ELISA
Enabling Linux in Safety Applications

WORKSHOP

NASA Goddard

# Investigating the Implementation of Linux-based Payload Computers

Dongshik Won
TelePIX
& Korea Advanced Institute of Science and Technology
dw@telepix.net (dwon@kaist.ac.kr)
December 11th 2024

# Agenda

- Status quo of Satellite payload electronics
- Motivations for Linux-based Computers in Space
- Development Objectives
- Timeline
- Implementation Details: Hardware, Software, Interfaces
- Operation Principles
- Key Results
- Collaboration Opportunities
- Discussion Points
- Closing Remarks

# Status quo of Satellite payload electronics

- Historically, payload electronics relied on FPGA, limiting image processing capabilities.
- Raw data sent to ground stations often unusable due to cloud obstructions and transmission loss.
- Proprietary RTOS faced security vulnerabilities leading to costly patches in air-gapped environments.
- Rise of AI/ML in aerospace shifts developers towards Embedded Linux over proprietary or baremetal OS.

**Solution:** Modern high-performance Edge AI Computer

# Motivations: Linux-based Computers for Space

**SpaceX Adoption on Linux**

- Easy development for G-FOLD (Reentry guidance algorithm)
- x86 SOM for real-time guidance algorithms with PowerPC-based GNC Computer
- *The 3.2 kernel with real-time patches*
- Reference: How Embedded Linux is used in Spacecrafts!

# Motivations: Linux-based Computers for Space

**NASA White Paper Insights**

- White Paper: Challenges Using Linux as a Real-Time Operating System
- Transition Linux from RTOS
- Example: NASA Ingenuity, and many other missions
- Kernel optimization experiences is key

# Motivations: Linux-based Computers for Space

**Lessons from SmallSat Builders**

- Modern days SmallSat Compoment provider already shipped Linux based computers for Space
  - For Example: Nanoavionics (Lithuania), Endurosat (Bulgaria), Unibap (Ireland), etc…
- KAIST RANDEV: (Stock Raspberry Pi CM4 with Vanila Ubuntu were used, and have been working over 2 years!)

# Motivations: Linux-based Computers for Space

**New World, New Approach**

- Development cost considerations
- Transitioning from Desktop to Embedded systems

**Modern Software Infrastructure**

- CI/CD for Space (Docker, AI deployment)
- Examples: Klepsydra, Palantir MetaConstellations, SpaceCloud

# Development Objectives

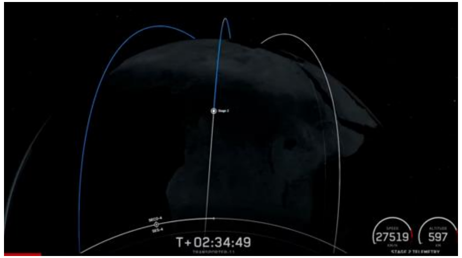
- Accelerate development by adopting innovative solutions. -> *just do it, and try*
- Reduce costs by prioritizing build-and-send over extensive space-qualified testing -> *Cost of Space Qualification > Cost of In-orbit Demonstration.*
- Implement rapid iteration: deploy, test, and refine quickly.
- Utilize COTS and AI boards with necessary space-grade modifications.
    - Microchip PolarFire SoC offers QML V equivalent, allowing minimal board design changes for space applications later, if needed.
    - NVIDIA Jetson platform offers no Space products, but it worth taking risks to test them in-orbit (A few reported case, but details weren't found): USC, OPS-SAT, etc. -> **Therefore, it maybe cheaper to find it out by ourselves**

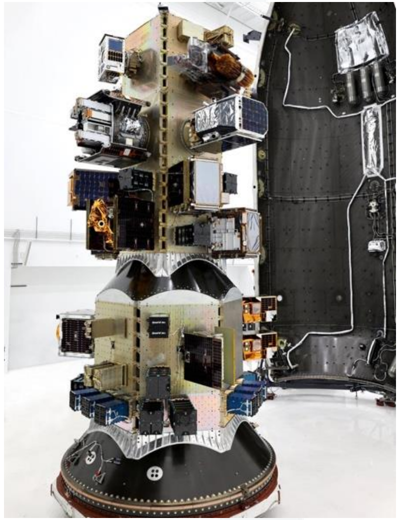
ELISA Enabling Linux in Safety Applications

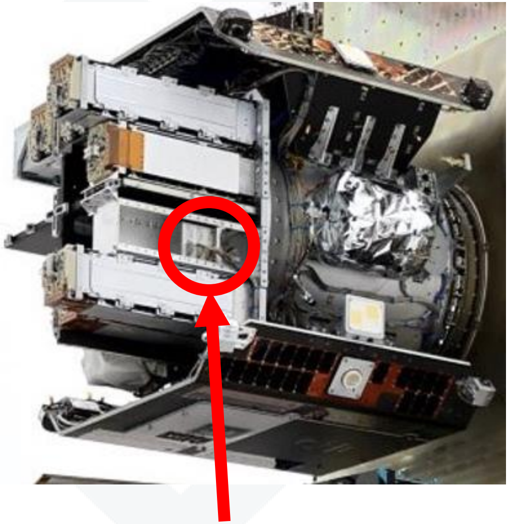
WORKSHOP

# Overview

- High-Performance AI Computer for Future Satellite Missions
- D-Orbit's ION Spacecraft (NORAD ID 60573) as a Host Payload
- Launch date: 08/16/2024 (SpaceX Transporter Rideshare 11)



Here!

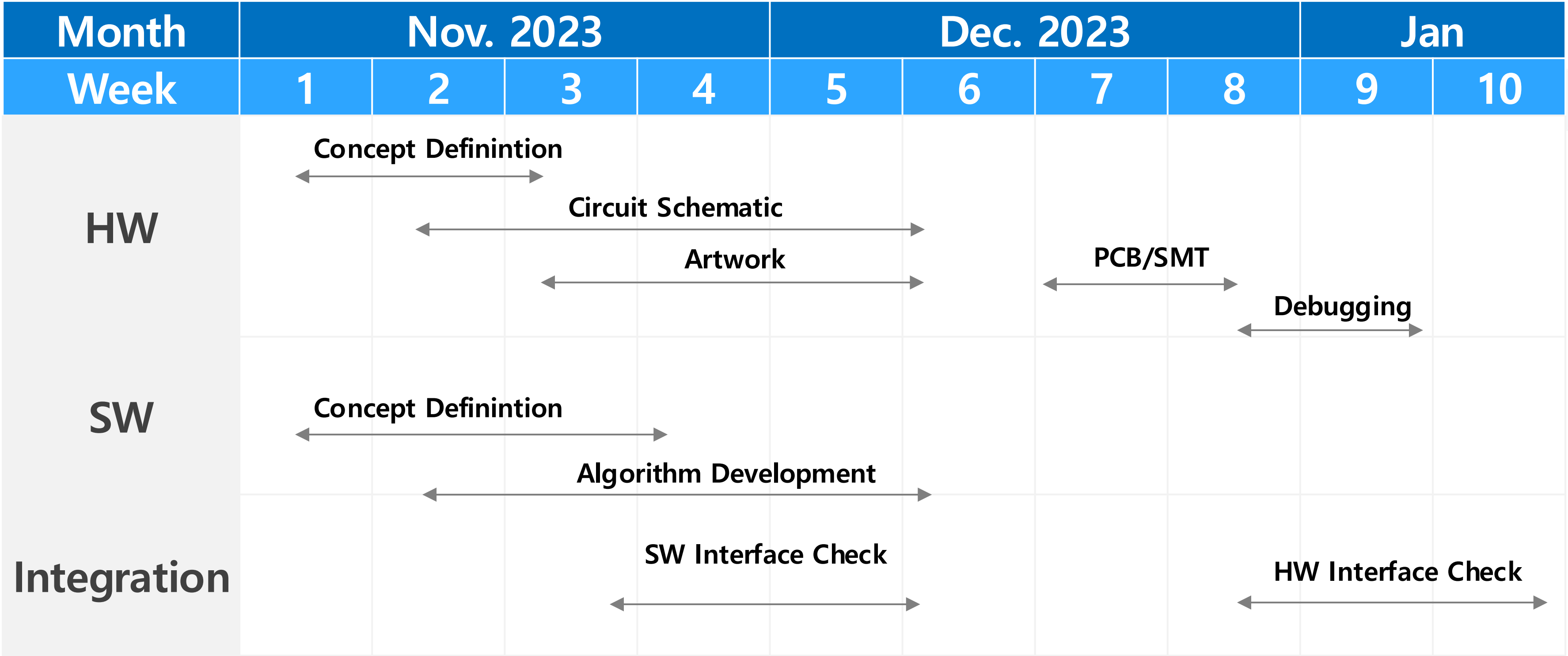
ELISA Enabling Linux in Safety Applications

WORKSHOP

**One shot, one opportunity!**

10 weeks: From Concept design to Satellite integrations.

# Timeline

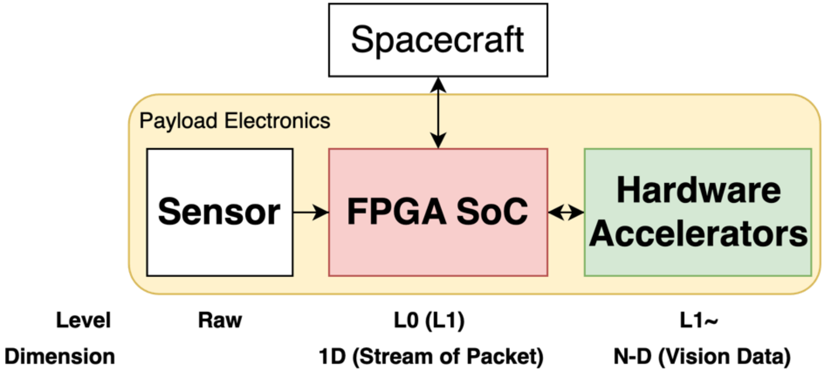



ELISA Enabling Linux in Safety Applications

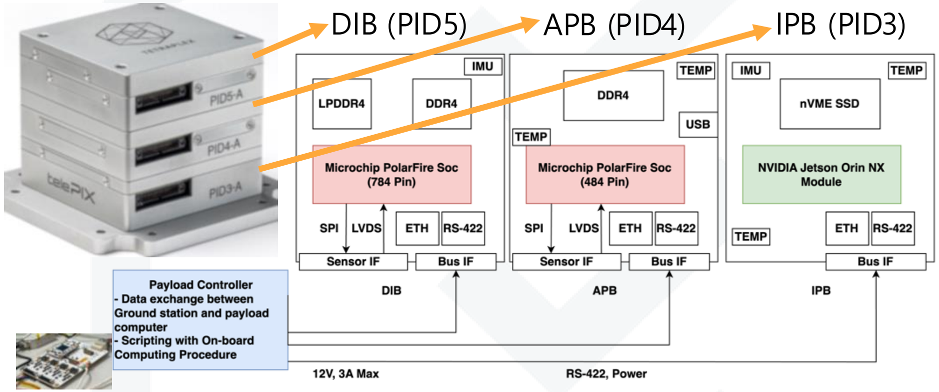
WORKSHOP

# Design and Implementation

**On-board Data Processing Architecture**




**Implementation for IOD**




- Mechanical: Simple, 1U within 160 mm x 160 mm base plate
- Electrical: Choose 1 from RS-422, RS-485, CAN

# Intelligent Processing Board

## Main Processor: Jetson Orin NX 16GB (SOM)

| Item | Spec |
|---|---|
| CPU | ARM Cortex-A78E v8.2 64-bit CPU |
| RAM | 16 GB LPDDR5 |
| GPU | 1024 NVIDIA® CUDA® cores \| 32 Tensor cores (MAX Operating Frequency 918 MHz) |
| DLA | 2x NVDLA \| Maximum Operating Frequency: 614 MHz \| 20 TOPS each (Sparse INT8) |
| Networking | 10/100/1000 BASE-T Ethernet \| Media Access Controller (MAC) |
| Peripherals | xHCI host controller with integrated PHY (up to) 3x USB 3.2, 3x USB 2.0 \| 3x1 (or 1x2 + 1x1) + 1x4 (GEN4) PCIe \| 3x UART \| 2x SPI \| 4x I2C \| 1x CAN \| DMIC \| DSPK \| 2x I2S \| 15x GPIOs |
| Storage | Supports External Storage (NVMe) via x2 or x4 PCIe |
| Temperature | Temp. Range (TJ)*: -25°C – 105°C \| Maximum Orin SoC Operating Temperature = Slowdown Temp = 99°C |
| Power | 12V, Max 25W |


ELISA Enabling Linux in Safety Applications WORKSHOP

# Application Processor Board

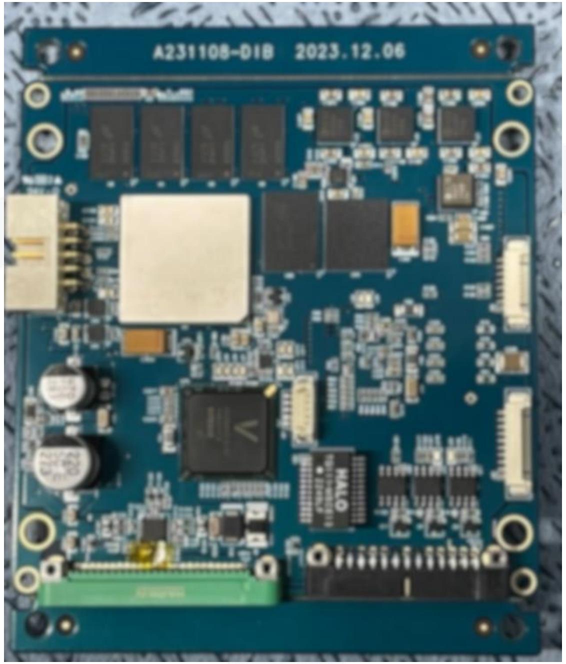
## Main Processor: Microchip PolarFire FPGA SoC (250T)

| Item | Spec |
|---|---|
| CPU | 1x 64-bit RV64IMAC monitor/boot core<br>4x 64-bit RV64GC Application cores<br>Fmax of 667 MHz (–40 °C to 100 °C Tj),<br>3.125 CoreMarks/MHz, 1.714 DMIPS/MHz **MPFS250T-FCVG484** |
| OS | Microchip Linux (Buildroot) |
| RAM | 16 Gbit LPDDR4 (512M x 32) |
| FPGA | 254K logic elements (4-input LUT + DFF)<br>784 Math blocks (18x18 MACC)<br>16 SerDes lanes of 12.7 Gbps |
| Peripherals | 2x GigE MACs, USB 2.0 OTG, 5x multi-mode UARTs, 2x SPI, 2x I2C, 2x CAN 2.0 Controllers.<br>2x PCIe Gen2 End Points/Root Ports **(for IPB Interface)** |
| Storage | MMC 5.1 SD/SDIO **(SD Card, eMMC 8GB)**<br>1 Quad SPI flash controller **(1 Gb, Serial NOR Flash)**<br>128 KB eNVM<br>56KB sNVM |
| Temperature | –40 °C to 100 °C Tj |
| Power | 12V, Max 7W |


# Data Interface Board

## Main Processor: Microchip PolarFire FPGA SoC (250T)

| Item | Spec |
|---|---|
| CPU | 1x 64-bit RV64IMAC monitor/boot core<br>4x 64-bit RV64GC Application cores<br>Fmax of 667 MHz (–40 °C to 100 °C Tj),<br>3.125 CoreMarks/MHz, 1.714 DMIPS/MHz (**MPFS250-FCVG784E**) |
| OS | Baremetal (C/C++) |
| RAM | 16 Gbit LPDDR4 (512M x 32) |
| FPGA | 254K logic elements (4-input LUT + DFF)<br>784 Math blocks (18x18 MACC)<br>16 SerDes lanes of 12.7 Gbps |
| Peripherals | 2x GigE MACs, USB 2.0 OTG, 5x multi-mode UARTs, 2x SPI, 2x I2C, 2x CAN 2.0 Controllers.<br>2x PCIe Gen2 End Points/Root Ports |
| Storage | MMC 5.1 SD/SDIO **(SD Card, eMMC 8GB)**<br>1 Quad SPI flash controller **(1 Gb, Serial NOR Flash)**<br>128 KB eNVM<br>56KB sNVM |
| Temperature | 0 °C to 100 °C Tj |
| Power | 12V, Max 7W |





ELISA Enabling Linux in Safety Applications

WORKSHOP

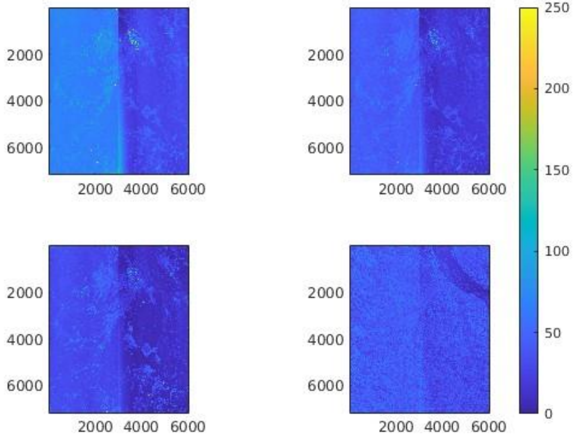
# SW Test Overview

On-board Image Data Pre-processing (CAS500 RAW -> L0)

- Fix Gain and perform correction to flatten images of TDI images

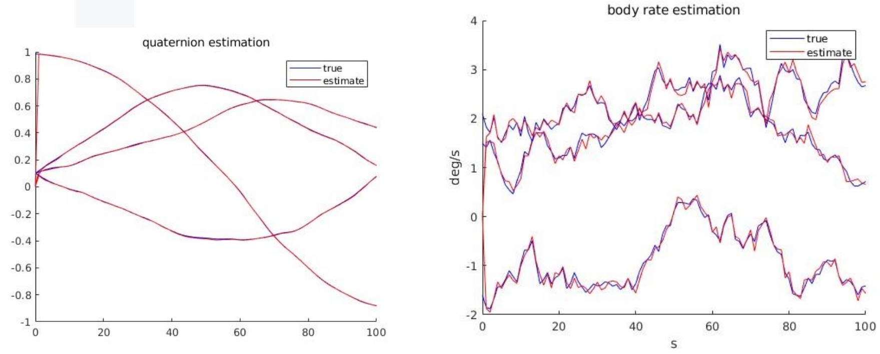
Spacecraft attitude estimation computation testing

- Quaternion and body rate estimation algorithms with EKF




On-board (C++)

Ref (MATLAB)

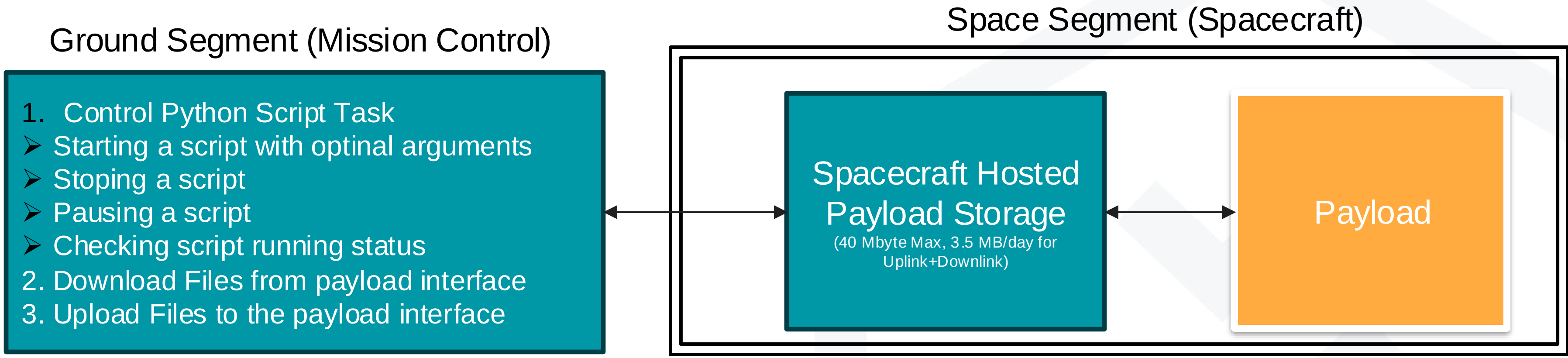


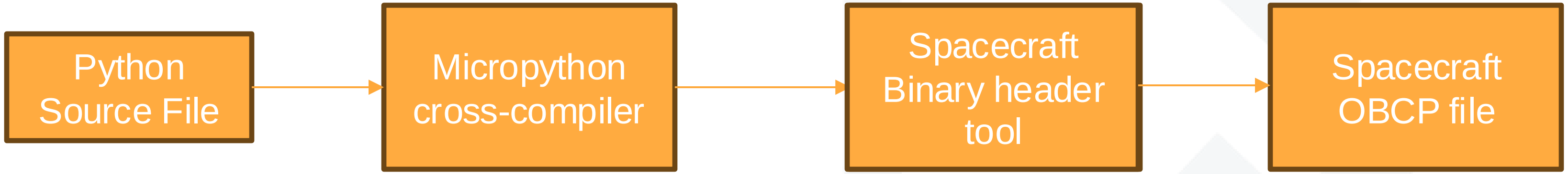

Spacecraft Attitude Estimation Computing Demo

ELISA Enabling Linux in Safety Applications

WORKSHOP

# Operation Principles



# Key Results

## Ground Tests

| Test Type | Date | Results |
|---|---|---|
| Thermal Cycling (Unit Level) | 2024-01-11 | Passed |
| Electrical Test (System Level with Bus EM) | 2024-01-15 | Passed |
| Electrical Test (System Level with Bus FM) | 2024-01-30 | Passed |
| Static Load (System Level with Bus FM) | 2024-02-12--2024-02-18 | Passed |
| Random Vibration (System Level with Bus F M) | 2024-02-12--2024-02-18 | Passed |
| Therval Vacuumn (System Level with Bus FM) | 2024-03-20 | Passed |
| Launch Campaign | 2024-05-3--2024-06-04 | Passed |

## In-Orbit Demonstration Results

| Test ID | Date | Reps | OBCP | Results |
|---|---|---|---|---|
| Test-000 | 2024-08-21 -- 2024-08-22 | 2 | v1 | Passed (PID4) |
| Test-001 | 2024-09-02 | 3 | v1 | Passed (PID5, once) |
| Test-002 | 2024-09-03 | 3 | v1 | Passed |
| Test-003 | 2024-09-24 | 1 | v2 (Added resets logic ) | Passed |
| Test-004 | 2024-09-30 | 1 | v3 (OBCP buffer flush) | Passed |
| Test-005 | 2024-10-11 | 1 | v4 (Wait for boot) | Passed |
| Test-006 | 2024-10-15 -- 2024-10-20 | 24 | v5 | Passed |
| Test-007 | 2024-10-22 -- 2024-10-27 | 32 | v5 | Passed |
| Test-008 | 2024-10-29 -- 2024-11-09 | 32 | v5 | Passed |
| Test-009 | 2024-11-12 – 2024-11-18 | 32 | v5 | Passed |
| Test-010 | 2024-11-18 – 2024-11-22 | 31 | v5 | Scheduled |
| Test-011 | 2024-12-?? | ?? | v5 | Planned (Extension) |

# Lessons Learned: Challenges and Mitigations

- **Vague Requirements:** Fixed deadlines with flexible implementations
  - Additional AI Model testing with replica model on ground.
- **Design Limitations:**
  - I2C line issues: IMU, Temp sensor not working -> Fixed in the next revision
  - PCIe remain untested: Boards split in half -> Test on-going with replica model
  - A few signal nets misplaced: eMMC didn’t work (SD Card work around) -> 1 out 5 chances, board requires resets.
- **Limited Information on Spacecraft**
  - LEOP phase data was not received: Payload requires recvbuf flush before receiving new data -> Undocumented and passed to us after the launch -> eventually fixed.

# Open Questions for Future Collaborations

- Importance of on-device AI for Space
- Defining on-device AI for Space
- Implementation strategies

# We are looking for collaborators

- Kernel optimization: Memory, boot times, recovery
- OTA updates (FW, FPGA bitstreams)
- Formal verification for SW and beyond (GenAI driven SW)
- Real-time response for space conditions (COTS based quantitative measurements)
- Lightweight containers or VMs
- AI integration alternatives

# Discussion Points

- Concepts for Intelligent payloads: IMU inclusion to payload?
- On-board mission planning (e.g. D-SHIELD, DSA, Starling, etc)
- Porting Zephyr to NVIDIA’s AOMMC (from FreeRTOS)
- RTOS/Linux for mixed critical applications
- (Forward looking) Software testing and formal verification: How do we formally verify software that has Generative AI component?
- ClangBuiltLinux Consideration?
- “mini” (RF+OBC+ADCS+Payload Computer) Flatsat for SGL RnD
- Compiler maturization for RISC-V Architecture

ELISA Enabling Linux in Safety Applications WORKSHOP

# Closing Remarks

For further questions, please contact Dongshik Won

- dw@telepix.net
- dwon@kaist.ac.kr

# Licensing of Workshop Results

All work created during the workshop is licensed under Creative Commons Attribution 4.0 International (CC-BY-4.0) [https://creativecommons.org/licenses/by/4.0/] by default, or under another suitable open-source license, e.g., GPL-2.0 for kernel code contributions.

You are free to:

- Share — copy and redistribute the material in any medium or format
- Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.