown in Figure 24a. Using the move tool highlights the board border and resizes the board to the desired amount (Figure 24b).

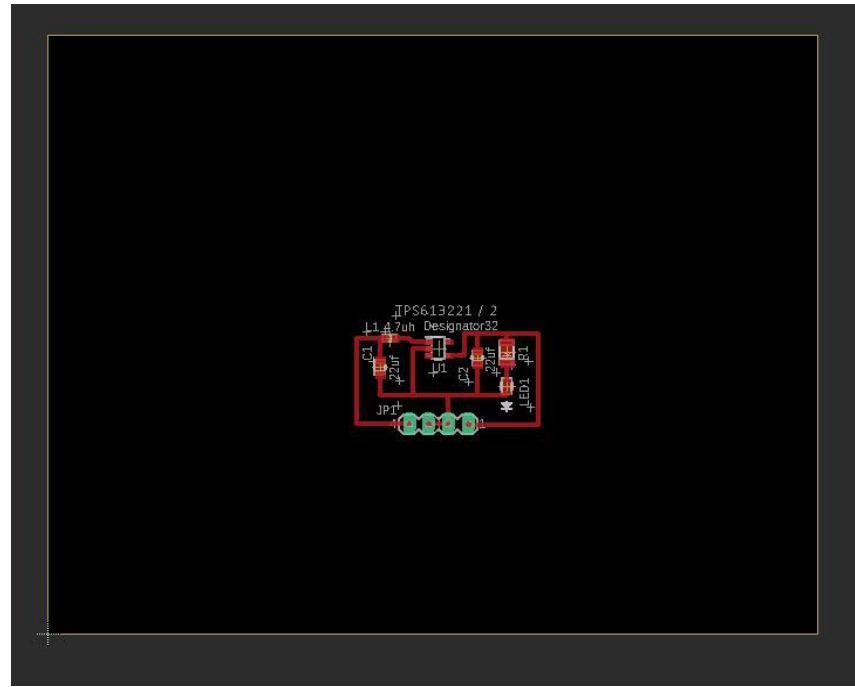


Figure 24a: Board window showing total board area.

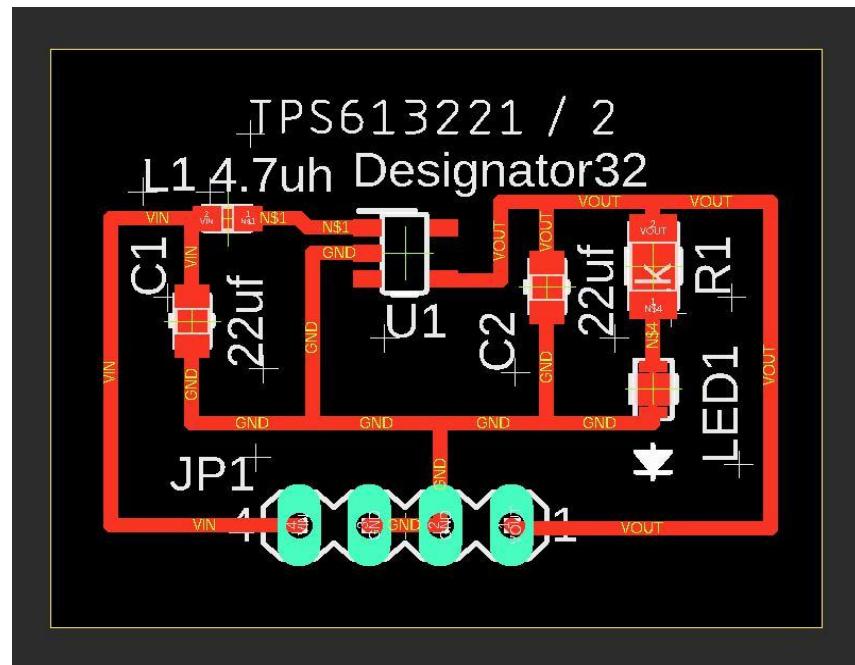


Figure 24b: Board window showing resized board.

3.24: The Display Layers menu item (the menu item on the left tab) gives us a look at the layers associated with this board. Figure 25 shows the various layers. Clicking on the eye icon turns the layers on and off.

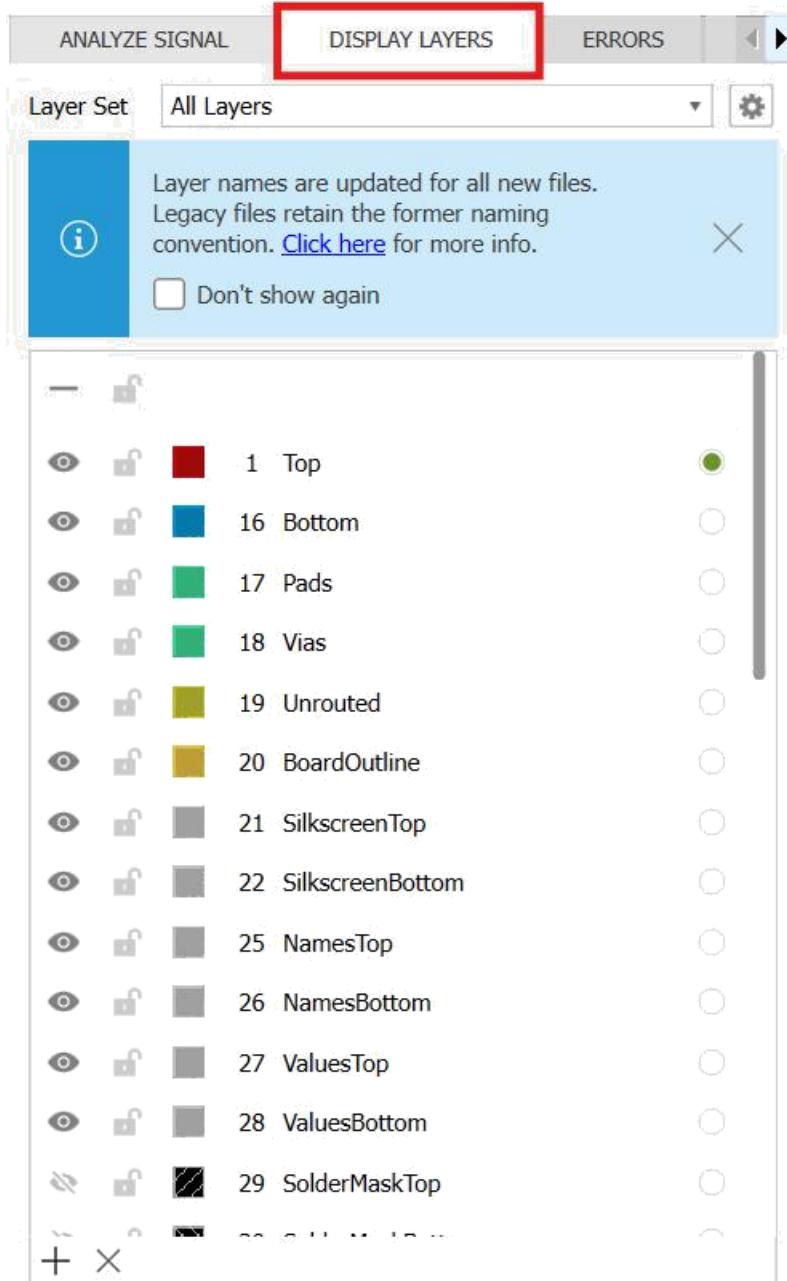


Figure 25: The various layers associated with the regulator board.

3.25: As a final step in our board design, we need to add a ground plane to the board. In the board window select the polygon tool. Next, draw a border around the board and double click the mouse when you have enclosed the entire board (it must be a closed contour). A pop-up window will appear asking for the polygons name. Choose the name “GND”. Using the change tool **Change**, select the isolation option and change it to 20 mils. This will be the spacing between the ground plane and the traces. Finally, select the Polygon Pour (**Dash lines**) and the ground plane will appear.

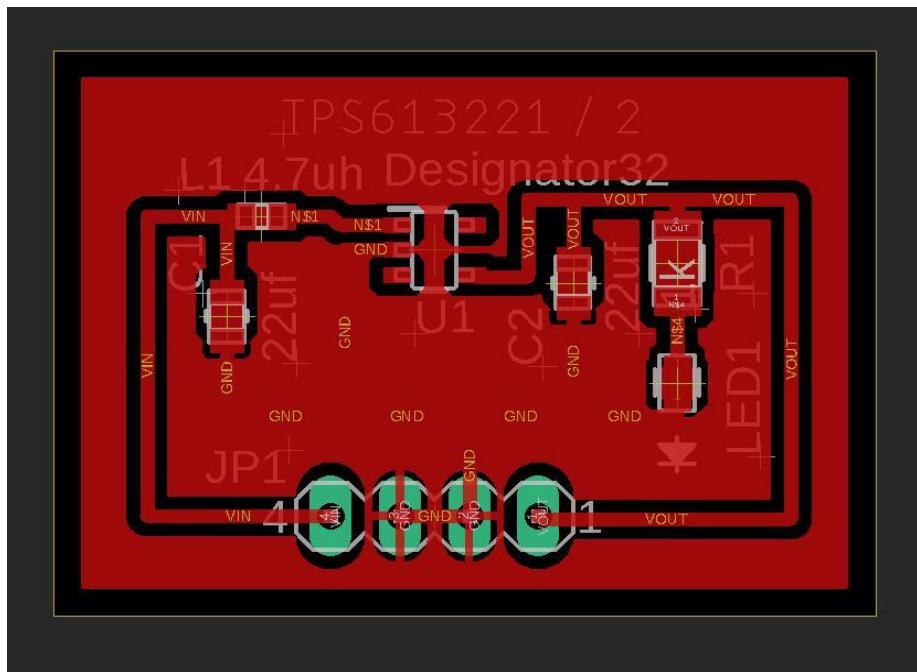


Figure 26: The regulator PCB showing the added ground plane.

3.26: On the top bar there is a Manufacturing tab that allows you to obtain a preview of the board that will be built manufacturer of your choice (Figure 27).

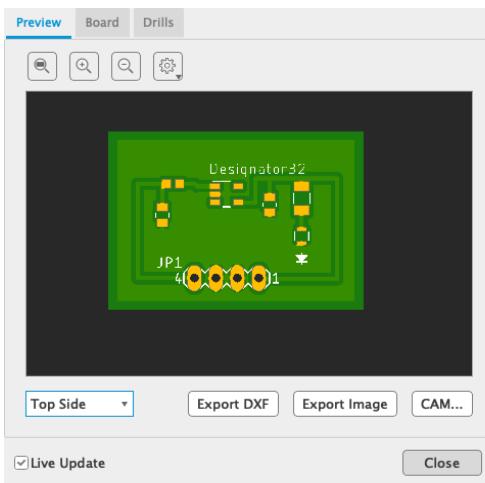
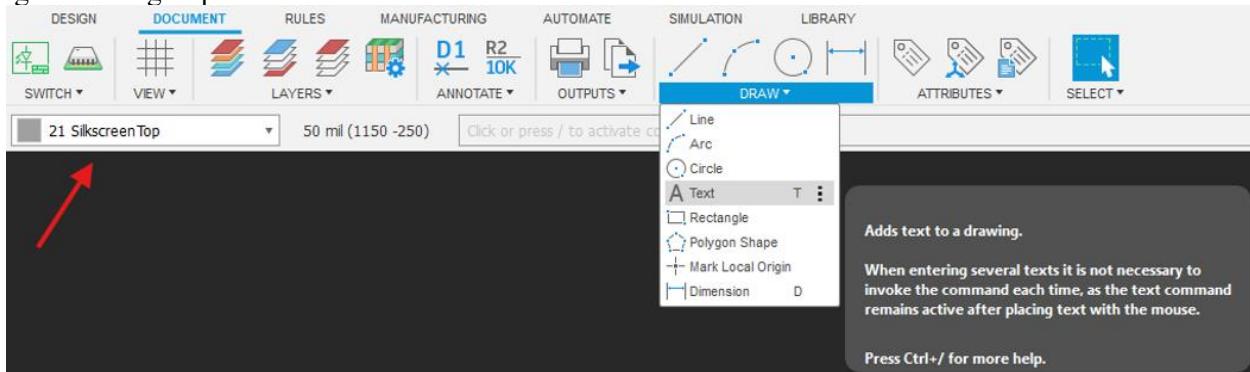


Figure 27: Manufacturer preview window

3.27: To add your parts to the silkscreen layer it takes several steps. First open the layer menu and deselect all the layers except the **ValuesTop** layer. Next, using the Select tool highlight all the components in this layer. The components selected should turn slightly brighter. Using the change tool under the Modify dropdown in Design tab, select the layer dropdown. Select the **SilkscreenTop** layer, select OK and then right click over the board. Fusion will ask you to “**Change: Group**”

click on this window. Your **ValuesTop** layer will be moved to the **SilkscreenTop** layer. This layer is the silk screen layer used by the PCB board manufacturers. At this point, you should see in the manufacturers preview window that your components have been added to the silk screen layer. Finally, turn on all the layers that you have turned off by selecting Preset Standard under your layer set drop down menu.

3.28: Make sure to change the working layer to **21 SilkscreenTop**, otherwise, your text will be on the top copper layer and can create potential shorts. Then switch to Document tab, under draw using **Text Tool** to add your name on to the board. This will help you distinguish between the board when you ordered them together as a group.



3.29: Using the Error tab menu item has Fusion check for any errors in your PCB design. If there are any errors, they need to be corrected before creating your Gerber files so you can order your PCB.

3.30: At this point we are ready to create the Gerber files and the Bill of Material (BOM) for our regulator board. In the board window, from the top menu go to Manufacturing tab select the CAM

processor icon . First select at the option at top of the window “Export as Zip”. Click on the process job button and a file menu will open asking for the location to place the file. The cam processor generates a zip file needed by the PCB manufacturers to build the blank board containing the required Gerber files. Select cancel and answer yes to save the updated CAM file. The bill of material file is in the assembly folder of this zip file. The BOM can also be generated separately from the schematic window by using the Bill of Material output located in the top Document tab. After selecting the Bill of Material a selection menu will pop up allowing us to choose the BOM option. At this point, just choose OK. Another menu window will pop up allowing the user to select the file type. Pick the file type the most appropriate PCB manufacturer (text, csv, or html).

4.0: Project Assignment: What is to be uploaded to Webcourses:

A report containing a summary of the steps in section 3.

- a. Electronic image of your final schematic.
- b. Electronic image of your final board layout.
- c. Included in your report your BOM.
- d. A short summary of what you have learned.

5.0: Accepted File Format: PDF

6.0: Due Date: Tentative and available as Webcourses Assignment.

NOTE: Please read carefully all the steps for this laboratory assignment so not to miss any of the steps.

LABORATORY 5: PICK AND PLACE INSTRUCTIONS

1.0: Description – This laboratory exercise is designed to help students get familiar with the pick and place machine NEODEN YY1, (<https://neodenusa.com/neoden-yy1-pick-place-machine/>). Students will obtain a basic understanding of how to operate the machine and running a python script to convert pick and place export file from fusion to appropriate format.

2.0: Goal – The first objective of this assignment is to align PCB layout to the designated origin to ensure the position of all components stays consistent when place by NEODEN YY1. Once aligned, a csv file is exported from Fusion 360 design containing the location and rotation of all components. Some minor modifications are required for the pre-written script. NOTE: this script is specifically designed for this lab's regulators PCB, additional detail on the file will be provided. The modified csv file is processed by the provided script to generate a pick and place file compatible with NEODEN YY1. Finally, the generated files are transferred to the machine, and students will be guided through the basic steps of how to operate NEODEN YY1.

3.0: Procedure –

3.1: Align PCB to Origin

On Fusion 360, there is an origin point of the design which is demarcated with this symbol:

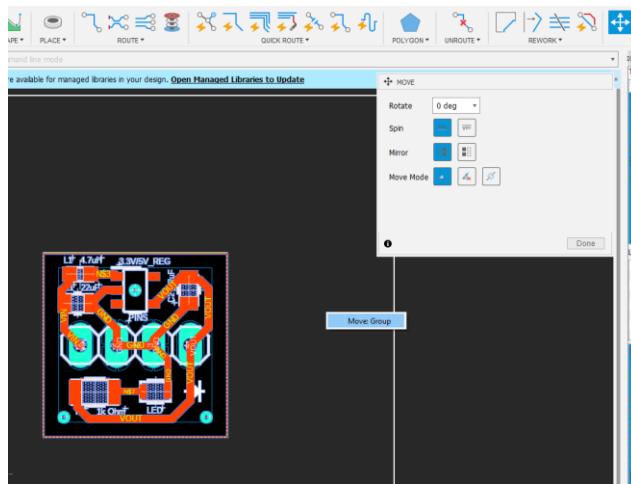


, we need to align our PCBs bottom left corner to this point for usage with the pick and place because this serves as the reference point of the coordinates. You can do this as follows:

- a. Use the select tool to select the entire board

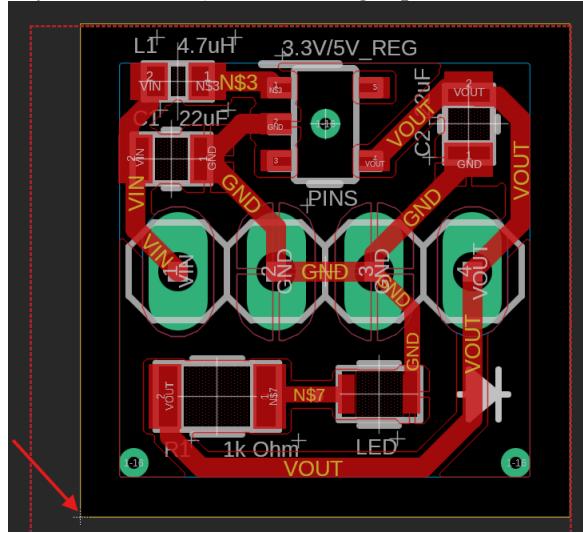


- b. Select the move tool and right click and select the “Move: Group” option which should pop up.



You should now be able to move your mouse and see the design moving with your mouse, the physical board will not move while you are doing this.

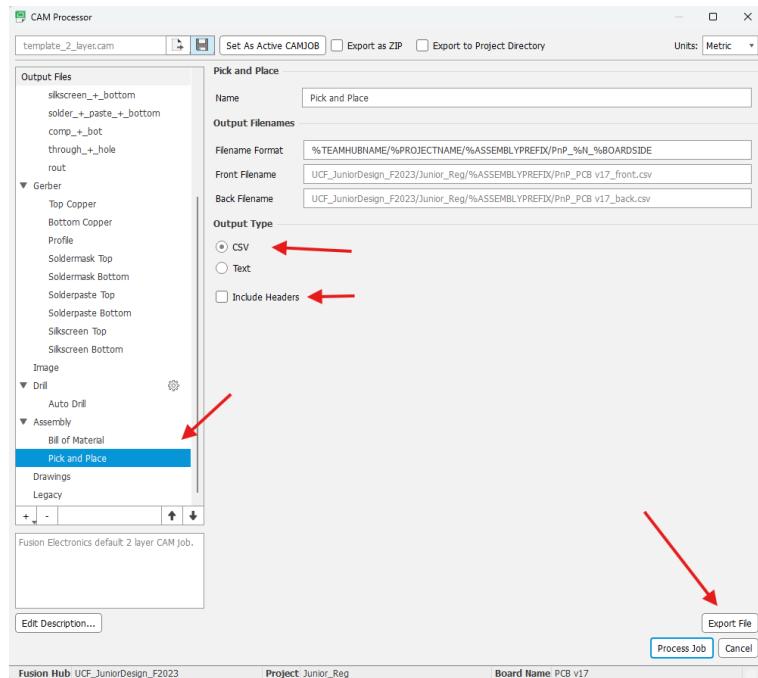
- c. Locate the origin point of your design, you may have to zoom out to find it and align the bottom left corner of your board (the yellow outline) with the origin point and click to place it.



- d. If you have not ordered, carry out the CAM processor job again for a new Gerber file and use this for ordering, if you have already ordered then you do not need to worry about the Gerber file.

3.2: Exporting a Pick and Place File and Formatting it for the Pick and Place Machine

- Save your updated PCB before any of the next steps.
- Under the manufacturing tab->manufacturing, select the CAM processor.
- Under the output files on the left side of the CAM processor, there should be a file called Pick and Place under the Assembly option, select this and under output type, select CSV, make sure “Include Headers” is NOT selected, select “Export File”, and save this file to somewhere you can locate it again.



- d. Open all the folders in the file until you get to the CSV, open the front CSV in Excel, in **column G**, add the appropriate numbers for the components as follows, **do not change anything else (leave other rows blank)**:

U1 (Regulator): 1

C1 (First capacitor): 2

C2 (Second Capacitor): 3

L1 (Inductor): 4

LED1: 5

R1 (Resistor): 6

- e. Rename this file as **PnP_Junior_Reg_front.csv**
- f. Download the **PnPScript** zip file from Webcourses -> Lab Resource, and extract the files.
- g. Copy your **PnP_Junior_Reg_front.csv** into the PnPScript folder that you've extracted, after copying the csv into the folder, you should see these files in the folder:

- h. Run **PnPscript.py** (if you need help with this, send a message to TAs), after running this script you should now see the following files in the folder:

Below is a demo file from the pick and place machine. Our objective is to convert the output csv file into this same format. The numbering is for ease of sorting and matching with our designated reels. The first row is the machine information and can be ignored in most cases. The third row indicated panelized function, which is used to place components on multiple of the same board, that have been panelized by the manufacturer. Unit length and width denote the size of a single board in the panel. Rows and columns denote the number of boards in the panel.

A	B	C	D	E	F	G	H	I	J
1	NEODEN YY1	P&P FILE							
2									
3	Panelized	UnitLength	0	UnitWidth	0	Rows	1	Columns	1

Next, fiducial is a point of reference from a high contrast component to the surrounding (preferably a hole), so that the machine can calculate the offset of each component to the indicated position in the pick and place file. Since each student has a different size board, the script does not include the adjustment of this line, and the fiducial position will be done manually.

5	Fiducial	1-X	13.09	1-Y	55.01	OverallOff	0.06	OverallOff	0.16
---	----------	-----	-------	-----	-------	------------	------	------------	------

Row 7 to 10 are used for changing different nozzles from the designated station. Each component generally has different sizes, and the nozzle used for picking them up will be different. Hence, this function is implemented for board that has a variety of components. NEODEN YY1 can utilize up to 4 nozzles at the same time. For our regulator board design, no more than 2 nozzles are needed, hence this function is left inactive.

7	NozzleChg	OFF	BeforeCon	1	Head1	Drop	Station2	PickUp	Station1
8	NozzleChg	OFF	BeforeCon	2	Head2	Drop	Station3	PickUp	Station2
9	NozzleChg	OFF	BeforeCon	1	Head1	Drop	Station1	PickUp	Station1
10	NozzleChg	OFF	BeforeCon	1	Head1	Drop	Station1	PickUp	Station1
...									

From row 12 and below, information of each component is denoted. Designators are the name of the components, comments usually use for the values of the components, footprint indicated the size of the components (can be modify and have no affect on the actual placement), Mid X is the x coordinate of the component, Mid Y is the y coordinate, rotation are the relative rotation of the components to the set direction of the reel, head is the head use to pick up the components, feeder number denoted the location of the reel, mount speed is the moving speed of the nozzle, pick height is the distance between the nozzle and the component at reel, place height is the distance between the nozzle and the component at the board, mode indicate the active function of the pick head (vacuum and/or camera), and skip indicate if the component should be skip.

12	Designator	Comment	Footprint	Mid X(mm)	Mid Y(mm)	Rotation	Head	FeederNo	Mount Spd	Pick Height	Place Height	Mode	Skip
13	R1	1K	0603D	31.23	17.52	-90	0	1	100	0	0	1	0
14	R2	1K	0603D	29.58	17.6	-90	0	1	100	0	0	1	0
15	R3	1K	0603D	26.03	17.5	-90	0	1	100	0	0	1	0
16	R4	1K	0603D	27.76	17.5	-90	0	1	100	0	0	1	0
17	R5	1K	0603D	27.76	20.72	-90	0	1	100	0	0	1	0
18	R6	1K	0603D	27.76	23.95	-90	0	1	100	0	0	1	0
19	R7	1K	0603D	27.76	27.17	-90	0	1	100	0	0	1	0
20	R8	1K	0603D	27.76	30.4	-90	0	1	100	0	0	1	0
21	R9	1K	0603D	26.03	30.4	-90	0	1	100	0	0	1	0

NOTE: The script is not necessary, and all information can be entered directly into the templates. Additionally, all information can be manually entered when running NEODEN YY1.

3.3:

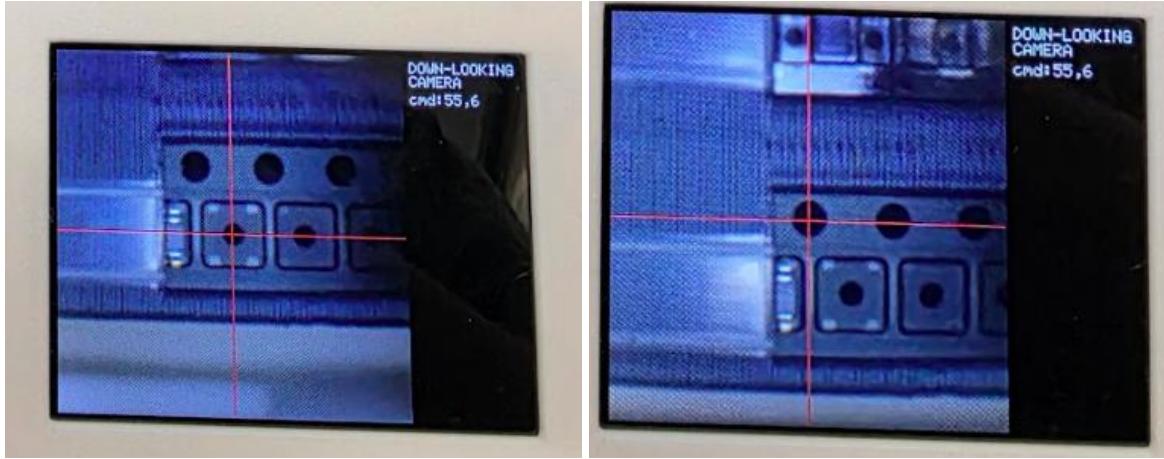
NEODEN YY1 Operation

- First the regulator will be placed at a designated position, such that the corner of the board is aligned with the set origin on the machine. This setting can be changed in system parameters shown below.

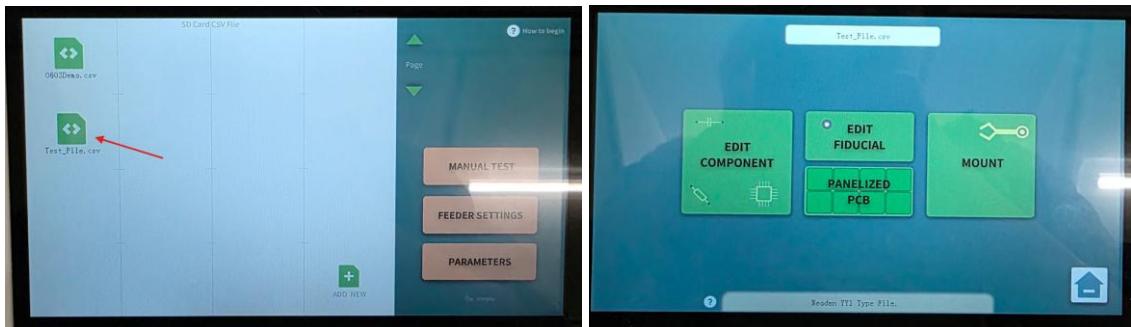


- In feeder setting the alignment of each reel should be calibrated between each use in case of misalignment when the machine is operated. This includes the pick position, where the machine picks up the component; and needle position, where the machine uses a needle to scroll to the next component in the reel. A downward view camera will point to the chosen position. The direction and pitch of each specific part can be set up and each reel can be scroll through.





- c. Then the pick and place file will be chosen and go to edit components to adjust and check if all the components are in the correct position. The cross hair on the camera should be in the middle of the components as shown.



- d. Then as mentioned, the fiducial need to be changed manually. The reference component is usually chosen with the very first component. Then the fiducial normally set with a through hole or even via because of the high contrast between the hole and the surrounding when image processing is taken as shown below.



- e. Lastly, chose the mount option, and the machine will automatically place the components when press play.



4.0: Project Assignment: Upload the **StudentOutputFile.csv** to the Webcourse assignment “Pick and Place files”.

5.0: Accepted File Format: .CSV

NOTE: Please read carefully all the steps for this laboratory assignment so not to miss any of the steps.

WEEK 6

LABORATORY 6: DATASHEET AND RESEARCH

1.0: Description – A crucial skill all engineer needed are research and understanding documentation. In this laboratory, students will perform research based on components in the BOM and the basis understanding of the system. A draft layout of the final layout of the range finder circuit will be designed by the student. Then, a library is implemented using such layout by modifying provided code.

2.0: Goal – The goal is getting students familiar with the researching process to fabricating a system from provided information. In this assignment, the student first needs to access the datasheet of MSP430G2553 and other relevant devices to gain basic understanding of the chip and its pins output and input. There are several areas that need to be considered based on the functionality of the project. The students will design a hand sketch schematic align with said function for later usage in board design.

3.0: Procedure –

3.1: Define the signal associated with each pin using Ultra Librarian and Data Sheets for the components below:

- A. Battery Pack
 - a. Terminal 1:
 - b. Terminal 2:
- B. Switch
 - a. Pin 1:
 - b. Pin 2:
 - c. Pin 3:
- C. 3.3V regulator PCB pinouts (as female pin headers)
 - a. Pin 1:
 - b. Pin 2:
 - c. Pin 3:
 - d. Pin 4:
- D. 5V regulator PCB pinouts (as female pin headers)
 - a. Pin 1:
 - b. Pin 2:
 - c. Pin 3:
 - d. Pin 4:
- E. LCD display (as male pin headers)
 - a. Pin 1:
 - b. Pin 2:
 - c. Pin 3:
 - d. Pin 4:
- F. Ultrasonic Sensor (as female pin headers)
 - a. Pin 1:
 - b. Pin 2:
 - c. Pin 3:
 - d. Pin 4:

G. Potentiometer

- a. Pin 1:
- b. Pin 2:
- c. Pin 3:

H. BS170 MOSFET

- a. Pin 1:
- b. Pin 2:
- c. Pin 3:

3.2: Using the [MSP430G2553's datasheet](#) and user's guide, which can be accessed through the TI website. Locate the power system of the chip, start with input voltage pin and ground pin. Make sure to match the pin with the chip package provided by Ultralibrarian (20 pin chip). Fill out table 1 below.

3.2.1: Find pins for I2C connections to the LCD display, I2C mode includes a SDA and a SCL pin. Fill out table 1, taking note for the coding section and connecting wires to appropriate pins on LCD and MSP430.

3.2.2: For the ultrasonic sensor, there will be 2 GPIO pins connected to the MSP for Echo and Trigger. Select two Port 2 pins for this and fill out your chosen pins in table 1. The trigger line can be directly connected to MSP chip as only confirming signal needs to be sent back and forth. However, there is a voltage divider for the echo line as you can find in its datasheet. We will use a 510 Ohm and 1k Ohm for the division. Make note of this for later.

3.2.3: An ADC connection is used for dimming one LED. First, a voltage value needs to be read from the potentiometer and taken as an input to one of MSP ADC port. Fill out table 1 (**Only use Port 1 for your selection and choose the correct port**). Read through ADC section in the datasheet to gain a better understanding of the process. A label of INCH_x is set to your chosen channel (replace x with your chosen port) take note for coding section). Then an output port is chosen to output ADC signal to an LED make sure there is no conflict with previously chosen port and remember to bias the LED with a 10k Ohm resistor. (**Only use Port 1 for your selection**).

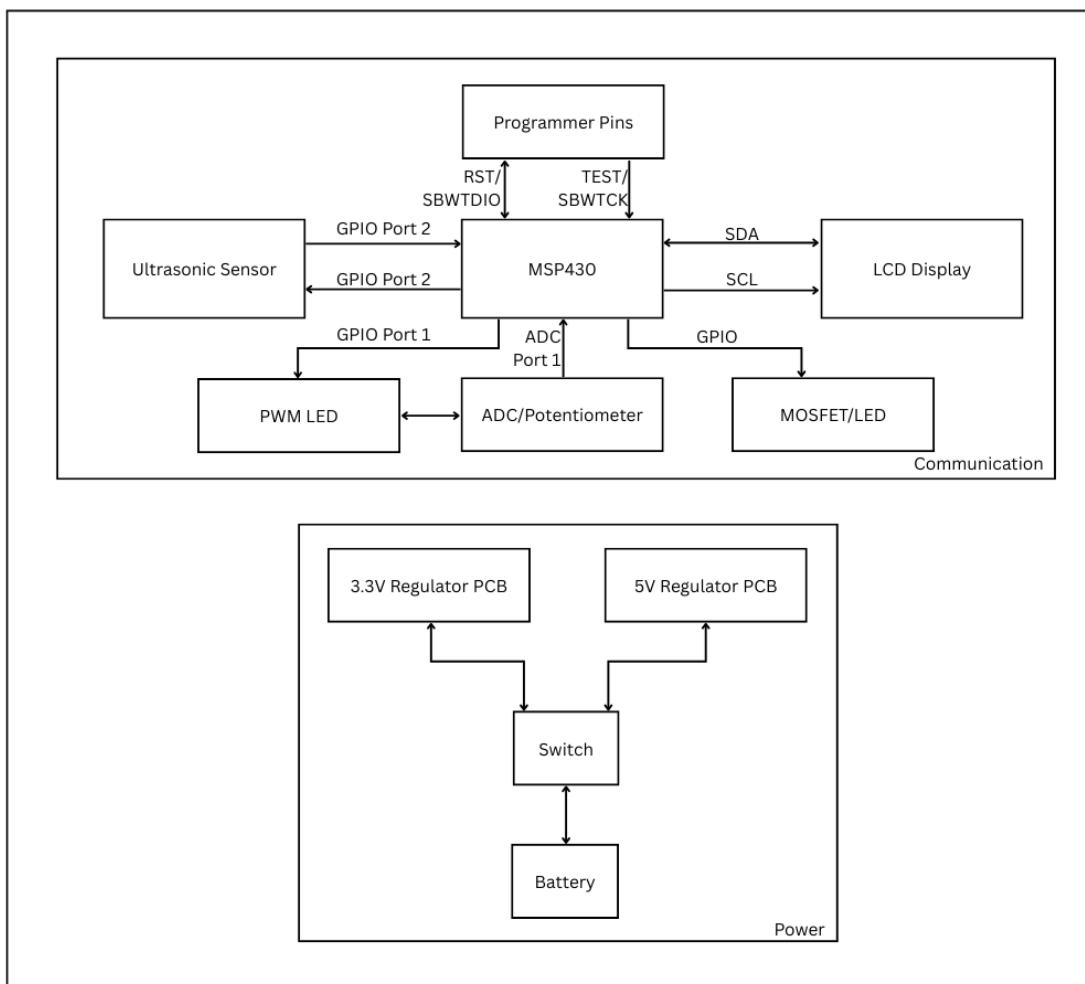
3.2.4: A port from MSP is chosen to blink LED control through a MOSFET. This pin will connect to a 1k resistor before connecting to gate terminal of the MOSFET to control the open and closing of the MOSFET. A 3.3V supply through an LED (with 1k Ohm bias resistor) is connected to the drain so they can flow to source (connect to ground) when the gate is open.

3.2.5: The MSP chip is not programmed by default, we need to write the code to the chip. A four-pin header connects to the UART line from the development board needed to do this programming step. There must be a connection to a common ground with both the designed PCB and the development board, hence one pin needs to connect to the GND. A test line is used to deliver the program packets from the IDE and debugger and requires a pin to connect to the MSP. Lastly, a reset line needs to connect to the development board with the output pin and pulled up (As per the datasheet for this IC and can vary depending on the IC) using a 1k Ohm resistor.

Function	Pin name	Pin number	Connect to	Description
Voltage input				
Voltage output				
SCL line				
SDA line				
Echo line				
ADC input line				
ADC output line				
Blinking LED line				
Programmer Test				
Programmer Reset				

Table 1: MSP430G2553 Pins out

3.3: Start creating a hand drawn schematic sketch with the associated pins for the components needed for the Range Finder and draw any necessary connections based on your defined pins from steps 3.1 and 3.2. A block diagram has been provided to help you get started and visual some of the connections which you should have listed above.



3.4: Download the template header library from Webcourses and modify lines 88 to 96 appropriately according to your chosen pins. Make sure you save this header in a location where you can access it again, you will need this when you begin your software prototype.

```
84 //-----+---  
85 //-----|---  
86 // #defines DONE BY STUDENT  
87 //-----|---  
88  
89 #define LCD_SCL      // Place your chosen SCL port here with the format of BITx  
90 #define LCD_SDA      // Place your chosen SDA port here with the format of BITx  
91 #define USENSOR_TRIGGER // Place your chosen Trigger port here with the format of BITx  
92 #define ECHO_INTERRUPT // Place your chosen Echo port here with the format of BITx  
93 #define ADC_INPUTPIN   // Place your chosen ADC input port here with the format of BITx  
94 #define ADC_INPUT     // place your chosen ADC input channel here in the format of INCH_x  
95 #define ADC_LED       // Place your chosen ADC LED port here with the format of BITx  
96
```

4.0: Project Assignment - What is to be uploaded to webcourses:

A report containing a summary of the steps in section 3.

- a. Electronic image of your final schematic.
- b. The list of pins in step 3.1.
- c. Table 1 filled out.
- d. Screenshot of modified header in step 3.7.
- e. A short summary of what you have learned.

5.0: Accepted File Format: PDF

6.0: Due Date: Tentative and available as Webcourses Assignment.

NOTE: Please read carefully all the steps for this laboratory assignment so not to miss any of the steps.

WEEK 7&8

LABORATORY 7: JUNIOR DESIGN PROJECT PCB

1.0: Description – In this assignment, the students will design and draw the schematic for the junior design project PCB. This will require the students to acquire the parts from the BOM (on webcourses) from Ultra-Librarian website. The students must consider the mechanical size aspect for the various parts such as the battery holder, the LCD display, and the ultrasonic sensor. By the end of this lab, the student will be able to order their blank PCB from one of the PCB manufacturers of their choice.

2.0: Goal – In this assignment the student will use the junior design PCB software to develop the PCB for their junior design project PCB (Range Finder). The functional block diagram for the range finder board is shown in Figure 1. The heart of the project is the MSP430G2553 microprocessor used to control a LCD, an ultrasonic sensor, and several LED's. There are several different functions implemented in this project. Developing software to read the analog voltage produced by the potentiometer by the MSP430 analog to digital (ADC) system, the displaying of character (ASCII) data on the LCD display, and reading the ultrasonic sensor to determine position location.

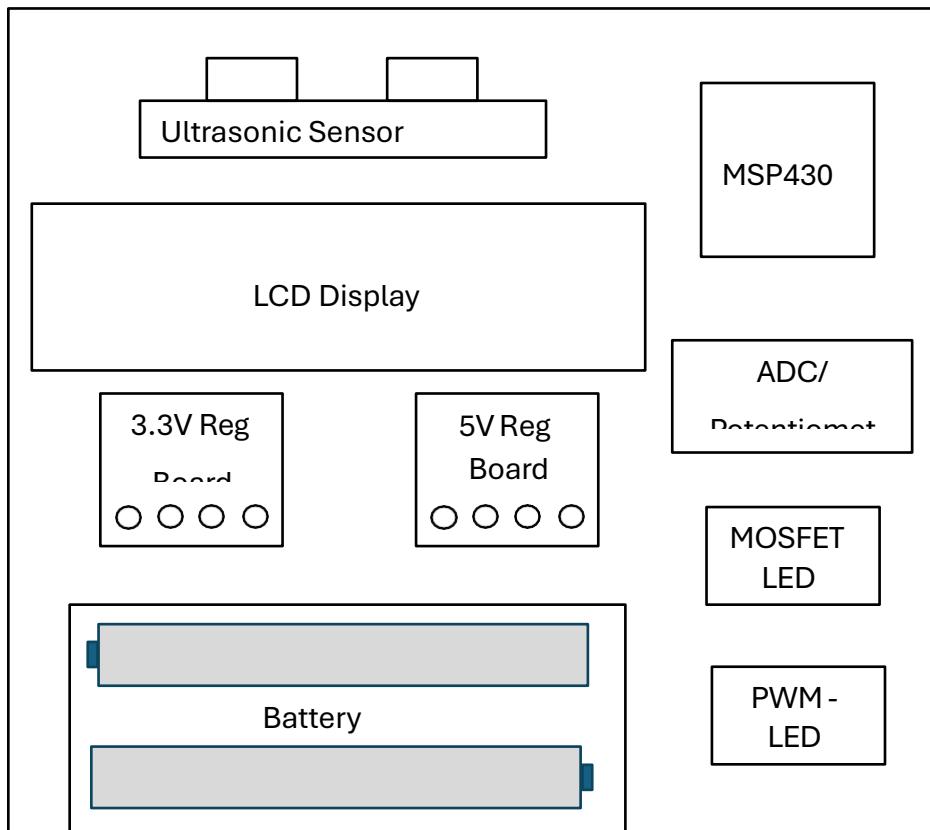


Figure 1: Functional block diagram for the junior design range finder project.

There are two LEDs included in this project: one controlled by a pulse width modulator that is used to control the brightness of this LED's and the other LED is implemented as a flashing LED to indicate that MSP430 is operational.

Figure 2 shows the implementation of range finder project using a TSSOP-20 pin package for the MS430G2553 (the breakout board on the right). The two breakout boards located to the left on the prototype board are the 3.3V and 5V regulators. The red LED is used as a power indicator and is not needed for your project as there are LED power indicators on each of your regulator boards.

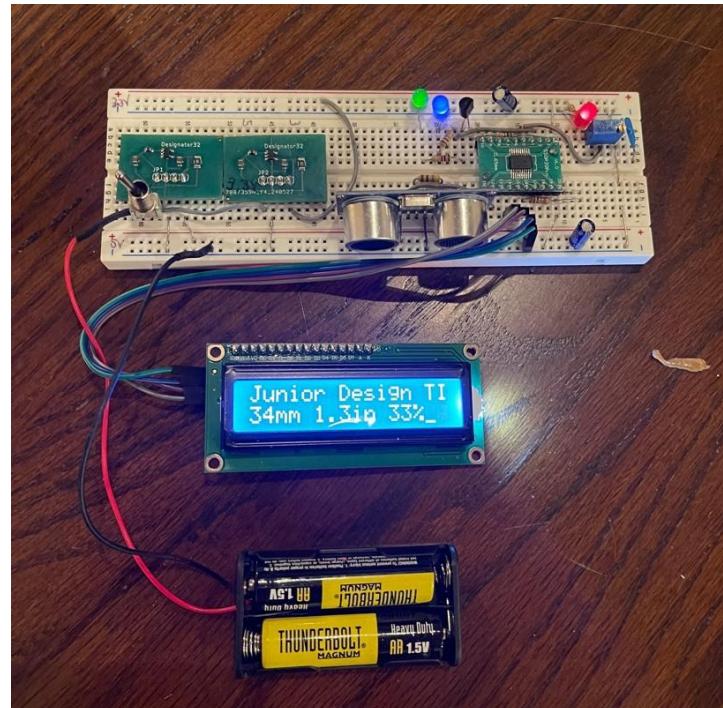


Figure 2: Prototype version of the junior design range finder project.

3.0: Procedure –

3.1: Starting with the enclosed BOM go the Ultra Librarian website and download the script files for each of the required components.

<https://www.ultralibrarian.com/>

(Do not forget that you need to have an account to download the Ultra Librarian file, make sure to create an account before proceeding).

3.2: The schematic that describes the above parts are designed by the student in previous lab (**CHECK WITH A TA BEFORE IMPLEMENT YOUR DESIGN**).

3.3: Open your regulator project and obtain the size of your regulator board(s) (Dimension tool or Board information from Cam Preview). You will need these dimensions of your regulator boards to properly size the junior design board.

3.3: The range finder board will contain an on / off switch, two 4 pin male 0.1" headers, and two 4 pin 0.1" female headers.

3.4: Using the dimensions for the LCD, ultrasonic sensor, the battery holder, your 3.3V regulator, your 5V regulator, and the potentiometer, layout approximately where the parts are to be located on your PCB. You can use WORD, Fusion 360 or any drawing package of your choice. Figure 3 shows a previous version of the junior design range finder project. Keep in mind that your 3.3V and 5V regulators will be on separate boards



Figure 3: Junior design range finder board.

3.5: From step 3.3 – 3.5 determine approximate size of your PCB. Do not forget to include room for the MSP430, one MOSFET (BS170), two LEDs, 2 resistors for biasing the 2 LEDs, and two resistors need for the voltage divider for the echo signal form the ultrasonic sensor. In addition, one resistor will be need as a pullup resistor for the reset-not pin and one for the gate of the MOSFET. You can get the size / footprint of the parts from Ultra Librarian.

3.6: Create a new parts library for this project and import into this library the parts are required for this project (see BOM).

3.7: Create a new project in your PCB design program. Also create a new schematic and a coupled board.

3.8: Using the PCB design tools along with the enclosed schematic draw the schematic in the PCB program. **Be careful in drawing the schematic as it needs to be correct for the board to function properly.**

3.9: Have the lab assistant review your schematic for correctness.

3.10: Using the board window in the PCB software, layout your parts corresponding to Steps 3.7- 3.9. Make sure to used the obtained board dimension and Dimension tool (in either silkscreen or value layer) to reserve space for the regulator.

3.11: Repeating the steps of week 2 PCB assignment build your PCB, generating the BOM and the required Gerber files. Do not forget to check your board for errors with your PCB design software (ERC).

3.12: Have the lab assistant review your board layout for correctness.

3.13: Lastly, the zip file output from the CAM processor, which is also called the gerber file, can be used to place an order through manufacture. We recommend using <https://jlpcb.com/> or

<https://www.pcbway.com/>. You will need to upload your zip file for the regulator as well as this project's zip file. For specification of the regulator order, place a quantity of 5, choose any color except for white, and select the stencil option. Choose custom size for your stencil and add 10mm to your board dimensions, i.e. h+10mm, w+10mm and not framed stencil. For the project board order, place a quantity of 5, again choose any color except for white, and select the stencil option with the same custom sizing of an additional 10mm. Please keep these parameters (beside the color) the same to avoid extra charges.

The screenshot shows the PCBWay order configuration interface. It includes sections for Base Material (FR-4), Layers (4), Dimensions (100x100 mm), PCB Qty (5), and Product Type (Industrial/Consumer electronics). The 'PCB Specifications' section covers Different Design (1), Delivery Format (Single PCB), PCB Thickness (1.6mm), PCB Color (Green), Silkscreen (White), and Surface Finish (HASL(with lead)). The 'High-spec Options' section includes Outer Copper Weight (1 oz), Via Covering (Tented), Min via hole size/diameter (0.3mm/(0.4/0.45mm)), Board Outline Tolerance (±0.2mm(Regular)), Confirm Production file (No), and various marking and test options like Order Number, Flying Probe Fully Test, and Edge Plating. An 'Advanced Options' section is also present.

4.0: Project Assignment - What is to be uploaded to webcourses: A report containing a summary of the steps in section 3.

- f. Electronic image of your final schematic.
- g. A comparison of the included schematic and your PCB schematic
- h. Electronic image of your final board layout.
- i. Included in your report your BOM.
- j. A short summary of what you have learned.

k. Ordering information for a **blank board for your regulator and project board** from a PCB manufacturer of your choice.

5.0: Accepted File Format: PDF

6.0: Due Date: Tentative and available as Webcourses Assignment.

NOTE: Please read carefully all the steps for this laboratory assignment so not to miss any of the steps.

WEEK 9&10

LABORATORY 8: JUNIOR DESIGN PROJECT PROTOTYPE / SOFTWARE

1.0: Description – In this assignment, the students will build a prototype of their junior design project using the MSP-EXP430G2ET development board from Texas Instrument and one of the breadboards from the junior design laboratory (Figure 1). The students will then develop the software required for the junior design project.

2.0: Goal – In this assignment the student will build the functional sections of the junior design range finder project. The functional block diagram for the range finder board is shown in Figure 2. The heart of the project is the MSP430G2553 microprocessor used to control a LCD, an ultrasonic sensor, and several LED's. There are several different functions implemented in this project. This assignment will develop software to read the analog voltage produced by the potentiometer using the MSP430 analog to digital (ADC) system, the displaying of character (ASCII) data on the LCD display, and reading the ultrasonic sensor to determine position location.

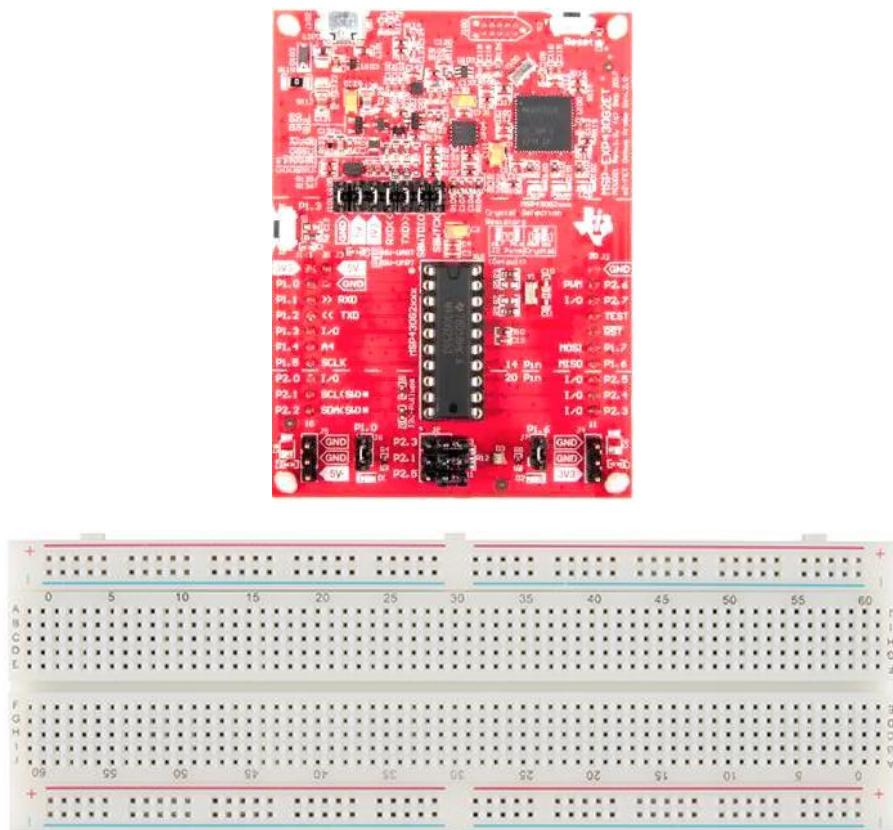


Figure 1: The MSP-EXP430G2ET development board and breadboard.

https://www.mouser.com/ProductDetail/Texas-Instruments/MSP-EXP430G2ET?qs=%252BEew9%252B0nqrDSYsq38fqB2w%3D%3D&mgh=1&utm_id=17222215321&gad_source=1&gclid=EAIAIQobChMIu73imtLnhgMViK1aBR0EFQ7-EAQYBSABEgLY8_D_BwE

There are two LEDs included in this project: one controlled by a pulse width modulator that is used to control the brightness of this LED's and the other LED is implemented as a flashing LED via a MOSFET to indicate that MSP430 is operational.

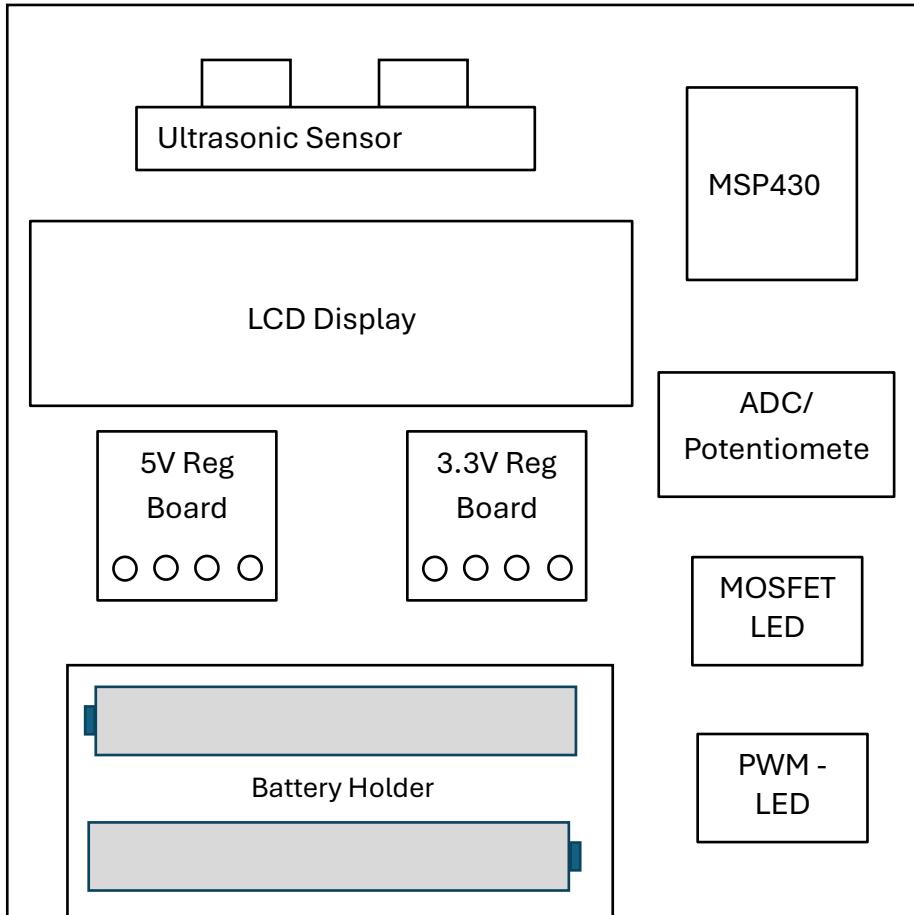


Figure 2: Functional block diagram for the junior design range finder project.

This assignment will be implemented in sections building each one of the functional blocks of the junior design project by writing the software for each section. Finally, each one of the functional blocks will be combined to produce an integrated prototype with software that represents the junior design project.

Figure 3 shows the implementation of range finder project using MSP-EXP430G2ET development board and breadboard. The breadboard contains two LED's; one for blinking and one where its brightness is adjustable via the potentiometer also located on the breadboard. The ultrasonic sensor along with the required voltage divider is located on the breadboard. The BSD170 MOSFET is used to turn on and off the blinking LED. The LCD display connects directly to the MSP430 via port 1 P1.6 and P1.7, 5v, and ground.

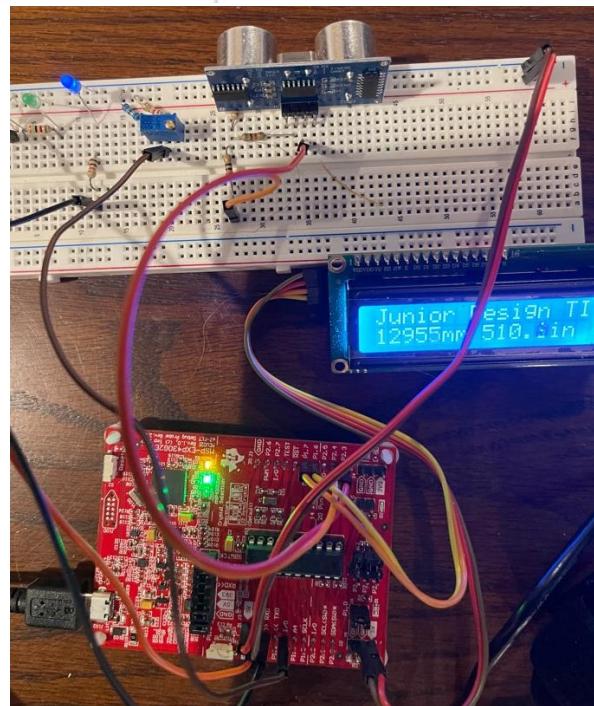
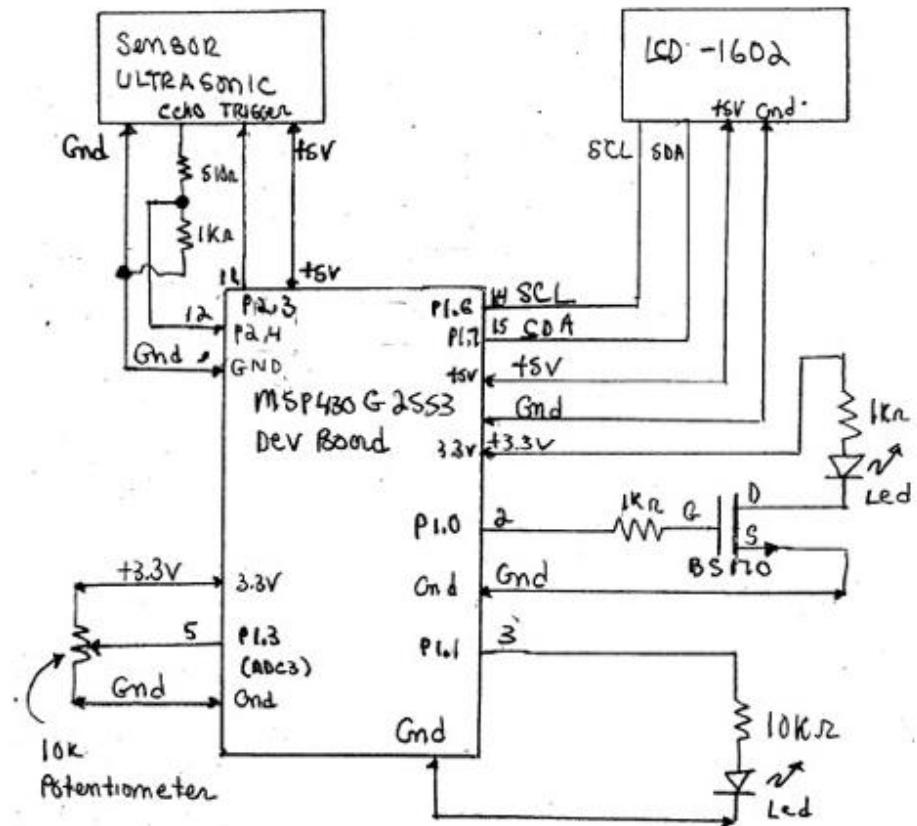


Figure 3: Prototype version of the junior design range finder project.

3.0: Procedure –

NOTE: Code Composer Studio version 12 and older (Eclipse based) is recommended for this project to minimize any complication can happen in (Theia).

3.1: Attached with this assignment is a list of library functions created for this laboratory assignment. Please become familiar with these functions. To use these functions, which are part of the included file that is on Webcourses, the included file must be included as part of your code composer project.

3.2: Start code composer and take notes in the start menu where the project workspace is located on your PC. Create a new code composer project and select the MSP430G2553 as the processor. Next, copy the included file to the workspace directory of where your project is located given in open menu. Now add this included file to your project. Your first lines of code need to have the following files included in your program.

```
#include "msp430.h"
// Needed by the sprintf function
#include "stdio.h"
// Header file needed for junior design project -
// Includes the LCD library
#include "Junior_Design.h"
```

To use the library functions, the hardware on the MSP430G2553 development board must be initialized. Let's comment out the `LCD_Initialize();` We will uncomment these lines later.

```
void main(void)
{
    // Go initialize the MSP430G2553 hardware to
    // setup I2C and the timers
    // Set the MCLK to 8 MHz and setup GPIO
    // for the ultrasonic sensor

    Init_HW();
    // Go initialize the LCD display
    LCD_Initialize();
    while(1)
    {
        // you code needs go here
        // to continuously run
        // you may to use the timer delay function
        // to control the speed of this loop
    }
}
```

3.3: To turn on port 1 P1.0 LED, the following instruction would be used.

```
// Turn on the LED on Port P1.0
P1OUT |= BIT0;
// And to blink the LED on and off
// toggle P1.0 LED on and off
P1OUT ^= BIT0;
```

3.4: Write a program to blink the LED connected to P1.0 on and off at a rate of 500 milliseconds on and 500 milliseconds off. Use the timer delay function **Delay_Timer** (more information can be found in Junior Design Software document) to control your delay time. Run the program and observe that the LED on the dev board is blinking on and off.

```
void main(void)
{
    // Go initialize the MSP430G2553 hardware to
    // setup I2C and the timers
    // Set the MCLK to 8 MHz and setup GPIO
    // for the ultrasonic sensor
    // Also, setup the GPIO as an interrupt input
    Init_HW();
    // Go initialize the LCD display
    LCD_Initialize();
    while(1)
    {
        // Turn on and off the LED here
        // to continuously run
        // Add your 500 millisecond delay here
        // to control the speed of this loop
    }
}
```

3.5: *Remove the two jumpers P1.1 and P1.6 on the dev board. They are used to turn on the two LED's on the dev board. Add to your breadboard, the BSD170 MOSFET, two 1K resistors and one LED as shown in Figure 3 (make sure you connect the MOSFET and LED in the correct orientation). Rerun your program and observe that the LED on the breadboard is blinking. Take two images of the LED on and then off and save them for your report. Also save a copy of your program as you will need to include it in your report.

3.6: Add a second LED using a 10K resistor for the LED bias current to port 1 P1.1 to the breadboard.

3.7: *Start a new project using steps 3.1-3.3. This time we are going to turn on the LED connected to Port 1 P1.1. We will need to turn on the LED:

```
// Turn on the LED on Port P1.1
P1OUT |= BIT1;
```

The brightness for this LED is controlled by the variable `Duty` defined in the `Junior_Design` header file. This variable controls the duty cycle of the pulse width modulation (PWM). The valid range for this variable is 0 to 100. Run your program several times with different values for the variable `Duty`. Also put your program in a 500-millisecond loop like you did to flash the LED on port 1 P1.0. Take some images of the LED with various brightness levels and save them for your report. Also save a copy of your program as you will need to include it in your report.

```
void main(void)
{
    // Go initialize the MSP430G2553 hardware to
    // setup I2C and the timers
    // Set the MCLK to 8 MHz and setup GPIO
    // for the ultrasonic sensor
    // Also, setup the GPIO as an interrupt input
```

```

    Init_HW();
    // Go initialize the LCD display
    LCD_Initialize();
    // Turn on Port 1 P1.1 LED on here
    while(1)
    {
        //
        // Set the Duty variable here
        // between 0 to 100;
        // Add your 500 millisecond delay here
        // to control the speed of this loop
    }
}

```

At this point we are ready to install the LCD display to the MSP430G2553 dev board. There are four wires require +5V, Gnd, SDA, and SCL. Connect these pins from the LCD to Port 1 P1.6, port 1 P1.7, +5V and ground as shown in Figure 3. Have the lab assistant (GTA) check your wiring. Let's take the comments out from `LCD_Initialize();`

3.8: Start a new project using steps 3.1 and 3.2. This time we are going to display the message on the LCD display “Junior Design”. We will put this character string in a string array and then using the `LCD_String` function we will display this message to the LCD display. First, we must clear the LCD screen. These instructions should be implemented outside the while loop part of your code.

```

// Store message in a character array
char Data[] = "Junior Design ";
// these lines are for the LCD display
LCD_CLRscreen();
// Junior Design Message
LCD_String(Data);

```

Write a program that display the message “Junior Design” to the LCD display. Have is update the display every 500 milliseconds. You may need to adjust the brightness of the LCD display. The brightness control is located on the back of the LCD panel on the I2C controller board via a potentiometer.

```

// Add character array definition here for Data
void main(void)
{
    //
    // Go initialize the MSP430G2553 hardware to
    // setup I2C and the timers
    // Set the MCLK to 8 MHz and setup GPIO
    // for the ultrasonic sensor
    // Also, setup the GPIO as an interrupt input
    Init_HW();
    // Go initialize the LCD display
    LCD_Initialize();
    while(1)
    {
        //
        // clear the LCD screen here
        // Use the LCD_String() function here
        // Add your 500 millisecond delay here
        // to control the speed of this loop
    }
}

```

```
}
```

3.9: *Adding to the program in step 3.8, create a counter that every time it executes through the 500 milliseconds loop it increments from 0 to 90 by 10 and then rolls over to 0. Set the Duty variable equal to this value and then display the Duty variable value on the LCD display. The modulus function can be used to rollover the variable back to 0 whenever a value of 100 is reached.

```
int i;  
i= (i+10) % 100;  
Duty = i;
```

To display a variable on the LCD display we will use the sprint() function. First, we need to reserve space in a character array for 32 characters (32 characters for the 1602 LCD display).

```
// Reserve an array of 32 characters  
char Data2[32] = "";
```

The sprint() function is the same as the printf function except it does not display the results to the screen but puts the results in a string. Code composer limits the sprint() function to only strings and integer numbers not floating-point numbers for the MPS430G2553 processor.

```
// Store the value of Duty in a character array Data2  
sprintf(Data2, "%d", (int)Duty);  
// Let's move the cursor to the second line  
LCD_Cursor_Position(0x40);  
// Set the LCD cursor to no blink  
LCD_Cursor_On_No_Blink();  
// To Display the value of the variable Duty on the LCD  
LCD_String(Data2);
```

An example of the program:

```
// Add character array definition here for Data2  
void main(void)  
{  
    // go define the variable i  
    // Go initialize the MSP430G2553 hardware to  
    // setup I2C and the timers  
    // Set the MCLK to 8 MHz and setup GPIO  
    // for the ultrasonic sensor  
    // Also, setup the GPIO as an interrupt input  
    Init_HW();  
    // Go initialize the LCD display  
    LCD_Initialize();  
    while(1)  
    {  
        // Clear the LCD screen here  
        // Increment i by 10 here  
        // Set the variable Duty to i here  
        // Use the sprintf() function here  
        // Move the cursor to the second row  
        // Set the cursor to no blink here
```

```

    // use the sprintf() function to display your result
    // Add your 500 millisecond delay here
    // to control the speed of this loop
}
}

```

Take some images of the LCD display showing different values for the variable i displayed on the LCD display. Save these images for your report. Also save a copy of your program as you will need to include it in your report.

3.10: *Create a new project that lets you write messages to the LCD as in step 3.9. If you wish you can copy the section of code from step 3.9 that accesses the LCD display to this project. Now connect the 10K ohm potentiometer to the breadboard as showing Figure 3. To read the voltage on the potentiometer you can use the ADC_Read() function. The value returned by this function is 0v = 0 and 3.3v = 1023.

```

// The variable returned by the ADC is defines as
// needs to be declared in the beginning of the program
static unsigned int ADC_Result;

// Go read the potentiometer
ADC_Result = ADC_Read();

```

Next, scale the value read by the ADC between 0 and 100 and set it equal to the Duty variable.

```

// go scale the duty cycle between 0 and 1023/103 = 99%
Duty = (int)((float)ADC_Result/(float)10.23);

```

As the potentiometer is changed from 0 to 3.3 volts the LED on port 1 P1.1 will change in brightness from off to full brightness. Make sure you program runs in a 500-millisecond loop. Also display on the LCD display the value of the Duty variable. Add a percent symbol after the Duty variable.

Take some images of the LCD showing different values for the Duty variable and for different brightness levels of the LED on the breadboard. Save these images for your report. Also save a copy of your program as you will need to include it in your report.

3.11: Now it's time to add the ultrasonic sensor to the breadboard. The sensor uses four connections: +5v, Gnd, trigger and echo. The echo pin uses a voltage divider composed of 510 ohms and 1k ohms to reduce the 5-volt logic level from the ultrasonic sensor to 3.3v which is compatible with the MSP430G2553. Have the lab assistant (GTA) check your wiring.

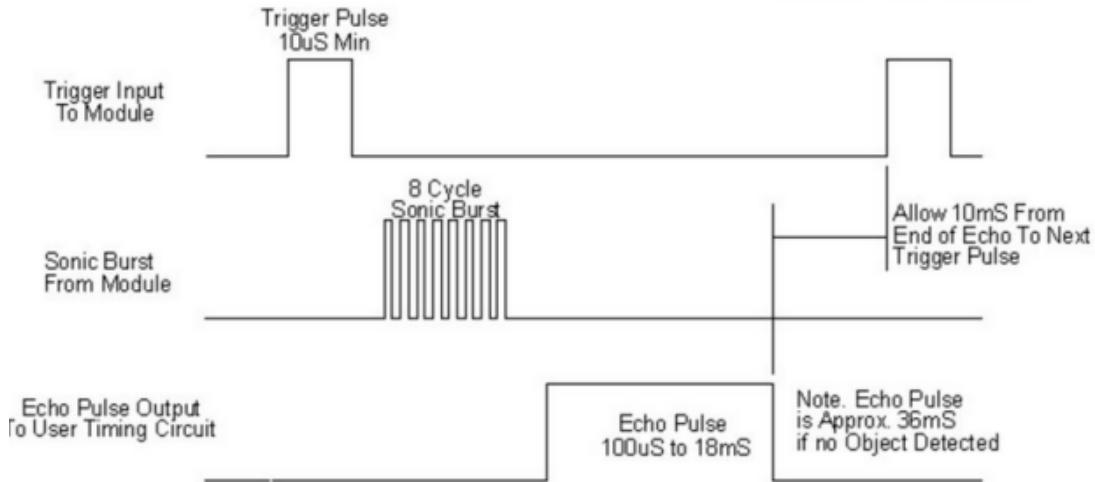


Figure 4: Timing diagram for the ultrasonic sensor.

<https://www.handsontec.com/dataspecs/HC-SR04-Ultrasonic.pdf>

3.12: *Create a new project that lets you write messages to the LCD as in step 3.9. If you wish you can copy the section of code from step 3.9 that accesses the LCD display to this project. You can remove the increment i variable and the setting the Duty variable part of the program. In this program we are going to send a trigger pulse to the ultrasonic sensor and then read the ultrasonic sensor echo signal and display the result in millimeters and inches (integer and one digit of the fractional part) on the LCD display.

```
// Start and trigger an ultrasonic sensor read
Ultrasonic_Trigger();
// need to wait at least 60 msec
Delay_Timer(DELAYT_100ms);
// go read the ultrasonic sensor
// each count = 10 usec
// the variable D1 is defined in the begining of the
// program as static unsigned long int d1;
d1= Ultrasonic_Echo_Read();
// 340 m/s for a 10 usec per count gives
// 3.4 mm (0.1338 in) per count both directions
// to and from the object or 1.7mm (0.066929 in)
// per 10 usec in one direction
```

To print fractional part of the inches and since the sprintf() function does not allow floating point numbers we have to convert the parts of the fractional part of the number to an integer value.

```
// 10 x result gives 0.66929
// this line converts the 10 usec counts to inches
// and gives the integer part in d_1
d_1 = (int)((float)d1*0.066929);
// multiplying by 0.66929 give 10 time the result
// the units are the first fractional digit
// subtracting 10 x the integer result (d_1) extracts
out the
// first fractional digit
```

```

d_10 = (int)((float)d1*0.66929) - d_1*10;
// both d_1 and d_10 are defined as: int d_1, d_10;
// in the beginning of the program.

```

We now can print the ultrasonic sensor result in mm and inches using the `sprint()` and the `LCD_String()` functions().

```

sprintf(Data2,"%dmm %d.%din
", (int)((float)d1*3.4/2.0), d_1, d_10);
LCD_String(Data2);

```

Here is an example of this program:

```

// define the variables d1,d_1,d_10 here
// Let's move the cursor to the second line
LCD_Cursor_Position(0x40);
// Set the LCD cursor to no blink
LCD_Cursor_On_No_Blink();

// Add character array definition here for Data2
void main(void)
{
// Go initialize the MSP430G2553 hardware to
// setup I2C and the timers
// Set the MCLK to 8 MHz and setup GPIO
// for the ultrasonic sensor
// Also, setup the GPIO as an interrupt input
Init_HW();
// Go initialize the LCD display
LCD_Initialize();
while(1)
{
// Clear the LCD screen here
// Start an ultrasonic trigger here
// wait 100 milliseconds
// read the ultrasonic sensor
// Move the cursor to the second row
// Set the cursor to no blink here
// scale the ultrasonic result to mm and inches
// use the sprintf() function to display your result
// Add your 500 millisecond delay here
// to control the speed of this loop
}
}

```

Take some images of the LCD showing different values for the display position from the ultrasonic sensor. Save these images for your report. Also save a copy of your program as you will need to include it in your report.

3.13: *This is the last step in the breadboard junior design ranger finder assignment. Starting with step 3.5(the blinking LED project), step 3.7(turning on the dimmable LED), step 3.9(varying the duty cycle), step 3.10 reading the ADC (analog to digital converter), and step 3.12(reading the ultrasonic sensor),

combing all the projects together into one project that implements the following for your junior design project. Figure 5 gives an example output from the range finder breadboard implementation.

- Displays on the first line of the LCD “Junior Design”.
- Blink the blink LED.
- Reads the potentiometer using the ADC_Read() function.
- Change the brightness of the dimmable LED by varying the potentiometer.
- Displays on the second row of the LCD the ultrasonic sensor distance in mm and in inches (with 1 fractional digital).
- At the end of the second row of the LCD is displayed the duty cycle (Duty) in percentage. All these requirements are implemented at a 500-millisecond rate.

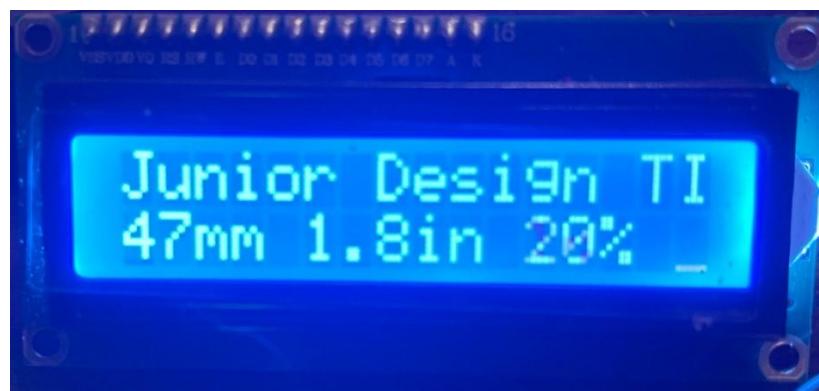


Figure 5: An example output from the breadboard range finder project.

4.0: Project Assignment - What is to be uploaded to webcourses:

This assignment will be divided into four parts:

- Part 1: Steps 3.5 and 3.7 - The blinking LED and dimmable LED.
- Part 2: Steps 3.9 and 3.10 - Displaying strings / integer numbers on the LCD / ADC input.
- Part 3: Steps 3.12 - Reading and displaying the ultrasonic sensor data.
- Part 4: Step 3.13 - The integrated software program for the junior design project.

For each part, a report containing:

- A short summary describing the step(s) due.
- Images taken for each step along with a copy of your program for each step.

5.0: The students will have the ability to check out of the lab the MSP430G2553 development board as well as one breadboard to complete this project at their own pace. To receive a grade for this lab the MSP430G2553 development board and the breadboard must be returned at the end of the semester.

6.0: Accepted File Format - PDF

7.0: Due Date - Tentative and available as Webcourses Assignment.

NOTE: Please read carefully all the steps for this laboratory assignment so not to miss any of the steps.

WEEK 11-14

LABORATORY 9: PCB FABRICATION AND SOLDERING (REGULATOR BOARD)

1. **Description** – In this assignment, the students will build their regulator boards learning the process of reflow soldering.
2. **Goal** – At the end of this assignment the student will learn the reflow soldering process in the assembling of their regulator boards (3.3v and 5.0v). Next, the student will verify that their regulator boards are functioning correctly.
3. **Procedure** – Methods used in fabricating your regulator boards.

3.1. SAFETY FIRST:

- Nitril gloves and safety glasses must be worn when using solder paste.
- Laboratory coats must be worn during the solder paste process.
- Only long pants can be worn during the soldering process.
- Only the approved GTA for the reflow oven can reflow the PCB's.

3.2. Take photos and images showing your process steps. You will need these images for your report.

3.3. Using extra PCB boards to hold your regulator PCB and using blue masking tape attach your regulator board to a flat surface.

3.4. Align stencil over the board, matching the holes in the stencil mask to the corresponding SMD pads on the board (Figure 1a).

3.5. Tape the stencil down over the PCB in place, preferably on a sturdy surface like a table (Figure 1b) keeping the stencil aligned to the PCB.

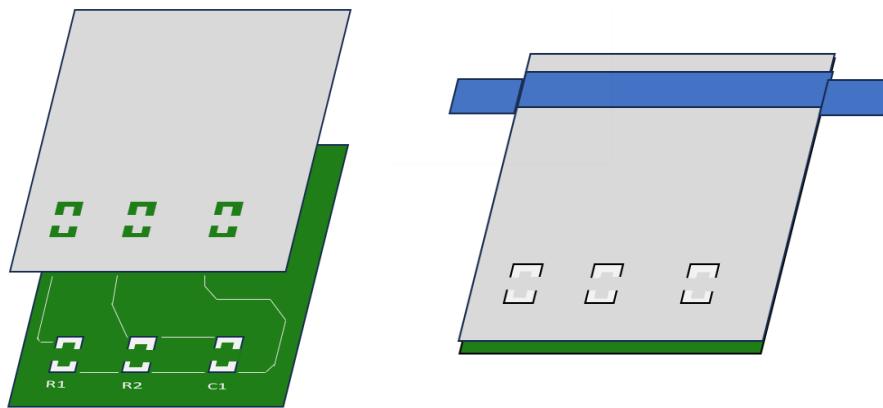


Figure 1: a. Placing the stencil over the PCB and b. taping the stencil to the PCB.

3.6. Using either a jar of solder-paste or a solder paste syringe, place or dispense a pea-sized amount of solder paste on the stencil, above the row of holes you wish to apply the solder paste to as shown in Figure 2.

- Do not place the solder paste directly into the holes, as this causes uneven application of solder paste.
 - Do one row of components at a time.
- 3.7. Use an item such as a flat plastic card to “squeegee” the solder paste across the stencil holes. In the junior design laboratory, we have plastic blade scrappers that can be used.

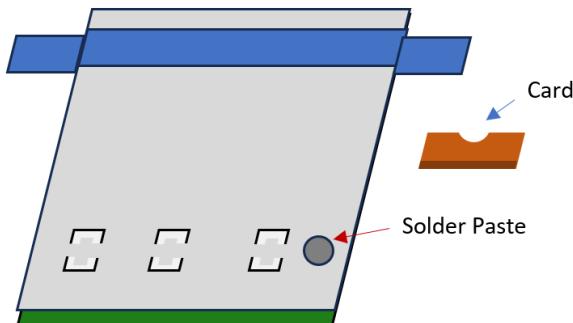


Figure 2: Using a solder paste and a plastic card applying to a PCB.

- 3.8. Check whether the solder paste application was successful. You should not see the metal of the SMD pads through the stencil mask. You should only see the solder paste on the pads.
- 3.9. If more solder paste is needed on the PCB’s pads, reuse the excess paste from the first batch (“squeegee” attempt).
- It is unlikely that it was all used. Repeat as needed to ensure good coverage of the PCB’s pads.
 - Be careful not to use too much solder paste on the board, as this may cause issues with component adhesion or create a bad electrical connection such as a bridge or a short.
- 3.10. Repeat this process for as many rows of SMD components that are on your board.
- 3.11. Once the solder paste is adequately applied to the PCB, remove the tape and take the stencil off the board.
- 3.12. Clean the stencil using isopropyl alcohol wipes. Be sure that there is no remaining solder paste in the **stencil** mask holes.
- 3.13. Carefully place the SMD components on their designated positions on the regulator boards using tweezers.
- Do not apply force when doing so, only place the components onto the solder-pasted pads. Applying force will cause the solder paste to spread, which is undesirable and can lead to SMD not being solder correctly or solder shorts.
 - Do the best job that you can to make sure that the components are centered, not misaligned, and that each terminal on the SMD components are touching their corresponding pads on the PCB.

- The surface tension of the solder paste with the pad will perform some self-alignment.
- Figure 3 is an example of our regulator board, and it shows in yellow the pads that should be covered with solder paste except the through-hole parts (4 pin header at the bottom). These parts will be soldered using wire solder after the surface mount parts are reflowed soldered.

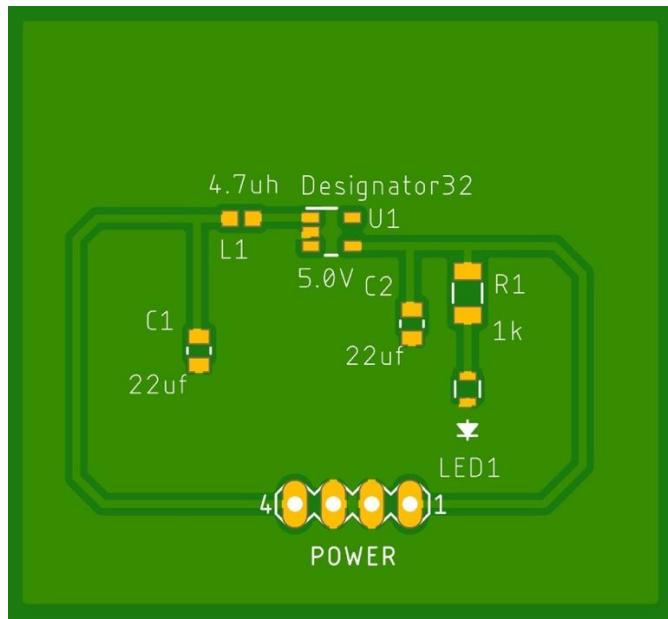


Figure 3: An example circuit board showing the pads to be solder paste.

- 3.14.** Next, you will need to use Kapton tape, to hold the thermocouple to the board to that the temperature of the board can be measured during reflow soldering process. Place the board on the belt of the solder reflow oven so your board can be soldered.
- 3.15.** Once the SMD components have been reflowed, the through-hole pin header can be hand-soldered to the board using wire soldering techniques. Have the GTA for the lab help you in this wire soldering process.

4. Procedure (3.3V and 5.0v regulator testing) –

- 4.1.** Using the breadboard in the laboratory place your 3.3volt regulator on this board.
- 4.2.** Please take note of the pinout that you used for your regulator.
- 4.3.** Have the laboratory GTA verify your wiring.
- 4.4.** Apply 2.9 volts to the input of the regulator using the laboratory power supply, limit the maximum current to 0.5 amps for the power supply.
- 4.5.** Turn on the laboratory supply on and take note that your LED is on (take a photo).
- 4.6.** Measure V_{out} from your regulator (Record this value).
- 4.7.** Next, measure a 100-ohm resistor and place this 100-ohm resistor across V_{out} and ground.
- 4.8.** Measure V_{out} from your regulator (Record this value).
- 4.9.** Calculate the current through this 100-ohm resistor.
- 4.10.** Have the laboratory GTA take a thermal image of your regulator board powered on.
- 4.11.** Have the laboratory GTA verify that your regular is working.
- 4.12.** Repeat steps 1 – 11 for the 5volt regulator.

5. Project Assignment – What is to be uploaded to webcourses:

- a. A short summary describing your regulator fabrication process steps.
- b. Include photos of these steps.
- c. The results of Procedure steps 4.1 - 4.12.
- d. Include the measured V_{out} for the 3.3v and 5.0v regulators
- e. Give the calculated current through the 100-ohm resistor for the 3.3V and 5v regulators.
- f. Include your thermal image of your regulators with your report.
- g. Discuss the results you obtained in Procedure 4.
- h. A short summary on what you have learned.

6. Accepted File Format – PDF

7. Due Date: Tentative and available as Webcourses Assignment.

NOTE: Please read carefully all the steps for this laboratory assignment so not to miss any of the steps.

LABORATORY 10: PCB FABRICATION AND SOLDERING (FINAL PROJECT BOARD)

5. **Description** – In this assignment, the students will build their junior design project board using the reflow soldering process.
6. **Goal** – At the end of this assignment the student will have completed their junior design project which includes fabricating their PCB's, uploading the project software, and integrating their regulators into their PCB.
7. **Procedure** – Methods used in fabricating your regulator boards.

3.16. SAFETY FIRST:

- Nitril gloves and safety glasses must be worn when using solder paste.
- Laboratory coats must be worn during the solder paste process.
- Only long pants can be worn during the soldering process.
- Only the approved GTA for the reflow oven can reflow the PCB's.

3.17. Take photos and images showing your process steps. You will need these images for your report.

3.18. Using extra PCB boards to hold your regulator PCB and using blue masking tape attach your regulator board to a flat surface.

3.19. Align stencil over the board, matching the holes in the stencil mask to the corresponding SMD pads on the board (Figure 1a).

3.20. Tape the stencil down over the PCB in place, preferably on a sturdy surface like a table (Figure 1b) keeping the stencil aligned to the PCB.

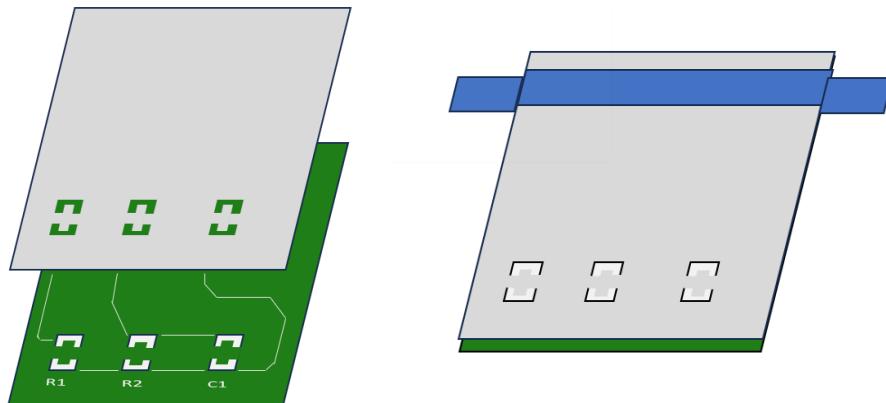


Figure 1: a. Placing the stencil over the PCB and b. taping the stencil to the PCB.

3.21. Using either a jar of solder-paste or a solder paste syringe, place or dispense a pea-sized amount of solder paste on the stencil, above the row of holes you wish to apply the solder paste to as shown in Figure 2.

- Do not place the solder paste directly into the holes, as this causes uneven application of solder paste.
- Do one row of components at a time.

3.22. Use an item such as a flat plastic card to “squeegee” the solder paste across the stencil holes. In the junior design laboratory, we have plastic blade scrappers that can be used.

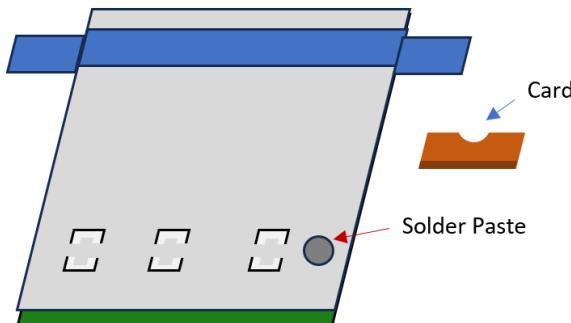


Figure 2: Using a solder paste and a plastic card applying to a PCB.

3.23. Check whether the solder paste application was successful. You should not see the metal of the SMD pads through the stencil mask. You should only see the solder paste on the pads.

3.24. If more solder paste is needed on the PCB’s pads, reuse the excess paste from the first batch (“squeegee” attempt).

- It is unlikely that it was all used. Repeat as needed to ensure good coverage of the PCB’s pads.
- Be careful not to use too much solder paste on the board, as this may cause issues with component adhesion or create a bad electrical connection such as a bridge or a short.

3.25. Repeat this process for as many rows of SMD components that are on your board.

3.26. Once the solder paste is adequately applied to the PCB, remove the tape and take the stencil off the board.

3.27. Clean the stencil using isopropyl alcohol wipes. Be sure that there is no remaining solder paste in the **stencil** mask holes.

3.28. Carefully place the SMD components on their designated positions on the regulator boards using tweezers.

- Do not apply force when doing so, only place the components onto the solder-pasted pads. Applying force will cause the solder paste to spread, which is undesirable and can lead to SMD not being solder correctly or solder shorts.
- Do the best job that you can to make sure that the components are centered, not misaligned, and that each terminal on the SMD components are touching their corresponding pads on the PCB.
- The surface tension of the solder paste with the pad will perform some self-alignment.

3.29. Next, you will need to use Kapton tape, to hold the thermocouple to the board to that the temperature of the board can be measured during reflow soldering process. Place the board on the belt of the solder reflow oven so your board can be soldered.

3.30. Once the SMD components have been reflowed, the through-hole pin header can be hand-soldered to the board using wire soldering techniques. Have the GTA for the lab help you with this wire soldering process.

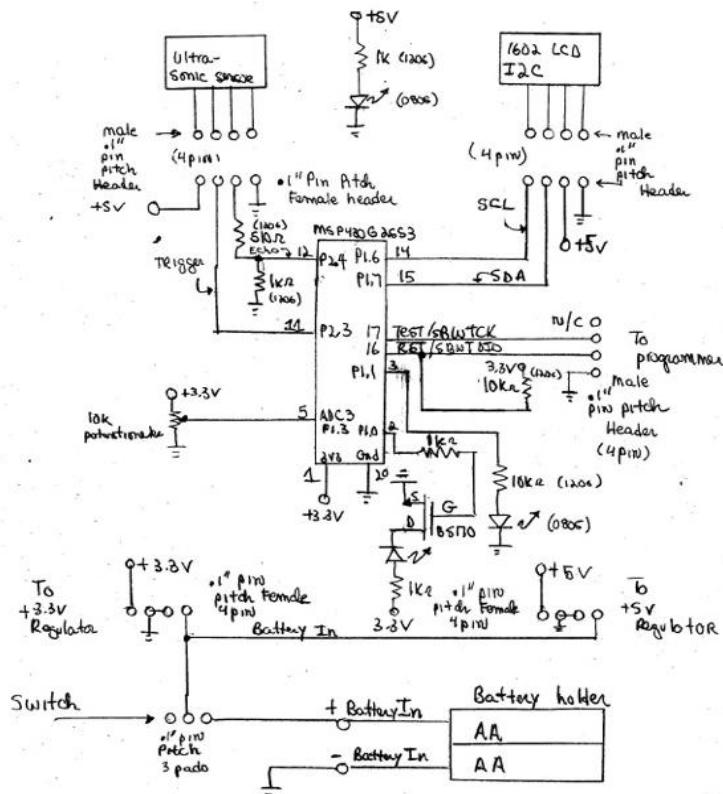


Figure 3: Schematic of the MSP430G2553 interface pins

8. Procedure (junior design project board testing):

- 4.1.** Once the junior design project PCB has been assembled, have the laboratory GTA verify that parts have been assembled correctly.
- 4.2.** Insert your regulators into the two connectors on your project board. Be careful and make sure to place the 3.3 volt and 5 volt regulators into the correct connectors.
- 4.3.** Next, insert batteries into your project board, turn on your project board and verify that it turns on. Verify the LEDs on both regulators turn on and the LCD panel turns on.
- 4.4.** With the help of the laboratory GTA download your project software onto the MSP430g2553 processor using the MSP-EXP430G2ET development board. You will need to connect wires from your project board programming header to P1.6 (SWBTUDIO), P1.7 (SWBTCK), and ground. Using code composer studio to download your software to the project board (Figure 3).
- 4.5.** Turn on the project board and verify that you project works correctly. Have the laboratory GTA verify that your project works. You should see the range finder work on the LCD, the blinking LED should blink and by turning the potentiometer the brightness of the other LED should change.

9. Project Assignment – What is to be uploaded to webcourses:

- a. A short summary describing your junior design project fabrication process steps.
- b. An image of your project showing it working
- c. A short summary on what you have learned.

10. Accepted File Format – PDF

11. Due Date – Tentative and available as Webcourses Assignment.

NOTE: Please read carefully all the steps for this laboratory assignment so not to miss any of the steps.