PyCubed - Introduction

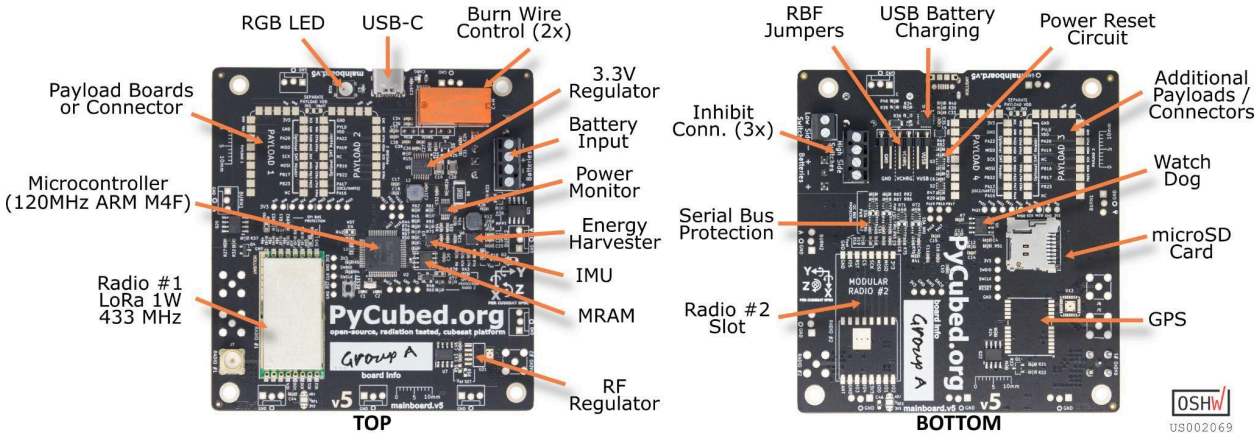
to a CubeSat Development Platform

Created by Andreas Chr. Dyhrberg - ufelectronics.eu

Version: 2025.09.25

Introduction **2**
What PyCubed Is **2**
What Functionality PyCubed Includes **3**
1. On-Board Computer (OBC) 3
2. Electrical Power System (EPS) 3
3. Radio Communication (COMMS) 3
4. Data Storage 3
5. Sensors 3
6. Software Environment 3
7. Open-Source Hardware + Community 3
What You Can Make With PyCubed **4**
Examples: 4
In other words 4
Limitations - What PyCubed Cannot Do **5**
1. Radiation Tolerance 5
2. Limited Power 5
3. Limited Computing Power 5
4. Radio Range and Data Rates 5
5. Size Constraints 5
6. Not a Complete Satellite 5
What Else You Need Besides PyCubed to Send a CubeSat into Orbit **6**
1. Structure (Mechanical Frame) 6
2. Solar Panels 6
3. Batteries 6
4. Antennas 6
5. Payload 7
6. Attitude Determination and Control System (ADCS) (optional but common) 7

7. Thermal Management..... 7
8. Ground Segment..... 7
9. Mission Operations Planning..... 8
10. Launch and Deployment..... 8
11. Regulatory Approvals..... 8
Abbreviations and Terms Explained..... 9



Introduction

This document is written for beginners in satellites and CubeSats. Its purpose is to explain, in simple terms, what PyCubed is, what it can do, and what it cannot do. It also introduces the most important satellite and CubeSat terms so that anyone, even without prior knowledge, can follow along.

PyCubed is a small, open-source electronics platform designed to make it easier to build and operate CubeSats. Think of it as a tiny “computer + power system + communications board” that you can buy and immediately start programming, instead of designing everything from scratch.

What PyCubed Is

• PyCubed is a CubeSat avionics board: a single printed circuit board (PCB) that combines the most important electronics a CubeSat needs to function.

• It was developed at Stanford University (Space Rendezvous Laboratory) as an open-source project to help student and educational CubeSats reduce cost and complexity.

• PyCubed boards use microcontrollers (computers-on-a-chip), run CircuitPython (a beginner-friendly programming language), and include power management and communication hardware.

What Functionality PyCubed Includes

PyCubed is designed as a CubeSat core board. Depending on the version, it includes:

1. On-Board Computer (OBC)

◦ A microcontroller (e.g., ATSAMD51 in early versions) that runs your satellite's software.

◦ This controls all satellite functions: turning systems on/off, managing communications, running experiments.

2. Electrical Power System (EPS)

◦ Power regulation circuits: convert solar panel voltage to safe voltages for the satellite.

◦ Battery charging and monitoring: handles Li-ion battery packs.

◦ Power distribution: sends power to different subsystems.

3. Radio Communication (COMMS)

◦ Integrated UHF/VHF transceivers (depending on version).

◦ Lets you send data to Earth (telemetry) and receive commands.

4. Data Storage

◦ On-board flash memory and SD card support to log science data and housekeeping data.

5. Sensors

◦ Built-in sensors such as IMUs (Inertial Measurement Units: gyros, accelerometers, magnetometers) for orientation sensing.

◦ Temperature sensors for monitoring hardware.

6. Software Environment

◦ Runs CircuitPython: a Python dialect designed for microcontrollers.

◦ This makes coding accessible—no need for embedded C skills.

◦ Pre-written libraries for common functions (power, comms, sensors).

7. Open-Source Hardware + Community

◦ Schematics and PCB design files are available online.

◦ Code examples and documentation help beginners get started.

What You Can Make With PyCubed

With PyCubed, you can build and operate a basic CubeSat mission without designing your own avionics from scratch.

Examples:

- **Educational CubeSats**: test sensors, cameras, or simple science experiments.

- **Technology demonstration satellites**: test a new antenna, solar panel, or material in orbit.

- **Student projects**: run software you've written in Python on real space hardware.

- **Testbeds**: practice building spacecraft subsystems on the ground before scaling up.

In other words...

PyCubed makes it possible for small teams, even beginners, to:

- Run code in space.

- Collect data from sensors.

- Communicate with Earth.

- Manage power from solar panels and batteries.

Limitations - What PyCubed Cannot Do

1. Radiation Tolerance

◦ PyCubed uses a commercial microcontroller (ATSAMD51).

◦ ATSAMD51 was originally radiation hardened, but it seems the latest batches have some problems. - https://pycubed.org/ is working on a solution.

2. Limited Power

◦ PyCubed can handle small solar panels and batteries, but not large power-hungry payloads (like electric propulsion or radar).

3. Limited Computing Power

◦ Microcontrollers running CircuitPython are excellent for simple tasks, but not suited for heavy computing (image processing, AI, etc.).

4. Radio Range and Data Rates

◦ Radios are basic (UHF/VHF). Good for telemetry and simple payload data, not for high-speed downlinks like photos or video.

5. Size Constraints

◦ Designed mainly for 1U to 3U CubeSats. Larger spacecraft require more scalable/custom avionics.

6. Not a Complete Satellite

◦ PyCubed is only the avionics (OBC + EPS + comms). You still need:

- Structure (the CubeSat frame).
- Solar panels.
- Antennas.
- Payload (camera, sensor, experiment).
- Launch provider to put it into orbit.

What Else You Need Besides PyCubed to Send a CubeSat into Orbit

PyCubed provides the **avionics** (computer, power regulation, communications, sensors), but a working CubeSat mission requires many additional systems and considerations.

1. Structure (Mechanical Frame)

- The **CubeSat frame** is the standardized box that houses all electronics, batteries, and payloads.
- It must comply with the **CubeSat Design Specification (CDS)** so it fits into launch deployers.
- Typically made of aluminum alloys with surface treatments for strength and thermal balance.

2. Solar Panels

- Provide electrical power by converting sunlight into usable voltage.
- Must be sized for the mission's power budget.
- Can be **body-mounted** (on the CubeSat faces) or **deployable** (folded-out panels for more area).

3. Batteries

- Store energy for eclipse periods (when the satellite is in Earth's shadow).
- Usually **Li-ion or LiPo cells**, with protections for charging and discharging.
- Integrated with PyCubed's EPS.

4. Antennas

- Needed for communication with ground stations.
- Can be simple **whip antennas** (spring-deployed) or more complex patch arrays.
- Must be stowed during launch and deployed in orbit.

5. Payload

- The mission-specific hardware. Examples:
    - Camera or imaging system.
    - Science sensor (magnetometer, spectrometer, particle detector).
    - Technology demonstrator (new material, radio, or component test).
- Payload often dictates the mission's orbit, data requirements, and satellite size.

6. Attitude Determination and Control System (ADCS) *(optional but common)*

- Controls the CubeSat's orientation (which way it points).
- Components can include:
    - **Magnetorquers** (electromagnets interacting with Earth's field).
    - **Reaction wheels** (spinning masses for fine control).
    - **Star trackers or sun sensors** (to know orientation).
- Not all missions need precise pointing, but imaging and comms often do.

7. Thermal Management

- Satellites experience rapid temperature swings in orbit.
- Thermal design includes coatings, surface finishes, and sometimes heat pipes.
- CubeSats typically rely on passive management (careful design, no moving parts).

8. Ground Segment

- A CubeSat is useless without a **ground station**.
- Ground segment includes:
    - **Antennas on Earth** (UHF/VHF or S-band, depending on comms).
    - **Software-defined radios (SDRs)** for sending/receiving signals.
    - **Mission control software** for scheduling passes, decoding telemetry, and uploading commands.

9. Mission Operations Planning

- Define what the CubeSat does on each orbit.
- Includes data collection schedule, downlink windows, battery charge management, and fault recovery.

10. Launch and Deployment

- CubeSats need a **launch provider** (SpaceX, Rocket Lab, ESA, etc.).
- They are deployed from standard deployers (like **P-PODs**).
- Each deployer has strict **Integration & Test (I&T) requirements** for mechanical, electrical, and safety compliance.

11. Regulatory Approvals

- You need licenses and coordination before launch:
    - **Frequency licensing** from ITU/national regulators for radio use.
    - **Space debris mitigation plans** (e.g., de-orbit within 25 years).
    - Export controls in some countries (ITAR, EAR).

In short

PyCubed gives you the brain and heartbeat of the CubeSat, but you still need the body (structure, solar panels, batteries, antennas, payload), the support systems (ADCS, thermal, ground station), and the paperwork (licenses, launch provider agreements) to actually put it in orbit.

Abbreviations and Terms Explained

• **CubeSat**: A small standardized satellite, usually 10×10×10 cm per "Unit" (1U). Larger CubeSats are 2U, 3U, etc.

• **Avionics**: The electronics that let the satellite "think" and "communicate."

• **OBC** (On-Board Computer): The satellite's brain (runs the software, controls subsystems).

• **EPS** (Electrical Power System): Handles power: solar input, battery charging, distribution.

• **COMMS**: Communication subsystem—radios that link the satellite with Earth.

• **Payload**: The mission-specific hardware (camera, sensor, experiment).

• **IMU** (Inertial Measurement Unit): A sensor that measures acceleration, rotation, and orientation.

• **Telemetry**: Data sent from the satellite to Earth about its status (voltages, temperatures, errors).

• **Command uplink**: Instructions sent from Earth to the satellite.

• **UHF/VHF**: Radio frequency bands commonly used for small satellite communication.

• **CircuitPython**: A simplified version of Python that runs on microcontrollers, easy for beginners.

• **Radiation effects** (SEU/SEL):

  ◦ **SEU** (Single Event Upset): A bit flip in memory caused by cosmic radiation.

  ◦ **SEL** (Single Event Latch-up): A radiation-induced short circuit that can permanently damage chips unless power is cut.