ZReach Z-Wave Controller Reference Design

# Introduction

## Features

* Public GitHub repository – Open-Source Repository – MIT License
* KiCAD schematic & PCB layout
  + Easily imported into other tools like Altium
  + Includes gerbers for immediate production
* Roughly hockey puck in size with an antenna poking out of the center 

Not the real thing - but similar with USB cable

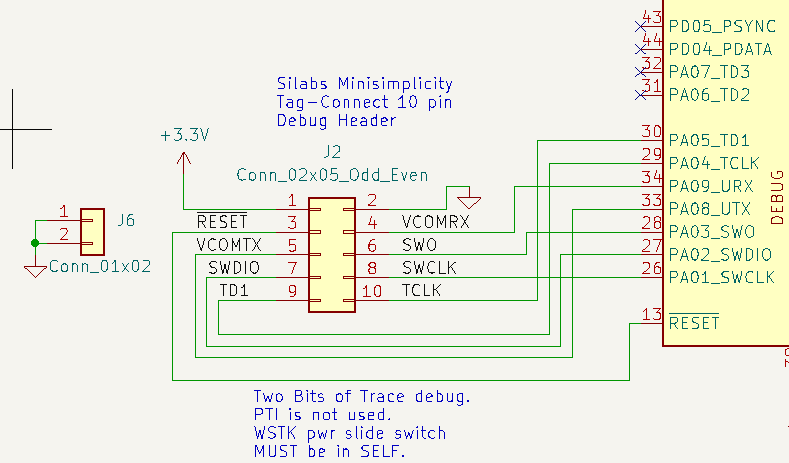
* EFR32ZG23 based +20dBm & CP2102N USB->UART
* Firmware is the binary SerialAPI in the SDK
* Documentation with full details for customization and replication
* Ideally pre-certified by the Alliance and used as a test case for the SerialAPI controllers & CTT
* Development budget: $30k
  + Schematic, PCB layout, 10 protos, functional testing, RF range testing, Documentation
  + Target Assembled/tested kit cost less than $50 (not including enclosure)
    - Target BOM cost under $10@10K
* White Paper on Antenna Best Practices with simulation results from antenna expert
* Development Time Line
  + 10 weeks from agreement to delivered prototypes

# Theory of Operation

## Z-Wave MCU

The UART Rx/Tx pins are connected to the CP2102 thru 100 ohm resistors to allow the Tag-Connect debug cable to drive the pins during manufacturing. RailTest is needed to send/receive UART commands during crystal calibration.

A Tag-Connect 10 pin header with alignment pins is used for programming and test during production and debug. Two Trace pins are wired to the header to enable Trace debugging using the adaptor board in the [etm\_zwave](https://github.com/drzwave/etm_zwave) repo. The PTI pins are not wired as Z-Wave typically uses a separate standalone Zniffer. The 2 ground pins are often needed for Trace debugging to improve signal integrity.



The ZG23 DCDC switching power supply is utilized to convert the 3.3V to 1.8V for the VDD power supply. The USB 5.0VDC is passed thru a linear LDO to drop the supply voltage to 3.3V. There will be significant power spikes of about 130mA when the radio transmits at +20dBm. The LDO and USB must be able to handle that sudden change cleanly to ensure good RF range. The PAVDD is powered with the 3.3V supply to be able to power the RF amplifier to +20dBm. The analog supply and IOVDD are on the 3.3V to provide better analog voltage range and connectivity with other external devices. The internals of the ZG23 are powered with 1.8V. Sensitive power supplies have additional ferrite beads and capacitors to ensure clean supply voltages.

## Antenna Filter and Match

The RF path from the ZG23 to the RF50 signal is straight from the Silabs DevKit reference designs. This path filters the RF and shifts the high impedance pins to be close to 50 ohms. From there to the SMA connector is a standard 3 element pi filter used to match the impedance of the antenna. The values for these 3 elements will be determined once the PCB and antenna have been tuned. A U.FL footprint is in this path to make tuning easier via a U.FL cable to a network analyzer. The U.FL is only mounted to bare PCBs to help with tuning. Thus, it is listed a NM in the schematic and is why there is a red X thru it. The SubGx1 pins are not used.



## USB-C Interface

A Silicon Labs CP2102N USB to UART chip is used to connect the SerialAPI to the host computer. A USB-C connector is used but only USB 2.0 is supported. USB3 is not supported or needed as the data rate of the UART is 115200 baud. The design follows the reference example in the CP2012 datasheet. This [article](https://hackaday.com/2023/08/07/all-about-usb-c-example-circuits/) was used as a guide.

## Power Supplies

USB-C provides +5V power in the typical application of a controller. The +5V is reduced to 3.3V via an LDO to power the on-chip switching regulator on the ZG23. The radio is powered from +3.3V to achieve a transmit power of +20dBm to the PAVDD pin.

When prototyping a low-power end device, the battery holder can be populated and connected to a CR123A or similar 3V battery. No reverse battery protection is provided so the battery holder must ensure the polarization. The LDO and USB devices are not populated when developing a battery powered device.

## PCB Ground Plane

A very important part of an IoT radio is the ground plane. Ideally the ground plane is perpendicular to the antenna and is half-wavelength. At 868MHz the wavelength is 34.5cm and 920Mhz is 32.6cm. The exact dimension isn’t that critical as long as it’s close. The ½ wavelength of 17cm but a square is more cost effective than a disk, thus divide by the diagonal (1.41) to get a 12cm on a side or roughly 5”. The ground plane fills the entire bottom side of the PCB to ensure there are no eddy currents or blockages. The next layer is used to distribute power. The upper two layers are used for interconnection of the ICs. All layers are flood-filled with ground as much of the area is open to provide a large ground plane for ideal RF performance. A smaller PCB size can be easily prototyped by simply making the PCB smaller with the potential of a loss in RF performance.

A related calculation is the width of the infeed trace. That it calculated using one of the many coplanar waveguide calculators such as: <https://chemandy.com/calculators/coplanar-waveguide-with-ground-calculator.htm>.

For [OSHPark](https://docs.oshpark.com/services/four-layer/), the relative dielectric constant is 3.61. The thickness of the prepreg is .2021mm. Usually the gap is set to .3mm then try various Widths until the calculator comes up with close to 50 ohms. In this case, it’s .44mm. I might want to lower the gap to .25 or even .2 but that might make the impedance too low.

## RGB Color LED

An RGB Color LED can be used to provide visual status of any sort. Utilizing this LED requires customizing the SerialAPI. The LED can be used when Z-Reach is used as an end-device for Indicator CC support.

## Optional Connections

Extra connections are incorporated in the design to allow prototyping of many other types of Z-Wave devices. These connectors and devices are normally NOT mounted for SerialAPI applications.

* I2C pins on a QWIIC connector easily attached many Sparkfun sensor and display boards
* 32KHz crystal supports Time CC. The on-chip low-speed oscillator drifts by a few minutes per day making it not usable for schedules or other time-of-day uses. The crystal reduces drift to a few seconds per day.
* Two 5 pin GPIO headers provide generic connections to other devices

# Reference Documents

1. EFR32ZG23 Datasheet
2. EFR32xG23 reference manual
3. Z-Reach github repository
   1. See docs directory for this document and the Z-Reach datasheet
4. [CP2102N](https://www.silabs.com/documents/public/data-sheets/cp2102n-datasheet.pdf) Datasheet

# Journal

Details of the development and timeline are described here in reverse chronological order.

## 2024-03-18 – Rev A PCB debug – Rev B update

What are the possible reasons the CP2102 doesn’t connect to a PC? The most likely problem is that D- isn’t connected or is shorted to something as pin 5 is one of the recessed corner pins. First thing to view is to see if reset is being held low during power-on. Then look at the D+/D- pins during power on and see if they toggle. Not much else I can debug as the CP2102 is fully encapsulated – no oscillator or anything else coming out. Configuration is done via USB. Check that pin 13 (WAKEUP\_N) is high which it should be. Should see the TX/RX uart pins toggle as well by the ZG23 sending power on commands. On power up I see the USB pins toggling probably sending a wakeup sequence. I also see the ZG23 sending the wakeup commands on the TX. Reset quickly follows powerup so I think that is fine. I posted this [question](https://community.silabs.com/s/question/0D5Vm000005tCdsKAE/cp2102-unresponsive) to the community page. I searched the community but didn’t find anything similar.

I might as well fix all the issues and release Rev B as I can’t see what else might be wrong. Either the CP2102 connects, or it doesn’t – no way to debug it if it fails.

Rev B issues are in github:

1. The refdes are not ordered relative to their position on the PCB. I forgot to run the reorder after placement so the refdeses are the random order they were placed on the schematic.
2. C11 and C12 are 220pF caps but have a 0603 footprint instead of 0402. The 0402 can be installed in the 0603.  
   Fixed in Schematic, not PCB.
3. The color LED is the wrong pinout in the schematic - the LED needs to be a common anode, not cathode. The footprint has the wrong pinout for the selected LED - QLSP14RGB\_B (7 cents) which is a good LED. Might have to make a custom footprint and symbol to fix it.

* A LED I've used before is: CLV1A-FKB-CJ1M1F1BB7R4S3CT ($0.38) Though it does have 3.2V forward drop which won't work if on battery power where the voltage might be only 2.4V.

1. The infeed track width to be 50 ohms should be .44mm instead of .3. Move L1 to the right just enough to get the flood fill to fill in between L1 and L2. Ideally squeeze a via in there. The traces out of the ZG23 have to start at .2mm (same size as the pad) but should widen immediately to .44mm.
2. Switch from the CP2102N QFN20 to the QFN24 package. The QFN20 has 4 pins that are below and recessed slightly within the package. The QFN20 is a standard pin arrangement so the solder joints can be visually inspected and reworked as needed.
3. The QWIIC connector footprint has thru-holes but the part is SMT.
4. Label the signals on the Tag-Connect
5. Add an optional reset switch (normally not mounted). It's just super handy during debug to be able to reset the chip by pressing a button.

Rev B upgrades the KiCAD database to V8.0.0. This required updating a number of symbols and footprints which are much more standardized in V7 – specifically pin 1 markers are in most footprints.

Somehow pins 29 and 30 are backwards in Rev A? I corrected this in Rev B. TD1 and TCLK.

The U.FL connector for whatever reason has a DRC error in it that the solder mask bridges 2 nets. I made it a little smaller. As a result, there is an exclusion that the footprint doesn’t match the one the library.

The 32KHz crystal also has an exclusion because I edited it again as the silk screen lines are too close to the pad.

QWIIC connector – easier to just modify the one I already have to be SMT than try to import one. Well, turns out in KiCAD V8 there already is a JST connector library and it has the correct footprint!

DRC Report: The 2 symbols were slightly fixed to prevent errors. The extra footprints are the fudicials.

\*\* Drc report for ZReach.kicad\_pcb \*\*

\*\* Created on 2024-03-18T15:06:45-0400 \*\*

\*\* Found 2 DRC violations \*\*

[lib\_footprint\_mismatch]: Footprint 'Crystal\_SMD\_3215-2Pin\_3.2x1.5mm' does not match copy in library 'Crystal'.

Local override; warning (excluded)

@(207.0000 mm, 114.0000 mm): Footprint Y1

[lib\_footprint\_mismatch]: Footprint 'U.FL\_Hirose\_U.FL-R-SMT-1\_Vertical' does not match copy in library 'Connector\_Coaxial'.

Local override; warning (excluded)

@(203.0000 mm, 97.2000 mm): Footprint J4

\*\* Found 0 unconnected pads \*\*

\*\* Found 2 Footprint errors \*\*

[extra\_footprint]: Extra footprint

Local override; warning (excluded)

@(174.4000 mm, 74.6000 mm): Footprint Fid1

[extra\_footprint]: Extra footprint

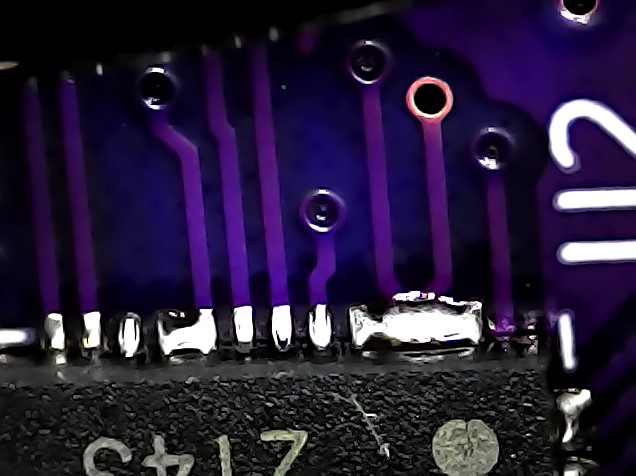
Local override; warning (excluded)

@(223.0000 mm, 145.2000 mm): Footprint Fid2

\*\* End of Report \*\*

Release board for fab via OSHPark and ordered a new stencil. Updated BOM with new RefDes.

## 2024-03-15 – Rev A PCB debug

Solder blobs on boards 1 & 2: Cleared them without too much trouble using soldering iron and a solder sucker. The USB connector was the most problematic but seems OK.

Plugging in the Minisimplicity header and powering the board via the WSTK SSv5 is able to detect the part and pull the info from it. Since it’s an A part, it seems none of the precompiled Demos are available so I have to create a SerialAPI and bootloader. Or is simply that this is a custom board so SSv5 doesn’t assume the GPIOs are where they are on a devkit? Naturally the bootloader project won’t compile because there’s no EUSART assigned. This is because all of Silabs projects are very centered on their devkits and if you don’t use one of those then you are 100% on your own with no help at all.

I built a BL and SAPI using the ZGM230 DK2603 board as the target. Flashed the binaries into the board and it powers up and has configured a DSK so it must be doing something. But it sits waiting for the BURTC to sync which it never will because I don’t have the 32KHz oscillator installed. But since I built the project using a devkit board it assumes it is setup. Ugh. Pintool doesn’t show anything in the SAPI project (or the BL). I can’t easily copy that to an empty SAPI project. I then create a blank SAPI project then try to figure out what else needs to be configured. The version via the devkit installs Platform->board->starter kit->BRD2603A which in turn enables a ton of stuff but not in a way I can then copy parts I need in a blank project. I created another SAPI project and this time it compiled! WTF? There are no VCOM or EUSARTs in the .slcp file? How is the UART being wired thru? How would we change it? It is in the pintool! Not sure how. It does appear to be running. I want to see if it’s coming out the UART. PC Controller does not find it thru the WSTK. Config/serial\_api\_config.h uses USART0 PA08 Tx and PA09 Rx.

USB is not working – the PC doesn’t detect a plugin in either orientation. Power is working. +5V VBUS comes in and the LDO makes 3.3V just fine. But nothing shows up in Device Manager on my PC. Using SSv5 Console it doesn’t connect. Not sure what that means. Is Tx/Rx backwards? The CP2102 is a QFN20 but the 4 corner pins are BELOW the package and not visible at the edge. Would be much better to use the QFN24 instead which is slightly larger but all the pins come to the edge of the package and none are recessed. They are the same price on Digikey and we have to room on the board. USB D+ pin is on the corner of the package. I would have thought the CP2102 will show up as a USB device when plugged into a computer without any interaction via the UART. I could wire an FTDI to the resistors to get something to work and check more things out. But that isn’t a good test of the RF range since so many things are hacked.

Board #1 & #2 are recognized via SWD and Commander: EFR32ZG23A020F512GM48

Ordered the missing components from the BOM. Not needed immediately. I got a bunch of antennas in the mail today.

## 2024-03-14 – Rev A PCB build

I built 3 boards. Board #3 has been cut down to roughly the minimum size to compare with the full size boards to measure the RF range difference with a smaller ground plane. Basically answering the question – does the additional cost for PCB real estate significantly improve the RF range? Overall size of the reference design isn’t that much of a concern but cost is an important factor though range is the priority.

I forgot to order 1.3nH inductors and 7.2pF caps. I had 1.2nH and 6.8pF which are close enough. I’ll order the correct ones for the next build. The QWIIC connector I ordered is SMT but the PCB is TH so need to pick a different connector. I didn’t order the 32khz crystals but those are not needed at this point. The QWIIC connectors appear to all be SMT so I’ll change the PCB and keep these components.

The solder paste seemed a bit thick and resulted in several shorts. I ordered the .003” thick mylar stencil. The only other option is a .005” so I am using the thinner one. My solder paste is a little thick as the basement is 55F so maybe warming the solder paste up will help is spread more evenly.

## 2024-03-13 – Antennas

I have a box of various antennas. However, they are not labeled and I don’t have the part numbers nor the specs. Some of them might be 2.4GHz, 868, 915. I really can’t tell. So, I ordered some and will keep the part number with the antenna so we know for sure which one is which. Digikey has 15,000 antennas! Filtering that down to SMA in the 900/868MHz bands with an SMA connector and in stock gets us down to 43. I ordered the following:

|  |  |  |  |
| --- | --- | --- | --- |
| Part Number | Price  @1k | Len mm | Comments |
| [**TI.92.2113**](https://www.digikey.com/en/products/detail/taoglas-limited/TI-92-2113/11197416) | $6.20 | 198 | Looks like the one shipped with Silabs DevKits |
| [**TS.89.4113**](https://www.digikey.com/en/products/detail/taoglas-limited/TS-89-4113/15284364) | $40.56 | 118 | Directional antenna with 3.5/2.7dBi gain in 1 direction |
| [**ANT-8/9-MMG1-SMA-1**](https://www.digikey.com/en/products/detail/te-connectivity-linx/ANT-8-9-MMG1-SMA-1/16592298) | $5.66 | 83 | Cable to a mount |
| [**FW.95.B.SMA.M**](https://www.digikey.com/en/products/detail/taoglas-limited/FW-95-B-SMA-M/6362786) | $9.13 | 226 | Thin fixed |
| [**DELTA12C/X/SMAM/S/S/17**](https://www.digikey.com/en/products/detail/siretta-ltd/DELTA12C-X-SMAM-S-S-17/6096292) | $9.65 | 130 | Classic Rubber Ducky |
| [**JCG402LR-2**](https://www.digikey.com/en/products/detail/jc-antenna/JCG402LR-2/15814458) | $2.42 | 110 | Tiltable – Low cost! |
| [**2600130083**](https://www.digikey.com/en/products/detail/w%C3%BCrth-elektronik/2600130083/11680415) | $14.88 | 179 | Covers 868 thru 928 |
| 868 antennas: |  |  |  |
| [**ANT-868-CW-HW-SMA**](https://www.digikey.com/en/products/detail/te-connectivity-linx/ANT-868-CW-HW-SMA/5592340) | $8.74 | 135 |  |
| [**FW.86.B.SMA.M**](https://www.digikey.com/en/products/detail/taoglas-limited/FW-86-B-SMA-M/6362785) | $9.13 | 240 | Thin fixed (similar to the 915 one above) |
| [**FLEXI-SMA-868**](https://www.digikey.com/en/products/detail/rf-solutions/FLEXI-SMA-868/5845738) | $5.12 | 73 | Short Rubber Ducky |

## 2024-03-03 – Component selection & ordering

Exported the BOM out of KiCAD and used it to manually create the BOM. Ordered enough components for 10+ boards. I already have 15 ZG23s from previous projects and samples. Entered an Issue on github that a couple of the caps are 0603 but should be 0402. The 0402 will still fit but better to have the proper footprint. I expect to spin the PCB at least once more.

The plan is to take one of the 3 boards coming in now and cut it to a much smaller size and compare the range. No sense in making the board bigger than it has to be but we are looking for maximum range. If the full size board is at least 3 dBm better than definitely keep the full size board.

KiCAD has been upgraded to version 8.0.0 which came out a week ago. I had to upgrade to checkin the ZG23 component as all libraries were upgraded to V8. The ZG23 will now be included in the KiCAD default libraries. My PR [4289](https://gitlab.com/kicad/libraries/kicad-symbols/-/merge_requests/4289) has already been merged in. I assume it’ll be included in the next tagged release.



## 2024-02-23 – PCB release to fab

There are 3 excluded DRC violations:

1. The Y2 Crystal footprint was slightly modified to clear the silkscreen error in the footprint
2. The 2 Fiducial marks are not in the schematic

These “warnings” can be ignored.

The PCB is 120mm square which I am certain is overkill. But we can cut a board to be smaller and see the net impact. Might also make a different proto board with a smaller PCB. The corners have 10mm radiuses to make a clean point for the enclosure to latch onto the PCB and make it easier to separate the boards when built as arrays.

Three PCBs and a stencil ordered. Expected in about a week. These 3 boards are just for initial testing to make sure the design doesn’t have some major problem like an incorrect pinout. One board will be for tuning the antenna(s), one is a proto and I expect to cut one to be much smaller to see if the large ground plane makes a difference. Might also want to make a 2 layer PCB version which is cheaper. I’ll order components on Monday.

## 2024-02-22 – PCB layout

DRC clean but in need of cleanup and more documentation.

## 2024-02-21 – Schematic capture

Schematic capture completed and ERC is clean.

Footprints to be picked out:

* USB-C receptacle – GCT [USB4110](https://www.digikey.com/en/products/detail/gct/USB4110-GF-A/10384547)-GF-A is SMT with 2 small mounting pins less than a buck
* [39MHz](https://abracon.com/Resonators/abm8.pdf) crystal – Crystal:Crystal\_SMD\_3225-4Pin\_3.2x2.5mm
* 32KHz crystal – Crystal:Crystal\_SMD\_3215-2Pin\_3.2x1.5mm - SC32S-7PF20PPM
* RGB LED – LED\_SMD:LED\_WS2812B-2020\_PLCC4\_2.0x2.0mm - [QLSP14RGB\_B](https://www.digikey.com/en/products/detail/quelighting-corp/QLSP14RGB-B/15848703)
  + Might need to make a custom footprint
* The RF RCs are 0402 which are cheap and still OK to hand solder. 0201s are tough to handle.
* Other RCs are 0603 and large caps are 0805 which makes them easy to handle and cheap.

PCB board created and a few components placed. Need to adjust the DRC rules to match [OSH Park](https://docs.oshpark.com/design-tools/kicad/kicad-design-rules/).

## 2024-02-20 – Schematic capture

Connected the USB UART to the same pins as the debug header but thru 100 ohm resistors. I could have put the USB UART to a different set of pins but then the firmware could NOT use the standard SerialAPI and would have to be customized with each release. Didn’t think that was a good idea. The series resistors are to provide some isolation when BOTH USB and the debugger are connected. The main issue is that during production, RailTest has to be downloaded and then commanded via the debug connector to the UART to Tx ON for crystal calibration. But since USB is also connected, it is also driving the UART pins. Simply adding some series resistors will provide sufficient isolation for the short time during production. Even during debug it’s not a big deal. During normal operation the resistors will help with EMI. UART speed is only 115.2K baud so the resistors will have no impact on that. The other option is to use a switch but that would be overkill and just 1 more place for things to fail and increase cost.

USB pins require ESD protection as described here: <https://www.digikey.com/en/articles/why-usb-type-c-circuit-protection-is-vital> but this is overkill IMHO. There are many triple TVS diodes but they do not seem to have a common pinout. Prices for these triple diodes are around 30 cents. Only need USB2 level ESD as we do not need to support USB3 which has tighter restrictions on the capacitance of the TVS diode. This [article](https://www.electronicproducts.com/how-to-protect-usb-type-c-connectors-from-esd-and-overtemperature/) has a good description of requirements. Want to have something with lots of 2nd sources and not be locked into a single part. KiCAD doesn’t have any diodes in 0402, mostly SOD-xxx. The CP2102 recommends SP0503BAHTG which is in the KiCAD library and digikey has 114K of them so it is easier to use this part.

The Tag-Connect debug header VAEM is tied directly to the 3.3V from the LDO. This is normally not a problem but could result in both the WSTK and the LDO driving the 3.3V pins. Since they are both at the same voltage it usually doesn’t matter. The WSTK should normally NOT drive the VAEM pin (slide the switch to USB or battery). The only time you would drive the VAEM pin is if you are debugging a battery powered device and want to measure the power which would be done using the AEM in SSv5. But in that case, the LDO would not be installed. The only question is during programming will the LDO short 3V3 to GND? That would require the USB to also be plugged in or powered via a test point which is acceptable but makes things more difficult. However, a functional test of the USB would require the cable being plugged in so maybe it is OK.

There is no QWIIC connector symbol or footprint in KiCad. I copied the ones from the sparkfun repo but I didn’t want to import the entire thing so I created project specific libraries which will also hold some other project specific stuff like the Z-Wave logo and things.

## 2024-02-16 – Project Start

Wrote the datasheet. Started Schematic capture and Theory of Operation (ToO) development.

Lots of LDOs to choose from. Diodes Inc [**AP2125K-3.3TRG1**](https://www.digikey.com/en/products/detail/diodes-incorporated/AP2125K-3-3TRG1/4470777) is 10 cents and will be the initial LDO to use (Digikey has 45K in stock). The TI [**TPS7A0333PDBVR**](https://www.digikey.com/en/products/detail/texas-instruments/TPS7A0333PDBVR/12165108) LDO I’ve used before is 33 cents but does have better Iq, temperature range and Vdropmax but the Diodes Inc is fine.

For the USB-C interface, I used this example: <https://hackaday.com/2023/08/07/all-about-usb-c-example-circuits/>. The classic question is what to connect the shield of the connector to? The Shield is the outside layer of the cable thus it can act as an unintentional antenna and then fail FCC. The example wires it to GND but that is likely to radiate more than leaving it unconnected. Nice discussion [here](https://electrical.codidact.com/posts/279876#:~:text=%E2%88%920-,Connect%20the%20shield%20directly%20to%20ground%20plane,to%20the%20PCB%20ground%20plane.) which mostly recommends connecting SHIELD to GND. There are 2 purposes for this connection – EMI and ESD. For ESD it makes sense to have a solid connection to have the entire board remain at the same potential which might shoot up thousands of volts in an ESD event. But I am more concerned about EMI which in an abundance of caution I put a small ferrite bead leading to the shield to reduce any EMI. Can replace this with a zero ohm resistor later, or leave it open or insert a capacitor or even just short the pads out. I’m not too worried about ESD as ZReach is pretty much plugged in once and left there for the duration.

## 2024-02-07 – Create Repository

Created the Repo, this file and started the hardware design in KiCAD.