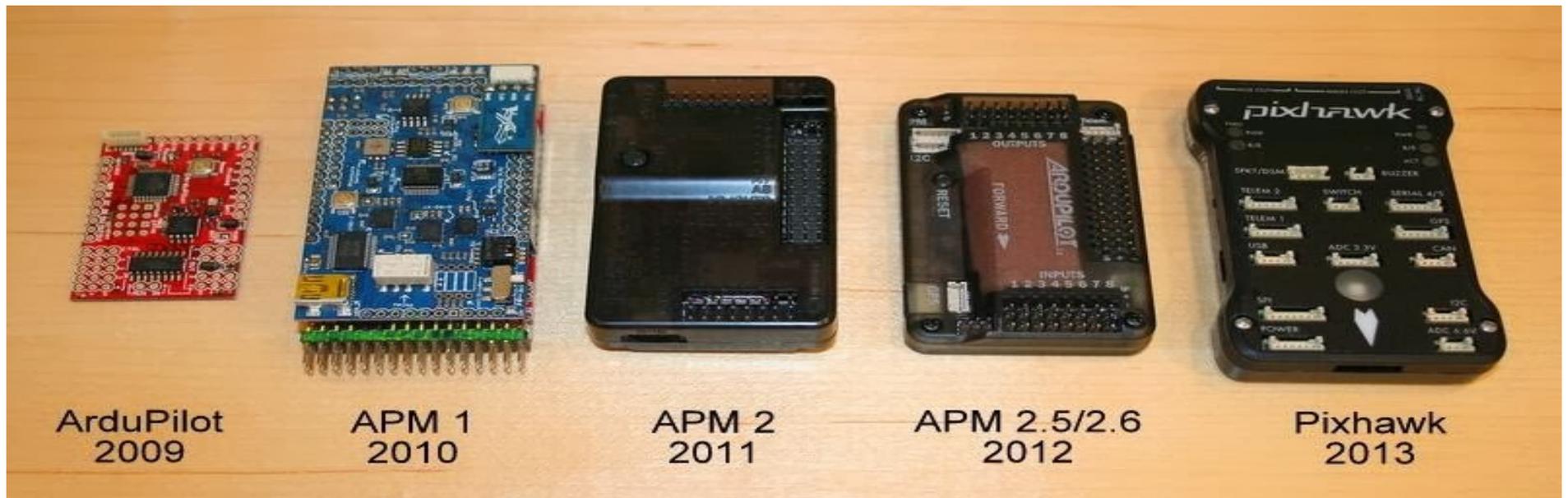


Flying with Linux

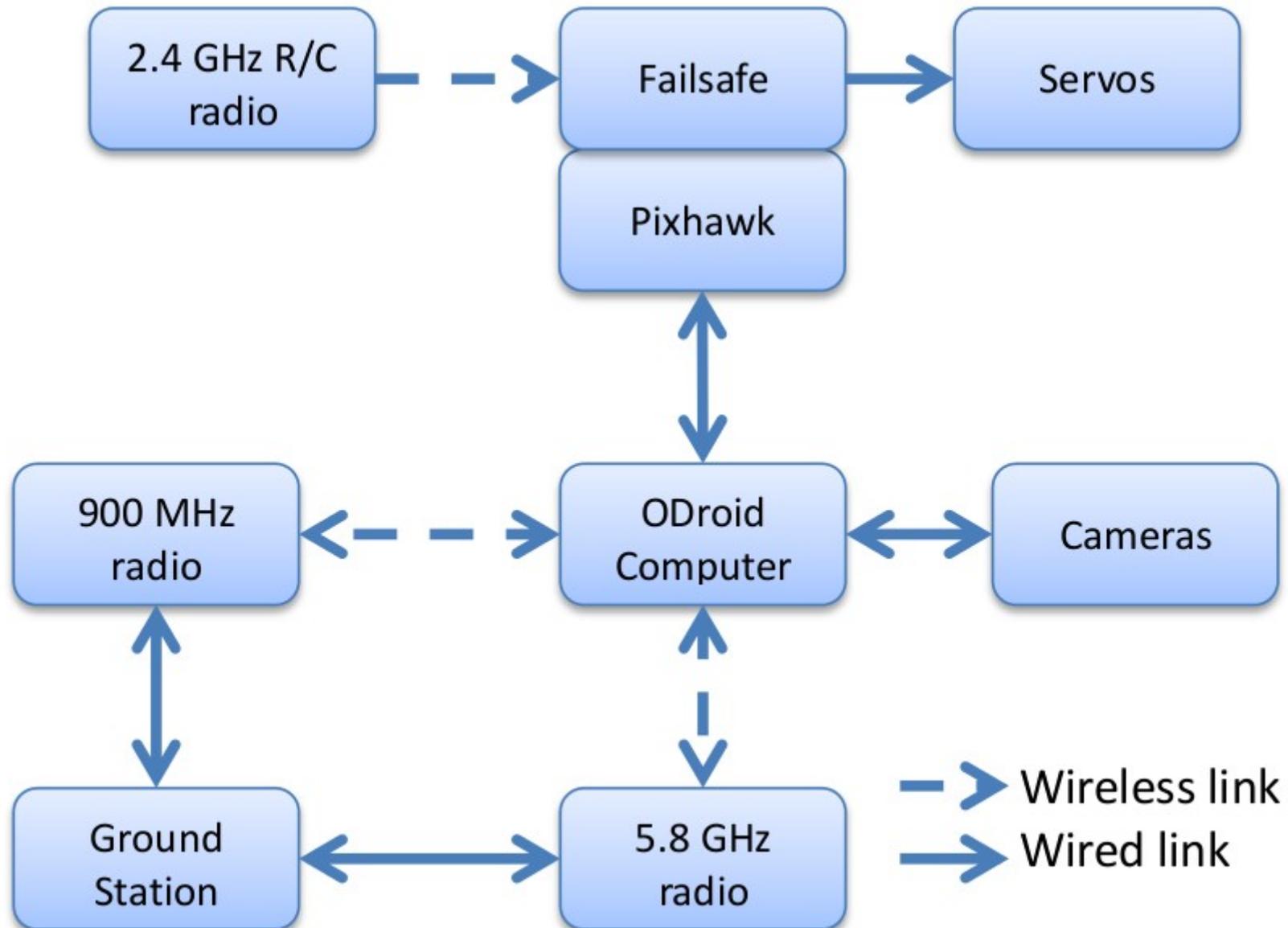
Porting an autopilot to Linux – Part 2
Andrew Tridgell

LCA last year

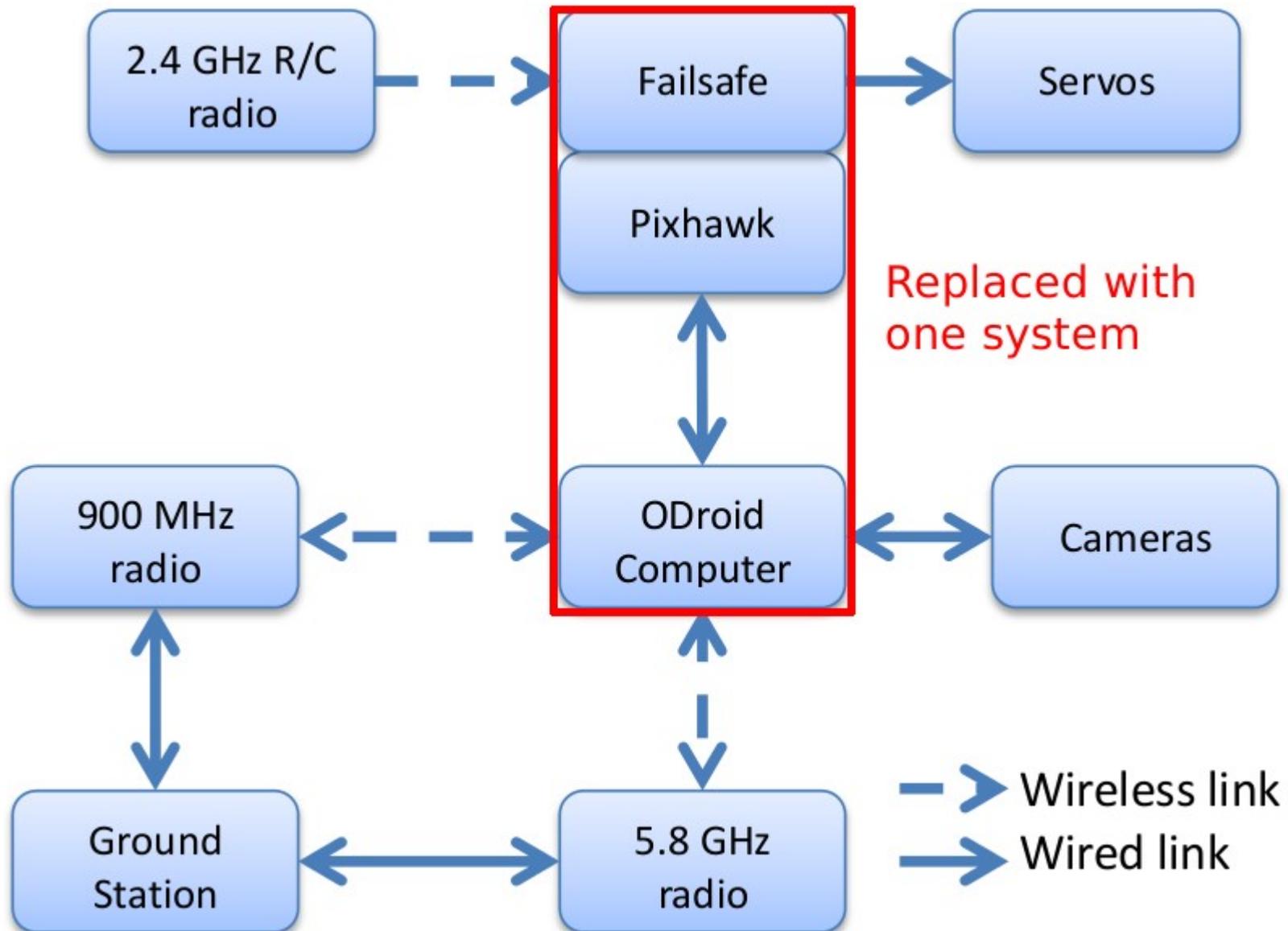
- Initial port of ArduPilot to Linux
- Base design of PXF sensor cape done, but not built yet
- Didn't know how much of the autopilot code could realistically run on Linux



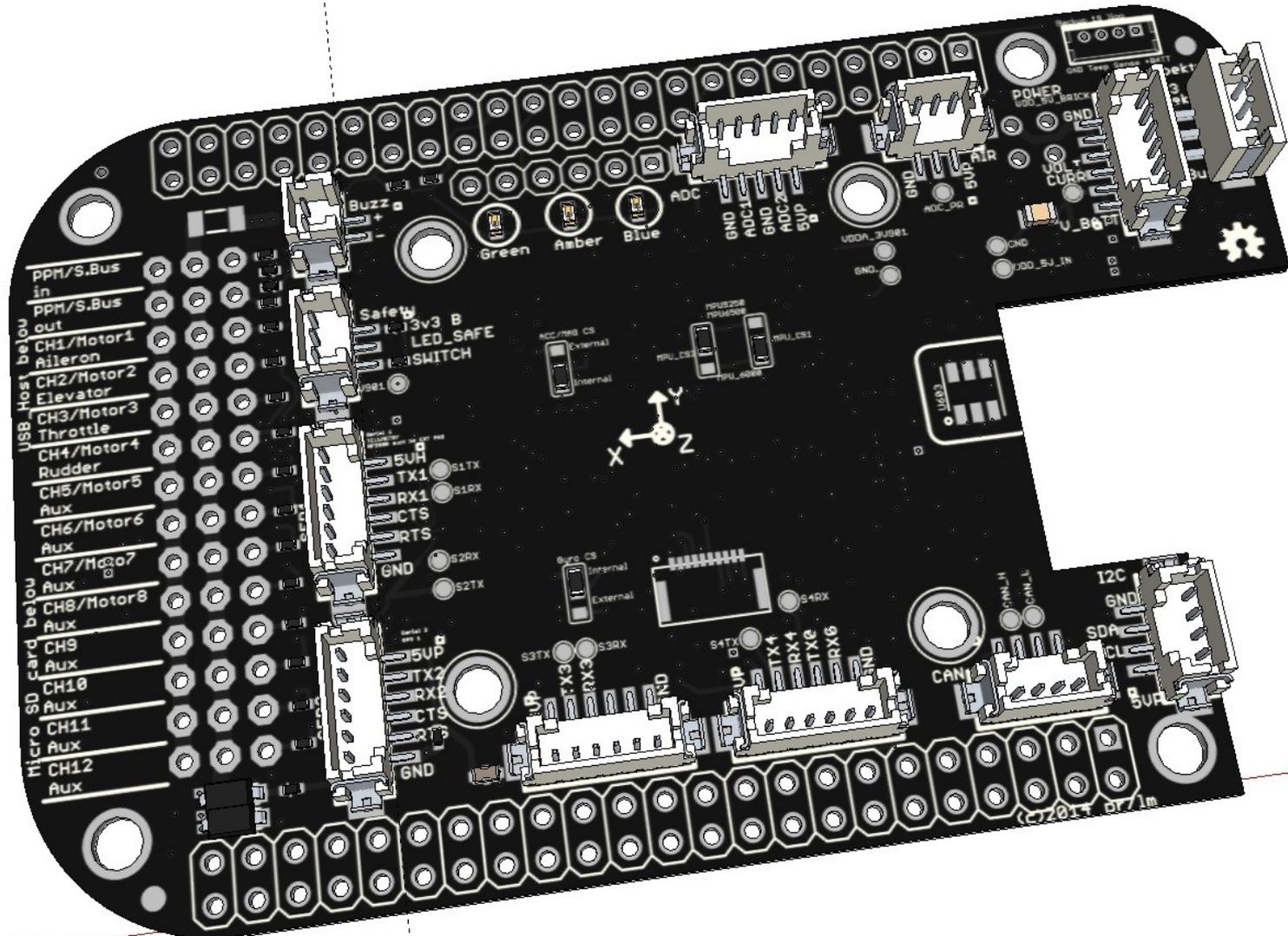
Existing System Design



New System Design



PixHawk Fire Cape



In the last year...

- ArduPilot is now regularly flying on Linux
- multiple ports of ArduPilot have been done to a variety of different boards
- Autopilots based on ArduPilot on Linux are now commercially available

Flight Demonstration

Live demo from Canberra, Australia

- Skywalker 2013 electric model
- BeagleBoneBlack with PXF cape
- ArduPilot 3.2.1
- Compiling Linux kernel while flying on same CPU

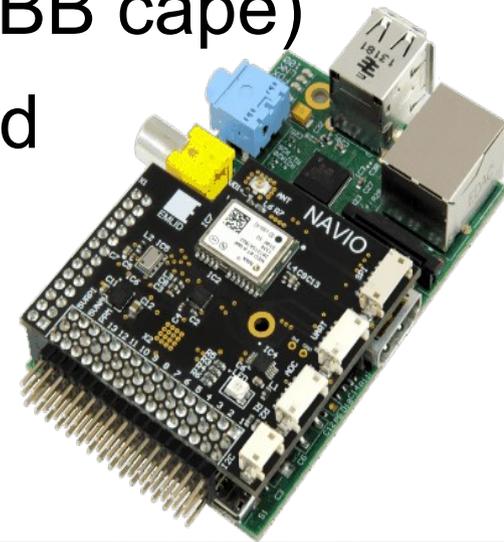


Demo Setup

- BeagleBoneBlack running Debian
 - 3.8.13-RT kernel
 - ArduPilot 3.2.1
- Sensors
 - MPU9250 accel/gyro on SPI
 - MS5611 barometer on SPI
 - Ublox Lea6H GPS on 38400 UART
 - HMC5883 compass on I2C
 - MS4525DO airspeed sensor on I2C
- IO
 - SBUS input via PRU2
 - PWM output via PRU1
 - two telemetry radios (for MAVLink and shell access)

Autopilot boards and ports

- Flying ports developed over the last year
 - PXF/BBB from 3DRobotics
 - Erle Brain from Erle Robotics (PXF on BBB cape)
 - NavIO and NavIO+ RPi capes from Emlid
- Prototype ports
 - I.MX6 port within 3DRobotics
 - Zynq ARM+FPGA port by John Williams
 - BBBMini port by Mirkix



I2C and SPI

- Fast sensors on SPI
 - using /dev/spidev interface, user-space drivers
 - works very well!
 - able to handle 4k SPI transactions per second with 25% CPU load on BBB
 - no DMA used due to DMA overhead for small transfers (typically a transfer is around 20 bytes)
- Slower sensors on I2C
 - using /dev/i2c smbus API, drivers in user space
- Why user space?
 - common drivers across multiple operating systems, using AP_HAL abstraction
- Moving to uavcan in future to replace most I2C

Scheduling

- 6 realtime (FIFO scheduled) threads
 - timer thread (1kHz timer, for regular tasks)
 - UART thread for all UART serial operations
 - RCIN thread for processing RC input pulses
 - main thread for core autopilot code
 - tonealarm thread for buzzer sounds
 - IO thread for all filesystem IO (logging, parameters and terrain data)

Scheduling Overrun

- Long scheduling overrun discovered
 - discovered while preparing for this talk
 - 11 hour test, building kernel on microSD while ardupilot running on BBB
 - 50Hz main loop, so 20ms expected loop time
 - Out of 2M loops executed, 19 were over 30ms
 - one was over 40ms – that took 1.7 seconds!
 - the challenge now is to find the cause and fix it

PRU Code

- Programmable Realtime Units
 - two PRUs on BeagleBoneBlack
 - 200MHz simple CPUs
 - access to 8k of shared memory with ARM
 - direct access to I/O pins
 - C Compiler available (not complete, but usable)

High rate timing task – RC Input

- Some tasks need microsecond precision
 - RC Input
 - PPM-SUM (multi-channel pulse width based RC-input)
 - SBUS (100kbaud inverted serial RC input)
 - DSM (115200 baud serial RC-input with framing)
 - We needed a solution that was as generic as possible
 - In AP_HAL each board calls process_rc_pulse() for each state change of pin
 - Commonly implemented using a ring buffer (pulse train)
 - Generic software decoder handles all 3 protocols in parallel
 - Decoding of multiple baud rates on one pin (protocol framing to avoid ambiguity)
 - Avoids the need for multiple UARTs for RC input

BeagleBoneBlack PRUs

- PRU1 used for RC Input
 - watches for state change on 1 pin
 - writes timing of state changes to a ring buffer
 - ARM code consumes entries from ring buffer, calling `process_rc_pulse()`
 - just 70 lines of C code on PRU
- PRU2 used for PWM output
 - shared buffer of PWM channel pulse width frequency
 - continuously reads shared buffer and updates 12 channels
 - just 235 lines of C on PRU

Outback Challenge 2014











DroneCode.org

- New umbrella organisation for free software UAV development
 - Part of Linux Foundation Collaborative Projects
 - Forum for collaboration between projects, users and companies using the technology
- First conference
 - First DroneCode conference at ELC in San Jose in March 2015
 - Come along!