TEXAS A&M UNIVERSITY Engineering

Disaster Response Observation Network (DRON)

Project Lead: Ian Wilhite

Subteam Leads: Chris Ambroziak, Aidan Briggs, Ryo Kato

Seungbin Ahn, Ryan Axum, Om Bhatt, Caleb Santiago, Jackson Shouba, Asbah Talal, Owen Thomas, Drew Wheaton, Colin Xu

TEXAS A&M UNIVERSITY
ROBOTICS TEAM & LEADERSHIP EXPERIENCE

Problem Definition

The Disaster Response Observation Network (DRON) is a proof-of-concept initiative that aims to leverage unmanned aerial vehicles (UAVs) to gather intelligence during structural fires to aide first responders in their scene assessment and emergency response.

Methodology

Autonomous swarm functionality allows DRON to assist in emergency situations with minimal required human input. DRON is designed around ease of use, speed of deployment, and quality of data gathered and presented.

Functional Requirements

- A network where each individual node can function independently of each other for redundancy.
- Data transmitted to a centralized Ground Control Station (GCS) for interpretation and use by responders without a technical background.
- Ability to display hotspots on interactive 3d structures to model surrounding hazardous areas.
- Ability to carry payload of instrumentation (~500g) while maintaining flight at ~100 ft long enough to appropriately gather critical data.

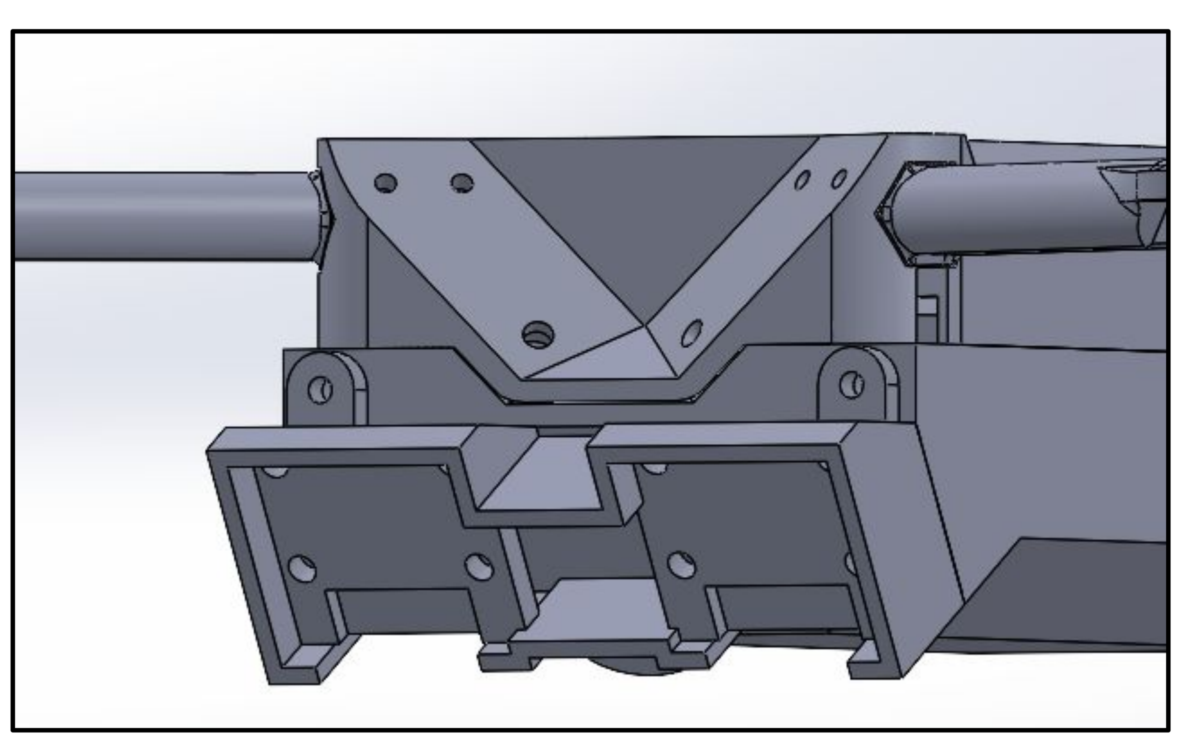

***Figure 1:** Updated Chassis Front View*

Mechanical

DRON is designed to balance weight, structural stability, and modularity to withstand the high stress involved in a flight, namely torque from the propellers and internal stresses from payloads, and continue to function in order to complete a mission.

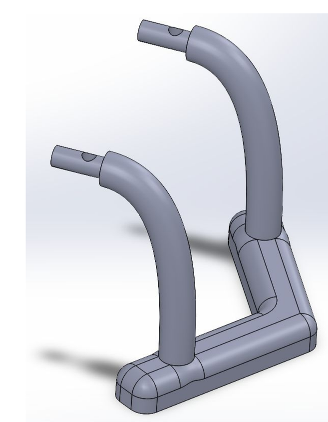

***Figure 2:** Legs*

DRON has adopted the use of PA6 Glass Fiber in the arms to minimize potentially devastating vibrations during flight that can cause failure, in addition to a supporting frame to brace the arms together. All of our parts are connected via heat inserts and screws for ease of repair.

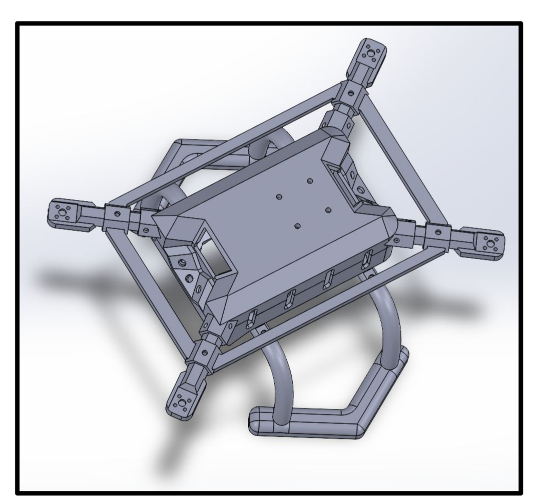

***Figure 3:** Top View of Updated Chassis*

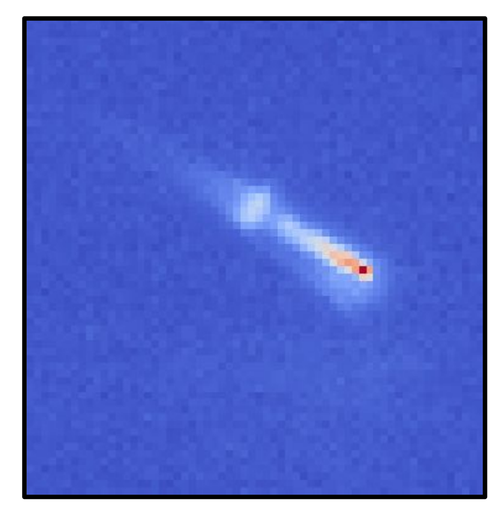

***Figure 4:** Thermal View of Soldering Iron in Fig 5*

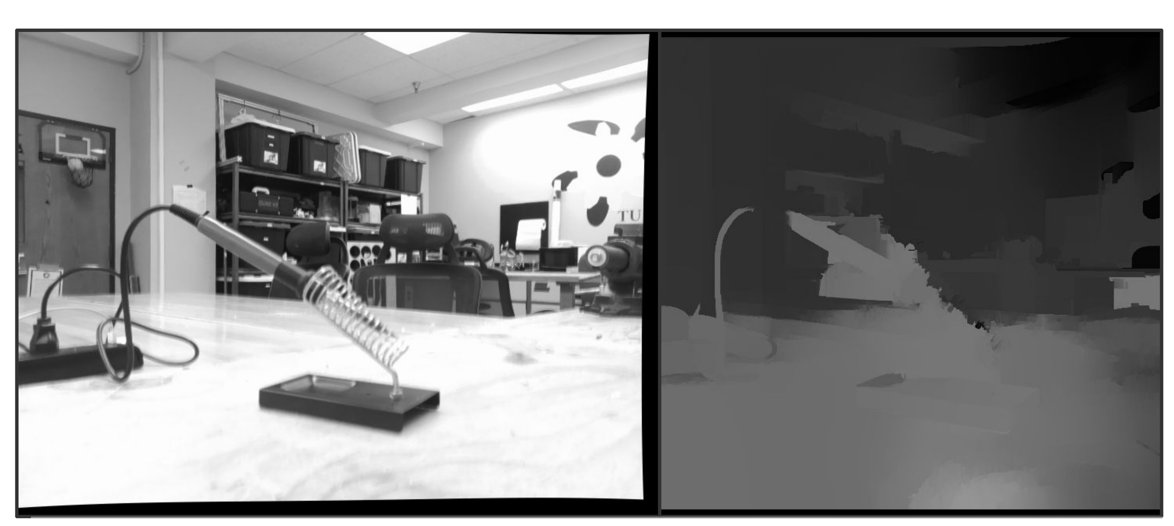

***Figure 5:** Generated Depth Map*

Visualization

The software has the role of managing communications between components, detecting hotspots, and 3D mapping the environment. Two stereo cameras were successfully calibrated to capture depth of their surroundings. Stereo images are published through a ROS2 node into Unity where they are visualized.

The software stack relies on many modules and libraries for flight control, data gathering, image processing, and environment visualization in 3D:

- **A. ROS2 Humble:** Open-Source Framework to handle communication between nodes.
- **B. Senxor:** Library to interface with the thermal camera to gather video feed via USB serial.
- **C. Open CV:** Computer Vision library used to create 'Depth Maps' from Stereo Images to generate the 3D point clouds.
- **D. Unity**: 3D point cloud visualization.
- **E. PX4**: the autopilot firmware that will control the drones movement and communication, interfaced with ROS2 for offboard control

Autonomy

UAV autonomy is being achieved via PX4 firmware simulation and deployment, and is interfacing with PX4 via the uORB messaging API. The team demonstrated multi-agent Software-In-The-Loop (SITL) simulation through Gazebo, with sensor output being published to ROS 2 topics.

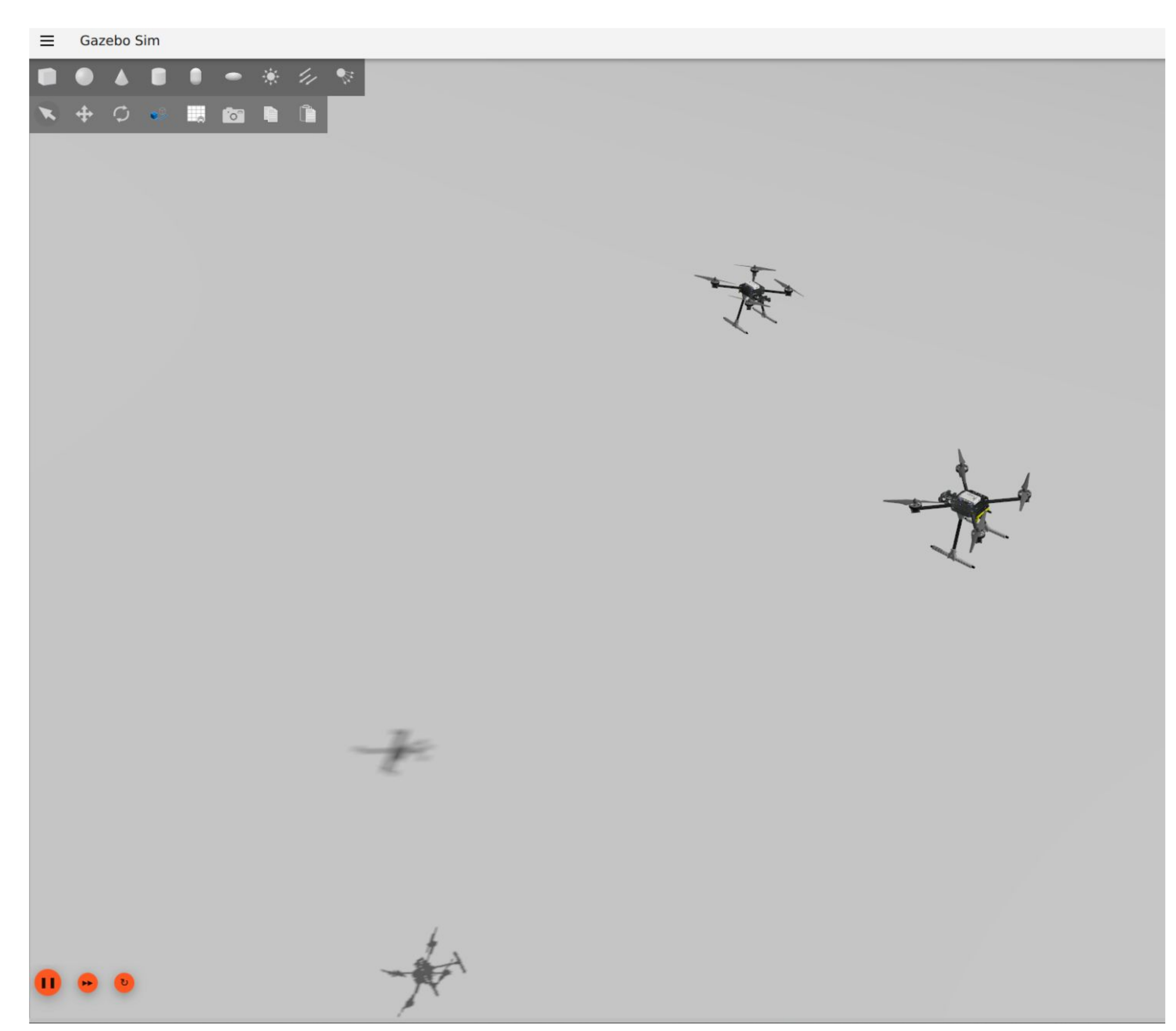

***Figure 6:** Multi-Drone simulation through Gazebo, using PX4*

Electrical

Configuring INAV as the centralized Ground Control Station (GCS) for use with the Flight Controller (FC) and Electric Speed Controller (ESC) stack.

The GPS and IMU provide low-level waypoint based autonomy, and the Raspberry Pi collects sensor data and uses wifi to transmit data and receive flight commands.

The battery selected is rated for 3000mAh and can deliver enough power to sustain maneuvering flight for ~12 min, and low speed flight and hovering for ~15 min.

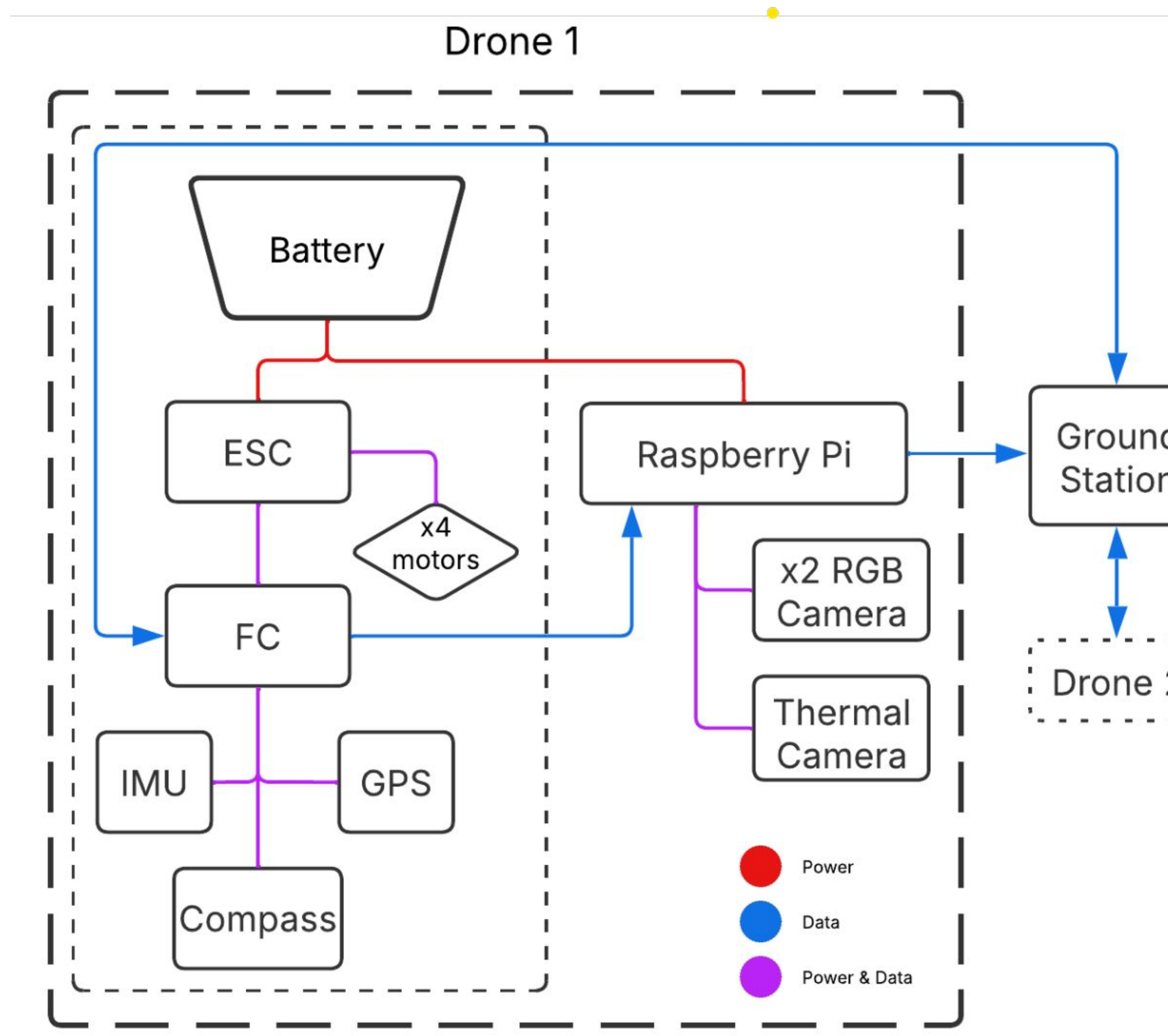

***Figure 7:** Revised system architecture illustration*

Next Steps:

Mechanical: Stress testing for final revisions to chassis and moving towards mass production.

Software: Processing a live data feed and consolidated tests of sensor fusion in plotting and visualization from multiple frames. Optimizing autonomous control and deploying to hardware via OBC & FC.

Electrical: Establish onboard computer publisher nodes, and validate controller on bench tests.

Texas A&M University Robotics Team and Leadership Experience (TURTLE) Lab