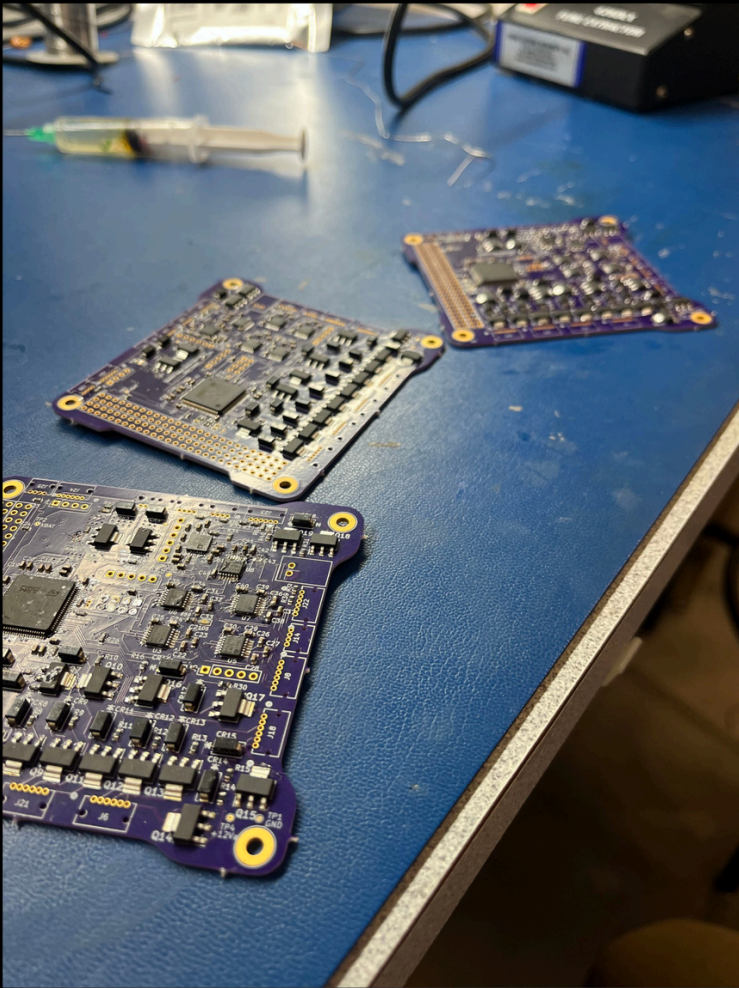


Kacper Paraniuk

ENGINEERING
PORTFOLIO



I am an engineer who loves to create systems and solve complex problems with an ultimate goal of enhancing the human experience. My focus is in avionics and building reliable embedded systems that keep flights safe, efficient, and connected.

2026
ELECTRICAL ENGINEERING
<https://thewiredjourney.com/>

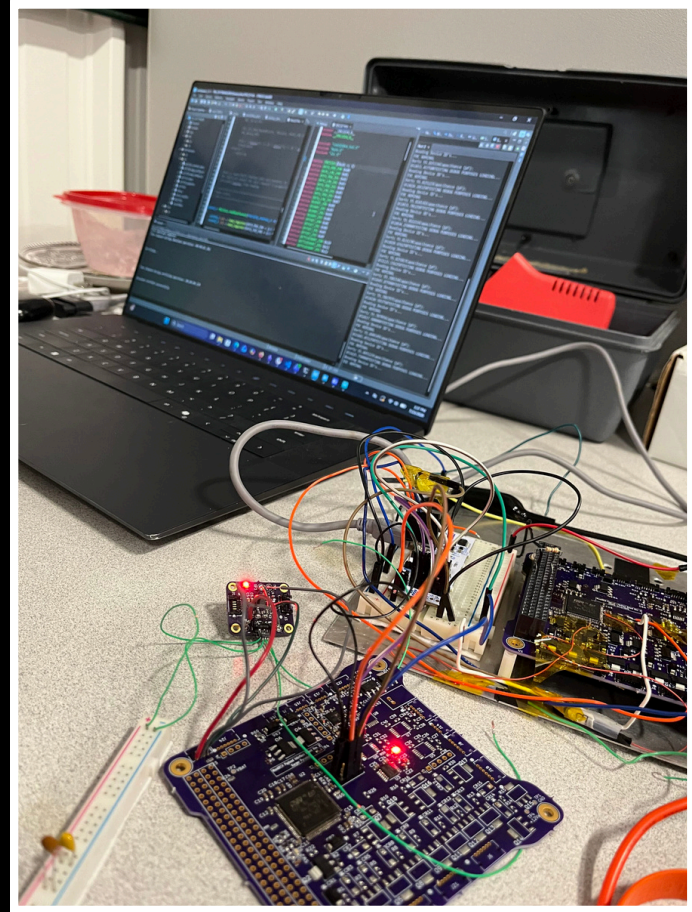
Payload Interface Board

2026 JANUARY - PRESENT

OVERVIEW

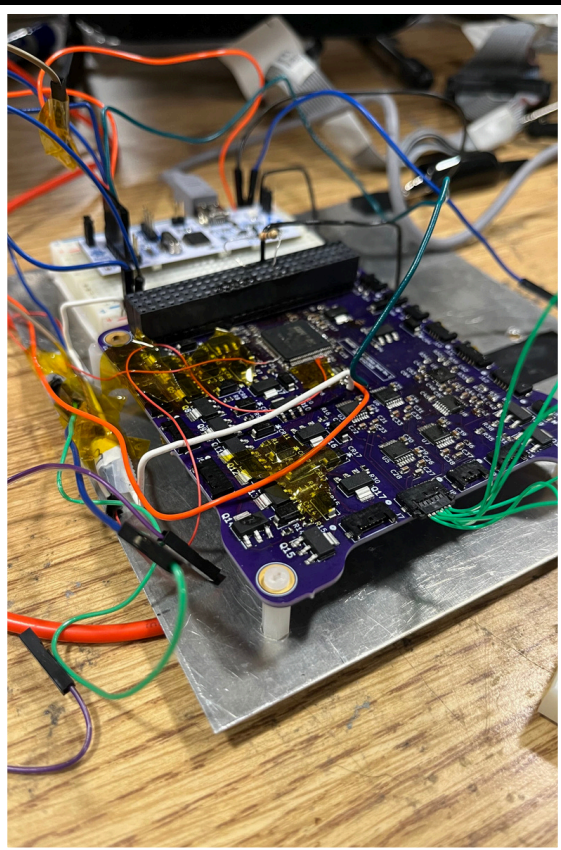
The payload interface board's (PIB) purpose is to control and instrumentation of the onboard payload of MonARCH. The PIB acts as the primary communication and control bridge between the spacecraft bus and the MonARCH payload.

The four layer board design consists of AD7124, FDC2214, MAX31856, and ADXL355 for reading from a variety of sensors such as K-type thermocouples, 4-20mA pressure transducers, capacitance sensors, and an accelerometer. The PIB monitors pressure, temperature, voltage, and fuel levels, and then provides data packets to the Onboard Flight Computer (OBC) through UART communication. The PIB directly controls chemical thruster firing via heat tape, drives solenoid valves for the propellant feed system, and commands the PPU. Additional functionality includes electrospray plume detection and payload current measurements. The PIB is a peripheral to the OBC and it waits for specific commands. In case of emergencies, the PIB is required to continuously listen for UART commands to be able to halt or execute a different task at any moment. Lastly, the system needs autonomous error handling such as in the event where the OBC doesn't catch a data outlier the system automatically moves into a safe state and communicates the error.



DEBUGGING/REDESIGN

Initial testing showcased many flaws in the first revision of the PIB. Testing began with the primary MCU (STM32L4A6) failing basic code execution, such as a simple GPIO LED toggle. Extensive continuity checks, pin probing, and datasheet analysis revealed errors in PCB design causing memory placement errors and constant board resets that prohibited code execution. These flaws were fixable with fragile "bodged" wire connections and thus testing resumed to validate IC drivers. However, these modifications severely hindered debugging, as loose and intermittent connections introduced noise and bit errors that frequently caused ST-LINK communication failures. After extensive hand-soldering and significant debugging time, the decision was made to develop a new PIB revision. Though the board ultimately succumbed to hardware issues, achieving brief solenoid control and successfully reading device IDs provided the critical baseline knowledge required for the next PIB revision. With all mistakes documented the hardware redesign was simple and quick. The new revision was built and software drivers were corrected. As a result, temperature, pressure, voltage, and capacitance measurements were all successfully read from sensors. Additionally, I implemented a command line interface for easy commanding. This enabled easy solenoid valve control and will be used for integrating all MonARCH components together, testing feed systems, and propulsion elements in fall 2026.



Payload Interface Board

2026 JANUARY - PRESENT

CHALLENGES MCU Selection -

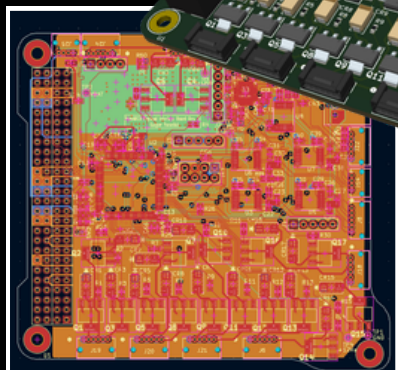
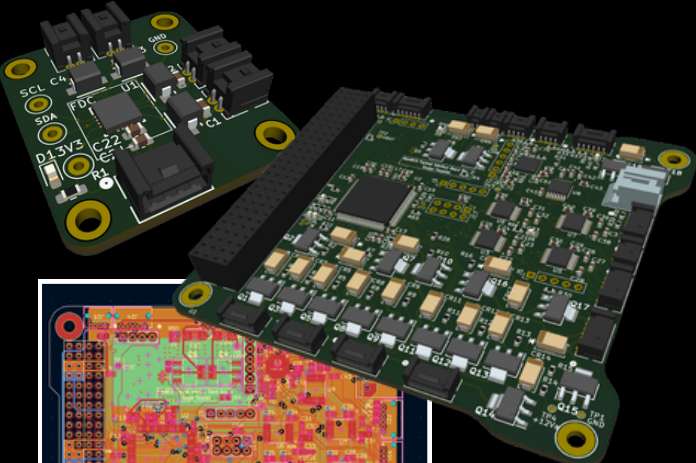
The entire 2026 spring semester I researched how the space environment affects electronics and realized that major issues to address, many of which made it difficult to subsequently make design choices. Such as picking out the microcontroller unit (MCU) which involved investigating the trade study between a COTS MCU and a rad-hard mcu. The micro-controller unit (MCU) is the most critical component of the payload interface system and is vulnerable to the space environment. A single event effect could cause mission failure if not taken into consideration during the design process, therefore, MCU selection required careful evaluation. The tradeoff between the risks posed by radiation exposure and the increased cost of radiation-hardened MCUs was the most important consideration. There are lots of commercial off the shelf (COTS) components available on the market that are not inherently designed for space flight nonetheless designed for radiation tolerance. Thus, the driving factors of selecting an MCU from the ST-Microelectronics (STM) series was the documentation behind these MCUs surviving and thriving in the LEO environment. Not only, STMs are very powerful, reliable, and accurate MCUs that we evaluated can perform every task the payload interface system needs to accomplish. Through rigorous research, the STM32L4A6 was chosen for the electrical payload system. The MCU consists of four I2Cs, three SPIs, three USARTs, and two UARTs which are essential communication protocols for interfacing with sensors, OBC, and PPU. The STM32L4A6 is an ultra-low-power processor and thus consumes exponentially less power than an STM32H7 series. The STM32L4A6 typically consumes an average of 20mA with a maximum of 150mA. It is a great processor when power consumption is a concern and perfectly satisfies the <1 Watt power PIB requirement. Also, it is important to note this MCU has an explicitly defined low power UART which is needed to communicate with the OBC. There are two 16-bit PWM timers which are needed for IEP valve pressure regulation. H7s consume around 264 μ A/MHz typical value at VDD = 3.3 V and 25 °C in Run mode (peripherals off) almost 3x the amount compared to the L4s which have around a power consumption of 91 μ A/MHz run mode (LDO mode). A more thorough explanation and validation of the STM32L4A6 and in-depth analysis of how components were chosen is available in the research paper [linked on my website](#). This Research Paper outlines the Space Environmental Effects on Electronic Components and details the development process of the payload interface board..

Developing Drivers for STM32 HAL

Every IC had a unique process for integrating its associated driver and bringing it to a functional state. This process differed significantly from high-level Arduino libraries I was accustomed to, which abstracted much of the underlying communication and data handling between ICs. For example, when communicating over I2C, the device address needed to be shifted left by one bit, with the least significant bit indicating a read or write operation. To develop reliable and functional drivers, I used preliminary skeletal drivers as references and adapted their structure to create my own custom drivers or implementations. For the FDC2214, I used an existing Arduino C++ library and converted its functionality into C-based functions compatible with the STM32 HAL. Additionally, I used device datasheets to develop error-detection functions, ensuring that each status and error bit was properly identified and interpreted.

REFLECTION

Working for the Electric Propulsion Lab on project MonARCH has been a revolutionary experience that I cannot be grateful enough for. From designing a PIB to helping build and operate electrosprays in the lab, my work has allowed me to apply and expand my knowledge across aerospace, computer engineering, physics, and beyond. I have gained hands-on experience with chemical safety while working with FAM142 propellant, as well as the safety protocols required when working with high and low pressures. Most importantly, MonARCH has strengthened my engineering problem solving skills by exposing me to new challenges on a daily basis and requiring me to diagnose, debug, and develop solutions with the team daily. Overall, my experience with MonARCH has deepened my understanding of aerospace systems while demonstrating my ability in debugging and problem solving issues in the field of embedded systems.



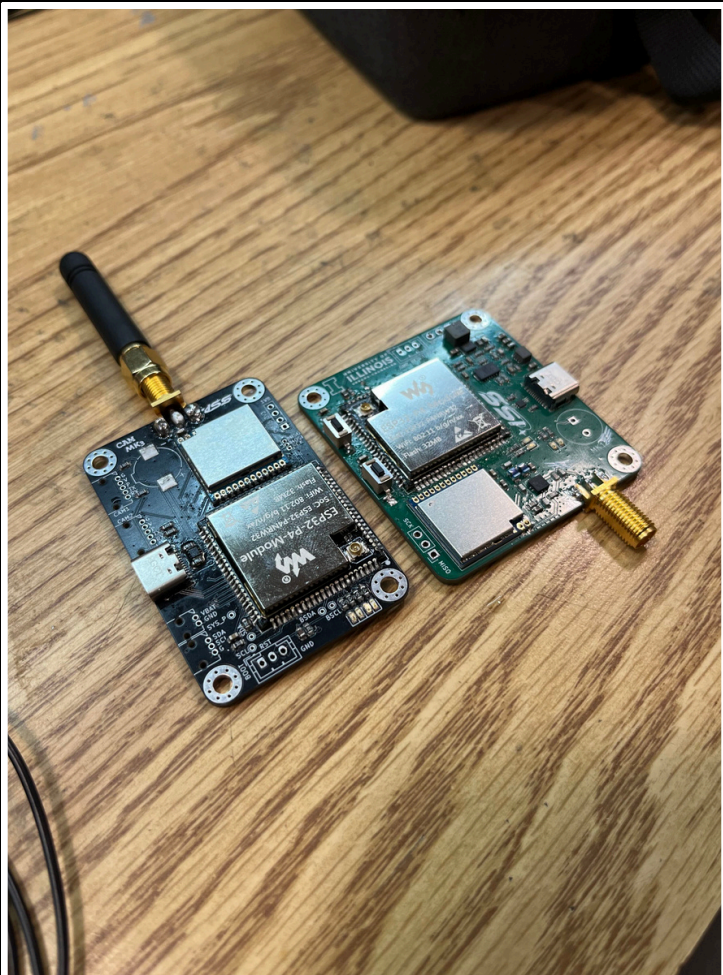
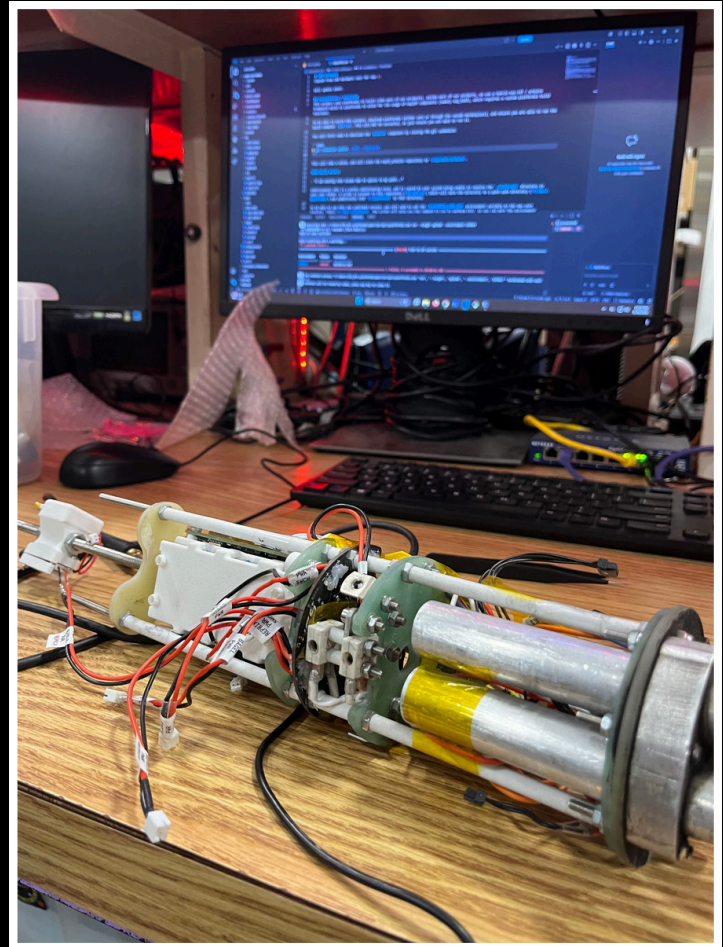
Live Video System

2025 OCTOBER - PRESENT

OVERVIEW

Project Spaceshot's main goal is to reach the Karman Line by building incredibly powerful and cool looking rockets. We, Illinois Space Society, push the limits of hobby rocketry to the next level. Containing custom flight electronics, our rockets push the limits in fields ranging from SDA, GNC, Fabrication, to E-Hardware. For example, I am honored to share my work on the live video system something that no other collegiate team is attempting (to my knowledge).

The live video systems mission is to capture the incredible nature and experience of launching rockets in real time and post launch. The live video system is a fully custom design that incorporates cheap COTS components. CAM MK3 or the small form factor custom transmitting PCB sits in the avionics bay and transmits live video over 434 MHz frequencies. CK3 is a peripheral to the main flight computer (COTS) and is commanded over the I2C protocol. EAGLE, the live video receiver, is connected to the ground station and receives data packets containing live frames which are then reconstructed in a python script. My primary role in the live video system project was designing and creating the live video receiver, EAGLE, while also contributing to the development of the transmitter and supporting software.



COMPONENT SELECTION & DESIGN

Heavy research was conducted at the beginning of the 2025 fall semester. Our goal was to create a custom digital video system learning from mistakes and hardships of other years. The TVP5151 video decoder was selected due to its ability to take in NTSC video formats and convert them to a digital YUV422 format. The ESP32P4 was chosen to be the MCU as it contained a H.264 hardware encoder, something that is incredibly rare for MCUs in this price range and form factor. Additionally the chip supports the LCD & CAM Controller which has the ability for fast polling of digital video within the DMA interface. The initial transceiver chosen was the Si4463 as it supported 1Mbps FSK, a speed no other transceiver could top at this form factor and price at the time. With an amplifier added to EAGLE we were confident in the system architecture. However, after testing, issues with EAGLE's amplifier, failing to write successful drivers to transmit at 1 Mbps, and finding the LR2021 a significantly better transceiver we decided to create new revisions of the boards. We decided to switch to the LR2021 due the reliability heritage of Semtech LoRa products. We have already used it on our main flight computer which has proven to provide reliable telemetry data over countless flights and we were confident working with similar products. On top of that, it supports FLRC at 2.6 Mbps as well as FSK at 1 Mbps at 434 Mhz frequencies and features such as ranging.

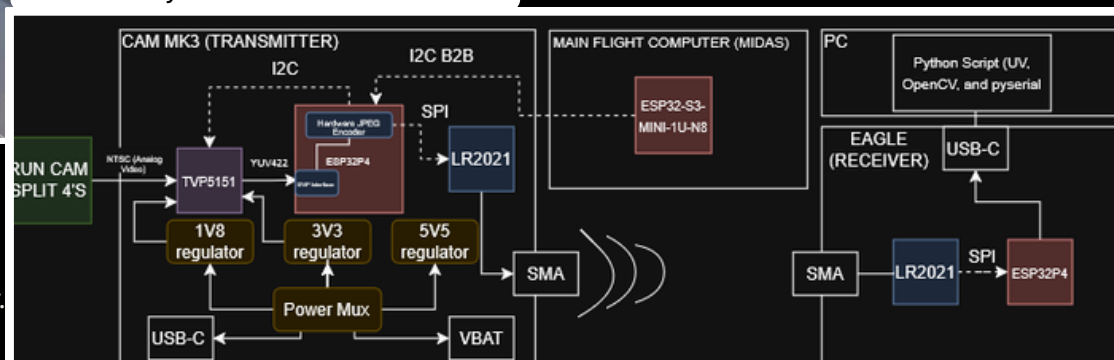
Live Video System

2025 OCTOBER - PRESENT

VIDEO PIPELINE

The video pipeline mission is to convert analog video coming from the Run Cam Split 4's into a digital format to receive and decode at a computer. The process begins at the Run Cam Split 4's which output a NTSC analog video format. This feeds into the TVP5151 video decoder chip which converts NTSC to YUV422, a digital video format. The ESP32P4 DVP interface copies outputted YUV422 into ESP32P4 memory which is then fed into a JPEG Hardware encoder. After encoding, data packets were formed and sent over SPI to the LR2021 to be transmitted over 434 MHz to EAGLE which receives the data through an LR2021 and feeds it to the ESP32P4 MCU chip. The MCU then prints data through a USB-C connector onto the serial monitor of a PC connected to EAGLE which is running a python script that continuously reads the serial monitor and reconstructs data into images which is then shown using OpenCV.

Live Video System Hardware Architecture



CHALLENGES

Developing a live video system in a small form factor with affordable COTS components is a greatly difficult endeavor. Managing big data sets within constrained space led us to scratch our heads every single meeting. However, there were three main challenges we encountered.

Compression of Video

Many attempts were made creating video compression software for our live video system as we overlooked the datasheet for the accepted ESP32P4 H.264 Hardware encoder data format. Our initial plan of using the hardware encoder failed as we attempted many solutions such as using the LCD & CAM Controller to convert YUV422 to YUV420 format and then feeding that data into the H.264 hardware encoder. However through implementation of the LCD controller and analysis of the data formats being outputted through python scripts and of the datasheet we determined that the order of the YUV format that is outputted from the LCD Controller is not in the correct order that the H.264 hardware encoder accepts. Making an attempt to flip the bytes unfortunately did not deem any success nor using the H.264 software encoder. After being stumped we came across a JPEG linux driver which we determined could work on the ESP32P4. We decided to try implementing the JPEG hardware encoder compromising speed for a functional system. This decision deemed the ESP32P4 unnecessary for our application. Although for a small compact system like this one H.264 encoding is the most practical, we ultimately decided on JPEG encoding as it worked for our application.

Developing drivers for JPEG, DVP, UVC, and TVP5151

Although the selected electronics were capable of being very powerful, limited documentation made developing custom software for the live video system particularly challenging. When creating the TVP5151 driver, I did extensive analysis of its datasheet and took inspiration from the general structure of C++ drivers. With few examples for the interfaces we needed to implement, our team went through multiple iterations of flashing and testing code to evaluate different features and verify their behavior. This iterative process allowed us to gradually understand how to create software for the hardware.

Avionics Bay Integration

After weeks of finalizing the live video system with the new LR2021, we had a reliable and functional product ready for testing at our high-power Summer Friends of Amateur Rocketry (FAR) launch. However, during rocket integration of the rocket and system testing, we encountered several unexpected issues involving the RunCam cameras and I2C buffer chip. Because we hadn't tested the system with other cameras, including the ones integrated into the avionics bays, we encountered compatibility issues that affected video output. The video we were getting was corrupted, however the recorded SD card video looked fine. We suspected the issues were related to camera wiring. We also found that certain pairs of cameras interfered with one another, with some combinations functioning correctly while others "corrupting" each other. Despite attempts to isolate the cause this we never figured out.

The avionics bay architecture also made troubleshooting difficult because physically swapping the Run Cams was challenging. Additionally, the architecture limited how tightly the SMA connectors could be secured, dropping the TX power output. Ultimately these issues prevented the live video system from operating during the 2026 FAR Launch.

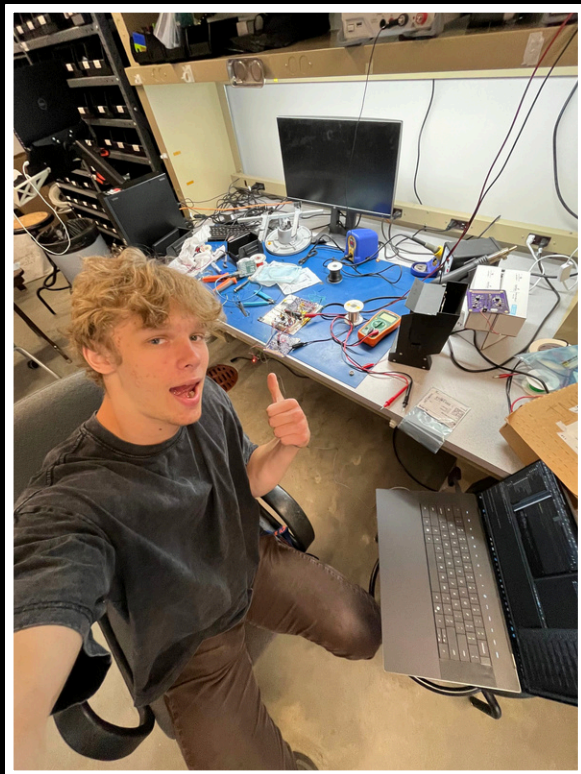
REFLECTION

Although a disappointing ending, the experience I gained from this project is invaluable. The FAR launch event highlighted the importance of testing hardware configuration under realistic integration conditions and designing avionics architectures to allow for accessibility. Surprisingly, the live video system project was my introduction to embedded systems. I will forever cherish the moments of hardship and success throughout the process of building this system as it created an incredible foundation introducing me to the entire world of hardware and software. From learning to read datasheets, video encoding schemes, communication protocols, to just creating a PCB from the ground up the experience was incredible.



Closing

Spending most of my free time tinkering with hands-on projects, I'm passionate and super excited to work on building, testing, and documenting avionics and electrical systems. I'm currently seeking opportunities in avionics and electrical system development and testing. I am always open to new experiences and creative challenges--feel free to explore my work or reach out directly!



KACPARANIUK@GMAIL.COM



[KACPERPARANIUK](https://www.linkedin.com/in/KACPERPARANIUK)



[KACPERPARANIUK](https://www.instagram.com/KACPERPARANIUK)



WWW.THEWIREDJOURNEY.COM