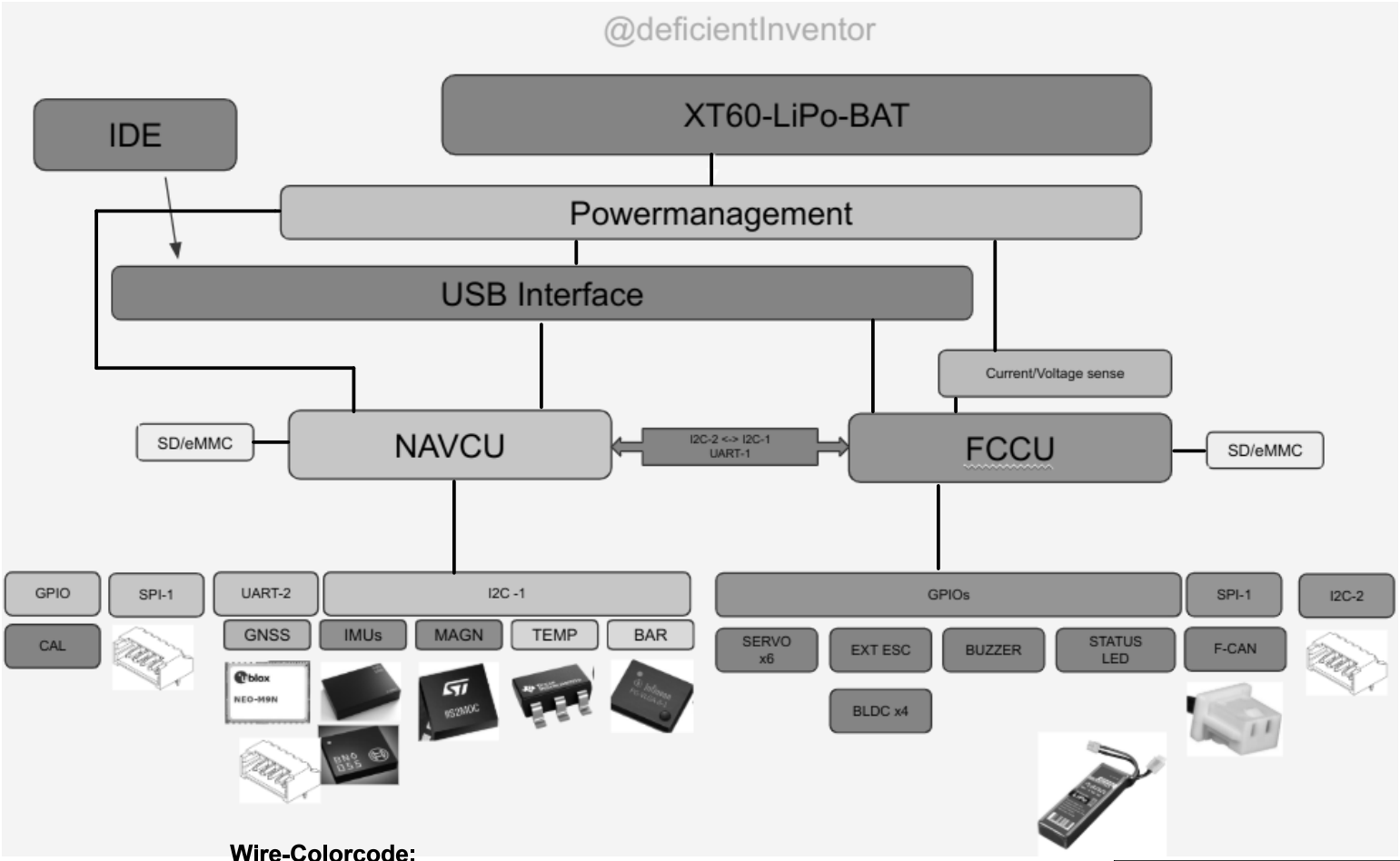



Designnotes

BATT_PWR_MANAGEMENT

FCCU_USB_INTERFACE

NAVCU_USB_INTERFACE

FCCU

NAVCU

FCCU_PHERIPHERALS

NAVIGATION_PHERIPHERALS

## Wire-Colorcode:

### POWER

| Type | Color | Width |
|---|---|---|
| VBAT | ——— | 2.5mm |
| +5V | ——— | 0.5-2.5mm |
| +3V3 | ——— | 0.5mm |
| REG_FB | ——— | 0.35mm |
| GND | ——— | 0.5mm |

### SIGNAL

| Type | Color | Impedance | Width |
|---|---|---|---|
| UART | ——— | 50Ω | 0.327mm |
| I2C | ——— | 50Ω | 0.327mm |
| SIgnal | ——— | 50Ω | 0.327mm |
| USB | ——— | 90Ω | 0.35mm |
| RF/SPI | ——— | 50Ω | 0.327mm |

### TODOS:

-Schematic Review

**Comments:**

Text-based project definition in design notes

deficientInventor

Feza

Sheet: /
File: feza_fcu_pcb.kicad_sch

***Title: FEZA FLIGHT COMPUTER***

| Size: A4 | Date: 2024-11-18 | **Rev: 1.0** |
|---|---|---|
| KiCad E.D.A. 8.0.7 | | Id: 1/9 |

# ABOUT ME / THIS PROJECT

I am a former car mechatronic technician and car wrapper, currently a mechanical engineer who started this project to create a learning platform for the next 5-10 years. My goal is to evolve into a general engineer or a full-stack engineer. The main purpose of this project is to learn Electrical Engineering in the first stage, followed by Physics, Math, Control Algorithms, Software Development, Robotics (e.g., Sensor Fusion, Thrust Vector Control, Reaction Wheel), and many more fields of engineering in the later stages.

The main purpose of this PCB is to serve as the primary development platform to learn all these interesting fields and act as a general-purpose computer. It could be used as a model rocket, fixed-wing or multicopter flight computer, or literally any kind of robot. Due to the CAN bus capability, it is possible to design dedicated task-focused PCBs and use this board as the central brain of the system. Surely, there will be iterations if flaws and hardware limitations are noticed during further development. I'm sure there are better ways of designing a board like this, but I want to mention that I'm not an electrical engineer, and this is the best I could achieve based on my current knowledge gained during the last ~200 hours of schematic design. A lot of this schematic is designed by referencing other schematics and trying to understand the reasoning by looking at the specific datasheets of the used components and asking relevant questions in subreddits like r/PrintedCircuitBoards, r/AskElectronics, r/Robotics, r/Electronics, and r/Arduino. The main reference is the Hades flight computer from Phil's Lab, and the peripherals are kind of mixed references from Reddit, Adafruit, SparkFun, and so on. The idea of choosing these specific components is that there are already ready-to-use libraries to start developing via PIO, ESP-IDF, and the Arduino Platform because I don't have the ability to write bare-metal programs yet. This will be important in later stages and will help me learn bare-metal programming by looking at existing libraries and figuring it out.

# DESIGN CHOICE:

## POWERMANAGEMENT

This PCB is going to be powered by a LiPo and a switching regulator capable of 5V @ 9 Amps max. This should be sufficient to power at least three servos with a stall current of 2.5-3 Amps; however, the correct amount of servos will be known after doing real measurements. The ESCs are going to be connected to the servo connectors, and they will have their own LiPo. I'm most likely going to design a slave PCB via CAN bus to operate the servos' BLDCs, which have decoders in later stages. The switching regulator part is the most complex topic for me on this schematic and is a 1:1 copy of Phil's Lab Hades flight computer.

There are two LDOs: one for 3.3V logic level (MCU and sensors) and one with 1.8V logic level for the CANSilent level shifter of the CAN transceiver because two pins of the ESP32S3 have 1.8V logic, and only one of them is used; all the others are NC (not connected). There is also an INA219 current sensor.

## USB-INTERFACE

The USB-Interface is simply a 1:1 copy of the dev board schematics; the only difference is that I used USB-C ports instead of Micro USB and added TX/RX transmission LEDs to the USB-UART converter. I decided to keep the auto programmer logic for development purposes so I can reduce the ESD risks while handling it and create a 3D-printed housing for this PCB.

## NAVIGATION CONTROLLER UNIT

The NAVCU (Navigation Controller Unit) serves as the central brain that communicates with the FCCU via UART and I2C. It handles the computation-heavy sensor fusion and has a lot of peripheral sensors. It is dedicated solely to navigation.

-There is an SD card serving as a read/write eMMC to do data logging and, if needed, to save calibration data on it to avoid saving stuff in the internal EEPROM of the ESP. I chose to put a dedicated SD card slot for the NAVCU during the development stages to log a more detailed log than in the FCCU for further data analytics and error handling. The FCCU will only log mission-critical information, which is also handed over to the ground station computer in later stages.
-CAL_BTN: Simply a calibration button for manual calibration if needed.
-There is an extra SPI board-to-wire Molex connector, which also could be used as an extra four GPIO due to the pin multiplexing nature of the ESP32S3.
-GNSS: The NEO-M9N with an extra board-to-wire Molex connector for dedicated setup via external USB to TTL adapter.
-IMUs: BNO055 9DoF as the primary IMU, BMI088 6DoF as the redundant secondary IMU.
-Magnetometer: IIS2MDCTR 3DoF as a complementary device for the BMI088.
-Temperature Sensor: TMP100 to measure board temperature if there are some issues with heat absorption.
-Barometer: DPS310 barometric pressure sensor to gain information about altitude.

## FLIGHT COMPUTER CONTROLLER UNIT

The FCCU (Flight Computer Controller Unit) serves as the central brain, which will have the main algorithms on it.

-There is an SD card serving as a read/write eMMC to do data logging and, if needed, to save calibration data on it to avoid saving stuff in the internal EEPROM of the ESP.
-Current sensor to monitor battery state.
-The CAN bus serves as an extension slot for task-dedicated slave PCBs.
-There is a buzzer for feedback (battery state, specific states, errors, if it lands somewhere unplanned).
-The LEDs are status LEDs (calibration, communications, etc.).
-3x10-pin 2.54mm (0.1") servo header (SIG, 5V, GND) for six servos and four BLDCs with external ESCs capable of 5V @ 9 Amps max.
-There is another board-to-wire I2C Molex connector for anything else I might have forgotten.

## Uncertainties

There are a total of three things about which I have some uncertainties.
1.Schottky Diodes: Have I placed all six Schottky diodes correctly, and are there any missing? I have placed one Schottky diode for each power source: 4x USB, 1x Switching Regulator, and 1x Hot-Start GNSS external battery connector.
2.ESD Protection: I placed 2x ESD diodes on each SD card slot, 4x ESD arrays for each USB slot, 1x ESD diode on the GNSS antenna, and 1x ESD diode on the CANBUS Molex connector. I don't have ESD diodes on the SPI test point headers, the I²C external connector, the external SPI Molex connector, the boot, reset, and calibration buttons, the safeboot header for the GNSS module, the Molex connector of the GNSS module, or the hot-start battery connector for the GNSS module.
3.Test Points: I'm not sure if the test points are sufficient. I feel like they are, but I'm still unsure about it.

deficientInventor

Feza

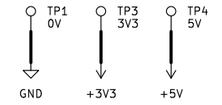
Sheet: /Designnotes/
File: designnotes.kicad_sch

***Title: FEZA FLIGHT COMPUTER***

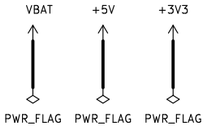
| Size: A3 | Date: 2024-11-18 | **Rev: 1.0** |
|---|---|---|
| KiCad E.D.A. 8.0.7 | | Id: 2/9 |

# POWERMANAGEMENT

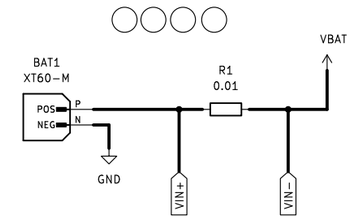
## TESTPOINTS




## POWERFLAGS




## BATTERY INPUT & CURRENT SENSE RESISTOR

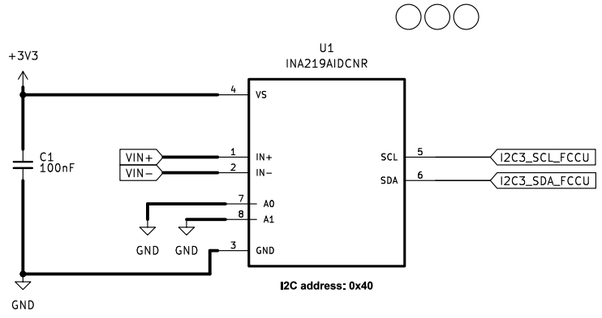



Maximum Input voltage: 26V

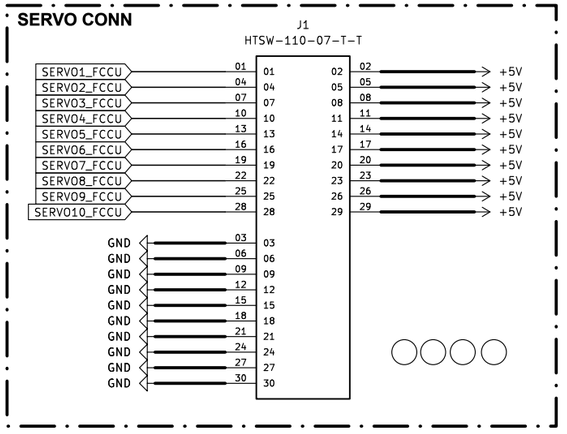
R1 10mOhm 1W 2512

BATTERY PROTECTION??

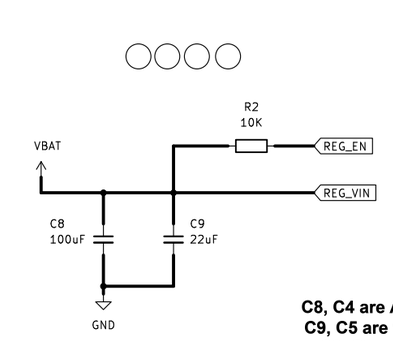
## VOLTAGE/CURRENT/POWERSENSE IC



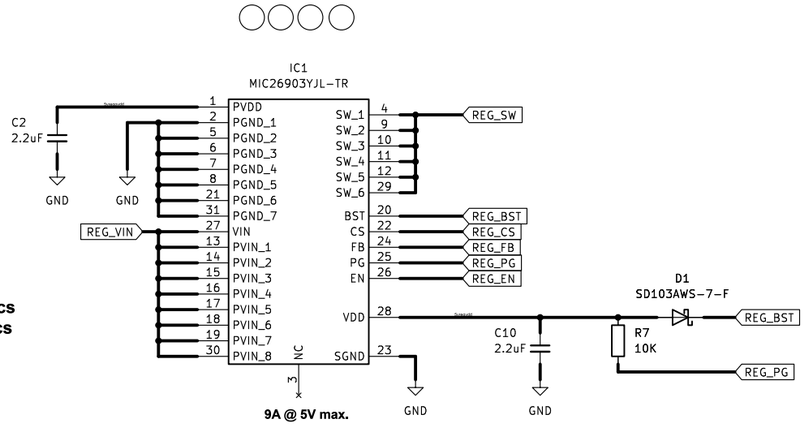

## SERVO CONN



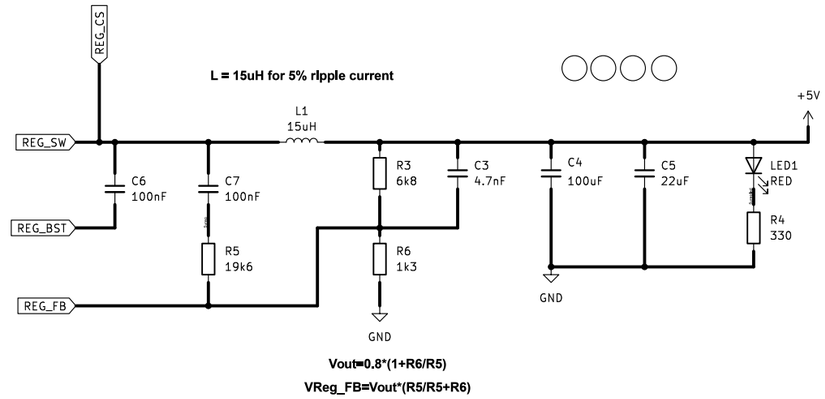

## SWITCHING REGULATOR




C8, C4 are Al Electrolytics
C9, C5 are 1210 ceramics
C8, C9 ratet at >= 35V



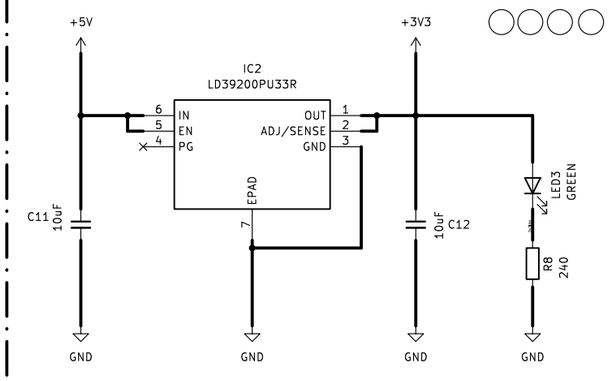

9A @ 5V max.

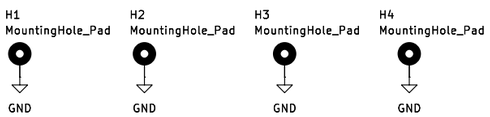



Vout=0.8*(1+R6/R5)

VReg_FB=Vout*(R5/R5+R6)

## LDO/PWRLED3v3 (MCUs and Sensors)




**SANITY-CHECK LEGEND:**

- Associations
- ESD
- Schottky
- Testpoints
- Functional
- Nets

**References:** 1:1 PHILS LAB HADES

**Comment:**

I understand everything, except the switching regulator.
The Datasheet was highly complicated for me to understand
It is saying that i should keep SGND and PGND seperate.
Rick Hartley says, that splitting grounds creates more problems than it solves. The Datasheet says if VIN of the REG is >5.5V PVDD should be connected to VIN, but Phil did connected It to GND.
- Is the Schottky Diode on the REG placed correctly?

deficientInventor

Feza

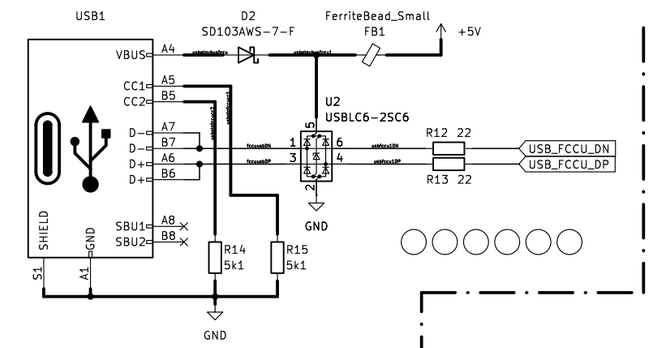
Sheet: /BATT_PWR_MANAGEMENT/
File: BATT_PWR_MANAGEMENT.kicad_sch

***Title: FEZA FLIGHT COMPUTER***

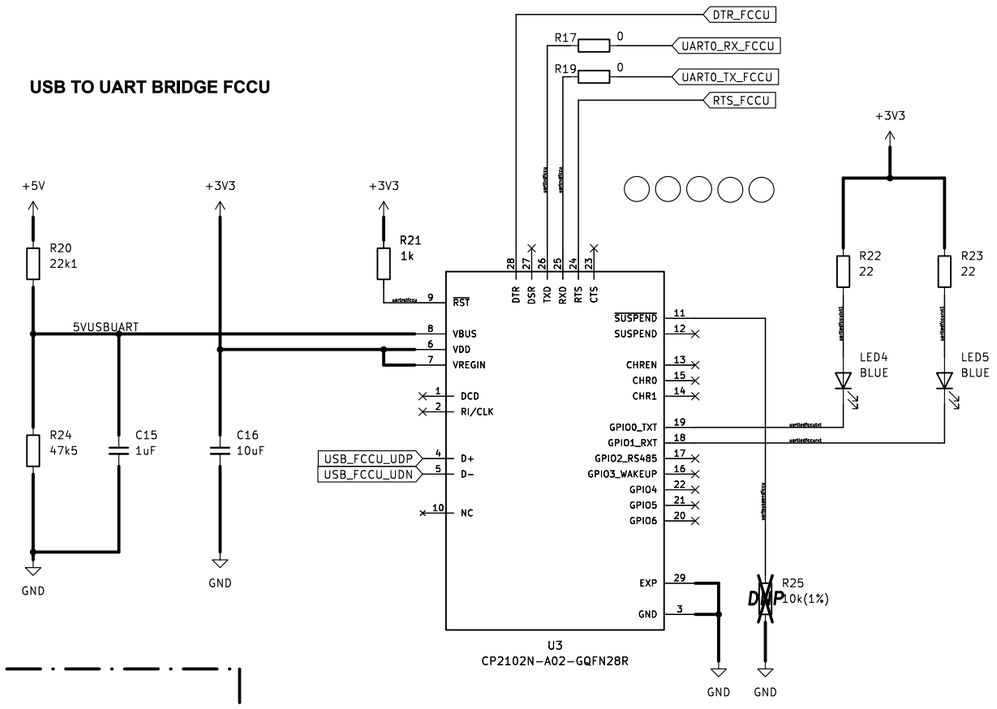
| Size: A3 | Date: 2024-11-18 | **Rev: 1.0** |
|---|---|---|
| KiCad E.D.A. 8.0.7 | | Id: 3/9 |

# FCCU-USB-INTERFACE

## USB-OTG-FCCU




## USB TO UART BRIDGE FCCU



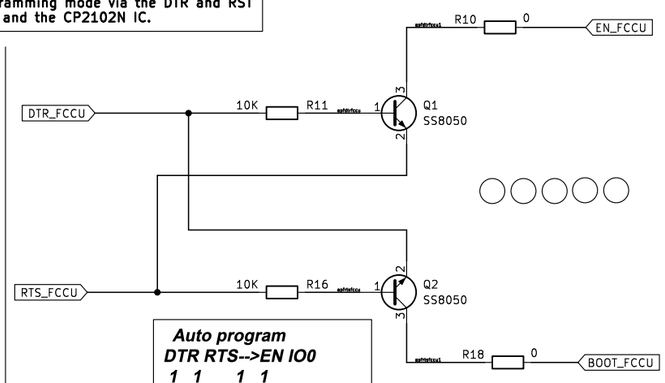

GPIO_TXT and GPIO_RXT are LED status indicators for data transmission. They are connected according to the technical documentation of the CP2102N-A2-xxx28R. See page 23 of the technical documentation.

LED resistors were chosen based on reference values to produce a dimmed light.

Click here to access the documentation.

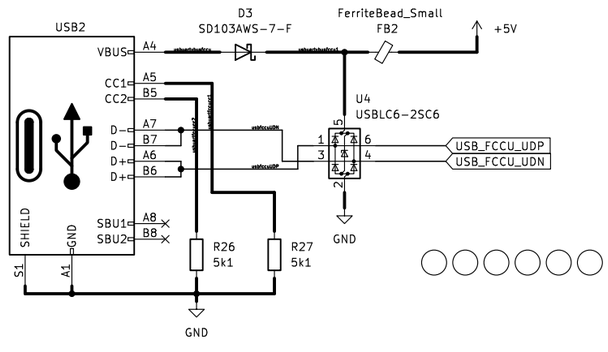
## AUTO PROGRAMMER FCCU

With the logic shown below, the software—IDEs PIO, Arduino, or ESP—IDF can automatically put the board into programming mode via the DTR and RST pins and the CP2102N IC.




| Auto program DTR RTS-->EN IO0 | |
|---|---|
| 1 1 | 1 1 |
| 0 0 | 1 1 |
| 1 0 | 0 1 |
| 0 1 | 1 0 |

## USB-UART-FCCU




**References:**

**https://www.silabs.com/documents/public/data-sheets/cp2102n-datasheet.pdf**
**https://dl.espressif.com/dl/schematics/SCH_ESP32-S3-DevKitC-1_V1.1_20221130.pdf**

***SANITY-CHECK LEGEND:***

- Associations ◯
- ESD ◯
- Schottky ◯
- Testpoints ◯
- Functional ◯
- Nets ◯

**Comments:**
I understand everything on this page,
except the Ferrite Bead values, however i understand why the Ferrite Beads are used. They are just Filtering
the signal reflexion caused by the USB wire lenght
I got told multiple times that I should avoid using Ferrite beads, and I listened to it on the other pages. I kept the FBs here, because since the +5V is going through the LDOs anyway.

This Page is almost a exact copy of the Dev Board except
-the additional LEDS for the TXT and RXT for UART
-USB-C instead of uUSB

deficientInventor

Feza

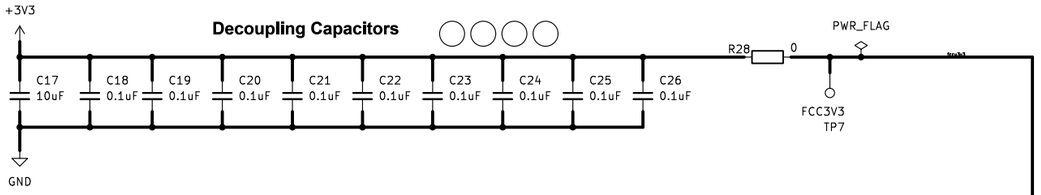
Sheet: /FCCU_USB_INTERFACE/
File: FCCU_USB_INTERFACE.kicad_sch

***Title: FEZA FLIGHT COMPUTER***

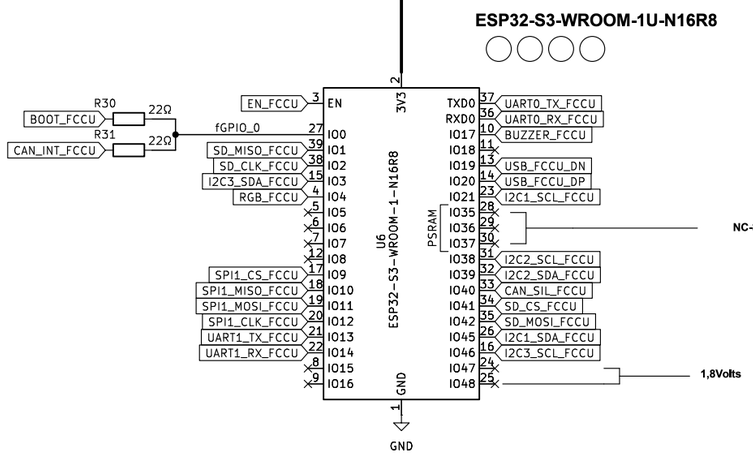
| Size: A3 | Date: 2024-11-18 | **Rev: 1.0** |
|---|---|---|
| KiCad E.D.A. 8.0.7 | | Id: 4/9 |

# FLIGHT CONTROLLER UNIT

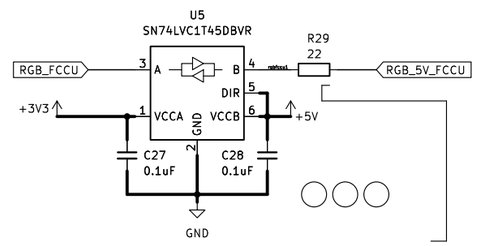
## Decoupling Capacitors




## ESP32-S3-WROOM-1U-N16R8

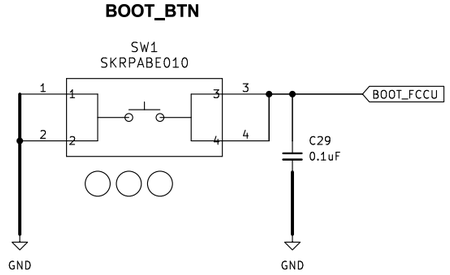



## 3V3-5V LEVEL SHIFTER (LED)

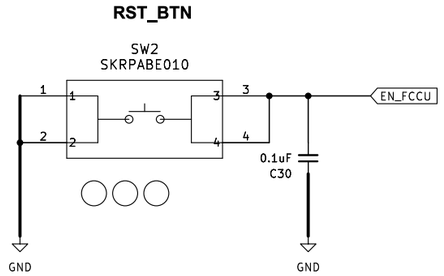



**Series resistor recommendation of neopixels best practices says 300-500ohm series resistor (I guess for longer data lines) since this is a PCB design I decided to use 22ohm series resistor. https://learn.adafruit.com/adafruit-neopixel-uberguide/best-practices**

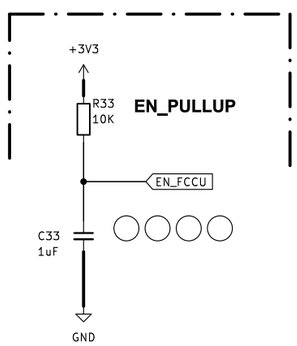
## BOOT_BTN



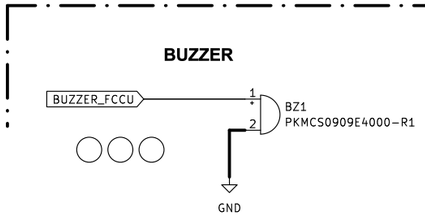

## RST_BTN



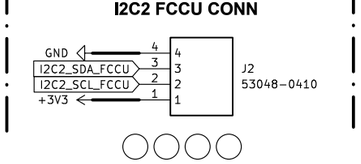

## EN_PULLUP



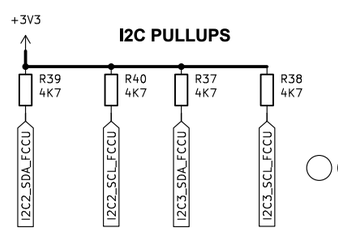

## BUZZER




## I2C2 FCCU CONN

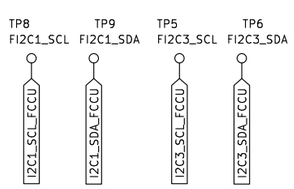



## I2C PULLUPS

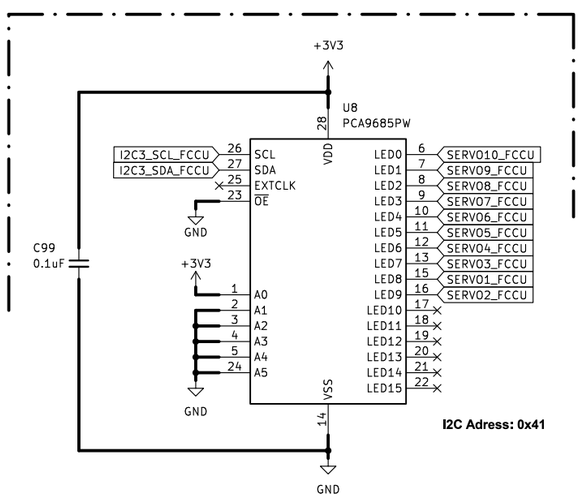



**I2C2 PWRSENSE & EXT-CONN**
**I2C1 FCCU <--> NAVCU ( NAVCU SCHEMATIC )**

## I2C Testpoints

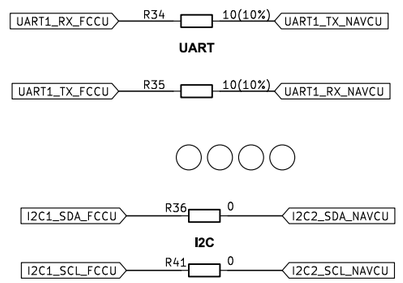



**I2C2 CAN BE TESTED WITH EXT CONN**




**I2C Adress: 0x41**

## FCCU <--> NAVCU




***SANITY-CHECK LEGEND:***

- Associations
- ESD
- Schottky
- Testpoints
- Functional
- Nets

**Comments:**
I understand everything on this page, however,
I have difficulties choosing the right value for series resistors.
I choose the values based on reference
from Phils lab and Robert Feranec.

-The array of decoupling capacitors maybe is overkill,
I choosed it based on Hades from Phils lab.
-I am not sure about the 3x2.54mm Header Testpoint for SDSPI
-CAN can be tested via the external conn.
-I2C2 also can be tested via ext. conn
-UART1 Testpoints? also 2x2.54 Headers?

deficientInventor

Feza

Sheet: /FCCU/
File: FCCU.kicad_sch

***Title: FEZA FLIGHT COMPUTER***

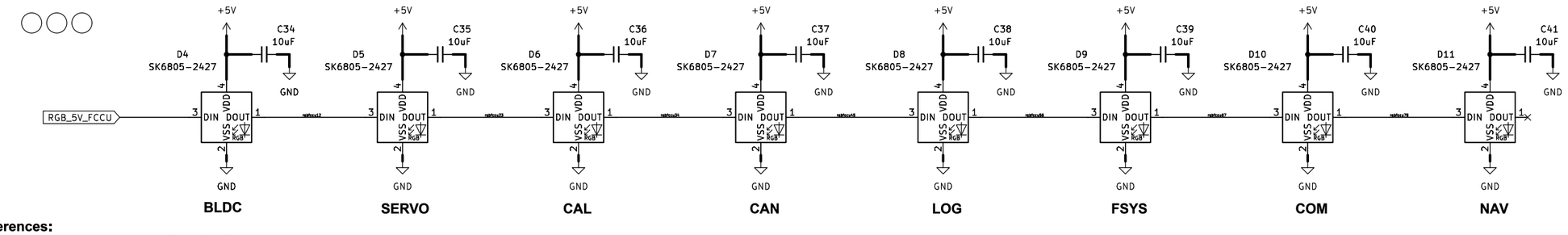
| Size: A3 | Date: 2024-11-18 | **Rev: 1.0** |
|---|---|---|
| KiCad E.D.A. 8.0.7 | | Id: 5/9 |

**References:**


**https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf**
**https://docs.espressif.com/projects/esp-hardware-design-guidelines/en/latest/esp32/esp-hardware-design-guidelines-en-master-esp32.pdf**
**https://dl.espressif.com/dl/schematics/SCH_ESP32-S3-DevKitC-1_V1.1_20221130.pdf**
**https://learn.adafruit.com/16-channel-pwm-servo-driver?view=all**

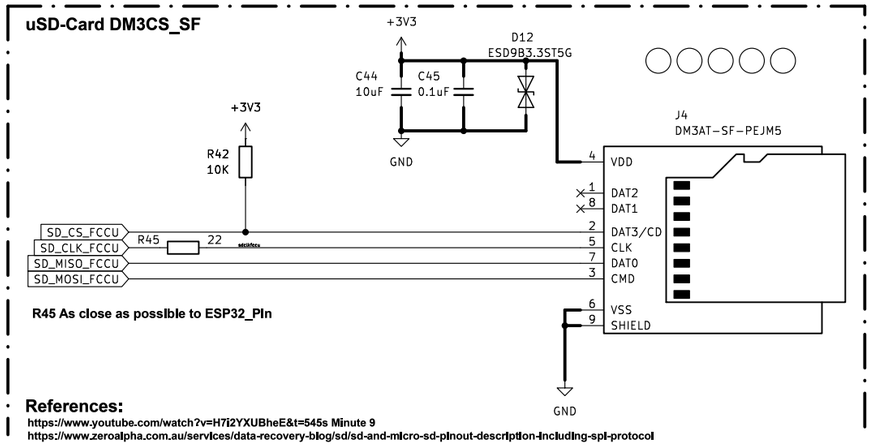
# FLIGHT CONTROLLER UNIT PHERIPHERALS

## STATUS LEDS




**References:**
https://cdn-shop.adafruit.com/product-files/3484/3484_Datasheet.pdf
https://dl.espressif.com/dl/schematics/SCH_ESP32-S3-DevKitC-1_V1.1_20221130.pdf

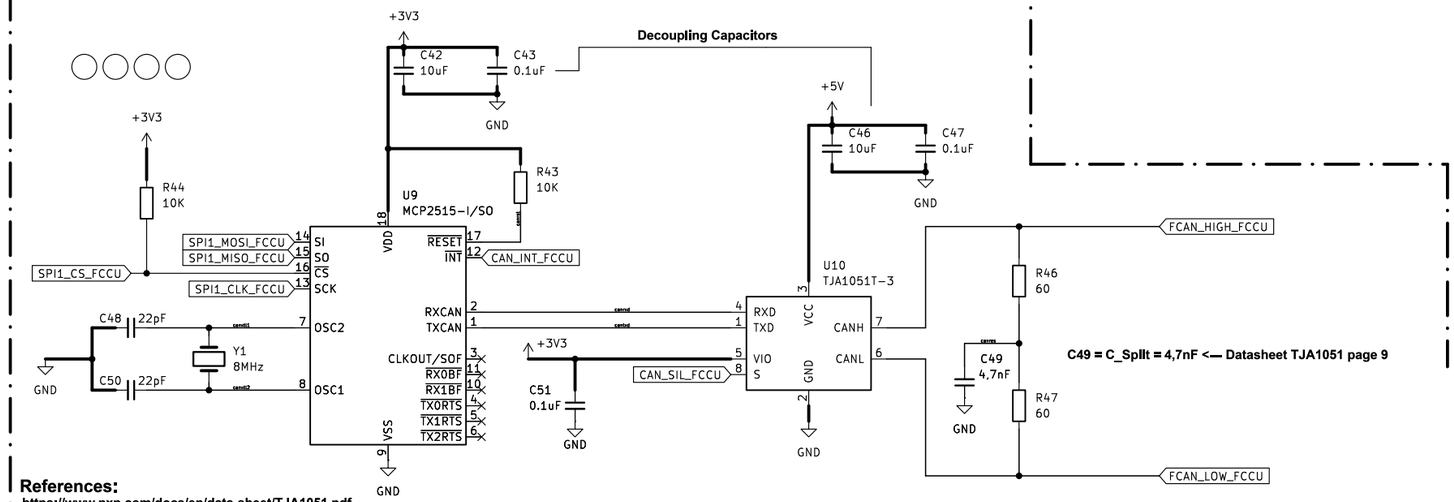
## uSD-Card DM3CS_SF




**R45 As close as possible to ESP32_Pin**

**References:**
https://www.youtube.com/watch?v=H7i2YXUBheE&t=545s Minute 9
https://www.zeroalpha.com.au/services/data-recovery-blog/sd/sd-and-micro-sd-pinout-description-including-spi-protocol

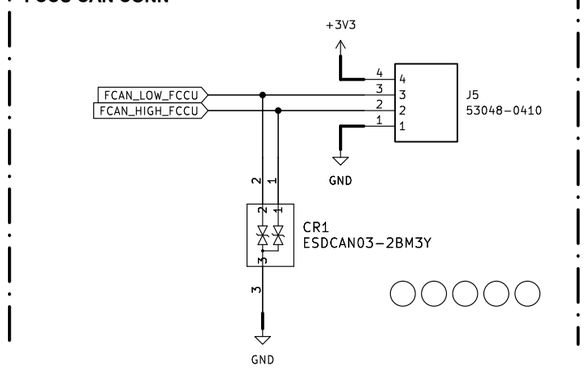
## CAN BUS FCCU




**C49 = C_Split = 4,7nF <— Datasheet TJA1051 page 9**

**References:**
- https://www.nxp.com/docs/en/data-sheet/TJA1051.pdf
- https://e2e.ti.com/support/interface-group/interface/f/interface-forum/290299/iso1050-can-tx-rx-filter
- https://how2electronics.com/interfacing-mcp2515-can-bus-module-with-arduino/

## FCCU CAN CONN




**CANBUS CAN BE TESTED WITH EXT CONN**

***SANITY-CHECK LEGEND:***

- Associations ◯
- ESD ◯
- Schottky ◯
- Testpoints ◯
- Functional ◯
- Nets ◯

**Comments:**
I understand everything on this page,
but im still kinda unsure when it comes to the uSD-Slot

deficientInventor

Feza

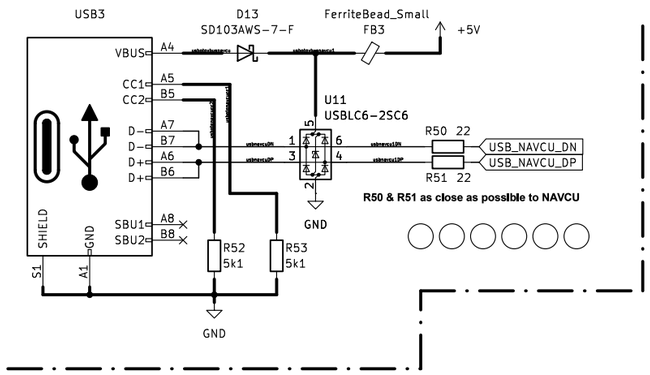
Sheet: /FCCU_PHERIPHERALS/
File: FCCU_PHERIPHERALS.kicad_sch

***Title: FEZA FLIGHT COMPUTER***

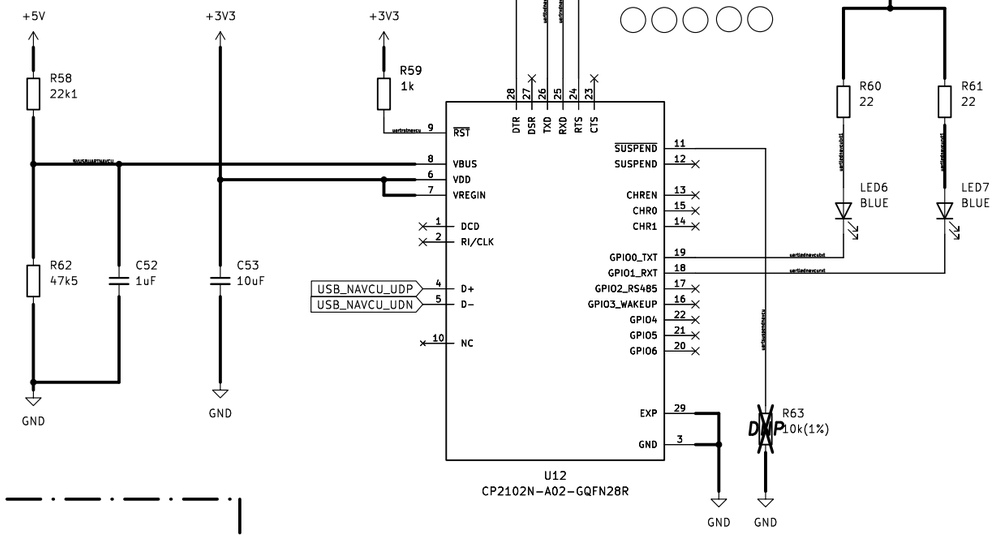
| Size: A3 | Date: 2024-11-18 | **Rev: 1.0** |
|---|---|---|
| KiCad E.D.A. 8.0.7 | | Id: 6/9 |

# NAVCU-USB-INTERFACE

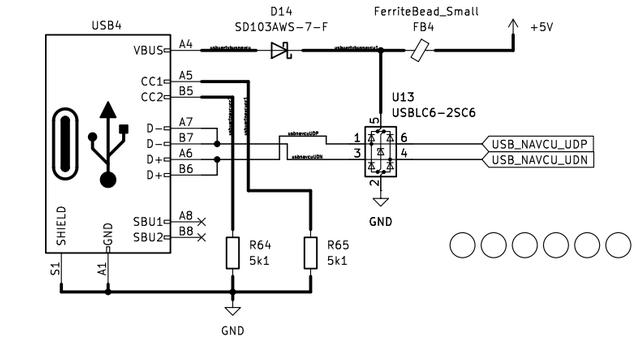
## USB-OTG-NAVCU




## USB TO UART BRIDGE NAVCU

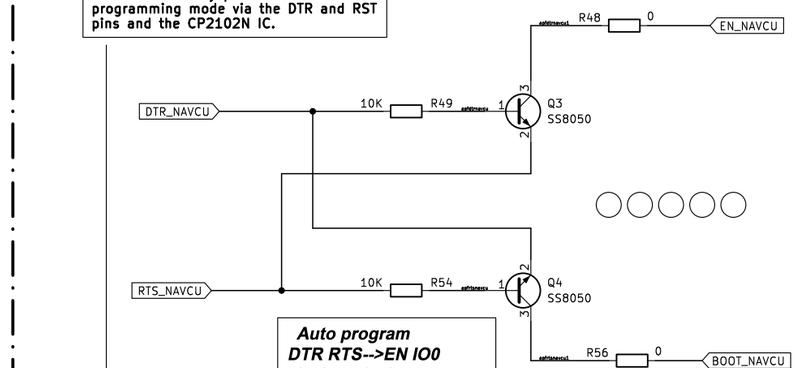



## USB-UART-NAVCU




**With the logic shown below, the software—IDEs PIO, Arduino, or ESP—IDF can automatically put the board into programming mode via the DTR and RST pins and the CP2102N IC.**

## AUTO PROGRAMMER NAVCU




***Auto program***
***DTR RTS-->EN IO0***

| | | | |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |

**GPIO_TXT and GPIO_RXT are LED status indicators for data transmission. They are connected according to the technical documentation of the CP2102N-A2-xxx28R. See page 23 of the technical documentation.**

**LED resistors were chosen based on reference values to produce a dimmed light.**

**Click here to access the documentation.**

***SANITY-CHECK LEGEND:***

- Associations ◯
- ESD ◯
- Schottky ◯
- Testpoints ◯
- Functional ◯
- Nets ◯

**References:**

**https://www.silabs.com/documents/public/data-sheets/cp2102n-datasheet.pdf**
**https://dl.espressif.com/dl/schematics/SCH_ESP32-S3-DevKitC-1_V1.1_20221130.pdf**

**Comments:**
I understand everything on this page,
except the Ferrite Bead values, however i understand why the Ferrite Beads are used. They are just Filtering the signal reflexion caused by the USB wire lenght
I got told multiple times that I should avoid using Ferrite beads, and I listened to it on the other pages. I kept the FBs here, because since the +5V is going through the LDOs anyway.

This Page is almost a exact copy of the Dev Board except
-the additional LEDS for the TXT and RXT for UART
-USB-C instead of uUSB

deficientInventor

Feza

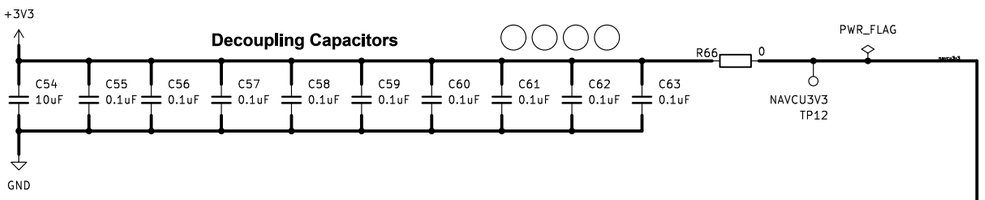
Sheet: /NAVCU_USB_INTERFACE/
File: NAVCU_USB_INTERFACE.kicad_sch

***Title: FEZA FLIGHT COMPUTER***

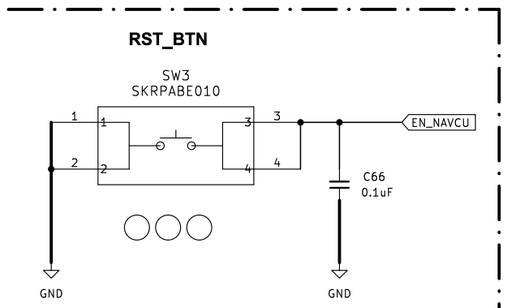
| Size: A3 | Date: 2024-11-18 | **Rev: 1.0** |
|---|---|---|
| KiCad E.D.A. 8.0.7 | | Id: 7/9 |

# NAVIGATION CONTROLLER UNIT

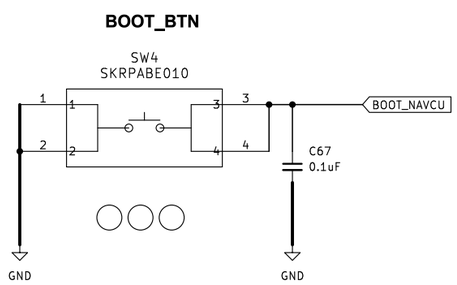
## Decoupling Capacitors



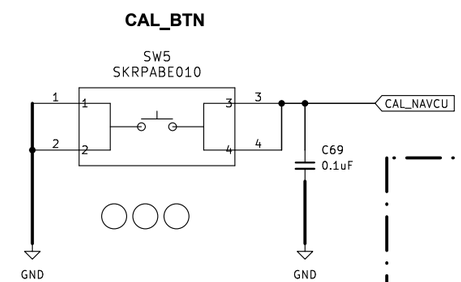

## RST_BTN



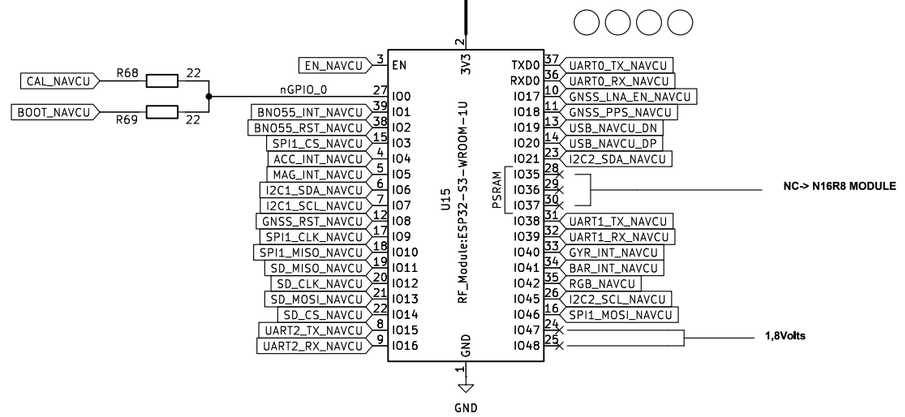

## BOOT_BTN



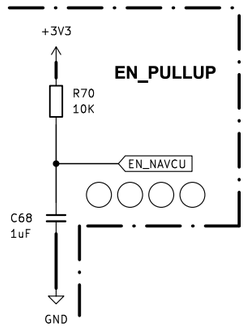

## CAL_BTN



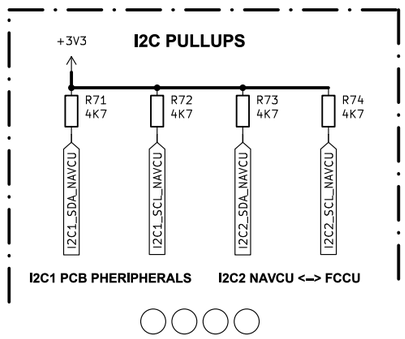

## ESP32-S3-WROOM-1U-N16R8



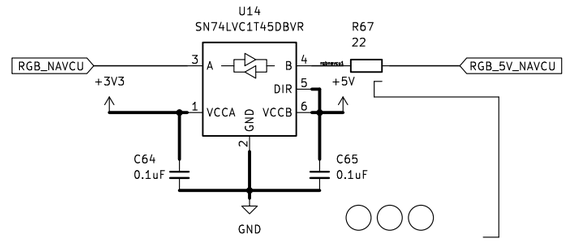

## EN_PULLUP




## I2C PULLUPS



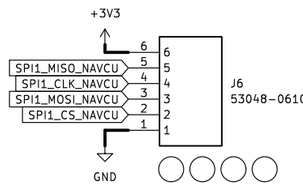

## 3V3-5V LEVEL SHIFTER (LED)



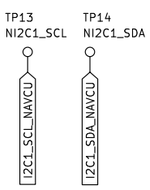

**Series resistor recommendation of neopixels best practices says 300-500ohm series resistor (I guess for longer data lines) since this is a PCB design I decided to use 22ohm series resistor.**

**https://learn.adafruit.com/adafruit-neopixel-uberguide/best-practices**

## SPI/GPIO CONN




## I2C Testpoints




**I2C2 ALREADY ON FCCU SCHEMATIC**

## SANITY-CHECK LEGEND:

- Associations
- ESD
- Schottky
- Testpoints
- Functional
- Nets

**Comments:**

I understand everything on this page, however,
I have difficulties choosing the right value for series resistors.
I choose the values based on reference
from Phlslab and Robert Feranec.

-The array of decoupling capacitors maybe is overkill,
i choosed it based on Hades from Phlslab.
-I am not sure about the 3x2,54mm Header Testpoint for SDSPI
-I2C2 testpoints on FCCU Page
-UART1 Testpoints? also 2x2.54 Headers?

deficientInventor

Feza

**References:**

**https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf**
**https://docs.espressif.com/projects/esp-hardware-design-guidelines/en/latest/esp32/esp-hardware-design-guidelines-en-master-esp32.pdf**
**https://dLespressif.com/dl/schematics/SCH_ESP32-S3-DevKitC-1_V1.1_20221130.pdf**

Sheet: /NAVCU/
File: NAVCU.kicad_sch

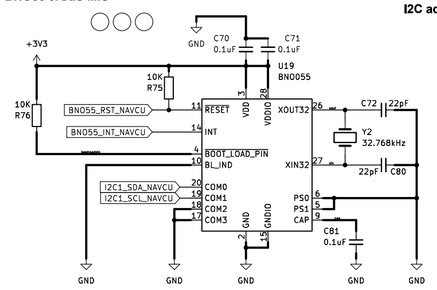
***Title: FEZA FLIGHT COMPUTER***

| Size: A3 | Date: 2024-11-18 | **Rev: 1.0** |
|---|---|---|
| KiCad E.D.A. 8.0.7 | | Id: 8/9 |

# NAVCU PHERIPHERALS

## BNO55 9AXIS IMU

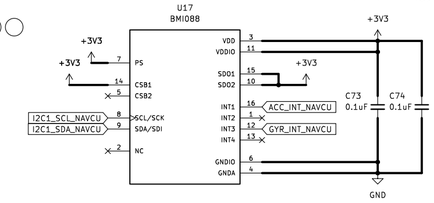
PS0 & PS1 GND = I2C
I2C address: 0x28




**References:**
https://www.bosch-sensortec.com/media/boschsensortec/downloads/application_notes_1/bst-bno055-an007.pdf
https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bno055-ds000.pdf

## IMU BMI088 GYRO & ACC

I2C address: 0x19 (acc) 0x69 (gyr)

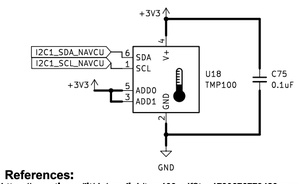



SDO1 & SDO2 HIGH FOR ADDR: 0x19 & 0x69
CSB 1 HIgh(acc) & PS HIgh(gyr) = I2C MODE

**References:**
https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmi088-ds001.pdf PAGE48 & 13

## PCB TEMP SENSE TMP100

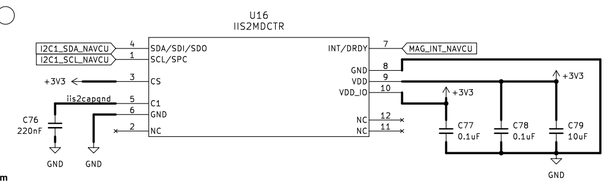
I2C address: 0x4E




**References:**
https://www.ti.com/lit/ds/symlink/tmp100.pdf?ts=1730676779420 page 10 & 21

## MAGN IIS2MDCTR

I2C address: 0011110b = 0x1E



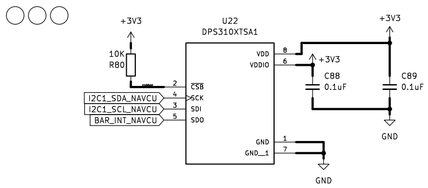

C76 ESR < 200mOhm

It must be connected as close as possible to pins 5 and 6 since very high current pulses flow from C1 to pin 5 and 6. This avoids problems caused by inductive effects due to the length of the copper strips.
-It is highly recommended to use low ESR (max 200 mOhm)

**References:**
https://www.st.com/resource/en/datasheet/iis2mdc.pdf
page 19 & 22 & 24

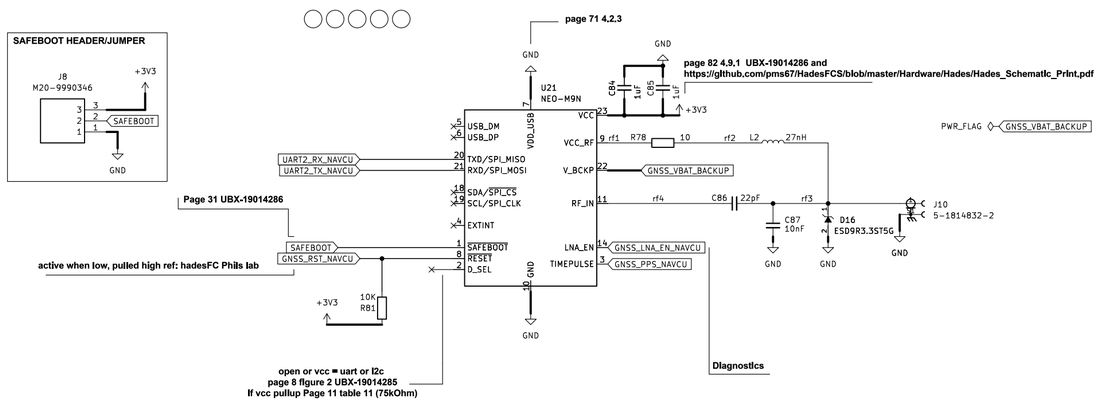
## Pressure Sensor DSP310

I2C address: 0x76




**References:**
https://www.infineon.com/dgdl/Infineon-DPS310-DS-v01_00-EN.pdf?fileId=5546d462576f3475015775082 6c42242
page 18-19
https://learn.adafruit.com/assets/87733

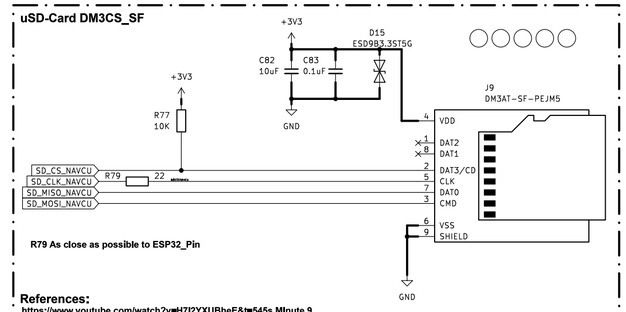
## GNSS u Blox




**References:**
https://content.u-blox.com/sites/default/files/NEO-M9N_IntegrationManual_UBX-19014286.pdf
page 73 design, page 12 4.2.1

**References:**
https://content.u-blox.com/sites/default/files/NEO-M9N-00B_DataSheet_UBX-19014285.pdf
Page 8 FIgure 2

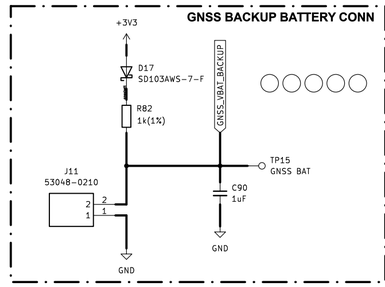
## uSD-Card DM3CS_SF



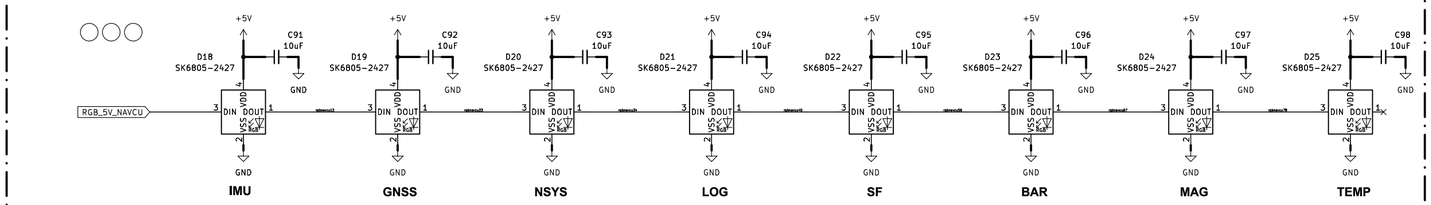

R79 As close as possible to ESP32_Pin

**References:**
https://www.youtube.com/watch?v=H7I2YXUBheE&t=545s Minute 9
https://www.zeroalpha.com.au/services/data-recovery-blog/sd/sd-and-micro-sd-pinout-description-including-spi-protocol

## GNSS BACKUP BATTERY CONN

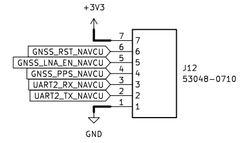



## STATUS LEDS




**References:**
https://cdn-shop.adafruit.com/product-files/3484/3484_Datasheet.pdf
https://dl.espressif.com/dl/schematics/SCH_ESP32-S3-DevKitC-1_V1.1_20221130.pdf

## uBLOX GNSS CONN




**SANITY-CHECK LEGEND:**

- Associations
- ESD
- Schottky
- Testpoints
- Functional
- Nets

**Comments:**
I understand everything on this page.

I choose 2 IMUs:
-BNO055 Main IMU 9 DoF.
-BMI088 Redundant IMU 6 DoF with IIS2MDCTR Magn

second SD-Card-Slot for Dataheavy Navigation Logging

I feel kinda unsure about the ESD on the external SMA-Antenna

deficientInventor

Feza

Sheet: /NAVIGATION_PHERIPHERALS/
File: NAVIGATION_PHERIPHERALS.kicad_sch

***Title: FEZA FLIGHT COMPUTER***

| Size: A2 | Date: 2024-11-18 | **Rev: 1.0** |
|---|---|---|
| KiCad E.D.A. 8.0.7 | | Id: 9/9 |