AUTONOMOUS GROUND RECONNAISSANCE DRONE USING ROBOT OPERATING SYSTEM (ROS)

Undergraduate Students: David Hung, Cinthya Tang, Coby Allred, Kennon McKeever, James Murphy, Ricky Herriman
Faculty Advisors: Dr. Michael W. Marcellin
Department of Electrical and Computer Engineering
The University of Arizona, Tucson, AZ 85721

ABSTRACT

The Arizona Autonomous Club is a student organization at the University of Arizona which designs, builds, and competes with Unmanned Air Systems (UAS). This year, a 25% scale Xtreme Decathlon model aircraft was selected and successfully converted into a fully autonomous UAS for the AUVSI Student Unmanned Aerial Systems (SUAS) 2017 competition. The UAS utilizes a Pixhawk autopilot unit, which is an independent, open-hardware project aiming at providing high-end autopilot hardware at low costs and high availability. The Pixhawk runs an efficient real time operating system (RTOS) and includes sensors such as a GPS unit, IMUs, airspeed, etc. The UAS also includes an onboard imaging system, which is controlled by an onboard computer (OBC). The Pixhawk and OBC are interconnected with two ground control stations (GCS) using the Robot Operating System (ROS) framework, which is capable of extending overall system capabilities to include an expanded telemetry downlink, obstacle avoidance, and manual overrides.

INTRODUCTION

Background

The Association for Unmanned Vehicles Systems International (AUVSI) has hosted the Student Unmanned Aerial Systems (SUAS) Competition annually since 2002. The competition is centered around the design, fabrication, and deployment of unmanned aerial systems (UAS). The UAS are tested on various mission tasks such as autonomous flight, obstacle avoidance, ground target identification and classification, and payload delivery.

This year the University of Arizona Autonomous Vehicles Club was organized into three sub-teams: hardware, flight control software, and imaging. The hardware team was responsible for the end-to-end design and fabrication of the airframe and avionics. The flight control software team was responsible for all software aspects for flight, including autopilot setup, mission planning, and creating custom airframe control configurations. The imaging subteam was responsible for the entire imaging stack, including imaging hardware both onboard and offboard the system as well as all peripheral software such as image processing and wireless communications. The imaging subteam was also in charge of the networking interface with the AUVSI SUAS judges' server.

Problem Statement

Success at the AUVSI SUAS competition requires a robust platform capable of accomplishing a wide breadth of tasks. As first time competitors, Arizona Autonomous prioritized the production of a solid basic platform to act as a launchpad for future development. As a result, the primary development focus for the 2017 AUVSI SUAS competition was placed on deploying a minimally featured framework which would be added to in subsequent years.

Objectives

The competition poses several mission tasks to be completed including fully autonomous flight, target recognition and classification, and payload delivery. This year's platform was also designed to prove infrastructure for the first two tasks, while secondary tasks such as payload delivery were excluded. A complete list of mission tasks may be found in the official AUVSI SUAS competition rules [1].

DESIGN SOLUTIONS

Airframe

This year, a 25% scale Xtreme Decathlon RC Acrobatics aircraft was selected as the base for the airframe. It has a hobby plywood and balsa construction with Ultracote covering for aerodynamics. The airframe has an overall wingspan of 92.2 inches and a fuselage length of 66 inches. It is powered by a Desert Aircraft 50cc engine with a 23x8 inch two blade propeller and is controlled using flaps, ailerons, elevators, and a rudder. In order to fit within the custom interior, a custom servo mount was designed and 3D printed and installed onto the airframe.More information about the 25% scale Xtreme Decathlon RC Acrobatics aircraft can be found at the vendor's website [2].

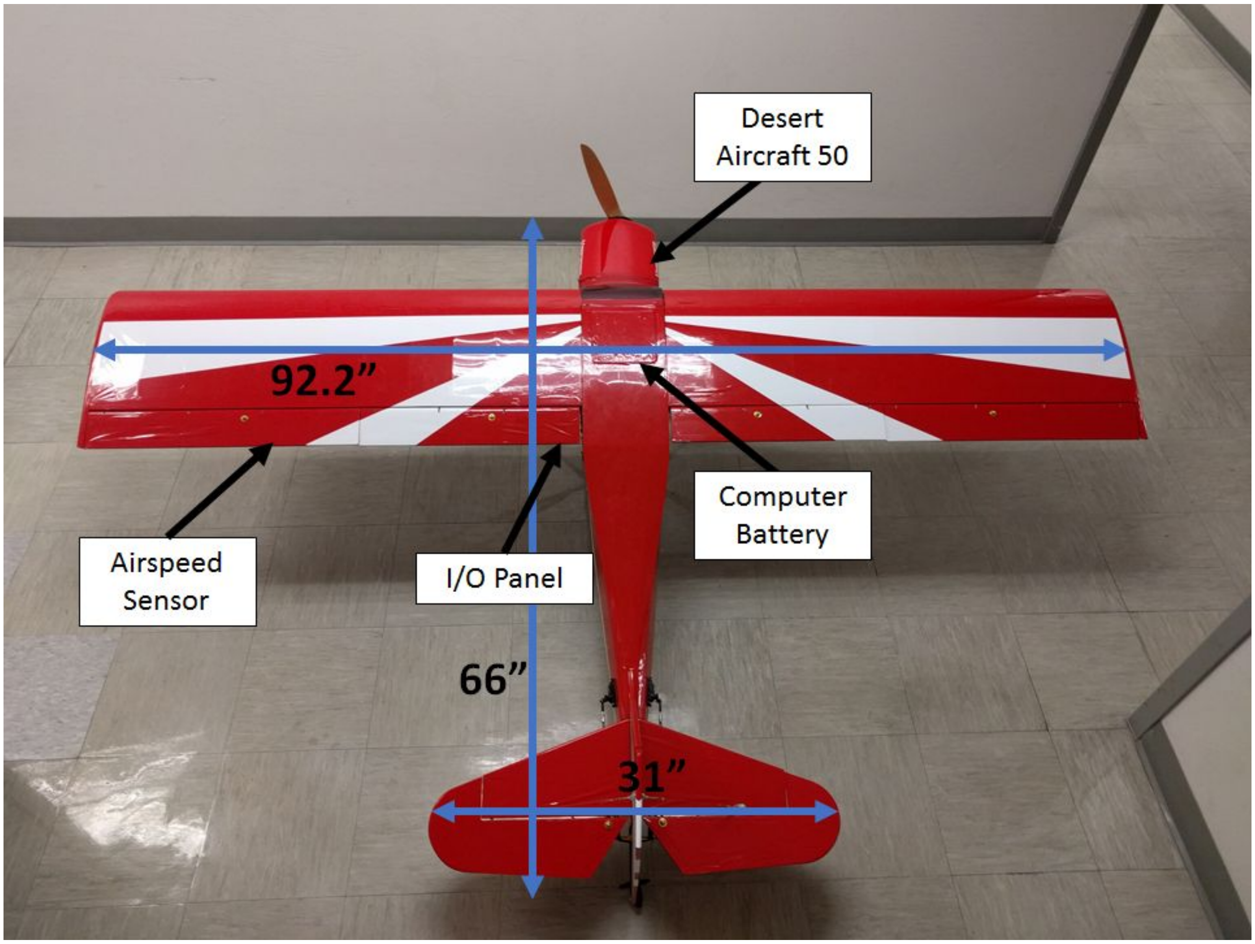

Figure 1. Top-down image of the UAS with measurements (in inches) and labels.

The airframe was modified with custom mounting solutions in order to support the various avionics and imaging equipment necessary to complete the mission tasks. Externally, a portion of the aircraft’s belly was cut out and replaced with a flat piece of hobby plywood in order to act as a mounting location for the gimbal assembly. On the wings, a mounting position and routing path were created for an airspeed sensor with accompanying pitot tubes and electrical wiring. Internally, plywood plates were added to create a three layered interior structure. This structure was leveraged to mount all avionic and computational equipment, such as the Pixhawk autopilot, Odroid on-board computer, and three independent, battery power circuits. Finally, a custom I/O panel was added to the airframe and was mounted on the side of the aircraft to provide an easily accessible interface for regular maintenance and software updates. It houses three switches to power for each of the three independent power systems, three charging ports, five indicator LED’s to indicate system power states and status, and various data ports. The addition of the I/O panel greatly decreased development overhead and streamlined the preflight setup process.

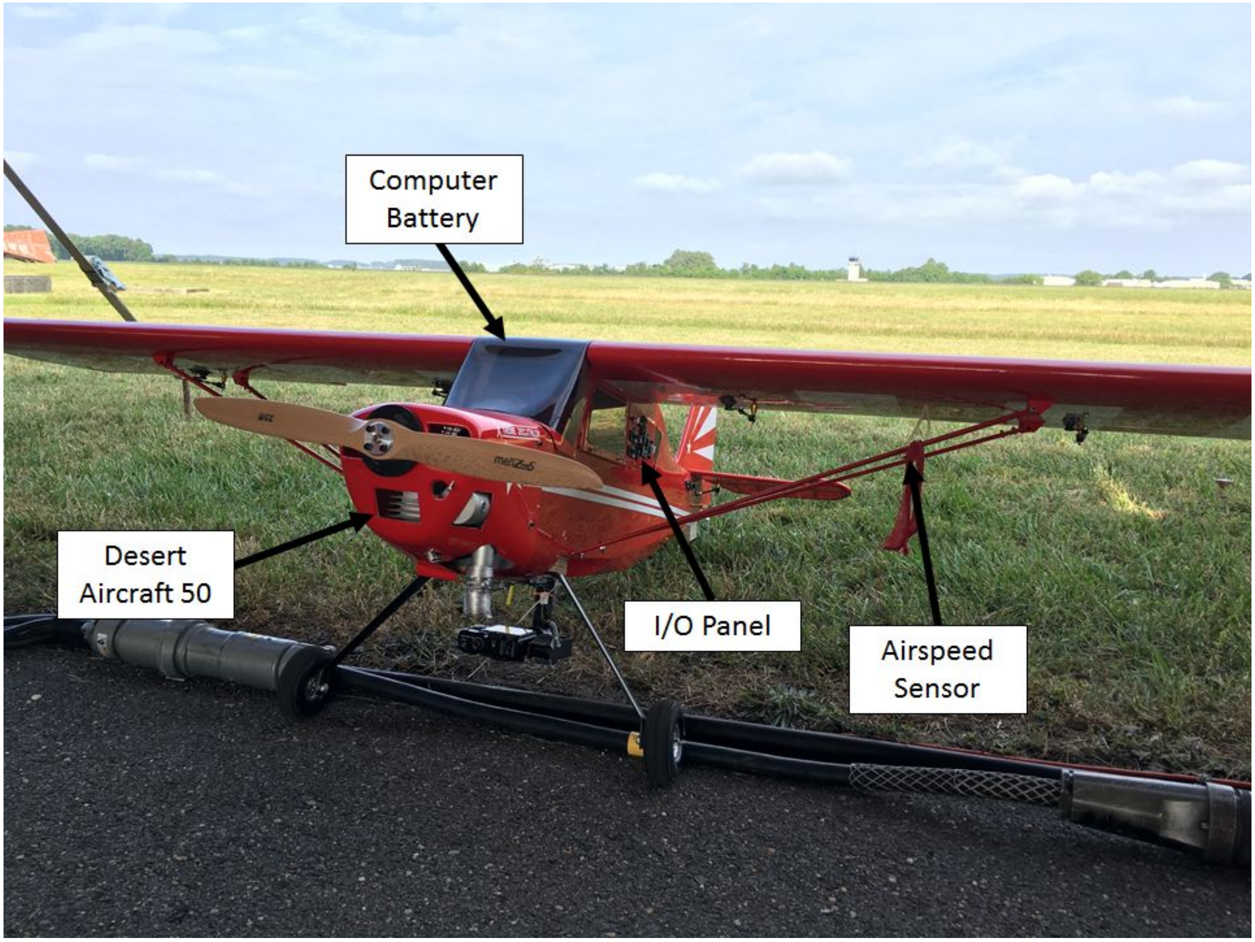

Figure 2. Front view image of the UAS with labels.

Flight Control Software

To control the UAS, the affordable, open-source Pixhawk autopilot platform was installed on the airframe with the PX4 flight stack. It utilizes a 32-bit ARM Cortex M4 core with 256KB of RAM, 2MB of flash memory, and a secondary 32-bit failsafe co-processor and is capable of automated, real-time control of all of the control surfaces on the airframe through the use of PWM servo outputs. The Pixhawk includes a variety of sensors including a 16 bit gyroscope, a 14-bit accelerometer and magnetometer, a 3-axis accelerometer and gyroscope, and a barometer. The Pixhawk platform also supports a variety of peripherals through the use of auxiliary servo outputs to control the mounted camera gimbal and other equipment. A custom airframe configuration was created to allow for the higher number of servos onboard the UAS than the included airframe options. The Pixhawk also supports offboard control which is accomplished using an ODROID XU4 running Ubuntu and ROS. When offboard control is desired, a MAVLink message is sent from the XU4 to the Pixhawk which gives control of the UAS to ROS from the Pixhawk [4].

The primary software used to control the Pixhawk during flight is the ground control software QGroundControl, an application specifically targeted at the Pixhawk and other similar platforms. This software is used for the initial configuration of the Pixhawk and calibration of its onboard sensors along with support for further parameter and control loop tuning. Telemetry data that is sent from the Pixhawk to QGroundControl is displayed in a GUI which shows airframe position, airspeed, mission waypoints, and other telemetry data from the UAS. QGroundControl also supports mission planning through the use of an interactive GUI that allows for waypoint navigation, takeoff and landing, and special mission triggers.

Figure 3. QGroundControl mission planning of 2017 AUVSI SUAS competition.

In addition to the core autopilot software, an additional utility named Path-Generator was developed. Path-Generator is used to parse input coordinates into waypoints, boundary vertices, and search area vertices in order to produce a sequence of waypoint coordinates usable by the Pixhawk autopilot. While input waypoint coordinates are trivial to translate into autopilot coordinates, search area vertices do not directly map to waypoints. In order to properly handle this, Path-Generator leverages QGroundControl's existing Survey command. By storing each of the search area vertices within a particular format QGroundControl is commanded to draw the proper ladder path within the mission's search area. Path-Generator also reads in and stores standard mission waypoints into a single list of waypoints which can be read by QGroundControl. The program will then prompt the user for a decision on whether the search area or the mission waypoints will be executed first. This function was desired in order to achieve the optimal mission time and minimum fuel usage after deciding which runway would be used for takeoff after evaluating wind conditions on the flight line. These two lists are then pushed to Path-Generator's formatting function in the appropriate order. Finally, the program will publish the final path to a .mission file in a JSON format which the flight software, QGroundControl, is able to read. After successfully writing the mission, the mission is then pushed to the UAS where the Pixhawk will begin its processing of the flight.

Telemetry

To communicate with the UAS, three separate links were established. The primary communication method for the UAS employed the 915 MHz band which utilizes frequencies from 902-928 MHz and was responsible for relaying telemetry data between the GCS and the UAS. This data includes all of the telemetry collected from the Pixhawk's built-in sensors and other sensors onboard such as the equipment responsible for airspeed readings. This data is packaged as MAVLink messages, which is a lightweight communication method that converts data from the XML format to language specific source code. The 915 MHz band is also utilized as the primary communication method from the GCS to the UAS regarding mission objectives in the form of MAVLink waypoint and event messages created by QGroundControl. The second link established with the UAS is over the 2.4 GHz band which allows for manual takeover of the UAS by a safety pilot using the Futaba T10J in case of data link loss or autopilot failure. Also controlled by the 2.4 GHz link and the Futaba T10J is a fibre-optic killswitch for the UAS's engine in case of complete system failure.

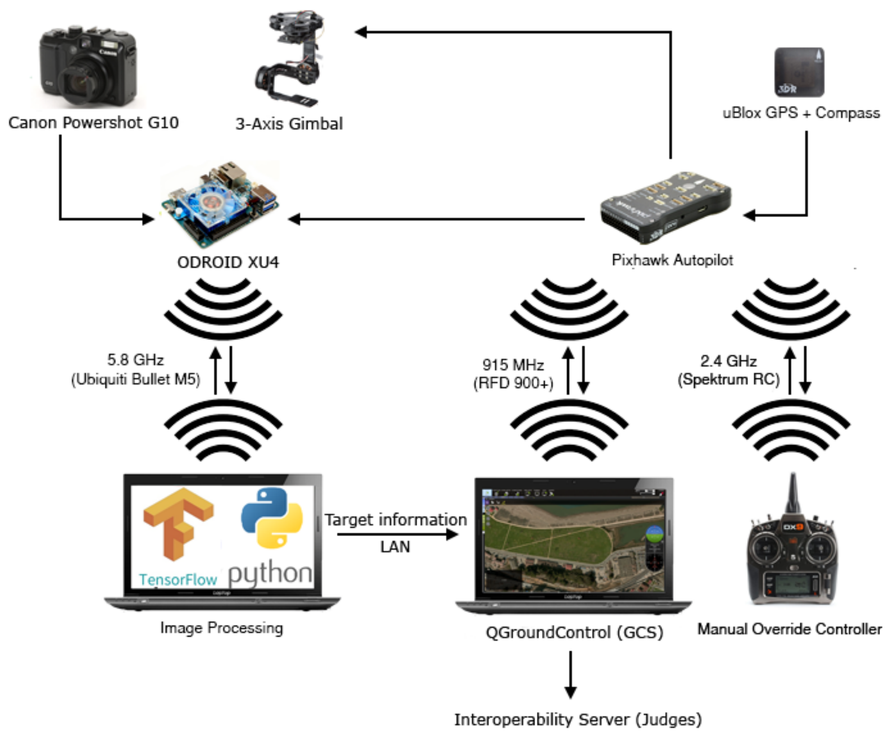

Figure 3. Communication system overview

A third communication link was established between the UAS's on-board imaging computer and the classification ground station. The connection consisted of an Ubiquiti Bullet M5 with an omni-directional antenna on-board and an Ubiquiti Nanostation loco M5 on the ground. These devices utilized the 5 GHz band with channels operating in the frequency range of 5170 to 5825

MHz. These transceivers connect to their respective machines via ethernet, so the established link will make the devices appear to be on the same Local Area Network.

Robot Operating System

Networking for all communications throughout and between the UAS and ground station were managed by the Robot Operating System (ROS), a framework that simplifies communication between processes distributed across different machines. In addition to abstracting most of the networking details away from the software developers and significantly reducing the imaging system's development time, the ability to automatically monitor and restart processes in case of failure, and the ability to easily communicate with the Pixhawk autopilot, ROS also provided a few features which were invaluable to this particular application.

One useful feature of ROS was its ability to automatically route data. Aside from simplifying the communication between the on-board computer and the classification ground station, it helped increase the on-board image processing stack's throughput by spawning multiple instances of the image processing routine and intelligently balancing workload between nodes. Aside from a slightly longer delay between any given input and its corresponding output, the end result was similar to enabling the use of multiple threads in the sense that multiple images could be processed simultaneously, but the developers never had to face the design challenges associated with multithreading. Furthermore, this communication structure required no additional bandwidth from the 5 GHz wireless data link due to ROS' peer-to-peer model for establishing routes for data [4].

Another useful feature that ROS provided was protection against brief network interruptions. In the case of short network interruptions, ROS can automatically store any data that failed to transmit correctly in a queue. Once the network becomes healthy, ROS then transmits all of the data to any machines affected by the outage. Due to the high likelihood of network interruptions in the imaging system, this helped to ensure that all packets containing target data would be received by the classification ground station.

RESULTS AND CONCLUSION

Arizona Autonomous successfully competed at this year's AUVSI SUAS competition, receiving an overall placement of 37 out of more than 50 teams internationally. The team's performance was negatively impacted by several unexpected technical complications, such as inhibitively high GPS noise, mechanical failure in the propellor mounting bolts, and a software synchronization error. With the exception of these technical difficulties, Arizona Autonomous' participation in this year's SUAS competition demonstrated some of the features offered by the team's ROS based framework, and the issues experienced require only nominal fixes for correction. Overall, the 2017 AUVSI SUAS competition proved to be a highly educational experience and allowed the team to collect valuable data for future development efforts.

ACKNOWLEDGEMENTS

The Arizona Autonomous Vehicles Club would like to thank our faculty advisor **Professor Michael Marcellin** for his endless support, patience, and guidance. We would also like to thank him for being our safety pilot throughout the year and for performing outstanding emergency landings on separate occasions. Additionally, we would like to thank all of our team members for their hard work, dedication, and perseverance. We would also like to thank the members who attended the 2017 AUVSI SUAS Competition. Lastly, we would like to thank the University of Arizona for their support.

REFERENCES

1. "2017 Rules for the 15th Annual Student UAS (SUAS) Competition." *AUVSI SUAS Competition*. AUVSI Seafarer Chapter, 12 June 2017. Web. <http://www.auvsi-suas.org/static/competitions/2017/auvsi_suas-2017-rules.pdf>.

2. "The New Bill Hempel's "Super Decathlon"." Bill Hempel's Team Edge! - Super Decathlon. R&D VISIONS, n.d. Web. 20 Aug. 2016. <http://www.billhempel.com/decathlon/>.

3. "Home." Home - Pixhawk Flight Controller Hardware Project. N.p., n.d. Web. 26 Nov. 2016. <https://pixhawk.org/>.

4. “ROS/Introduction.” ROS/Introduction - ROS Wiki. *Open Source Robotics Foundation.* n.d. Web. 22 May 2014. <http://wiki.ros.org/ROS/Introduction>