SENSE AND CONTROL

DELFT HYPERLOOP VIII

Preliminary Design Document

16 November 2023

DELFT
HYPERLOOP

Contents

1 Introduction

The Sense & Control (SC) subsystem is an integral part of the pod. The system collects data about the internal and external conditions the pod is operating in, and makes sure all data is communicated throughout the pod. Additionally, the SC subsystem enables wireless control and monitoring of the pod. The most important components of the subsystem are **sensors, control** and **communication**.

1.1 Sensors

There are many important variables that need to be taken into account when controlling the pod. These include the vertical and lateral offset, the longitudinal location as well as the internal and external temperature and pressure.

1.2 Control

The subsystem focuses on the top-level control of the pod, leaving low-level control to the other subsystems for enhanced modularity. The main controller PCB executes is in charge of this top-level control, it makes use of all sensory information to determine the most logical next state the pod should transition to. An example of this is when the pod should stop accelerating.

1.3 Communication

A CAN-network between the different sensor or control systems around the pod enables communication of important information. Additionally, the ground station can, true WiFi, wirelessly connect with the main PCB to control and monitor the pod's status.

2 Requirements and Risks

ID	General Requirement
SC.1	The system shall communicate the status of the different subsystems at all moments after booting with the ground station.
SC.2	In case of emergency, the emergency braking system should be deployed within 100ms
SC.3	The system should be able to reliably transmit information from the main PCB to the ground station
SC.4	The system must be able to have remote firmware updates
SC.5	All components shall operate at a pressure 10 mbar.
SC.6	The system should be able to communicate real-time information to the ground station in less than 100ms from the moment it was captured by the sensor
SC.7	All critical data should be stored on the main controller as well as in the ground station
SC.8	The system shall provide the levitating system with the vertical and horizontal offset location with a minimum accuracy of 0.5% of the air gap
SC.9	The system shall weigh less than TBD kg.
SC.10	For any case when a data value goes out of its required safety range, a safety procedure should be available and described in the FDD.
SC.11	The system shall not cost more than TBD.
SC.12	All cables in the vacuum box should have slack.
ID	**Requirement S&C for Lane-Switch**
SC.LS.1	The subsystems should be able to detect an upcoming lane-switch
SC.LS.2	The subsystems should be able to recognize the pod going off-path and deploy brakes without having any electronic control involved
SC.LS.3	The subsystems should be able to decide on which lane the pod enters before the pod starts accelerating
ID	**Requirement S&C for Localization**
SC.LOC.1	Real-time pod location shall be measured with a minimal precision of 4mm all time
SC.LOC.2	A lane-switch should be announced using the localization system.
SC.LOC.3	The system should have a working range at least as wide as the offset range of the levitation system
SC.LOC.4	The system should be self-calibrating, meaning with different offsets, the signal can produce meaningful information
SC.LOC.5	When the pod loses power mid-run, the system should be able to recalibrate the pods position
SC.LOC.6	The marker's and track's thermal expansion coefficients should be roughly the same since changes in temperature then would not mean misalignment.
SC.LOC.7	The material should be non-magnetic to not interfere with the motor or HEMS modules
SC.LOC.8	The material should be non-corrosive, so that the reflectibility doesn't change over time
SC.LOC.9	The area of the beam should be less than the width of the marks
SC.LOC.10	The localization system shall announce a lane-switch and gaps in the strip
SC.LOC.11	The localization system shall have the incremental distance as well as an absolute distance being measured
SC.LOC.12	The pod shall be included with an accelerometer, confirming the acceleration of the pod
SC.LOC.13	The laser shall be placed at an angle, such that the photodiode can be placed next to it. This angle is in the lateral direction, such that pivoting of the pod doesn't influence the readouts
ID	**Requirement S&C for Software**
SC.SW.1	The system should be able to transmit information from the main PCB to the ground station after the booting state.
SC.SW.2	A stop command must be implemented, such that the demonstrator can be commanded to come to a safe stop.
SC.SW.3	All data should be logged.
SC.SW.4	A graphical user interface must be implemented in order to visualize in real time all the data requested on the requirements.
SC.SW.5	A diagram of the system architecture and implementation of the graphical user interface should be included in the FDD.
SC.SW.6	A correct error detection and notification mechanism should be implemented. Any error on the system should be visualized on the graphical user interface including cause and timestamp.
ID	**Requirement S&C for Control**
SC.CON.1	The emergency braking mechanism should be automatically activated when the power or a control signal to the braking activation actuator is interrupted.
SC.CON.2	The emergency braking should be accessible by the person monitoring the speed.
SC.CON.3	It should NOT be possible in any state for the emergency braking system and propulsion system to be powered at the same time.
SC.CON.4	No control systems shall be located on track.
ID	**Requirement S&C for Communication**
SC.COM.1	The speed of the pod should be monitored during the demonstration.
SC.COM.1	Latency of the orders sent to the prototype must be less than 100 ms.
SC.COM.1	Data should be refreshed on the monitoring application with a minimum frequency of 2Hz.
ID	**Requirement S&C for Sensing**
SC.CON.4	The ground fault detection system shall be able to detect abnormally low electrical isolation between the electric system and the pod chassis. Value TBD.
SC.CON.4	The system shall be able to accurately determine the yaw, pitch, and roll of the pod.
SC.CON.4	The system is able to sense when levitation fails and pod drops off the track, initiating an emergency procedure.
SC.CON.4	The system is able to identify on which track the pod is located.
SC.CON.4	The emergency brake pressure shall be measured.

Table 2.1: Requirements Sense & Control

2.1 Risks

There are several things that can go wrong working towards, and at the EHW. These include technical as well as health risks. Since the whole subsystem functions on LV the health risks are not too severe. The technical risks mainly involve parts not functioning, any subsystem component not functioning could, and most likely will, result in the pod not functioning. A way to prevent these events from influencing our demonstration is to bring spare parts to the EHW, so that these parts can be replaced.

ID	Risk
SENS-TECH-1	Main PCB malfuctions
SENS-TECH-2	Components burn through
SENS-TECH-3	Ground Voltage Sensor malfunctions
SENS-TEST-1	After receiving a PCB, we cannot establish any communication
SENS-TEST-2	After receiving a PCB, certain functionales don't work
SENS-TECH-4	Laser gets missaligned, meaning the encoder strip cannot be read out anymore
SENS-COS-1	We go over our budget
SENS-SCH-1	We get behind on our schedule by 2 weeks or more

3 Concept

The design of the Sense & Control subsystem focuses on reliability and less on unprovoked innovation. The subsystem is however not the same as previous designs, since the pod is designed to performed a lane-switch there are enough changes that have to be made to the SC subsystem already.

3.1 Control

The Sense & Control department is responsible for all top-level control of the pod. There is one central node that decides what state the pod will be in, this is the main PCB. Every department is responsible for their own low-level control (f.e. the wavefunctions that are send to the motor). Figure 3.1 below shows all communication lines that are necessary for proper control. Every subsystem of the pod is connected with the main PCB which tells each subsystem what state the pod should transition to. Other important control mechanisms of the pod are the braking control (braking PCB), the temperature control (of the thermal system, the vacuum box and other parts of the pod).

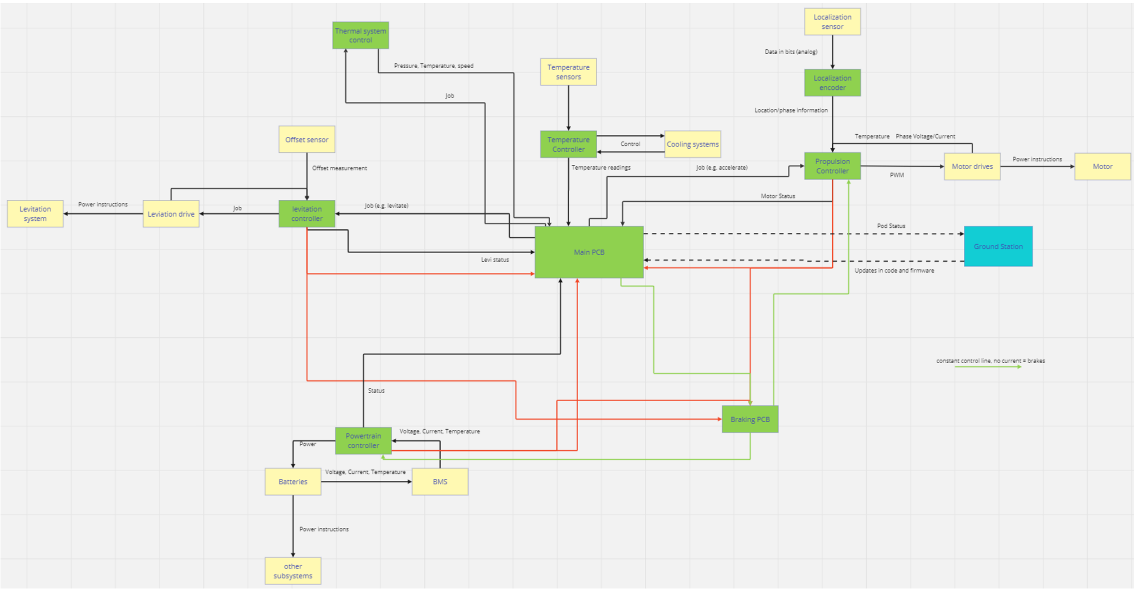

Figure 3.1: Top level control communication schematic

The black arrows correspond to data communication, the green lines are the 'sanity check' connections that the braking PCB has with each subsystem, and the red lines correspond to the error message lines, these transmit error messages with the causes

3.1.1 Main PCB

Main PCB will act as the hub to control all peripherals for control and communication of the overall Hyperloop POD. It will consist of the following systems integrated with it:

- MicroController STM32H743ZIT6

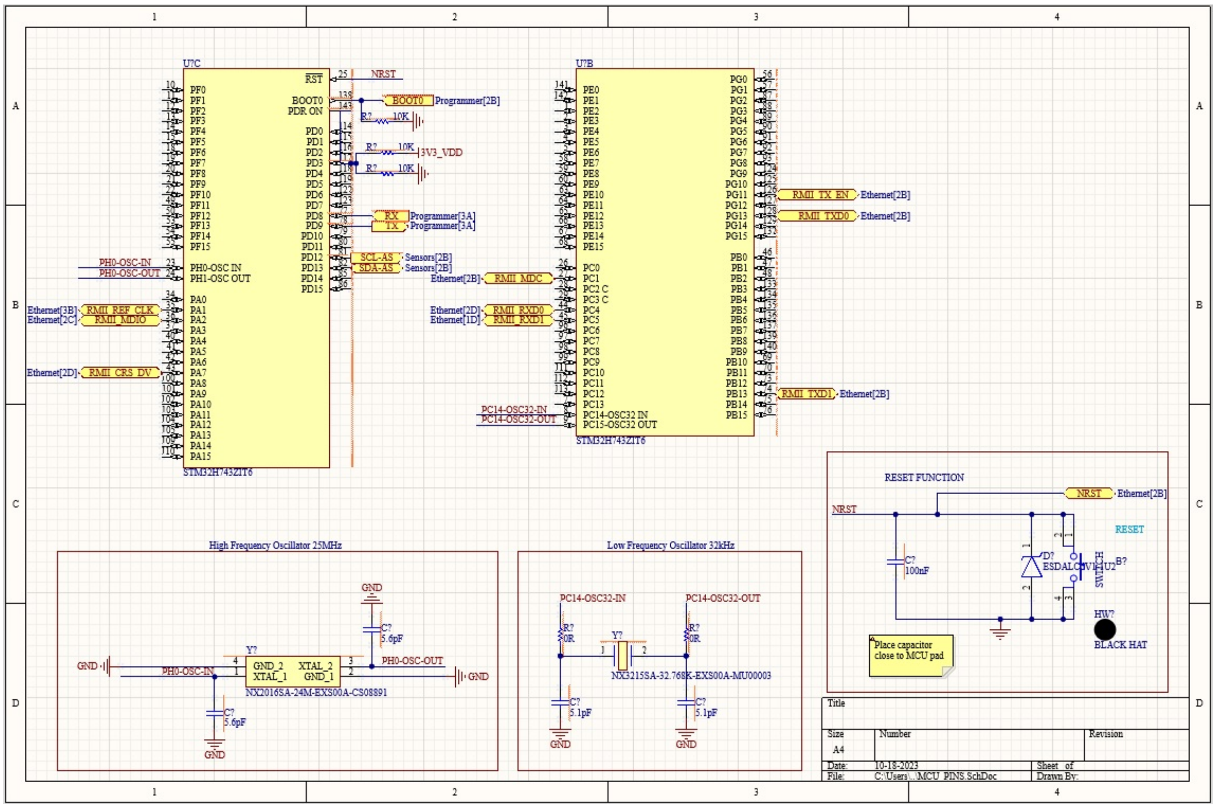

Figure 3.2: Configuration of Microcontroller Pins

NX2016SA-24M-EXS00A-CS08891 (24 MHz)

1. Main System Clock: This is likely used as the main clock source for the microcontroller, providing the clock pulses that drive the core CPU and peripherals. The 24 MHz frequency is a commonly used value for many MCUs.
2. Frequency Scaling: Depending on the architecture and peripherals, 24 MHz could be a base frequency that is scaled up or down to provide different clock domains within the chip.
3. USB Operation: If the MCU has USB capabilities, a 24 MHz crystal might be required for the precise timing required for USB data transmission.
4. Communication Protocols: Higher frequency crystals are commonly used for fast serial communications like SPI, UART, etc.

NX3215SA-32.768K-EXS00A-MU00003 (32.768 kHz)

1. Real-Time Clock (RTC): This frequency is almost always used for real-time clock functionality because 32,768 is 2^{15}, making it convenient for dividing down to 1 Hz for an RTC.
2. Low Power: 32.768 kHz crystals are used in low-power applications where you need accurate timing but not the computational power (and therefore energy consumption) provided by a faster clock.
3. Timers and Delays: Sometimes used for providing a clock source to timers that do not require the high speed of the system clock.
4. Watchdog Timer: Can be used to clock an independent watchdog timer that resets the MCU in case of a software failure.

- MicroController STM32H743ZIT6 Power

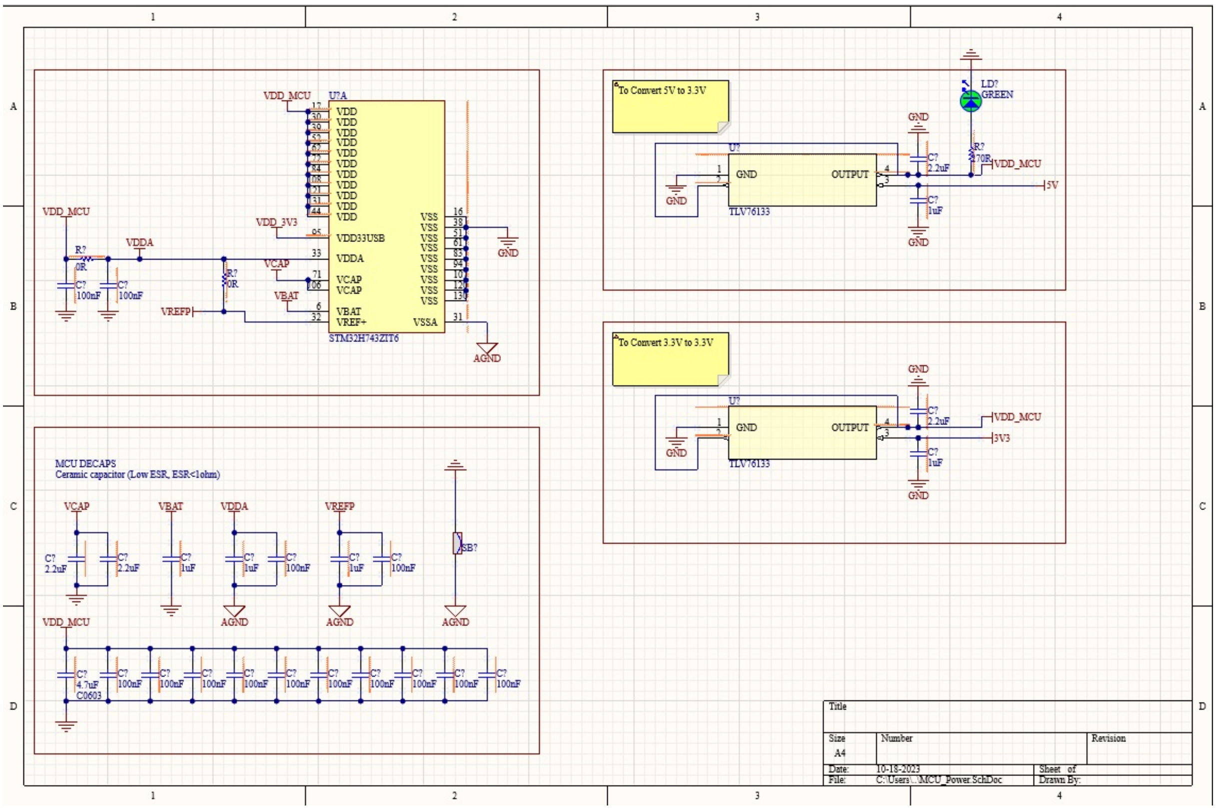

Figure 3.3: Design for Microcontroller Power

The Schematic consists of IC TLV76133 and decoupling capacitors ensuring a constant 3.3V DC Supply. ‘

- Ethernet

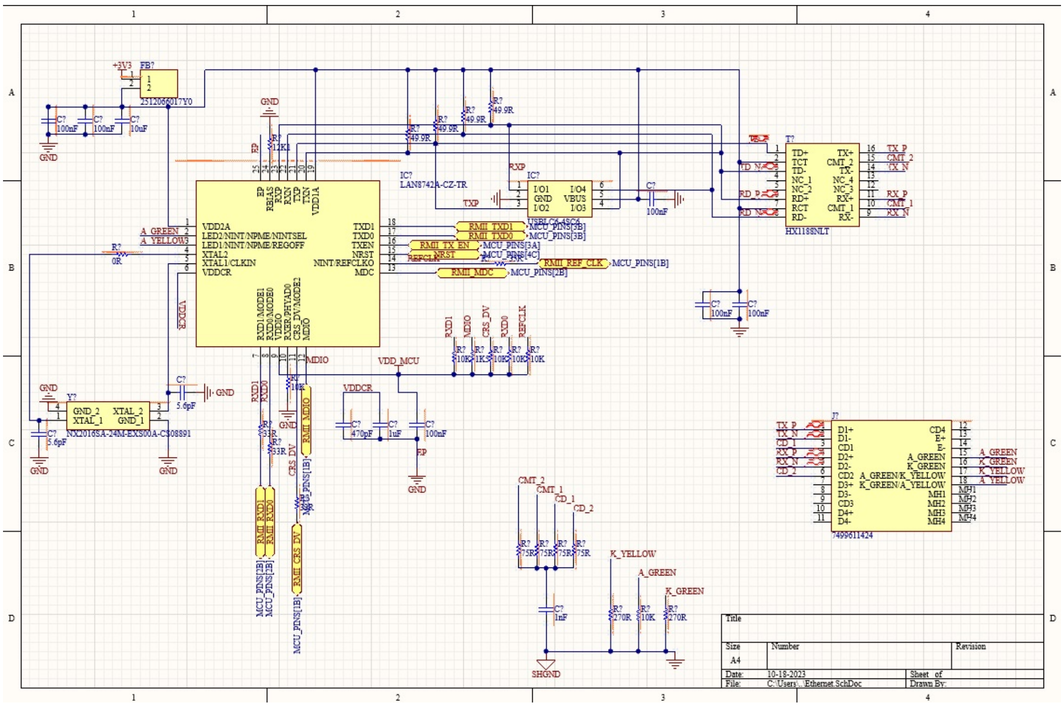

Figure 3.4: Design for Ethernet

LAN8742A-CZ-TR: It is an Ethernet PHY (Physical Layer Transceiver) integrated circuit. It is typically used in conjunction with an Ethernet MAC (Media Access Controller) in microcontrollers to implement network communications based on the Ethernet standard.

USBLC6-4SC6: It is a low capacitance ESD (Electrostatic Discharge) protection device, commonly used to protect sensitive electronic components from static electricity or other transient voltage spikes.

HX1188NLT: It is an Ethernet magnetic transformer module, commonly referred to as a "MagJack," which provides the required isolation and coupling for Ethernet communication. s

- Programming Over Air and STLINK-V3MINI

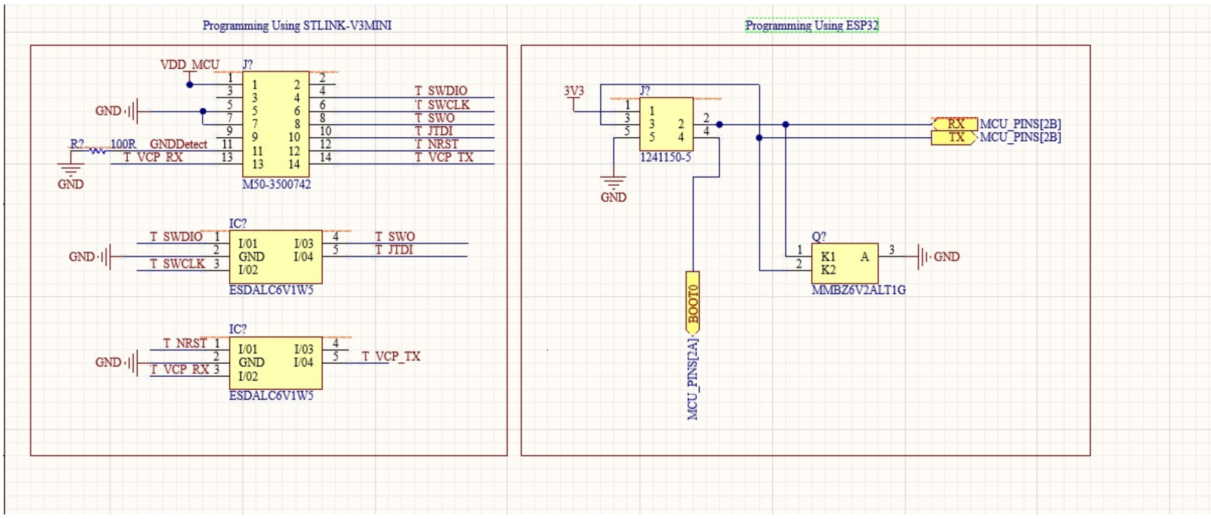

Figure 3.5: Programming Connectors

It consists of all the connections that are required to flash the microcontroller using STLINK-V3MINI (debugger module) and Over Air Using ESP32.

- Data Logging using SD Card

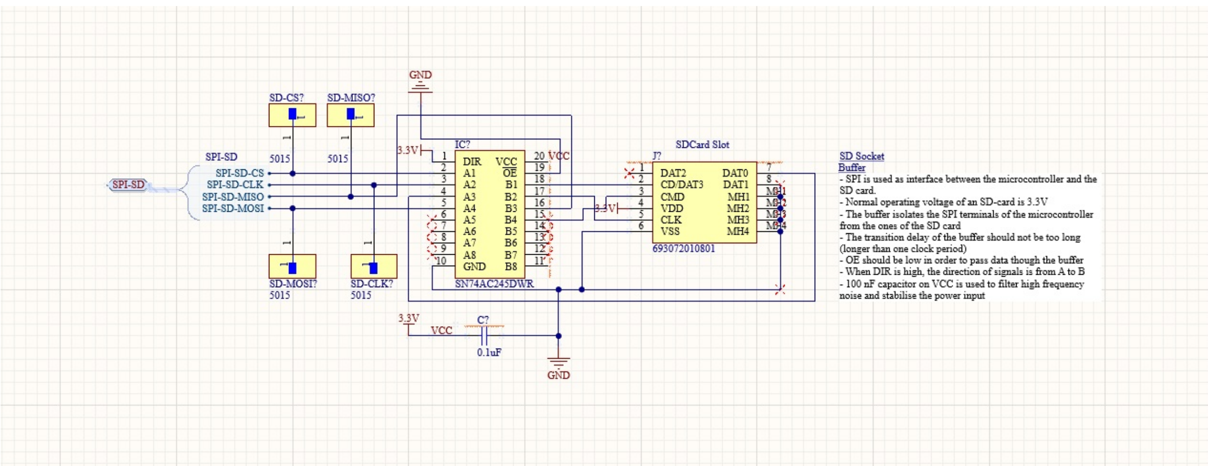

Figure 3.6: Design for SDCard

Includes all the peripherals for logging in data to SDCard.

- Sensors

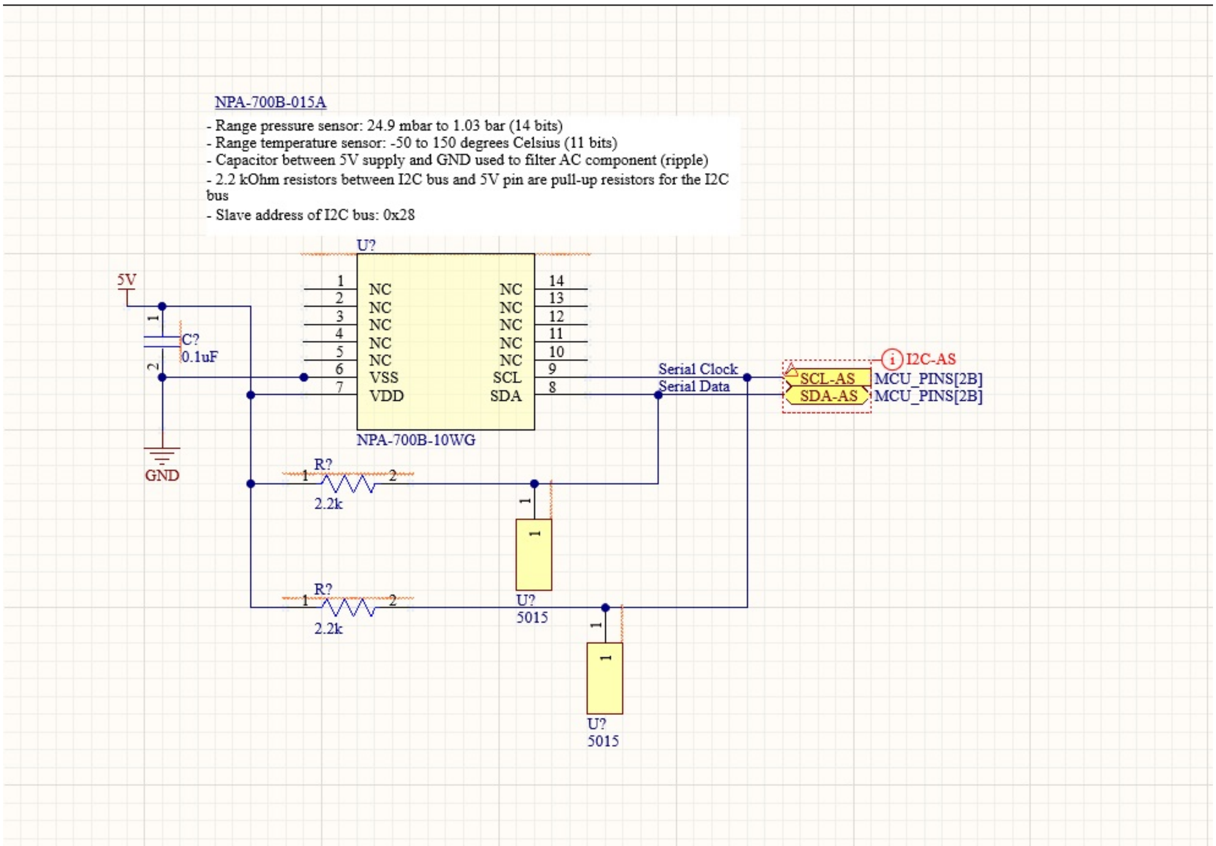

Figure 3.7: Design for Microcontroller Power

NPA-700B-10WG: It is a MEMS-based pressure sensor offered by Amphenol Advanced Sensors. This device is designed to provide high-accuracy pressure measurements in a compact form factor, making it suitable for various applications ranging from medical devices to industrial systems. It operates over a specified pressure range and offers digital output for easy interfacing with microcontrollers or other digital systems. Additionally, the sensor often features built-in temperature compensation, ensuring that pressure readings are accurate over a wide range of operating temperatures. Its low power consumption makes it ideal for battery-operated applications.

- Design of PCB The PCB will be a 6 layer PCB with 3 GND Layers which are Shield GND for Ethernet, Analog GND for analog signals and GND for microcontroller. Each layer will have GND Layers for minimizing EMI issues with high frequency data signals. The PCB will have CAN Connectors and Ethernet connectors at the borders of PCB for short distance and reliable communications. Testing points have been added for easy debugging of the PCB.

3.1.2 Braking PCB

The braking PCB controls the pneumatic braking system. It will likely be an analog circuit that is connected to all control systems.

- Constant monitoring the activity of other control systems (levi, propulsion etc.) and detect a loss of connection to these.
- In the case that any subsystem fails, the brakes should be deployed without using any active control.
- Shut down all HV systems in the case of any failure
- Log the reason for the activation and send an error message to the ground station

Logging data and sending error messages is something that only active components can do, good integration with the main PCB is thus necessary, or the decision for a "smarter" braking PCB should be made.

3.1.3 Localization PCB

The localization system works as described in Chapter 3.2. To enable this system to work a localization master will be created that can interpret the photodiodes signal. More specifically, the functionales of the localization master will be to:

- With the use of a photodiode, measure the modulated laser signal and convert it to a current signal.
- Convert the current signal to a voltage signal.
- Obtain the original modulation signal (which is carried by the laser).
- Using the moving average of the signal the two different cases (reflection and no reflection) are distinguished.

The way this is done is by using the setup in

3.1.4 Temperature and Pressure Control

The temperature of all subsystems need to be monitored and possibly controlled. The most important systems to do this for are:

- The vacuum box(es)
- The thermal control system
- The HEMS cores
- The EMS cores
- The propulsion motor

The BMS measures the temperature of the HV and LV batteries.
At locations outside of the vacuum box small temperature sensor nodes will be made. Using an ESP32 module together with a temperature and pressure sensor these parameters can be measured and communicated with the main PCB through WiFi. These modules will be connected to a LV power node and can be operated without data cables.
The thermal system's water temperature and pressure need to be monitored to determine its efficiency and operating status. Sensors for this purpose will be placed along these pipes.

3.2 Localization

To accurately determine the location of the pod in the direction it is moving in an optical encoder setup will be used.

3.2.1 Setup

The setup consists of an optical encoder strip (both an absolute and incremental one) together with a laser, a beamsplitter and a diode. The schematic in Figure 3.8 shows what the setup looks like.

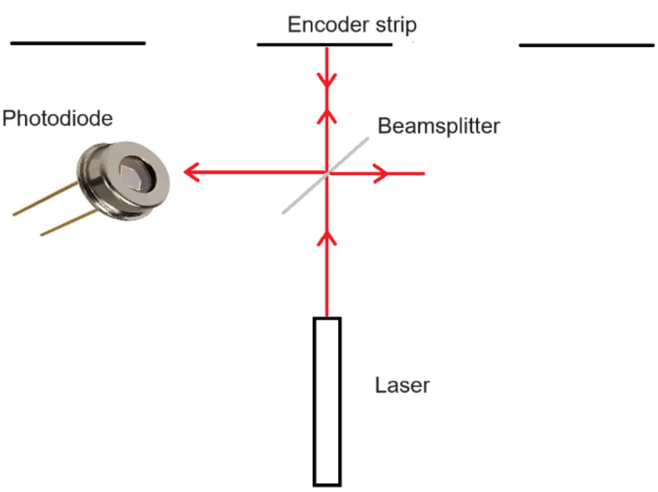

Figure 3.8: Setup of the localization system used to determine the location in the direction of movement.

The encoder strip consists of evenly spaced gaps in the direction of movement. when the lasers light meets the beamsplitter (which is placed at a 45° angle) 50 % of the lasers light (intensity) gets reflected off the plate, and 50 % goes through. The laser light then gets reflected back and 25 % gets reflected from the beamsplitter into the photodiode. If there is a gap in the strip no light gets reflected back and thus a low intensity is measured by the photodiode. Measuring the amount of times that a photodiode measurement flips from low intensity (0) to high intensity (1) we can determine the amount of distance the pod has traveled.

3.2.2 Beamsplitter

The beamsplitter is used instead of putting the laser at an angle and simply putting the photodiode next to the laser. The reason for this is that in the latter case, the offset of the pod with respect to the strip will influence the place the light lands on. In the setup used, this offset doesn't matter and thus only one axis needs to be taken into account for.

3.2.3 Encoder strip

The encoder strip is presented in Figure 3.9. The left strip represents the incremental encoder, counting increments of 3mm. The right strip is the absolute encoder, this strip signals stator teeth (to reset the incremental encoding) and signals the direction in which the pod is going using unsymmetrical gaps.

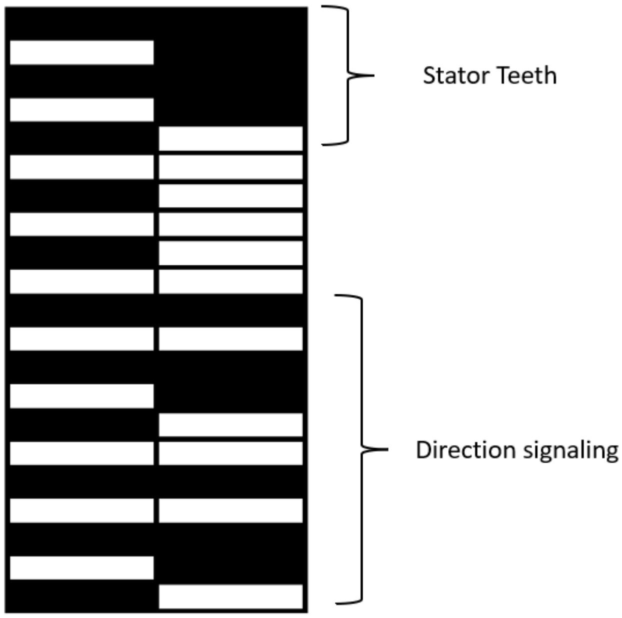

Figure 3.9: Encoder strip

3.2.4 Circuitry

To obtain the modulation signal from the laser signal, the signal needs to be processed using a series of components. The graphic in Figure 3.10 shows the circuit used to do this.

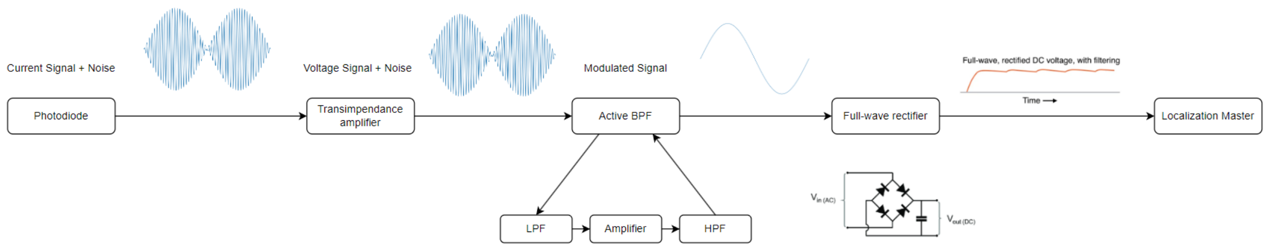

Figure 3.10: Signal prcoessing schematic

First the photodiode current signal gets converted to a voltage signal using a transimpendance amplifier. Secondly, The noise and other frequency (other than $f_{modulation}$) signals get removed using an active band-pass filter (BPF). The last step is to rectify the signal (AC to DC) to obtain a constant DC value corresponding to the power of the sine wave. Using this circuit it will be possible to destinguish a sine signal from no signal.

3.2.5 Pod layout

Because during the lane switch the localization strip cannot be present (due to crossing beams). The laser setup as described will be placed on the front as well as the back of the pod. This way, we can 'bridge' the crossing beams and always have localization.

3.2.6 IMU

As a sanity check, an IMU sensor will be installed in the localization system, monitoring of the second derivative of the localization system corresponds to the IMU readings.

3.3 Sensors

In the early design only the most integral sensors with specific implementations are included, due to the project still being in the preliminary design phase, a lot of necessary variables are still unknown (like f.e. the braking pressure).

3.3.1 Ambient Pressure and Temperature

Simple temperature and pressure sensors will be placed on the edge of the pod to monitor the environment that the pod is operating in. This is important to ensure that a the pod is operating in a safe environment and no anomalies are present. In the detailed design document these will be presented.

3.3.2 Lateral and Vertical Offset

The operating range of the vertical air gap is 11.5mm - 18.5mm, for the lateral air gap this is 3-13mm. The **Althen Sensors FDRF603HS Series** has a 70kHz measurement frequency, a range of 2mm - 750mm and a non-linearity of $\pm$ 0.1 %. If we can calibrate for this linearity and correct for it, this sensor would be perfect for its application.

3.3.3 Safety Switches

Two types of safety switches will be used to ensure that the pod doesn't go out of its intended operating space. Additionally, a safety button will be installed on the outside of the pod as well as on the ground station. If any of these switches or buttons get activated the HV and LV systems will be disabled and the emergency brakes activated. The latency for this can be no longer than 100ms.

3.3.4 Ground Fault Detection

The HV power supply has maximum voltage and current outputs of 400V and 20A. The decision has been made to not die this year and thus we need to make sure no current leaks into the parts of the pod that should be at 0V (like the chassis). Ideally, there would be infinite resistance between HV components and grounded components because in that case the current between the two will be 0A. This sensor is simply for security reasons and will be bought. Any ground fault should lead to a HV system shutdown.
It is very important that when a ground fault is detected that this information is communicated to the operator. Since other subsystems can also initiate a shutdown knowing that a ground fault was the cause is crucial.

3.3.5 Thermal Management

The thermal system consists of pipes with water running around the pod, to ensure that the thermal system is working like it is supposed to the temperature as well as the pressure on different parts of the system should be monitored, this way it is possible to measure (only big) leaks, the temperature (and thus efficiency) of the system and the mass flow speed. The control of the system is propulsion's responsibility. In the detailed design document the exact sensors will be presented.

3.4 Software

As a team, it was decided to choose Rust as our main programming language. It has been a great choice so far, with detailed and intuitive documentation from the embedded rust book[1], and ease of collaboration between team members. The other programming language considered was C++, so below is a list of comparisons and trade-offs between both. As shown in the table Rust has many advantages over C++ while serving similar purposes and still supporting embedded

Aspect	**C++**	**Rust**
Year of Initial Release	1985	2010
Memory Safety	Prone to memory safety issues (e.g., segfaults) without careful management	Designed to prevent memory safety issues, ensuring safety through its ownership and borrowing system
Community and Ecosystem	Large community and vast ecosystems with a large number of libraries	Growing community and ecosystem, increasing number of libraries and tools available
Performance	High performance, close to the hardware	Comparable to C++ in terms of performance, with safety guarantees
Tooling	Established tooling with various compilers and debuggers available	Modern tooling including package manager (Cargo), built-in test suite, and more
Learning Curve	Steep, especially understanding pointers and manual memory management	Steep, due to its unique ownership and borrowing system, but with extensive documentation and community support

Table 3.1: Comparison of C++ and Rust programming languages

systems greatly. For the scale of our project a true object-oriented program structure is not necessary and would convolute and complicate the process of programming each controller. This change was also decided due to the team's thrill to innovate and as Rust gets more and more popular there is a strong belief among the department members that it will be as influential as C++ in the future.

3.4.1 Communication

Communication between microcontrollers will use either CAN or Ethernet. The trade-off between the two is examined below (Table 3.1), however a final decision has not been made. The discussion is only between point-to-point CAN and point-to-point Ethernet, since a CAN bus will create all sorts of planning and integration problems between the departments, while being unscalable and potentially dangerous if an emergency message from one controller is delayed.

Table 3.2: Comparison between CAN and Ethernet

Attribute	**CAN (Controller Area Network)**	**Ethernet (Local Area Network)**
Type	Single-line bus	Multi-line bus
Speed	Top Speed with CAN 1 Mbps and 10 Mbps with CAN FD.	Top speed of 4096 Mbps (4 Gbps).
OSI model	Physical and Data Link layer of the OSI model.	Physical and Data Link layer of the OSI model.
Point communication	Multiple controllers can access the bus from any point	Point-to-point communication
Cable	Twisted Pair cable	Co-axial, twisted pair (unshielded), or fiber optics
Master-Slave?	Multi-Master	Master-Slave
Cost	Cheaper to implement as it does not require many expensive add-ons.	Often requires a dedicated chip or a UNIX capable system and takes up many resources, making it costly.
Physical Layer	CAN is a voltage differential-based bus that requires transceivers and a termination resistor on both ends.	Ethernet is a current-driven bus inductively coupled and driven by transformers on both ends.
Messages priority	Messages are prioritised using the arbitration process.	Messages are transmitted randomly, hence the possibility of collisions
Packet Collision	Handles packet collision in real-time.	No provision to take care of collisions.
Collision	CAN is CSMA/CA: avoids any collision within the data coming from nodes.	CSMA/CD: Ethernet tries to transmit, and if a conflict occurs, performs random back-off.
Traffic planning	Traffic planning on a CAN network is simple: lower IDs always have priority over higher IDs.	Ethernet has no priority on the MACs; the transmission here is random.

3.4.2 Finite State Machine

A Hierarchical Event-Driven Finite State Machine will be used to control the various behaviors of the pod. It is an extension of the classical Finite State Machines (FSM) that are widely used to control embedded systems. The extensions came from the need to quickly react to sensor data and commands (which represent events in the system), hence the 'event-driven'. Furthermore, since there are multiple independent systems in the pod, each with its own specifics and its own Finite State Machine, the high-level abstraction is the usage of the Hierarchical Finite-State Machine that encapsulates all different control behaviors of the different subsystems, while maintaining the semantics of the main FSM. Any subsystem finite state machine can be easily showcased by extending the main Finite State Machine since the states have been created in such a way that every subsystem is dependent only on the main state of the machine, and not on other subsystem states. This way, each subsystem is exposed only to the pod controller Finite State Machine, removing inter-dependencies between subsystems, hence making the system scalable horizontally (the workload can be distributed between multiple 'machines' rather than making the current 'machine' handle more load). For simplicity, this chapter will explain only the high-level pod FSM Figure 3.2, with emphasis on the main states and the conditions that lead to transitions between them.

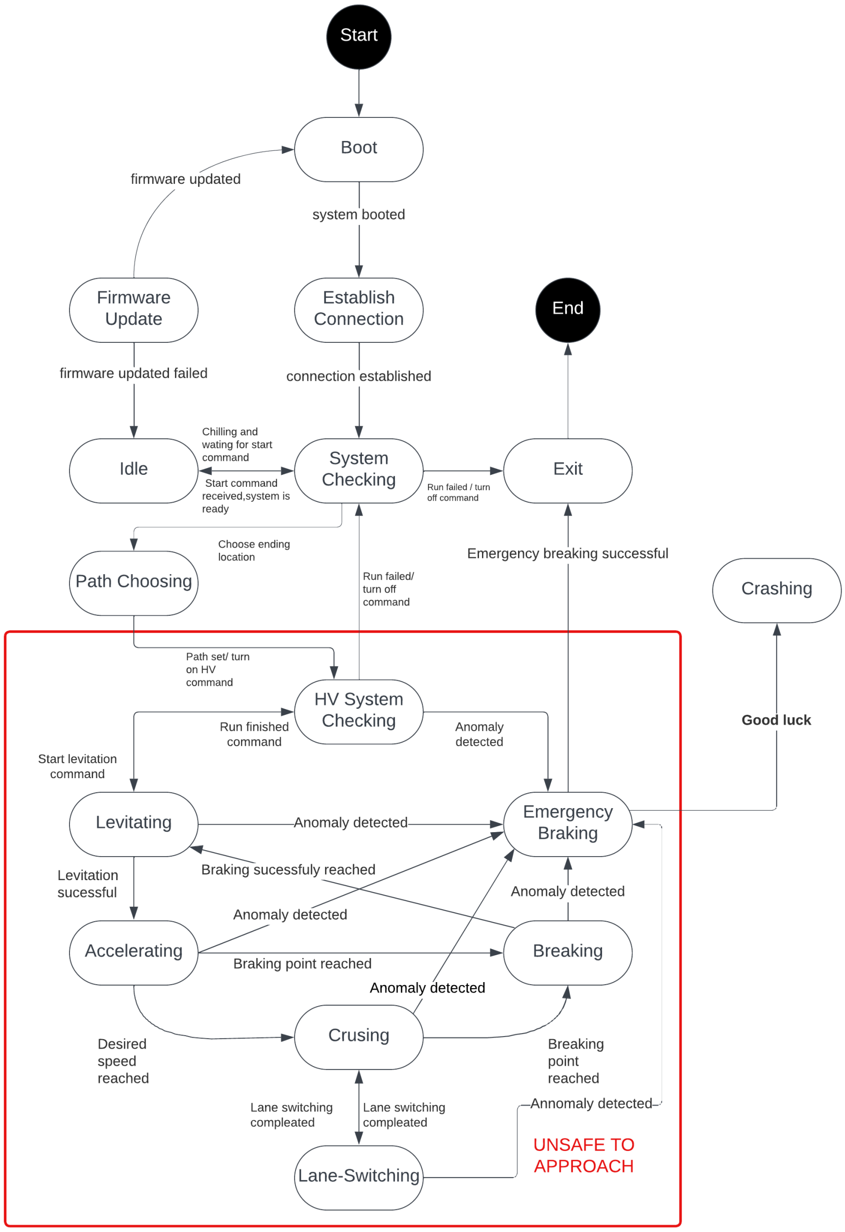

Figure 3.11: The Core Finite State Machine of the pod with all the states and event transitions

The system will start and it will go to the booting state, in this state the subsystem should be booted and it will move to establish connect

3.4.3 Ground Station

What to monitor, over the air protocols, all features The ground station software is responsible for all external communication to the pod, sending control commands and receiving all sensor information for monitoring the pod.
The ground station is going to be a cross-platform software that:

- establishes a TCP/IP connection with the wireless module of the main PCB
- can transmit a .bin (or .elf) to the wireless module, which will in turn flash the main PCB
- can transmit any from a list of predefined commands, such as stop, accelerate, cruise etc, which when received by the main PCB will switch to the appropriate state
- can receive, process, and display appropriately all sensor data streams from the pod to create a short summary of the state

The ground station should be the primary human-to-pod interface.

3.4.4 STM32

About coding on this mf, flashing memory etc. debuggers etcetc.

4 Process

During the preliminary design phase of the project the Sense & Control department focused on doing research as well as getting hands-on experience with the necessary control hardware. Additionally, designs of the localization system, the internal communication network and the main PCB were made. In the development of the subsystem the primary focus is to create a mock environment in which the hardware can be tested as soon as possible. The advancements and designs as well as the future development processes of the different subsystem components are described in the subsections below.

4.1 Trade off

Every decision is trivial

Table 4.1: Trade off for custom designs

Criteria	Althen FDR603HS	Althen FDR603
Frequency	70kHz	9.6kHz
Non-linearity	0.1%	0.05%
Range	2-750mm	2-1250mm

4.2 Cost

The most costly components of the subsystem are represented in the table below. Parts that are very likely to be obtained in-kind are not included (like cables and connectors). Note that this estimate is **very** rudementary.

Table 4.2: Cost estimation template

Expenses	Cost in euros
Laser	600
Localization Strip (test track)	400
Localization Strip (full track)	800
photodiode PCB	300
Localization master	300
laser housing	70
Lateral offset sensors	2500
Vertical offset sensors	2500
Main PCB	610
Ambient pressure sensor	5
Ambient Temperature sensor	5
Main PCB components	300
Microcontrollers	50
ESP32 (multiple	80
Ambient pressure sensor (ESP module)	15
Ambient Temperature sensor (ESP module)	15
Total cost	8850

5 Model

Table 5.1 below shows all components with high masses and their mass, together with the total mass of the subsystem.

Amount	Component(s)	Total lower mass (kg)	Total average mass (kg)	Total upper mass (kg)
1x	Main PCB	0.4	0.6	0.8
1x	Braking PCB	0.2	0.35	0.5
1x	Localization PCB	0.2	0.35	0.5
-	Cables	2	4.5	7
3x	Lasers (Localization)	0.2	0.6	1
1x	Laser housing	0.3	1.15	2
-	Sensors	0.5	0.75	1
	Total	3.8	8.3	12.8

Table 5.1: Mass estimates of the Sense & Control subsystem including all components with the most weight.

6 Detailed Design Phase

The main goal of the DDP is to create a mock environment with which we can simulate the pod's environment and we can start training our subsystem. To prepare our system for the other subsystem's testing phase, we need to have a working subsystem that can control these subsystems. Therefor the goals of the detailed design phase are the following:

- Complete the Main PCB design and have code running on it. The functionales of the PCB should include all from the prelim design as well as
- Make the localization system ready to be used in the propulsion track testing
- Finish the braking PCB schematic, all functionales as described in the Braking PCB subsection should be included in this schematic
- Manage a system of 10+ sensors from different sources and communicate sensory values to the GS to be monitored.
- Finish a design/map of all temperature sensor placements as well as identify which LV power nodes they can be connected to.

These goals can still be changed, since other departments don't fully know yet what their subsystems will exactly look like and what their testing setup will look like. Constantly monitoring their plans and changing accordingly will help us prepare for testing as well as possible.

7 Production

Below is a list of all components that need to be fabricated and a description of their production plan.

- PCBs: These will be produced and assembled by EuroCircuits since we have a partnership with them. Our second choice would be JLC PCB.
- Localization encoder strip:

8 Testing

In order to facilitate testing of the sense & control subsystem without the fully constructed pod, we aim to create a full emulation of the pod environment. Connecting all the microcontrollers and sensors from Figure 3.1 to each other outside the pod will allow thorough testing to take place during the detailed design phase for the rest of the pod. Using a raspberry pi to mock sensor inputs and spy on outputs will allow it to unit test the microcontrollers throughout the development phase.
Testing of the control algorithms consists of test cases, passed in as sensor input from the raspberry pi and asserted by the raspberry pi as communication output from the microcontroller. This process will effectively test our custom hardware wherever it is used, but additional verification might need to be put in place for custom hardware components.

8.1 Localization

The localization system consists of an optical and an embedded system. Both these need to be tested. Firstly, the embedded circuit can be tested using an oscilloscope that creates a mock signal with which we can then test the response of the circuit on high frequency, modulated signals. To change parameters even after we have produced the circuit, spare resistors and capacitors will be obtained with different parameters than the ones in the circuit. This way, components can be switched out and changed when necessary, leading to quick iterating and testing.
When the optical and embedded systems work well together, and it is possible to read the encoder strip, the setup can be tested on a small track together with the motor.

8.2 PCB testing

PCB testing can be done two ways where in one way test points can be added and checked on the Digital Oscilloscope for the integrity of signal as well as the frequency of the signal as well (especially for communication protocols). The other way is to have visual indication like LEDs which will be provided with supply voltage to check whether the supply provided is constant as well to the specified voltage level also.

9 Safety

safety is optional. if it works i am happy

10 Conclusion

Bibliography

[1] "The embedded rust book." (), [Online]. Available: https://docs.rust-embedded.org/book/intro/index.html.