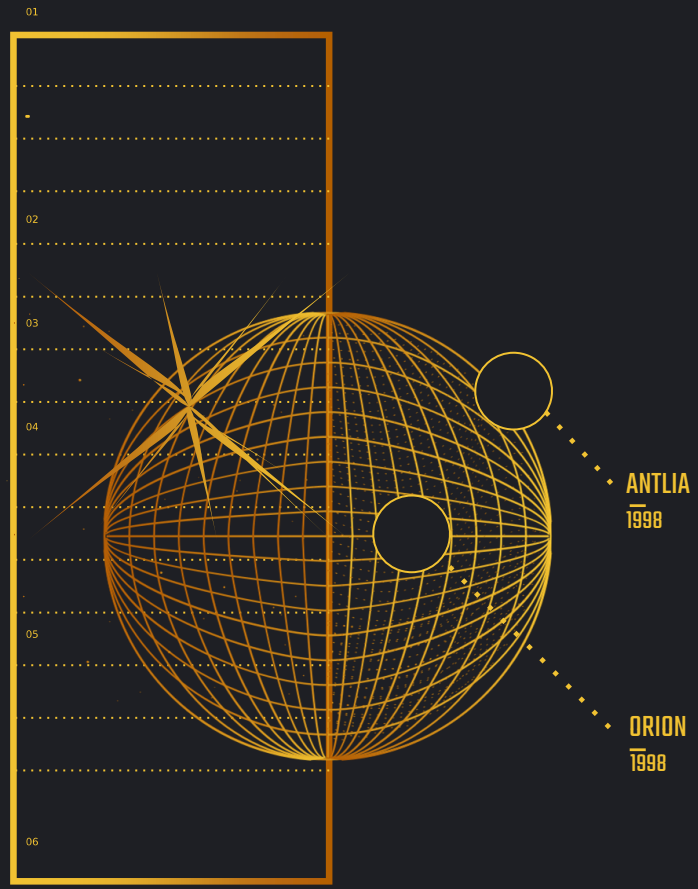
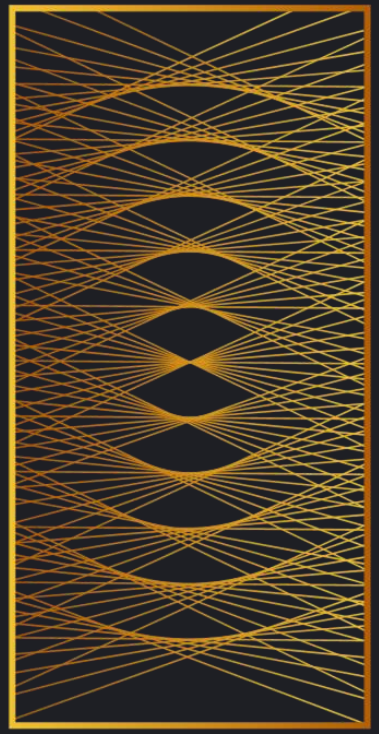


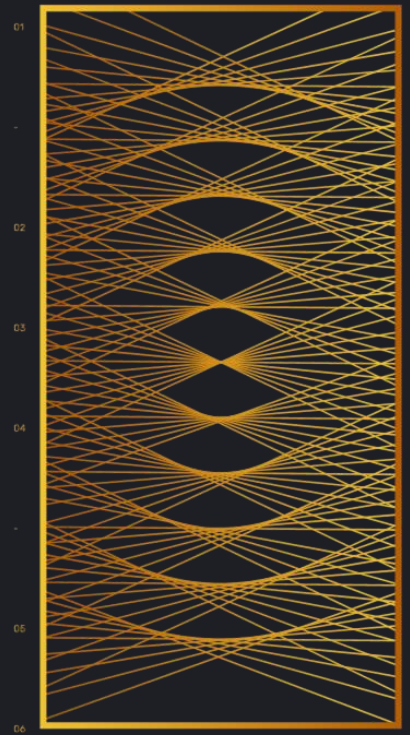
TEAM ANANT

ORIENTATION



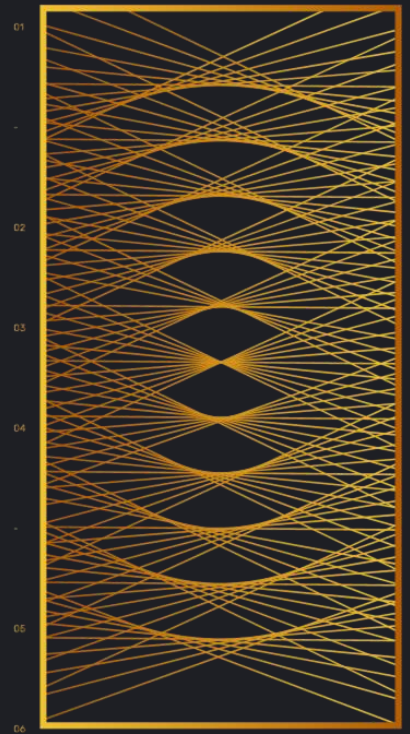
01.

Intro to Team Anant



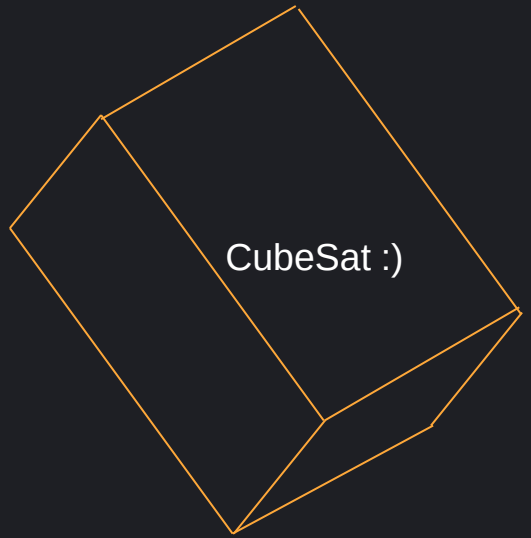
02.

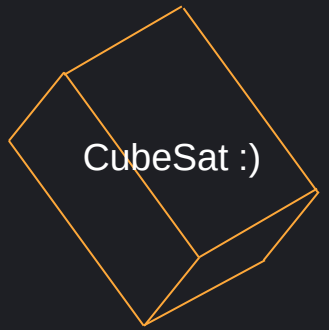
Systems Overview



03.

Attitude Determination and Control Systems

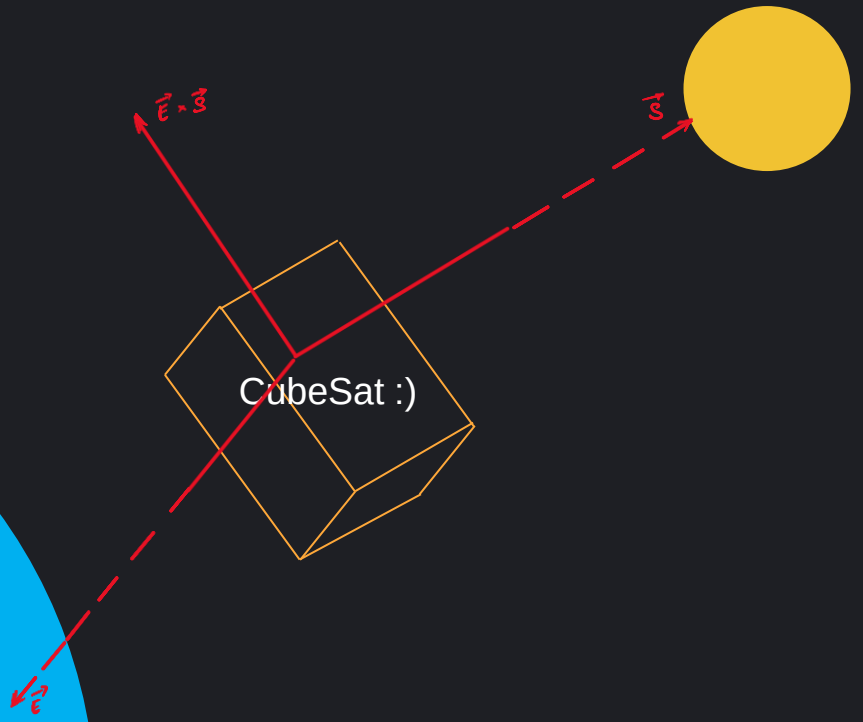

CubeSat :)



CubeSat :)

01 02 03 04 05 06

01 02 03 04 05 06



01 02 03 04 05 06

01 02 03 04 05 06

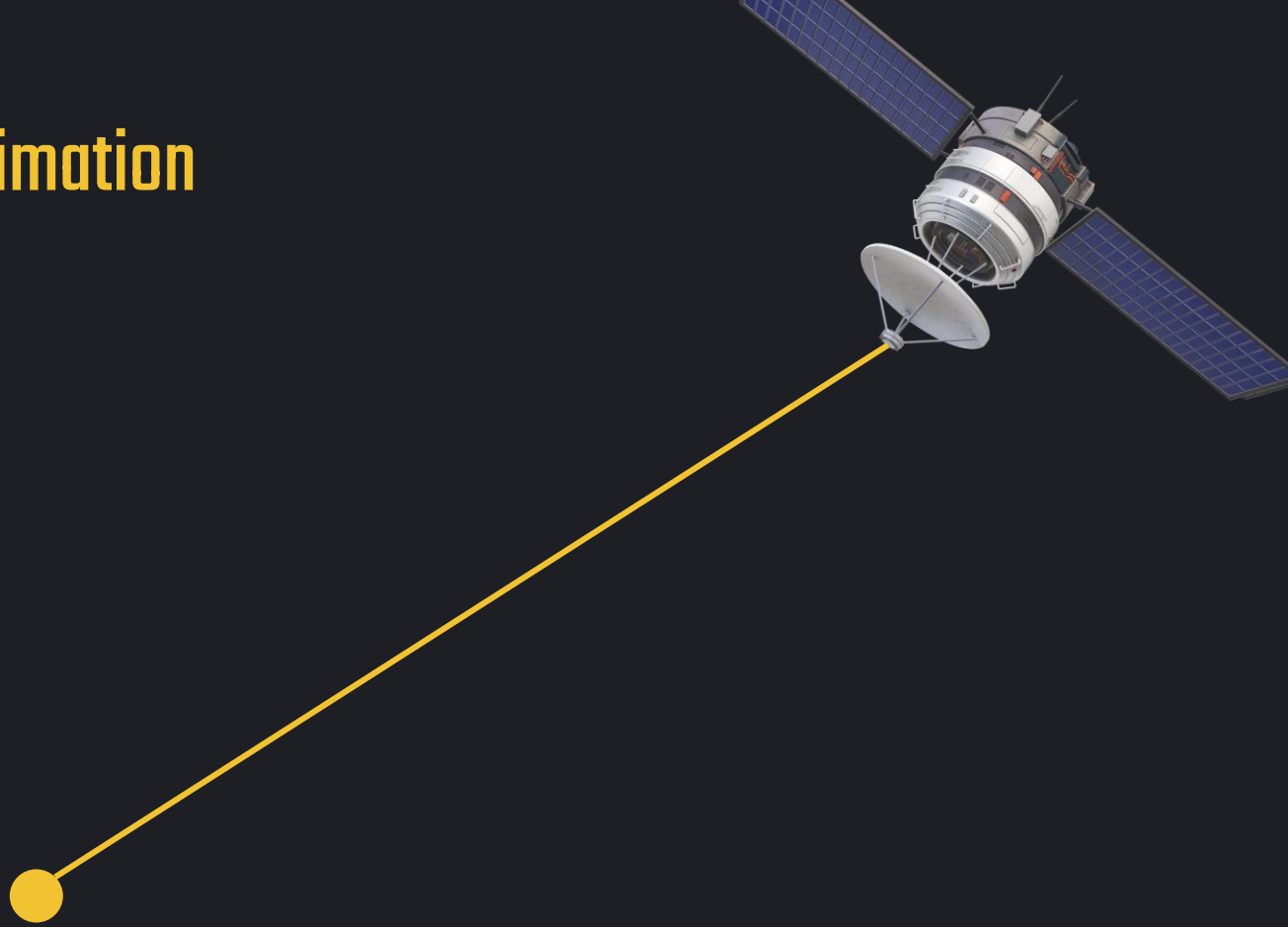
ATTITUDE DETERMINATION HARDWARE

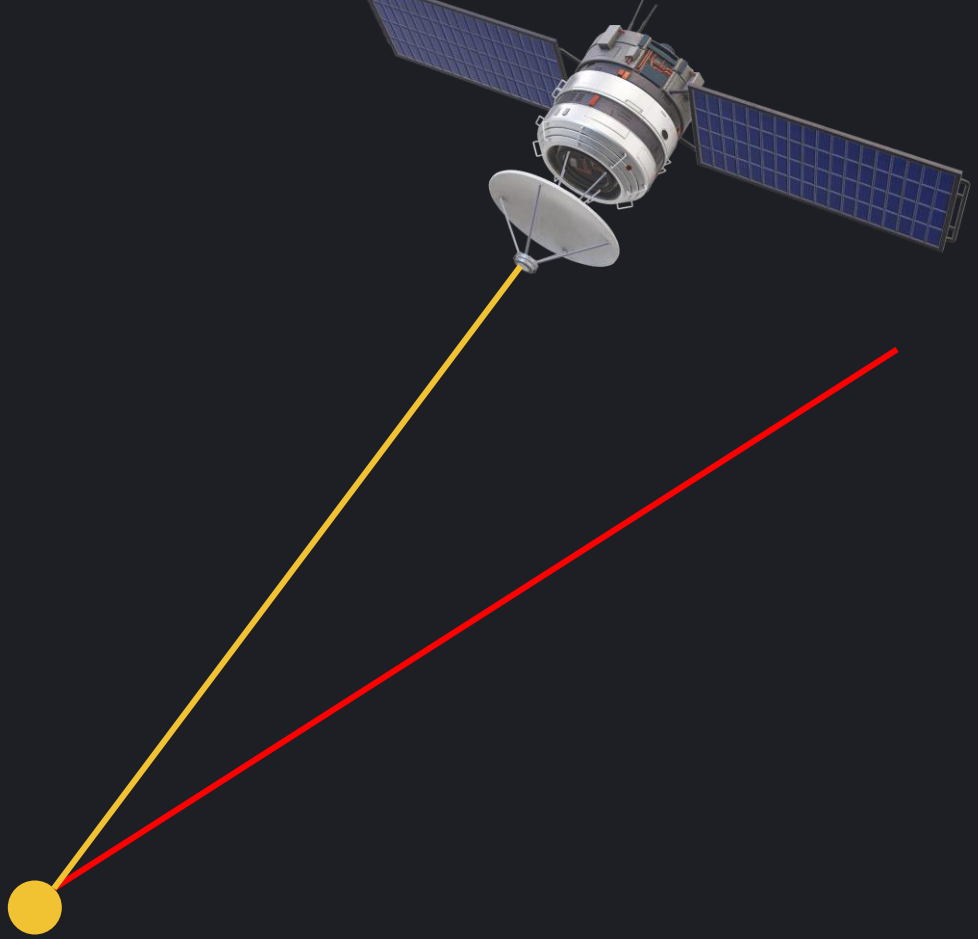
MAGNETIC FIELD SENSOR

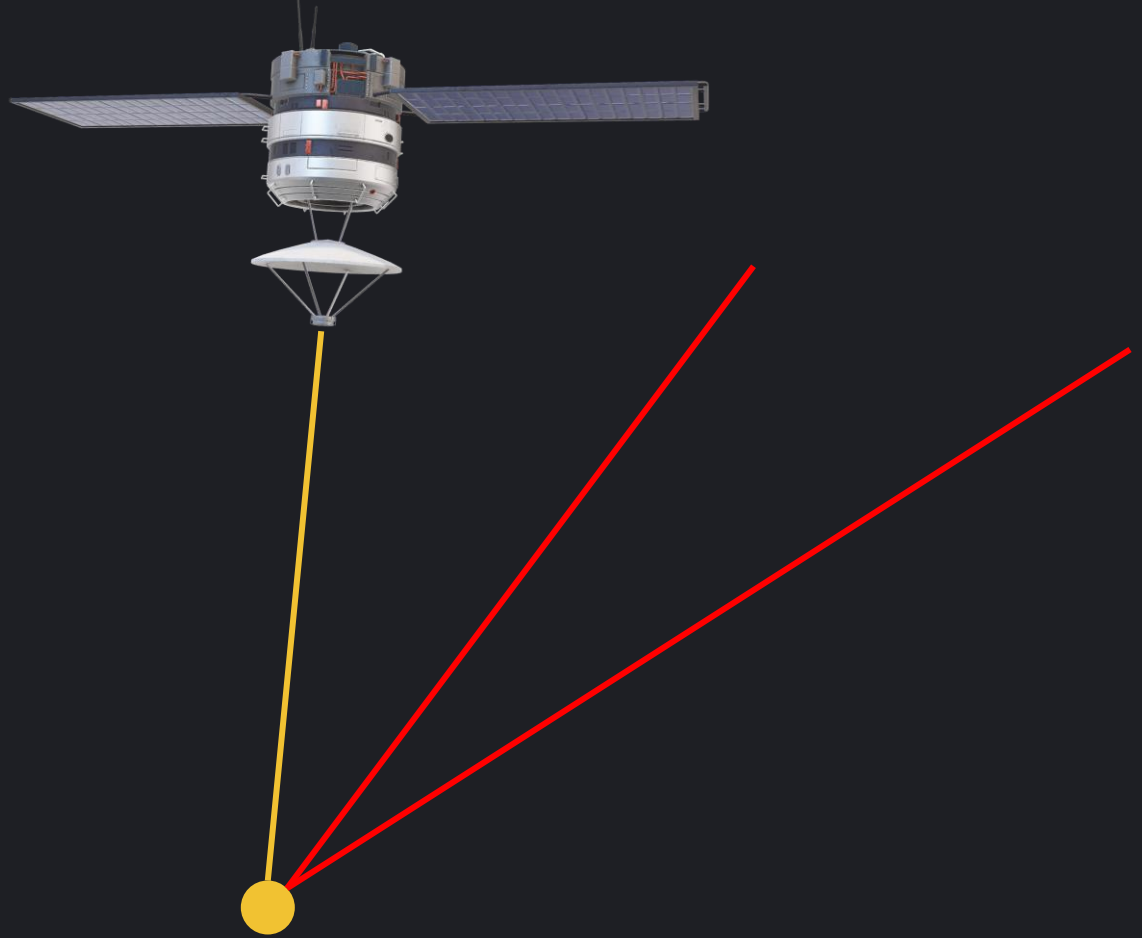
SUN SENSOR

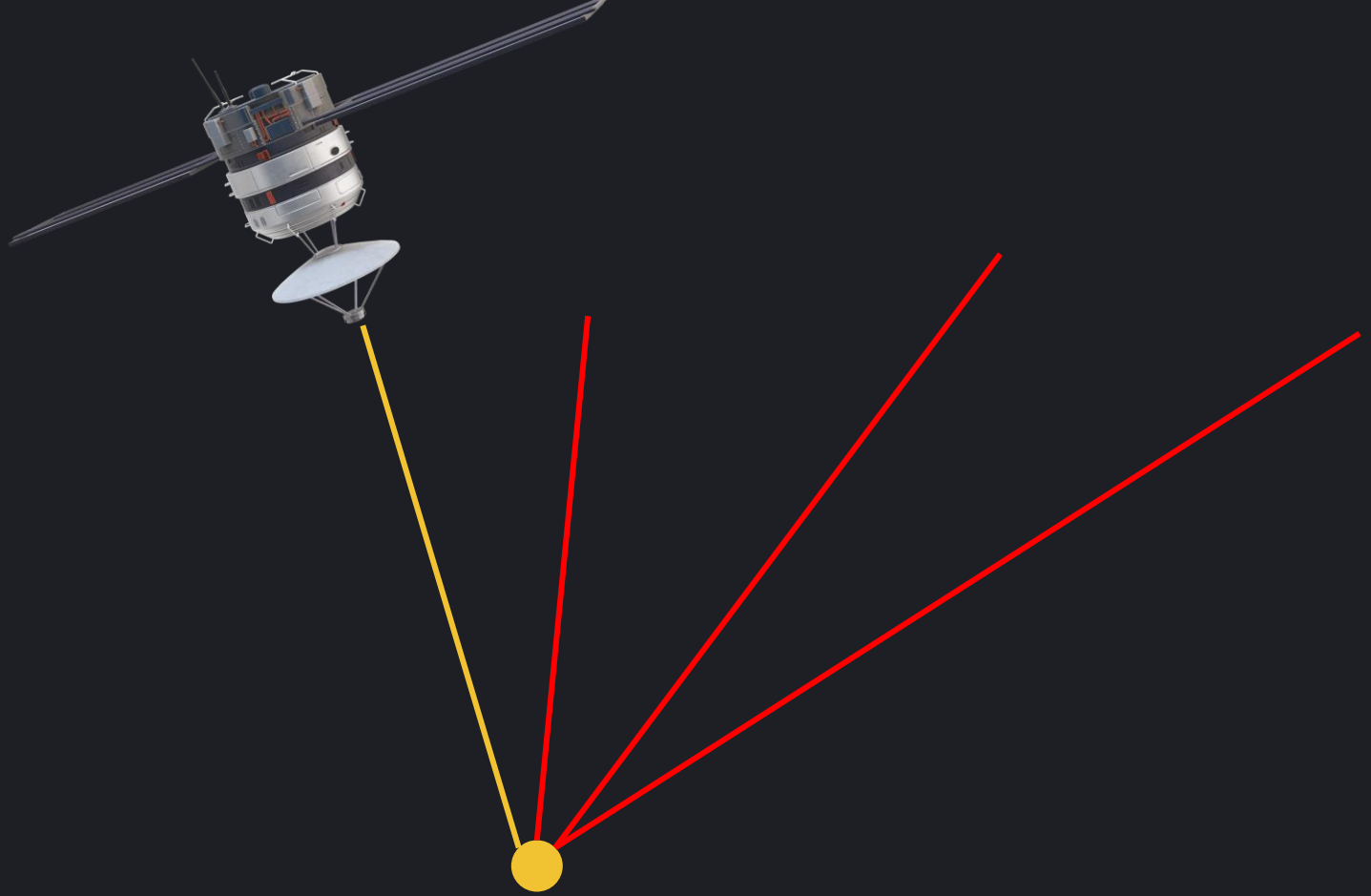
STAR SENSOR

The Need for Estimation









ESTIMATIONS

DATA PREDICTION

DATA FILTERING

Our filter is estimating the error state, and therefore the filter correction equations[26],

$$\mathbf{K} = \mathbf{P}\mathbf{H}^\top(\mathbf{H}\mathbf{P}\mathbf{H}^\top + \mathbf{V})^{-1} \tag{273}$$

$$\hat{\delta\mathbf{x}} \leftarrow \mathbf{K}(\mathbf{y} - h(\hat{\mathbf{x}}_t)) \tag{274}$$

$$\mathbf{P} \leftarrow (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P} \tag{275}$$

require the Jacobian matrix $\mathbf{H}$ to be defined with respect to the error state $\delta\mathbf{x}$, and evaluated at the best true-state estimate $\hat{\mathbf{x}}_t = \mathbf{x} \oplus \hat{\delta\mathbf{x}}$. As the error state mean is zero at this stage (we have not observed it yet), we have $\hat{\mathbf{x}}_t = \mathbf{x}$ and we can use the nominal error $\mathbf{x}$ as the evaluation point, leading to

where $h()$ is a general nonlinear function of the system state (the true state), and v is a white Gaussian noise with covariance $\mathbf{V}$,

$$v \sim \mathcal{N}\{0, \mathbf{V}\} \ . \tag{272}$$

Our filter is estimating the error state, and therefore the filter correction equations[26],

$$\mathbf{K} = \mathbf{P}\mathbf{H}^\top(\mathbf{H}\mathbf{P}\mathbf{H}^\top + \mathbf{V})^{-1} \tag{273}$$

$$\hat{\delta\mathbf{x}} \leftarrow \mathbf{K}(\mathbf{y} - h(\hat{\mathbf{x}}_t)) \tag{274}$$

$$\mathbf{P} \leftarrow (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P} \tag{275}$$

require the Jacobian matrix $\mathbf{H}$ to be defined with respect to the error state $\delta\mathbf{x}$, and evaluated at the best true-state estimate $\hat{\mathbf{x}}_t = \mathbf{x} \oplus \hat{\delta\mathbf{x}}$. As the error state mean is zero at this stage (we have not observed it yet), we have $\hat{\mathbf{x}}_t = \mathbf{x}$ and we can use the nominal error $\mathbf{x}$ as the evaluation point, leading to

$$\mathbf{H} \equiv \left.\frac{\partial h}{\partial \delta\mathbf{x}}\right|_{\mathbf{x}} \ . \tag{276}$$

6.1.1 Jacobian computation for the filter correction

The Jacobian above might be computed in a number of ways. The most illustrative one is by making use of the chain rule,

$$\mathbf{H} \triangleq \left.\frac{\partial h}{\partial \delta\mathbf{x}}\right|_{\mathbf{x}} = \left.\frac{\partial h}{\partial \mathbf{x}_t}\right|_{\mathbf{x}} \left.\frac{\partial \mathbf{x}_t}{\partial \delta\mathbf{x}}\right|_{\mathbf{x}} = \mathbf{H}_{\mathbf{x}}\ \mathbf{X}_{\delta\mathbf{x}} \ . \tag{277}$$

Here, $\mathbf{H}_{\mathbf{x}} \triangleq \left.\frac{\partial h}{\partial \mathbf{x}_t}\right|_{\mathbf{x}}$ is the standard Jacobian of $h()$ with respect to its own argument (*i.e.*, the Jacobian one would use in a regular EKF). This first part of the chain rule depends on the measurement function of the particular sensor used, and is not presented here.

The second part, $\mathbf{X}_{\delta\mathbf{x}} \triangleq \left.\frac{\partial \mathbf{x}_t}{\partial \delta\mathbf{x}}\right|_{\mathbf{x}}$, is the Jacobian of the true state with respect to the error state. This part can be derived here as it only depends on the ESKF composition of states. We have the derivatives,

$$\mathbf{X}_{\delta\mathbf{x}} = \begin{bmatrix} \frac{\partial(\mathbf{p}+\delta\mathbf{p})}{\partial\delta\mathbf{p}} & & & & & & \\ & \frac{\partial(\mathbf{v}+\delta\mathbf{v})}{\partial\delta\mathbf{v}} & & & & 0 & \\ & & \frac{\partial(\mathbf{q}\otimes\delta\mathbf{q})}{\partial\delta\boldsymbol{\theta}} & & & & \\ & & & \frac{\partial(\mathbf{a}_b+\delta\mathbf{a}_b)}{\partial\delta\mathbf{a}_b} & & & \\ & 0 & & & \frac{\partial(\boldsymbol{\omega}_b+\delta\boldsymbol{\omega}_b)}{\partial\delta\boldsymbol{\omega}_b} & & \\ & & & & & \frac{\partial(\mathbf{g}+\delta\mathbf{g})}{\partial\delta\mathbf{g}} & \end{bmatrix} \tag{278}$$

[26] We give the simplest form of the covariance update, $\mathbf{P} \leftarrow (\mathbf{I}-\mathbf{K}\mathbf{H})\mathbf{P}$. This form is known to have poor numerical stability, as its outcome is not guaranteed to be symmetric nor positive definite. The reader is free to use more stable forms such as 1) the symmetric form $\mathbf{P} \leftarrow \mathbf{P} - \mathbf{K}(\mathbf{H}\mathbf{P}\mathbf{H}^\top + \mathbf{V})\mathbf{K}^\top$ and 2) the symmetric and positive *Joseph* form $\mathbf{P} \leftarrow (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}(\mathbf{I} - \mathbf{K}\mathbf{H})^\top + \mathbf{K}\mathbf{V}\mathbf{K}^\top$.

- The perturbations $\mathbf{w}$ are never sampled.

As a consequence, the integration over Δt of these two stochastic processes differs. Let us examine it.

The continuous-time error-state dynamics (424) can be linearized to

$$\dot{\delta\mathbf{x}} = \mathbf{A}\delta\mathbf{x} + \mathbf{B}\tilde{\mathbf{u}} + \mathbf{C}\mathbf{w} \ , \tag{427}$$

with

$$\mathbf{A} \triangleq \left.\frac{\partial f}{\partial \delta\mathbf{x}}\right|_{\mathbf{x},\mathbf{u}_m} \quad , \quad \mathbf{B} \triangleq \left.\frac{\partial f}{\partial \tilde{\mathbf{u}}}\right|_{\mathbf{x},\mathbf{u}_m} \quad , \quad \mathbf{C} \triangleq \left.\frac{\partial f}{\partial \mathbf{w}}\right|_{\mathbf{x},\mathbf{u}_m} \ , \tag{428}$$

and integrated over the sampling period Δt, giving,

$$\delta\mathbf{x}_{n+1} = \delta\mathbf{x}_n + \int_{n\Delta t}^{(n+1)\Delta t} \left(\mathbf{A}\delta\mathbf{x}(\tau) + \mathbf{B}\tilde{\mathbf{u}}(\tau) + \mathbf{C}\mathbf{w}^c(\tau)\right) d\tau \tag{429}$$

$$= \delta\mathbf{x}_n + \int_{n\Delta t}^{(n+1)\Delta t} \mathbf{A}\delta\mathbf{x}(\tau)d\tau + \int_{n\Delta t}^{(n+1)\Delta t} \mathbf{B}\tilde{\mathbf{u}}(\tau)d\tau + \int_{n\Delta t}^{(n+1)\Delta t} \mathbf{C}\mathbf{w}^c(\tau)d\tau \tag{430}$$

which has three terms of very different nature. They can be integrated as follows:

1. From App. B we know that the dynamic part is integrated giving the transition matrix,

   $$\delta\mathbf{x}_n + \int_{n\Delta t}^{(n+1)\Delta t} \mathbf{A}\delta\mathbf{x}(\tau)d\tau = \Phi \cdot \delta\mathbf{x}_n \tag{431}$$

   where $\Phi = e^{\mathbf{A}\Delta t}$ can be computed in closed-form or approximated at different levels of accuracy.

2. From (426) we have

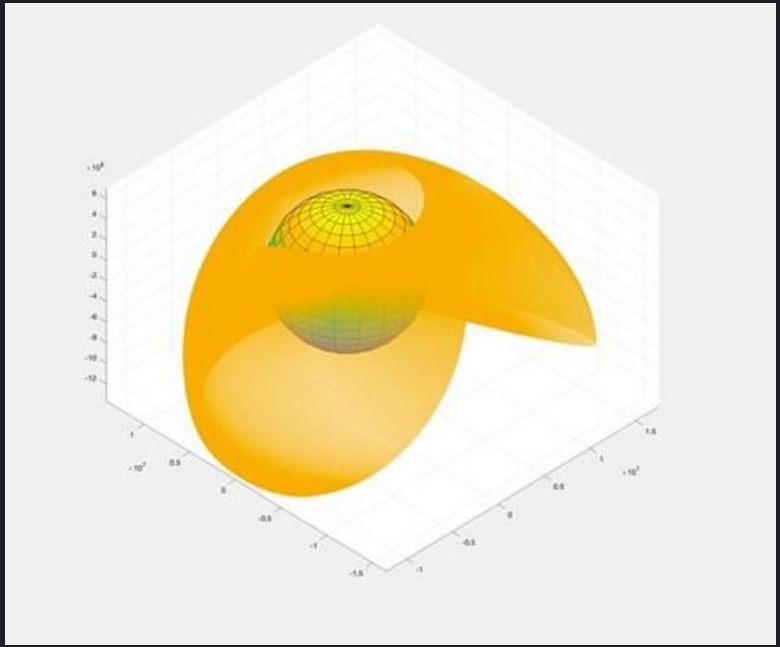
   $$\int_{n\Delta t}^{(n+1)\Delta t} \mathbf{B}\tilde{\mathbf{u}}(\tau)d\tau = \mathbf{B}\Delta t\tilde{\mathbf{u}}_n \tag{432}$$

   which means that the measurement noise, once sampled, is integrated in a deterministic manner because its behavior inside the integration interval is known.

3. From Probability Theory we know that the integration of continuous white Gaussian noise over a period Δt produces a discrete white Gaussian impulse $\mathbf{w}_n$ described by

   $$\mathbf{w}_n \triangleq \int_{n\Delta t}^{(n+1)\Delta t} \mathbf{w}(\tau)d\tau \quad , \quad \mathbf{w}_n \sim \mathcal{N}\{0, \mathbf{W}\} \quad , \quad \text{with } \mathbf{W} = \mathbf{W}^c\Delta t \tag{433}$$

   We obsereve that, contrary to the measurement noise just above, the perturbation does not have a deterministic behavior inside the integration interval, and hence it must be integrated stochastically.

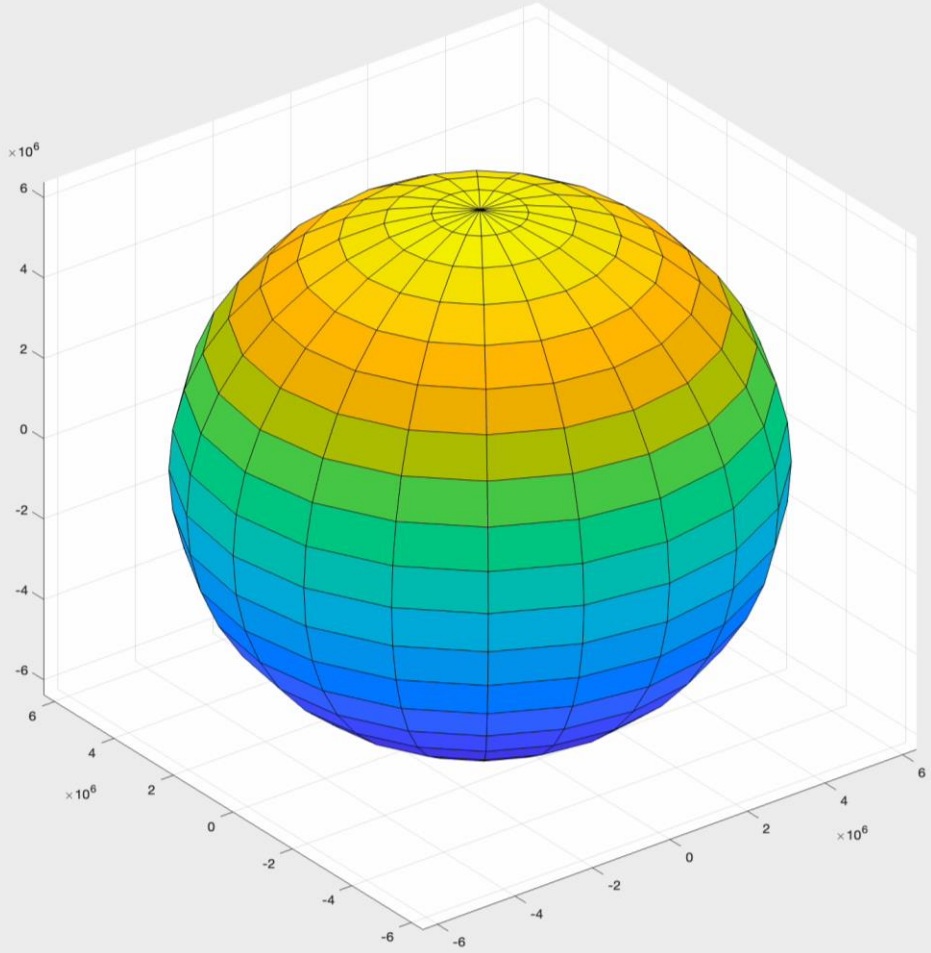


NAVIGATION

- ORBIT PROPAGATION
- ORBITAL MODES

Code_Final.m

```
42
43      arr1_v_mag = [];
44      arr2_ke = [];
45      arr3_g = [];
46
47      dt = 100;
48      total_time = 5e4;
49
50
51      overall = [];
52
53      for j = 1:dt:total_time
54          [a_g, a_J2] = acc(r(1),r(2),r(3),constants);
55          a_drag = drag(rho(norm(r), data_table), getNormals(), v, constants) / mass;
56          F_SRP = solarPressure(r(1), r(2), r(3), getTime(getJulian(j)), getNormals(), constants);
57          a_SRP = F_SRP / mass;
58          a = a_g + a_J2 + 0*a_drag + 0*a_SRP;
59          v = v + a*dt;
60
61          arr1_v_mag = [arr1_v_mag norm(v)];
62          arr2_ke = [arr2_ke mass*norm(v)^2/norm(r)];
63          arr3_g = [arr3_g a_g*mass];
64
65          r = r + v*dt;
66          pos = [pos r];
67
68          rho(norm(r), data_table);
69      end
70
71      [X,Y,Z] = sphere;
```

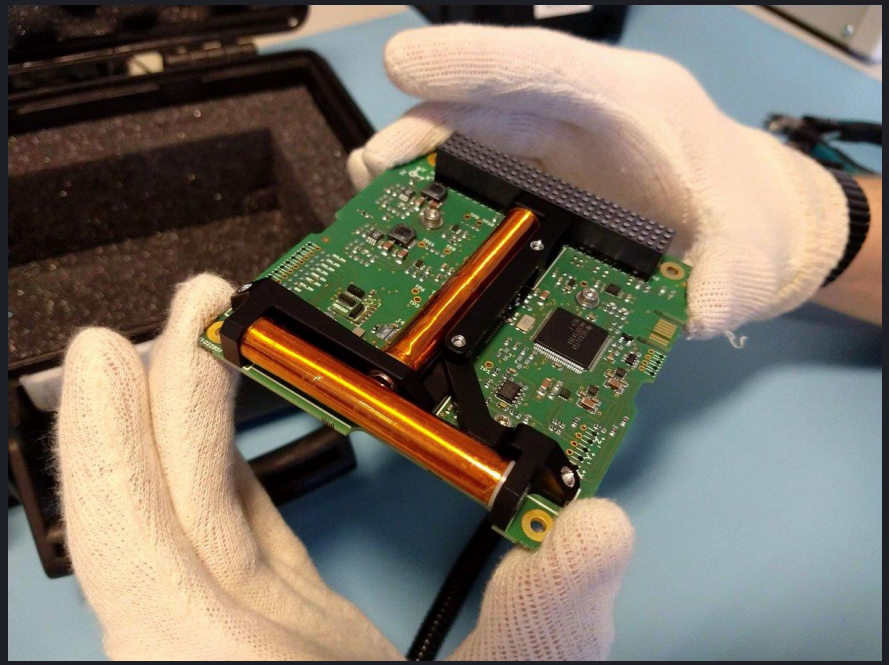

×10^6
6
4
2
0
-2
-4
-6
6
4
×10^6
2
0
-2
-4
-6
-6
-4
-2
0
2
4
×10^6
6

CONTROL SYSTEMS

REACTION WHEELS

MAGNETOTORQUERS

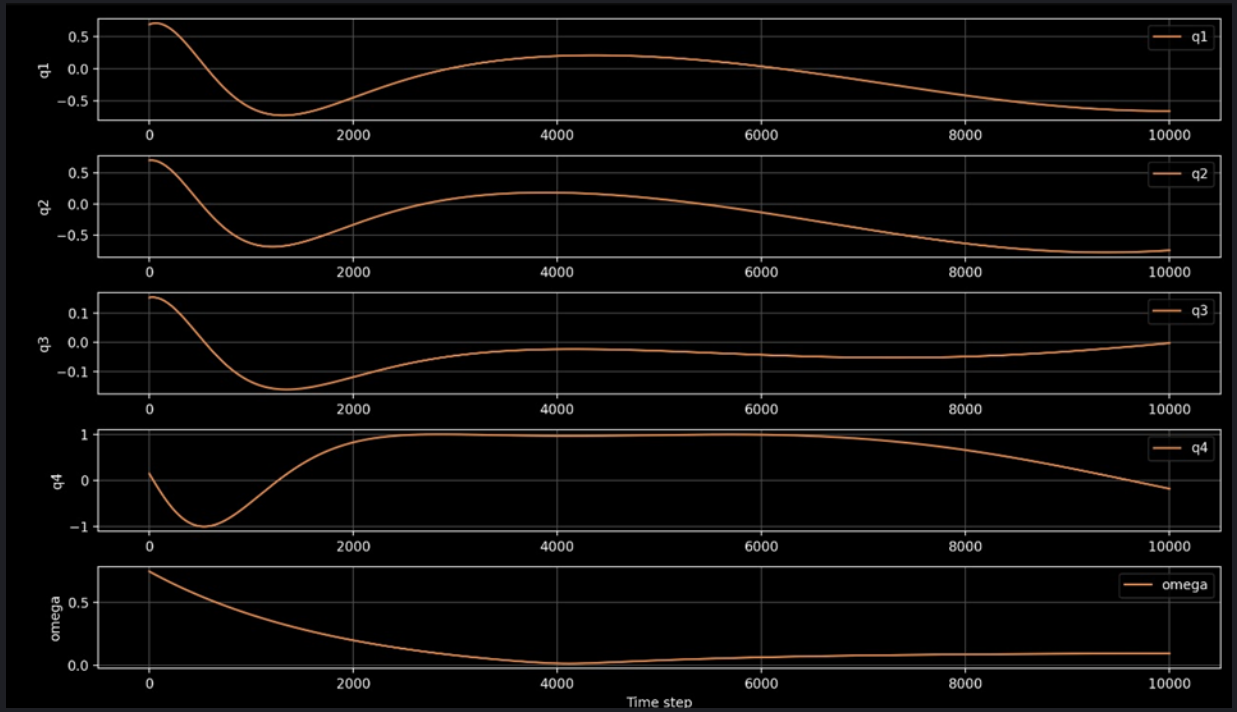
MAGNETORQUERS



REACTION WHEELS

REACTION WHEELS

CONTROL


q1
0.5
0.0
−0.5
0
2000
4000
6000
8000
10000
q1
q2
0.5
0.0
−0.5
0
2000
4000
6000
8000
10000
q2
q3
0.1
0.0
−0.1
0
2000
4000
6000
8000
10000
q3
q4
1
0
−1
0
2000
4000
6000
8000
10000
q4
omega
0.5
0.0
0
2000
4000
6000
8000
10000
omega
Time step

04.

TELEMETRY, TRACKING AND COMMAND

Telemetry Tracking & Command: WWWHAAATTT??!!!

T

T

C

Telemetry: Downlinking of Data from the satellite to the ground station.

Telemetry Tracking & Command: WWWHAAATTT??!!!

T

T

C

Tracking: Locating the position of the satellite in the orbit to ensure that the ground station antennas stay pointed towards the satellite.

Telemetry Tracking & Command: WWWHAAATTT??!!!

T

T

C

Command: Uplink data/instructions from the ground station to the satellite.

Forms of Data that we deal with:

Primary Data: The image. This is downlinked from the satellite to the ground station.

Forms of Data that we deal with:

Beacon: This signal is tracked, to locate the satellite in orbit and to ensure that the satellite is alive. Also provides basic housekeeping data about the satellite.

Forms of Data that we deal with:

Commands: These instructions are uplinked from the ground station to the satellite.

T

T

C

Software Defined Radios

T T C

Communication system that uses software for digital signal processing.

003-1040559 1250 003-77156.8 1760 0009-14563.7 73273

T

T

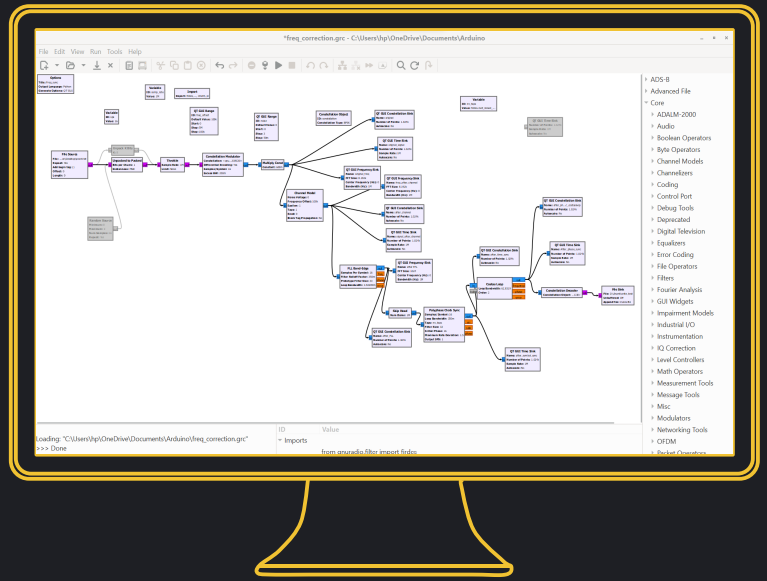
C

Communication system that uses software for digital signal processing.

003-1040559 1250 003-77156.8 1760 0009-14563.7 73273

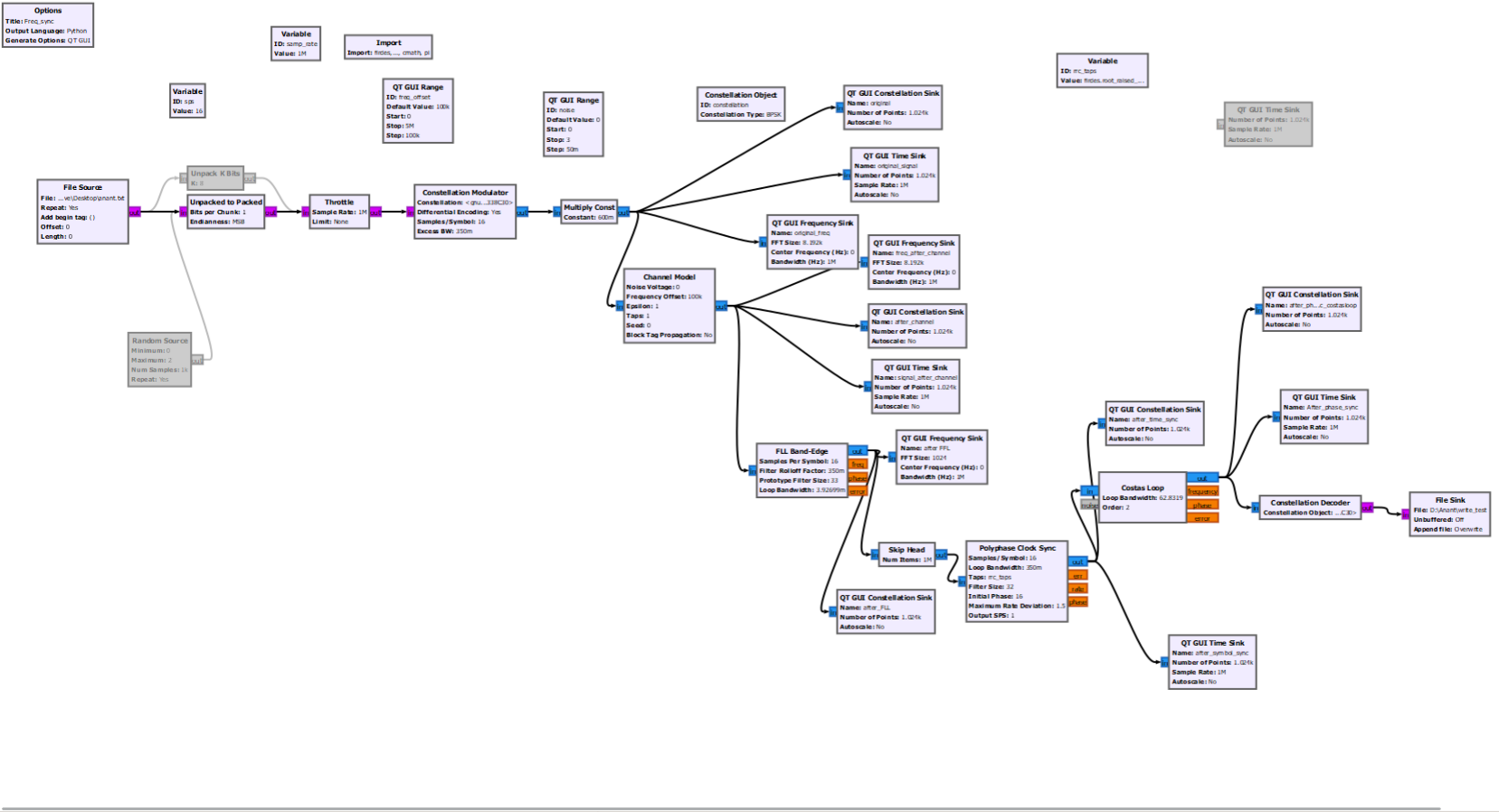
OK, but how do I use the SDR?

GNU Radio Companion is a visual tool for designing and simulating software-defined radios and signal processing systems using flowgraphs.



T T C

01 02 03 04 05 06



- ADS-B
- Advanced File
- Core
  - ADALM-2000
  - Audio
  - Boolean Operators
  - Byte Operators
  - Channel Models
  - Channelizers
  - Coding
  - Control Port
  - Debug Tools
  - Deprecated
  - Digital Television
  - Equalizers
  - Error Coding
  - File Operators
  - Filters
  - Fourier Analysis
  - GUI Widgets
  - Impairment Models
  - Industrial I/O
  - Instrumentation
  - IQ Correction
  - Level Controllers
  - Math Operators
  - Measurement Tools
  - Message Tools
  - Misc
  - Modulators
  - Networking Tools
  - OFDM

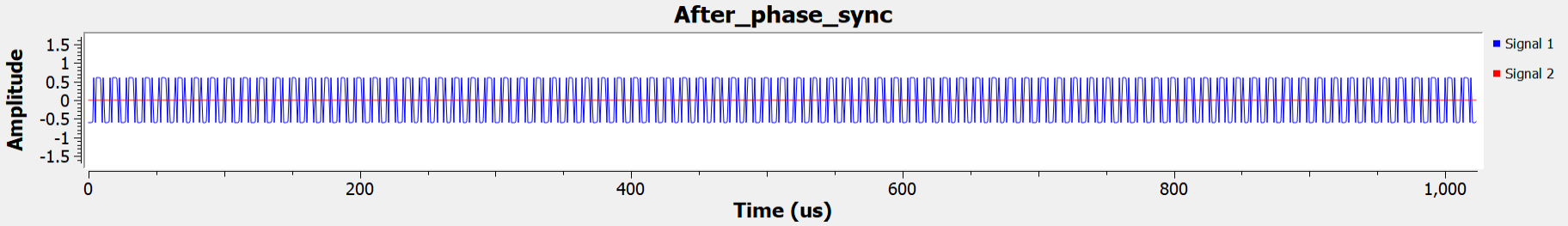
Loading: "C:\Users\hp\OneDrive\Documents\Arduino\freq_correction.grc"
>>> Done

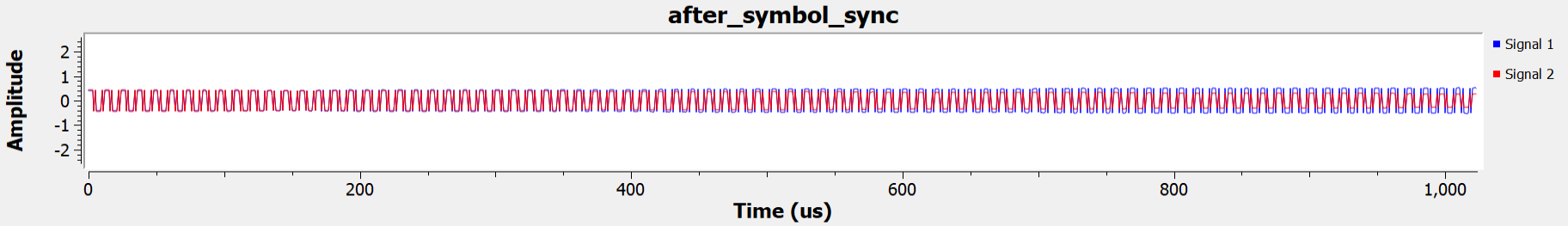
ID	Value
Imports	
	from gnuradio.filter import firdes

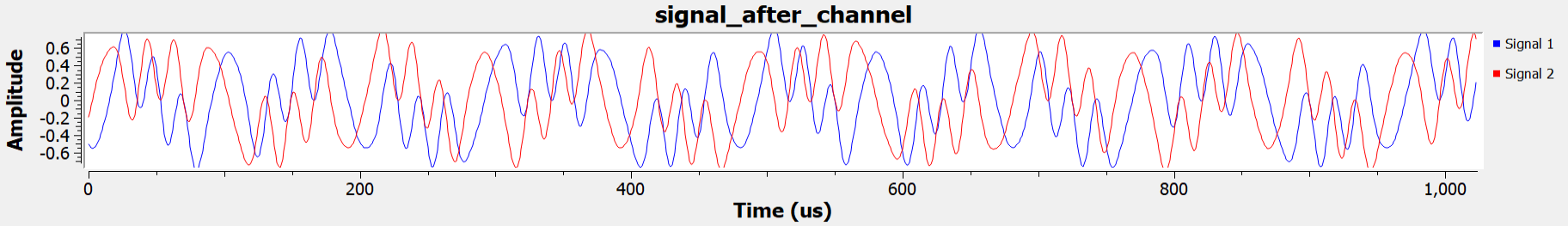
Freq_sync

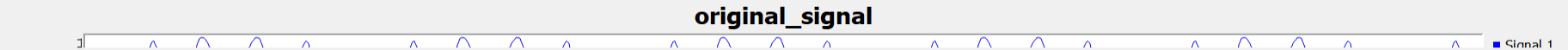
'noise' 0.0000

'freq_offset' 200000



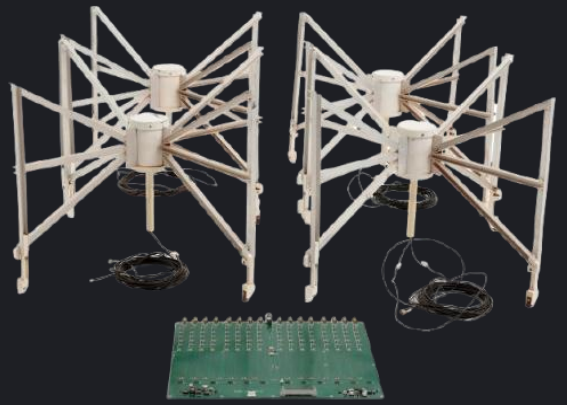






Antennas?!

A little (sometimes large) metal thingy that sends EM waves to a far away place.

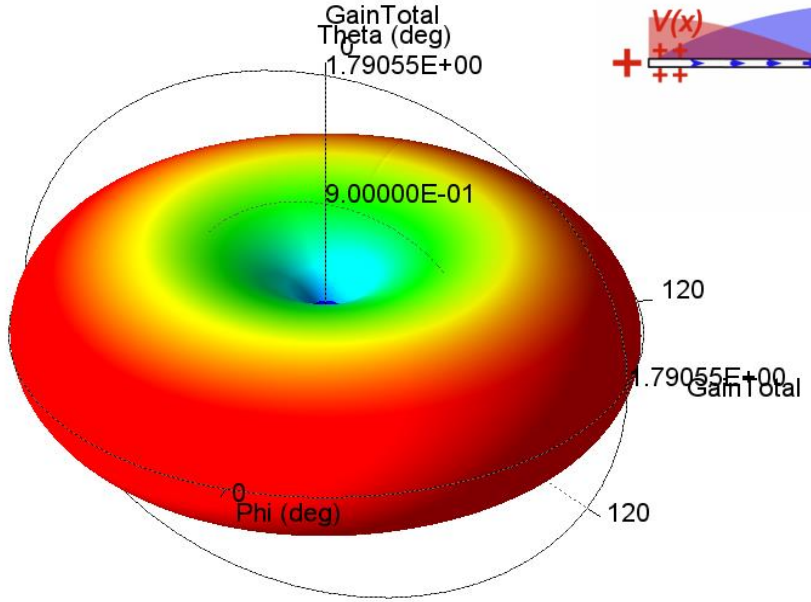


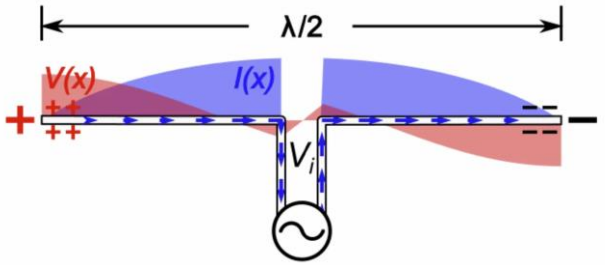
Ansys Inc.

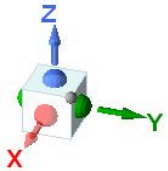
Gain Plot 1

Ansys
2023 R2
STUDENT

Max: 1.79
1.80000E+00
1.60000E+00
1.40000E+00
1.20000E+00
1.00000E+00
8.00000E-01
6.00000E-01
4.00000E-01
2.00000E-01
0.00000E+00
Min: 0.00







SIMULATED GAIN PLOT FOR A DIPOLE ANTENNA OPERATING AT 2.4 GHz

Show 3 Messages
Show Progress

Ansys Twin Builder Student 2024 R2 - patch_array - array - 3D Modeler - SOLVED - [patch_array - array - Modeler]

File Edit View Project Draw Modeler HFSS Tools Window Help

Save | Cut Copy Paste | Undo Redo Delete | Select: Object | Select by Name | Zoom | Pan Rotate Orient | Fit All Fit Selected | Unite Subtract Intersect | Split Imprint | Fillet Chamfer | Measure Ruler Units | Grid XY 3D | Model vacuum Material

Desktop View Draw Model Simulation Results Automation Ansys Minerva Learning and Support

- Model
  - Solids
    - Duroid (tm)
      - sub
    - vacuum
      - RadiatingSurface
  - Sheets
    - Lumped Port
      - port1
    - Perfect E
      - antenna
      - Ground
- Coordinate Systems
  - Global
- Planes
  - Global:XY
  - Global:XZ
  - Global:YZ
- Lists
  - AllObjects

Ansys 2024 R2 STUDENT

[3, 1] [2, 1] [3, 2] [1, 1] [2, 2] [3, 3] [1, 2] [2, 3] [1, 3]

Z Y X

A Vector

A Vector

SIMULATION FOR A 3x3 PATCH ANTENNA ARRAY OPERATING AT 10 GHz

0 3.5 7 (cm)

Nothing is selected

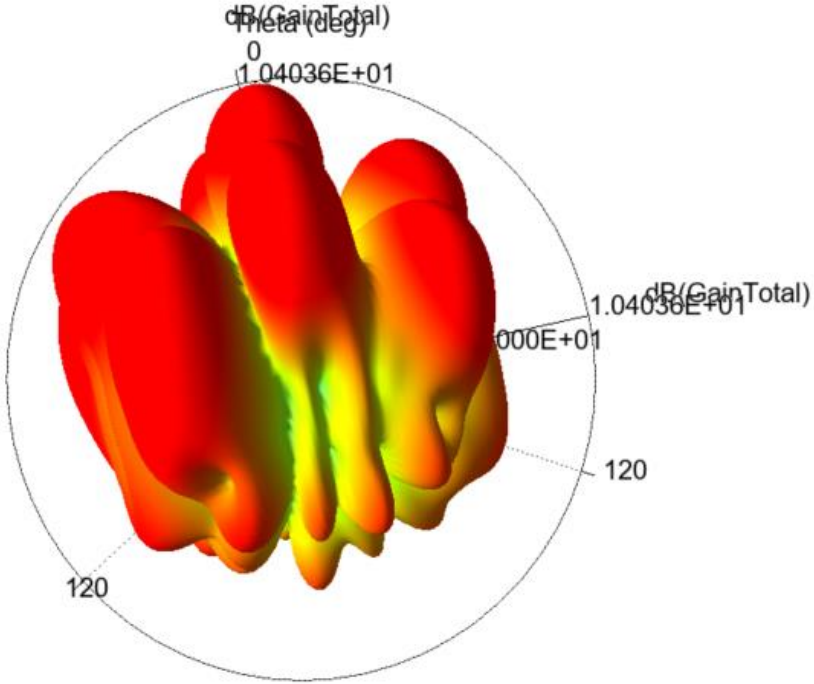
Show 1 Messages Show Progress

Ansys Inc.

Gain Plot 6

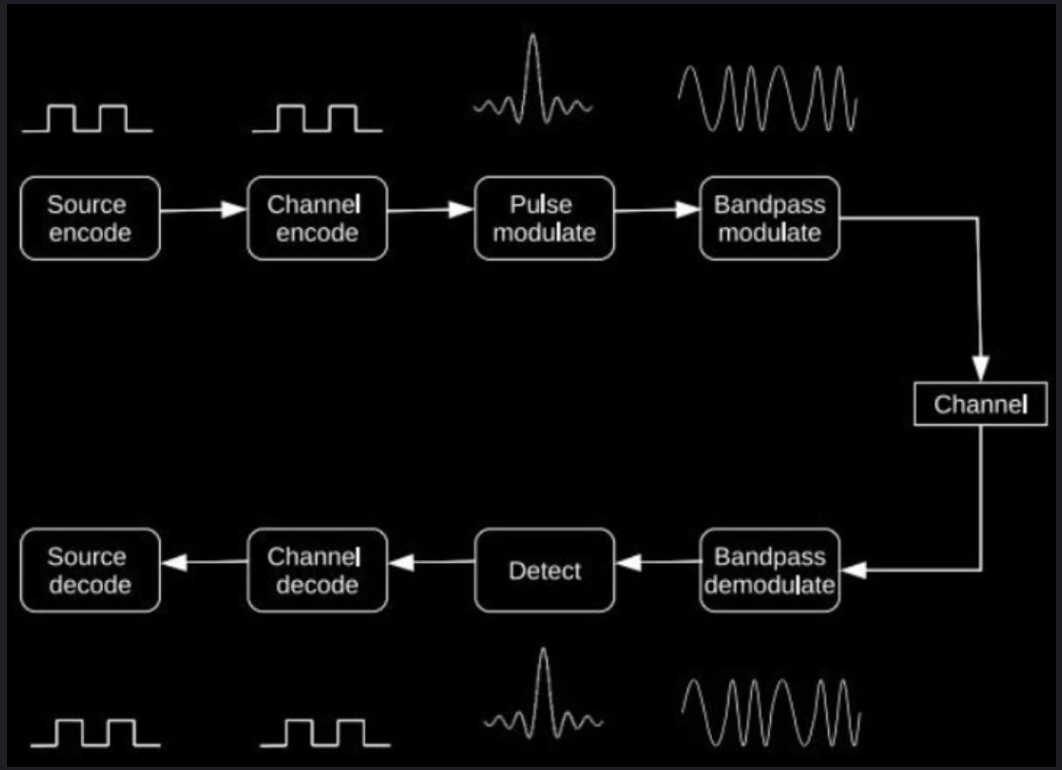
Ansys
2024 R2
STUDENT

Max: 10.40
12.5
0.0
-12.5
-25.0
-37.5
Min: -32.62

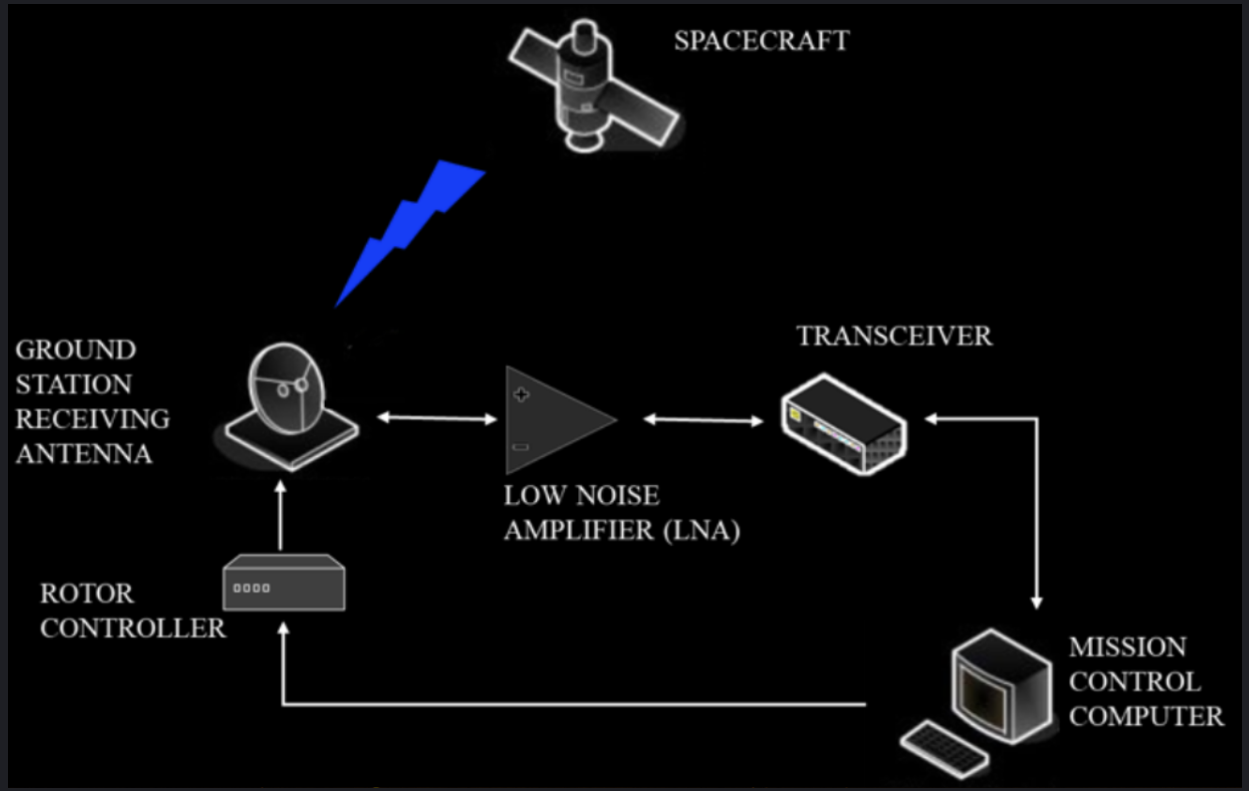


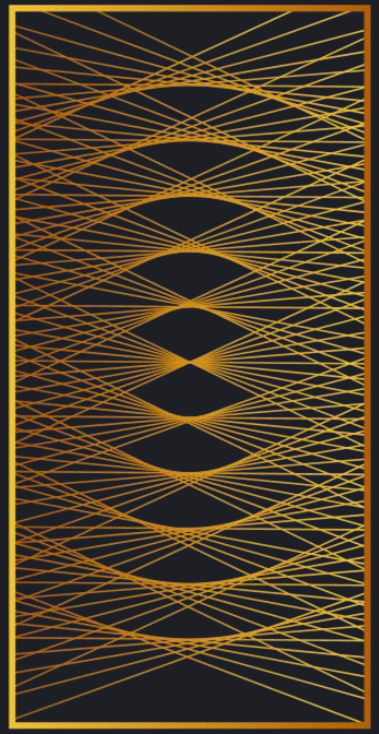
IT'S GAIN PLOT

Telemetry Tracking & Command: Digital Communication



Telemetry Tracking & Command: Ground Station



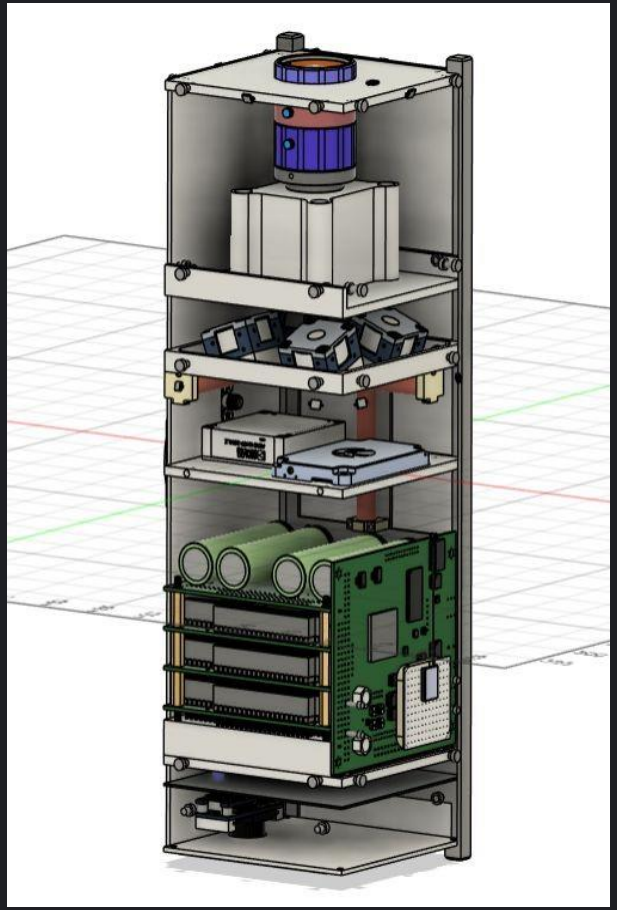


05.

STRUCTURAL AND THERMAL SUBSYSTEM

STRUCTURE

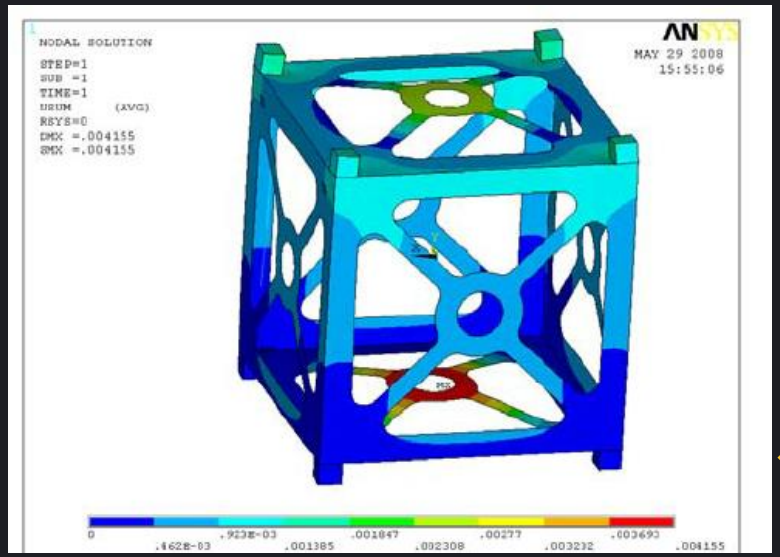
As part of the structure we need to design something that will survive launch loads, while providing an easily accessible data and power bus for debugging and assembly of components.



STRUCTURAL ANALYSIS

Static Loads

Quasi Static Loads



Random Vibrations

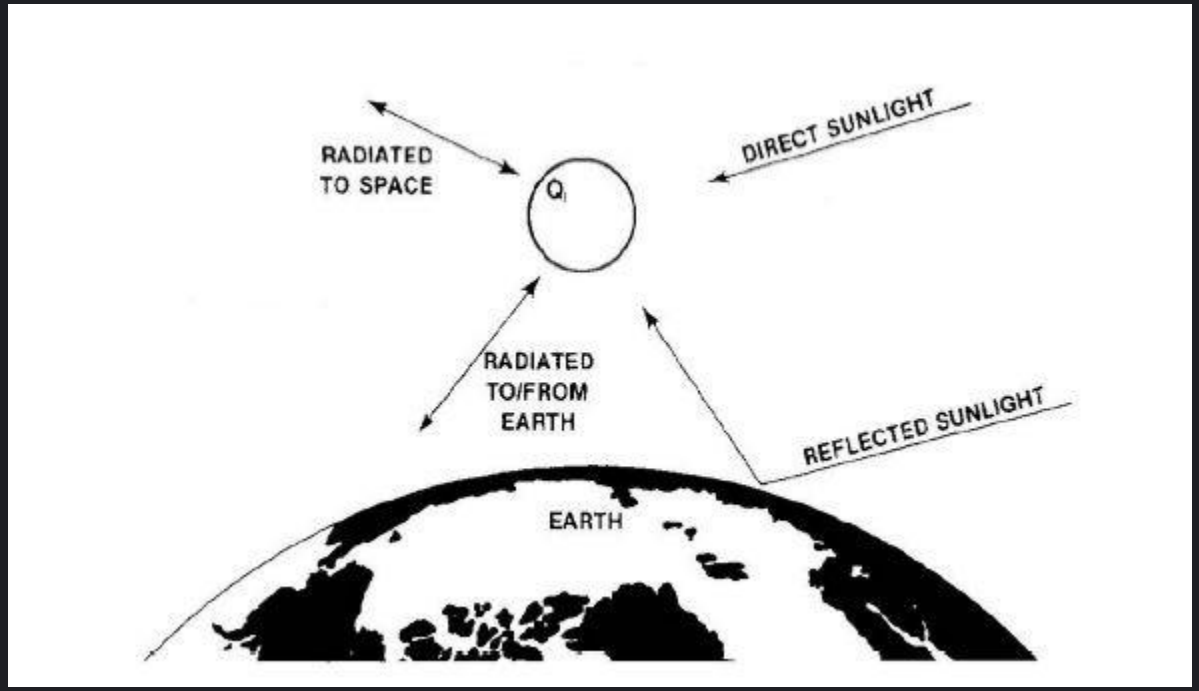
Sinusoidal Vibrations

003-1040559 1250 003-77156.8 1760 0009-14563.7 73273

IMPORTANCE OF THERMAL ANALYSIS

003-1040559 1250 003-77156.8 1760 0009-14563.7 73273

IMPORTANCE OF THERMAL ANALYSIS



SIGNIFICANCE OF THERMAL ANALYSIS

- Maintaining the temperatures for each component of the satellite is important for the safety of the components.
- Orbital parameters like different angle values, vectors and values like Ltan (local time at ascending node), influence the thermal loads that the CubeSAT experiences.

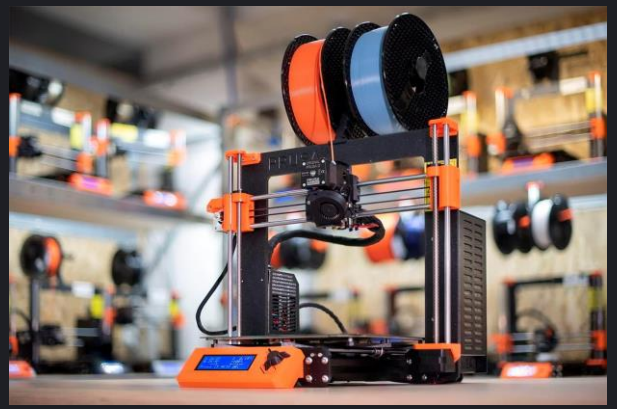
Component	Operating Temperature (degrees Celsius)
Batteries	10 to 45
Solar Panels	-25 to 50
Geomagnetic Sensors	-40 to 85
PCBs	-40 to 85

MANUFACTURING

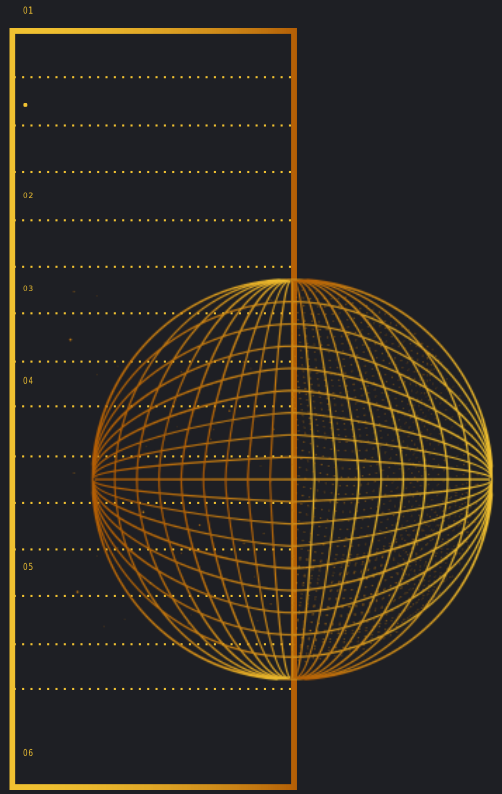
In STS Manufacturing is the ultimate constraint to a design.

You can have the most magnificent design but if it can't be made it isn't as useful as it can be.

We use both additive manufacturing as well as classical methods of manufacturing to make our prototypes for subsystems to use.



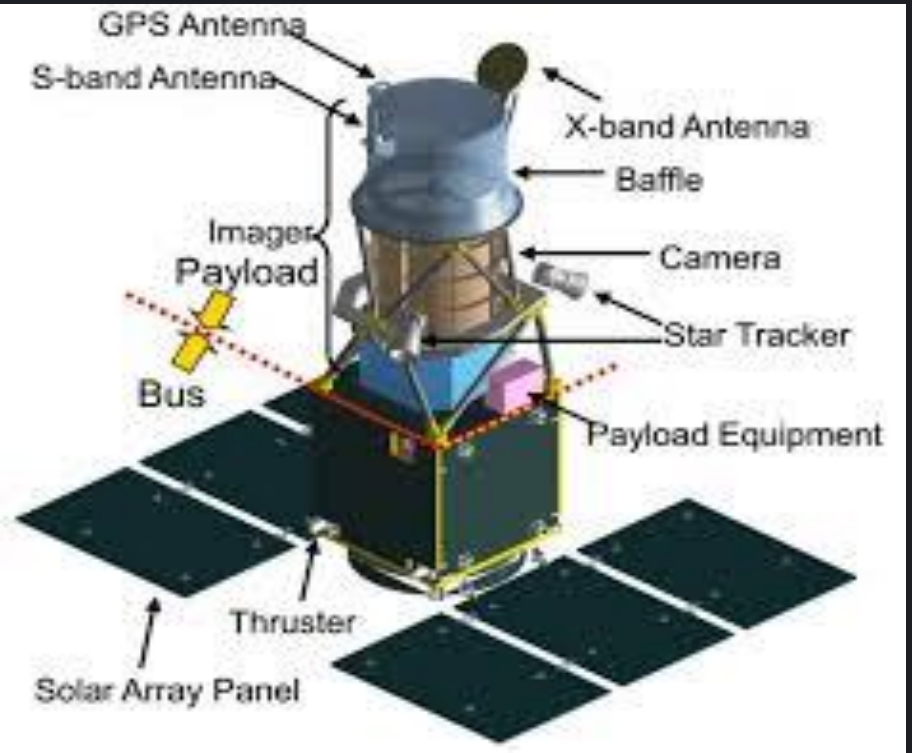
003-1040559 1250 003-77156.8 1760 0009-14563.7 73273



06.

PAYLOAD

PAYLOAD



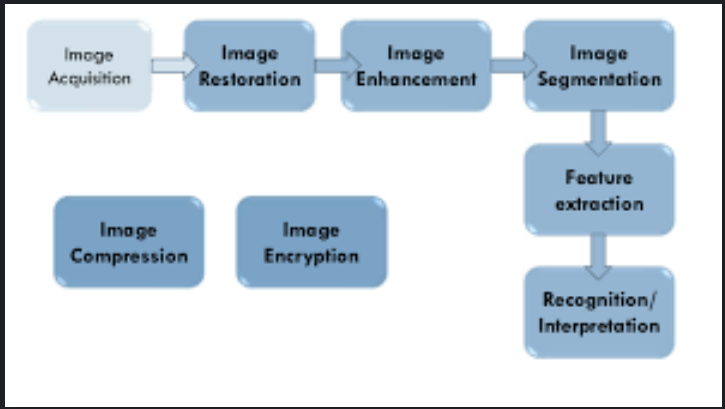
A scientific or technological instrument on a satellite that's designed to achieve the satellite's mission

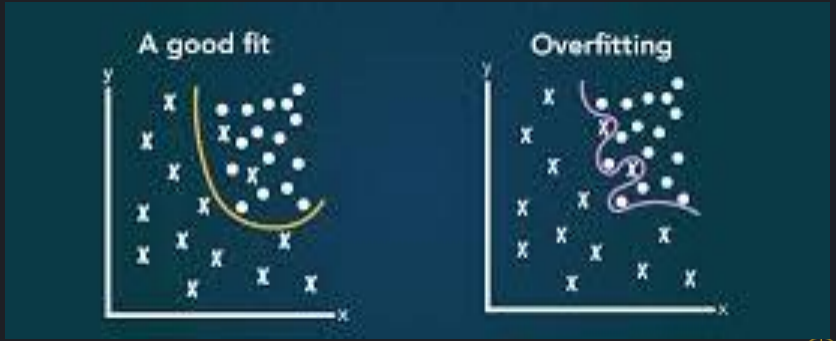
SKILLS REQUIRED

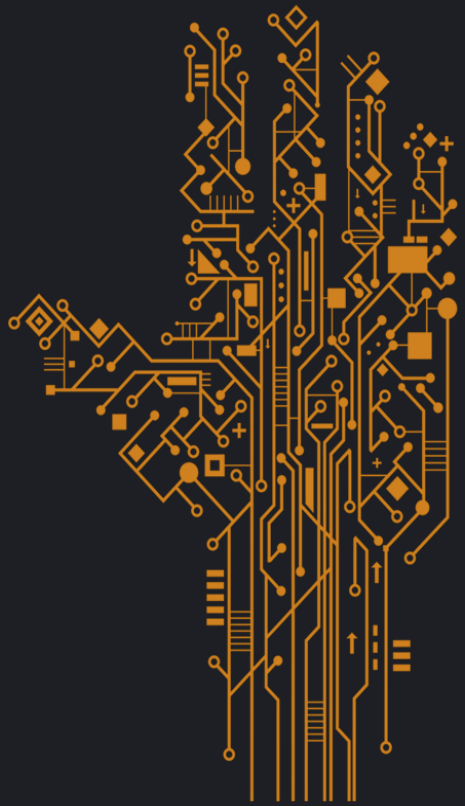
MACHINE LEARNING & DEEP LEARNING

IMAGE PROCESSING

WILL TO LEARN!!!







07.

ELECTRICAL POWER SYSTEM

01 02 03 04 05 06

01 02 03 04 05 06

ABOUT EPS

The Electrical & Power Subsystem (EPS) is one of the essential subsystems for the CubeSat since it handles **power generation**, **energy storage**, and **power distribution** to all other subsystems.

Therefore, the design of EPS becomes crucial for successful CubeSat mission, wherein the first step is the selection of EPS architecture.

DEVELOPMENT of advanced electronics have enabled building of CubeSats with powerful capabilities that were previously dominated by larger satellites.

CORE FUNCTIONS OF EPS

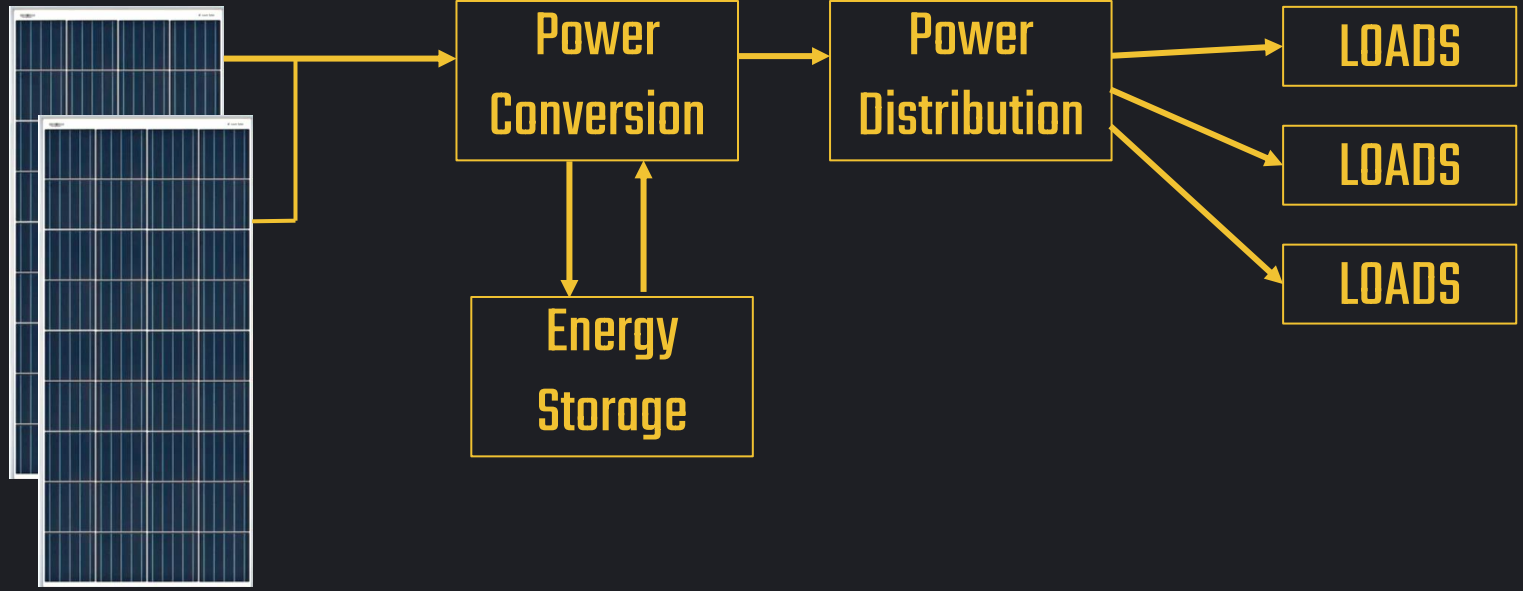
POWER SOURCE

ENERGY STORAGE

POWER DISTRIBUTION

POWER REGULATION AND CONTROL

Basic Components of CubeSAT EPS



POWER STORAGE

Why do we need Power Storage ?

As the CubeSat revolves in an orbit, it experiences both sunlit and eclipse periods.

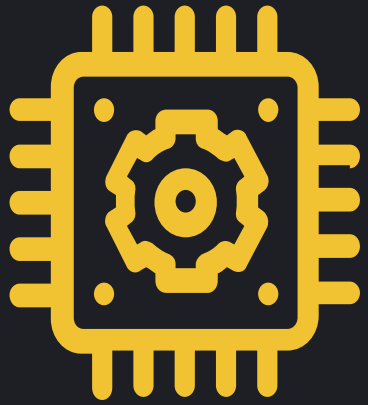
In **Sunlit Period**, the CubeSat is illuminated by sunlight and the angle of incidence of sunlight on the photovoltaic (PV) arrays varies from 66.55 to 90 degrees.

During **Eclipse Period**, the Earth blocks the sunlight from illuminating the CubeSat once in every orbit in all seasons and the temperature drops sharply. The eclipse period is dependent on the orbit altitude, inclination, and the sunlight incidence angle on the orbit plane.

The storage system is required during **eclipse** period and **peak load conditions**.

The energy storage system consists of lithium-ion (Li-ion) cells due to higher energy density, higher number of charge/discharge cycles, and lower self-discharge rate.

On the other hand, the **Li-ion batteries** have following **disadvantages**:
(1) the life-cycle of Li-ion battery degrades due to higher charge/discharge rates [23]
(2) Li-ion battery has limited energy density at lower temperatures (<-20oC)
(3) Charge rate of Li-ion battery is very low at lower temperatures with the possibility of life reduction.

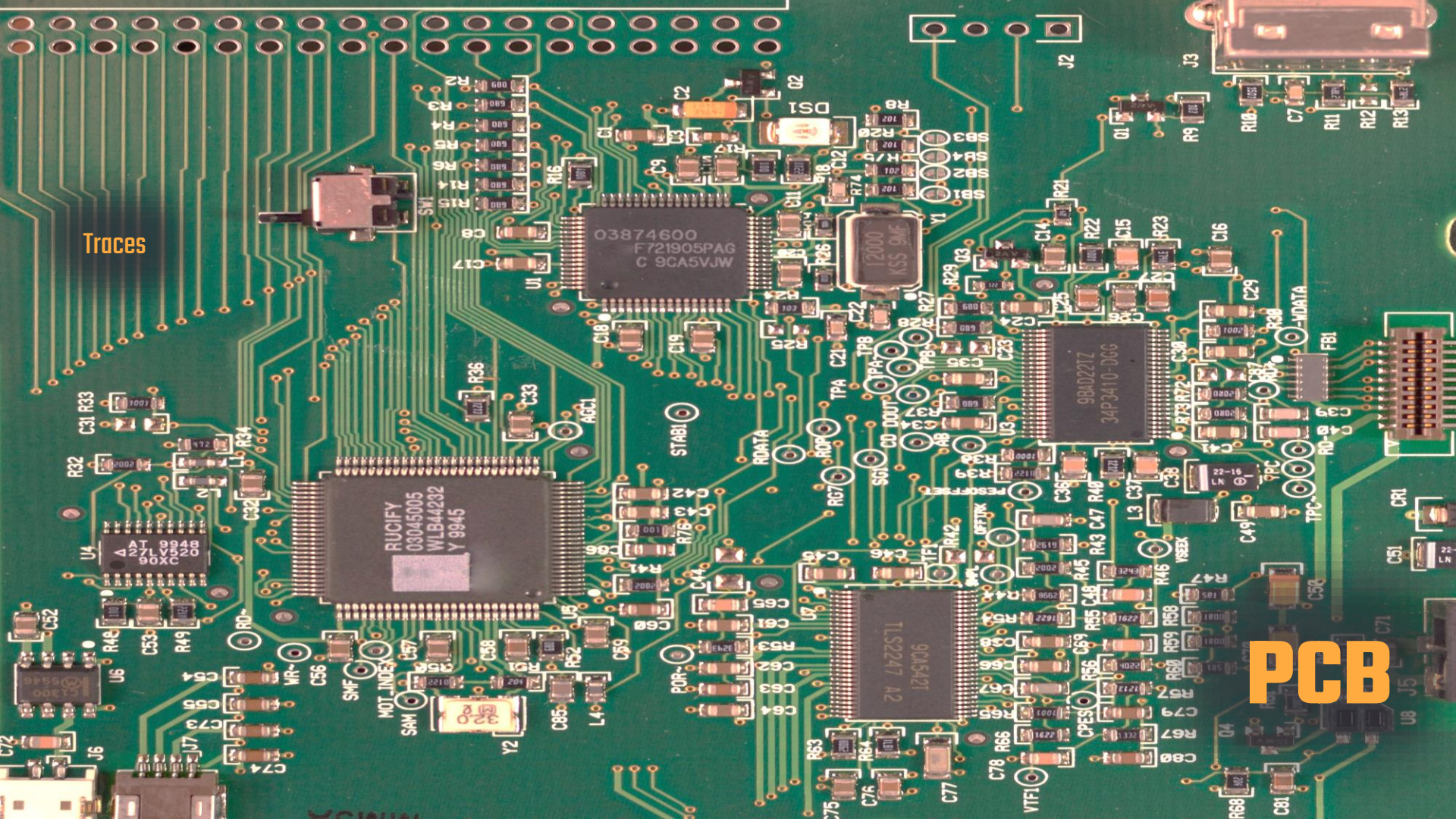


Printed Circuit Boards

Power Distribution

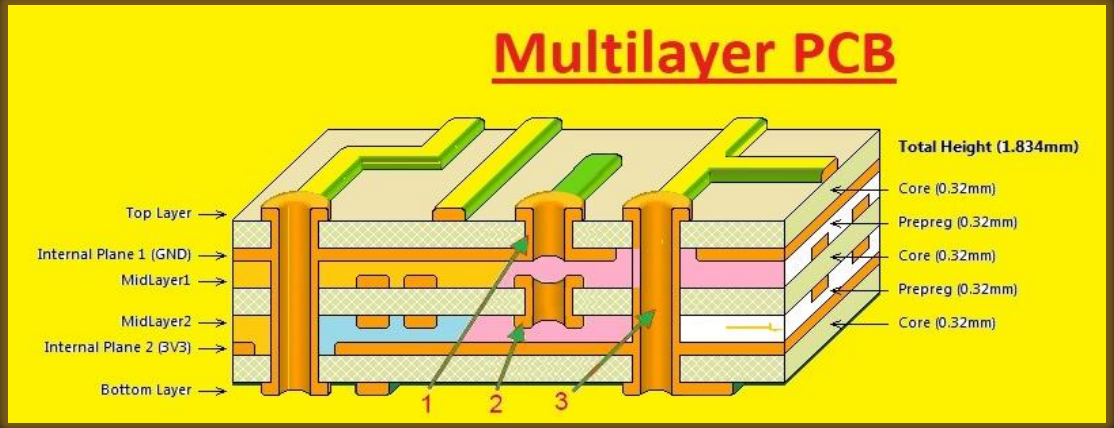
01
02
03
04
05
06

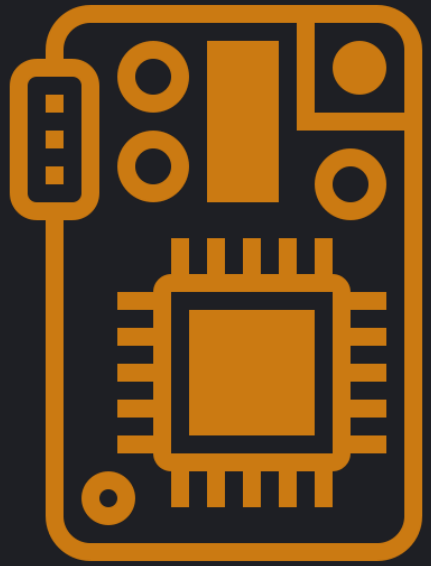
01 02 03 04 05 06


Traces
PCB

A **Printed circuit board (PCB)** is an electronic assembly that uses copper conductors to create electrical connections between components.

Printed circuit boards provide mechanical support for electronic components so that a device can be mounted in an enclosure. A printed circuit board design must include a specific set of steps that aligns with the manufacturing process, integrated circuit packaging, and the structure of the bare circuit board.



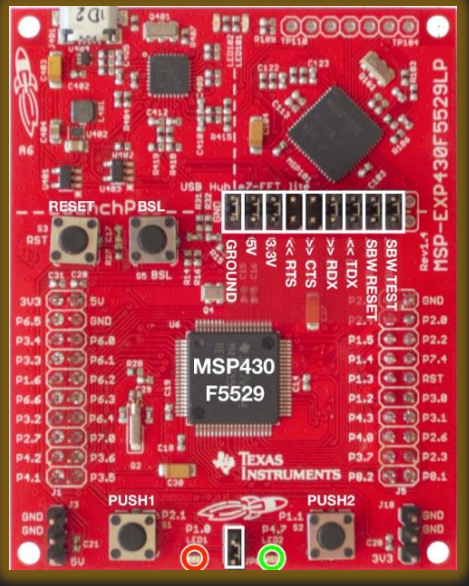


EPS Microcontroller

MSP430F5529

01 02 03 04 05 06

01 02 03 04 05 06



MSP430F5529

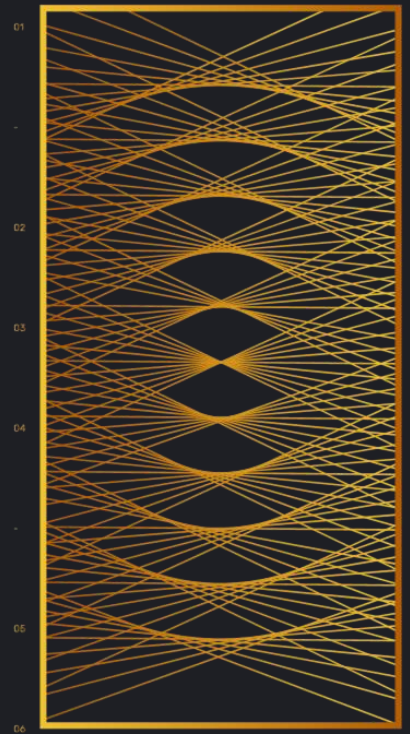
The MSP has a storage capacity of only 128 kilobytes and consists all of the **software** of the EPS subsystem.

The software used in the satellite is variation of the commonly used language ‘C’ and it’s called ‘**embedded C**’.

```
# This function adds two numbers
def add(x, y):
    return x + y

# This function subtracts two numbers
def subtract(x, y):
    return x - y

# This function multiplies two numbers
def multiply(x, y):
    return x * y

# This function divides two numbers
def divide(x, y):
    return x / y


print("Select operation.")
print("1.Add")
print("2.Subtract")
print("3.Multiply")
print("4.Divide")

while True:
    # take input from the user
    choice = input("Enter choice(1/2/3/4): ")

    # check if choice is one of the four options
    if choice in ('1', '2', '3', '4'):
        try:
            num1 = float(input("Enter first number: "))
            num2 = float(input("Enter second number: "))
        except ValueError:
            print("Invalid input. Please enter a number.")
            continue

        if choice == '1':
            print(num1, "+", num2, "=", add(num1, num2))
```

A Simple Python Calculator is 3 KB

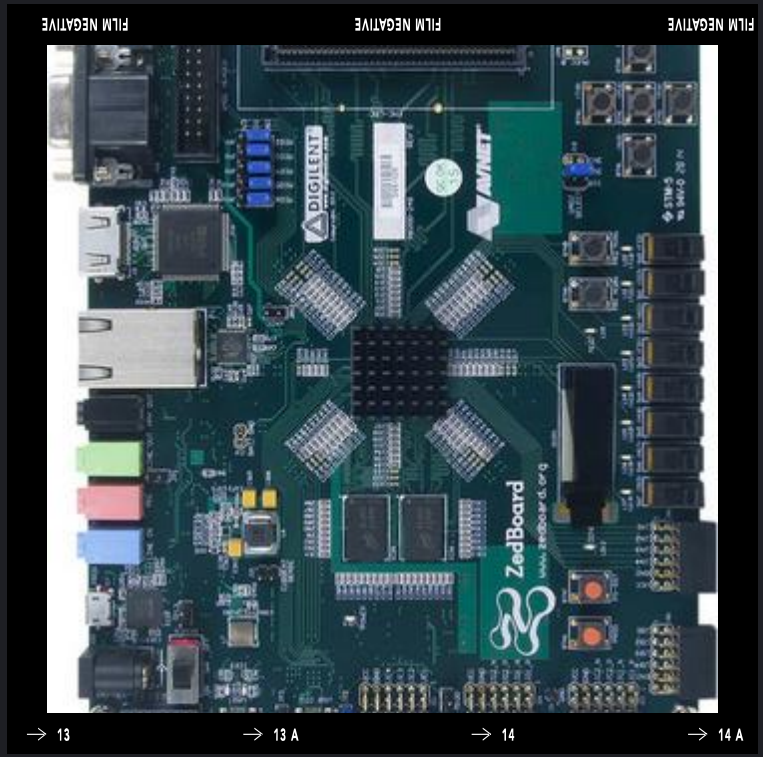
01 02 03 04 05 06



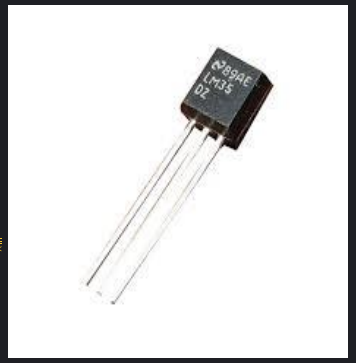
08.

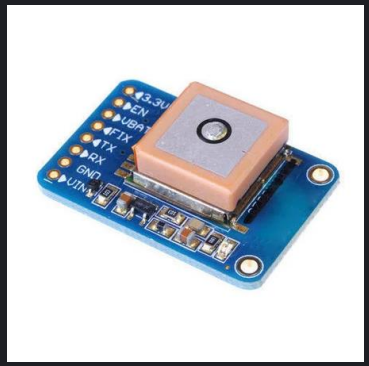
ON BOARD COMPUTING

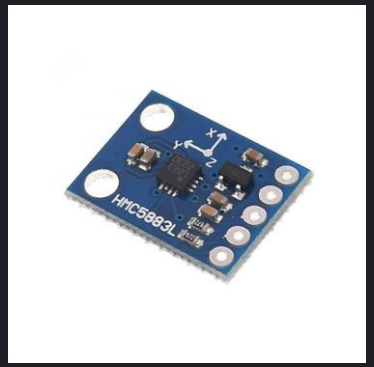
MOTHERBOARD OF LAPTOP



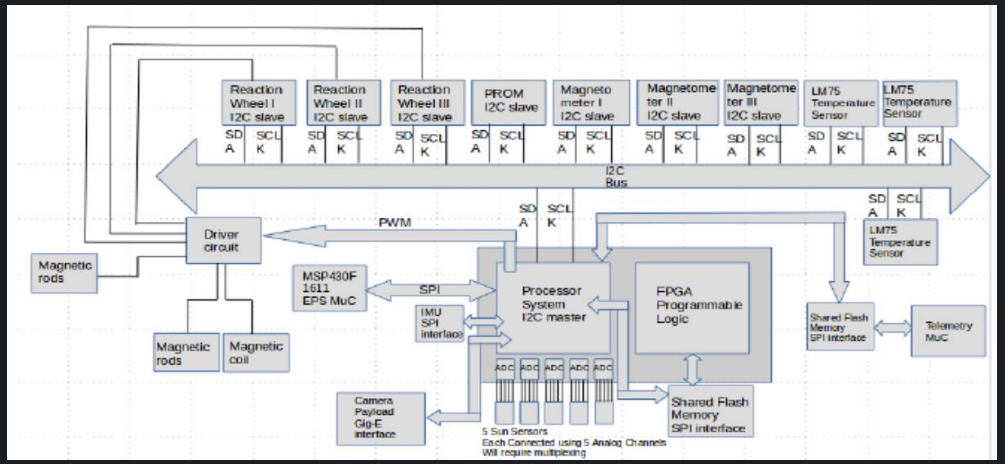
MOTHERBOARD OF CUBESAT

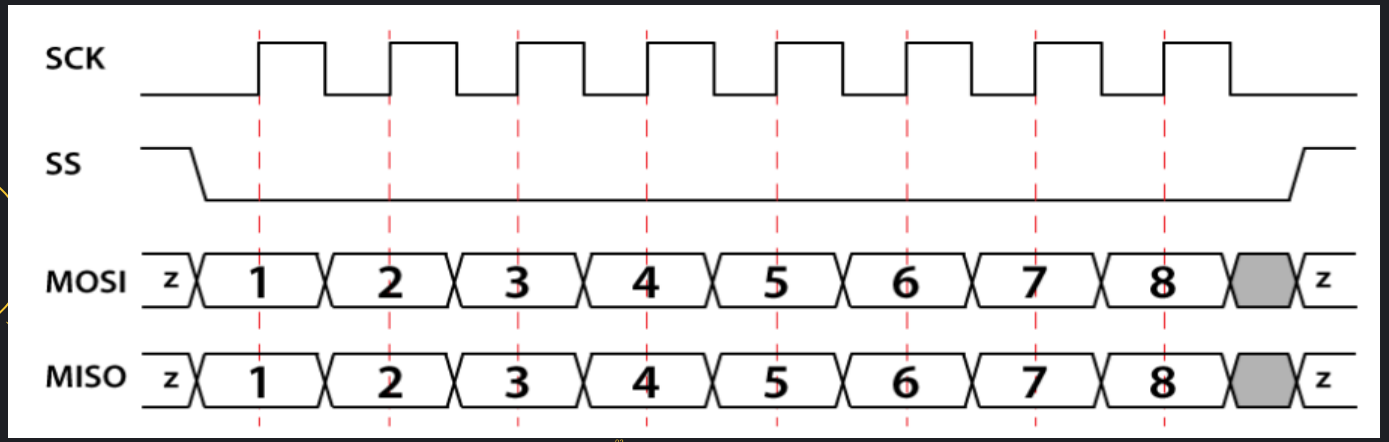




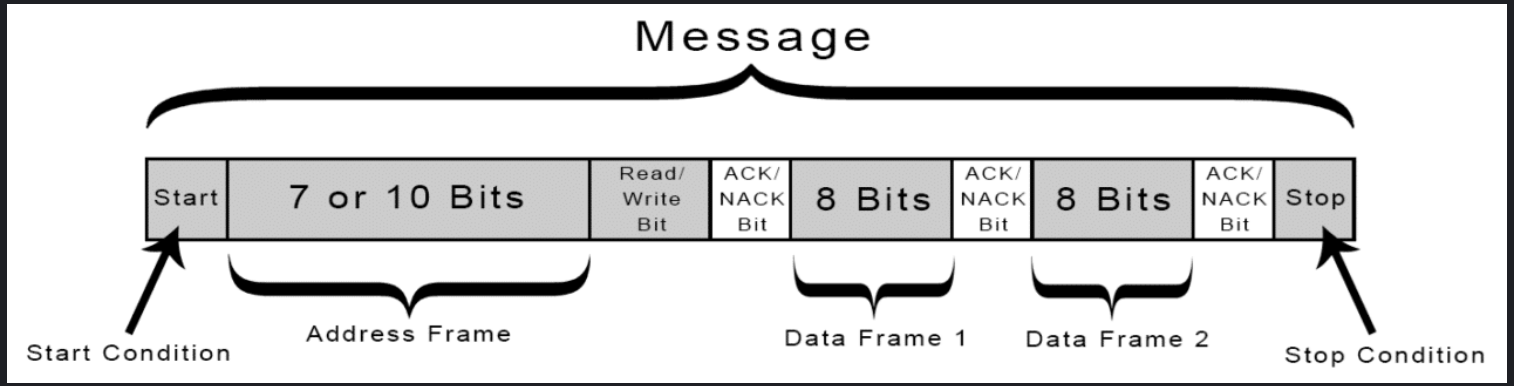


SENSOR INTERFACING





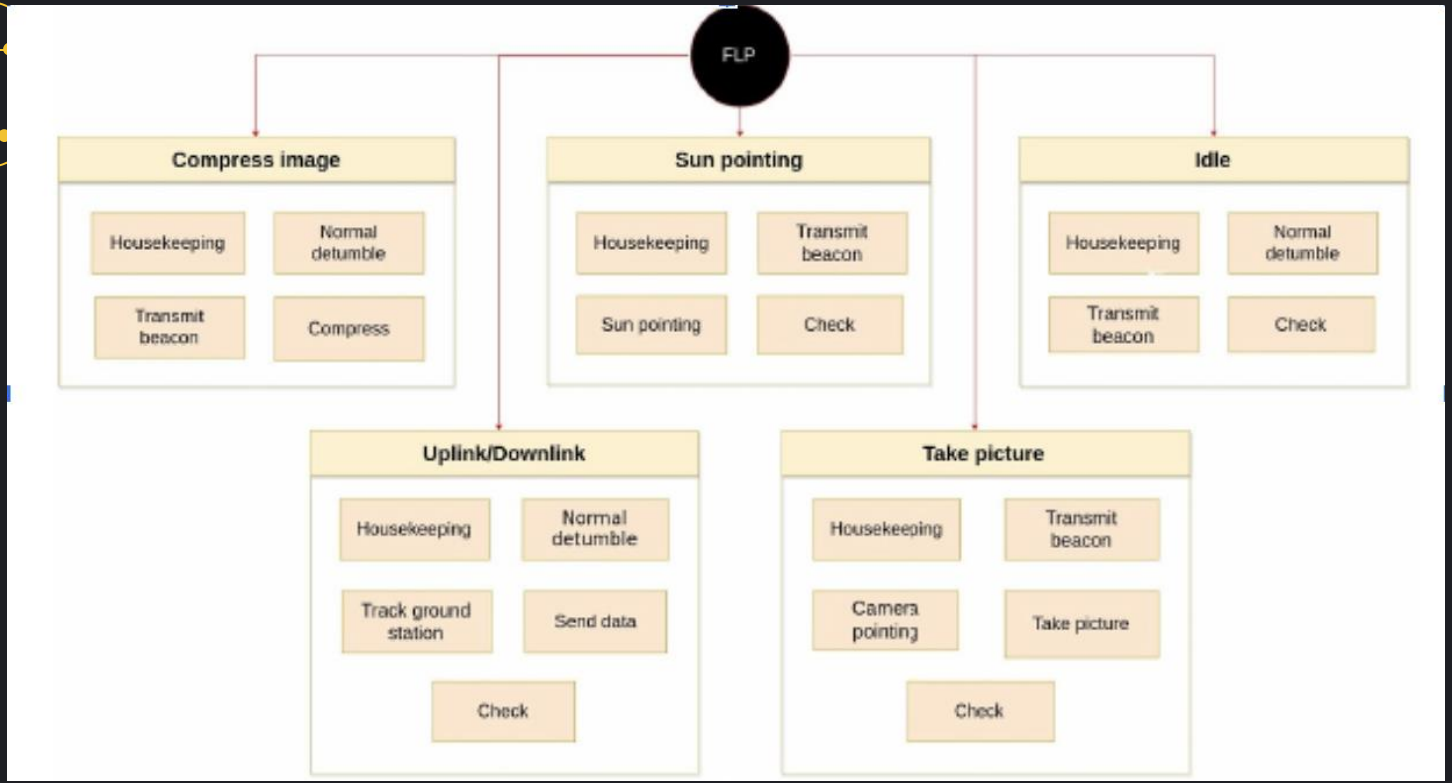
COMMUNICATION PROTOCOLS



```
DISTRO ⟼ Garuda Linux x86_64
  ⟼ Linux 6.12.8-zen1-1-zen
  ⟼ 2016 (pacman)[stable]
  ⟼ fish 3.7.1
DE/WM ⟼ Hyprland 0.46.2 (Wayland)
  ⟼ BeautyLine [Qt], BeautyLine [GTK2/3/4]
  ⟼ Sweet (24px)
  ⟼ JetbrainsMonoNL (14pt)
  ⟼ foot 1.20.1
SYSTEM ⟼ VivoBook_ASUSLaptop K3502ZA_K3502ZA (1.0)
  ⟼ 12th Gen Intel(R) Core(TM) i5-12500H (16) @ 4.50 GHz
  ⟼ Iris Xe Graphics
  ⟼ 1920x1080 @ 60Hz
  ⟼ 4.95 GiB / 15.24 GiB (32%)
  ⟼ 2.50 GiB / 32.01 GiB (8%)
  ⟼ 1 day, 3 hours, 52 mins
  ⟼ 1920x1080 @ 60Hz
AUDIO ⟼ Alder Lake PCH-P High Definition Audio Controller Speaker
```

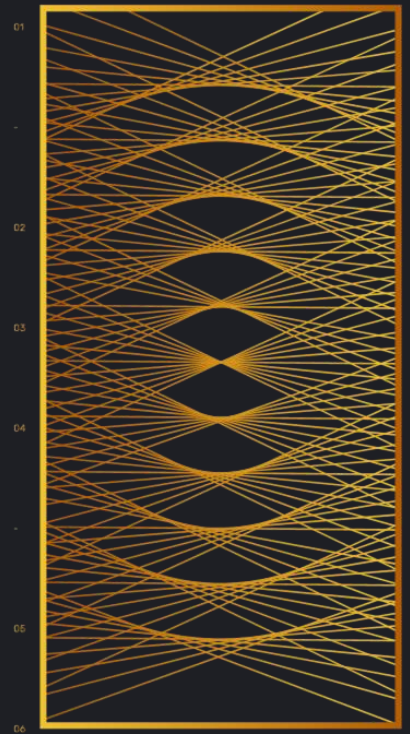
OPERATING SYSTEM - LINUX

FLIGHT PLAN


FLP
Compress image
Housekeeping
Normal detumble
Transmit beacon
Compress
Sun pointing
Housekeeping
Transmit beacon
Sun pointing
Check
Idle
Housekeeping
Normal detumble
Transmit beacon
Check
Uplink/Downlink
Housekeeping
Normal detumble
Track ground station
Send data
Check
Take picture
Housekeeping
Transmit beacon
Camera pointing
Take picture
Check

SO BASICALLY, WE

003-1040559 1250 003-77156.8 1760 0009-14563.7 73273



09.

PUBLICITY SPONSORSHIP AND DESIGN

VERTICALS

WEB DEVELOPMENT 01

02 UI/UX & GRAPHIC DESIGNING

003-1040559 1250 003-77156.8 1760 0009-14563.7 73273

UI/UX & GRAPHIC DESIGNING

The UI/UX & Graphic Designing Vertical of Team Anant is dedicated to creating visually stunning and user-centric experiences. We are a team of creative minds who blend art and technology to design intuitive interfaces and captivating visuals.

We work on maintaining Team Anant's online presence by creating engaging posts as you all might have seen through our official handles on LinkedIn and Instagram. We also design Team Anant's merch.

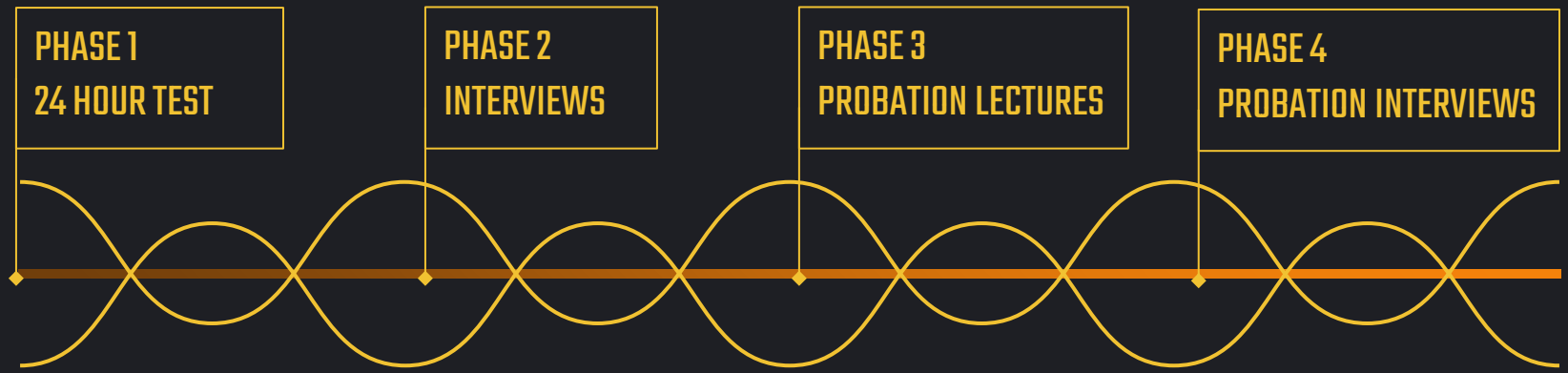
Joining the team will equip you with skills like Figma and Adobe Photoshop. Have a happy learning experience!

003-1040559 1250 003-77156.8 1760 0009-14563.7 73273

WEB-DEVELOPMENT

The Web Development Vertical of Team Anant is at the forefront of crafting the online presence of the team. We are a passionate group of developers who translate creative ideas into stunning, user-friendly websites. We are responsible for the development, and ongoing maintenance of the Team Anant's online platform. We're currently working on revamping the website. This will provide you with a great opportunity to learn frameworks such as React as well as HTML, CSS and JavaScript.

RECRUITMENT PROCESS



PHASE 1
24 HOUR TEST

A 24 hour long, open internet, open ChatGPT test, as a test of your dedication

PHASE 2
INTERVIEWS

Technical and Non-Technical interviews

PHASE 3
PROBATION LECTURES

A deeper dive into each of the technical subsystems

PHASE 4
PROBATION INTERVIEWS

A final set of tasks and interviews to evaluate your understanding of the team

01 02 03 04 05 06

SIGN UP!

FOR FURTHER DETAILS

Join our WhatsApp Group to stay updated

Check out our website