Przepis na zdjęcie Ziemi z orbity

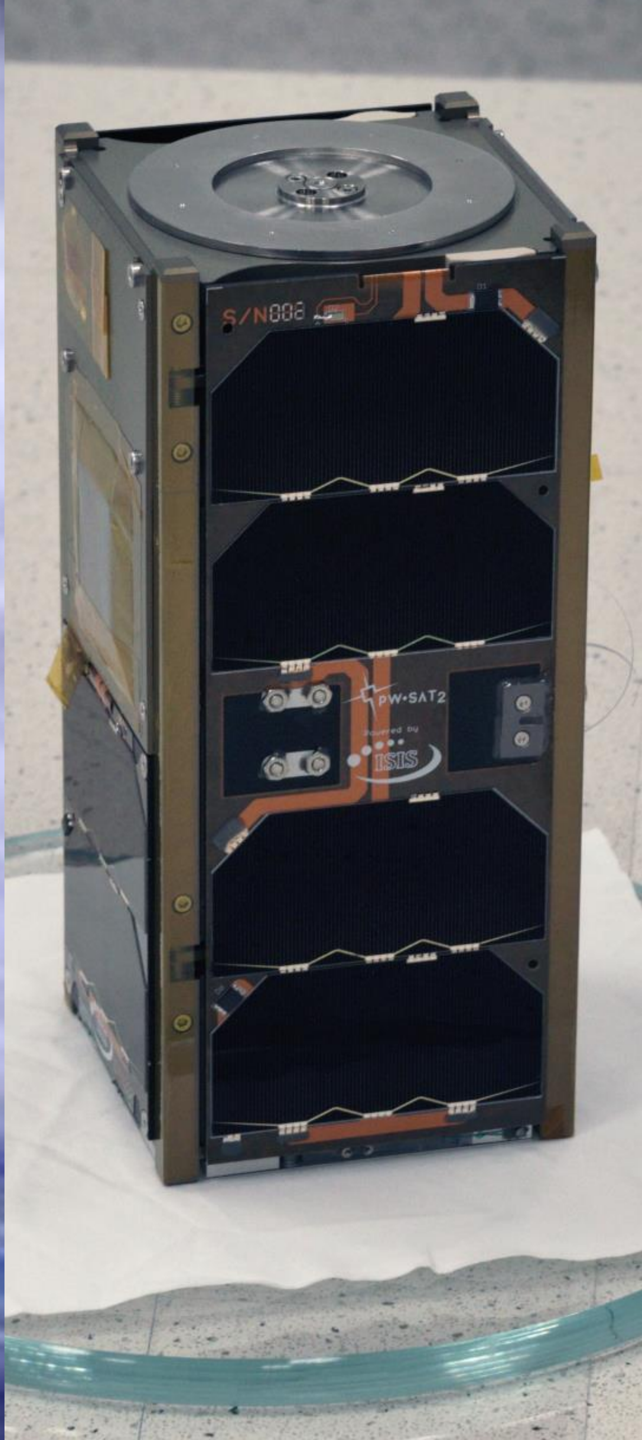
Marcin Drobik
Head of Flight Software Engineering @ KP Labs

2024-02-06

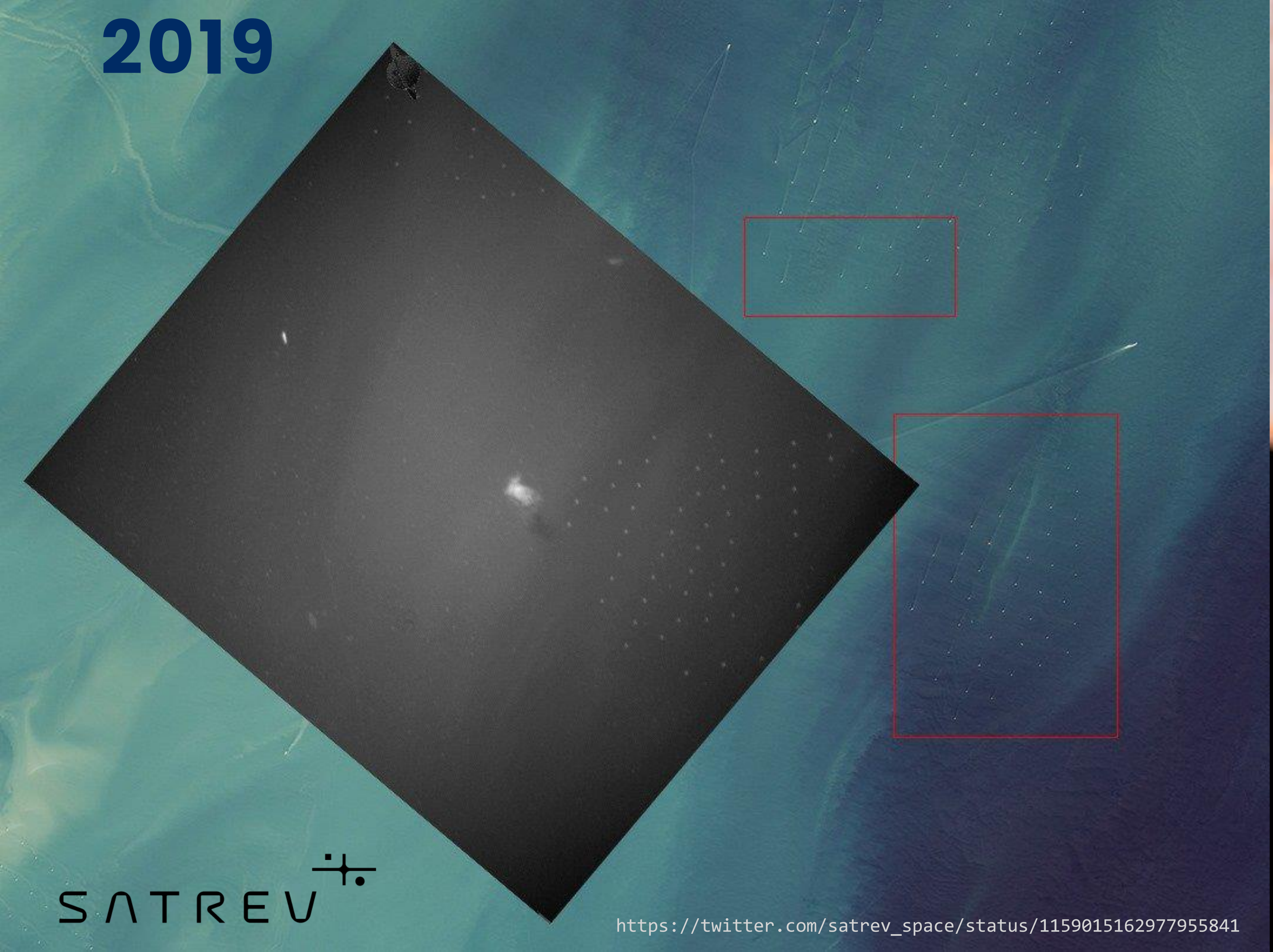
info@kplabs.pl
www.kplabs.pl

1978

2018



2019



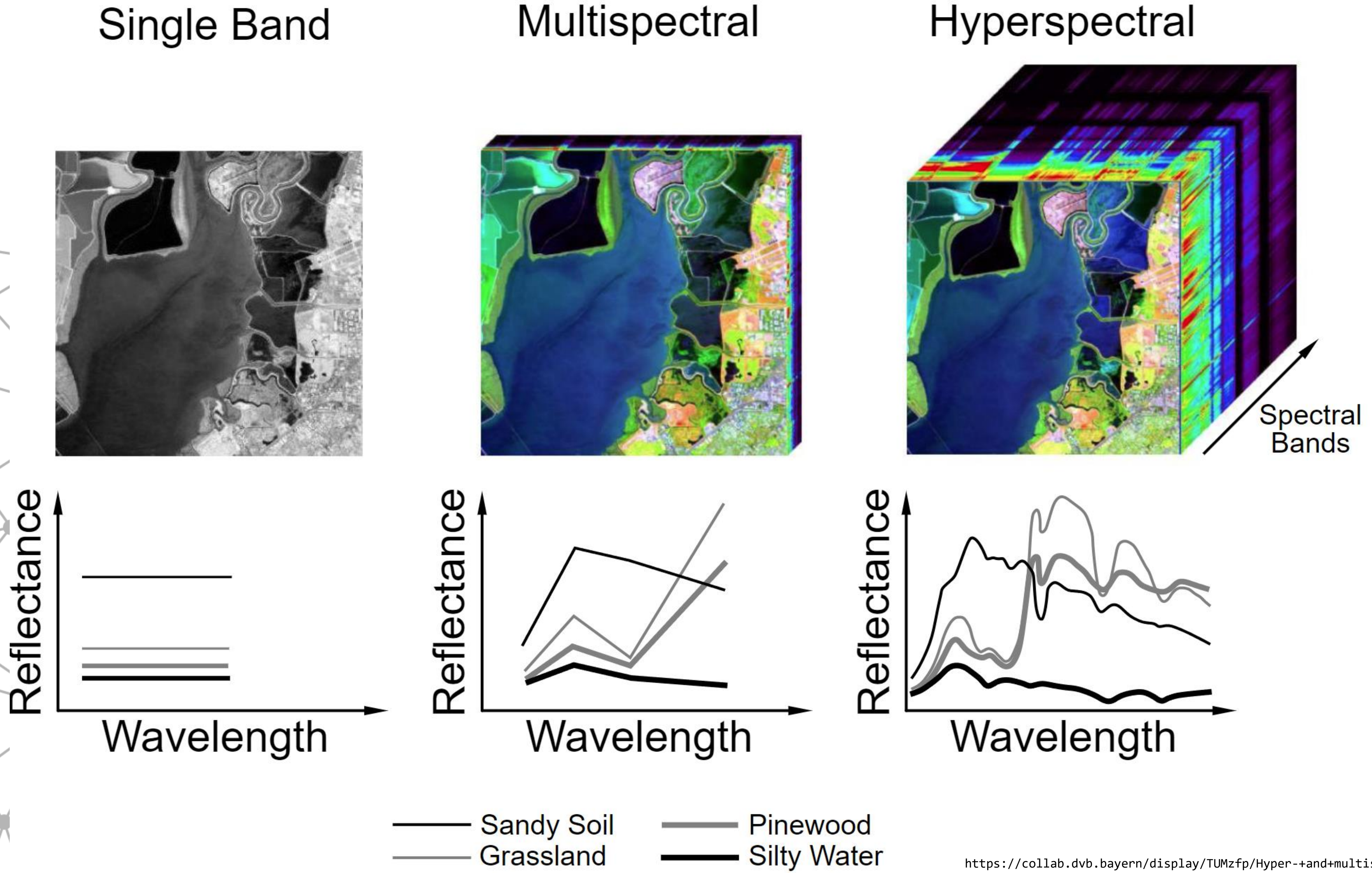
SATREV

https://twitter.com/satrev_space/status/1159015162977955841

2023

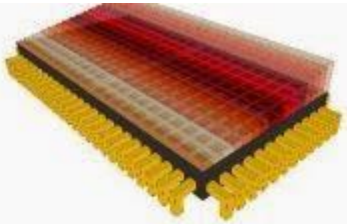
https://scanway.space/star-vibe-images/

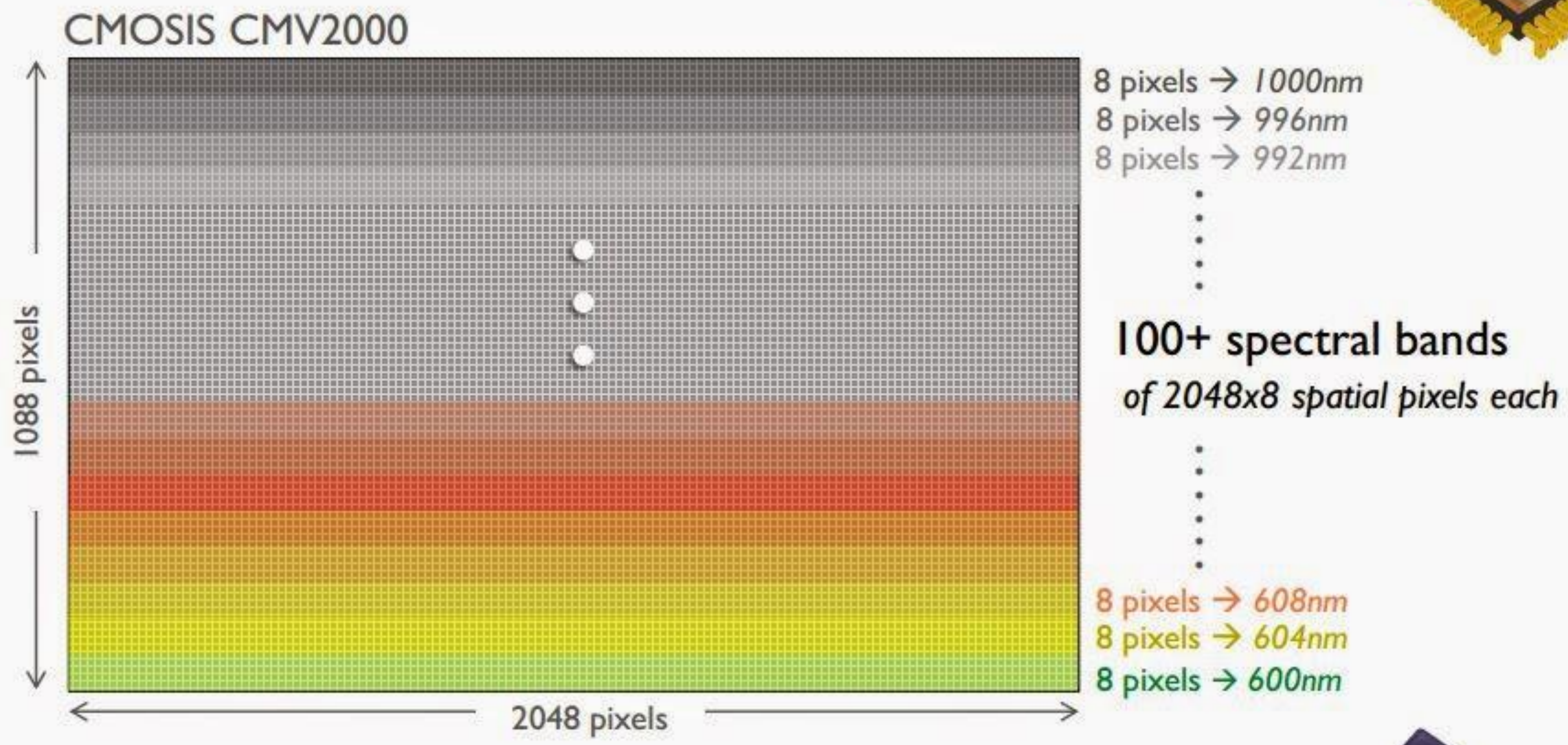
Po co kolejne zdjęcie?



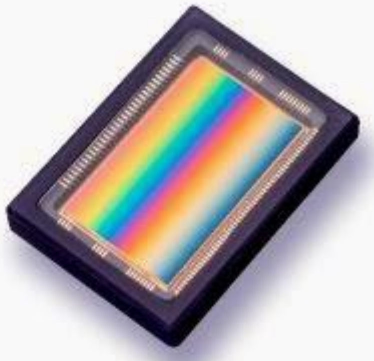
https://collab.dvb.bayern/display/TUMzfp/Hyper-+and+multispectral+imaging

LINE-SCAN **HSI SENSOR DESIGN** (GEN 1)





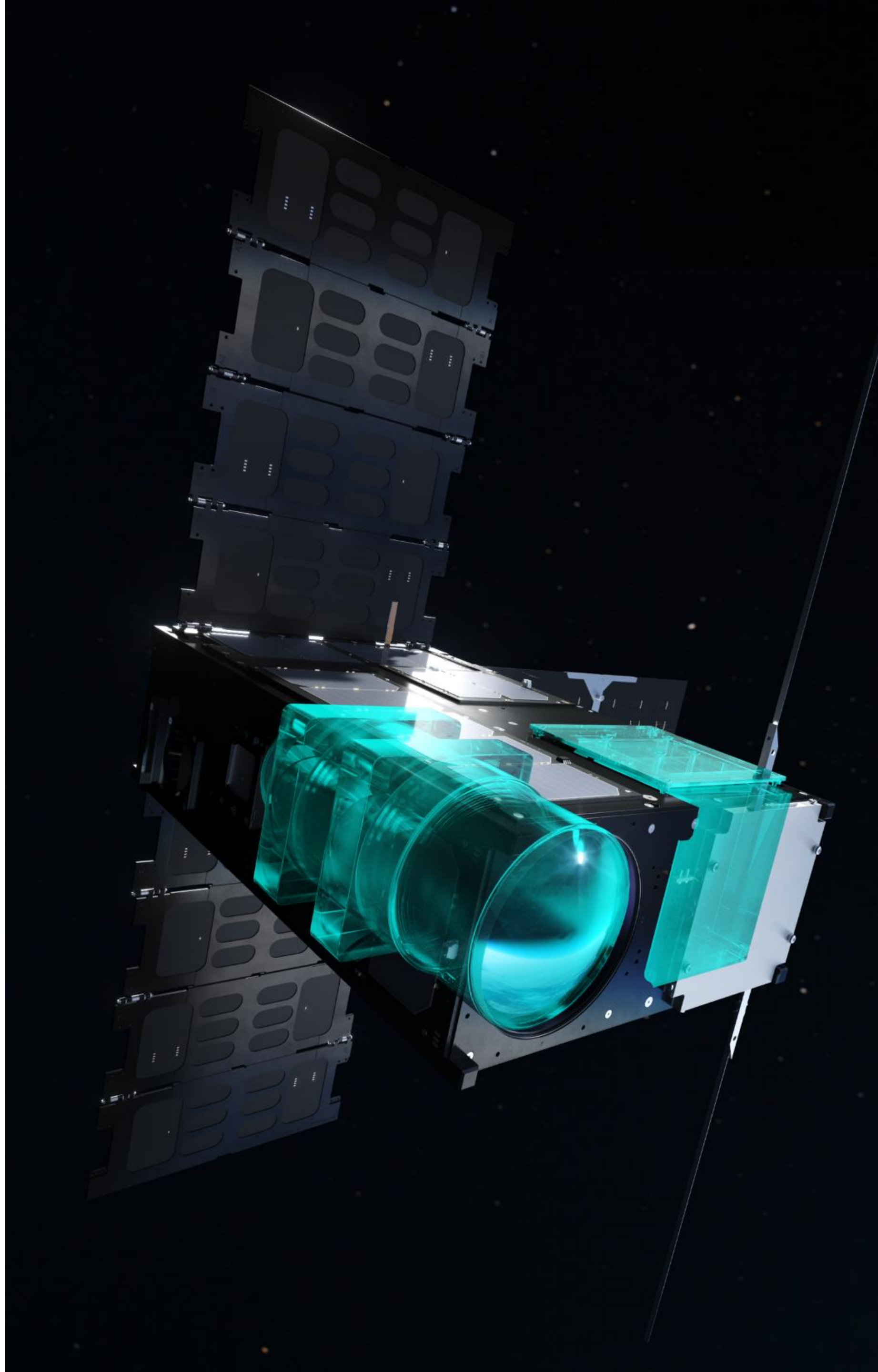
- Key specifications
  - **Spectral resolution**: 128 bands in 600-1000nm
  - **FWHM**: ~ 10-12nm
  - **Spatial resolution**: 2048 pixels x length of scan
  - **Speed**: up to 340 fps (full sensor frame)



Rozmiar danych

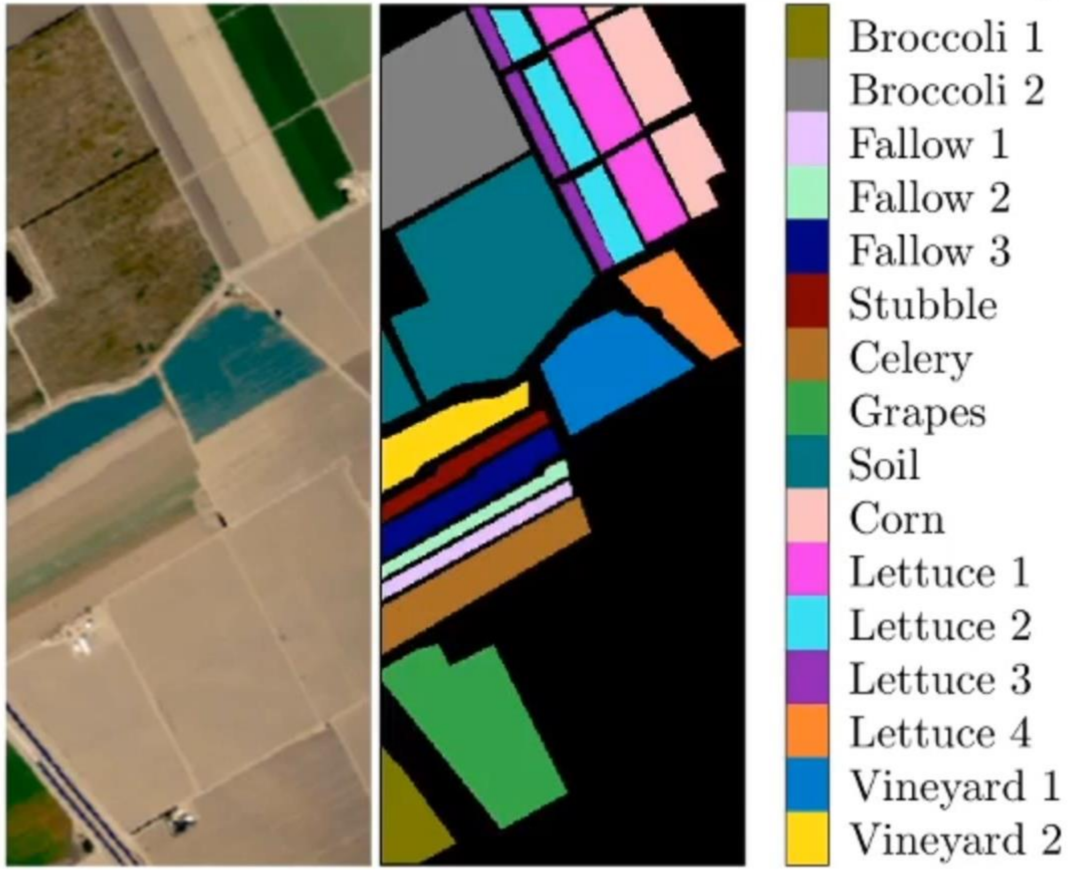
Minimalne zdjęcie hiperspektralne

- 2048*1088 x 10 bit x 190 pasm (min x 2)
- ~ 800 MB danych po kompresji (16bit PNG)
- Pełna kostka hiperspektralna wymaga okolo **1.5 GB** surowych danych

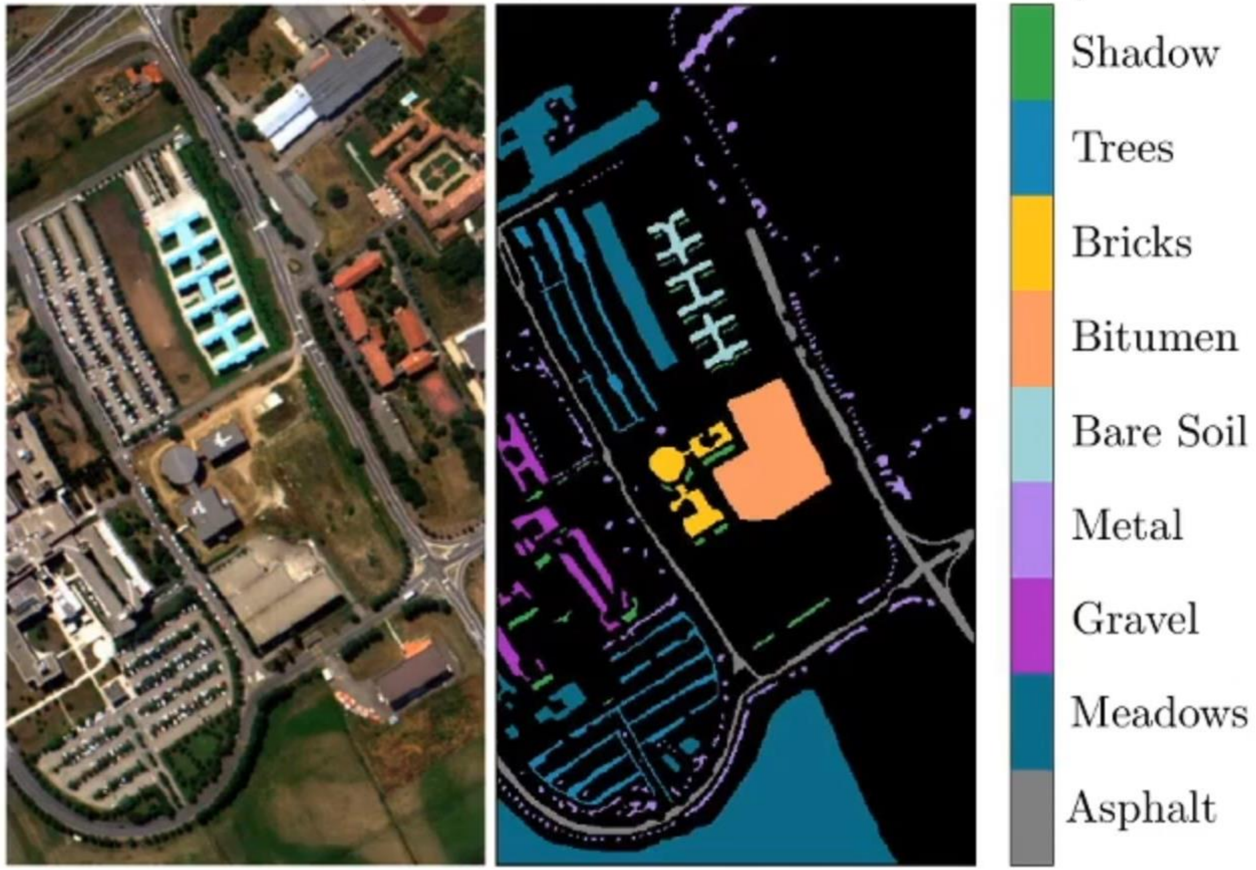


Benchmark hyperspectral images—examples

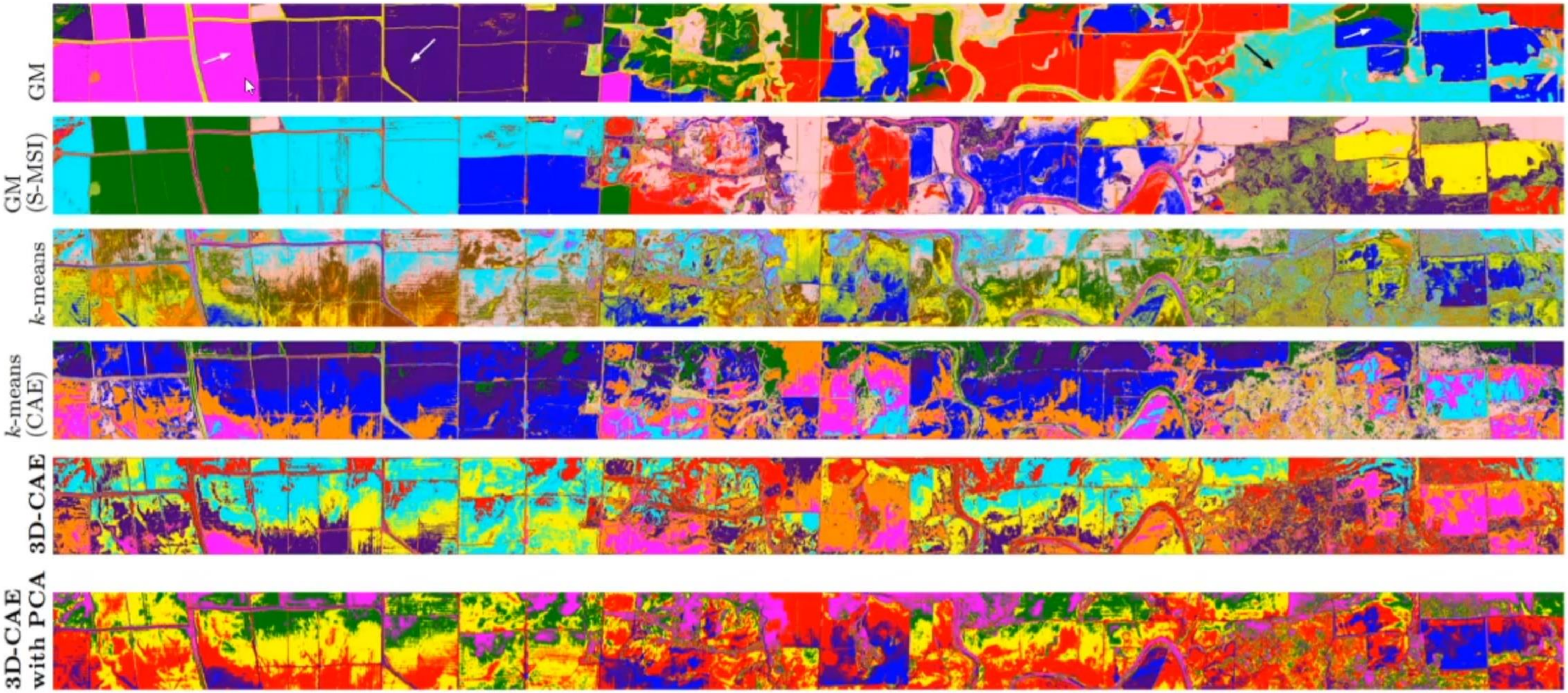
Salinas Valley (512x217px, AVIRIS, 3.7m, 224 bands)



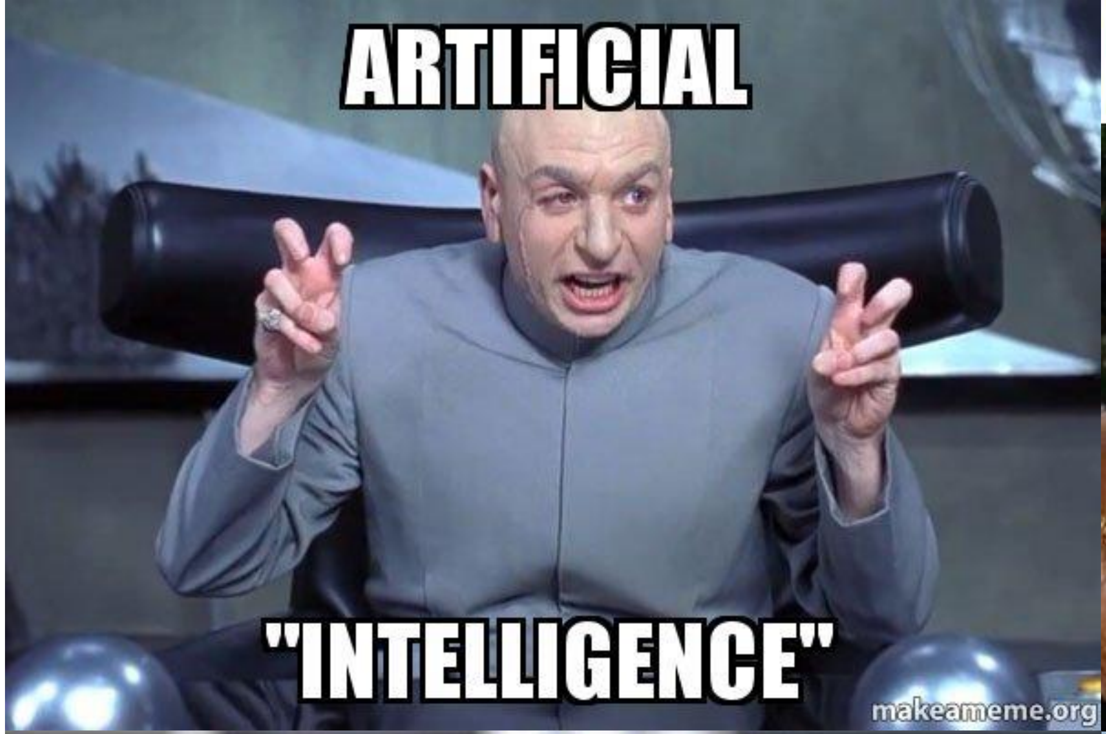
Pavia University (610x340px, ROSIS, 1.3m, 103 bands)

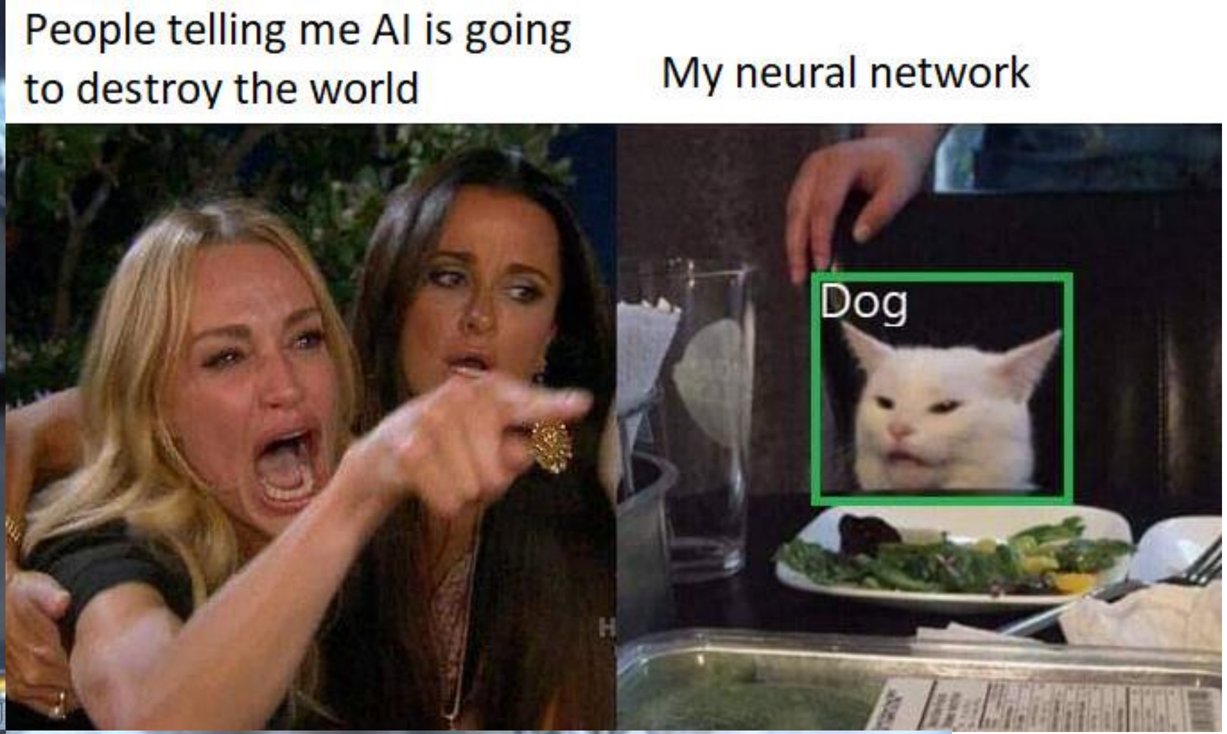


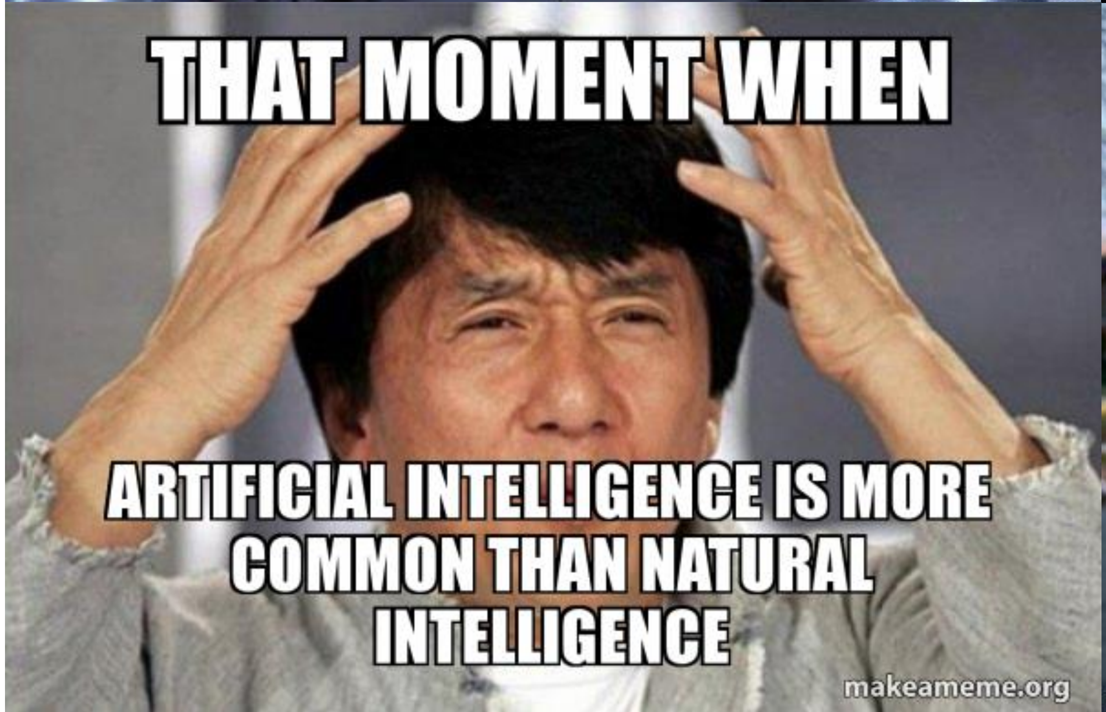
Example 2: Precision agriculture (Mullewa, Australia)



J. Nalepa et al.: Unsupervised Segmentation of HSI Using 3-D CAEs, IEEE GRSL, pp 1–5, 2020 (DOI: 10.1109/LGRS.2019.2960945).


ARTIFICIAL
"INTELLIGENCE"
makeameme.org


People telling me AI is going to destroy the world
My neural network
Dog


THAT MOMENT WHEN
ARTIFICIAL INTELLIGENCE IS MORE COMMON THAN NATURAL INTELLIGENCE
makeameme.org


5000 YEARS OF ACCUMULATED HUMAN KNOWLEDGE AND KNOWHOW
38
CHATGPT
imgflip.com

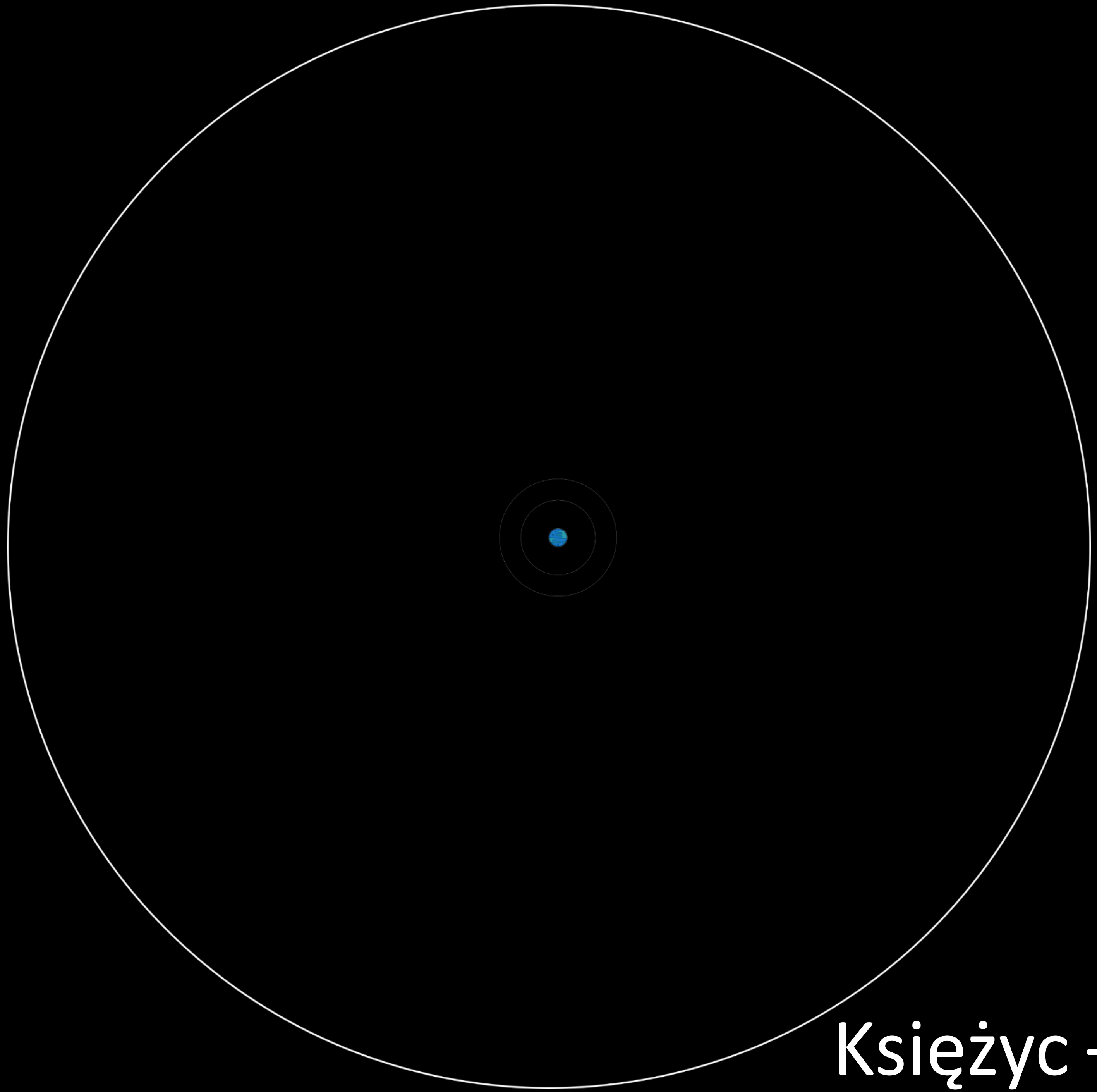
Dokąd Lecimy?

ISS – 400 km

GPS – 20200 km

Geo – 35800 km

KP LABS

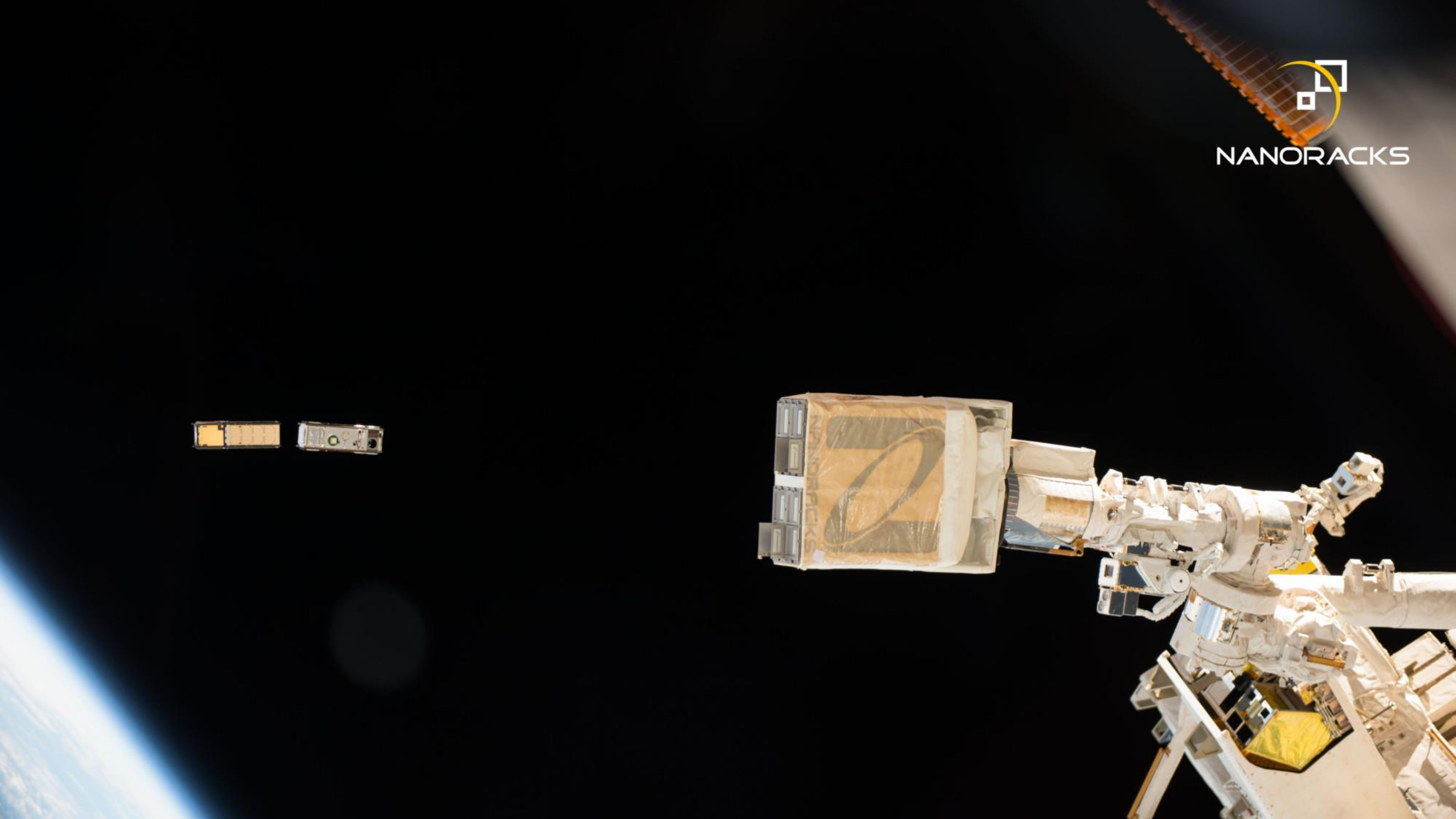


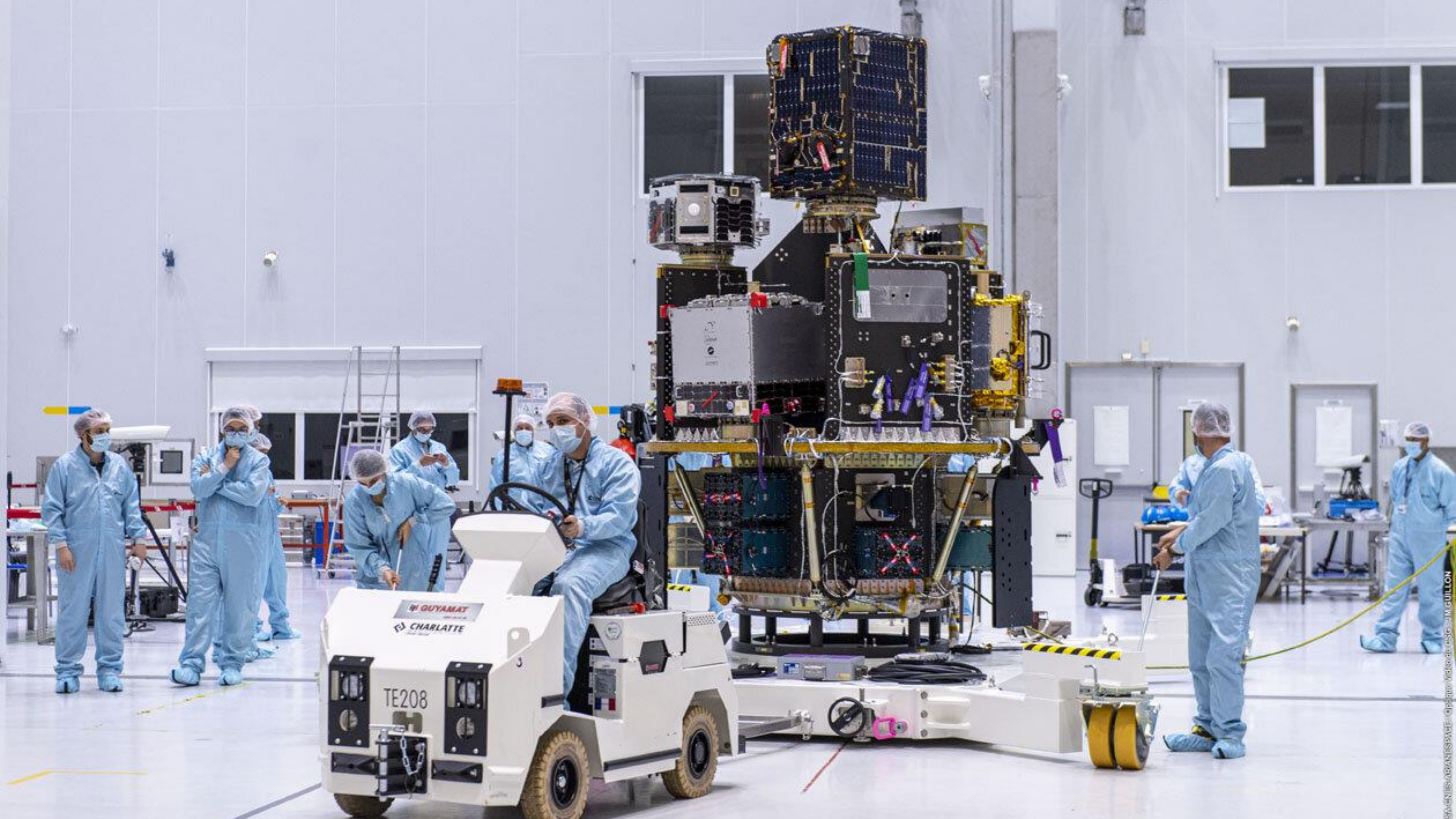
Księżyc – 378000 km

https://www.youtube.com/watch?v=ylvgxNF3C0c

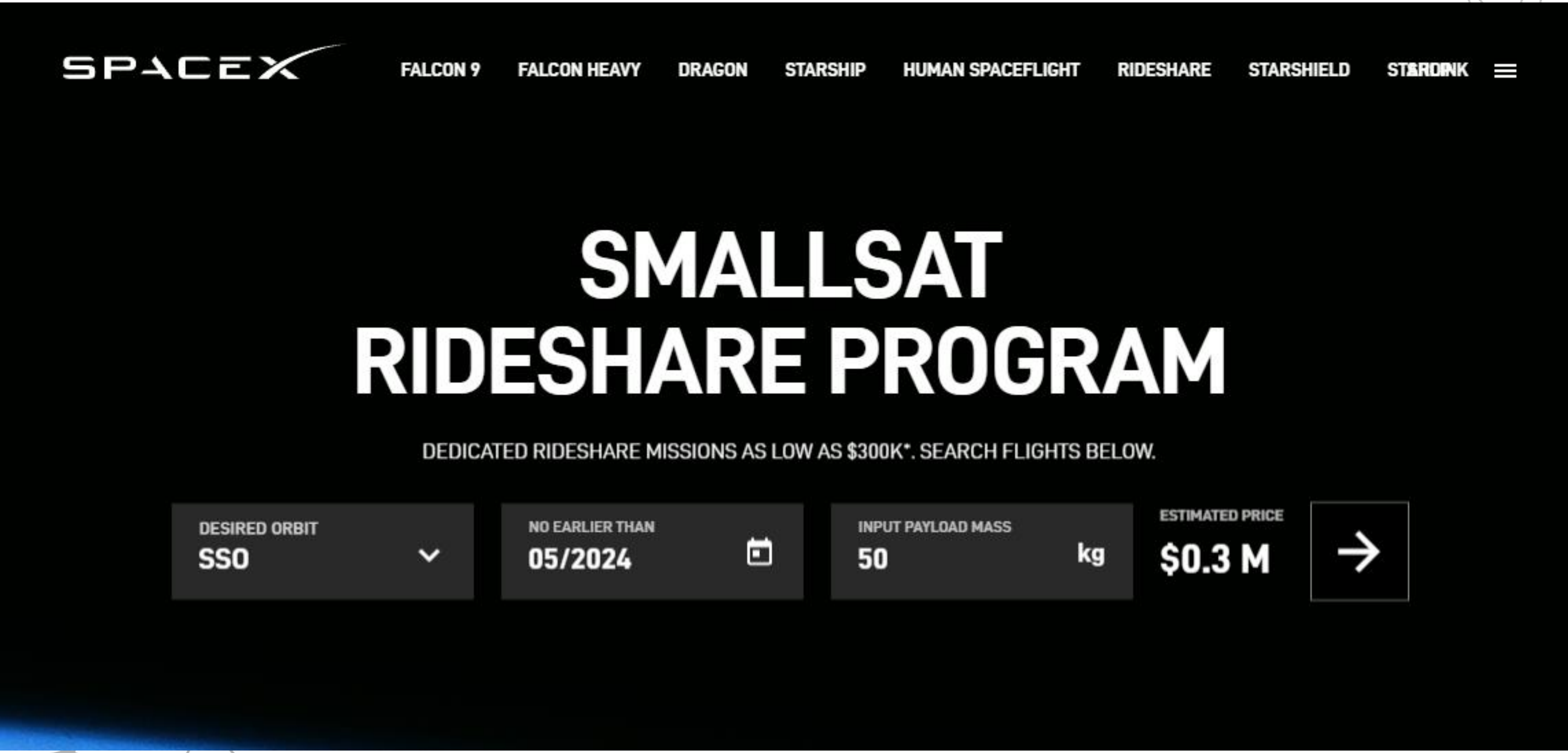
Jak Lecimy?


DEPLOYMENTS BEGIN
SES-2
SECO-2
T+01:08:18
TRANSPORTER-7
SPEED
27664
KM/H
ALTITUDE
510
KM
STAGE 2 TELEMETRY


NANORACKS


GUYAMAT
CHARLATTE
TE208
CNES/ARIANESPACE - Optique Vidéo du CSG - JM GUILLON

$$$


SPACEX
FALCON 9
FALCON HEAVY
DRAGON
STARSHIP
HUMAN SPACEFLIGHT
RIDESHARE
STARSHIELD
SMALLSAT
RIDESHARE PROGRAM
DEDICATED RIDESHARE MISSIONS AS LOW AS $300K*. SEARCH FLIGHTS BELOW.
DESIRED ORBIT
SSO
NO EARLIER THAN
05/2024
INPUT PAYLOAD MASS
50
kg
ESTIMATED PRICE
$0.3 M

Fundusze Europejskie
Inteligentny Rozwój

Narodowe Centrum Badań i Rozwoju

Unia Europejska
Europejski Fundusz Rozwoju Regionalnego

Fundusze Europejskie dla Nowoczesnej Gospodarki

Rzeczpospolita Polska

Dofinansowane przez Unię Europejską

PARP
Grupa PFR

esa
European Space Agency

Jak zbudować?

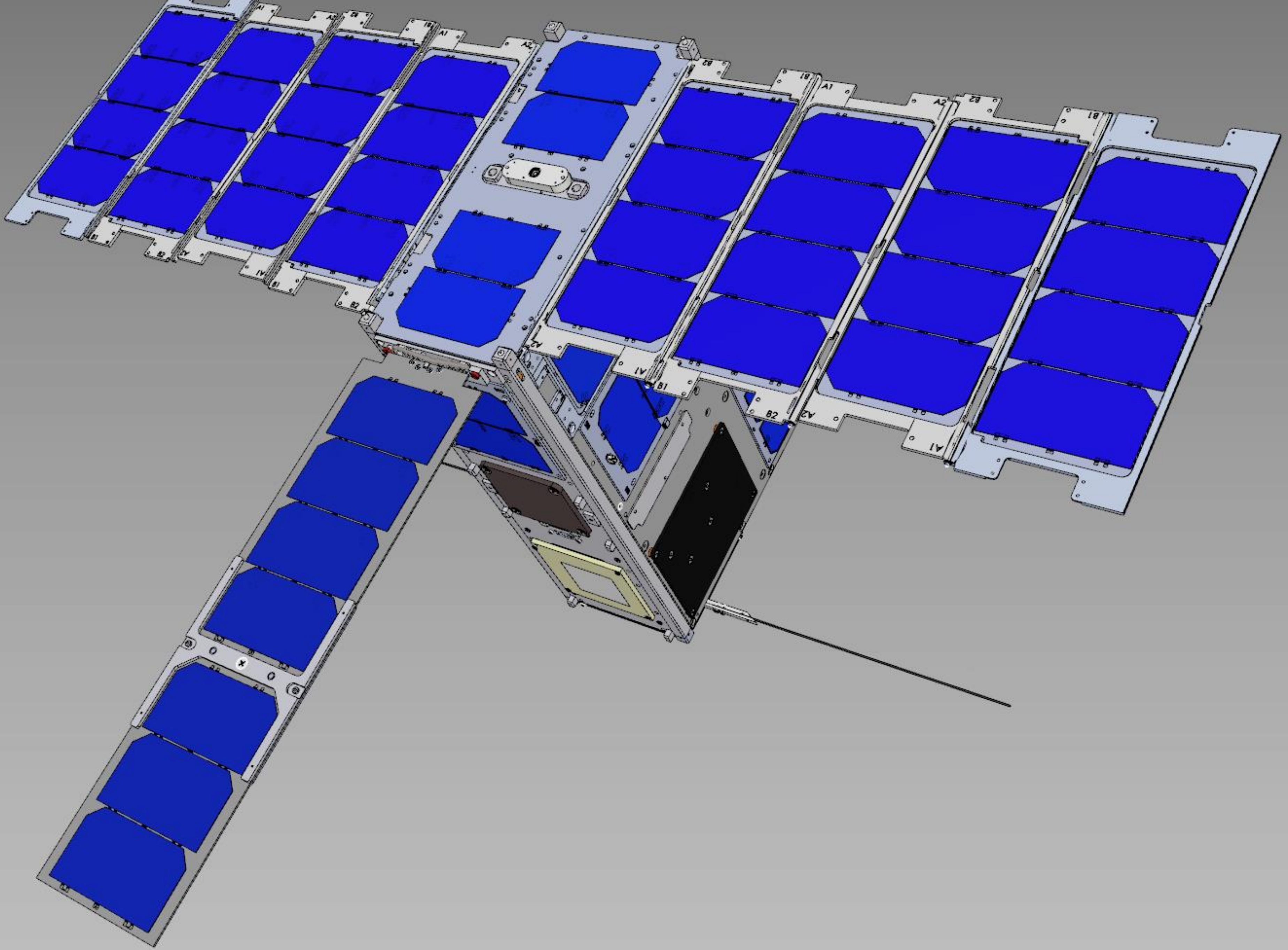
Krok 1: Rozmiary

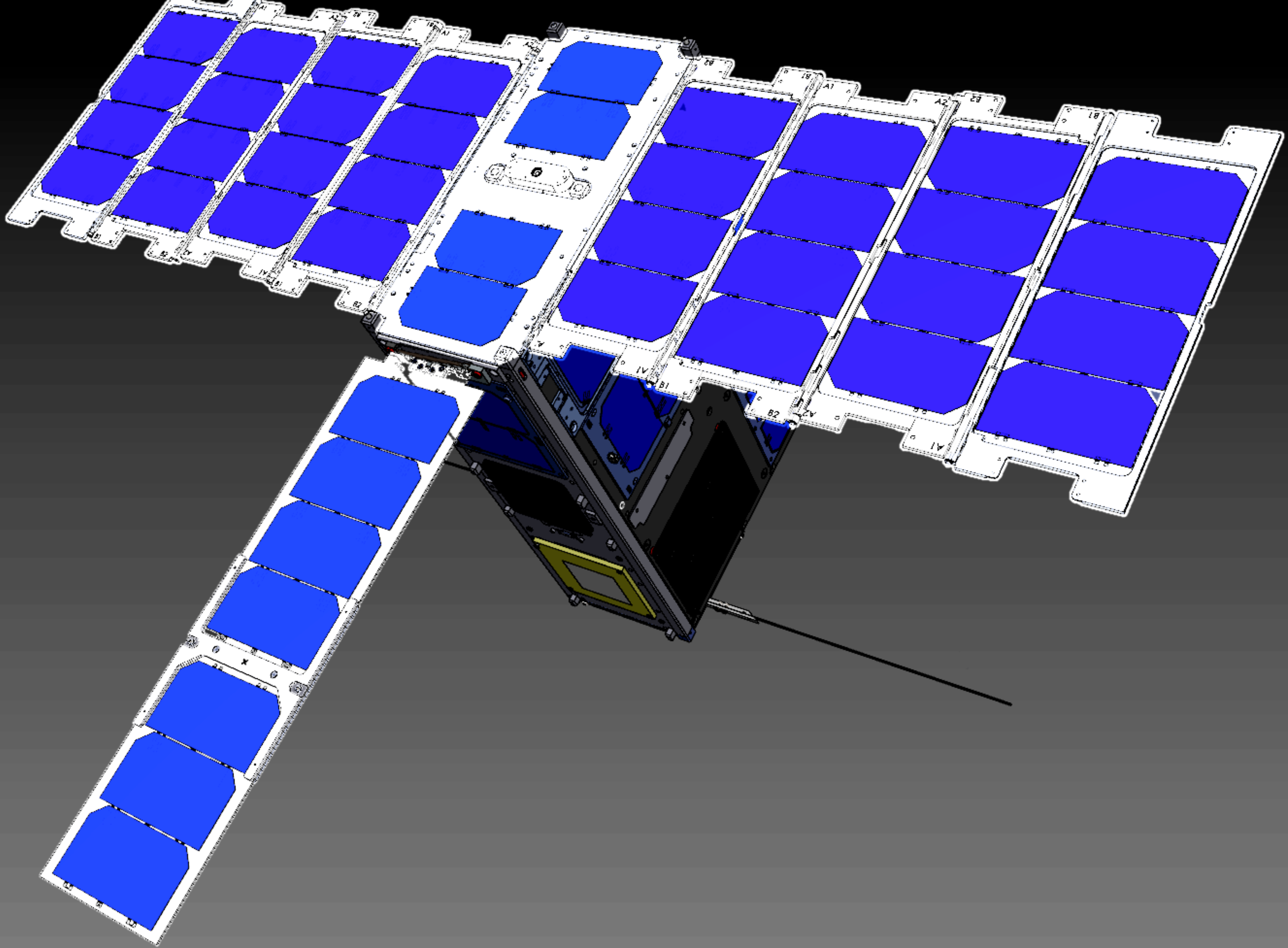

PW-SAT2

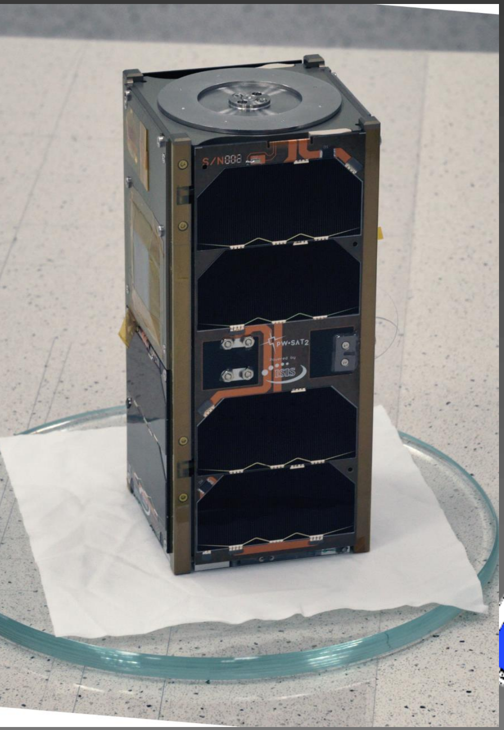
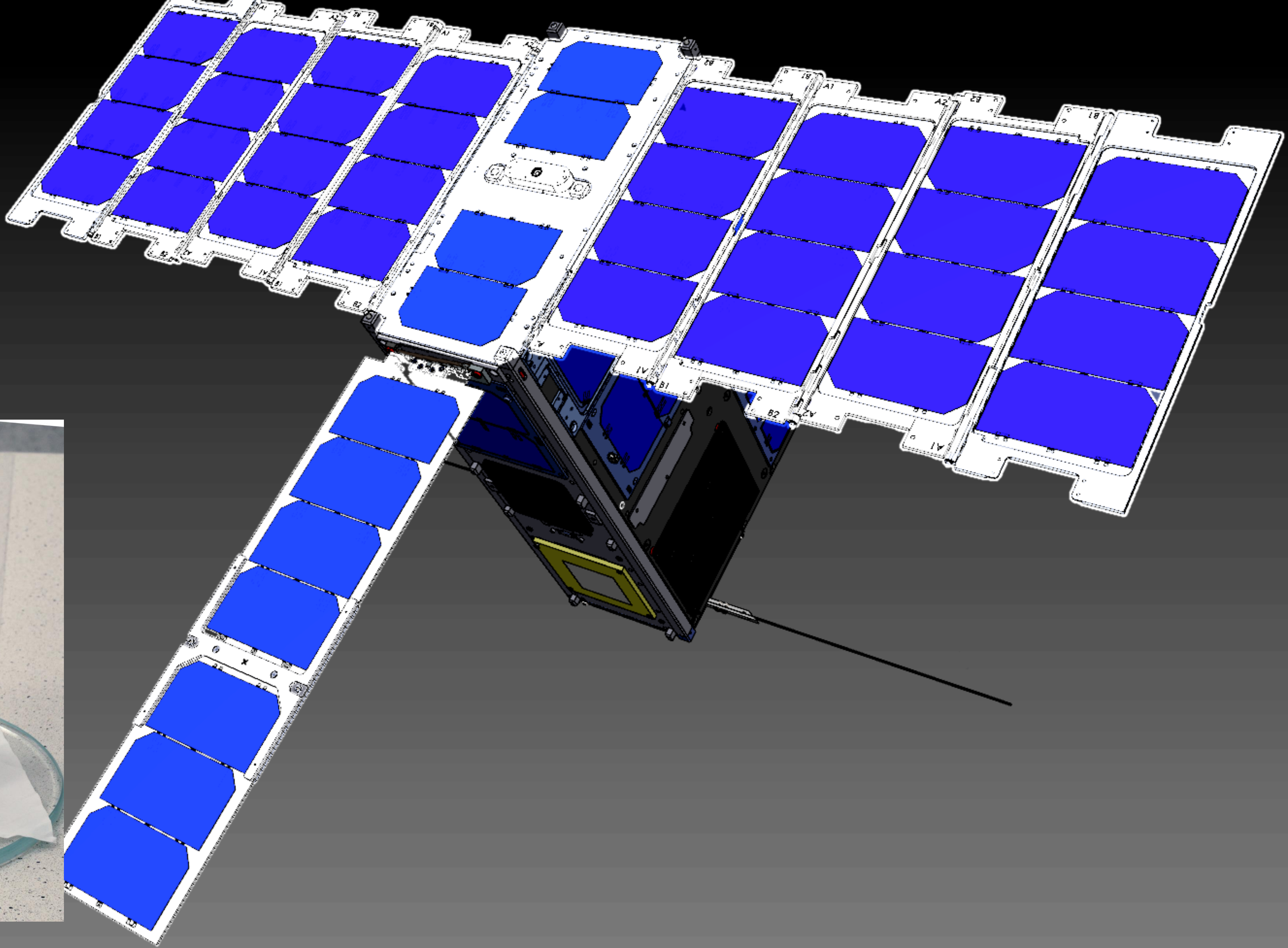

PW-SAT2

Jak zbudować?

Krok 2: Zasilanie

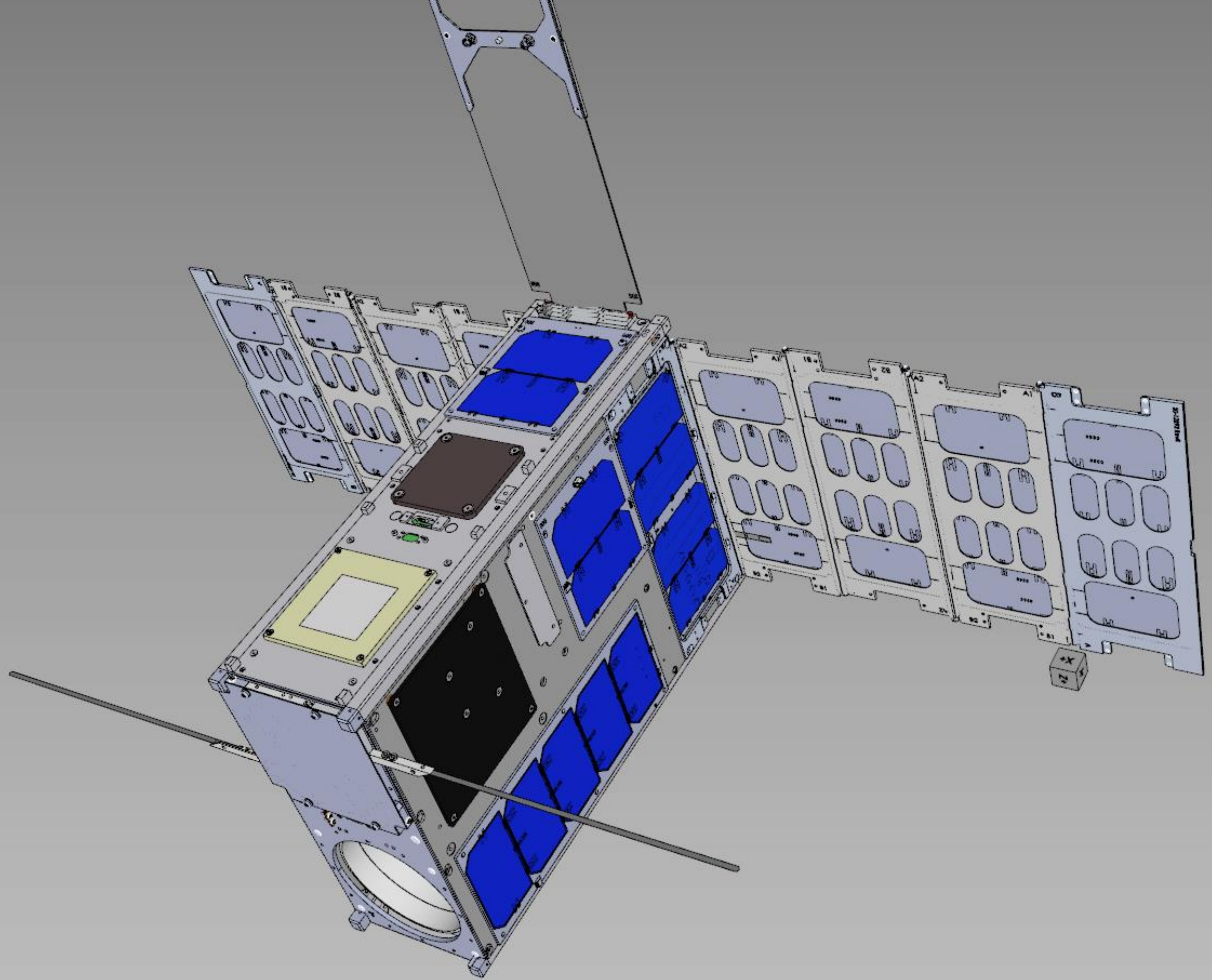


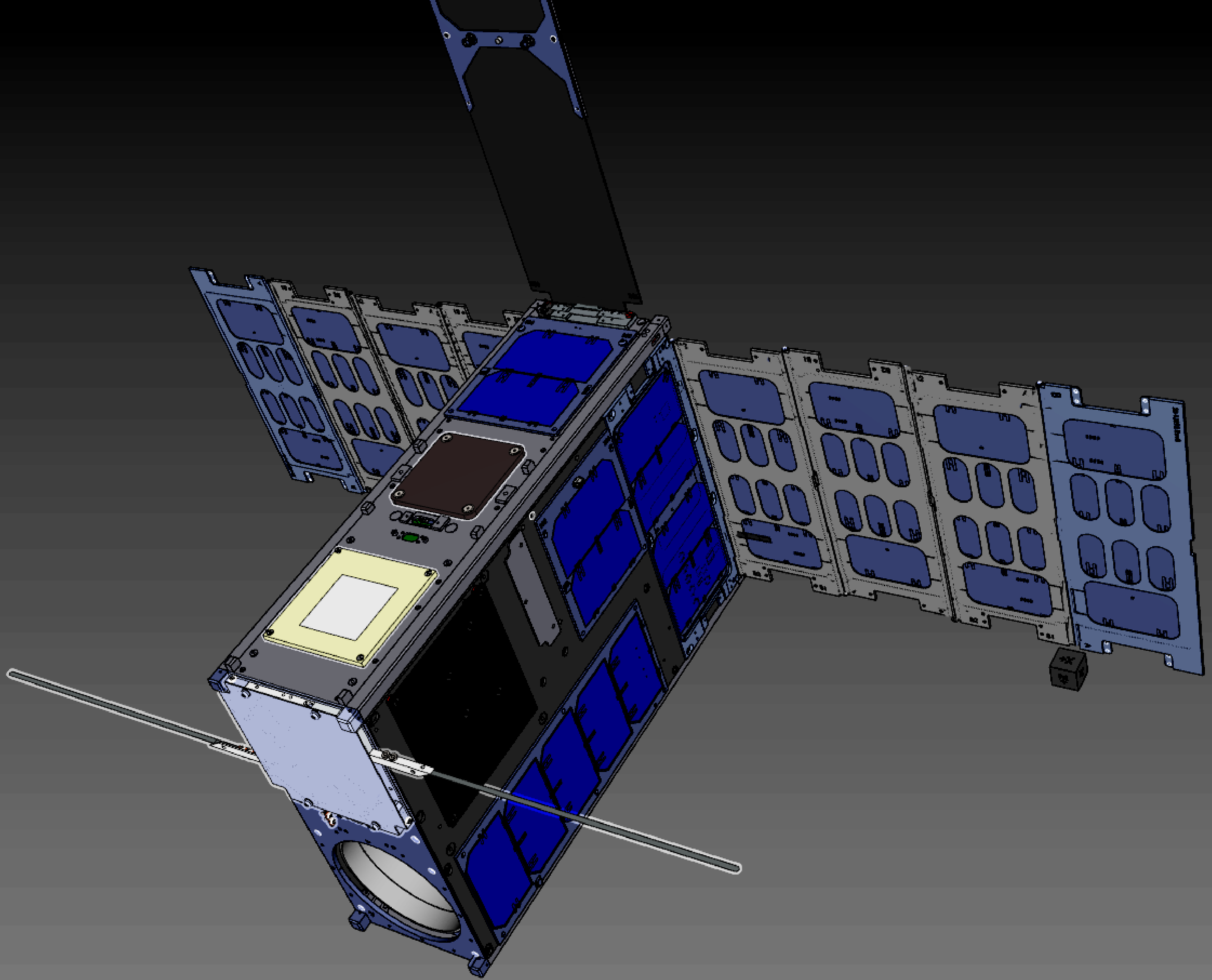


Jak zbudować?

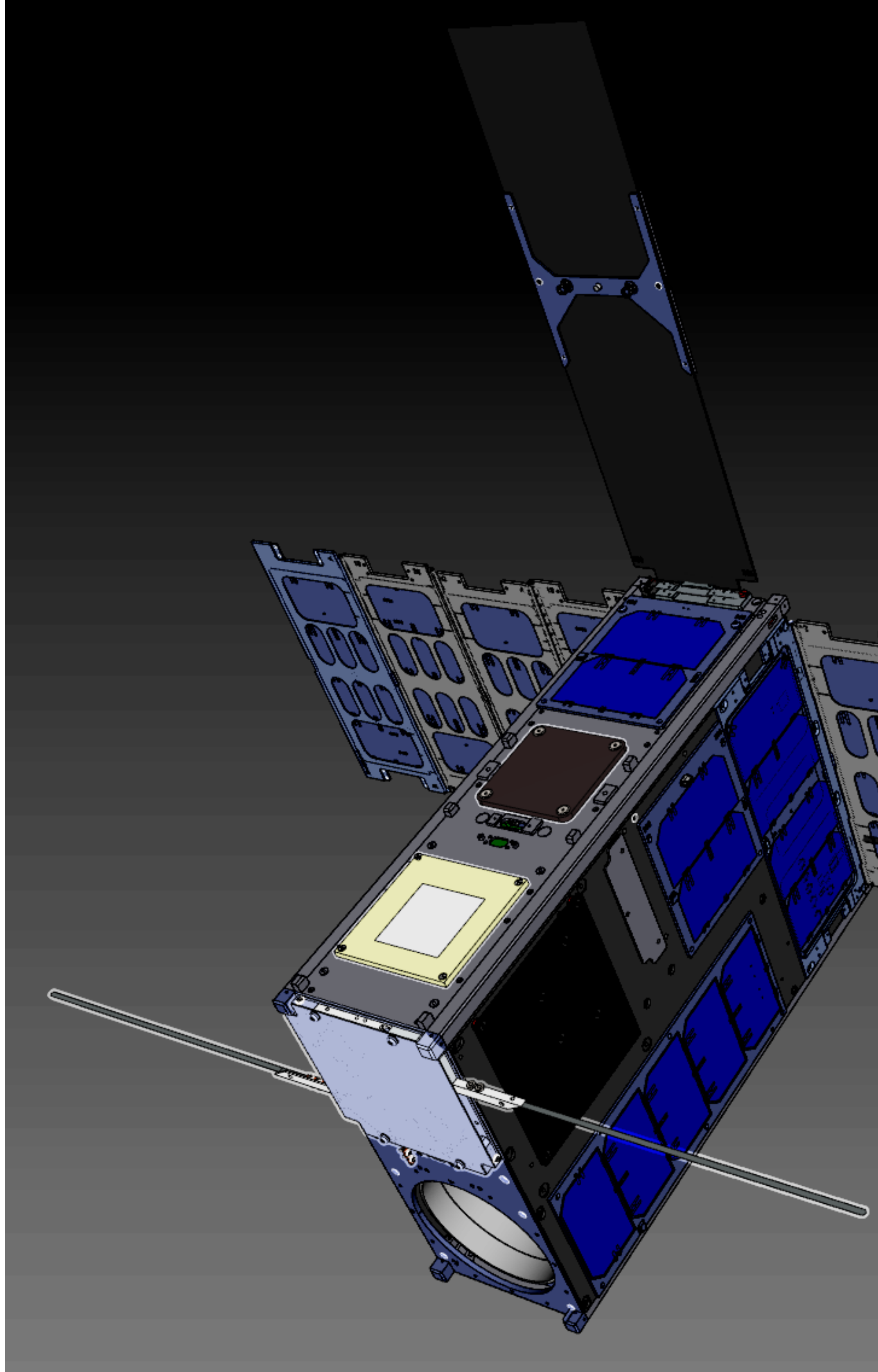
Krok 3: Komunikacja





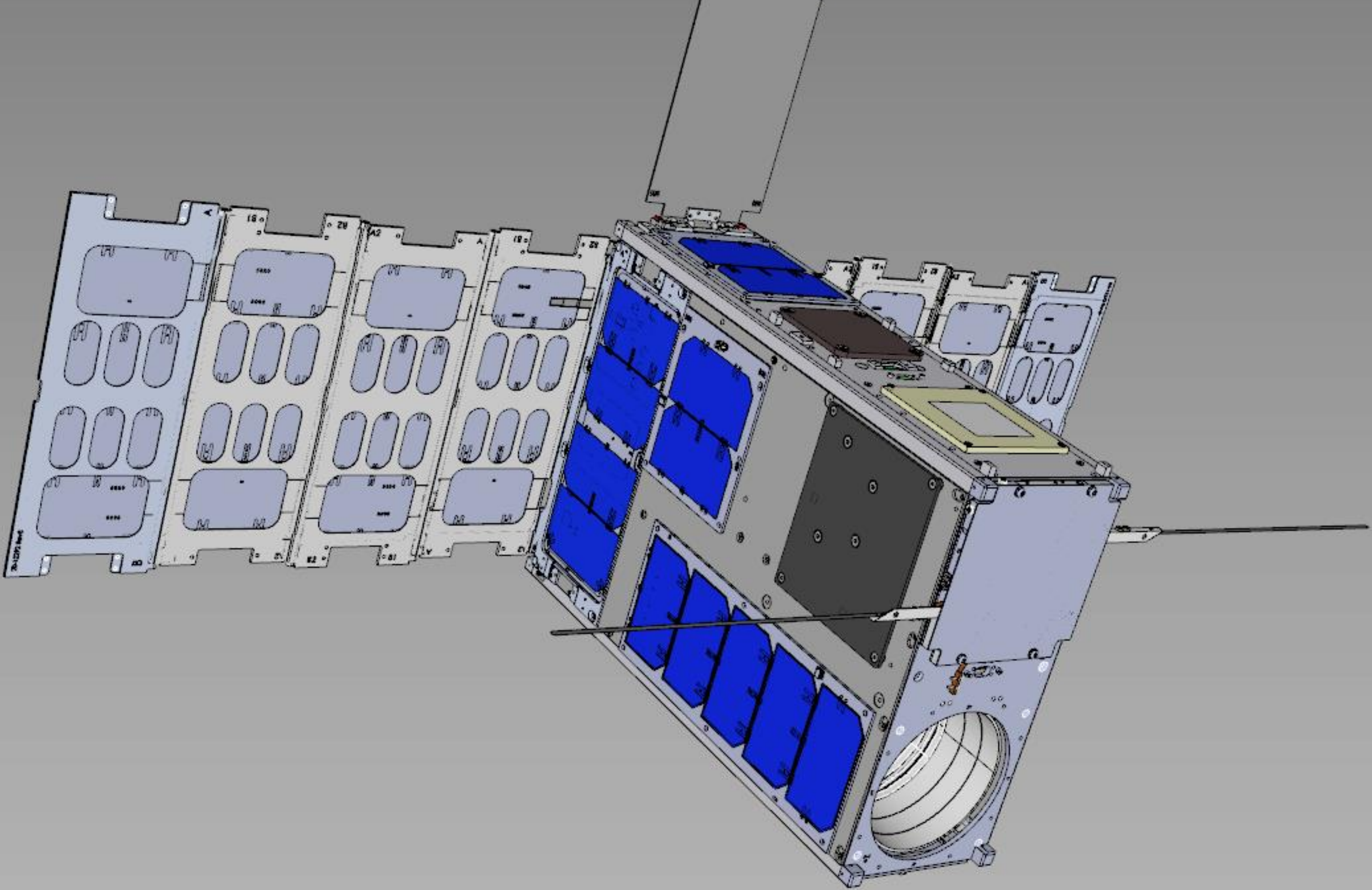
Komunikacja

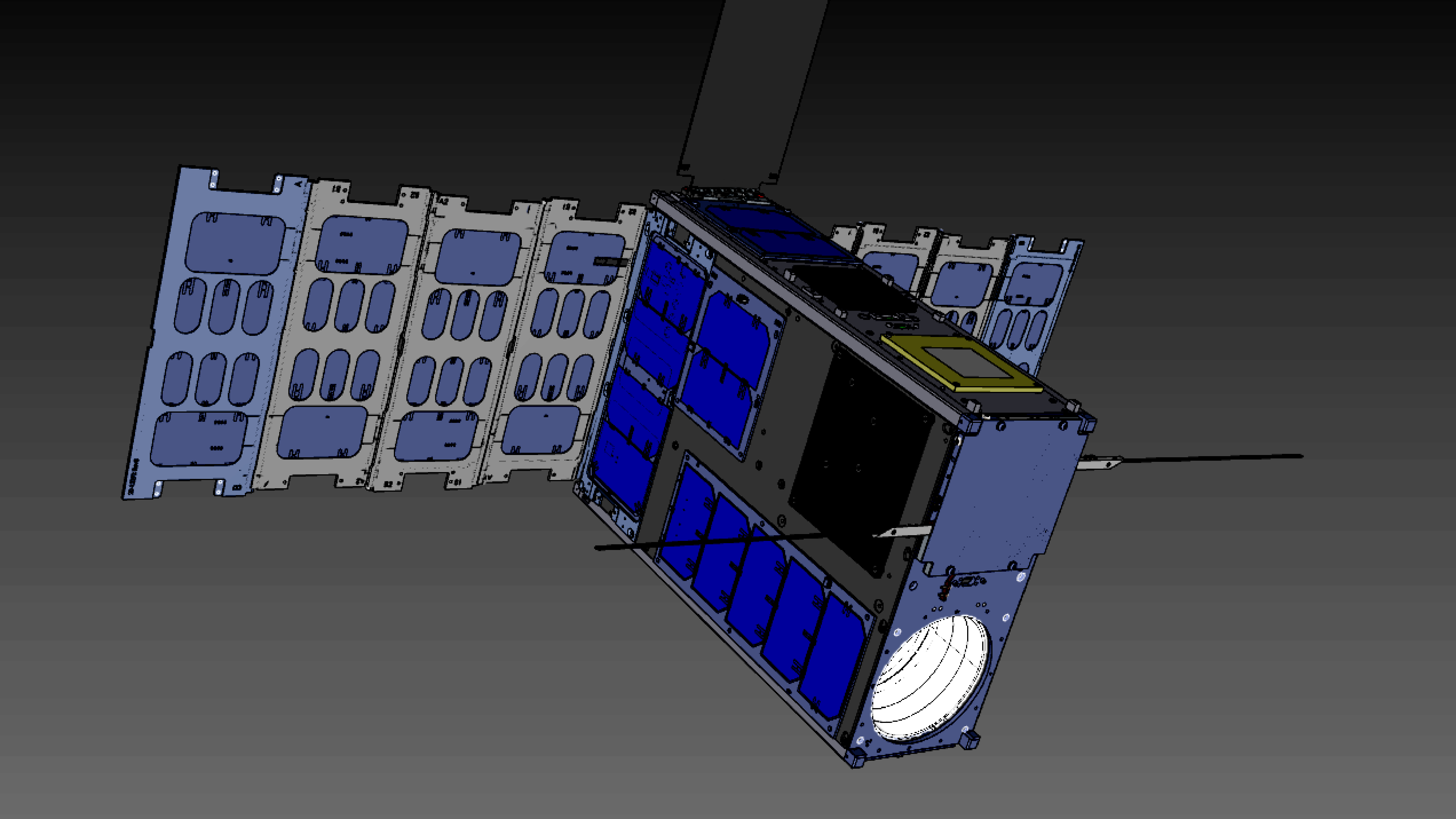
- **SSO @ 530 km**
- **1 stacja naziemna (Gliwice)**
- **3 przeloty rano + 3 przeloty wieczorem**
- UHF: 9600 bps half-duplex
- S-band: 250 kbps uplink
- X-band: 25 Mbps downlink

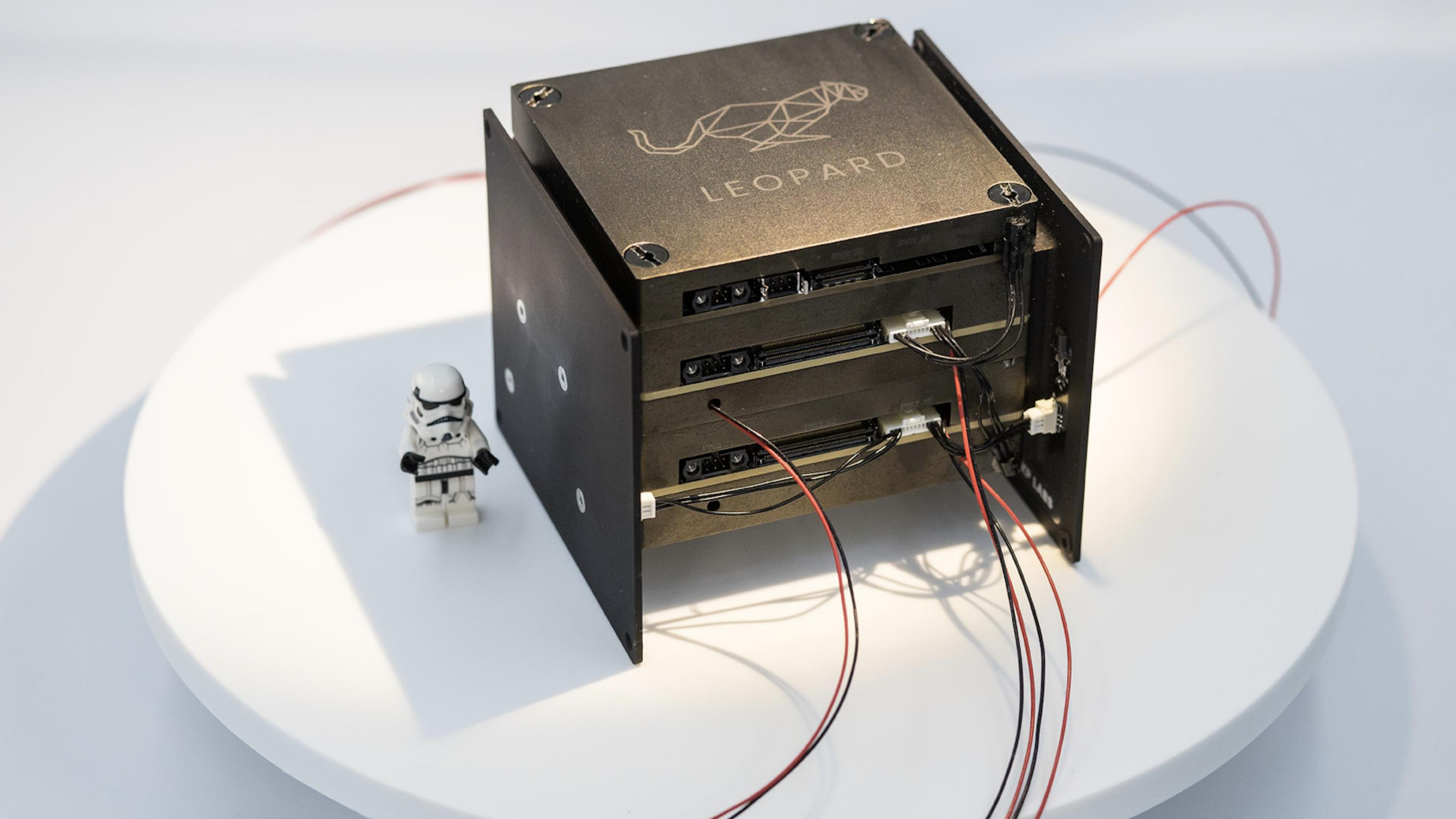


Jak zbudować?

Krok 3: Ładunek






LEOPARD

Jak zbudować?

Krok 4: Główny computer pokładowy

OBC – sprzęt

- Cortex M3 @ 50 MHz
- 8 MB MRAM (NVM), ECC
  - Program + RAM + Storage
- 4 GB NAND, ECC
  - Filesystem (UFFS)
- GPS
- GPIO + UART + I2C + SPI + CAN

OBC – software

- Bootloader + główne oprogramowanie
- FreeRTOS
- Funkcje:
  - Komunikacja
  - Storage (systemy plików, konfiguracja etc)
  - Scheduler
  - Telemetria
  - Mierzenie czasu
  - ...

OBC – komunikacja

- Telekomendy – polecenia wysyłane z Ziemi
- A co jak operator bedzie potrzebować wykonać sekwencje poleceń?
- A co jak sekwencja bedzie warunkowa?
- A co jak zapomnimy jakieś telekomendy?

ECSS-E-ST-70-41C
15 April 2016

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION

Space engineering

Telemetry and telecommand packet utilization

OBC – “aplikacje”

- Wymagania:
  - Nie używa globalnego stanu.
  - Można zarządzać pamięcią.
  - Brak innych zależności na system operacyjny.
  - Silnik oddzielony od interpretera.
  - Łatwość integracji własnego kodu.
  - Izolacja błędów.
- Co wybrać?
  - native?
  - TCL?
  - MicroPython?
  - …

the programming language

Lua

30 years

OBC + Lua

- Dystrybuowana jako biblioteka C
- Da sie łatwo skompilować na dowolnej platformie
- Własny wrapper w C++ dla funkcji biblioteki z Lua

Lua - instancja

```
std::array<std::byte, 1024 * 10> heapSpace;
Heap heap{heapSpace};
ScriptEngine engine{heap};
```

Lua – uruchomienie

```cpp
gsl::span<std::byte> scriptBinary;  // load the compiled script somehow
ScriptBufferReader reader{scriptBinary};

engine.LoadScript(reader);
engine.Execute<void>(); // Execute the whole script
auto result = engine.Execute<std::uint16_t>("function_name", 1, "param2");
```

Lua – integracja

```
onion::Function<std::int32_t(const std::int32_t&, const std::string_view&), 32> function =
    [](const std::int32_t& param1, const std::string_view& param2) -> std::int32_t {
        // Process arguments in param1 and param2
        return 0; // Return some value
    };

engine.Register("my_function", std::move(function));
// From now on, Lua scripts may use my_function
```

```cpp
static std::uint16_t Multiply(std::uint8_t val1, std::uint8_t val2)
{
    return val1 * val2;
}

static const Entry Module_entries[] = {
    {"__index", TableRef(Module_ROTable)},
    LROT_FUNC_WRAP(Multiply),
    TableEnd,
};

Static int MyModuleLoader(lua_State* luaState)
{
    return ModuleLoader(luaState, Module_ROTable);
}

void LoadMyModule(script_engine::ScriptEngine& engine)
{
    engine.LoadModule("MyModule", MyModuleLoader);
}
```

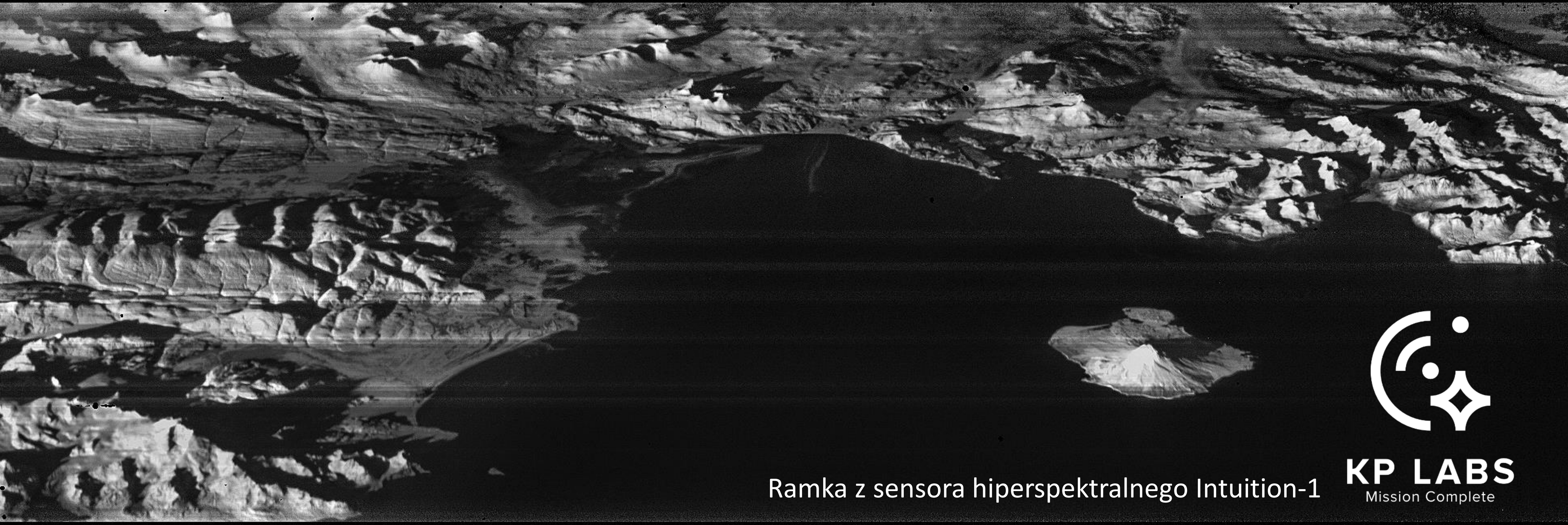
```lua
local filesystem = require 'filesystem'
local string = require 'string'

function action(token)
    logFile:Write('S\n')

    local errorCode, paramsFile = filesystem:OpenFile('/filter_telemetry.params', filesystem.FileMode.ReadOnly)

    local errorCode, params = paramsFile:Read(8)

    local errorCode, paths = paramsFile:Read(256)

    local startIndex, timestampsToDrop  = string.unpack('>LL', params)
    local sourceIdxFileName, sourceDatFileName, targetIdxFileName, targetDatFileName = string.unpack('>zzzz', paths)

    local result = filter(startIndex,
        timestampsToDrop,
        sourceIdxFileName,
        sourceDatFileName,
        targetIdxFileName,
        targetDatFileName)
    logFile:Write('\nE' .. result)

    logFile:Close()
end
```

Zróbmy wreszcie to zdjecie


SPEED
19
KM/H
ALTITUDE
0.1
KM
STAGE 1 TELEMETRY
STARTUP
LIFTOFF
MAX-Q
MECO
BOOSTBACK
FAIRING
T+00:00:03
TRANSPORTER-9
LIFTOFF
THE HOLDDOWN CLAMPS HAVE RELEASED
FALCON 9 AND WE HAVE BEGUN OUR FLIGHT


2024-01-10 11:34:47

2024



Ramka z sensora hiperspektralnego Intuition-1

KP LABS
Mission Complete

https://kplabs.space

Embedded Software Engineer

PRACA@KPLABS.PL

Job Description:

- Implementing low-level software ran on embedded devices using C++.
- Creating software for flight computers used in space missions (on-board computers, processing units and others).
- Implementing and maintaining automated tests using Python.
- Implementing simulators of satellite subsystems.
- Creating technical documentation for the software.

What we expect:

- Willingness to learn new things.
- Proficiency in C++ 17 and CMake.
- Experience in embedded software.
- Knowledge of embedded RTOS.

KP LABS
Mission Complete

THANK YOU

mdrobik@kplabs.pl

www.kplabs.pl

KP Labs
Bojkowska 37J
44-100 Gliwice,
Poland