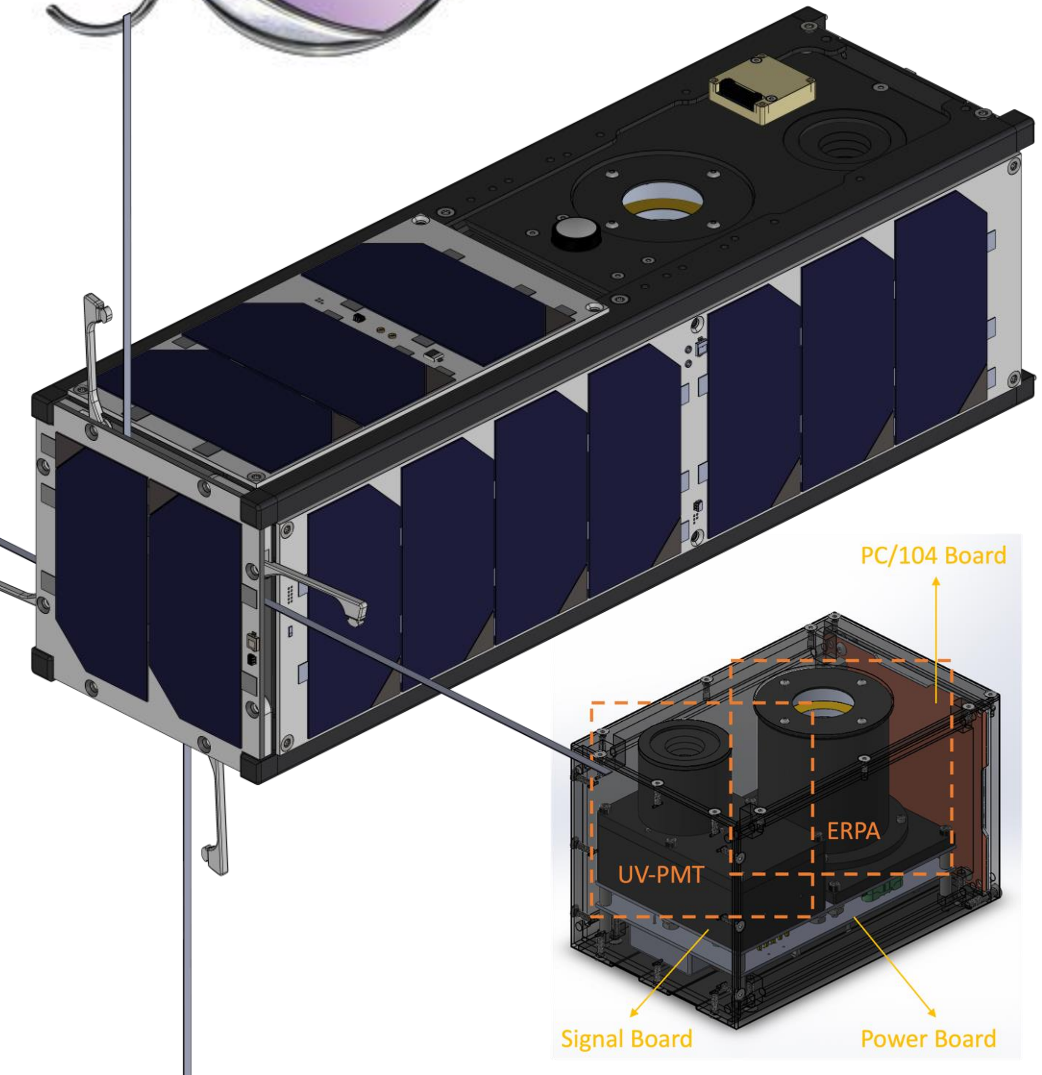
# 3UCubed: IMAP Student Collaboration CubeSat Software

Presenters: Jack Schneider, Anthony Santos, Jayme Brannan

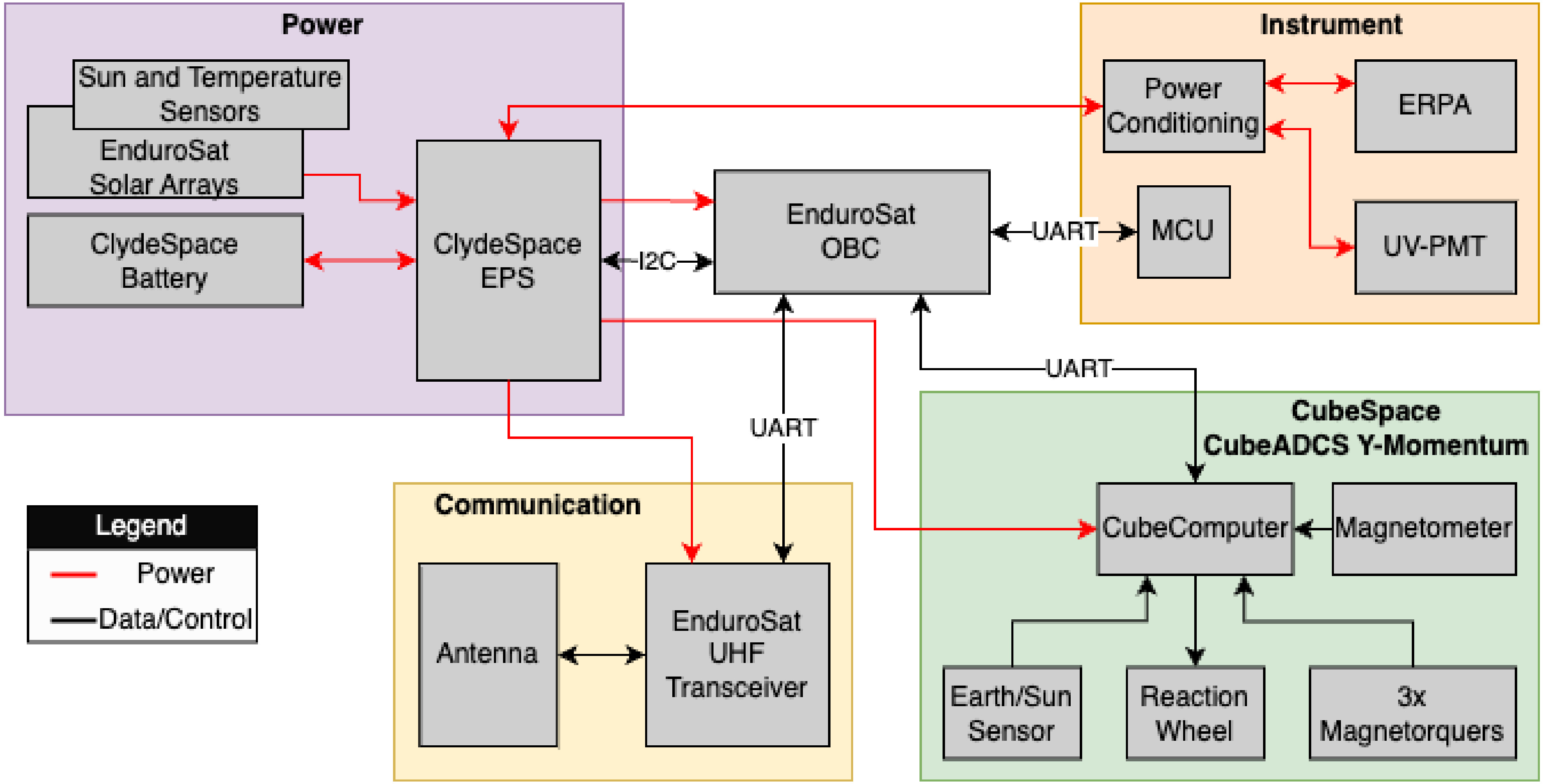
Contributors: Damian Fowler, Jared Morisson, Shane Woods, Michaela Power, Jared King, Sanjeev Mehta, 3UCubed Team

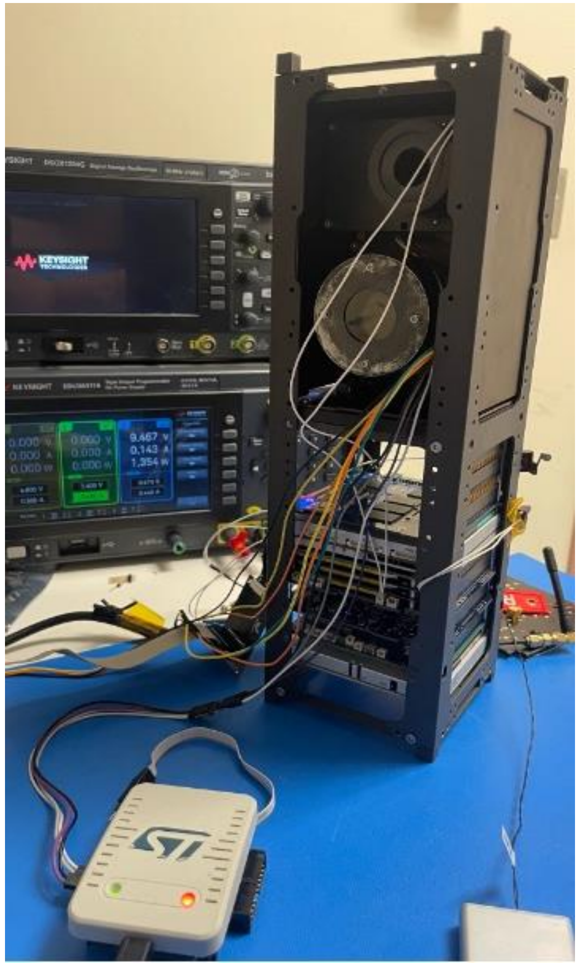
# Introduction



- The IMAP Student Collaboration goals are to build and launch a 3U CubeSat, augment IMAP science return, develop hands-on research experience for students.
- Goal: to understand how Earth's thermosphere responds to particle precipitation and solar wind forcing and internal magnetospheric processes.
- 3UCubed: 3 Universities; 3 CubeSats; Upwelling, Uplifting Undergraduates
- Three universities: Sonoma State University, Howard University, University of New Hampshire
- Manifested for launch on Transporter-15
- Spacecraft
  - Two instruments, combined size ≤ 1.5U
  - COTS subsystems for S/C bus

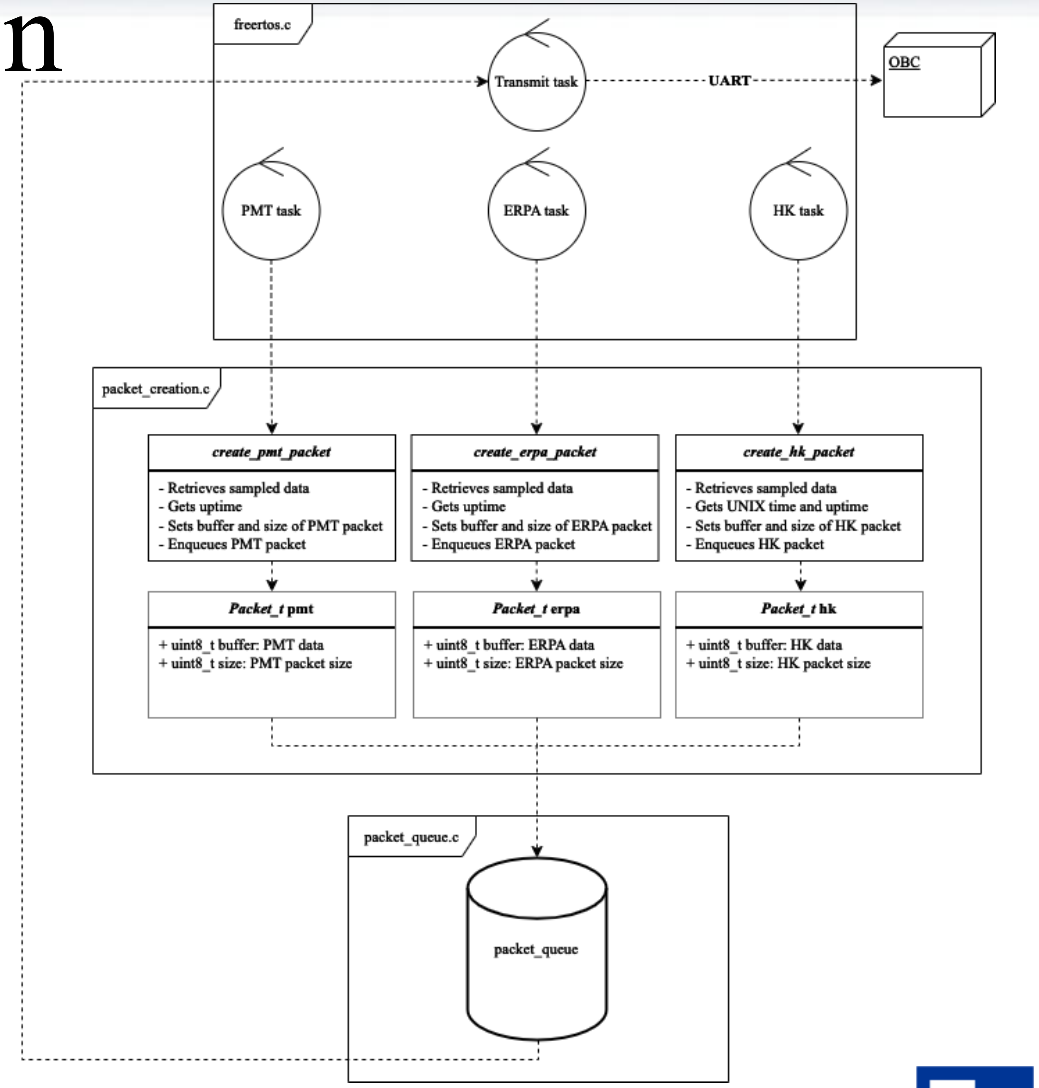
# Subsystem Overview





# Instrument Design

- Two Science Instruments:
  - Ultraviolet-Photo-Multiplier-Tube (UV-PMT)
  - Electron Retarding Potential Analyzer (ERPA)
- Instrument Microcontroller Unit (iMCU):
  - STM32 microcontroller
  - Collects analog data using onboard ADCs
  - Packetizes the data into telemetry formats
  - Sends raw packetized data to the Onboard Computer (OBC) over UART



***create_pmt_packet***

- Retrieves sampled data
- Gets uptime
- Sets buffer and size of PMT packet
- Enqueues PMT packet

***create_erpa_packet***

- Retrieves sampled data
- Gets uptime
- Sets buffer and size of ERPA packet
- Enqueues ERPA packet

***create_hk_packet***

- Retrieves sampled data
- Gets UNIX time and uptime
- Sets buffer and size of HK packet
- Enqueues HK packet

***Packet_t* pmt**

+ uint8_t buffer: PMT data
+ uint8_t size: PMT packet size

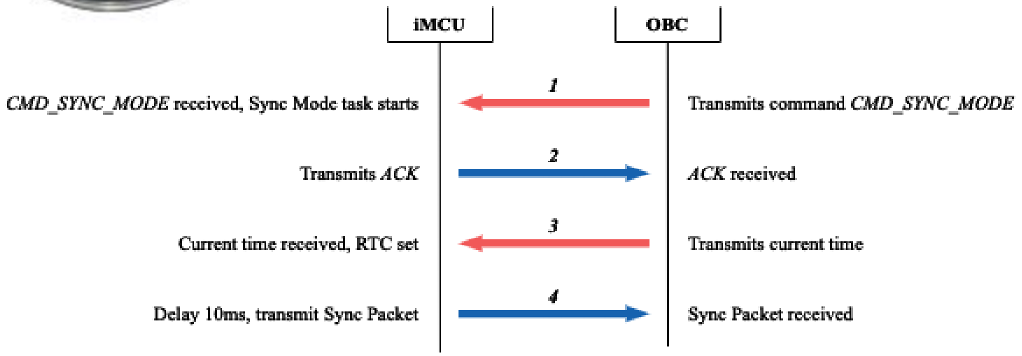
***Packet_t* erpa**

+ uint8_t buffer: ERPA data
+ uint8_t size: ERPA packet size

***Packet_t* hk**

+ uint8_t buffer: HK data
+ uint8_t size: HK packet size

# Instrument Operation



iMCU

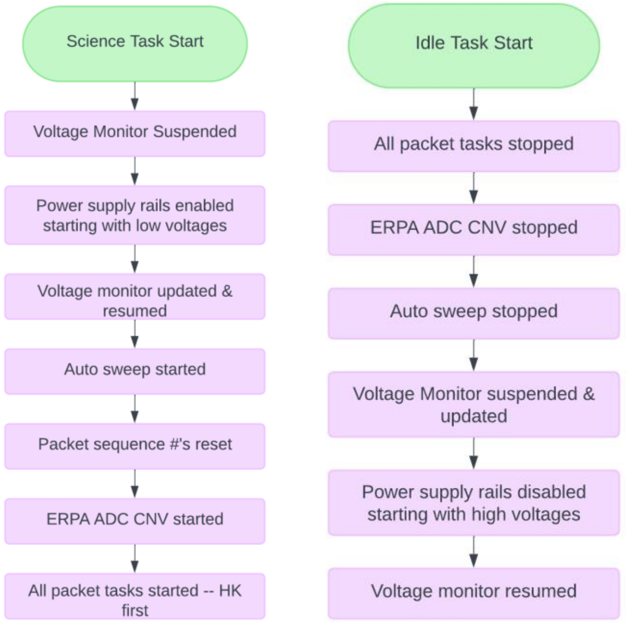
OBC

1 — *CMD_SYNC_MODE* received, Sync Mode task starts ← Transmits command *CMD_SYNC_MODE*

2 — Transmits *ACK* → *ACK* received

3 — Current time received, RTC set ← Transmits current time

4 — Delay 10ms, transmit Sync Packet → Sync Packet received



Science Task Start

Voltage Monitor Suspended

Power supply rails enabled starting with low voltages

Voltage monitor updated & resumed

Auto sweep started

Packet sequence #'s reset

ERPA ADC CNV started

All packet tasks started -- HK first

Idle Task Start

All packet tasks stopped

ERPA ADC CNV stopped

Auto sweep stopped

Voltage Monitor suspended & updated

Power supply rails disabled starting with high voltages

Voltage monitor resumed

- Operation Modes Triggered by Onboard Computer
  - Sync Mode
    - Syncs instruments' RTC with OBC
    - Sends version info + error counters to OBC
  - Science Mode
    - Turn on and monitor power rails
    - Analog data collection from sensors
    - Sends Housekeeping, ERPA, and PMT packets
  - Idle Mode
    - Powers down sensors and voltage rails
    - Halts data collection
  - Low-power (Sleep) Mode
    - A low power state that halts the MCU
    - Wakes on OBC command

# Instrument–OBC Interface

- Instrument and OBC communicate over UART
- Commands from the OBC are single-byte instructions
- Instrument data split into and sent as binary packets
  - Each ERPA packet: 14 Bytes
    - Sent every 3.125ms
  - Each UV-PMT packet: 10 Bytes
    - Sent every 62.5ms
  - Each Housekeeping packet (HK): 48 Bytes
    - Sent to OBC every 100ms

**ERPA** Packet

| SYNC | UPTIME | SEQ | SWEEP STEP | SWEEP MON | ERPA ADC |
|---|---|---|---|---|---|
| Fixed 2-byte hexadecimal | 4-byte uptime in ms | 3-byte packet counter | 1-byte sweep step | 2-byte sweep monitor | 2-byte ERPA ADC reading |

**UV-PMT** Packet Structure

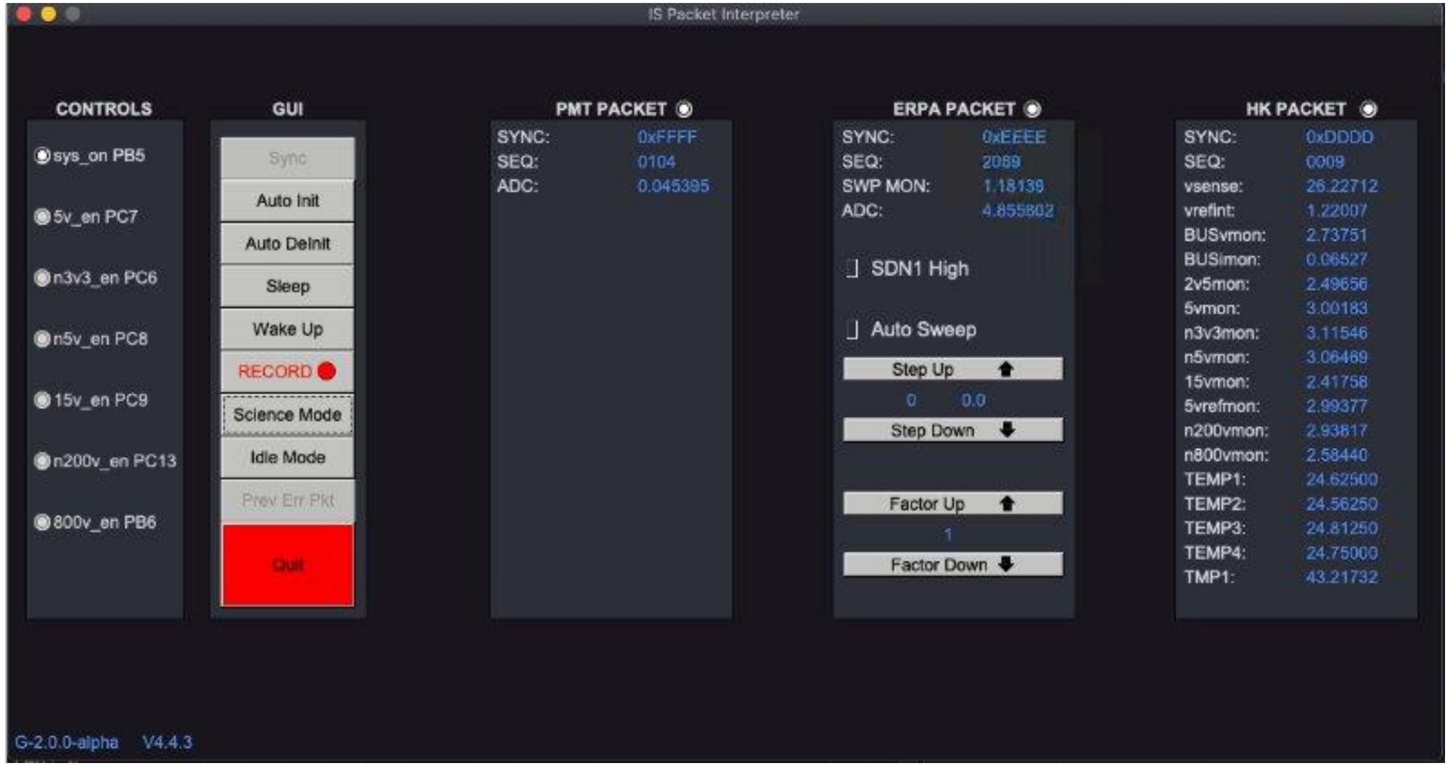
| SYNC | UPTIME | SEQ | PMT ADC |
|---|---|---|---|
| Fixed 2-byte hexadecimal | 4-byte uptime in ms | 2-byte packet counter | 2-byte PMT ADC reading |

**Housekeeping** Packet Structure

| SYNC | RTC | UPTIME | SEQ | MCUHK | VMON x8 | TMP x5 |
|---|---|---|---|---|---|---|
| Fixed 2-byte hexadecimal | 6-byte RTC in UTC | 4-byte uptime in ms | 2-byte packet counter | 8-bytes of MCU Housekeeping | 2-byte Voltage Monitor ADCs | 2-byte On-board temp sensors |

# GSE for Instrument Testing

- Custom PC-based GUI
- Control of iMCU over UART
- Displays live packet data from ERPA, PMT, and Housekeeping
- Monitor voltage levels, sensor output, and system status
- Validate and debug firmware and sensor operation before satellite integration



PMT PACKET ○

```
SYNC:   0xFFFF
SEQ:    0104
ADC:    0.045395
```

ERPA PACKET ○

```
SYNC:      0xEEEE
SEQ:       2089
SWP MON:   1.18139
ADC:       4.855802
```

0 0.0

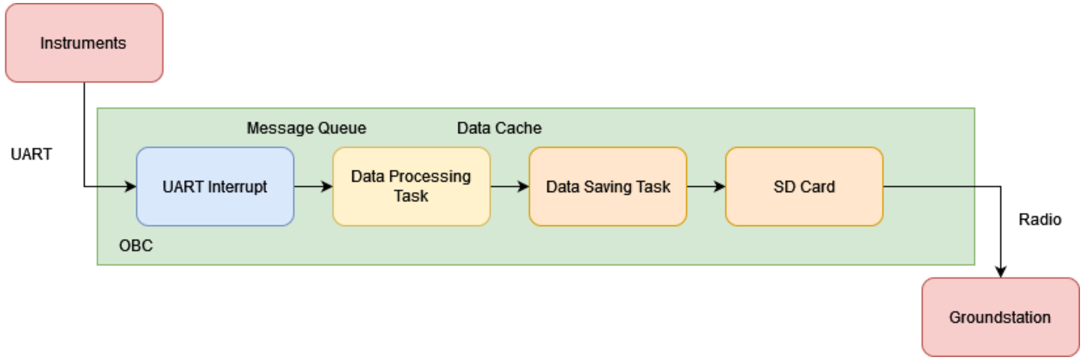
1

HK PACKET ○

```
SYNC:        0xDDDD
SEQ:         0009
vsense:      26.22712
vrefint:     1.22007
BUSvmon:     2.73751
BUSimon:     0.06527
2v5mon:      2.49656
5vmon:       3.00183
n3v3mon:     3.11546
n5vmon:      3.06469
15vmon:      2.41758
5vrefmon:    2.99377
n200vmon:    2.93817
n800vmon:    2.58440
TEMP1:       24.62500
TEMP2:       24.56250
TEMP3:       24.81250
TEMP4:       24.75000
TMP1:        43.21732
```

# On Board Computer (OBC)

- COTS– EnduroSat’s OBC
- Flight software built upon EnduroSat’s SDK
- Manages commanding and collecting telemetry, onboard logging, file system, and timing
- Interfaces with all subsystems: Instrument, ADCS, EPS+Batt, Transceiver
- Instrument data is processed on-board and saved on an SD card before being downloaded
- Note:
  - o Has inter-thread communication and event-driven scheduling
  - o Handles UART with interrupt limitations and buffering logic



# Instrument Data Post Processing on OBC

**PMT**

- Averages two raw samples to one post processed sample

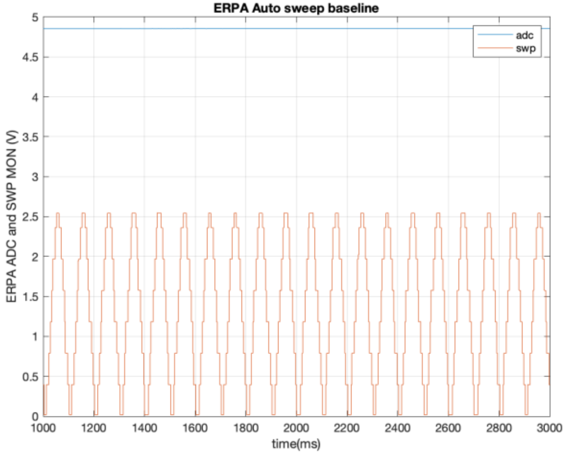
**ERPA**

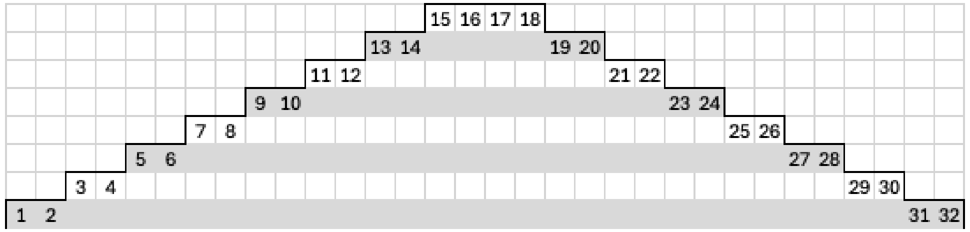
- Averages 4 raw samples per sweep step to one post processed sample
- 32 samples are averaged to 8 sample, one per sweep step
- Eg. AVG[1,2,31,32] output ADC0

NOTE: Processed Instrument packets are stored in corresponding buffers to reduce write times.

| SEQ | PMT_ADC |
|---|---|
| Fixed 2-byte hexadecimal | Averaged PMT ADC |

**PMT** Processed Packet





**ERPA** processing Technique

| SEQ | SWP0+ADC0 | SWP1+ADC1 | SWP2+ADC2 | SWP3+ADC3 | SWP4+ADC4 | SWP5+ADC5 | SWP6+ADC6 | SWP7+ADC7 |
|---|---|---|---|---|---|---|---|---|
| Fixed 2-byte hexadecimal | SweepMon and ADC for Step0 | SweepMon and ADC for Step1 | SweepMon and ADC for Step2 | SweepMon and ADC for Step3 | SweepMon and ADC for Step4 | SweepMon and ADC for Step5 | SweepMon and ADC for Step6 | SweepMon and ADC for Step7 |

**ERPA** Processed Packet

# iMCU and OBC– Error Handling

**Instruments**

- iMCU monitors power rail voltages, temperatures, ECC…
- iMCU generates an error packet, sends it to OBC, and turns the instrument off

**OBC**

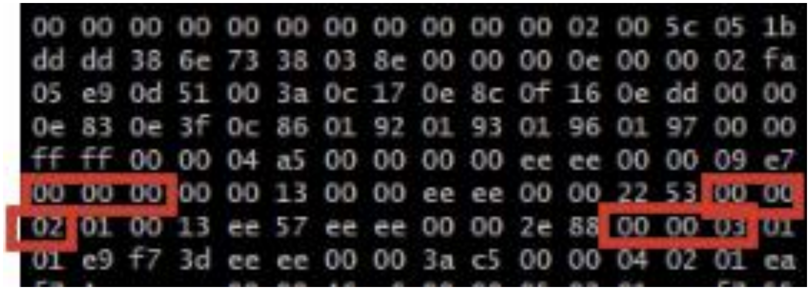
- Tracks error packets from iMCU
- Logs previous and current error packets
- Can request last known error data for debug or telemetry
- Re-attempts failed packet commands (e.g., sync and version)

**Missing Packets**

- Packets have the possibility of not being received or corrupted.
- Ranges of missing packets are recorded. Reduced write times from writing every missed packet.

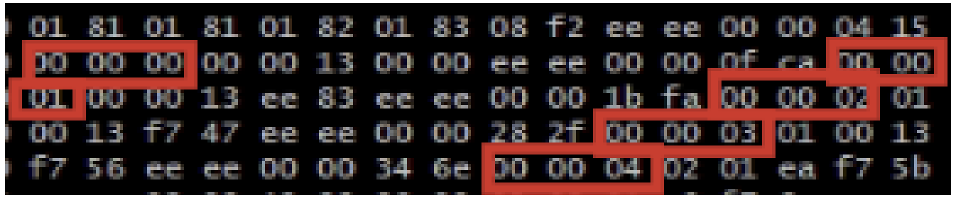
**Missing Files**

- Store the associated reported missing sequences.
- After processing has finished, the missing packet seq # are appended to the instrument file



```
00 00 00 00 00 00 00 00 00 00 00 02 00 5c 05 1b
dd dd 38 6e 73 38 03 8e 00 00 00 0e 00 00 02 fa
05 e9 0d 51 00 3a 0c 17 0e 8c 0f 16 0e dd 00 00
0e 83 0e 3f 0c 86 01 92 01 93 01 96 01 97 00 00
ff ff 00 00 04 a5 00 00 00 00 ee ee 00 00 09 e7
00 00 00 00 00 13 00 00 ee ee 00 00 22 53 00 00
02 01 00 13 ee 57 ee ee 00 00 2e 88 00 00 03 01
01 e9 f7 3d ee ee 00 00 3a c5 00 00 04 02 01 ea
```

***Missing Sequence*** *- Picture showing sequences missing the sequence number 1*

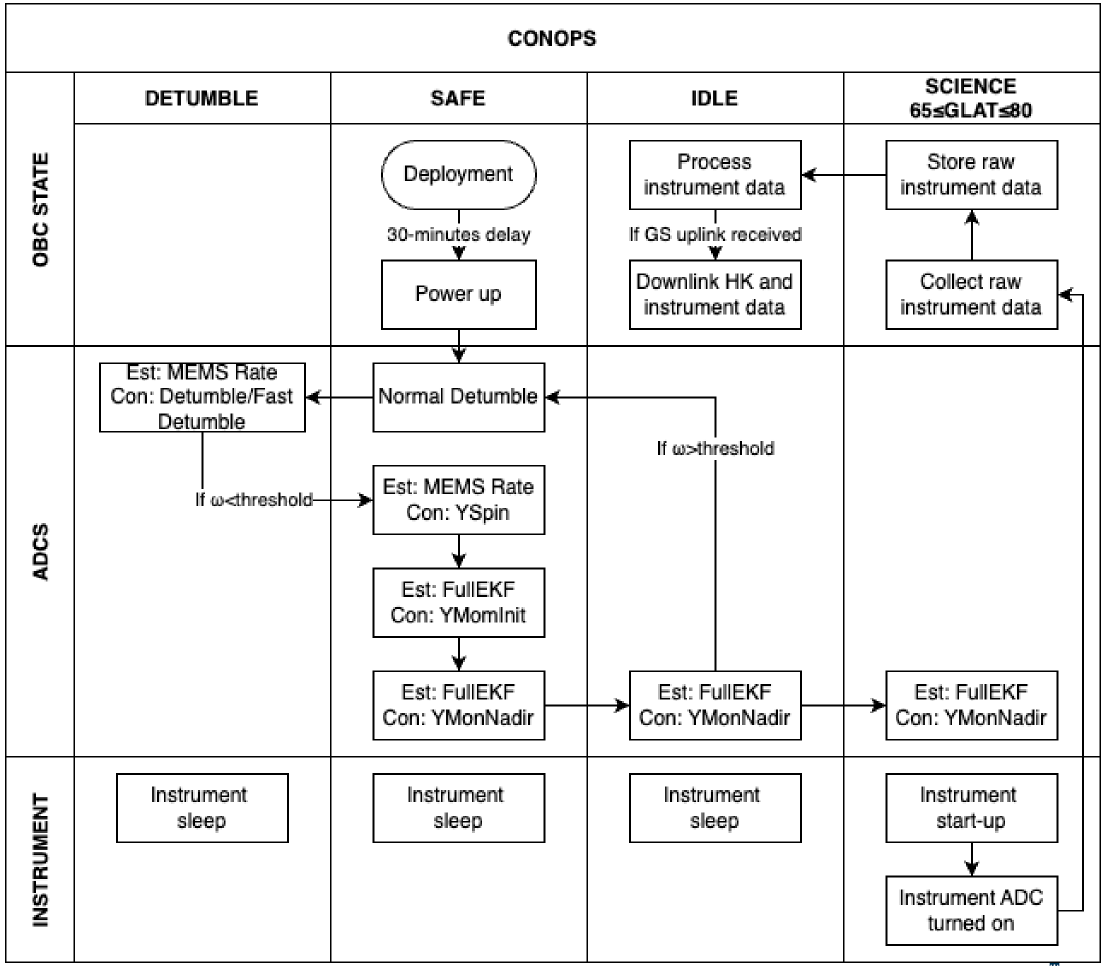


```
01 81 01 81 01 82 01 83 08 f2 ee ee 00 00 04 15
00 00 00 00 00 13 00 00 ee ee 00 00 0f ca 00 00
01 00 00 13 ee 83 ee ee 00 00 1b fa 00 00 02 01
00 13 f7 47 ee ee 00 00 28 2f 00 00 03 01 00 13
f7 56 ee ee 00 00 34 6e 00 00 04 02 01 ea f7 5b
```

***Correct Sequence*** *- Picture showing sequence 0 through 4*

# ConOps

- The OBC goes into different conops states depending on the status of the satellite:
- **Safe**: Power up/Error
- **Detumbling**: angular rates > threshold
- **Communication**: Ground station in range
- **Science**: Spacecraft in region of interest
- **Idle**: Default state
- Additional sub-states to manage smooth transitions and a variety of conditions.

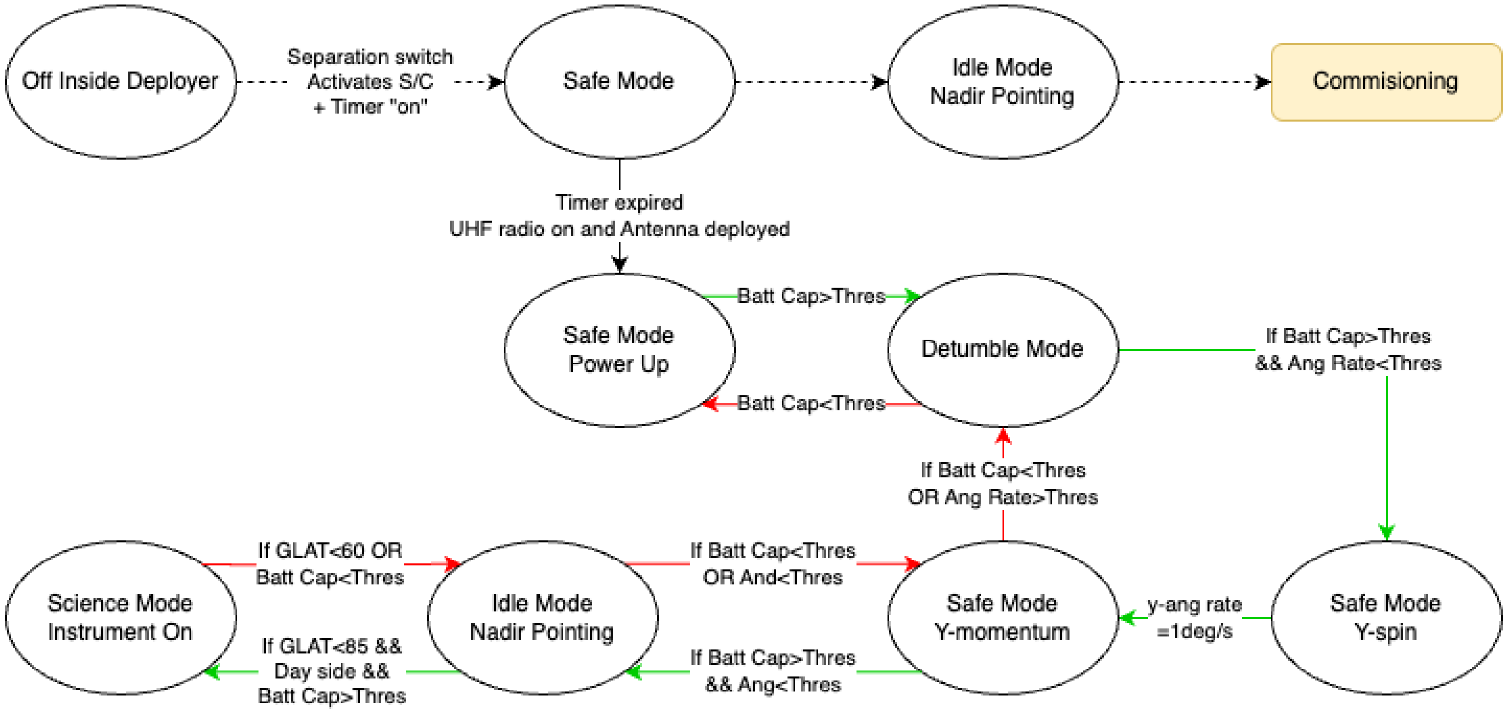


CONOPS

| | DETUMBLE | SAFE | IDLE | SCIENCE 65≤GLAT≤80 |
|---|---|---|---|---|
| OBC STATE | | Deployment<br>30-minutes delay<br>Power up | Process instrument data<br>If GS uplink received<br>Downlink HK and instrument data | Store raw instrument data<br>Collect raw instrument data |
| ADCS | Est: MEMS Rate Con: Detumble/Fast Detumble<br>If ω<threshold | Normal Detumble<br>Est: MEMS Rate Con: YSpin<br>Est: FullEKF Con: YMomInit<br>Est: FullEKF Con: YMonNadir | If ω>threshold<br>Est: FullEKF Con: YMonNadir | Est: FullEKF Con: YMonNadir |
| INSTRUMENT | Instrument sleep | Instrument sleep | Instrument sleep | Instrument start-up<br>Instrument ADC turned on |

# ConOps – Commissioning & More

- The OBC manages ConOps states by tracking subsystem thresholds.
  - ADCS– angular rates, body angles, LLH
  - Battery– Charge Capacity
- UHF Transceiver & Antenna deployment delay
  - post-launch initialization
- Daily system-wide Power cycle
  - Reboot for memory reset



Off Inside Deployer

Separation switch Activates S/C + Timer "on"

Safe Mode

Idle Mode Nadir Pointing

Commisioning

Timer expired
UHF radio on and Antenna deployed

Safe Mode Power Up

Batt Cap>Thres

Batt Cap<Thres

Detumble Mode

If Batt Cap>Thres && Ang Rate<Thres

If Batt Cap<Thres OR Ang Rate>Thres

Science Mode Instrument On

If GLAT<60 OR Batt Cap<Thres

If GLAT<85 && Day side && Batt Cap>Thres

Idle Mode Nadir Pointing

If Batt Cap<Thres OR And<Thres

If Batt Cap>Thres && Ang<Thres

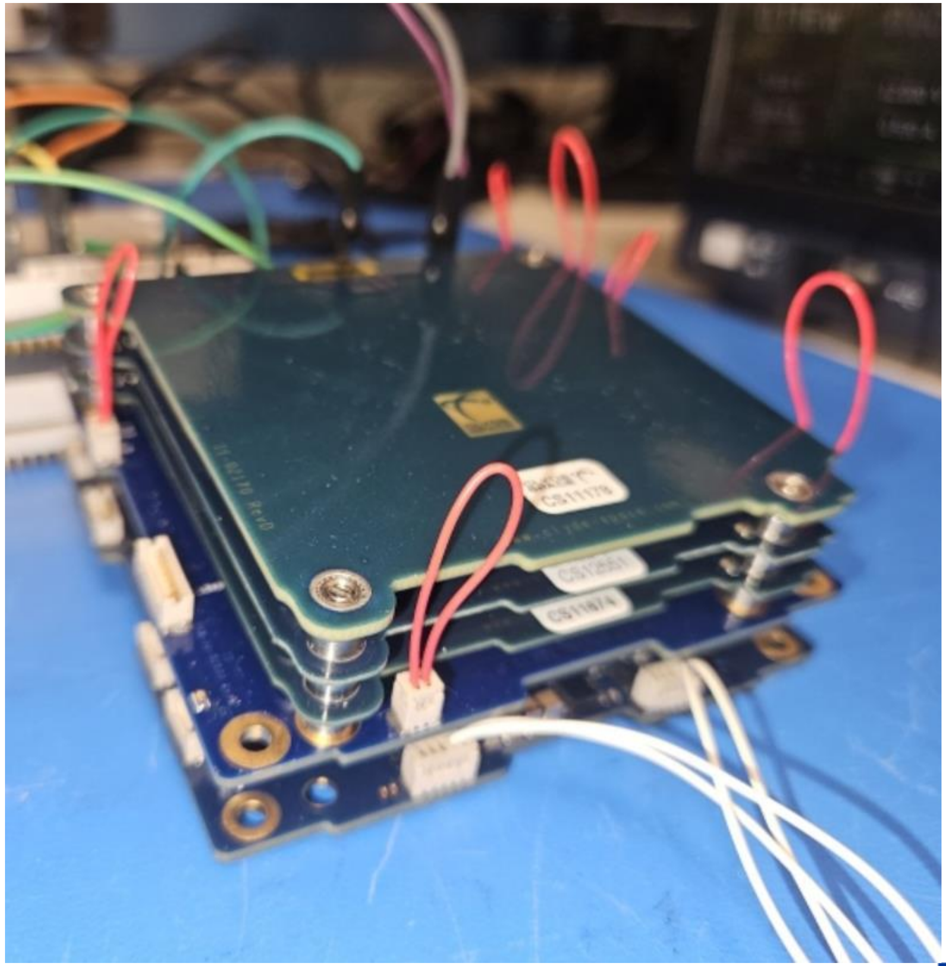
Safe Mode Y-momentum

y-ang rate =1deg/s

Safe Mode Y-spin

# Electrical Power System (EPS)

- Clyde Space - EPS & Battery
  - Communicate via I2C Bus
- **Telecommands** request data and command the EPS & Battery
  - Write commands: establish events
  - Read commands: return telemetry
- **Telemetry** data allows the OBC to monitor the EPS and Battery state



# EPS Integration Testing

- Custom driver to support constant communication between EPS & OBC based on SDK's system datacache and ConOps
- Reworked SDK datacache structure and data handling to support ClydeSpace EPS.
- System Testing:
  - OBC periodically requests EPS Data (Board Status, PDM Volt./Curr., etc) via I2C bus.
  - Response is stored in Datacache
  - Other tasks request current caches
    - i.e. Battery Capacity for Conops
- All EPS+Batt Telemetry is stored in data cache and eventually in telemetry files.

# Thank you

Any Questions?