S^2OUTH Documentation

written by WüSpace e. V.

15.06.2026

Contents

1. Mission Goal

The South project is to design and implement a reliable rocket avionics platform capable of acquiring, processing, and transmitting comprehensive flight data throughout all phases of the launch. The system shall integrate data from multiple onboard sensors to accurately estimate the rocket's position, velocity, and altitude in real time.

In addition to flight state estimation, the South system will capture and transmit live video footage during flight. As physical recovery of the rocket is not planned, the system shall ensure that all essential telemetry (including maximum altitude, peak velocity, and other critical parameters) is securely transmitted to the ground station via a robust radio communication link.

Through the successful execution of these objectives, South will demonstrate advanced capabilities in telemetry, sensor fusion, and communication systems, thereby contributing to the advancement of student-led aerospace research and the interconnection between German Aerospace Teams.

2. Avionics system overview

The Avionics system is made up of an air segment and a ground segment. The air segment is responsible for data acquisition, state estimation, recovery pyrotechnic's control and communication via radio and umbilical. The ground segment is responsible for data retention, mission control and communication with the air segment. This also includes a custom antenna and video tracking system.

3. Air segment

The South avionics stack is made up of a set of pcb's each with their own dedicated function. The system is modular by design, each pcb is a module with only one function, when additional functions are required additional modules can easily be added to the system. Communication between the modules happens via two central redundant CAN-FD busses.

3.1. Connectivity

Since the avionics stack's functionality is highly distributed along a lot of different dedicated subsystems, a lot of thought has been put into the connectivity between these subsystems. Both power and data connections need to be especially robust, as a failure of one of those will inevitably lead to the complete failure of the entire system.

3.1.1. Power architecture

The system is powered by one of three sources. One is the auxiliary power received via the umbilical cable during launch preparations. The other two are lithium ion battery packs compromising of five 18650 cells each. These battery packs provide the system with a redundant power source during flight.

All power inputs are run through the EPS which takes care of regulating the correct voltage levels needed by the different subsystems. Each subsystem has a dedicated output rail to isolate them from noise induced by other subsystems. Furthermore the EPS is able to individually enable and disable different power sources and sinks.

Most subsystem's PCBs use HARWIN M80-5400442 connectors, which are screwed into position.

3.1.2. Can architecture

For redundancy, two parallel CAN busses connect all pcbs and enable communication between all subsystems. For this, a long cable with in line board connectors is used. Both ends of the bus are terminated through 120Ω resistors. All controllers support CAN FD, which enables higher data rates and bigger individual frame sizes than traditional CAN.

3.1.3. Interface to ground segment

The avionic system is connected to the ground segment via three different ways. The main communication method is the radio link which provides downlink on the launch pad and in flight. Additionally on the launch rail an umbilical cable is connecting the rocket to the ground segment providing communication and power to the avionic system. As a backup the tank temperature and pressure can be read out via a secondary cable on the launch rail.

3.1.3.1. Radio Links

The radio communication consists of two separate links: a 430-440MHz (70cm) telemetry link, and a 5.8 GHz video link. Both are purely downlink, featuring no means of communicating from the ground to the rocket via RF.

For more information about the links see Table 2 and Table 3 in Section 3.4.5 and Section 3.4.6.1.2

3.1.3.2. Umbilical Wire

The rocket avionics are connected to the ground segment via an umbilical cable which provides the rocket with power, launch detection and an Ethernet connection with the Ground segment. The wire is connected to the rocket with a non locking fisher 102 series 9-pin connector that is passively disconnected via tension as the rocket starts. The connectors are selected for low-friction to prevent any lateral loads on the vehicle during the initial launch phase.

3.1.3.3. Backup sensor connection

Additionally, a separate ground support connection allows readouts of the most critical sensors, without the need of our avionics system working. This is just a backup for the most critical situations, and to judge if the rocket is safe to approach even when other communication is lost.

3.1.4. Interface to secondary avionics

The Lower Sensor Board features a UART Output in the form of a 2x2 Molex NanoFit connector. This will be used to forward sensor information and GNSS position as required by the experimental BLAST avionics system provided by a HyEnd subteam.

3.2. Structure

The avionics structure stack houses the main PCBs required for the onboard avionics system, including RocketHD, power supply, blackbox and umbilical board. In addition, the stack carries four 5.8 GHz WIFI antennas for RocketHD, one custom VHF antenna and at least one GPS antenna. The structure consists of multiple circular levels made from GFRP sandwich plates. These plates are stacked vertically and connected by GFRP hollow tubes. The connections are both bolted and bonded to provide sufficient stiffness and vibration resistance. The complete avionics structure will be verified on a shaker before flight. From bottom to top, the stack is built around four main levels. The lowest level acts as a mounting interface and is firmly connected to a structural ring at the top of the tank section. The battery powering the avionics stack is mounted to the underside of this plate. On the upper side, the main PCBs are mounted to the GFRP plate using standoffs. The second level carries the four 5.8 GHz WIFI antennas used by RocketHD. These antennas are mounted in 3D-printed housings and mechanically fixed to the plate. The third level supports the custom VHF antenna. It has a copper groundplane to improve the radiation characteristics of the antenna. The fourth and uppermost plate carries the GPS antenna. This places the GPS antenna at the top of the stack and provides the best possible position for GPS reception. The GPS antenna is also fitted with a copper ground plane for improved reception.

Figure 1: Full avionics stackup during a range test

3.3. Software

The avionics system's compute is purely based on stm32 microcontrollers running embedded programs based on Embassy, the cooperative concurrency framework written in Rust. This guarantees us minimal overhead due to scheduling operations while still allowing for the conveniences of a non-blocking hardware abstraction layer. Additionally the uniformity of the Embassy ecosystem makes it possible to reuse code across multiple different microcontroller models. In order to minimize the amount and severity of potential errors the software architecture of onboard systems (excluding the umbilical board and mcu) should fulfill the following paradigms:

3.3.1. Onboard software architecture paradigms

1. All onboard mcu's do not contain global allocators (heaps). This preempts out of memory errors, as the maximum stack size can be asserted at compile time. Instead static memory allocation is used.
2. The software should not depend on arbitrarily defined memory sizes, this means that when using limited size (stack based) collections the size should be calculated at compile time and (if applicable) automatically synchronized between crates. The exception to this rule are peripheral/dma buffers.
3. No constant, definition or configuration should be defined twice, if a constant, definition or configuration is needed across more than one crate it needs to be extracted into a common crate.
4. The functionality of the system should not depend on persistent states. Every state should be recoverable if a reboot of the relevant mcu occurs. If possible, stateless logic is preferred.
5. In order to maximize efficiency the system inputs (peripherals/sensors) should all be run in continuous mode, and the mcu's timings should depend on interrupts and timings dictated by the inputs. Arbitrary timer delays shoud be avoided. If the rate of input is higher than the required rate for calculations/communications, software oversampling (averaging) should be applied.
6. In order to maximize the value of sent/saved data raw values should be sent/saved and only converted to on the ground. In case where this principal collides with point 4. (as in cases where mcu specifc factory calibrated values are needed) point 4. takes precedence.

3.3.2. Chell system

In order to fulfill these paradigms meta-programming is employed as much as possible. For communications between boards a custom minimal non self descriptive serialization format is used. Serialization for higher order types is implemented via macros. The maximum possible bytesize of the types after serialization is calculated in constant context in order to guarantee correct fixed sizes for byte collections. All communicated values have a macro generated definition containing automatically assigned ids (unsigned int16) for communications via CAN as well as generated static string based addresses for communications via NATS message servers on the ground. Furthermore macro generated beacon types allow for batching of serialized values into a coherent structure with a maximum size known at compile time. Beacon types also contain identification and crc mechanisms for use with radios. All of these macros as well as some system independent generic utility functions and structs are bundled in the chell system crate.

3.3.3. Common

South specific chell values, chell definitions and chell beacons as well as common peripheral configurations and data handling tasks are bundled in the south common crate. This includes sensor configurations needed for data conversion on ground as well as time synchronization logic. Furthermore all conversion functions for the gse are defined in this crate and assigned to their respective chell definitions. All subsystem crates implement this crate, if two subsystem repositories implement the same commit hash of south common, compatibility in configuration and communication between the subsystems should be guaranteed. The chell system is always included as an exposed dependency of south common.

3.3.4. Subsystem software

All subsystems employ the usage of hardware watchdogs in order to reboot the mcu in case a hard fault occurs or one of the subsystem's tasks is blocking concurrent execution (necessary on non preemptive concurrent systems like embassy). Every software crate starts by initiating all required peripherals and sensors. During runtime every input and output, as well as every processing task should be separated into different async tasks, and only communicate via embassy-sync concurrency synchronization primitives and structures such as channels.

3.3.5. State estimation

In order to estimate the state of the rocket the Upper Sensor subsystem uses an Extended Kalman Filter (EKF). The aim of the EKF is real-time state estimation. This includes the position, velocity, orientation and sensor biases of a rocket during flight using data fusion. GPS provides absolute position data but has a slow update rate. The IMU on the other hand provides high-frequency data but suffers from drift due to sensor noise accumulating over time through mathematical integration. The barometer provides a stable relative altitude until it exceeds the maximum permissible measurement limit. To overcome these problems a direct EKF is implemented.

3.3.5.1. Architecture

Indices	State Symbol	Mathematical Description	Unit
0...2	p_x, p_y, p_z	Position in NED System	Meters (m)
3...5	v_x, v_y, v_z	Velocity in the local NED System	m/s
6...9	q_w, q_i, q_j, q_k alternative: q_w, q_x, q_y, q_z	Orientation quaternion (Transformation from NED to Body Frame)	dimensionless
10...12	$b_{\text{acc}_x}, b_{\text{acc}_y}, b_{\text{acc}_z}$	Axis-specific accelerometer biases	m/s^2
13...15	$b_{\text{gyro}_x}, b_{\text{gyro}_y}, b_{\text{gyro}_z}$	Axis-specific gyro biases	rad$/s$
16	b_{baro}	Mathematical bias for the barometric altitude sensor	Meters (m)

3.3.5.2. Algorithm

Like any normal Kalman Filter, the Algorithm has two steps. The prediction phase and the update phase. The prediction includes a state transition where the state vector is projected into to the future by the time step dt. Next the covariance propagation is calculated by state_transition_jacobian().

$$P_{k-1|k} = AP_{k-1}A^T + Q$$

$$P = \text{Prediction covarianz matrix}$$

$$A = \text{State transition jacobian matrix}$$

$$Q = \text{Process Noise}$$

In the update phase the filter compares the actual sensor measurements with the theoretical prediction from the estimated system state. To optimize computational efficiency and handle multi-rate sensors, a dynamic masking mechanism (mask) is implemented. This mask evaluates if a specific sensor channel has updated since the previous time step. If no new data is present, the corresponding rows and columns are bypassed.

3.4. Subsystems

This Chapter goes into greater detail about the functionality and scope of each individual subsystem.

3.4.1. Electrical Power System (EPS)

The EPS (Electrical Power Supply) provides centralized power management, regulation, and distribution. It accepts power from 2 independent 5s 18650 battery packs and an additional external power supply. Each input can be individually enabled or disabled

When multiple power inputs are enabled, the EPS automatically selects the active source based on input voltage. The input with the highest voltage supplies the system. If enabled inputs reach approximately the same voltage, they are paralleled and share the load. The EPS continuously measures the voltage of each input as well as the current flowing from each source into the board to support source selection, load sharing, and fault detection.

The selected input voltage is converted to regulated 5 V rails using onboard DC-DC converter. Which is then distributed to the other PCBs. Each regulated output is individually switchable and can be enabled or disabled via command.

In addition to the regulated outputs, the EPS provides two direct battery-voltage outputs. These outputs forward the selected input voltage without regulation and are intended for boards that require the full battery voltage. In this system that would be the RocketHD video streaming board and the lower sensor board. The direct outputs can also be individually switched on or off.

All power management, monitoring, and control functions are handled by an onboard STM32G0 microcontroller. The MCU manages input enable states, source selection, output switching, and real-time voltage and current monitoring. Communication with the rest of the system is performed over CAN busses using onboard CAN transceivers, enabling command, status reporting, diagnostics, and fault signaling.

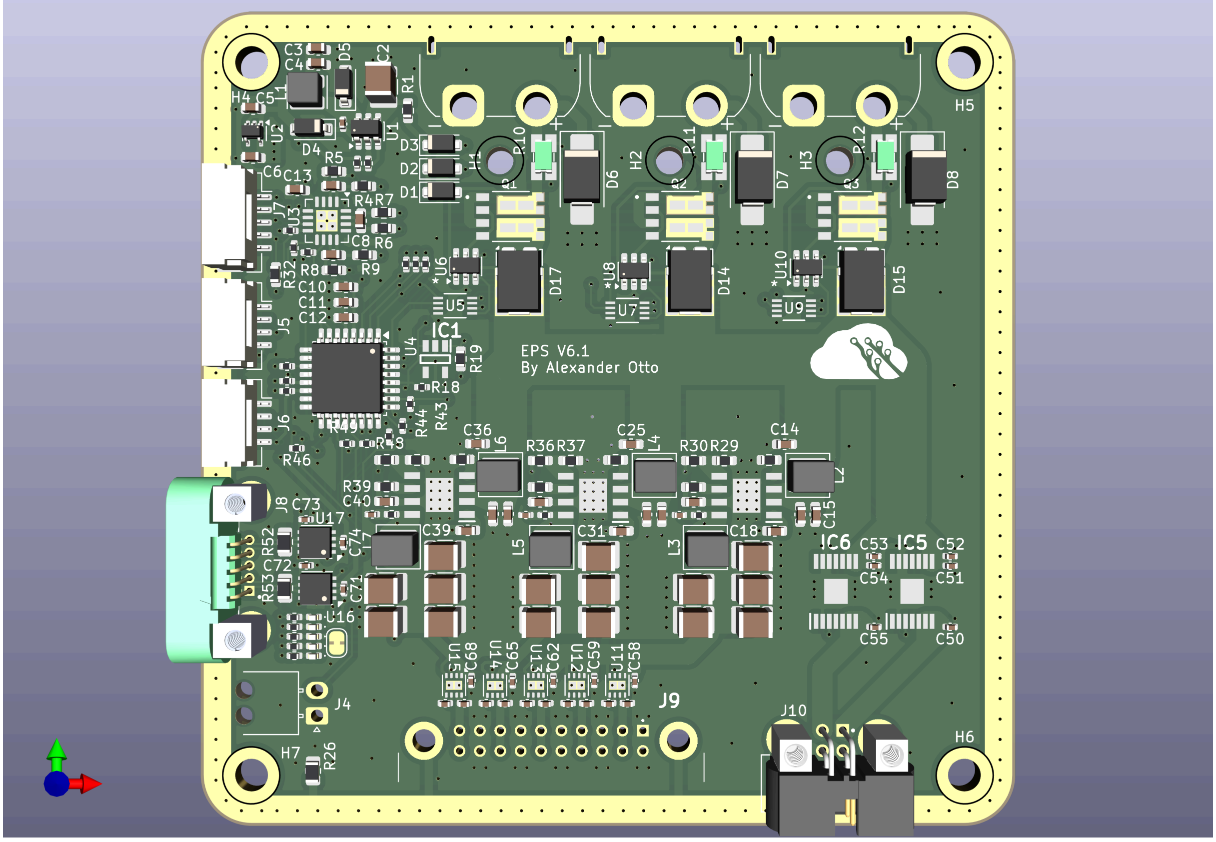

Figure 2: 3D view of the EPS PCB

3.4.2. Umbilical

The umbilical subsystem on the avionic stack is responsible for interfacing with the ground support segment while the rocket is on the pad. Its main functions include

1. Providing power to the avionic stack before launch
2. Bidirectional communication between the avionic stack and the ground segment during launch, without needing a radio link

3. Providing a physical means of detecting liftoff by disconnection of the Umbilical Cable

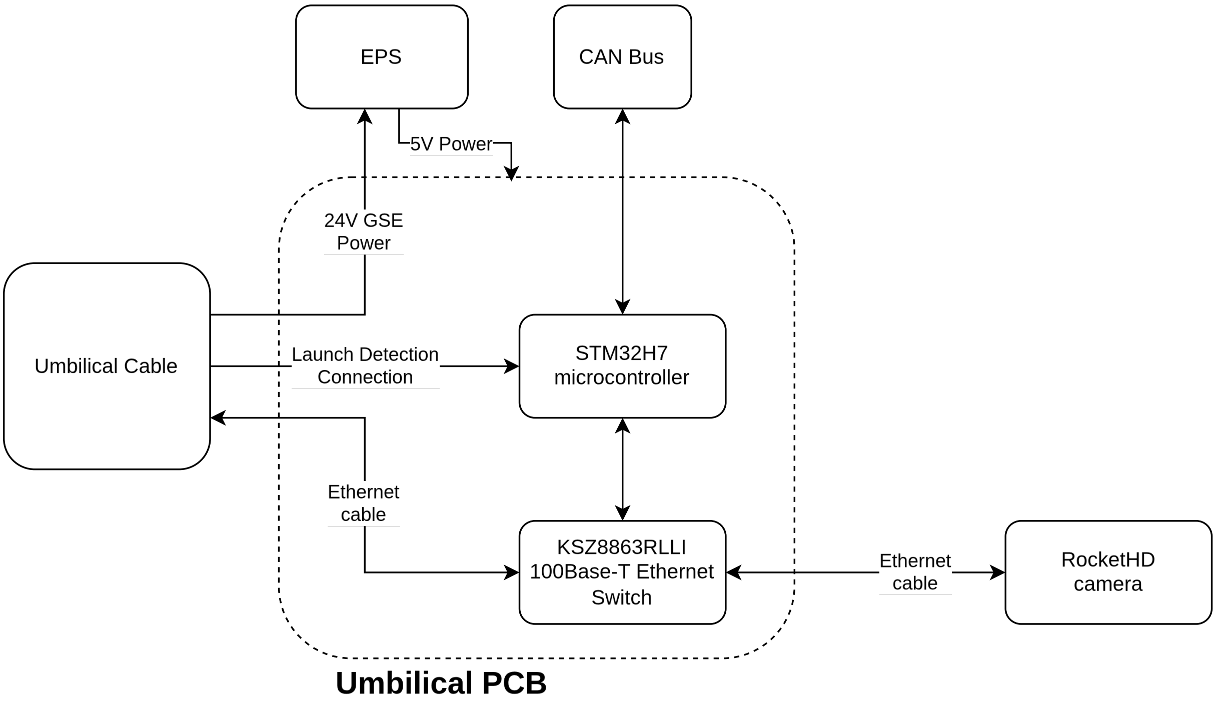

Figure 3: Functional diagram of the Air Segment's Umbilical Subsystem

The central component of the umbilical subsystem is a custom PCB, housing all electronic components required for the umbilicals function. This mainly includes a STM32H7 microprocessor, a KSZ8863RLLI 100Base-T Ethernet switch, and multiple power converters.

This PCB interfaces with a cable towards the rocket's umbilical connector, a 24V DC power cable towards the EPS, a 5V DC power cable from the EPS, an Ethernet cable towards the RocketHD camera, and the CAN Bus.

Name	Function	Type	Pin count
Umbilical Connector	Interface to the ground segment	Molex NanoFit	8
24V DC Power	Forward ground power to the EPS	Molex NanoFit	2
5V Power	Power to the umbilical subsystem	Standard Power Connector	4
CAN	Messaging to the umbilical subsystem	Standard CAN Connector	10
Ethernet Cable	Messaging to the RocketHD camera subsystem	Molex NanoFit	4

Table 1: Connectors of the umbilical subsystem

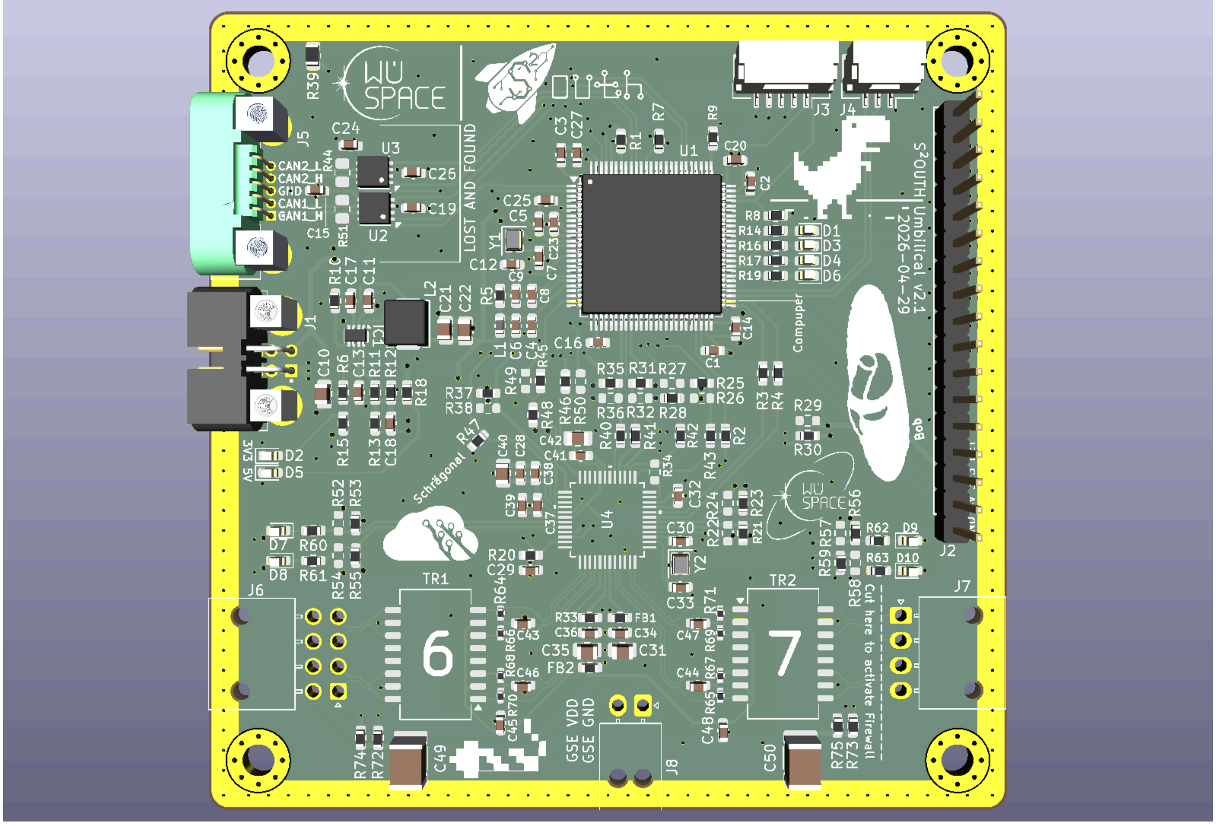

Figure 4: 3D view of the Umbilical PCB

3.4.3. Sensor Board & H7 skith

The primary sensors of the avionics system are split onto two PCBs:

The **Upper sensor board** contains sensors for position estimation and navigation, and performs the necessary sensor fusion to accomplish this task. It contains two high rate, high precision MEMS inertial measurement units (LSM6DSV32) providing the MCU with gyroscope and accelerometer data, a three axis magnetometer with external spools (RM3100) that our university already validated for space applications on their satellite Sonate-2, a high accuracy barometer we previously tested on our Daedalus project flying higher than 80km on the REXUS rocket, as well as a Phoenix GPS receiver from the DLR. These sensors allow us to retrieve high quality data at a rate of several kiloherz.

The **H7 skith** is an unofficial successor to the skith[1] (skip the harness) computation board from the university of würzburg, providing us with a modular computation board featuring a STM32H723VG chip with a single core and clock speeds up to 550 MHz. This Board can be used in combination with the abovementioned sensor board or standalone if dedicated computation capabilities are needed (for example for local trajectory estimation). When connected to the sensor board it is able to provide a high accuracy 6DOF state estimation.

---

[1] https://www.informatik.uni-wuerzburg.de/en/aerospaceinfo/main/projekte/skith/

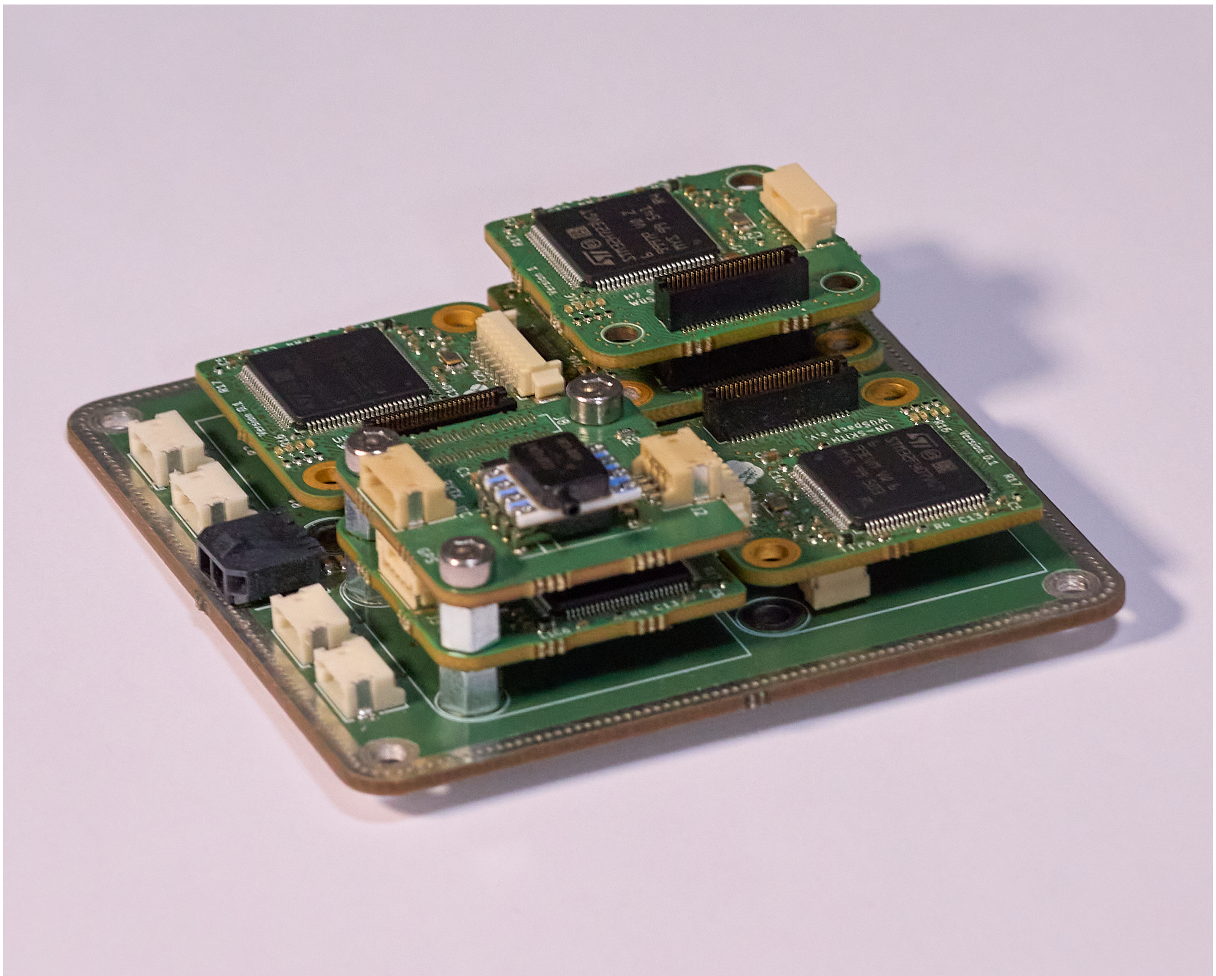

Figure 5: Picture featuring four H7 skiths, two of them equipped with an Upper sensor board

3.4.3.1. Phoenix

The DLR Phoenix Miniature GPS receiver[2] provides the system with a reliable source of absolute position, created with high dynamic environment in mind and free of the limitations of usual COTS GPS receivers.

3.4.3.2. IMU

For inertial sensing two LSM6DSV32 IMUs are used on the sensor board. It combines a 3-axis accelerometer and a 3-axis gyroscope with configurable full-scale ranges up to $\pm 32g$ and ± 4000 dps and output data rates in the kHz range, making it well suited for the highly dynamic environment. A secondary high-range $\pm 32g$ accelerometer channel is also available and can be used in parallel to improve robustness during high acceleration events. The sensor further provides low-noice measurements, integrated digital filtering and embedded features such as motion detection, which is important for vibration robustness during flight.

3.4.3.3. Barometer

For barometric pressure the sensor board uses the HSCMRNN030PA which is a high accuracy absolute pressure sensor. This sensor is already tested on our Daedalus project flying higher than 80km on the REXUS rocket.

---

[2] https://www.dlr.de/en/rb/research-operation/research-projects/flight-dynamics-navigation-and-orbital-sustainability/gnss-technology-and-navigation/phoenix-miniature-gps-receiver

3.4.3.4. Magneto

For the magnetometer the RM3100 is being used. It is a high resolution, high poling rate 3 axis sensor that uses external coils and is already validated for space application by the JMU University for their use on the satellite Sonate-2.

3.4.4. Recovery system

The Pyro board is responsible for igniting the recovery ignitors that deploy the parachute.

The fire mechanism is powered by two 18650 lithium-ion cells connected in series, providing a voltage of 8.2V (full) to 6V (empty). Thy ignitors are safe below 0.45A and fire above 1A. Their internal resistance is between $0.4 - 0.8\Omega$

These cells were chosen for their mechanical robustness and reliability. The rest of the board is powered externally via a power connector with 5 V. Onboard regulators step this voltage down to 3.3 V where required.

An STM32G0B1 microcontroller is used on the board. It communicates with the rest of the system via CAN and generates the fire and safe control signals. It also reads the two Sense_A signals. In addition, the ADC is used to monitor the voltage of each individual battery in order to detect potential battery discharge.

The board contains two identical ignition circuits, each dedicated to firing one initiator. Each circuit includes a Safe input, which is inverted to increase system safety.

The fire signal drives an N-MOSFET, which switches on and allows current to flow through the initiator, thereby igniting it. Both initiators are connected to the same battery and are additionally secured with a shorting plug that must be removed before launch. Diodes are used to prevent current from flowing in the wrong direction. Using the Sense_A points, the system can verify whether the initiators are correctly installed.

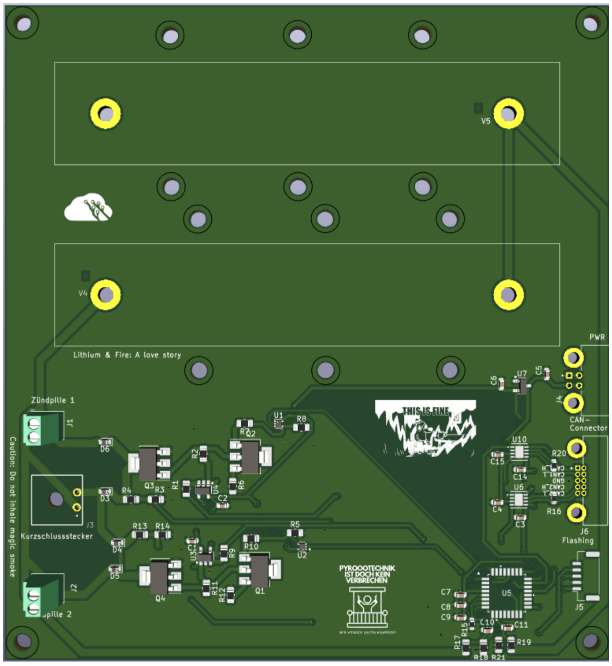

Figure 6: 3D view of the Pyro PCB

3.4.5. Video transmission system (RocketHD)

RocketHD is our video transmission system based on the OpenIPC project, a project that aims to provide open source firmware for IP cameras from different manufacturers. This enables running custom software directly on the IP cam chips to add to their functionality. One of the main use cases for this firmware is in FPV drone video transmission using Wifibroadcast, a package injection technique for general broadcasting of data packets using commercial-of-the-shelve wifi chips.

Our video camera is an Eachine Sphere Link, which records up to 4k30, has an h265 encoder on board, and has usb and ethernet interfaces. We use this device in combination with our custom pcb, which adds a microSD card slot, a CAN connector to the rest of our system, and power circuits. For video transmission, we use a Realtek wifi module (rtl8731bu), that is split into four different tx paths. We amplify each signal via the SE5023L amplifier featuring a compression point of 34dBm (2.5W).

Each signal is then fed into a patch antenna with 6.7 dBi gain, and about a $\pm 45°$ sending pattern on both axis. This antenna is a product of Team Black Sheep, a well known supplier for fpv drone hardware.

Since we use wifi modules for transmission and reception, we are limited to the default Modulation and Coding Scheme (MCS) of wifi. With only one spatial stream (hardware dependent) we will use **MCS index 1**, which uses QPSK modulation, and has about **12 Mbit/s** of data rate. Our bandwidth is 20MHz. The frequency used results from the wifi channel, which we chose as 149 (**5.745 GHz** center frequency).

Another side effect of using wifi cards is OFDM, which divides the data into multiple overlapping frequencies, resulting in possible power spikes at the intersections. Since the amplifier is hard limited, this data is lost if the amplifier is driven at its compression point. For that reason, we keep a distance of about 8dB[3] from the actual 34 dBm compression point.

The detailed RF signal path from the transceiver to the independent patch antenna feeds is structured as follows:

Component Stage	Gain / Loss	Absolute Power
Wi-Fi Card Max Output	—	+17.0 dBm
Fixed Attenuator Stage	−14.0 dB	+3.0 dBm
Stacked Wilkinson Splitter Network	−7.4 dB	−4.4 dBm
Active PA Gain Stage	+32.0 dB	+27.6 dBm
PCB Trace & Interconnect Insertion Loss	≈−1.0 dB	+26.6 dBm

1. **Total Peak Envelope Power (PEP):** Across all four parallel PA channels, the combined peak power evaluates to approximately $10.04\ \mathrm{W}$. This remains safely below the legal $75\ \mathrm{W}$ PEP limit allocated for the 6 cm amateur band in germany.
2. **Effective Radiated Power (ERP):** The four 6.7 dBi patch antennas are physically arranged in an orthogonal circular cluster to provide 360° horizontal coverage. Because the elements point in opposing vector directions, their wavefronts do not combine constructively in the far-field to increase directional gain.

The peak Effective Isotropic Radiated Power (EIRP) for any single spatial sector is evaluated as:

$$\mathrm{EIRP} = 26.6\ \mathrm{dBm} + 6.7\ \mathrm{dBi} = 33.3\ \mathrm{dBm} \quad (\approx 2.13\ \mathrm{W})$$

Converting this metric to Effective Radiated Power (ERP) relative to a half-wave dipole:

$$\mathrm{ERP} = 33.3\ \mathrm{dBm} - 2.15\ \mathrm{dB} = 31.15\ \mathrm{dBm} \quad (\approx 1.3\ \mathrm{W\ per\ sector})$$

Accumulating the total omnidirectional thermal energy radiated into the horizon yields a combined total of roughly **5.2 W ERP**.

---

[3] This 8dB distance is called Peak-to-Average-Power-Ratio (PAPR)

Frequency	5.735-5.755 GHz
Power (ERP)	5.2W
Modulation	QPSK
Bitrate	12Mbit/s
Bandwidth	20 MHz
Category	amateur radio

Table 2: Radio Capabilities of the Video Subsystem

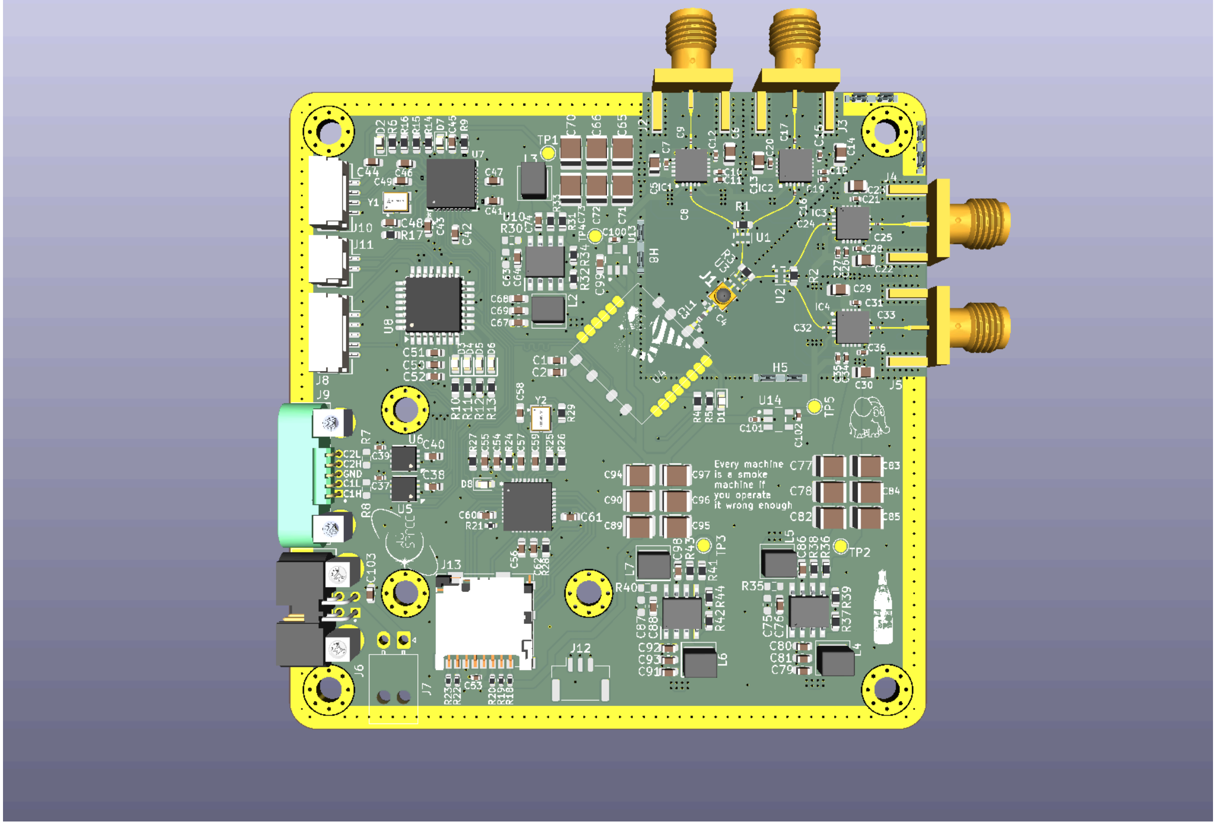

Figure 7: 3D view of the RocketHD RF PCB including a slot for the wifi card, four amplifiers and an stm32g0b1ke

3.4.6. Telemetry transmission system

The telemetry transmission system is made up of two custom radio boards (RocketLST) based on the OpenLST open source satellite radio. Both radios are combined via a power combiner to feed a singular main telemetry antenna.

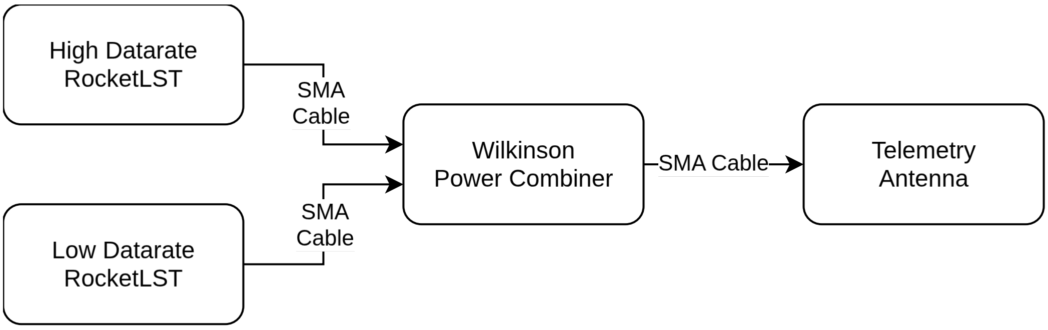

Figure 8: Functional diagram of the Telemetry Segment

3.4.6.1. Telemetry Sender (RocketLST)

The telemetry subsystem operates on custom-build hardware in the 70cm amateur radio band. The subsystem on the air segment features two PCBs with radio transceivers – operating at different baudrates and different frequencies (Section 3.4.6.1.2), a PCB for combining the two radio signals and matching the antenna (Section 3.4.6.3), and an end-fed antenna (Section 3.4.6.2)

The telemetry sender sends packets on-demand – the board computer has full authority over when and what packets are being sent. That way, radio silence can simply be achieved by not sending packets from the board computers.

3.4.6.1.1. OpenLST

The RocketLST PCBs are based off of the *OpenLST* project by *Planet*.[4] This project includes open-hardware PCB design files and an open-source firmware toolchain, providing an open radio solution for telemetry in the 70cm amateur radio band.

The *OpenLST* PCB features a *CC1110* radio MCU from *Texas Instruments*, a *RFFM6403* amplifier by *Quorvo*, an SMA jack, two UART pin headers for connectivity, a two-pin 5V DC power connector, and a debug connector for the radio MCU. This represents a good baseline for an amateur telemetry system, and initial verification and prototyping was done on the base system. However, for the proper avionic stack, many modifications and extensions proved necessary.

3.4.6.1.2. RocketLST

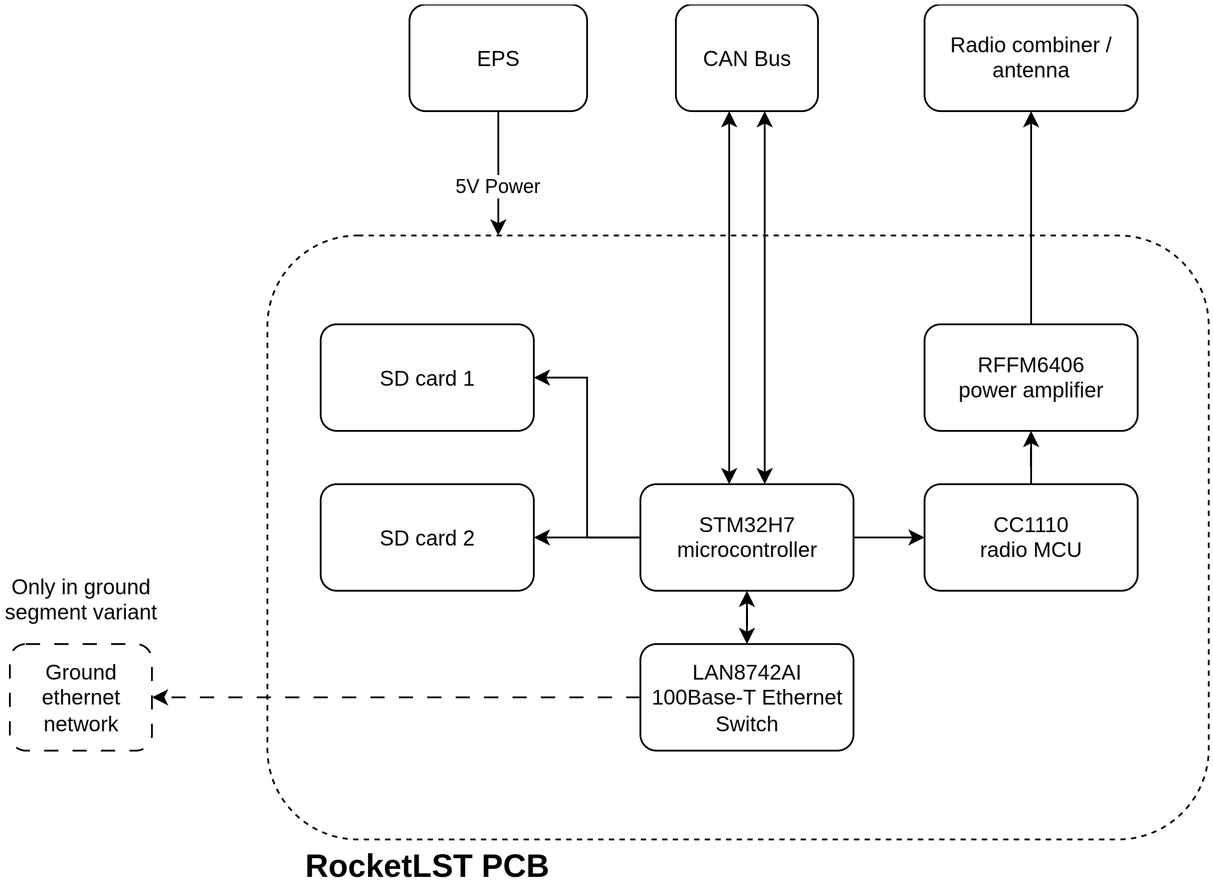

Figure 9: Partial functional diagram of the RocketLST PCB

---

[4] https://www.planet.com/pulse/planet-openlst-radio-solution-for-cubesats/

Like the *OpenLST* PCB, the RocketLST PCB is build around the *CC1110* radio MCU from *Texas Instruments*, a Sub-GHZ radio MCU, operating in multiple ISM and amateur radio frequency bands below 1Ghz. Instead of the no longer available *RFFM6403*, the *RFFM6406* is used, boosting the transmit power up to 1.5W. The board has been extended by an STM32H7 microprocessor, Ethernet and FD-CAN interfaces, SD-Cards, changed connectors.

Frequency	430 MHz - 440 MHz
Power (ERP)	up to 1.5W
Modulation	2-FSK, GMSK, MSK, OOK
Baudrate	up to 250 kBaud
Bandwidth	up to 562.5 kHz
Category	70cm amateur radio

Table 3: Radio Capabilities of the RocketLST PCB

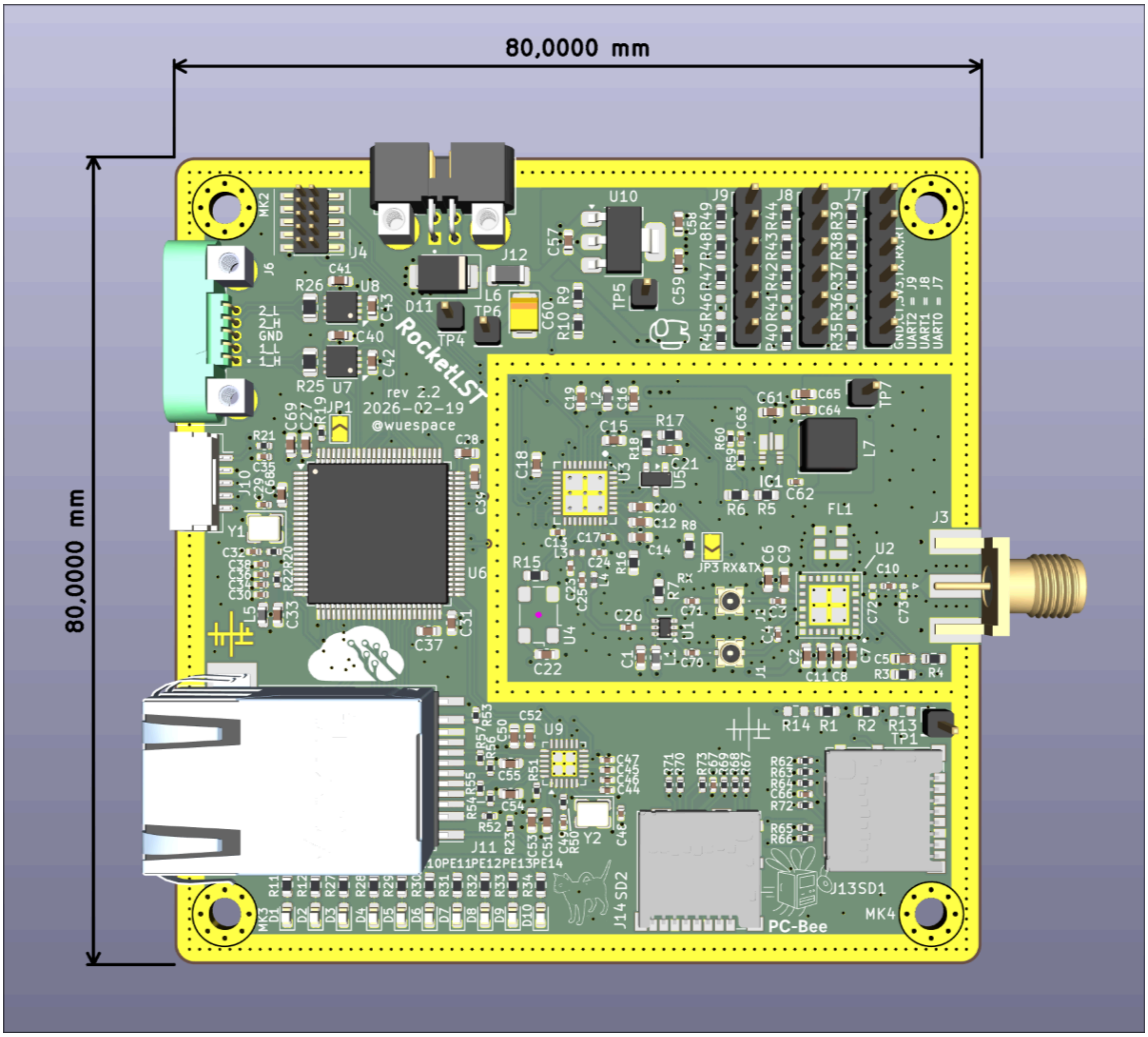

Figure 10: 3D view of the RocketLST PCB

Frequency	For testing: 434.200 MHz
Power (ERP)	1.5W
Modulation	GMSK
Baudrate	50 kBaud
Bandwidth	562.5 kHz
Category	70cm amateur radio

Table 4: Preliminary Radio Settings of the High Datarate RocketLST

Frequency	For testing: 437.123 MHz
Power (ERP)	1.5W
Modulation	GMSK
Baudrate	7.145 kBaud
Bandwidth	60.267 kHz
Category	70cm amateur radio

Table 5: PreliminaryRadio Settings of the Low Datarate RocketLST

3.4.6.2. Telemetry Antenna

As the telemetry antenna, a custom end-fed antenna is utilized. It is approximately 17.3cm long (1/4 of our telemetry system's wavelength) and is made out of a solid copper rod.

It is located in a reserved section in the nosecone of the rocket, and ensures the signal generated by both RocketLST boards is transmitted in an approximately isotropic fashion.

3.4.6.3. Wilkinson Power Combiner

As the avionics stack features two telemetry radios, but only one telemetry antenna, a power combiner is required to combine both signals before transmission.

For this we created a custom PCB featuring a lumped element Wilkinson Combiner, and a matching network for the antenna. The specific values were chosen for the center frequency of 435 MHz. The inductors for this were calculated as $L \approx 26nH$ and the capacitors before and after the inductor $C \approx 5.2pF$[5]

Due to sending two distinct waves on different frequencies and therefore phase mismatched signals, the input signal power is cut in half, where one half is dissipated as heat in the resistors.

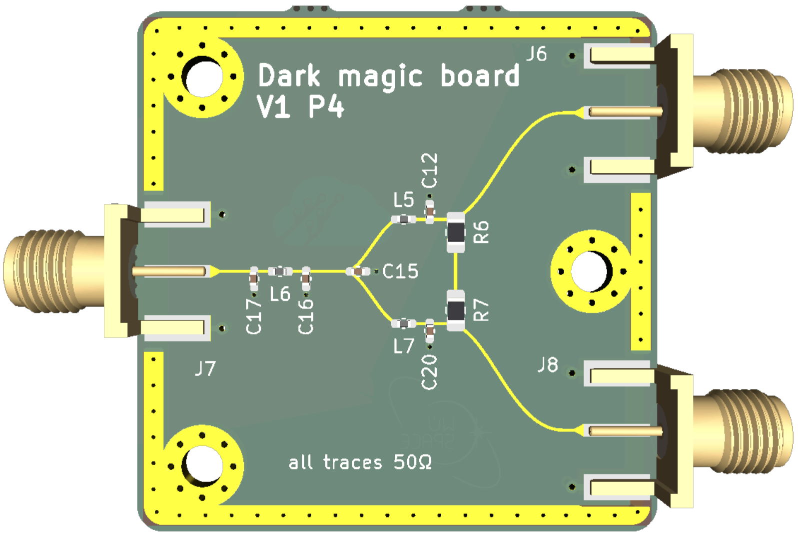

Figure 11: Custom lumped element wilkinson power combiner

---

[5] Wilkinson Calculator used: https://leleivre.com/rf_wilkinson.html

3.4.7. Blackboxes

The Blackbox is based on a modified M7 Skith sensor board, using its basic infrastructure for the STM32H7 chip to add independent power and data storage capabilities to ensure safe logging and shutdown in the case of total loss of supply power.

3.4.7.1. Blackbox Hardware

The PCB is equipped with a separation diode from the main power supply and two high-capacity capacitors, allowing for encapsulation and up to 10s of independent power upon separation of the main power supply. Additionally, a microSD card is soldered directly to the board for data storage.

For mechanical security, the PCB itself is encapsulated in an aluminum casing with 4mm thickness on all sides and potted in polyurethane (MG Chemicals 8800) for additional shock and impact protection.

Dimensions Assembled (exuding cables and Screws): Size: 48mmx48mmx23mm (±0.15mm) Weight: TBD Mount: M4*30+ (M4*25 Possible but not Recommened)

3.4.7.2. Pinouts Black box

External Cable

Pin	Function
1 Yellow	CAN1+
2 Green	CAN1-
3 Blue	CAN2+
4 White	CAN2-
5 Black	GND
6 Yellow	SWO
7 Green	NRST
8 Blue	JTMS
9 Black	GND
10 White	JTCK
11 Black	GND (Supply)
12 Red	+5V (Supply)

SD Card

Pin	Function
PD2	CMD
PC12	CLK
PC8	Data0
PC9	Data1
PC10	Data2
PC11	CD/ Data3

Notable

Pin	Function
PE4	Supply Power Sensor
PB3	SWO
PA13	JTMS
PA14	JTCK
PD0	CAN1_RX
PD1	CAN1_TX
PE2	CAN1_STBY
PB12	CAN2_RX
PB13	CAN2_TX
PE3	CAN2_STBY
PD12	LED_G (Deactivate after Potting)
PD13	LED_Y (Deactivate after Potting)
PD14	LED_R (Deactivate after Potting)
PD15	LED_B (Deactivate after Potting)

3.4.8. Lower Sensor Board

Between the tank and engine of the rocket, specialized sensors are used to measure tank pressure and temperature.

The Lower Sensor Board houses measuring circuits for those sensors. Two Wika A10 sensors are read out directly via a 16 bit ADC to deliver high accuracy pressure data. The temperature is measured via a PT1000 resistive probe, that is measured via a 1mA constant current source. The generated voltage is also measured via the 16 bit ADC. As a backup, the whole board can be externally powered and read out without disturbing the rest of the avionics system. The collected data is fed to the rest of our system via long CAN wires. These raceway wires run alongside the tank to the top of the rocket. There are also wires to feed the battery voltage required for the sensors and readout circuit.

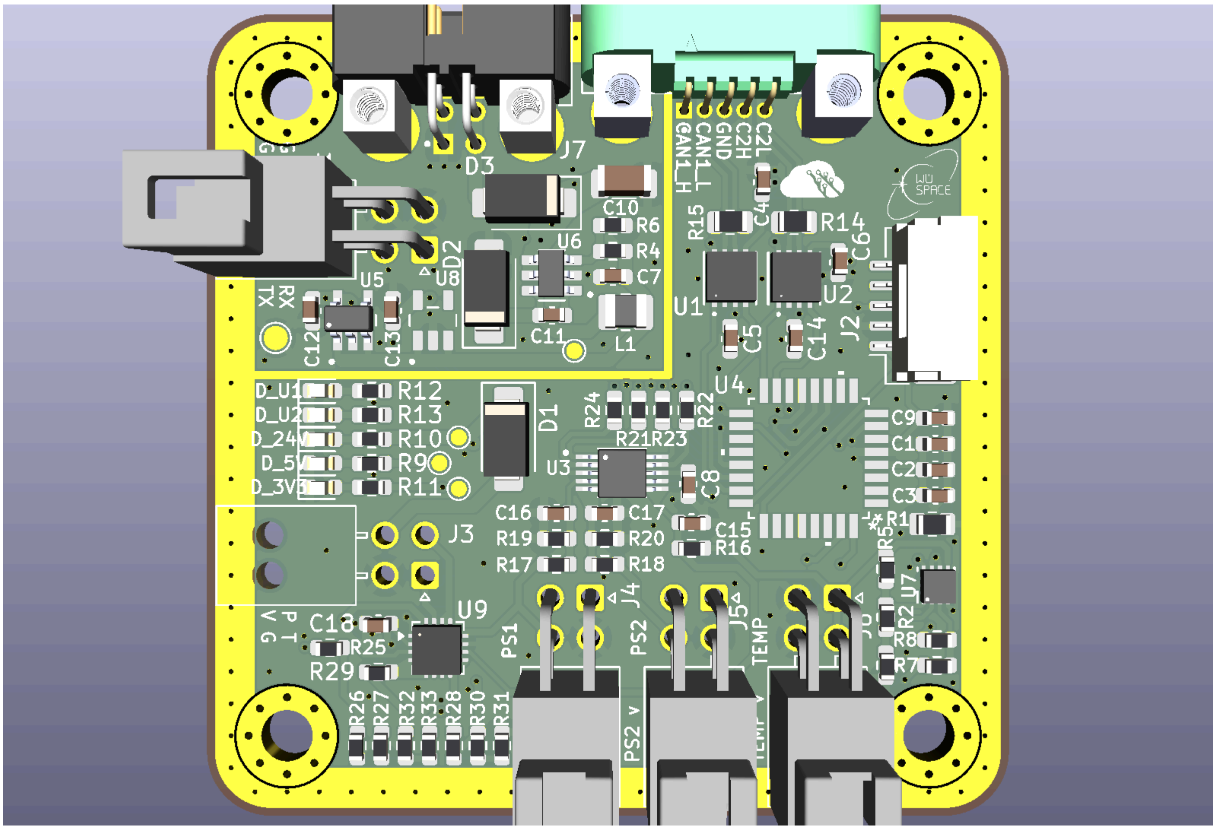

Figure 12: 3D view of the lower sensor board PCB

4. Ground segment

In order for the avionics system to serve it's purpose a robust and functioning ground support system is needed. The ground support system should be able to receive, store and display incoming data from several sources, be easily setup and moved and (similarly to the avionics system itself) as modular and extendable as possible. Furthermore it contains a tracker (T-Rex), tracking the rocket during flight in order to point a higher gain 5Ghz dish for the video transmission system.

4.1. Connectivity

Similarly to the air segment connecting the different ground subsystems to each other and to the air segment in a reliable fashion is crucial.

4.1.1. Interface to air segment

The primary goal of the ground segment is to receive, store and display data from the air segment. For that purpose different connections exist.

4.1.1.1. Umbilical wire

The umbilical wire is a seven strand shielded wire that directly connects power (12V and GND), data (4 wire Ethernet) and a separate launch detection wire. The umbilical wire directly connects to the umbilical board inside the rocket, and to the GSE provided by HyEnD outside of the rocket.

4.1.1.2. Radio Links

To receive the signals broadcasted by the air unit any number of receivers can be connected to the ground segment network.

4.1.1.2.1. Telemetry Link

For receiving the packets sent by the RocketLST air segment, the same board with different firmware can be used. In contrast to the air segments' RocketLST the ground segment LST board has no CAN trancievers, and are equipped with an ethernet port containing an integrated phy.

4.1.1.2.2. Video link

The RocketHD ground segment is equally distributed, the raw packets can be received on an arbitrary number of receivers, and will then be decoded and combined in one central location. That way there should be no packet loss in the actual video as long as all receivers together are receiving the minimal amount of fec packet for decoding.

4.1.2. Network

For communication between the different subsystems each subsystem is connected to the network via Ethernet. The network master is inside the main server in mission control

4.1.2.1. Internal network

For local communication at the launch site we are using an Ethernet network that spans from the Mission Control to the launch rail. All Clients in this Network are setup to work offline.

4.1.2.2. Public network

For providing Video streaming as well as for more redundancy for data storage we have a link to the public internet using Starlink. This is setup in a way that we only send video and telemetry from the network to a remote server. Controlling clients from the remote network is not possible.

4.2. Mission control

The primary goal of the mission control architecture is to connect all nodes of the GSE to one local area network and provide necessary services for communication and data retention. For communication between different nodes a publish/subscribe messaging server is used. All messages sent on the NATS are automatically logged into a timeseries database. Furthermore a video distribution server is setup to combine video streams from the video transmission system as well as potential launchsite cameras and drones. Via the network all nodes can access or send necessary data to ensure their respective functionality.

4.2.1. Main server

The main server runs the central components of our network. This includes routing, dns resolution, messaging, video distribution and the mission control gui server.

The main server is responsible for the primary central NATS messaging server.

4.2.2. Clients

In addition to the main server several hardcases can be connected to the system to provide additional specialized control hardware

4.2.2.1. TREX control

The T-Rex Control Hardcase provides the operator interface for controlling the tracking system. It is connected to the Main Server via Ethernet and includes a control stick, display and emergency shutdown. Through this interface, operators can switch between automatic and manually control of the tracker and monitor telemetry and video data in real time.

4.3. Tracking

4.3.1. Hardware

The primary objective of TREX is to receive the 5.8 GHz RocketHD video downlink during flight. To achieve this, the tracker actively points a 5.8 GHz dish antenna towards the rocket, which transmits the video signal through four onboard patch antennas. In order to maintain a stable live video transmission throughout the mission, the dish antenna must continuously follow the rocket with a pointing accuracy in the medium single-digit degree range.

As a secondary objective, TREX also points a Yagi antenna towards the rocket to receive flight telemetry data. This enables the ground segment to continuously receive, process, and monitor flight data during the mission.

The tracker is implemented as a two-axis system with independent horizontal and vertical movement. Both axes are driven by closed-loop NEMA 23 stepper motors. The horizontal axis uses a motor directly connected to a 20:1 gearbox, followed by a 3:1 belt drive, which rotates the tracker and dish antenna in azimuth. The vertical axis uses a NEMA 23 motor connected to a 10:1 gearbox, also followed by a 3:1 belt drive, to actuate the elevation movement.

The belt drives are used to minimize mechanical backlash while maintaining a dynamic and fast-moving system. To improve positioning behavior, the belts are tensioned both statically and dynamically. Final position feedback is provided by Hall effect sensors, allowing the tracker to achieve an approximate repeatable pointing accuracy of 2–3 degrees.

Figure 13: Picture of the TREX Tracker

4.3.2. Trajectory estimation

TREX is designed as a mostly self-contained subsystem. For target estimation, the system uses a basic ballistic prediction model as an initial reference trajectory. During flight, this prediction is corrected using state estimation telemetry received from the rocket. The tracker compensates for telemetry delay by predicting the rocket position over a short time horizon before calculating the required azimuth and elevation angles.

The STM32H7-based control board calculates the target pointing angles and commands the motor drivers for both axes. The control system includes timeout and fallback behavior so that the tracker can continue short-term prediction during brief telemetry interruptions and enter a safe state if valid tracking data is lost for too long.

To manually override tracking look at Section 4.2.2.1

5. Appendix

5.1. Appendix A: Requirements

5.1.1. Overarching

Requirement ID	Requirement Text
R0.0.01	System needs extensive testing in final configuration
R0.0.02	Components shall avoid export restrictions as much as possible
R0.0.03	the system architecture shall clearly classify all subsystems as mission-critical or non-mission-critical

5.1.2. Air Segment

5.1.2.1. Overarching

Requirement ID	Requirement Text
R1.0.01	The Avionic need to be easily transportable without special care
R1.0.02	The System shall be able to tolerate a failure of a complete subsystem without compromising the mission goal (Exception to this are EPS, Phoenix GPS and Pyro)
R1.0.03	The flight hardware shall be able to endure sustained acceleration of at least 10g
R1.0.04	Several Cameras shall record the flight

5.1.2.2. Rocket LST (mission critical)

Requirement ID	Requirement Text
R1.1.01	The Rocket LST shall transmit throughout the flight
R1.1.02	The system shall be designed to transmit data for at least 150km
R1.1.03	The system shall transmit all necessary data of the flight before radio communication is lost

5.1.2.3. Rocket HD (not mission critical)

Requirement ID	Requirement Text
R1.2.01	Rocket HD shall send a live video feed throughout the flight
R1.2.02	Rocket HD shall be able to send HD video with at least 60fps at 150km

5.1.2.4. EPS (mission critical)

Requirement ID	Requirement Text
R1.3.01	The Remove before flight shall be able to disconnect the batteries from all other loads
R1.3.02	EPS shall have redundant batteries
R1.3.03	Batteries shall be able to supply power during the flight for at least 3h power
~~R1.3.04~~	~~The EPS shall be able to top of the batteries on the launch rail~~
R1.3.05	The EPS shall monitor the Voltage and current of every power input(Batteries and External power on the rail)

5.1.2.5. Sensors (mission critical)

Requirement ID	Requirement Text
R1.4.01	The system shall reliably measure height and position
R1.4.02	Temperature and Pressure sensors in the lower part of the rocket shall be collected and distributed to all onboard systems that need this data (not mission critical)
R1.4.03	For the GPS position an Unlocked Phoenix Receivers shall be used
R1.4.04	At least one (if possible two) GPS receiver shall provide position and velocity data throughout the flight

5.1.2.6. Black Box (mission critical)

Requirement ID	Requirement Text
R1.5.01	Black boxes shall record all data on the rocket
R1.5.02	The black boxes shall be able to withstand hard impacts

5.1.2.7. Communication

Requirement ID	Requirement Text
R1.6.01	Communication between nosecone and bottom sensor board needs to use the two 4pin cables with fisher connectors
R1.6.02	Communications with the rocket on the rail need to happen via 7pin 2 fisher connector umbilical
R1.6.04	Telecommands shall be supported via umbilical connection and radio link

5.1.2.8. Pyro (mission critical)

Requirement ID	Requirement Text
R1.7.01	The Pyro system shall reliably deploy the drogue parachute at apogee
R1.7.02	The Pyro system shall not be able to deploy the parachute prior to apogee
R1.7.03	The Pyro system shall not be able to deploy the parachute on the ground

5.1.2.9. Telemetry

Requirement ID	Requirement Text
R1.8.01	Temperature and Housekeeping minimum frequency 1 Hz
R1.8.02	All other (sensor) data shall be at least 10Hz
R1.8.03	Continuously send the maximum achieved height and speed
R1.8.04	Tank pressure sensor shall supply tank pressure during fueling operation

5.1.3. Ground Segment

5.1.3.1. Overarching

Requirement ID	Requirement Text
R2.0.01	The ground segment shall be easily transportable
R2.0.02	The ground segment shall be able to be set up with minimal tools and without prior knowledge

5.1.3.2. T-Rex (not mission critical)

Requirement ID	Requirement Text
R2.1.01	T-Rex shall be able to track the rocket for the duration of the flight
R2.1.02	T-Rex shall support the weight and momentum of all antennas at any acceleration and velocity required
R2.1.03	T-Rex shall be able to rotate with a rate high enough to follow the rocket
R2.1.04	T-Rex shall be able to track the rocket within ± 90° azimuth and −20° to 90° elevation
R2.1.05!	The accuracy of the antenna pointing needs to be at most +-1deg - The resolution needs to be at least .25 deg
R2.1.06	T-Rex shall be able to use the telemetry data to reliably determine a tracking solution even if the telemetry data has data loss (extrapolation)

5.1.3.3. Mission Control (mission critical)

Requirement ID	Requirement Text
R2.2.01	Launch procedures need to exist that cover every possible scenario at launch site (error states, …)
R2.2.02	Mission control shall be able to reliably and easily send telecomands to the rocket during the launch preparation
R2.2.03	Mission control shall be able to easily verify all functionality of the system on the launch pad via umbilical and radio link
R2.2.04	The mission control shall display all relevant data in real time

5.1.3.4. Rocket HD Ground Segment (not mission critical)

Requirement ID	Requirement Text
R2.3.01	RHD GS shall record all received data
R2.3.02	RHD GS shall have multiple receiving chains which are able to receive the video stream
R2.3.03	One receive chain shall be able to cover the maximum distance

5.1.3.5. RocketLST Ground (mission critical)

Requirement ID	Requirement Text
R2.4.01	The RocketLST Ground shall be able to receive and decode the Telemetry data with at least 150km range
R2.4.02	RocketLST Ground shall be able to relay all received data to the other systems via Nats
R2.4.03	At least 2 RocketLST Ground receive chains shall be able to receive telemetry from at least 150km

5.1.3.6. Main Server

Requirement ID	Requirement Text
R2.5.01	The main server shall run a Nats server that provides the communication interface between all other ground subsystems

Requirement ID	Requirement Text
R2.5.02	The main Server shall run a Database service that collects all that is being send via Nats
~~R2.5.03~~	~~The main server shall display relevant data in real time~~ (changed to R2.2.04 in mission control)
R2.5.04	Needs to reliably supply the following TC to AIR on pad: • Ability to turn on/off individual systems • Parachute system arming/disarming/reset

5.1.3.7. Offsite Server

Requirement ID	Requirement Text
R2.6.01	The offsite Server shall make Video and NATS data publicly accessible
R2.6.02	The offsite Server shall not be able to send TC

5.1.3.8. Procedure Software

Requirement ID	Requirement Text
R2.7.01	The procedure software shall store procedure steps in a persistent database
R2.7.02	The software shall allow users to create, edit, and delete procedure steps in an intuitive interface
R2.7.03	The software shall display the full procedural flow and highlight the current active step
R2.7.04	The software shall support automated checks of telemetry data within procedure steps
R2.7.05	The software shall support branching procedures and allow switching between branches
R2.7.06	The software shall allow manual override and modification of the current procedure state
R2.7.07	The software shall support defining conditional logic for branching based on telemetry or user input
R2.7.08	The software shall log all changes to procedure steps and state transitions
R2.7.09	The software shall validate telemetry data against predefined thresholds or rules
R2.7.10	The software shall provide a visual representation of branching points and decision paths
R2.7.11	The software shall allow versioning of procedures and rollback to previous versions
R2.7.12	The software shall support user roles and permissions for editing and executing procedures
R2.7.14	The software shall allow pausing, resuming, and restarting procedure execution
R2.7.16	The software shall assign and store timestamps for each procedure step and state transition
R2.7.18	The software shall display elapsed and remaining time (T-time) for procedure steps where applicable

Requirement ID	**Requirement Text**
R2.7.19	The software shall allow filtering and navigation of the timeline based on timestamps and events