Linux and Dronecode
Lucas De Marchi, Intel
ELC 2016
Agenda

- Dronecode and Ardupilot
- Linux boards: why?
- Linux boards: where and how?
Linux and the future of drones

By Nathan Willis
October 7, 2015

At the 2015 Embedded Linux Conference Europe in Dublin, Lucas De Marchi presented an update on recent developments in the field of Linux-based drones, including a look at where things are headed in the near future.

De Marchi works on the ArduPilot flight-control package, which is one of several open-source drone projects under the umbrella of the Dronecode organization. He identified himself as a relative newcomer to drones, having only started work on ArduPilot in late 2014, but said he has several years of experience working on realtime embedded Linux.

Drones are, first and foremost, hardware projects, regardless of whether they are fixed-wing airplane designs or are quadcopters (or any other rotor-wing design). The same basic software architecture applies in either case, though. The flight-control package (such as ArduPilot) is tasked with keeping the drone aloft and in its intended position. That requires monitoring input from an array of sensors (GPS, altimeters, accelerometers, gyroscopes, magnetometers, and so on). The flight controller's "output" (so to speak) is commands that manipulate the drone's engines: pulse-width modulation (PWM) to control motors, commands sent to a UART or CAN Bus controller, and so forth. A tertiary input is the radio link that most drones have to a ground-control station or RC controller, but that is about the full extent of a flight controller's worries.

Recent developments

The year-to-year changes that can be observed in drones can happen in hardware or in software. In hardware, De Marchi noted that the first ArduPilot code ran on Arduino-compatible microcontrollers. The boards supported have grown in complexity in the years since, adding more processing power; more and better (usually faster) sensors, and additional memory. A beefier board can process more sensor samples, but it also allows the flight controller to use more sophisticated algorithms.
Dronecode

✧ Project under Linux Foundation
✧ Flight stacks
✧ Ground Control Station
✧ Communication protocols: mavlink, rtps, etc
✧ ...
Ardupilot

New boards (since ELCE 2015):

Erle-Brain2, BH-Hat, Navio2, PXFmini, Minlure, Bebop2, qurt, qflight
Linux boards: why?
Linux boards: why?
Linux boards: why?
Linux boards: why?

✧ Smart devices, drones
✧ Memory, flash, CPU
✧ Development convenience
✧ HW abstraction
✧ Security
✧ Connectivity
✧ More features!
Linux boards: why?

✧ Smart devices, drones
✧ Memory, flash, CPU
✧ Development convenience
✧ HW abstraction
✧ Security
✧ Connectivity
✧ More features!
Linux boards: where?

- State estimation
- Control
- Mapping
- Planning
Linux boards: where?

Low-level flight stack:
- Sensor sampling
- State estimation
- Control
- Pilot input

High-level flight stack:
- Mapping
- Planning
- Apps, cloud, ...

Pilot input
Linux boards: where?

Low-level flight stack:
- Sensor sampling
- State estimation
- Control
- Pilot input

Sometimes the separation is blurry

High-level flight stack:
- Mapping
- Planning
- Apps, cloud, ...

Control

Sensor sampling

Optical Flow

Vision-based Attitude Estimation

...
Linux boards: where?

✧ Task offload / partitioning
Drivers in userspace:
- New APIs using bus abstraction
- Sensors are grouped per bus
- Spawn a thread per bus

Linux boards: how?
Linux boards: how?

- Drivers in kernel: IIO subsystem
  - High level sensor abstraction
  - Some driver need tweaking due to high sample rate
  - ADC
  - IMU, Baro, ...

COMING SOON
Linux boards: how?

✧ I/O restrictions
✧ RT requirements
✧ Additional tasks
Linux boards: how?
I/O restrictions

✧ External microcontroller
  ✧ Additional ADC / I2C / SPI
  ✧ RCInput / RCOoutput
  ✧ Kernel abstraction vs userspace communication

✧ Internal microcontrollers
  ✧ PRU
  ✧ Sensor Hubs

✧ DSP

✧ FPGA
Linux boards: how?

RT requirements

✧ Good drivers
✧ Offload specific tasks
✧ Offload the flight stack?
  ✧ May not be ideal
  ✧ May be too much
Demo 1
Linux boards: how?
Additional tasks

✧ Video streaming
✧ Optical Flow
✧ Planning
✧ Mapping
✧ ...

---

**Intel**
Q & A