**INTER IIT SELECTION HACKATHON**

# DRONA AVIATION:

# PLUTO DRONE CHALLENGE

**-SS SQUADRON**

**Pluto Class:**

The pluto class serves as the primary interface for interacting with the Pluto drone. It encapsulates the connection to the drone and provides methods for sending commands and receiving data. Key functionalities within the pluto class typically include:

* **Connection Establishment:** Methods to establish a connection with the drone over Wi-Fi or other communication protocols.
* **Command Transmission:** Functions to send various commands to the drone, such as takeoff, landing, movement, and configuration settings.
* **Data Reception:** Methods to receive data from the drone, including sensor readings, status information, and telemetry data.
* **Error Handling:** Mechanisms to handle potential errors or exceptions that may occur during communication or drone operations.

**MSP Packets:**

MSP (Micro Serial Protocol) packets are standardized data structures used for communication between the drone and the ground station. They contain information such as the command type, arguments, and a checksum for error detection. The pypluto library likely provides functions for creating, encoding, and decoding MSP packets.

**Keyboard Control:**

Keyboard control allows the user to interact with the drone directly using a keyboard. It involves mapping specific keys to different commands or actions. For example, pressing the "W" key might make the drone move forward, while pressing the "A" key might make it turn left. The pypluto library might provide a built-in keyboard control mechanism or allow for custom key mappings.

**Command Generation:**

The command generation process involves creating MSP packets based on the user's input or predefined commands. This typically involves:

* **Parsing user input:** Interpreting keyboard inputs or other user interactions to determine the desired command.
* **Creating MSP packet:** Constructing the MSP packet with the appropriate command type, arguments, and checksum.
* **Encoding packet:** Encoding the MSP packet into a format suitable for transmission over the communication channel.

**Packet Transmission:**

Once the MSP packet is generated, it is transmitted to the drone over the established Wi-Fi connection. This typically involves:

* **Serializing data:** Converting the MSP packet into a byte stream for transmission.
* **Sending data:** Transmitting the byte stream to the drone using the appropriate communication protocol.
* **Error checking:** Implementing mechanisms to detect and handle transmission errors, such as retransmissions or error correction.

**Drone Response:**

The drone receives the MSP packet, decodes it, and processes the commands. The drone's firmware interprets the commands and executes the corresponding actions, such as controlling motors, adjusting flight parameters, or activating sensors.

**Data Reception (Optional):**

If data reception is enabled, the script can receive data from the drone. This data might include:

* **Sensor readings:** Data from sensors such as gyroscopes, accelerometers, barometers, and magnetometers.
* **Status information:** Information about the drone's current state, such as battery level, altitude, and flight mode.
* **Telemetry data:** Other relevant data that the drone transmits.

The received data can be processed and displayed in a human-readable format or used for further analysis or control purposes.