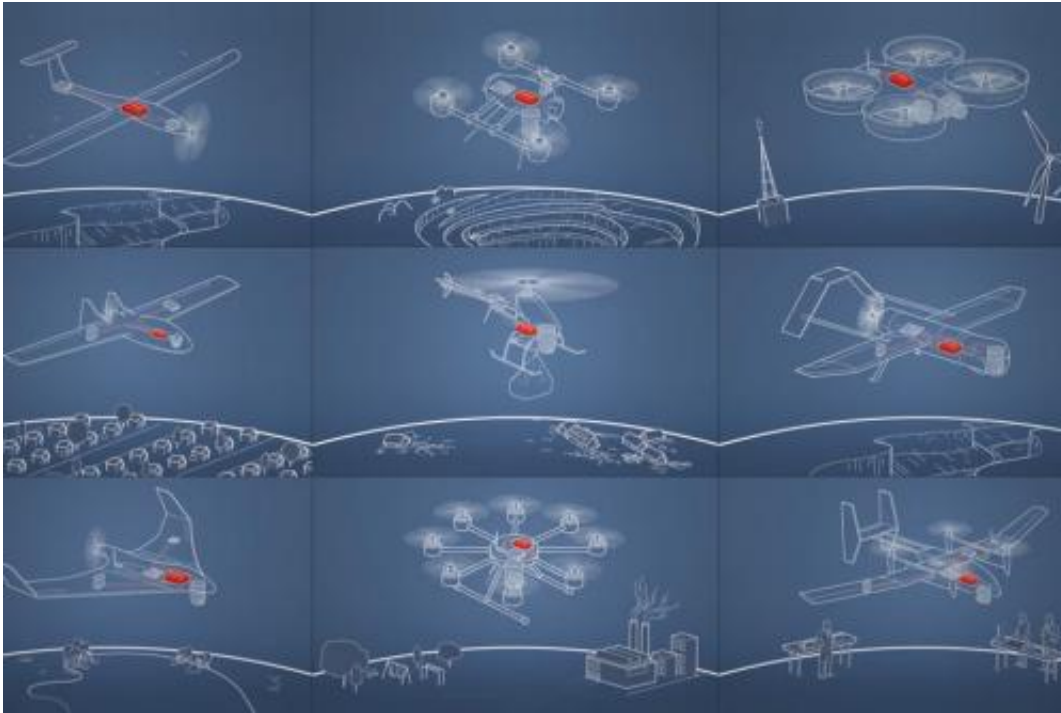


Making drones more customizable

September 12 2014, by Rob Matheson



Airware's autopilot device (red box) can be installed in commercial drones to collect data from various components, such as sensors, cameras, actuators, and communication devices. Software lets manufacturers pick and choose which components to add, and lets users plan and monitor missions in real time.

A first-ever standard "operating system" for drones, developed by a startup with MIT roots, could soon help manufacturers easily design and customize unmanned aerial vehicles (UAVs) for multiple applications.

Today, hundreds of companies worldwide are making drones for

infrastructure inspection, crop- and livestock-monitoring, and search-and-rescue missions, among other things. But these are built for a single mission, so modifying them for other uses means going back to the drawing board, which can be very expensive.

Now Airware, founded by MIT alumnus Jonathan Downey '06, has developed a platform—hardware, software, and cloud services—that lets manufacturers pick and choose various components and application-specific software to add to commercial drones for multiple purposes.

The key component is the startup's Linux-based autopilot device, a small red box that is installed into all of a client's drones. "This is responsible for flying the vehicle in a safe, reliable manner, and acts as hub for the components, so it can collect all that data and display that info to a user," says Downey, Airware's CEO, who researched and built drones throughout his time at MIT.

To customize the drones, customers use software to select third-party [drone](#) vehicles and components—such as sensors, cameras, actuators, and communication devices—configure settings, and apply their configuration to a fleet. Other software helps them plan and monitor missions in real time (and make midflight adjustments), and collects and displays data. Airware then pushes all data to the cloud, where it's aggregated and analyzed, and available to designated users.

If a company decides to use a surveillance drone for crop management, for instance, it can easily add software that stitches together different images to determine which areas of a field are overwatered or underwatered. "They don't have to know the flight algorithms, or underlying hardware, they just need to connect their software or piece of hardware to the platform," Downey says. "The entire industry can leverage that."

Clients have trialed Airware's platform over the past year—including researchers at MIT, who are demonstrating delivery of vaccines in Africa. Delta Drone in France is using the platform for open-air mining operations, search-and-rescue missions, and agricultural applications. Another UAV maker, Cyber Technology in Australia, is using the platform for drones responding to car crashes and other disasters, and inspecting offshore oilrigs.

Now, with its most recent \$25 million funding round, Airware plans to launch the platform for general adoption later this year, viewing companies that monitor crops and infrastructure—with drones that require specific cameras and sensors—as potential early customers.

A company from scratch

Airware's roots date to 2005, when Downey, who studied electrical engineering and computer science, organized an MIT student team—including Airware's chief technology officer, Buddy Michini '07, SM '09, PhD '13—to build drones for an intercollegiate competition.

At the time, drones were primarily used for military surveillance, powered by a "black box" that could essentially fly the drones and control the camera. There were also a handful of open-source projects—made by hobbyists—that let people modify drones, but the code was unreliable when tweaked. "If you wanted to do anything novel, your hands were tied," Downey says.

The group's decision: build a drone from scratch. But their advisor, Jonathan How, a professor of aeronautics and astronautics who directs of the Aerospace Controls Laboratory, told them that required too much time, and would cost them the competition.

"We said, 'You're right, but we're MIT students, and we'd feel better

getting last place and learning a lot doing it than winning the competition by repackaging a black-box solution," Downey says.

Sure enough, the team earned second-to-last place. "But we learned that black-box solution didn't work if you're trying to address new applications, and the open-source wasn't reliable even though you could change the software," Downey says.

A five-year stretch at Boeing—as an engineer for the U.S. military's A160 Hummingbird UAV and as a commercial pilot—put Downey in contact with drone manufacturers, who, he found, were still using black boxes or open-source designs.

"They were basically facing the same challenges we faced as undergrads at MIT," Downey says. Thus Airware was born in 2010—first run only by Downey, then with Michini and a team of Boeing engineers—to make a military-grade "black box" system, but whose capabilities could be tweaked and extended.

Early prototypes were trialed by How's group at MIT, before Airware entered two California incubators, Lemnos Labs and Y-Combinator, in 2013. Since then, they've raised \$40 million from investors and expanded their team from five to more than 50 employees. "The last 18 months has been a rapid rise," Downey says.

Not much of the early MIT drone designs made it into the final Airware platform. "But building that early drone at MIT, and having the idea to leverage an enterprise-grade platform that you can extend the capabilities of, very directly became what Airware is today," Downey says.

"The DOS for drones"

Today, Downey says, the development of a standard operating system for drones is analogous to Intel processors and Microsoft's DOS paving the way for personal computers in the 1980s. Before those components became available, hobbyists built computers using software that didn't work with different computers. At the same time, powerful mainframes were only available to a select few—and still suffered software-incompatibility issues.

Then came Intel's processors and DOS. Suddenly, engineers could build computers around the standard processor and create software on the operating system, without needing to know details of the underlying hardware.

"We're doing the same thing for the drone space," Downey says. "There are 600 companies building differing versions of drone hardware. We think they need the Intel processor of the drones, if you will, and that operating system-level software component, too—like the DOS for drones."

The benefits are far-reaching, Downey says: "Drone companies, for instance, want to build drones and tailor them for different applications without having to build everything from scratch," he says.

But companies developing cameras, sensors, and communication links for drones also stand to benefit, he adds, as their components will only need to be compatible with a single platform.

Additionally, it could help the Federal Aviation Administration (FAA) better assess the reliability of drones; Congress recently tasked the agency with compiling UAV rules and regulations by 2015. This could also help promote commercial drone use in the United States, which lags behind other countries around the world, primarily in Europe, Downey says.

"Rather than see a world where there's 500 drones flying overhead, and every drone has different software and electronics, it's good for the FAA if all of them had reliable and common hardware and [software](#)," he says. "We think it's valuable for everybody."

This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.

Provided by Massachusetts Institute of Technology

Citation: Making drones more customizable (2014, September 12) retrieved 25 April 2024 from <https://phys.org/news/2014-09-drones-customizable.html>

| |
|--|
| <p>This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.</p> |
|--|