Find my KITTEN

Otso Mikkilä, Roy Långsjö, Tomi Bilcu, Lassi Kokkonen, Yinghui Ni, Erik Wahlström, Christopher Romano

For: Antti Nurminen

https://github.com/CatScanners

Introduction

- Drones have increasingly been used for reconnaissance and search operations in environments where other's face difficulties
- The project's goal is to create a drone capable of autonomous navigation and location, recognition, and following of target objects and reporting the findings to the operator.
- The project's software, practical tests and it's related documents are made for research and for other developers to learn and create on top of it

Technologies

Software
ROS 2
Isaac ROS
PX4
PyTorch
Isaac Simulator
Numpy
Colcon
Pegasus Simulator
C++
Python
QGroundControl
Yolo V5

Hardware
Nvidia Jetson Nano and NX
ArduCam AR0234
Oak D pro camera
Pixhawk 6x revision 8
ICP 200
H-RTK ZED-F9P rover
ICM-45686
ArduCam 12 MP IMX708
HolyBro sik radio
Radiomaster TX12
BMP388
BMM150

Hardware setup

- The drone is controlled by a flight controller (Pixhawk), which can receive commands from an offboard computer (Jetson).
- A Ground Control Station (GCS) laptop is connected to the flight controller via telemetry radio for traditional flight control, another laptop is connected to the vehicle via a mobile hotspot to run and monitor scripts running on the offboard computer.
- A relay program provides virtual NTRIP (RTK) corrections source for RTK GPS. The drone is linked to a manual remote control radio for manual flight capabilities.
- The drone has a stereo camera (OAK-D) for use in VSLAM and a standard camera (IMX) for image recognition

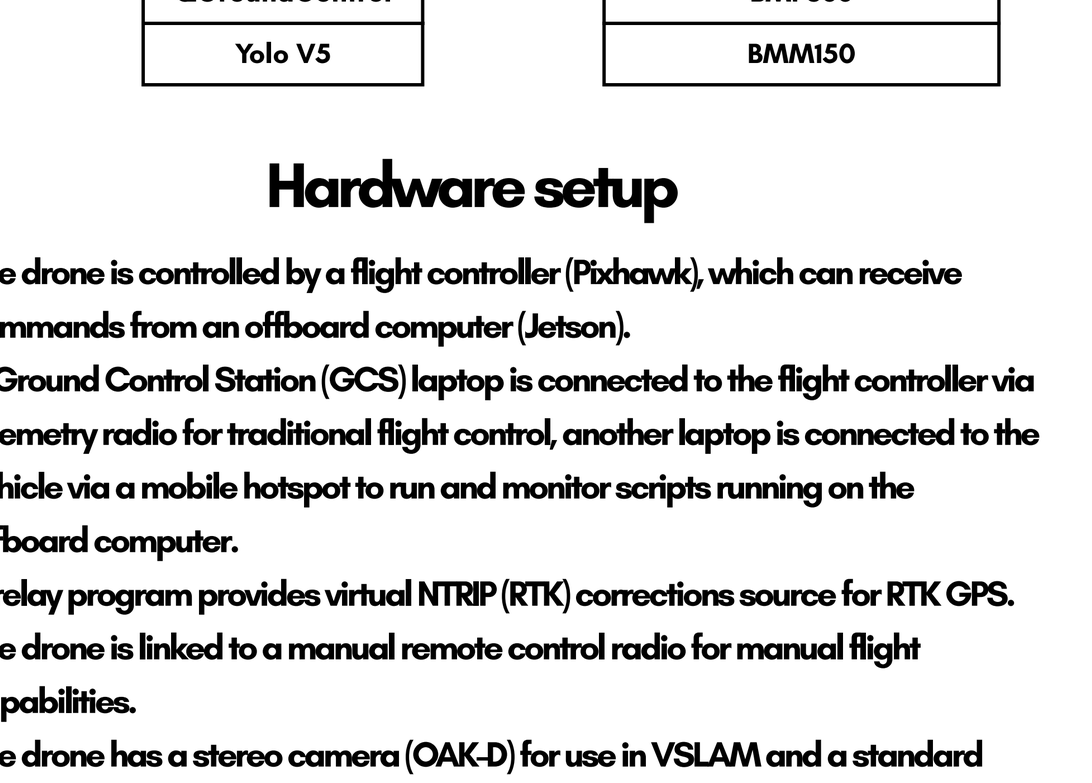

Figure 2: Hardware architecture

Figure 1: The drone

Objectives

1. Improve the drones image recognition, autopilot and real-time kinematic positioning software implemented by the predecessors
2. Create a simulator which can be used to test the drones image recognition features on various environments.
3. Convert existing capabilities , such as image capture, image recognition to Nvidia's alternatives to better utilise the drone's Graphics Processing Unit (GPU)
4. Implement autonomous navigation on the drone without
5. Create a custom location algorithm that is more accurate than existing solutions and can be tested with our own C++ and ROS 2 code.
6. Making clear and structured documentation that helps outside readers and future groups to understand the project and have an easy start
7. Create build pipeline for testing

Results

1. IsaacSim and Pegasus Simulator were tested to create a drone simulator that can be run on Ubuntu either in a local environment or in a docker container. Initial tests showed mixed results as the simulated drone did not move as intended
2. Added VSLAM onto the drone to handle independent movement and navigation
3. We conducted 4 flight sessions, which had at least 2 people present. During the flights, the drones movement was tested while collecting location, altitude and image data for further debugging and development of the software technologies.
4. Added a stereo camera, and a new standard camera whose output can be directly read and decoded by the GPU

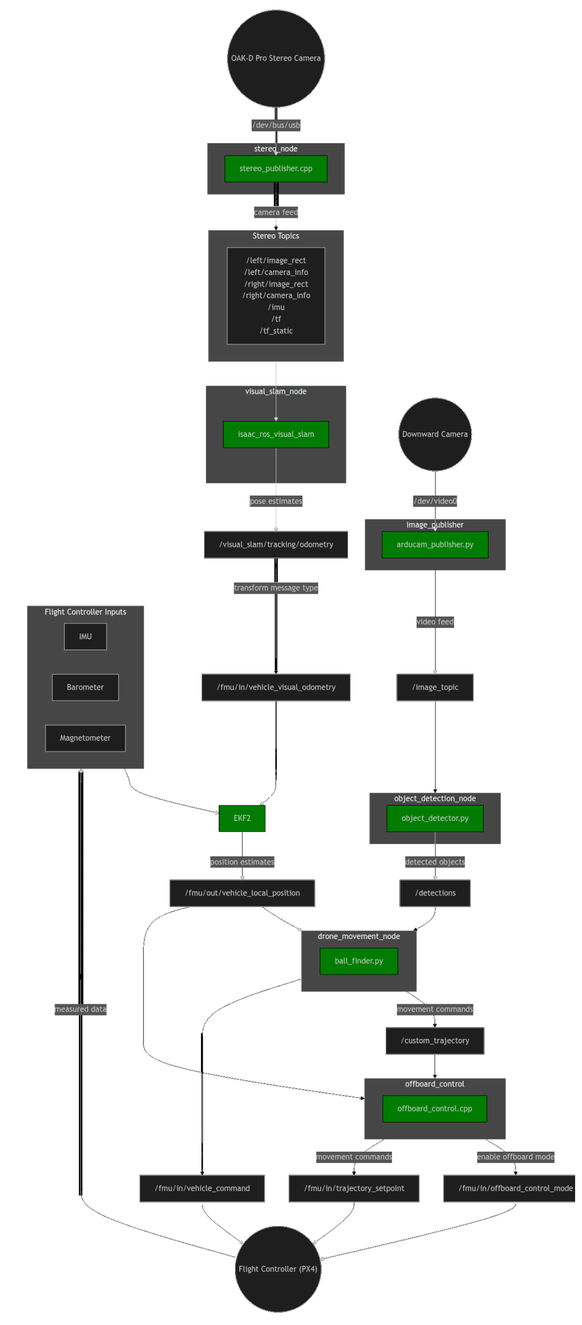

Figure 3: Software stack