Linux - the future for drones

Lucas De Marchi, Intel

ELCE 2015

Who am I

- Software developer
- Contributed to several open source projects throughout the Linux stack
- Recently joined projects under the Dronecode
- Linux maintainer for Ardupilot

Agenda

- Dronecode
- Hardware evolution
- Software evolution
- Handling the complexity and scaling
- Future

Dronecode

“If you want to go **quickly** go alone,

if you want to go **far** go together”

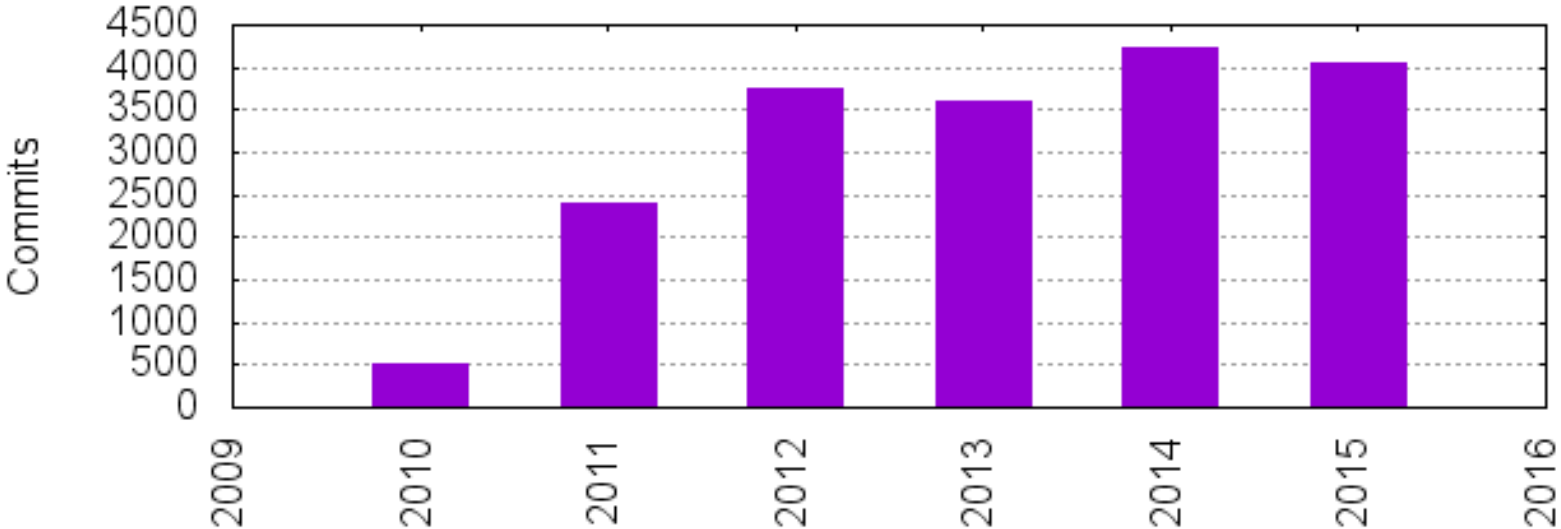
Dronecode

- 40+ members
- Composed of several projects, including

  2 flight stacks
- Contributions to each of them increasing

Dronecode

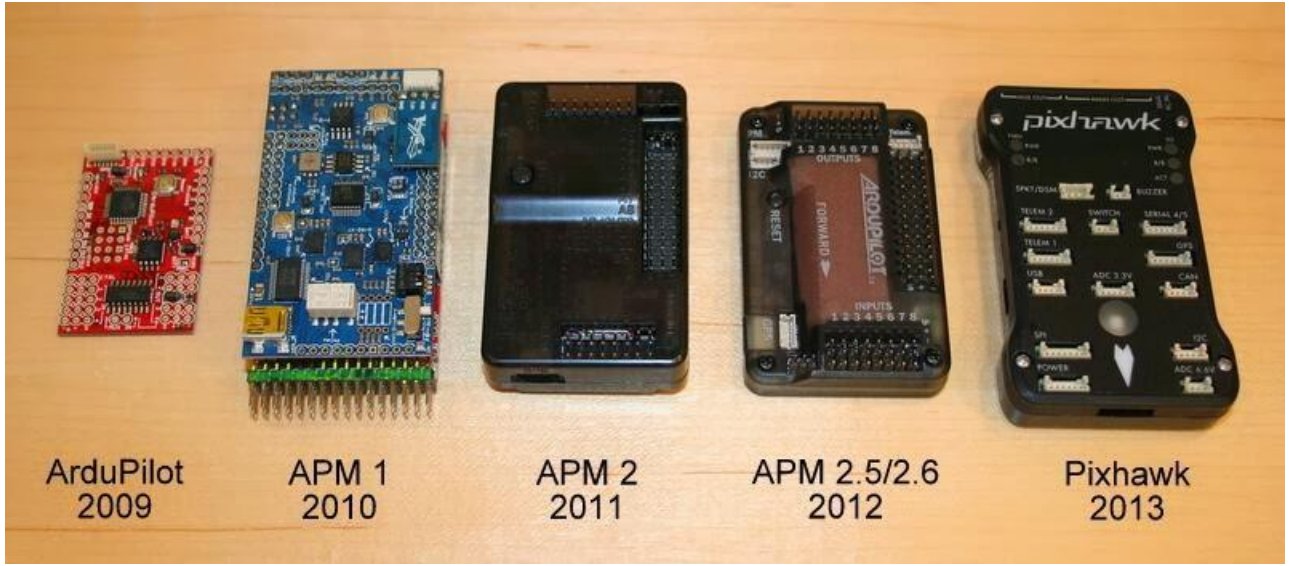
Ardupilot



Hardware evolution

Hardware evolution

Ardupilot



Pixhawk2
2015

Hardware evolution

Ardupilot - Linux Boards

It all started with a single board,

with a specific set of sensors in a daughter board:

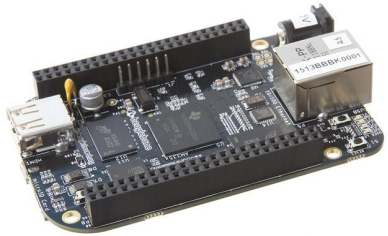
BeagleBone Black + PXF cape

Hardware evolution

Ardupilot - Linux Boards

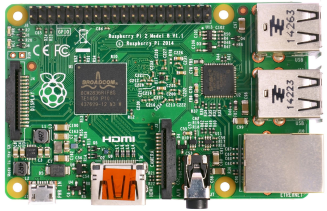
- PXF 2014
- ErleBoard 2014
- BBBMini 2015

Expansion boards for BeagleBone Black



- Navio/Navio+ 2014
- Raspilot 2015
- ErleBrain2* 2015
- VR Brain 5 LX* 2015

Expansion boards for Raspberry Pi



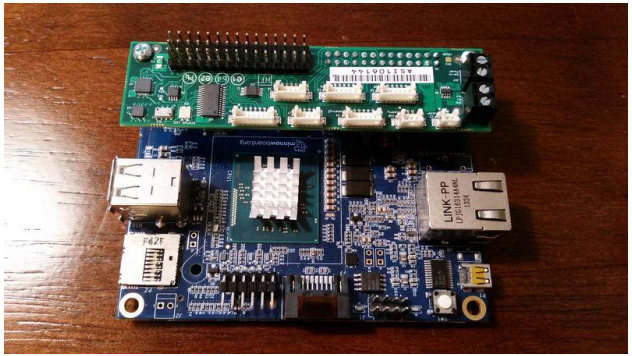
* Not merged yet

Hardware evolution

Ardupilot - Linux Boards

- Bebop
  - Own HW and Linux stack

- MinnowBoard Max*
  - Drone Lure with sensors

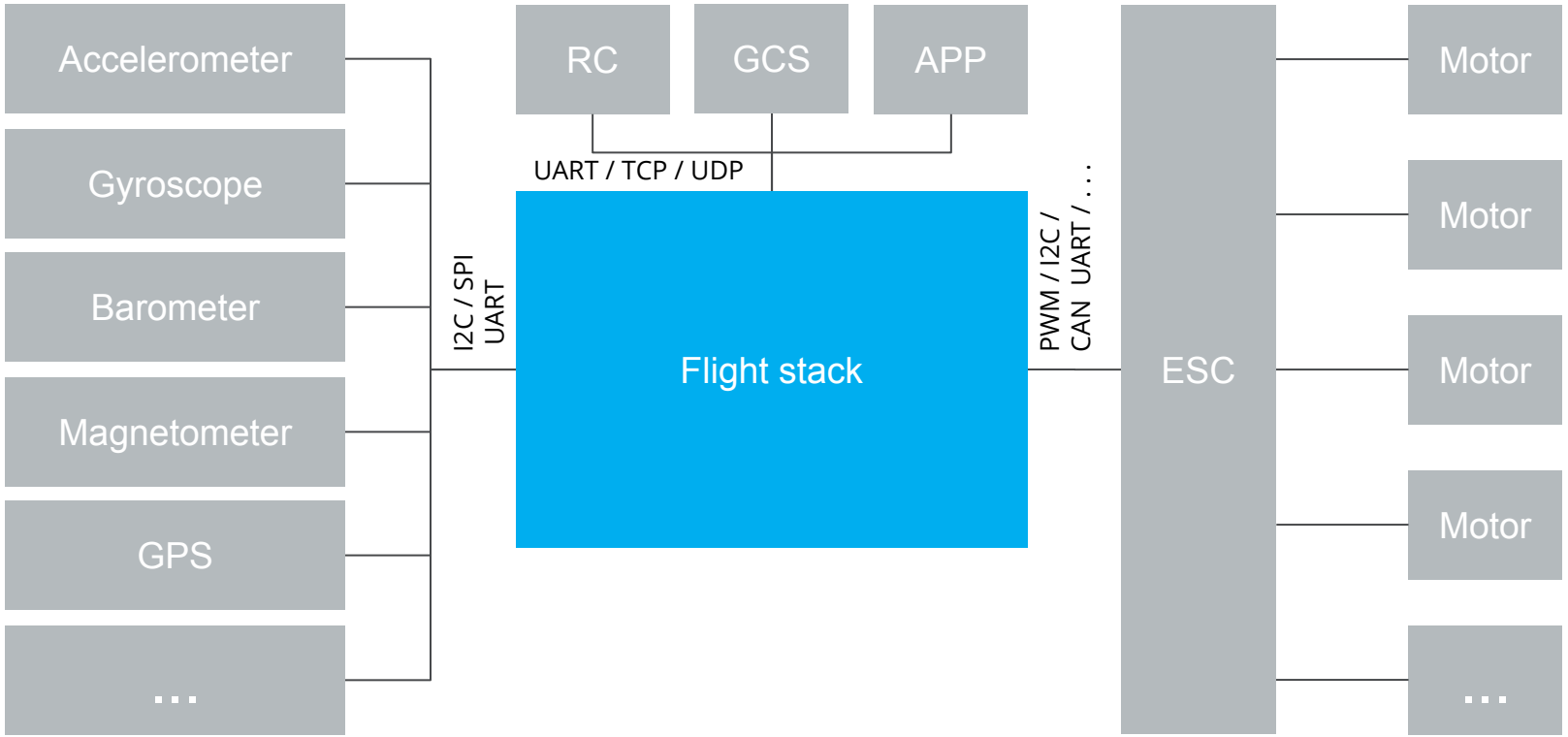


* Not merged yet

Overview how a drone works

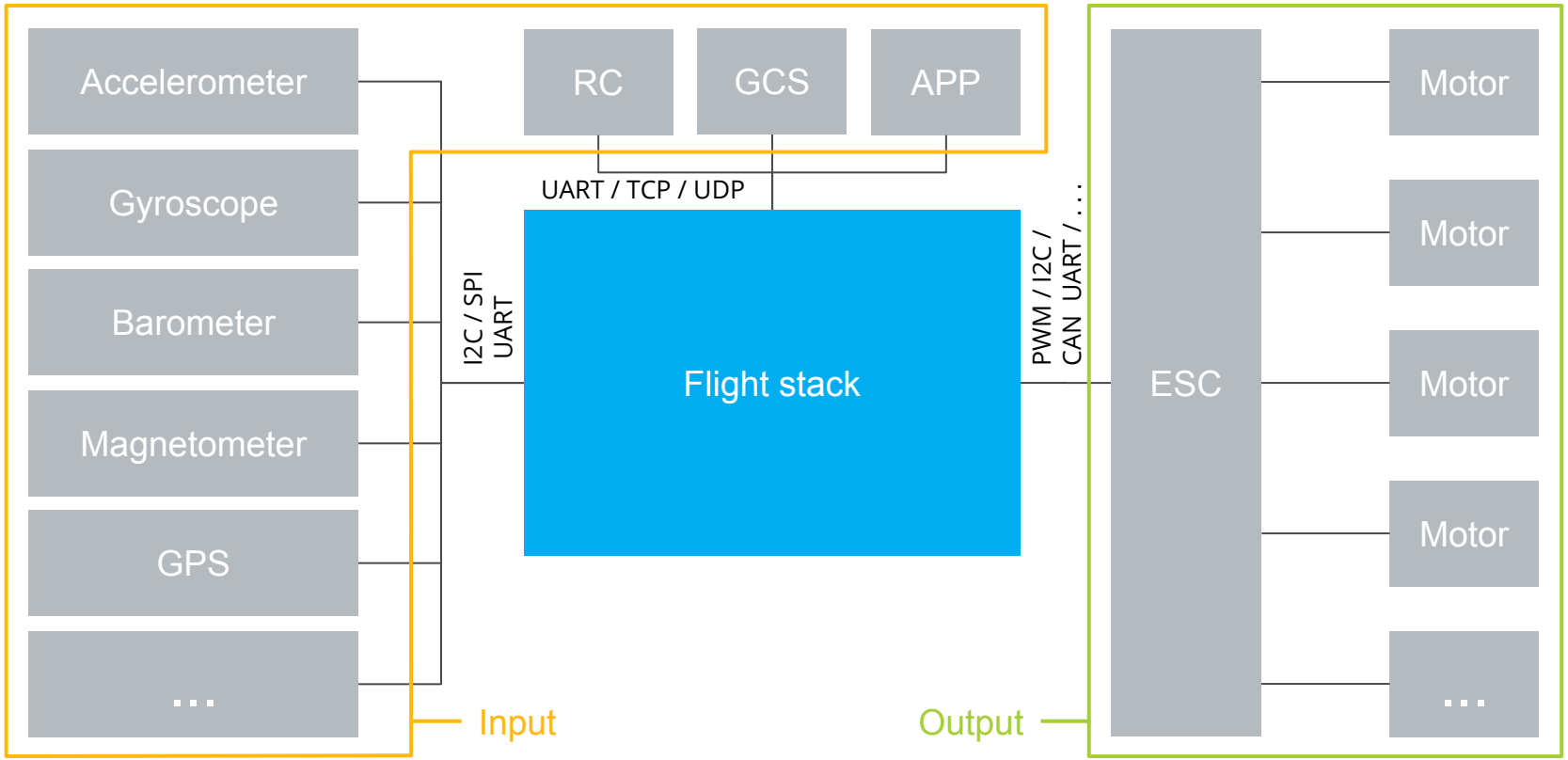
Hardware/software evolution

101 - How a drone actually works (simplified)



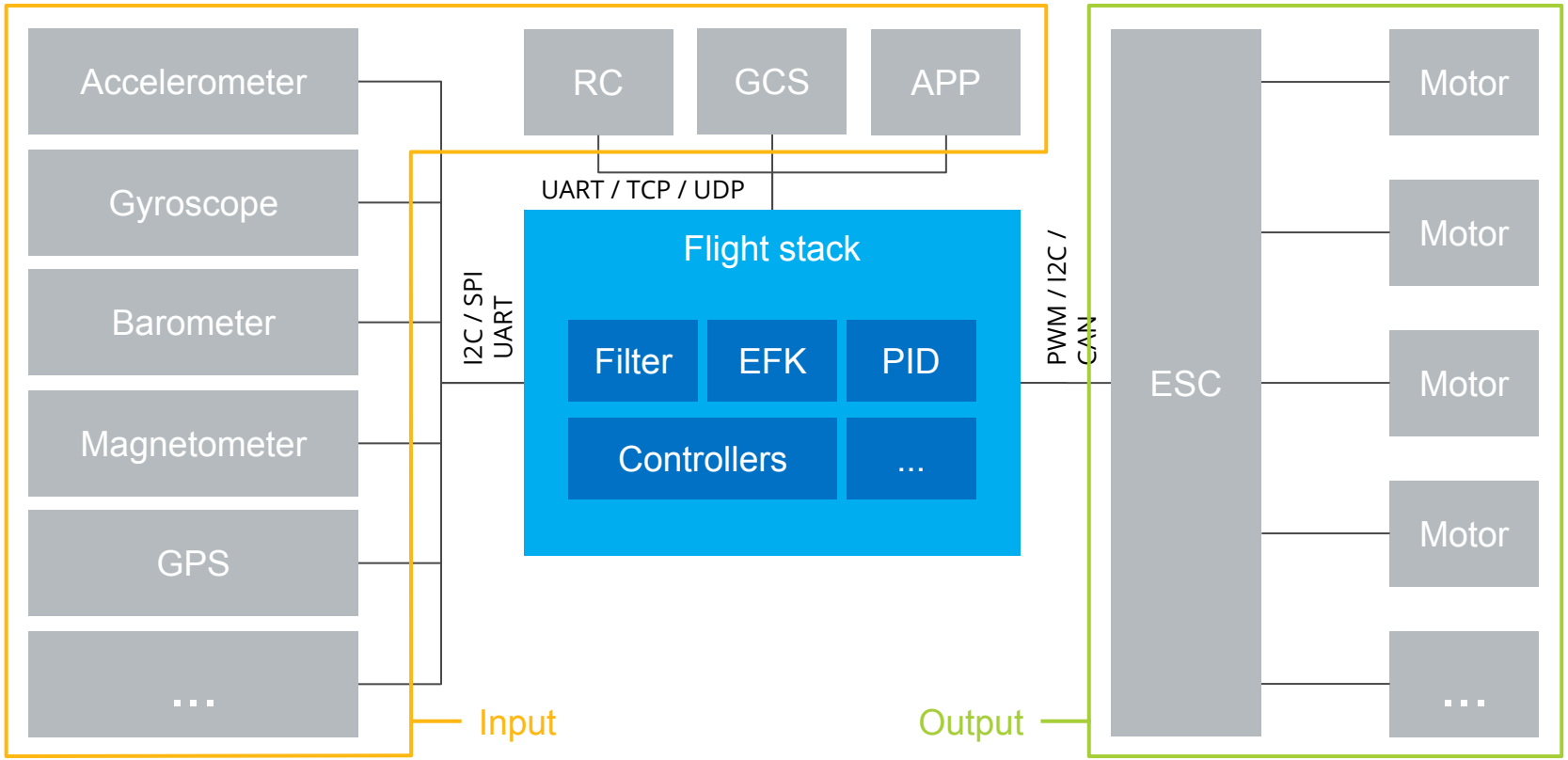
Hardware/software evolution

101 - How a drone actually works (simplified)



Hardware/software evolution

101 - How a drone actually works (simplified)



Software evolution

Software evolution

Sensors

- From few samples per second to thousands
- Redundancy
- More complex sensors
  - Lidar
  - Optical Flow
  - Depth cameras
  - Computer vision

Software evolution

"Low-level" flight stack

- Increasing accuracy (hence complexity) of control algorithms
  - E.g. the move to EKF for AHRS

Software evolution

Usages - pushing the complexity

- Photography
- Agriculture
- Survey / Mapping
- Inspection
- Deliveries
- Search and rescue

Software evolution

Outcome

- Drones becoming smarter
- Intelligence inside vs outside
- Increased CPU and memory requirements
- Need to scale for more hardware platforms

Handling the complexity and scaling

Handling the complexity and scaling

Boards in Ardupilot

- APM1 and APM2 are deprecated
  - Not enough RAM, flash and CPU anymore

Handling the complexity and scaling

Sensors in Ardupilot

- Support for more sensors, different manufacturers
- Linux boards becoming first class citizens
  - PX4-only features moving to common code
  - Linux-only features starting to appear (existing infrastructure in Linux)

Handling the complexity and scaling

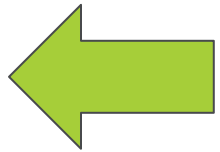
When microcontrollers are not enough anymore

- Companion computer
  - Move complex tasks to a separate Linux board
  - Move flight stack to a separate microcontroller
- Single board Linux solution
  - Both flight stack and other tasks on same board

Handling the complexity and scaling

When microcontrollers are not enough anymore

- Companion computer
  - Move complex tasks to a separate Linux board
  - Move flight stack to a separate microcontroller
- Single board Linux solution
  - Both flight stack and other tasks on same board



This is the solution taken for the Linux boards currently supported in Ardupilot

Handling the complexity and scaling

Single board Linux solution

- Realtime
- Offload specific part(s) of the stack
  - To separate microcontroller (even inside the SoC)
  - To dedicated off-the-shelf hardware

Handling the complexity and scaling

Single board Linux solution

1. Move single-digit µs precision off the CPU: PWM output, RC decoding (PPM, SBUS, DSMX), tone generator, etc.
2. Follow guidelines for RT tasks in Linux
3. Have the necessary buses exposed
4. Cheers your new Linux-based flight stack

Future

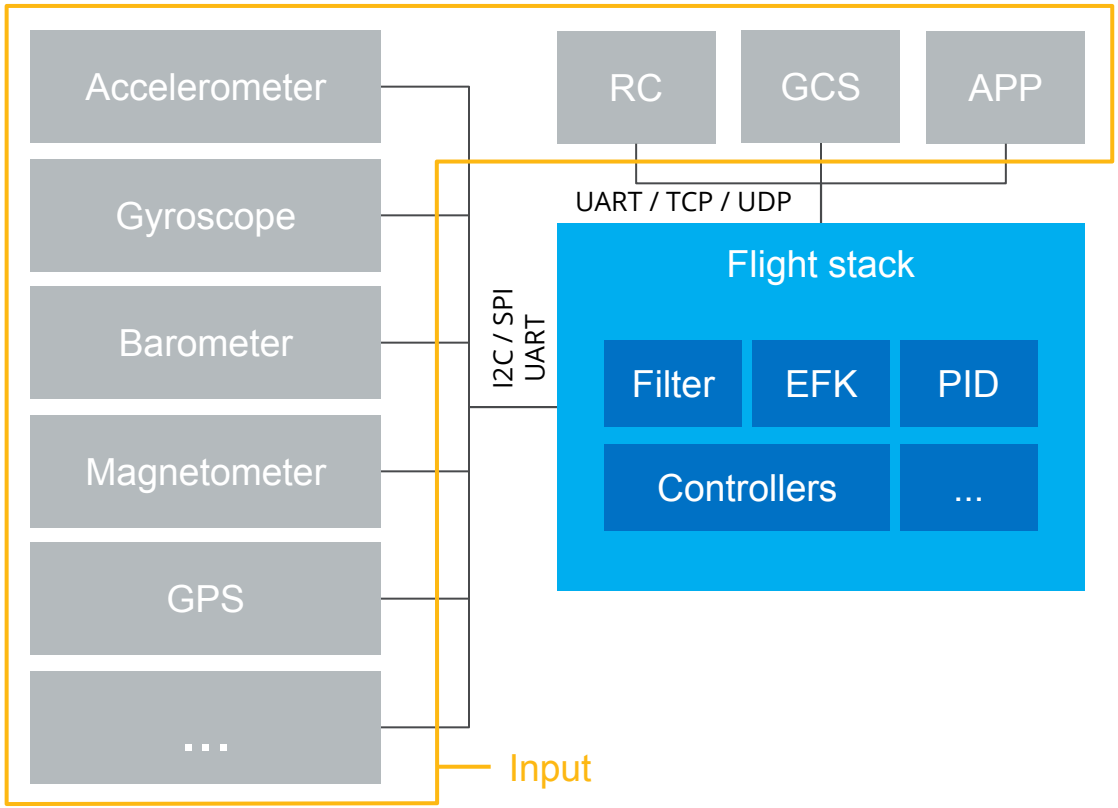
Future

Scaling for new boards

- Support for new boards (LIVE “DEMO”)
- Make adding new boards easier and scalable
  - Runtime detection / configuration
- Different platforms
- Increased complexity

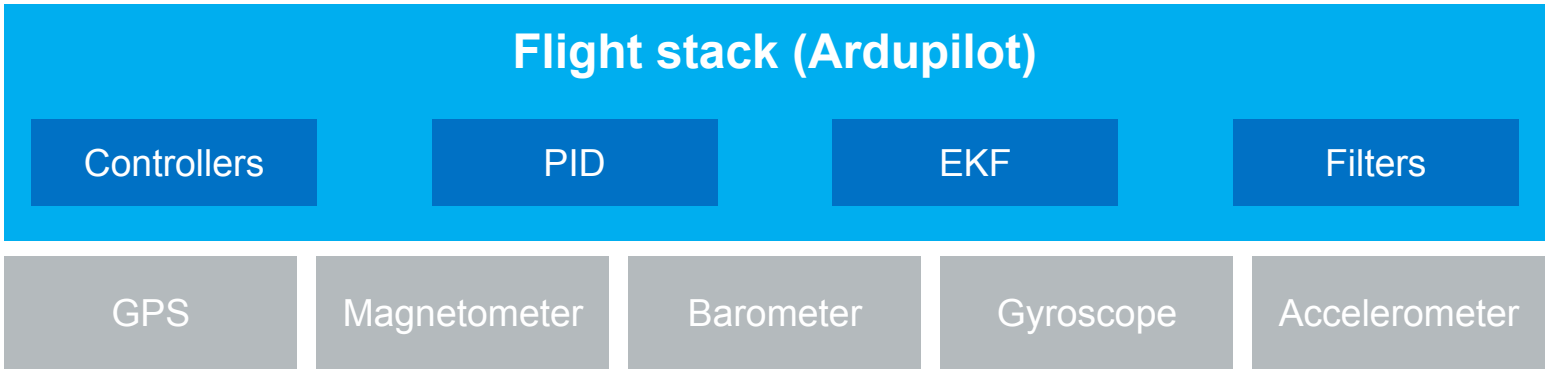
Future

Scaling for new sensors



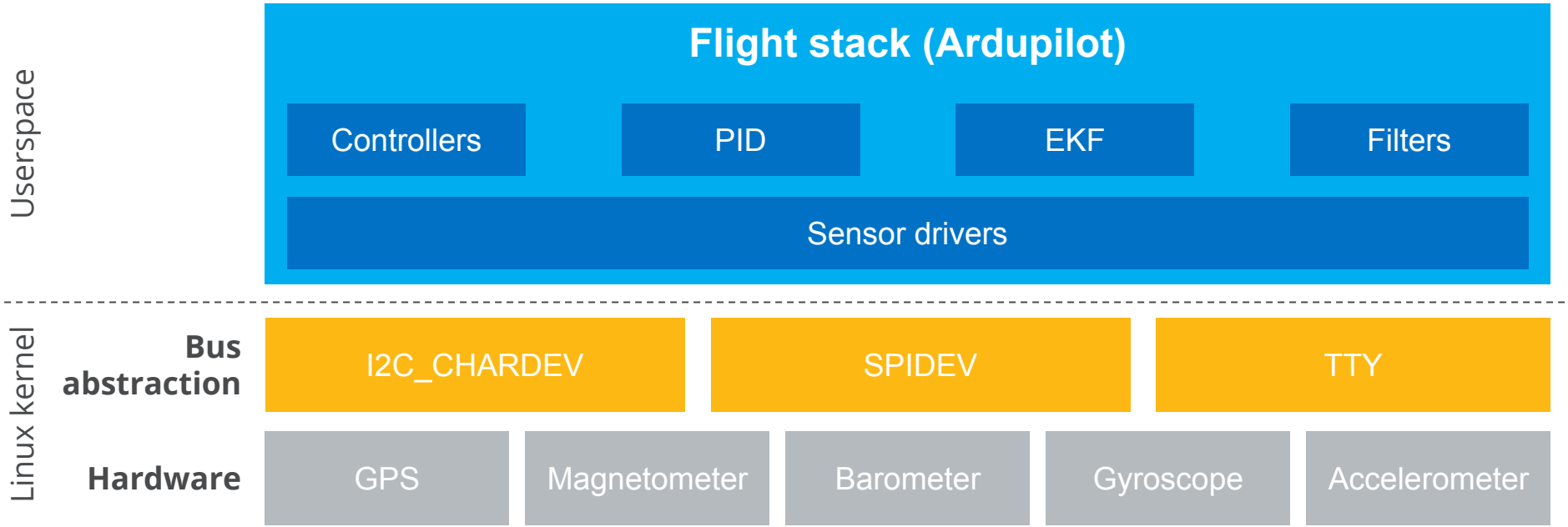
Future

Scaling for new sensors



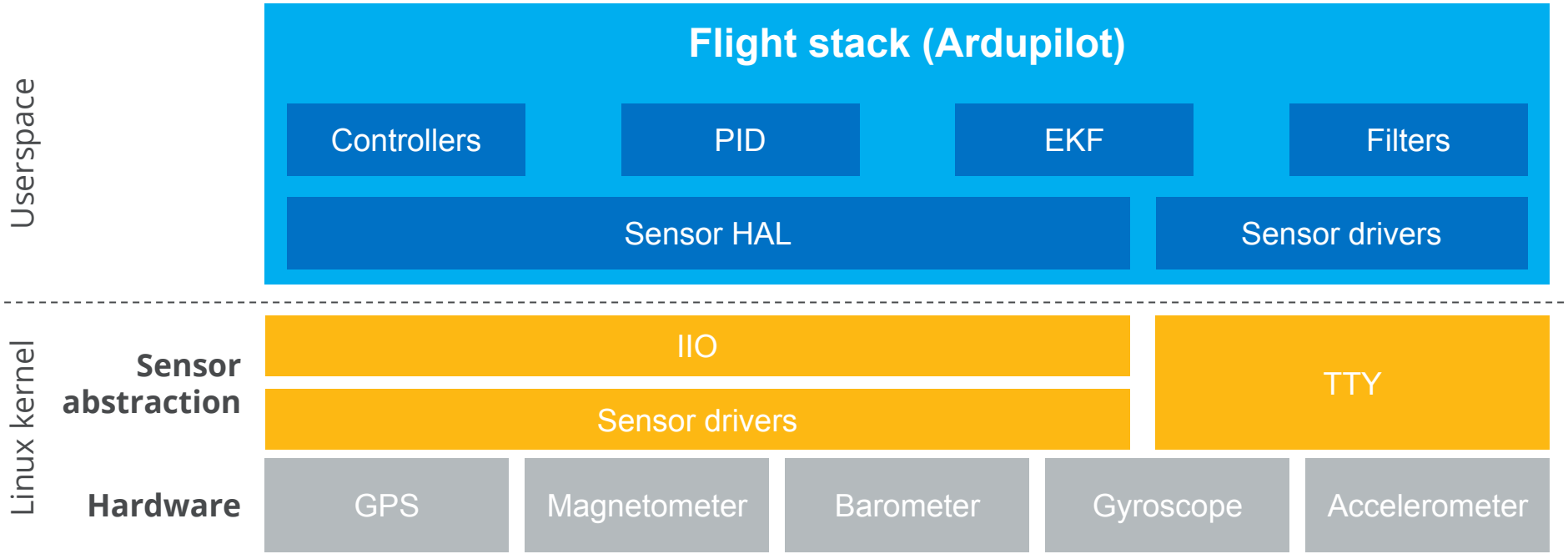
Future

Scaling for new sensors



Future

Scaling for new sensors



Future

Scaling for new sensors

Use kernel drivers (IIO subsystem)

Pros:

- Several drivers already available
- Share testing with other platforms (Linux desktop, Android)
- Reduce complexity on the flight stack
- Reduce overhead to communicate with sensor: flight stack access data stream

Cons:

- Can't share driver with other platforms (PX4 middleware / Nuttx)
- Harder to prototype new drivers
- Currently used sensors don't have kernel drivers or don't have the right interfaces

Middle ground: support both for separate buses

Future

aka dreams

- Linux boards to foster use of new algorithms
- New sensors
- Smarter autonomous drones

Wrap-up

- Drones growing in application and capabilities
- Linux provides scaling at HW and SW levels
- Sharing parts with other projects improves code quality and testability

Q&A

Links:

Dronecode: http://www.dronecode.org
Ardupilot: http://ardupilot.com/
Contact: lucas.demarchi@intel.com
Slides: conference site

drones-discuss mailing list
http://diydrones.com
Gitter Skype
IRC Mumble