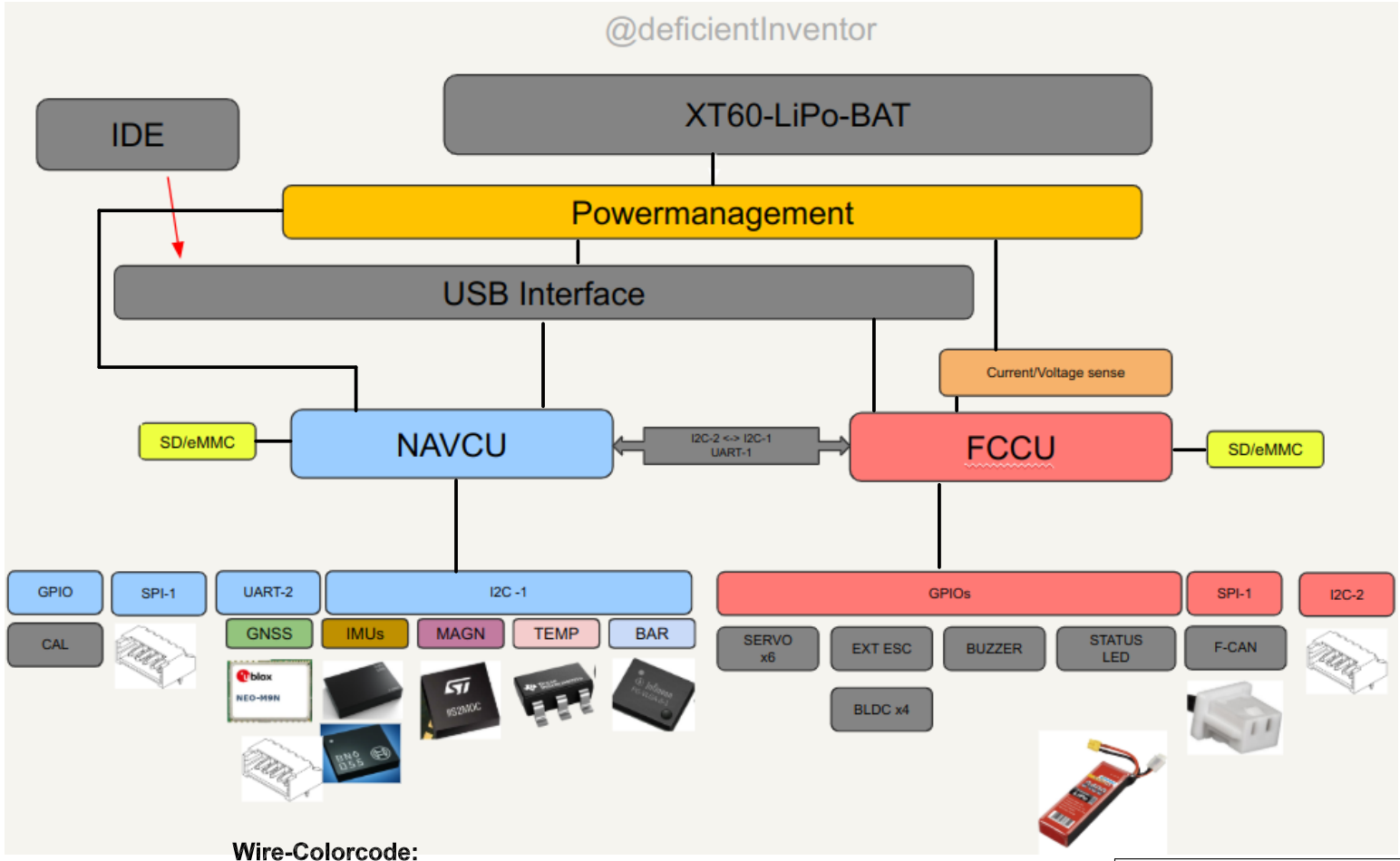


Designnotes

BATT_PWR_MANAGEMENT

FCCU_USB_INTERFACE

NAVCU_USB_INTERFACE

FCCU

NAVCU

FCCU_PHERIPHERALS

NAVIGATION_PHERIPHERALS

Wire-Colorcode:

POWER

Type	Color	Width
VBAT		?
+5V		?
+3V3		?
+1V8		?
REG_FB		?
GND		?

SIGNAL

Type	Color	Impedance	Width
UART		50Ω	?
I2C		50Ω	?
Signal		50Ω	?
USB		90Ω	?
RF/SPI		50Ω	?

TODOS:

-Schematic Review

Comments:

Trace width calculations will be performed once I have approval of the schematics and can proceed with the layout.

Text-based project definition in design notes

Sheet: /
File: feza_fcu_pcb.kicad_sch

Title: FEZA FLIGHT COMPUTER

Size: A4	Date: 2024-11-18	**Rev: 1.0**
KiCad E.D.A. 8.0.6		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. 8.0.6		Id: 2/9

POWERMANAGEMENT

TESTPOINTS

TP1 0V — GND
TP2 1V8 — +1V8
TP3 3V3 — +3V3
TP4 5V — +5V

POWERFLAGS

VBAT, +5V, +3V3, +1V8 — PWR_FLAG

BATTERY INPUT & CURRENT SENSE RESISTOR

BAT1 XT60-M (POS, NEG) — GND
R1 0.01 — VBAT, VIN+, VIN-

Maximum input voltage: 26V

R1 10mOhm 1W 2512

BATTERY PROTECTION??

VOLTAGE/CURRENT/POWERSENSE IC

U1 INA219AIDCNR

+3V3, C1 100nF, GND
VIN+, VIN-
VS, IN+, IN-, A0, A1, GND
SCL — I2C2_SCL_FCCU
SDA — I2C2_SDA_FCCU

I2C address: 0x40

SERVO CONN

J1 HTSW-110-07-T-T

SERVO1_FCCU
SERVO2_FCCU
SERVO3_FCCU
SERVO4_FCCU
SERVO5_FCCU
SERVO6_FCCU
SERVO7_FCCU
SERVO8_FCCU
SERVO9_FCCU
SERVO10_FCCU

+5V, GND

SWITCHING REGULATOR

R2 10K — VBAT, REG_EN, REG_VIN
C8 100uF, C9 22uF — GND

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

IC1 MIC26903YJL-TR

C2 2.2uF — GND
REG_VIN
PVDD, PGND_1..7, VIN, PVIN_1..8, NC
SW_1..6 — REG_SW
BST — REG_BST
CS — REG_CS
FB — REG_FB
PG — REG_PG
EN — REG_EN
VDD — C10 2.2uF — GND
SGND — GND

D1 SD103AWS-7-F — REG_BST
R7 10K — REG_PG

9A @ 5V max.

L = 15uH for 5% ripple current

REG_CS, REG_SW, REG_BST, REG_FB
L1 15uH — +5V
C6 100nF, C7 100nF
R5 19k6, R6 1k3 — GND
R3 6k8, C3 4,7nF
C4 100uF, C5 22uF — GND
LED1 RED, R4 330

Vout=0.8*(1+R6/R5)
VReg_FB=Vout*(R5/R5+R6)

LDO/PWRLED3v3 (MCUs and Sensors)

IC2 LD39200PU33R

+5V, C11 10uF — GND
IN, EN, PG, EPAD, OUT, ADJ/SENSE, GND
+3V3, C12 10uF — GND
LED3 GREEN, R8 240 — GND

LDO/PWRLED1,8v (CANSilent Levelshifter)

IC3 LD39100PU18RY

+3V3, C13 10uF — GND
VIN, EN, PG, EPAD, VOUT, NC, GND
+1V8, C14 10uF — GND
LED2 YELLOW-GREEN, R9 10 — GND

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

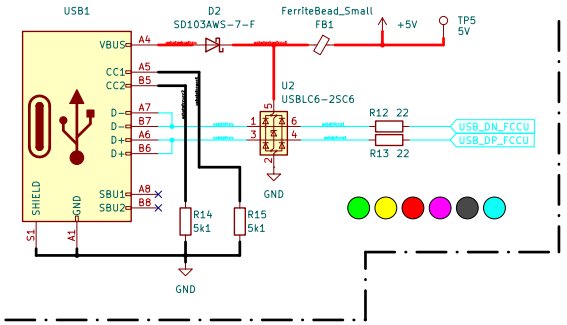
Sheet: /BATT_PWR_MANAGEMENT/
File: BATT_PWR_MANAGEMENT.kicad_sch

Title: FEZA FLIGHT COMPUTER

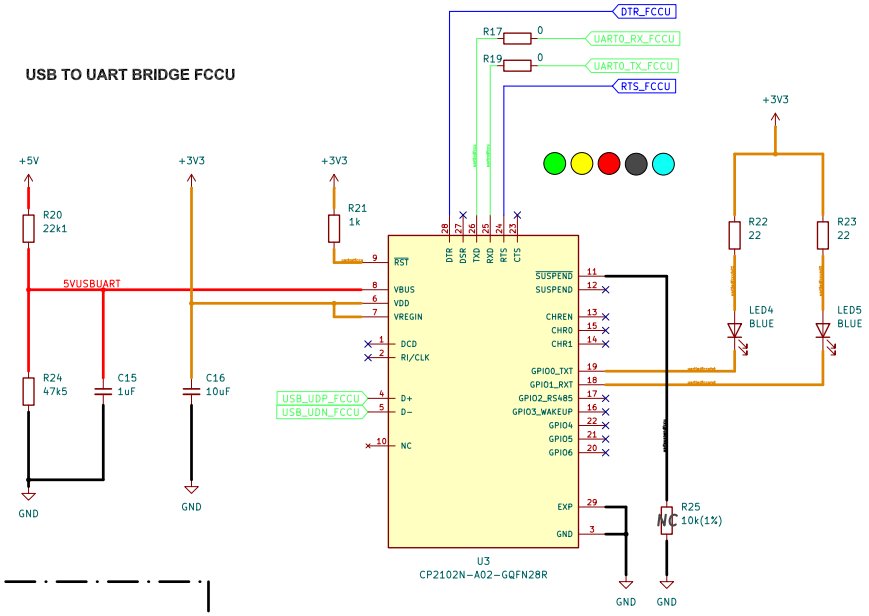
Size: A3	Date: 2024-11-18	**Rev: 1.0**
KiCad E.D.A. 8.0.6		Id: 3/9

FCCU-USB-INTERFACE

USB-OTG-FCCU



USB TO UART BRIDGE FCCU



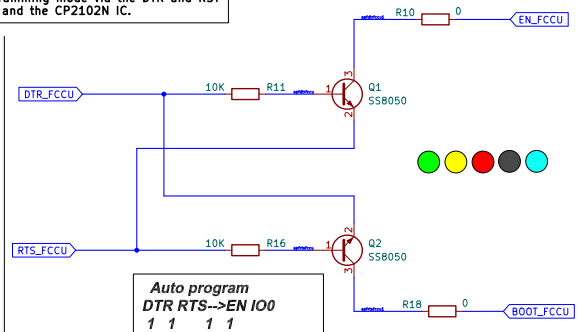
GPIO_TXT and GPIO_RXT are LED status indicators for data transmission. They are connected according to the technical documentation of the CP2102N-A2-xxx28R. See page 23 of the technical documentation.

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

Click here to access the documentation.

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.

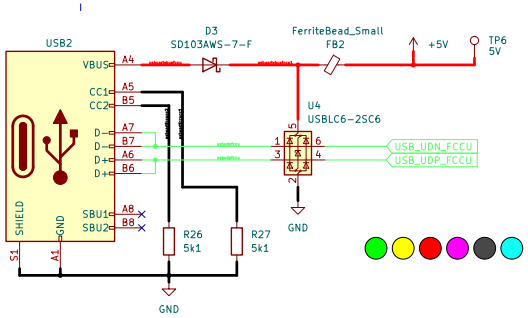
AUTO PROGRAMMER FCCU



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



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

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

deficientInventor

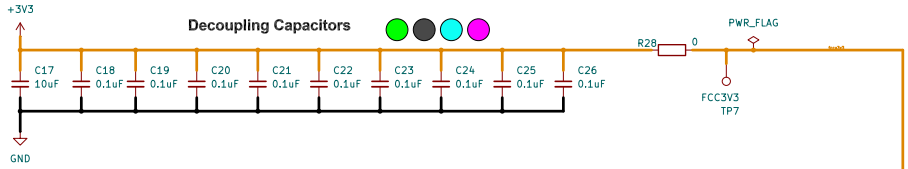
Feza

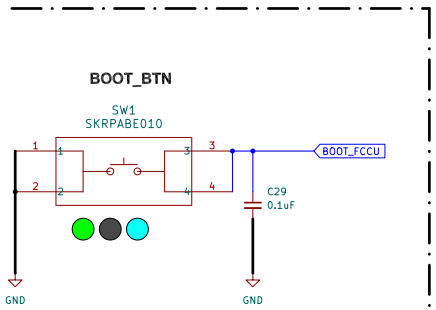
Sheet: /FCCU_USB_INTERFACE/
File: FCCU_USB_INTERFACE.kicad_sch

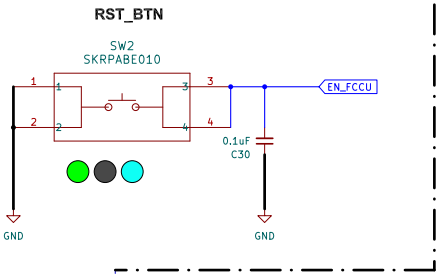
Title: FEZA FLIGHT COMPUTER

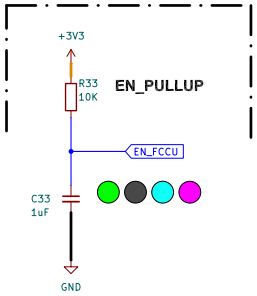
Size: A3	Date: 2024-11-18	Rev: 1.0
KiCad E.D.A. 8.0.6		Id: 4/9

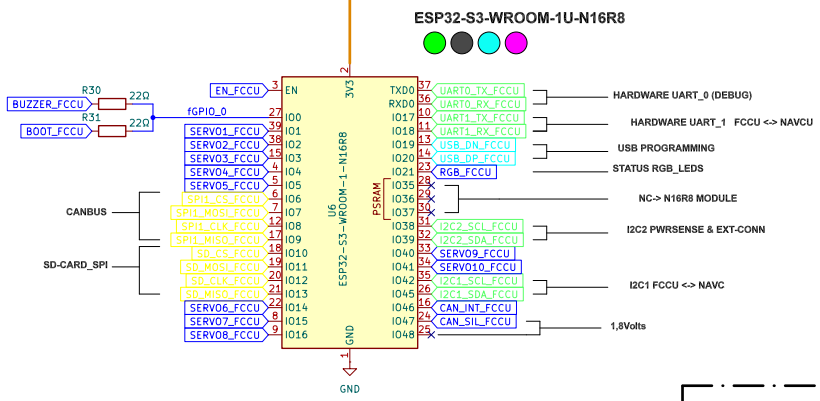
FLIGHT CONTROLLER UNIT

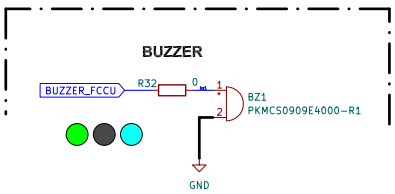


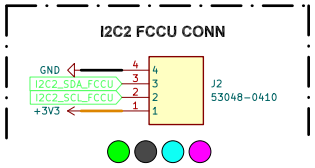


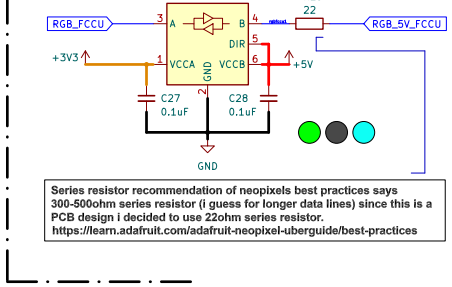


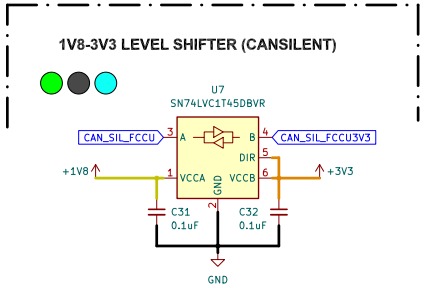


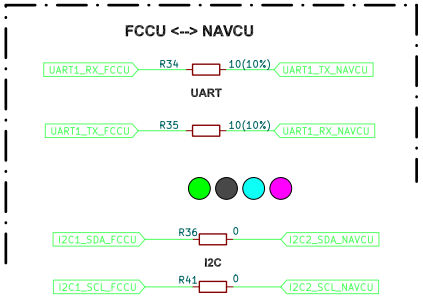


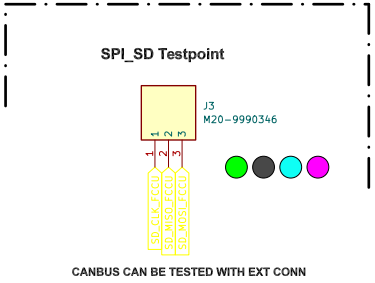


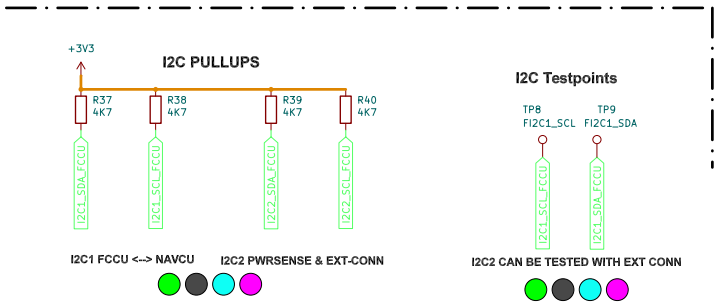


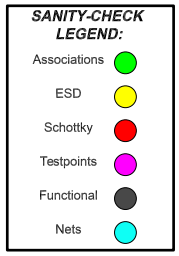












References:

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

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 Philslab and Robert Feranec.

-The array of decoupling capacitors maybe is overkill,
i choosed it based on Hades from Philslab.
-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?

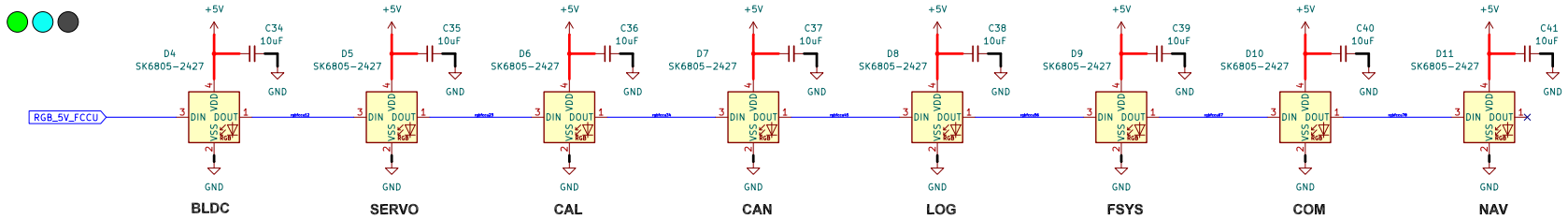
Sheet: /FCCU/
File: FCCU.kicad_sch

Title: FEZA FLIGHT COMPUTER

Size: A3	Date: 2024-11-18	**Rev: 1.0**
KiCad E.D.A. 8.0.6		Id: 5/9

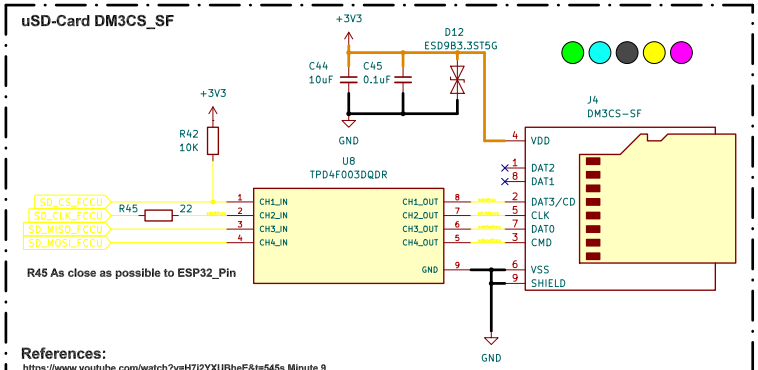
FLIGHT CONTROLLER UNIT PHERIPHERALS

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

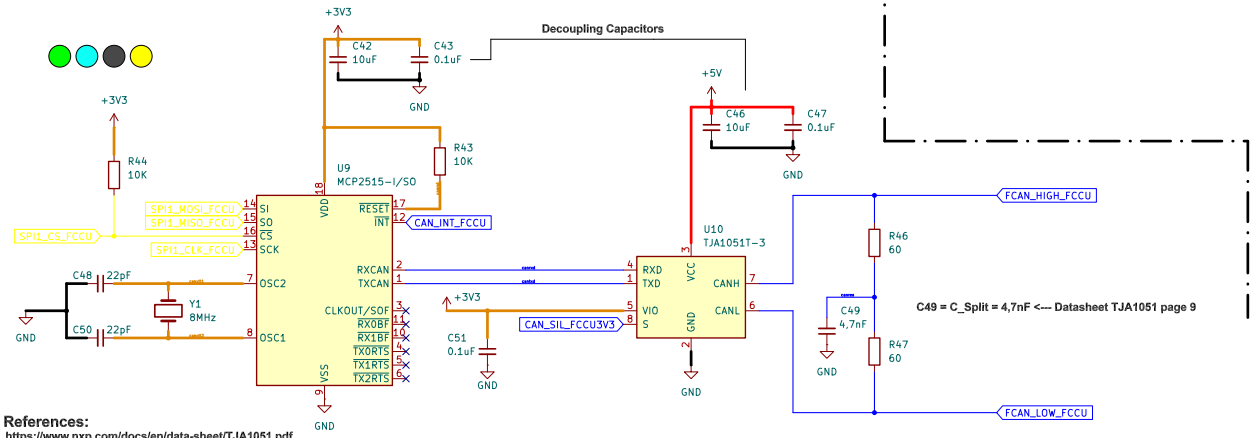
uSD-Card DM3CS_SF



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

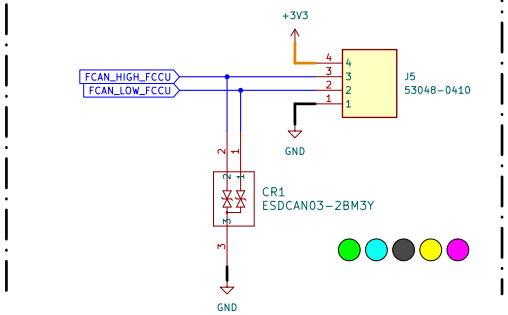
CAN BUS FCCU



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



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

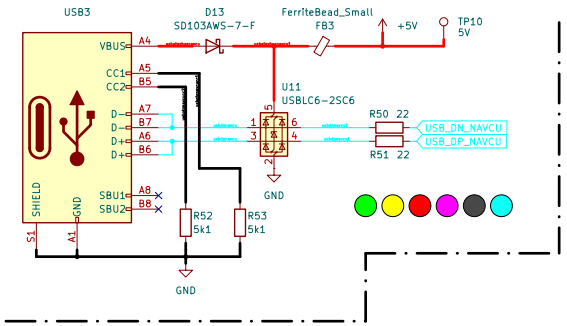
Sheet: /FCCU_PHERIPHERALS/
File: FCCU_PHERIPHERALS.kicad_sch

Title: FEZA FLIGHT COMPUTER

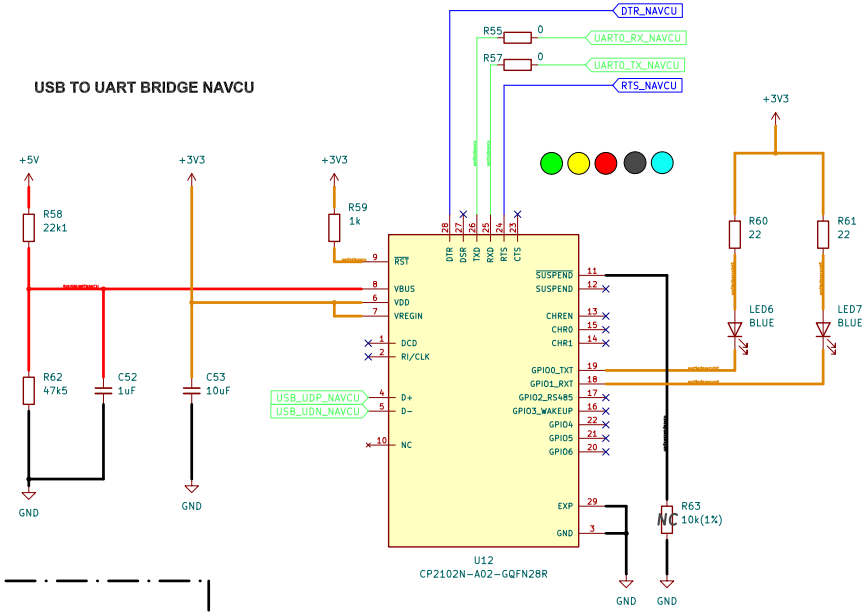
Size: A3	Date: 2024-11-18	**Rev: 1.0**
KiCad E.D.A. 8.0.6		Id: 6/9

NAVCU-USB-INTERFACE

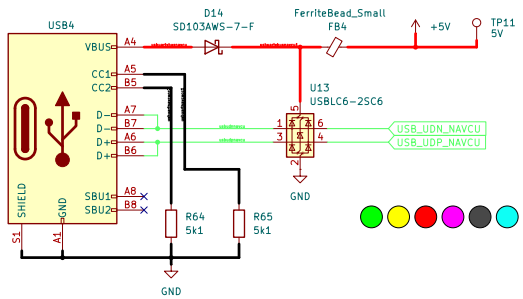
USB-OTG-NAVCU



USB TO UART BRIDGE NAVCU

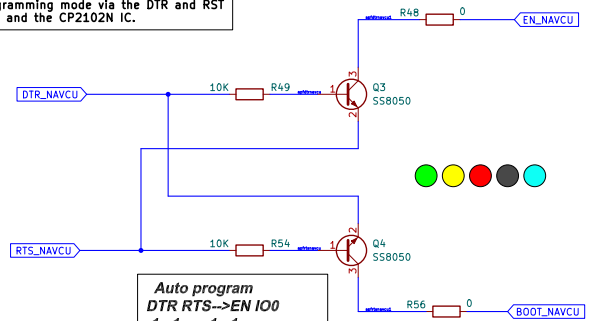


USB-UART-NAVCU



AUTO PROGRAMMER 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 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

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

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

deficientInventor

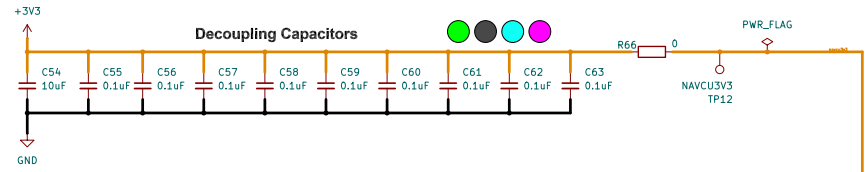
Feza

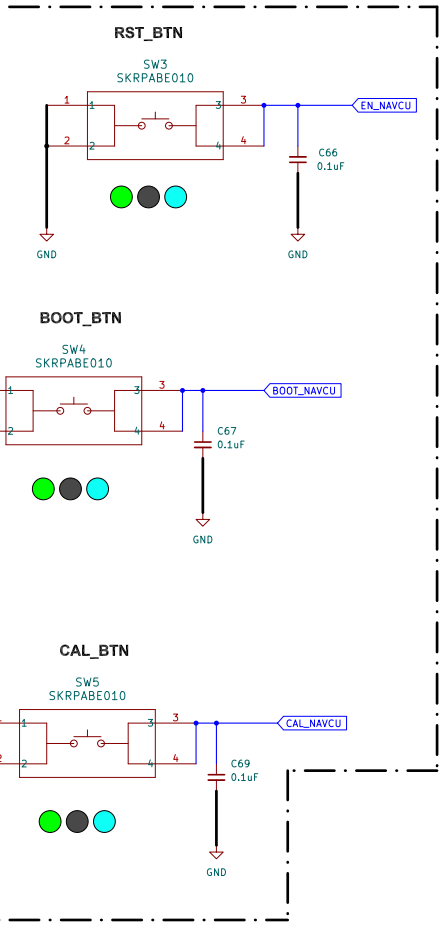
Sheet: /NAVCU_USB_INTERFACE/
File: NAVCU_USB_INTERFACE.kicad_sch

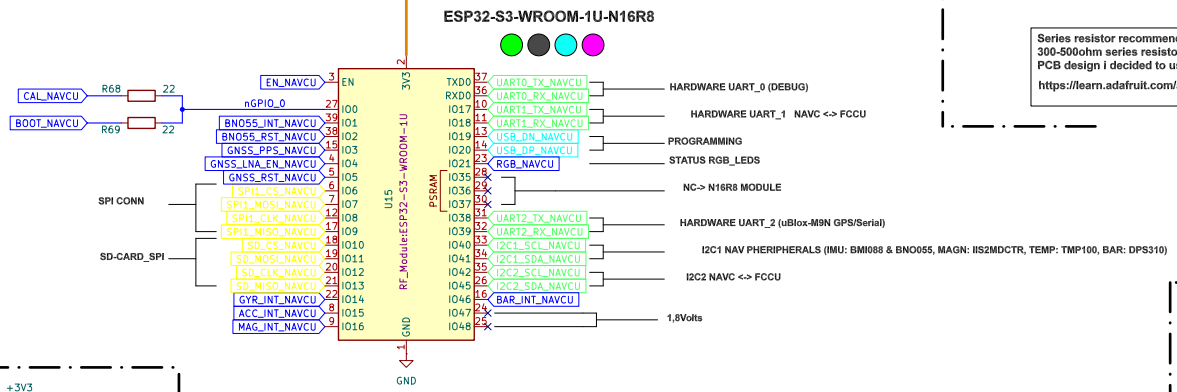
Title: FEZA FLIGHT COMPUTER

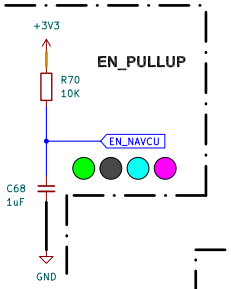
Size: A3	Date: 2024-11-18	Rev: 1.0
KiCad E.D.A. 8.0.6		Id: 7/9

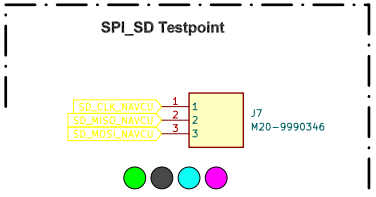
NAVIGATION CONTROLLER UNIT

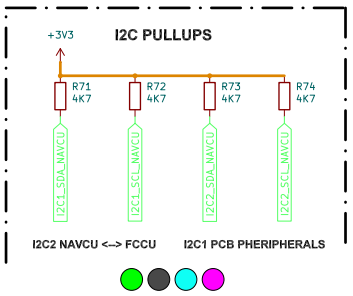


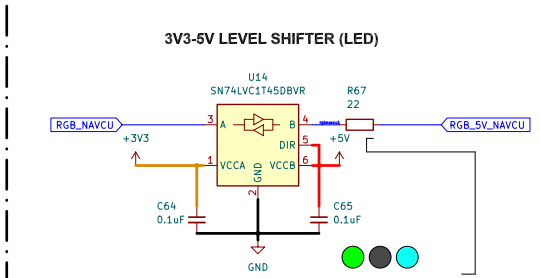


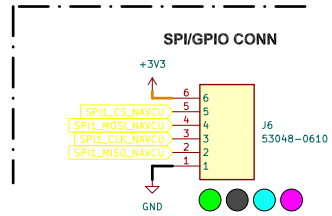


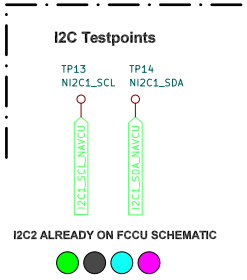


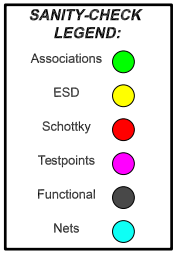












References:

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

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 Philslab and Robert Feranec.

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

deficientInventor

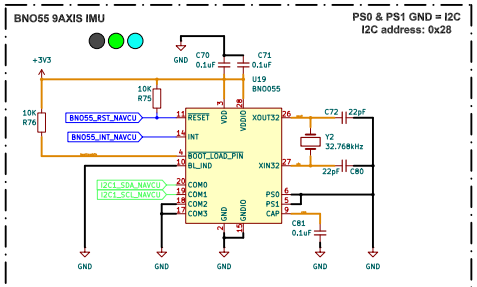
Feza

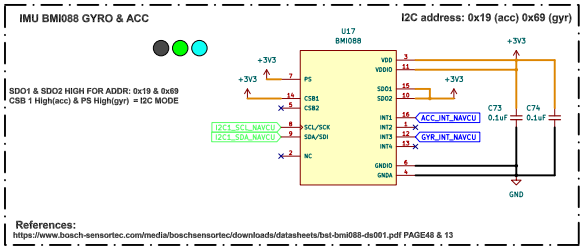
Sheet: /NAVCU/
File: NAVCU.kicad_sch

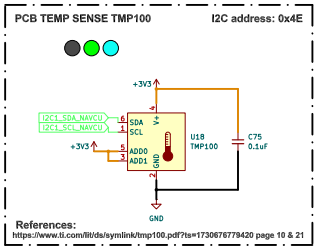
Title: FEZA FLIGHT COMPUTER

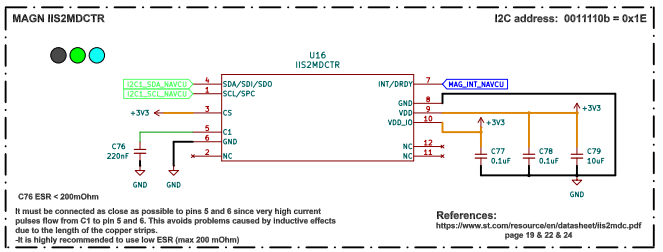
Size: A3	Date: 2024-11-18	**Rev: 1.0**
KiCad E.D.A. 8.0.6		Id: 8/9

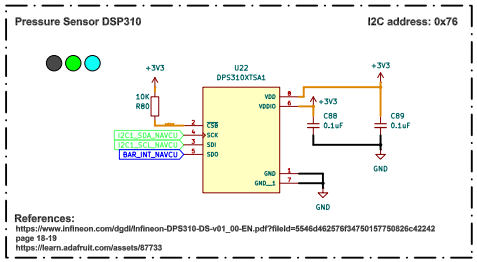
NAVCU PHERIPHERALS

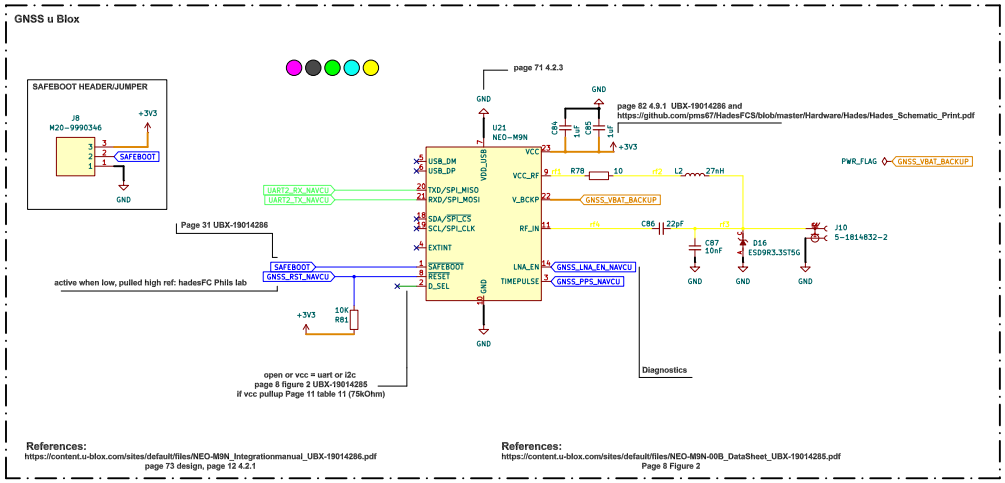


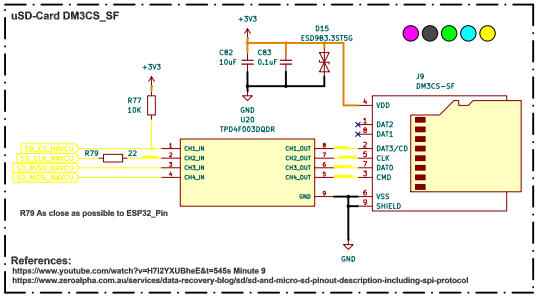


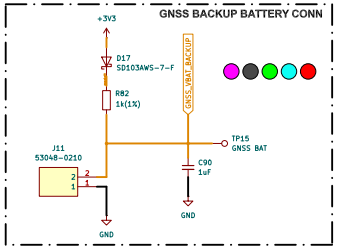


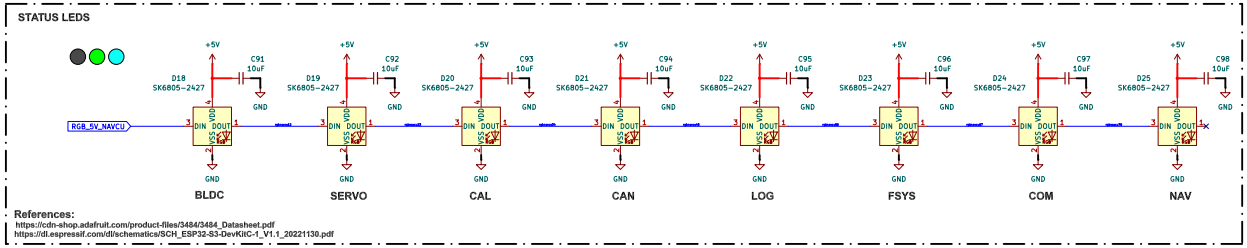


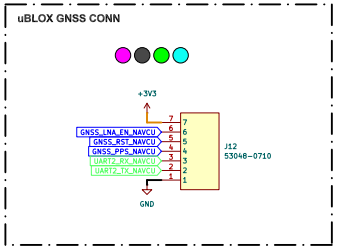












SANITY-CHECK LEGEND:

- 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. 8.0.8		Id: 9/9