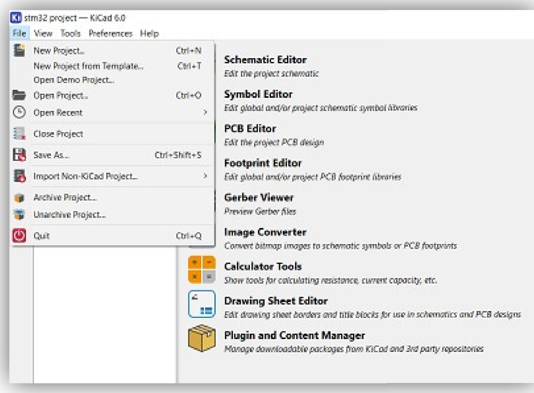
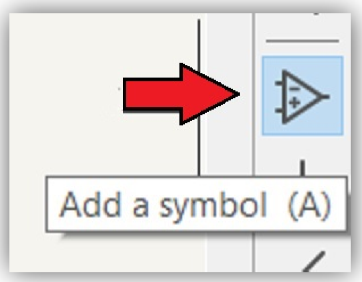
<u>Designing an STM32 Microcontroller board</u>

Here are the steps I took to create my first microcontroller board, the STM32 based board.
I used Kicad version 6 to create this board. The first thing to do is Open Kicad, Click 'File'

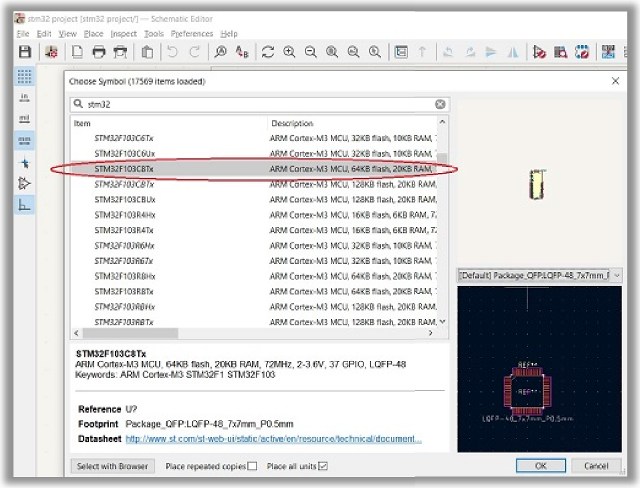


in the top left corner, next click 'New Project' then assign the project a name.
We will start out by creating a schematic. To do this we will click on the
'Schematic Editor' icon. To open the blank schematic editor canvas.
This board will be based on the STM32F103C8T6 original design.

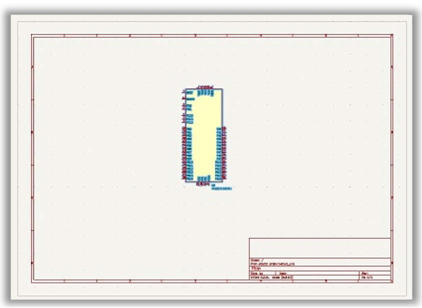
Now to begin placing symbols on our canvas we click on the 'Create a symbol' icon, located
On the right side of the screen near the top. These symbols may also be listed as components.



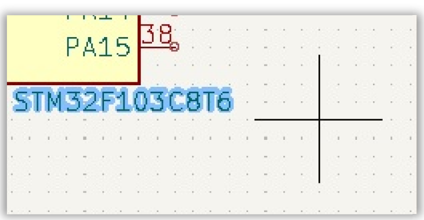
I may loosely interchange the two words, but they both mean the same in the scope of this project.
So after clicking on the create a symbol icon we will be greeted with a menu asking about the library,
Click yes or ok to proceed and wait a moment for the symbol library to load.
When the library loads, we will type STM32F103C8Tx in the search bar located at the top of the screen.
Scroll up/down until you find the file with the exact name and matches the image below.



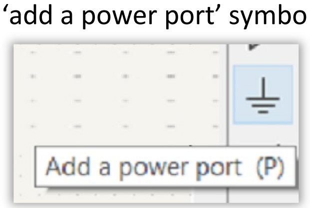
You can double-click on the file or press the 'OK' button to proceed to the canvas. Now that you are back on the canvas you notice the stm32 symbol attached to your mouse cursor. This means that you can click anywhere within the canvas window to place the symbol on your canvas.



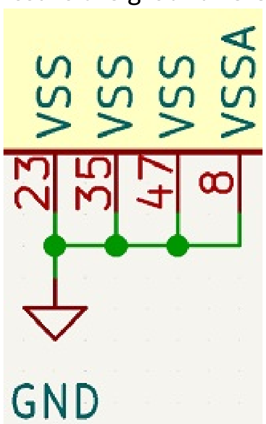
Next change your selection the pointer in the ribbon on the right of the screen, above the symbol placement icon. Now double-click on the text STM3F103C8Tx and change the 'x' letter at the end to the number '6'. The x was merely a placeholder in the filename.



Now we will add a ground and a power symbol. To do this we will select the 'add a power port' symbol that is located just below the symbol placement icon.



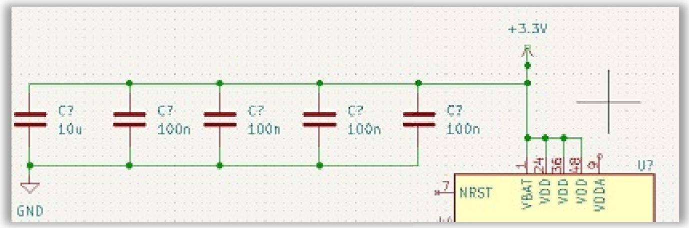
Once the power symbol library loads we will choose our ground symbol by typing GND into the textbox located at the top of the screen. We will notice the file present itself within the list of other files. Select the GND file by double-clicking it or by pressing the 'OK' button. The symbol looks like an upside down triangle with a line come from the base. Select this symbol.
Next place the GND symbol just below the 23 vss pin located at the bottom of the STM32 symbol. After doing so, click on the small circle on the GND symbol and move the mouse up and click again to create a small line. Connect this line to the 4 VSS labels as indicated in the image below. This will create a wire to connect the ground to the vss pins. The vss is the ground supply for the digital side and the vssa is the ground reference for the analog side.



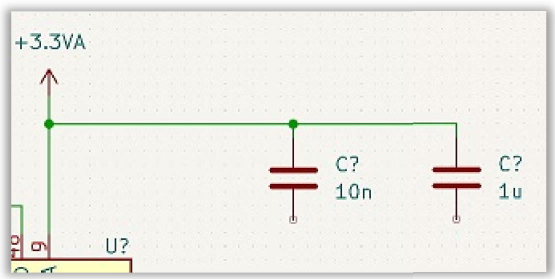
We will now add our power symbol and connect it to the proper ports. Click on the power icon symbol to access the power library and type +3.3v in the searchbox to bring up the list and select the file that matches +3.3v by double-clicking the file or by pressing the 'OK' button. Now you can place it above the #1 port located at the top of the STM32 symbol. At this point a wire must be created to connect the 'POWER' symbol to the Following ports 1,24,36,48 also labeled as VBAT,VDD,VDD,VDD.

After connecting the wire to the vdd ports we will no add a few capacitors With the value of 100n or 100 nanofarad. Click on the symbol library icon to bring up the library and type the letter 'C' in the textbox and the file capacitor should populate within the list. Select it by double-clicking the file or by pressing the 'OK' button. Click just above the #7 (NRST) port to place the capacitor. You must double-click on it and change the value to 100n. This means 100 nanofarads. Now right click on the newly place capacitor and select the 'duplicate' option to create another capacitor just a little to the left of the first capacitor, space them so they don't overlap one another. Repeat this process until you have 4 capacitors in a row, with the value of 100 nanofarads. Now create another one to left of the last capacitor and change the value of this 5th capacitor to 10u (10 microfarads). Create a a GND right clicking the first power symbol and selecting the duplicate option. Now place this GND symbol close to the bottom circle (pin) of the 10u capacitor and create a wire system connecting the GND to all 5 capacitors using the bottom circle point to connect them to the GND symbol.

Now connect the top small circle of the 5 capacitors to the +3.3v Power Symbol that is connected to the vbat & vdd pins. As seen in image below.

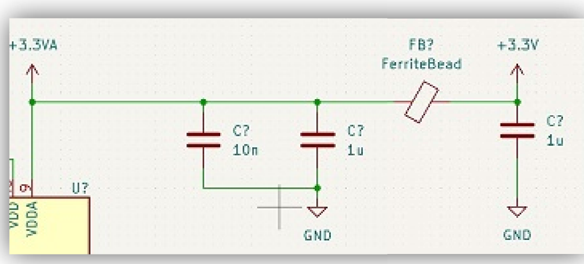


We will now add symbols to connect to the vdda port. We will create a 10n (nanofarad) capacitor & a 1u (microfarad) capacitor from the power symbol library. We must manually assign the value of these 2 capacitors individually. Click on the 'POWER' symbol icon to create a new symbol. Type in and select the file +3.3va and connect it to the 2 most recent capacitors that you placed as seen in the image below.



We will now add a Ferrite Bead. The purpose of the Ferrite Bead is to dissipate heat at high frequencies. Click on the add a symbol icon when the library loads go to the textbox and type in bead. You will see the filename, ferritebead. Select this file by double-clicking it or by pressing the 'OK' button. Now place a 1u capacitor to the right of the Ferrite Bead. Add 2 GND symbols as well as a 3.3v symbol.

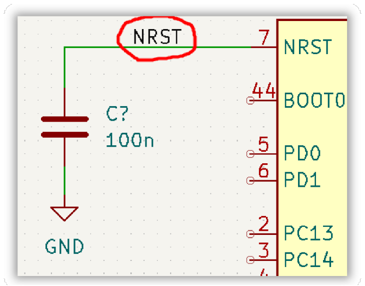
Connect these components together with a wire as show in the image Below.



FerriteBeads are typically given a resistance @ 100mhz. The Bead used here will be a 120ohms @ 100 mhz. So double-click on the FerriteBead and change the value to 102R.

Let's discuss the #7 (NRST) pin. N reset pin is pulled low it will put the microcontroller into reset mode. This pin is usually pull high so that the to make sure that the microcontroller runs the software or program that is flashed to it. You can connect it to an external tactile switch to perform a manual reset or connect to an io pin.

One thing that we can do if we decide not to use this pin is to attach a 100n capacitor to this pin to to prevent serious resets. So let's add that Capacitor and connect the top terminal of the capacitor to the NRST pin And place a GND symbol on the bottom terminal of the capacitor.
Lastly click on the net label icon in the ribbon on the right side of the screen.
When the menu pops up type NRST in the textbox and press ok. Now place this text label right above the wire that connects the NRST pin to the capacitor.



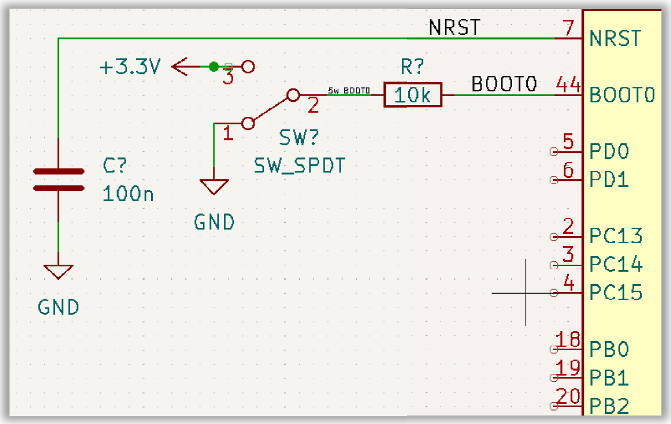
Now we will take a look at the BOOT0 pin. This pin enables/disables the internal Boot loader of the STM microcontroller. The main way to program this microcontroller is via jTag or serial wire debug. Serial wire debug and to program an STM microcontroller you will typically need an ST-Link debug probes they connect on one side to your host pc via USB and the other side via the serial wire debug signals to the microcontroller. You can also set hardware breakpoints, you can read live variables. This is not possible on Arduino devices.

If we don't want to use those methods to debug the device we can usually enable the internal Bootloader of the microcontroller by pulling the BOOT0 pin HIGH. This enables interfaces like UAT or USB to be able to program this microcontroller.

To do this via USB, we also want to use this capability to flash programs via usb to this device. In order to do this we must pull BOOT0 to HIGH, plug in the device and then it will appear as a programmable device to your host pc if you have installed the drivers. If you pull BOOT) LOW This microntroller will run the program that is currently flashed to it. To make this BOOT0 pin variable we can apply a switch.

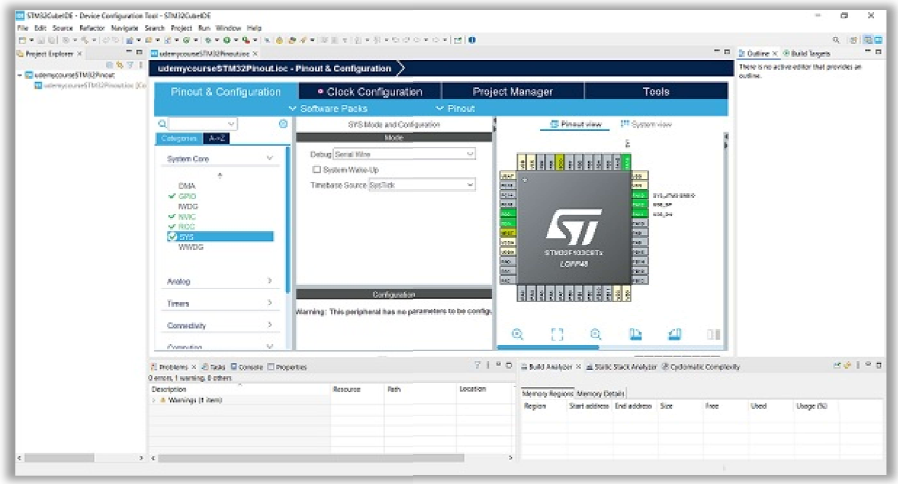
Go to add symbol icon and when the menu pops up type spdt in order to pull up an SPDT Switch. Select this file by double-clicking it or by pressing the 'OK' button. Place it a little ways away from the BOOT0 pin. After placing the switch press the 'X' button to flip it so that it matches the image below. In short SPDT simply means single Pole, Double Throw, referring to a type of electrical switch with one input (pole) that can switch to one of two different output connections (throws).

Next go to the add symbol icon and type the letter R in the textbox to bring up a resistor. Select the resistor and place it next to the BOOT0 pin, press the 'R' button to rotate the resistor. Now connect it to the #44 BOOT0 pin with a wire. Double-click on the resistor and change the value to 10k. Taking a look at the switch you will notice a 1 and a 3. The 3 reflects the HIGH and is the 3-volt (3.3v) current that will pass through. The 1 represents the BOOT0 pin is set to LOW. Click on the switch and press Y to clip the numbers so that the # is on top and the 1 is on the bottom. Add a GND symbol to the 1 side of the switch, and a +3.3v power symbol to the 3 side of the Switch. Connect the switch to these symbols with wires as seen below. You will also add Next labels to each side of the switch. Net labels are located in the ribbon on the right side of the screen. Label the first one BOOT0 and place it above the wire connecting the resistor and the BOOT0 pin. Add another net label, except this time shrink the text size down to something small like 0.5mm and type in the textbox 5w BOOT0. Place this 2nd net label on the other side of the switch slightly above the wire that connects the resistor to the '2' pin on the switch.



At this point I would like to point you to a very important program for completing the construction of this STM32 board. The program is called STM32CubeIDE it is an integrated dev environment for STM32. Open a web browser and navigate to the following page https://www.st.com/content/st_com/en.html You must create an account before you can download the program. You will receive an email with the link to download the software shortly after registering your account. Download and install the software. When the install is finished, open the software. When the program loads go to the top left corner and select 'File', 'New', STM32 project.
You will now be presented with a STM32 target selector. Select the MCU/MPU Selector tab, and enter The part number STM32F103C8T6, you will see this part populate in a list, click on it then click next. Give it a project name and click Finish.

Once you get to the screen where it shows a microcontroller pinout take a quick glance at some of the pins, some should look familiar. Go to the left tab to make a few changes.



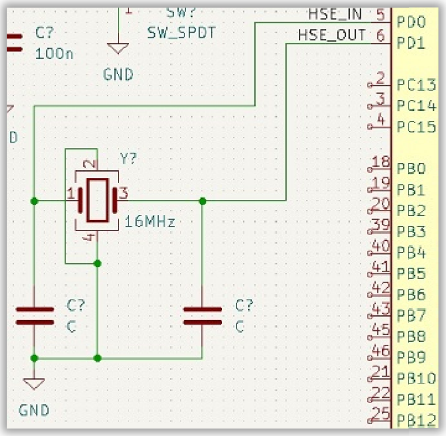
Select SYS in the system core tab, click on debug listbox and select Serial Wire. Leave ‘SYS’ and navigate to the RCC selection, click it and set High Speed Clock to ‘Crystal/Ceramic resonator. Scroll down to the Connectivity section and click on USB. In the ‘MODE’ section click the box next to Device (FS) to set to Full Speed. Now scroll up to UART1 and change the mode t o ‘Asynchronous’. Let’s go back to Kicad. Go to the net label icon, click it to create a net label. Set the name to HSE_IN and place it next to the PD0 pin. Create another net Label and set it to HSE_OUT and place it next to the PD1 pin. We will add a few more labels to a few more pins.

Create USB_D- net label and place it next to PA11
Create USB_D+ net label and place it next to PA12
Create SWDIO net label and place it next to PA13
Create SWCLK net label and place it next to PA14

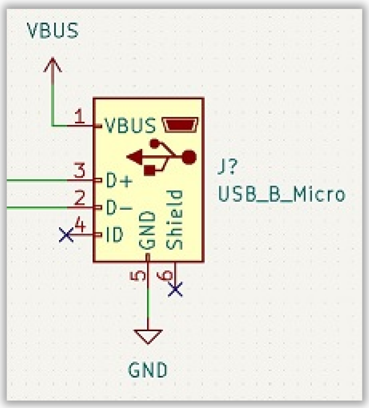
Lets go back to Kicad and get out crystal. Go to add a symbol, type crystal
And select the crystal file Crystal_GND24 by double clicking or selct ok.
Double-click on the crystal and change the value to 16MHz.
Next, make wired connection from the PD0 pin to the 1 pin on the crystal.
Now make a wired connection from the PD1 pin to the 3 pin on the crystal.

Now that the wires are connected we can move on to adding a couple load capacitors, a GND and connecting the GND to pins 2 & 4 of the crystal.
Add a GND symbol under the capacitor on the left.
Change the values of both of the capacitors connected to the crystal to 10p (picofarads)

Connect the wires as seen in the image below.

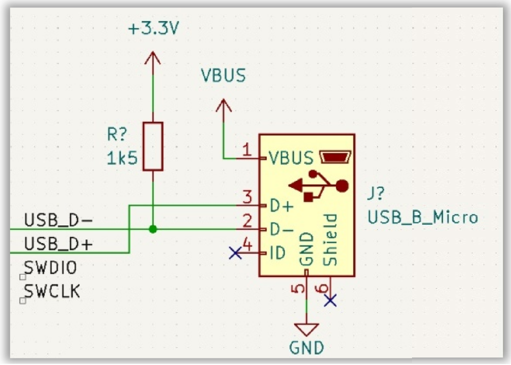


We will now add a USB connector from the add a symbal icon menu. We will be adding a micro usb connector for this setup. Type USB in the textbox and select the USB_B_Micro by double-clicking or selecting the 'OK' button. Place this usb connector on the right hand side of the microcontroller. Press 'X' or 'Y' button to flip the symbol so that it matches the image below. Add a GND symbal to the ground pin and a leave the shield floating, to do this highlight the shield pin and press the 'Q' to place a "DO NOT CONNECT " flag.
Place another one of these flags on to pin number 4. Now we must create a vbus symbol. Click on add power port icon. When the library loads, type vbus. When vbus appears select It and place it near the 1/vbus pin.

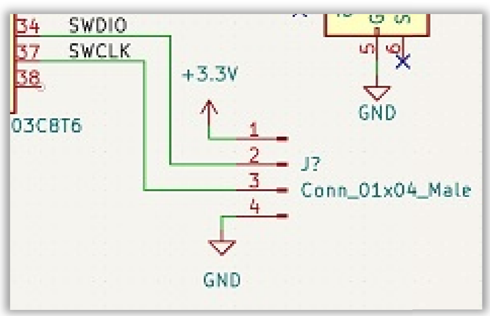


At this stage. The board still isn't ready. Lets add more symbols.
Next lets go to add a symbol and add a resistor by typing the letter 'R'.
Place the resistor between the usb connector and the microcontroller, but closer to the usb connector. There will be a picture to illustrate this.
On the lower terminal of the resistor, add a wire to connect it to the wire that connects to the D+ wire. On the other end of the resistor, and a +3.3v symbol.
Change the value of the resistor to 1k5.

Compare your schematic to the image below.

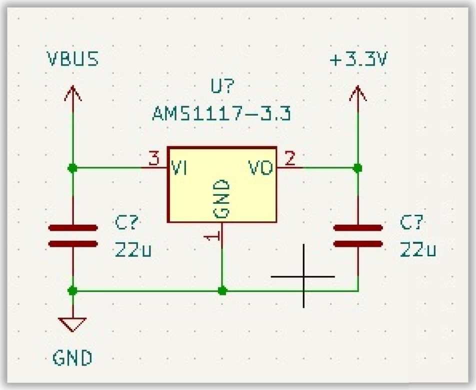


Now to get headers for the serial connections. Go to add a symbol and type conn. You want the Conn_01x04_Male file, double-click on this file to add it to the project. Place it beneath the D+ wire, give it plenty of space. Also use 'X' or 'Y' buttons to flip And orientate this connector so that it matches the image below. Add a GND symbol And connect it by wire to the 4 pin on the connector that was just added. The 4 pin should be on the left and at the bottom. Add a +3.3v symbol to the project and place it near the 1 pin on the connector and connect it via wire. Next add a wire connection from the SWDIO pin on the microcontroller to the number 2 pin on the connector. Now add a wire to connect the SWCLK pin to the number 1 pin on the connector.

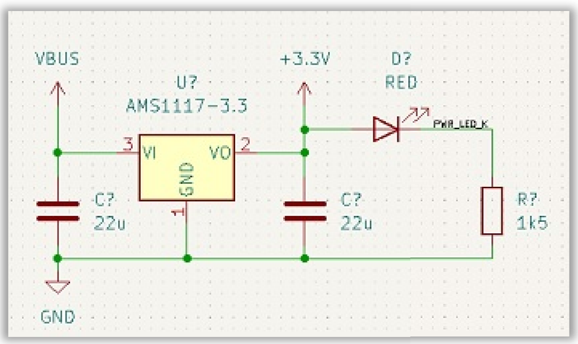


Next we will add a linear regulator because this is a low-power system. We will use a regulator is a little more than we need because it provides about an amp of current but has a low dropout voltage and it provides a 3.3 fixed output for pretty high input voltage range. Go to add a symbol and type ams1117, select it and place it far left of the microcontroller near the top left of the schematic canvas. One thing to note is that linear regulators always require input and output capacitors for stability.
We will add a capacitor and change the value to 22u and place it left of the ams1117.

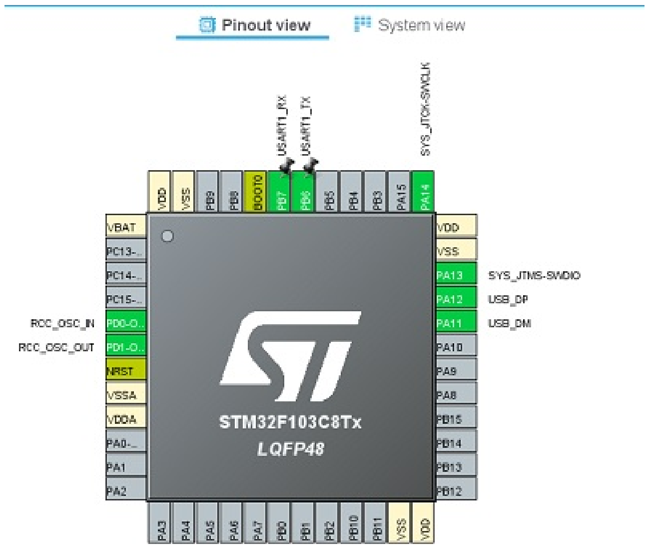
Now add a 22u capacitor to the other side of the ams1117. We still need to add a GND, a vbus and a +3.3v to this project and wire them together as shown in the image below.



We will add a Power-On LED. Go to add a symbol and type in LED, select it and place it close to the +3.3v symbol and connect it via wire. Change the value of the LED to RED. Red LED has a forward voltage of about 1.8v. So the voltage dropped across this connection will have a couple milliamps. Next we will add a resistor to prevent the LED from burning out. GO to add a symbol and add a Resistor and change the value of the resistor to 1k5. Next we will add a net label. Change the text In the net lable to PWR_LED_K. Refer to the image below.



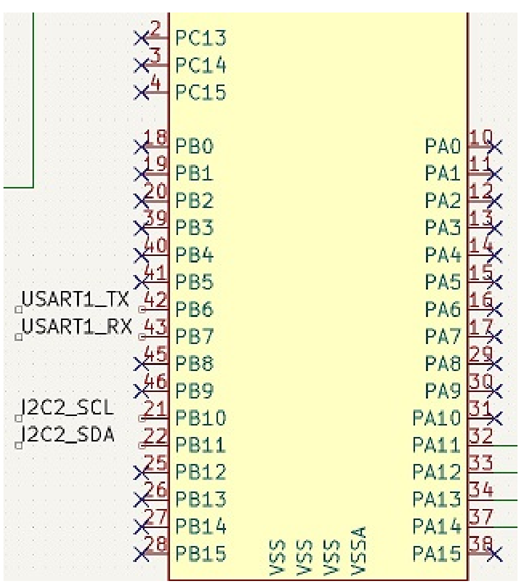
Next we will breakout some of the connections from the pin banks to the headers.
TO begin this lets go back to the STM32CubeIDE program. Go to the Pinout & configuration tab and within the Connectivity category on the left select 'UART' and change the Mode setting to Asyncronous. Next we will move some pins and change the pinout locations. To do this we will Control + click on the PA11 & PA12 pins then click on the PB7 & PB8 pins and swap them out as Seen in the image below.



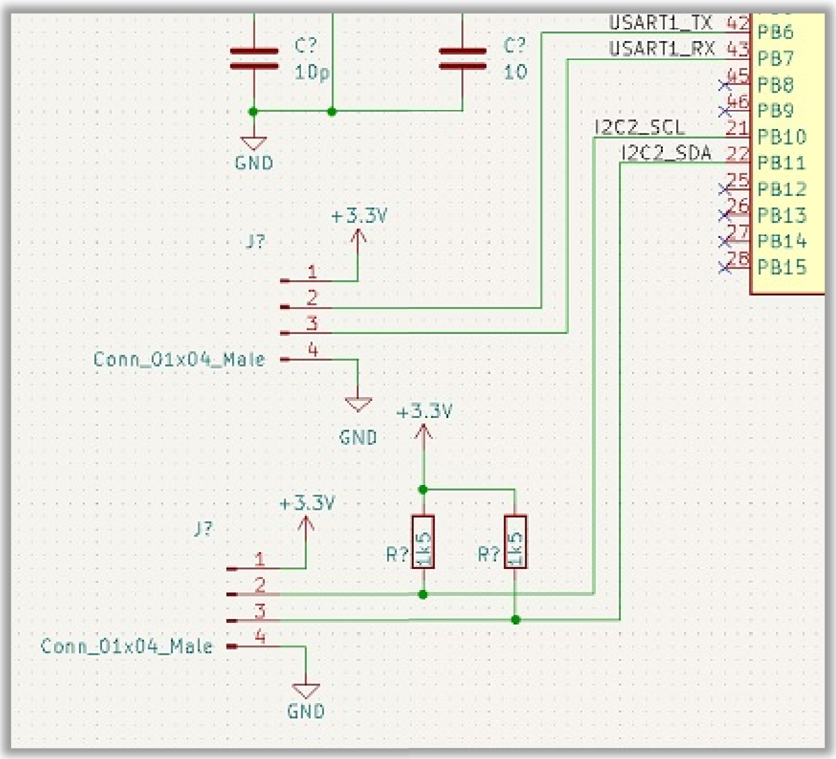
In the Connectivity column click on I2C1, change the Mode to I2C. Lets go back to Kicad and update the project with the recent changes that were recently made in STM32CubeIDE.
We will add net labels to some of the pins. Create net labels as listed below.

USART1_TX to pin 42
USART1_RX to pin 43
I2C2_SCL to pin 21
I2C2_SDA to pin 22

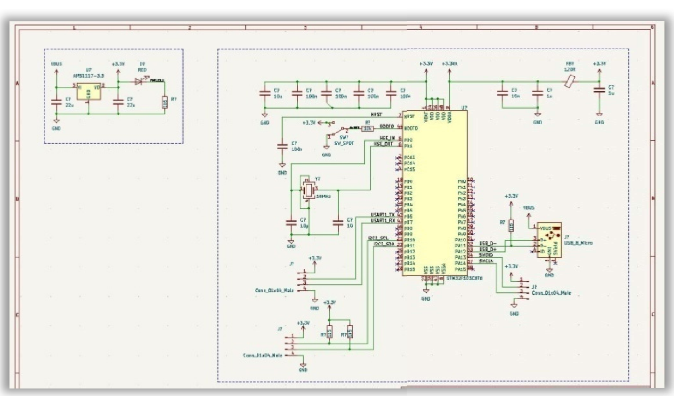
We will X= out the remainder of the pins by hovering and pressing the 'Q' button.
Compare your x out pins and net labels to the image below.



At this point we will make 2 connectors similar to the ones that we created earlier that will interface with UArt and I2C2. Copy and paste the connector, the GND and the +3.3v symbols. Move this connector beneath the linear regulator on the left side of the schematic canvas, this connector will be wired to pins42 & 43. Copy and paste another connector. This connector will be wired to pin 21 & 22 on the Wire the Also, add two 1k5 resistor and place them next to the new connector. We will wire these resistors to the microcontroller and the connector. See the image below as a guideline on wire placement.

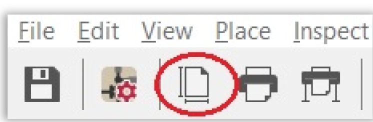


I also moved the resistor labels inside the resistor by clicking on it and pressing the letter 'R' to rotate the label and performed the drag action to place the label inside of the symbol. Alright now that we have a schematic, let's clean it up and make it easier for everyone to read and understand. To begin this process we will go to the right ribbon and select the 'add connected graphic' icon to begin partitioning the schematic. Lets draw bounding boxes or simple rectangles around our symbols as shown in the image below.

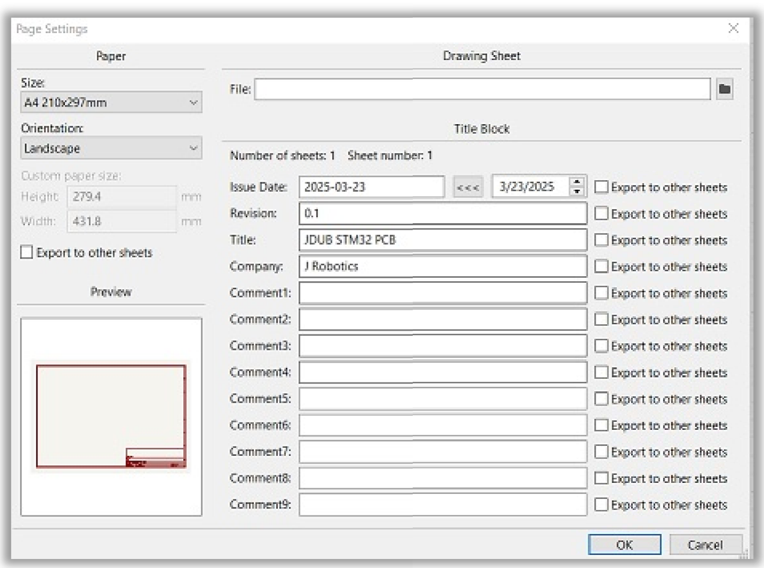


After partitioning the schematic, let's label the two sections for easier identification. To do this we will click on the 'T' icon on the ribbon, which will open a popup that contains a textbox which allows you to type what you want in that box to serve as a text label. You may want to adjust the size of the text by typing a larger number inside the textbox that is labeled 'Text Size'. Now that size is adjusted, Click inside the textbox, type Power Supply and place this text near the smaller of the two partitions so that the smaller section will display that it is the power supply. Now create another textbox, this time type the following inside of the textbox Microcontroller (and USB). Place this text near the top left corner of the bounding box. You can also make the text Bold if you feel it isn't visible enough. Switch back to the arrow pointer icon in the right ribbon and double click on the label you created and select the 'Bold' option.
Now you must add one more label to this schematic. This will contain the title, so the Text should be very large and easy to notice. It may help to make this text bold.

Next we will perform what is called a 'Title Block'. We will fill in necessary information about this schematic design that will make future production or even presentation for examples, this will give the reader a much clearer idea of what is going on before even looking at the schematic design area. To do this we will go to the ribbon near the top of the screen and select the icon that handles page settings. See the below image.

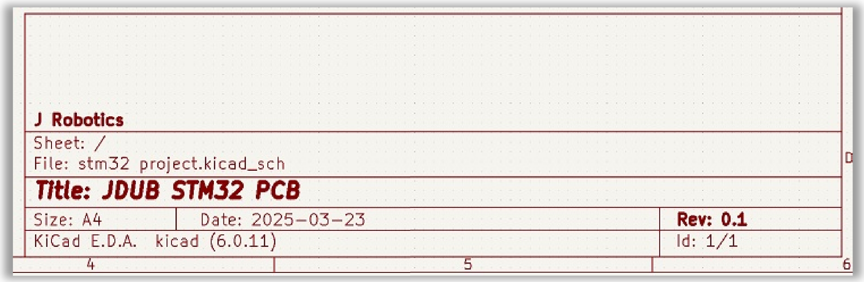


When the popup appears there will be textboxes for you to fill in the necessary data for this project. There is also an area in the popup to see live changes to this section being made in the lower left corner of the popup.



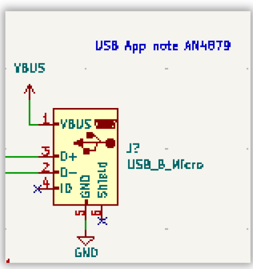
You can see the information that was typed inside of the textboxes.
This same information will now be available on our schematic in a

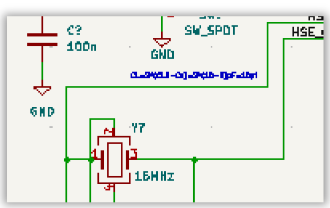
clean and professional manner. Next we will see how it looks up close. In this image we have zoomed in on the graphic. You can see how the Information that was previously entered has been formatted neatly so that it can be read and understood.

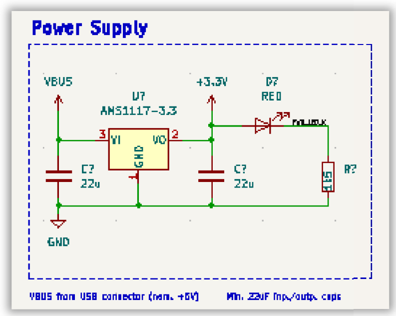


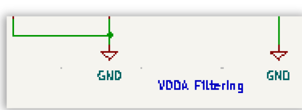
Now we will add a few text labels in the schematic to help identify what is going on in the project. This helps take out some of the guesswork that can sometimes occur in reading schematics.

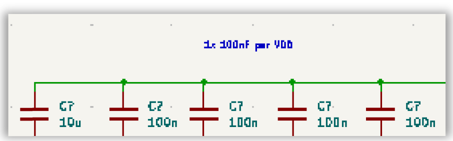
Here a few images that illustrate what I am talking about when labeling the partitions according to their purpose.

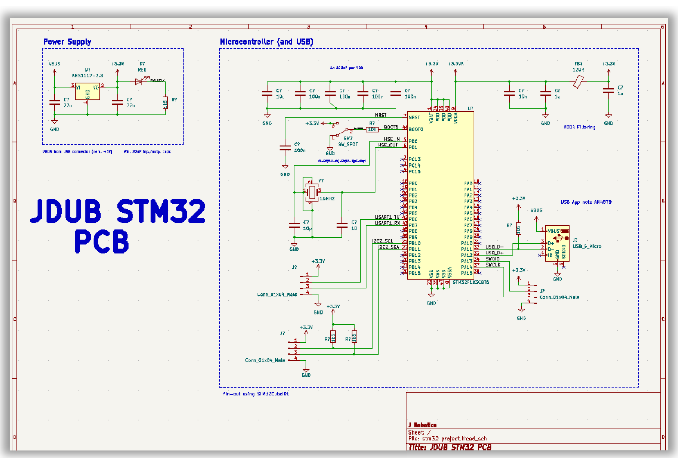






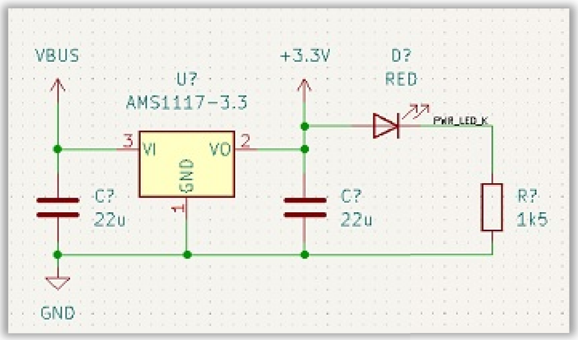




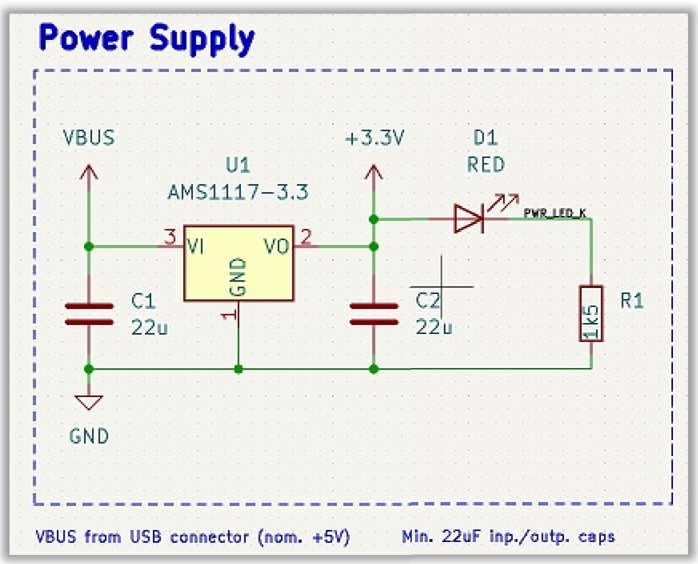


Now we will 'Annotate' to assign proper numbers to our symbols/components so that we will have a clearer idea for placement and assembly. To do this go to the top ribbon and select the icon that says 'Fill in the schematicsymbol'. After clicking this icon you will be greeted with a popup that will feature an 'Annotate' button to allow you to adjust the names of components in an orderly manner and proceed. Click the Annotate button. After the process has finished, you can press the close button. Now that you are back to your schematic window, use the mouse wheel to zoom in and out, as well as the right click button to drag the view port to another area of the schematic window. You will now notice that the (???) question marks are removed, and have been replaced with proper identifying numbers.

Compare the images, one is before annotation and one is after….



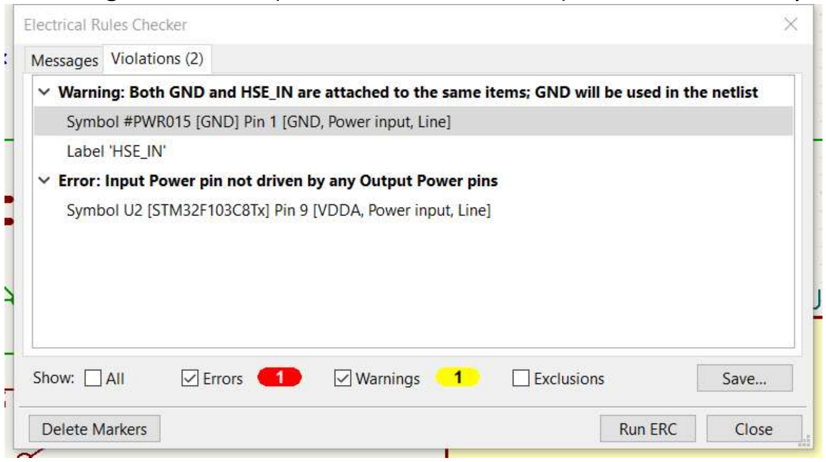
Before Annotation



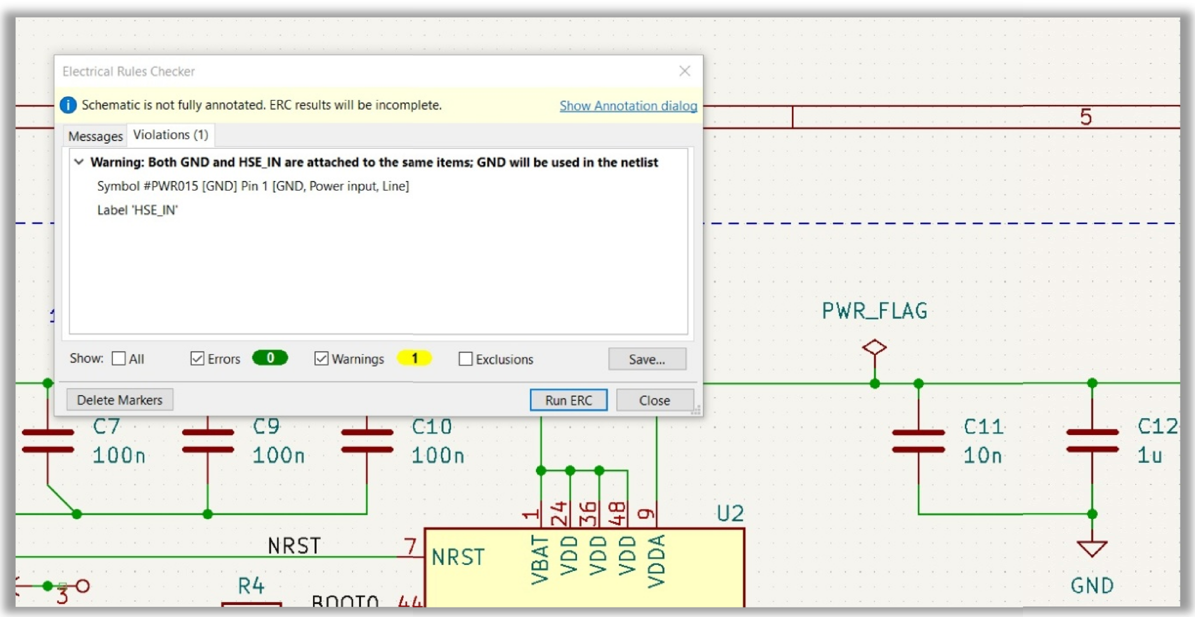
After Annotation

Next we will assign footprints to our symbols/components, The best way to describe footprints and symbols is simply this….Symbols are the entities that live on the schematic, and footprints are the entities that correspond to these symbols that live on the pcb.

Now we go to the ERC (Electronic Rules Check) to make sure everything on paper is good.

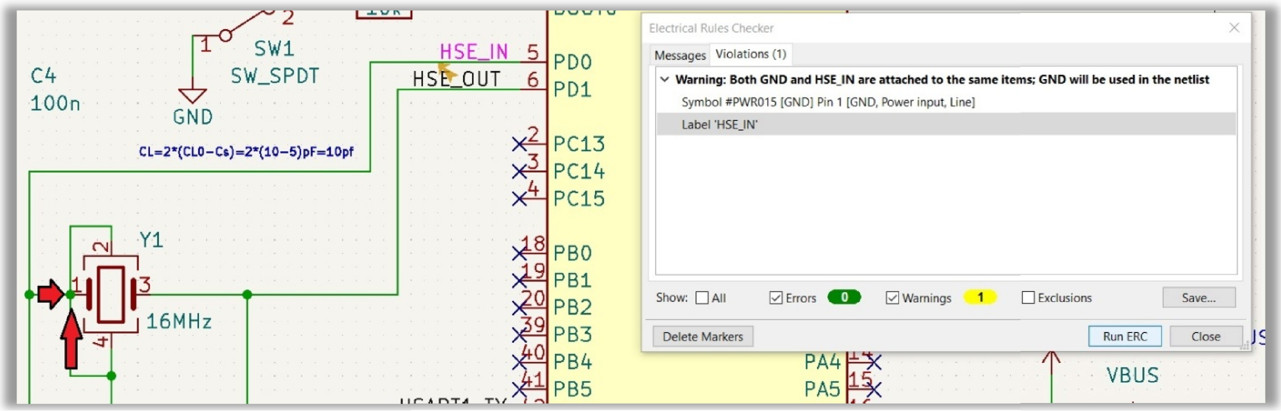


I received 2 errors. I will fix the input power pin not driven error first. To do this I will add a power flag. Go into the power symbol icon and type flag. A pwr flag file will populate from the list, select it and place it on the same wire as the FB1 resistor. Place it in the same area as seen in the image below. Notice the previous error has been removed.

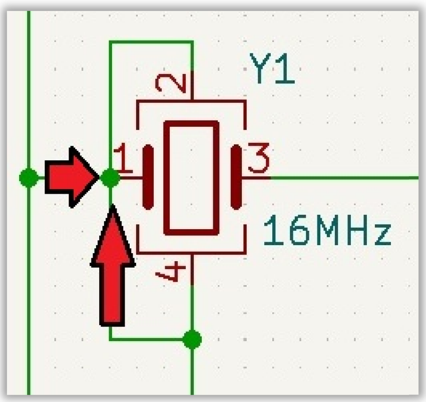


See the images below to see error 2 being fixed.

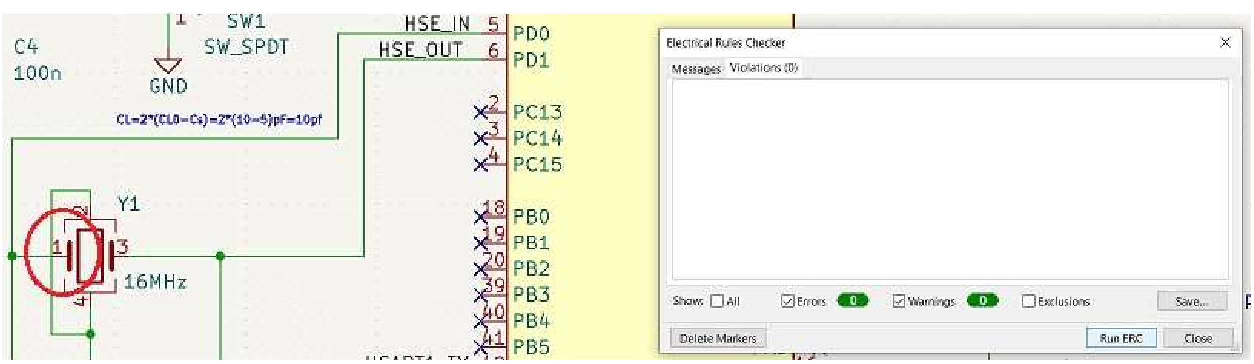
Now to fix the 2nd error. It appears that I have to remove a connection point because 2 connections are linked, and they should not be linked. To fix this, just simply click on the ball signifying a link and press the delete key on your keyboard and the error will be fixed when you run the ERC again.



The error says that GND and HSE_IN are attached to the same item. Lets remedy that.



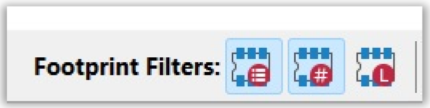
The green dot that is targeted by the arrows and must be removed in order to pass the ERC check. Once you remove it, the problem is solved.



Item is removed and the matter is resolved. We have passed the ERC and can now proceed.

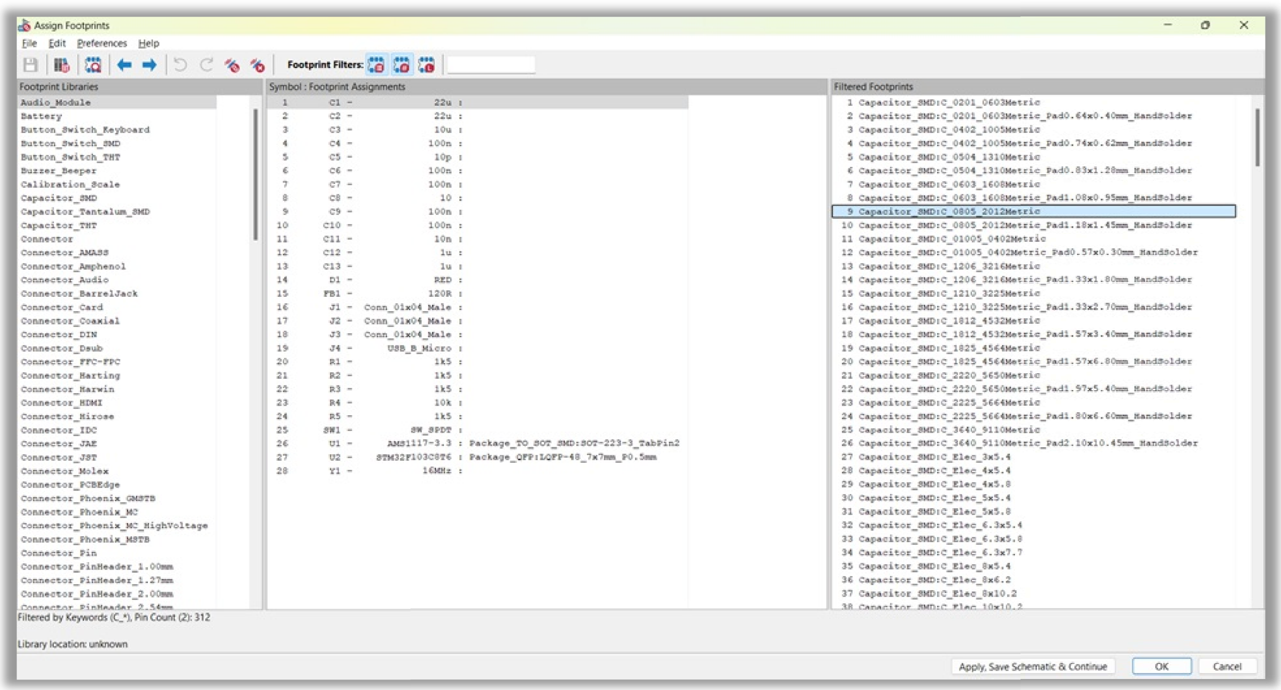
Assigning Footprints

We will now assign footprints that correspond to our relevant schematic symbols.
To begin this process we will go to the 'Run footprint assignment tool' in the top Ribbon. When the assign footprints popup appears you will see the following image



Click on these icons so that the first to icons are set as active, and the 3rd icon is not.

Now take a look at the 3 panels that we have in the Assign Footprints popup. The middle panel contains a list of all of the components that we will be using in our project. The Left panel contains lists of libraries and the right panel contains the list of available footprints.



Let's assign the first footprint. Starting at the top of our part list we will begin with a 22u capacitor. Let's select the top item, the capacitor and look in the right panel for the footprint that we need. We will choose 'Capacitor_SMD:C_0805_2012Metric'. Double click on this footprint to assign it to the capacitor that was selected. Do the same for the next capacitor.

For the 10u capacitor we will assign the footprint 'Capacitor_SMD:C_0603_1608Metric'.
I will provide an image that contains the proper footprint for each component selected.

Refer to this graphic to assign the proper footprints to your components.

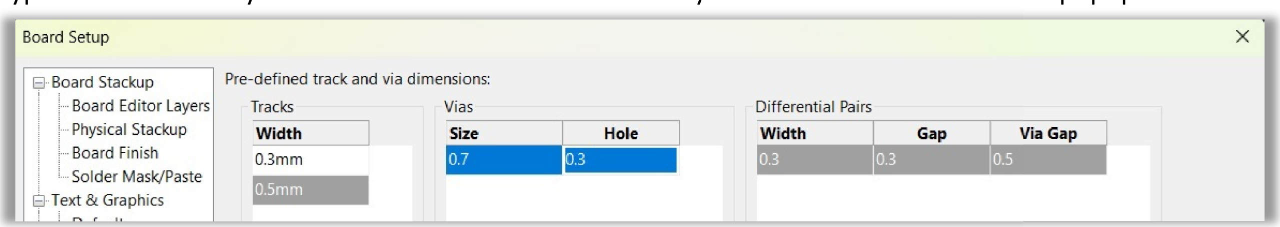
Symbol : Footprint Assignments

1	C1 -	22u	: Capacitor_SMD:C_0805_2012Metric
2	C2 -	22u	: Capacitor_SMD:C_0805_2012Metric
3	C3 -	10u	: Capacitor_SMD:C_0603_1608Metric
4	C4 -	100n	: Capacitor_SMD:C_0402_1005Metric
5	C5 -	10p	: Capacitor_SMD:C_01005_0402Metric
6	C6 -	100n	: Capacitor_SMD:C_01005_0402Metric
7	C7 -	100n	: Capacitor_SMD:C_01005_0402Metric
8	C8 -	10	: Capacitor_SMD:C_01005_0402Metric
9	C9 -	100n	: Capacitor_SMD:C_01005_0402Metric
10	C10 -	100n	: Capacitor_SMD:C_01005_0402Metric
11	C11 -	10n	: Capacitor_SMD:C_01005_0402Metric
12	C12 -	1u	: Capacitor_SMD:C_01005_0402Metric
13	C13 -	1u	: Capacitor_SMD:C_01005_0402Metric
14	D1 -	RED	: LED_SMD:LED_0603_1608Metric
15	FB1 -	120R	: Inductor_SMD:L_0603_1608Metric
16	J1 -	Conn_01x04_Male	: Connector_PinHeader_2.54mm:PinHeader_1x04_P2.54mm_Vertical
17	J2 -	Conn_01x04_Male	: Connector_PinHeader_2.54mm:PinHeader_1x04_P2.54mm_Vertical
18	J3 -	Conn_01x04_Male	: Connector_PinHeader_2.54mm:PinHeader_1x04_P2.54mm_Vertical
19	J4 -	USB_B_Micro	: Connector_USB:USB_Micro-B_Wuerth_629105150521
20	R1 -	1k5	: Resistor_SMD:R_0402_1005Metric
21	R2 -	1k5	: Resistor_SMD:R_0402_1005Metric
22	R3 -	1k5	: Resistor_SMD:R_0402_1005Metric
23	R4 -	10k	: Resistor_SMD:R_0402_1005Metric
24	R5 -	1k5	: Resistor_SMD:R_0402_1005Metric
25	SW1 -	SW_SPDT	: Button_Switch_THT:SW_E-Switch_EG1224_SPDT_Angled
26	U1 -	AMS1117-3.3	: Package_TO_SOT_SMD:SOT-223-3_TabPin2
27	U2 -	STM32F103C8T6	: Package_QFP:LQFP-48_7x7mm_P0.5mm
28	Y1 -	16MHz	: Crystal:Crystal_SMD_3225-4Pin_3.2x2.5mm

Now that we have placed our symbols in the schematic & assigned footprints, the next step is to prepare our pcb board layout for routing. To do this we will navigate to the top ribbon and click on the icon that says 'Open PCB in board editor'. You will be presented with a dark empty canvas. Now lets change that. To do so click on the button located in the far left side of the top ribbon that says 'Edit Board Setup'. You will notice a panel on the left that contains different settings to modify. We will start by clicking 'Board Editor Layers', then looking down at the middle panel and navigate to the 'B.CU' layer. Next to it is a value that can be assigned. Select the value 'power plane'.

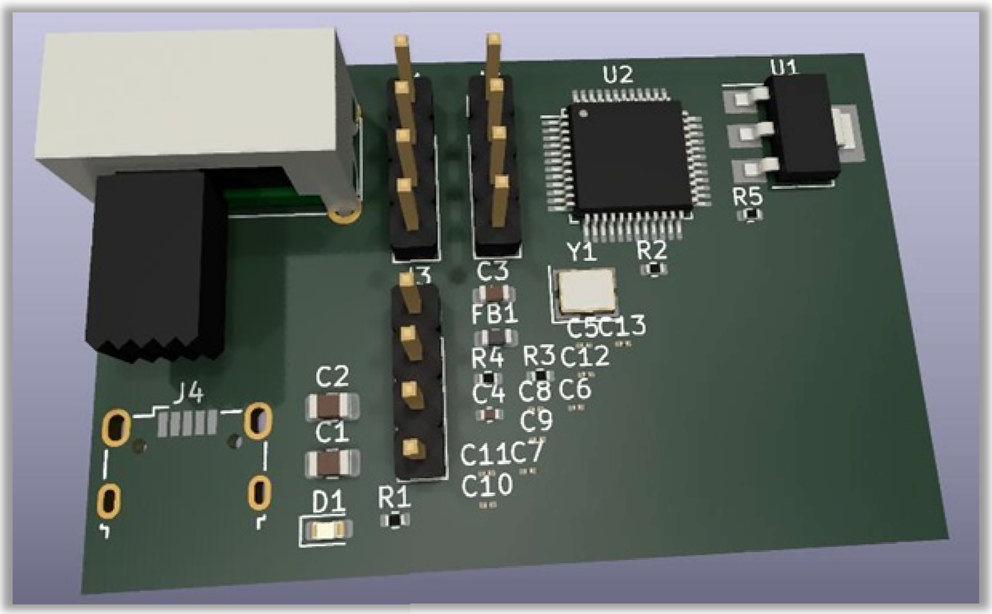
Let's take a look at the Design Rules. These rules are there to make sure that the board is able to be manufactured. We will move down to pre-defined sizes and setup sizes.

Type the values that you see in the boxes into the fields in your 'Pre-Defined Sizes' menu popup.

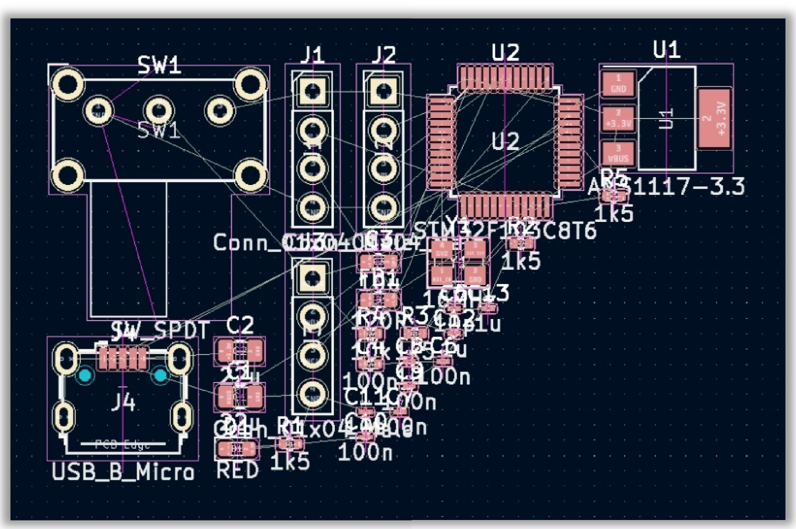


After you have entered those settings, you can press the 'OK' button to return to the pcb design canvas. Let get ready to import our designs and settings. To do this navigate to the top ribbon and click on the 'Update PCB' button. After clicking, a popup will appear showing the changes and updates for the pcb. Click on the button 'Update PCB', then click the 'Close' button. You will notice a graphic board attached to the mouse cursor. You can place this board by clicking anywhere to begin.

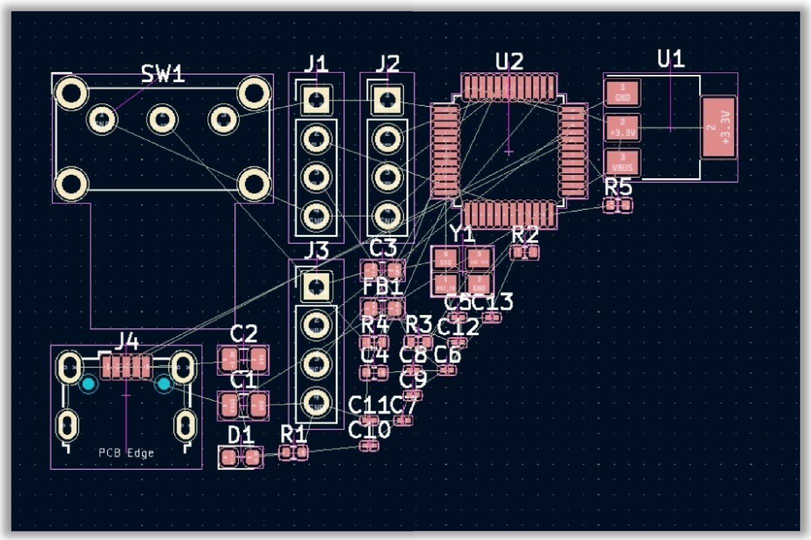
Go to the top ribbon and change the grid size to 0.5000mm. At this point we can see what our board currently looks like by navigating to the ribbon at the top of screen and select 'View', now select the option to view the board in 3d. You can move this board around in free 3d space to see how progress is coming so far. There are still a few important things that we need to do before this board is usuable.



Now we will have a look at our PCB and begin cleaning things up and adding traces.

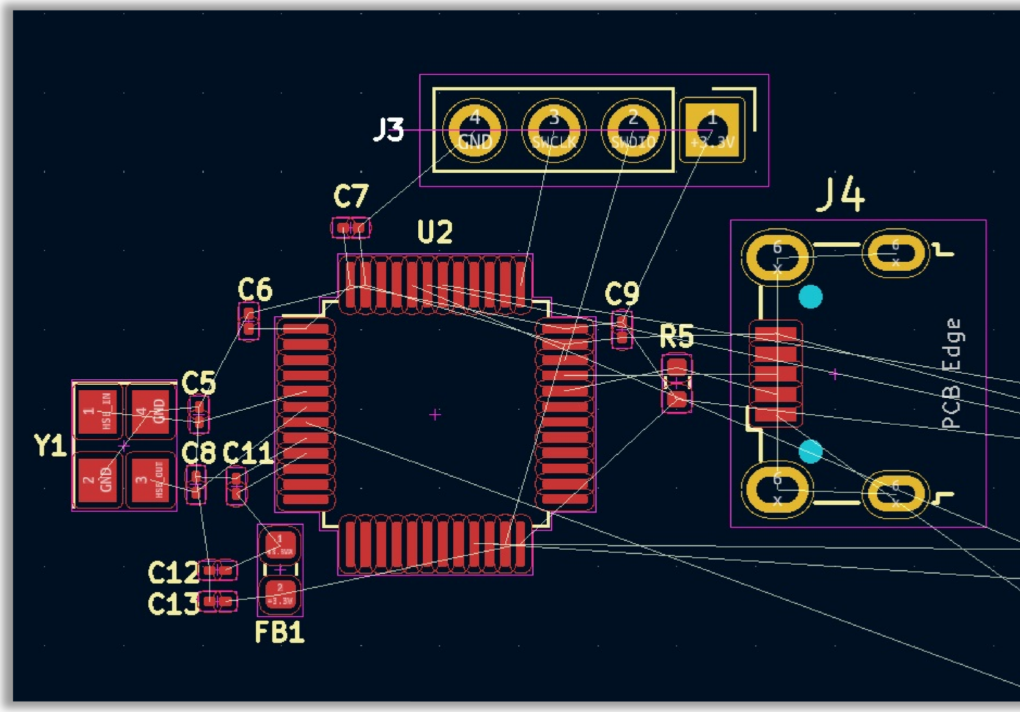


On the right side of the screen you will see a long list of layers. Find the 'F.fab & the B.fab" layers and click on the eye icon next to them to place them in 'hidden mode'. This will allow you to see the pcb without extra info getting in the way. Here is an image with the previous layer placed in hidden mode.

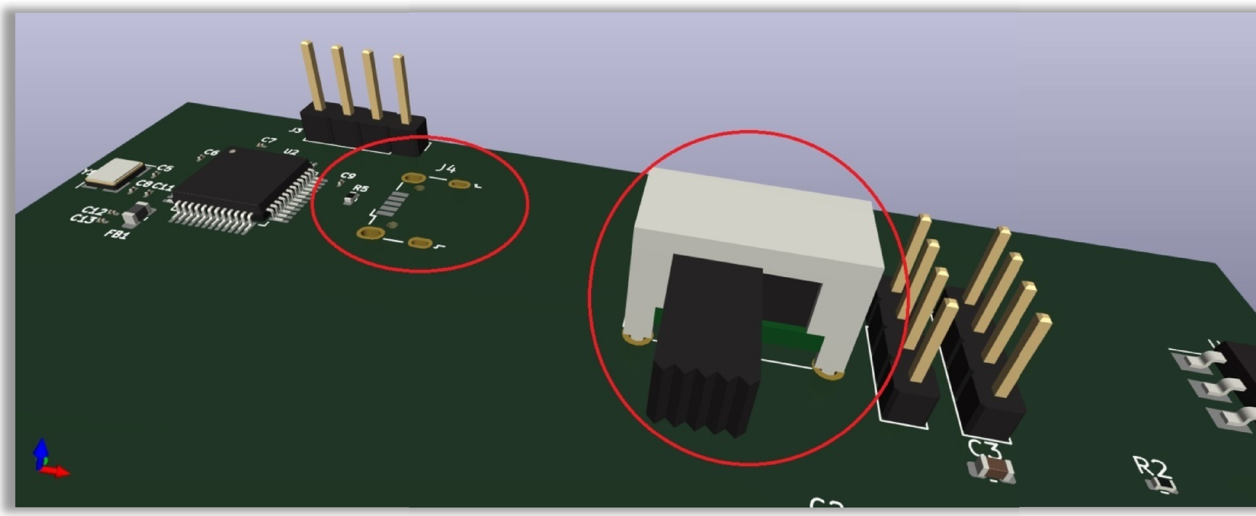


This image has much less visual clutter. That will make it easier to complete this part of the project. Next we will begin moving components, we will start out by moving the microcontroller to the left and giving it plenty of space away from the rest of the components.

You can press R to rotate an item or text. Now organize your components so that they match the image below.



Upon viewing the 3d model it is discovered that there are a couple errors that need to be addressed. The first is that there is a missing 3d object, and another 3d object is too big. They are circled in the image. Now we will provide solutions.

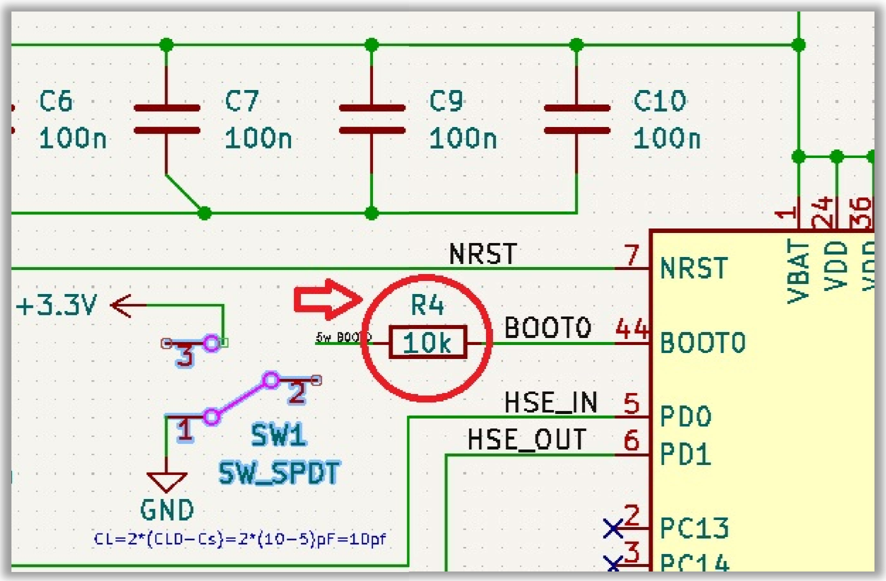


We will first fix the issue with the part being too big. Lets head to the schematic editor and double click on the switch. Go into the footprint section and assign the new footprint SW_SPDT_PCM12. Now go to the ribbon at top of the screen and choose the icon'Update pcb". Upon doing so the 3d file will be replaced with a much smaller switch. Now to fix the other issue about the missing part. We will go to the manufacturers website and get some information/files to load into Kicad and get a part that contains a 3d model.

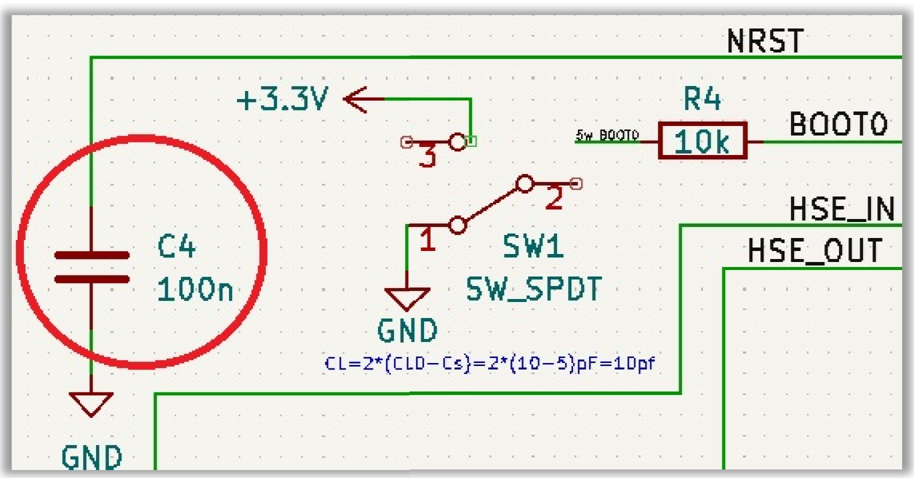
At this point you must navigate to the company website and type in this part number 629105501521 Select this part. Now scroll down to the downloads section and download the .stp file in the CAD section of the downloads. After downloading the file, place it in the same directory as your project.

Go back to the pcb design and double click on the usb plug that doesn't properly work. Go to the '3d model' tab and right click and delete the file that is in place. Now click on the folder icon and navigate to where you download the .stp file and select it. You now have a viewable 3d part that you can use. One last thing to do while we are in this menu. Go down to the 'Offset' section and change the y & z values from 0.0000mm to 1.0000mm. Now the part is properly seat on the board.

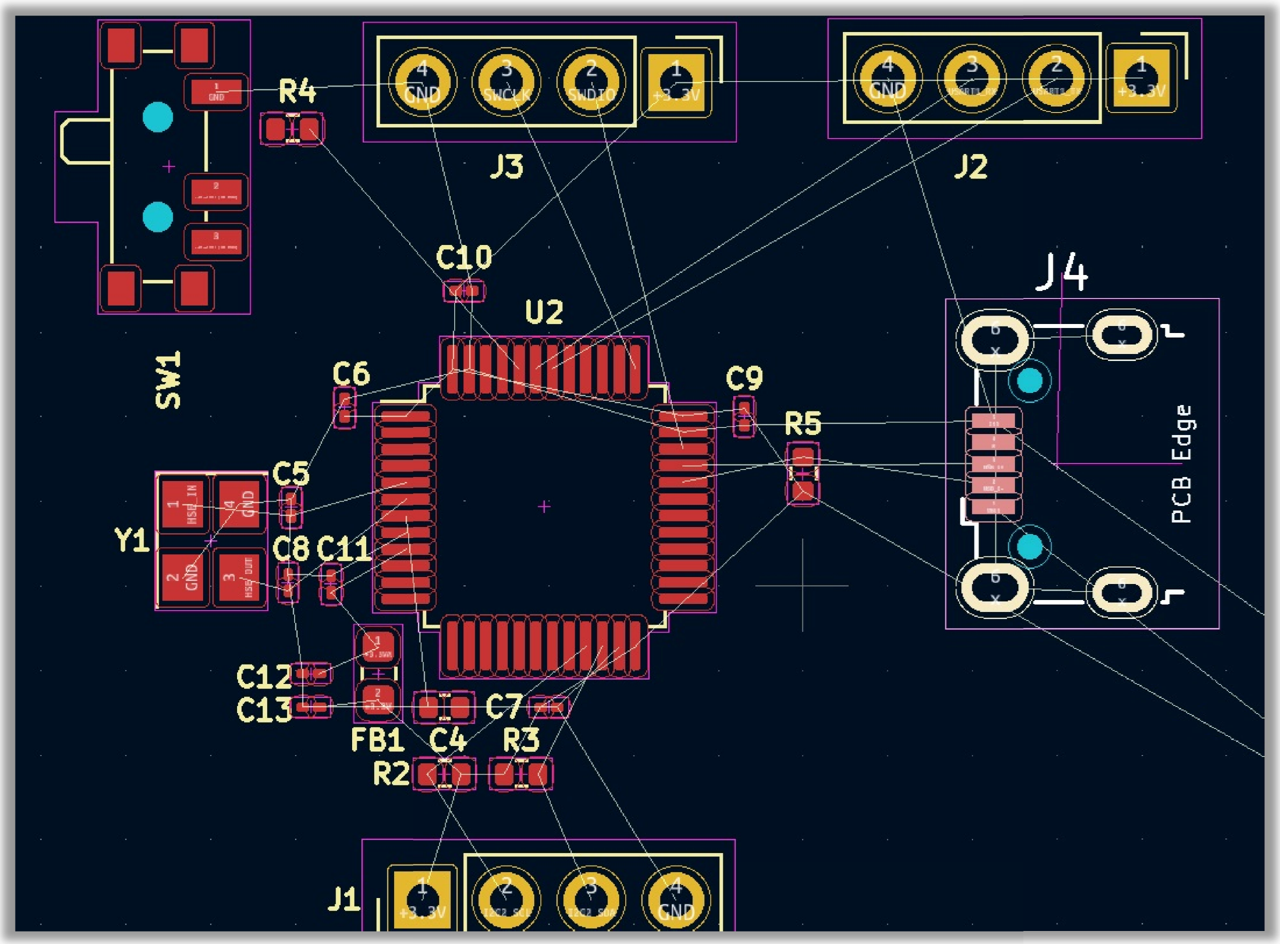
Press Ok, then go to the 3d viewer to see how the board looks with the new addition of the usb connector. Go back to the schematic editor and move the switch and the resistor that is show in the image below closer to the microcontroller.



Move this resistor near the microcontroller. Move switch near the resistor, and also move the capacitor in the image below near the microcontroller.



Move the two Uart connectors near the microcontroller. The one on the left goes above the microcontroller and the one on the right goes below the microcontroller. You will have to rotate and arrange the parts to get them to fit. Use the following image(s) as a guide.



The only thing left to do is add the traces. The board has all of the components aligned well and neat. After the traces are added, the project will be exported as a gerber file, which will then be uploaded to a pcb manufacturing website, such as Jlcpcb. The company will test your design for functionality, and upon passing the inspection your board will be created and sent to the address that was provided during the 'file upload/account setup' process.