

# DIY-EFIS Oshkosh 2019



**Kotuku Aerospace**  
**CanFly<sup>®</sup>**

[www.kotuku.aero](http://www.kotuku.aero)

<https://github.com/kotuku-aero/diy-efis>

# CAN and CANAerospace

- CAN developed by Bosch as a highly reliable multi-point control system. All modern vehicles use some form of CAN bus, frequently multiple.
- CANAerospace defined by Stock Flight Systems at Oshkosh 2009 as a protocol suitable for small aircraft.
- Rotax 912is uses CANAerospace to publish information system from the ECU. Stock Flight systems provides an engine monitor that connects to the ECU.
- Most EFIS vendors offer some sort of CAN support, however in general is proprietary.

# Kotuku Aerospace CanFly<sup>®</sup>

- Builds on CAN and CANAerospace. Adds important extensions:
  - Configuration Protocol
  - Standard connector pinout
  - Standard data rate
- Specifies same data rate as Rotax (125kbit/sec)
- Lycoming EDU
- AHRS
- SLCAN (serial CAN)
- Raspberry PI H.A.T
- All code and designs are open source.

# EFIS Source code

- Source code developed in C++ since 2000.
- Converted to using CANAerospace 4 years ago.
- Supported platforms in reference code is for Windows – Primarily for development  
Linux – for embedded systems or development
- Can be ported to other platforms easily.
- Supports Lawicel CANUSB, CanFly USB or Pi HAT for access to CANBus
- Reference designs are in the open-source repository.



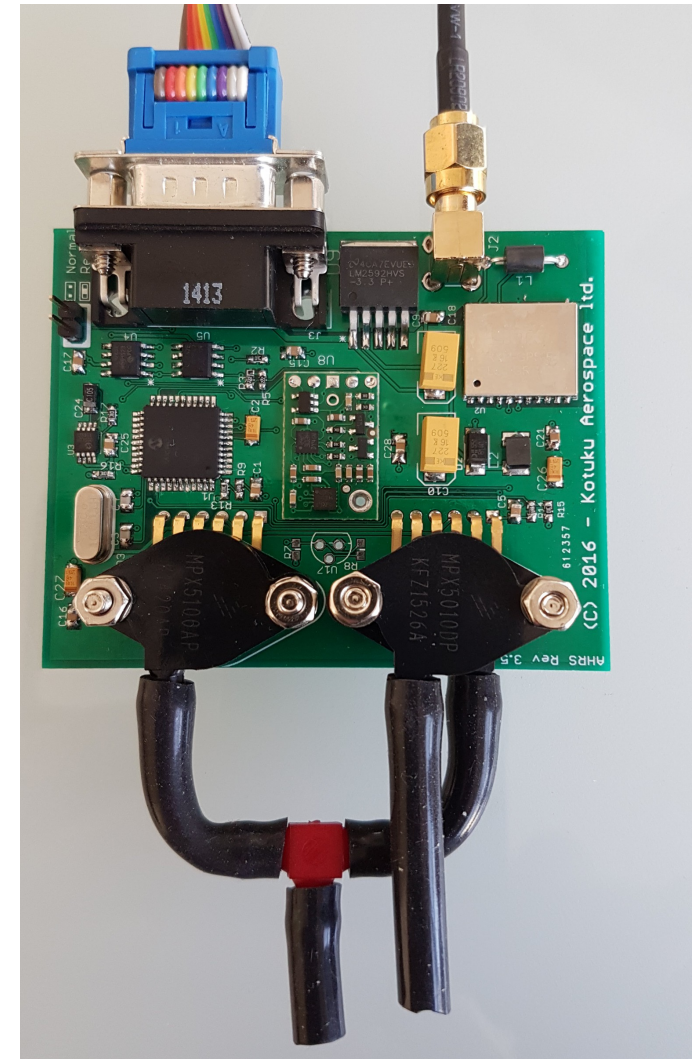
# CanFly USB

- Provides USB to CANAerospace conversion  
Serial commands compatible with  
CAN232/CANUSB from Lawicel.
- Contains terminal to allow access to CanFly  
devices for configuration.
- Panel mount version available with data  
logging and serial ports.
- Can be used with Android/iPad devices with  
a USB port, allows for flight plan uploads.



# Canfly AHRS

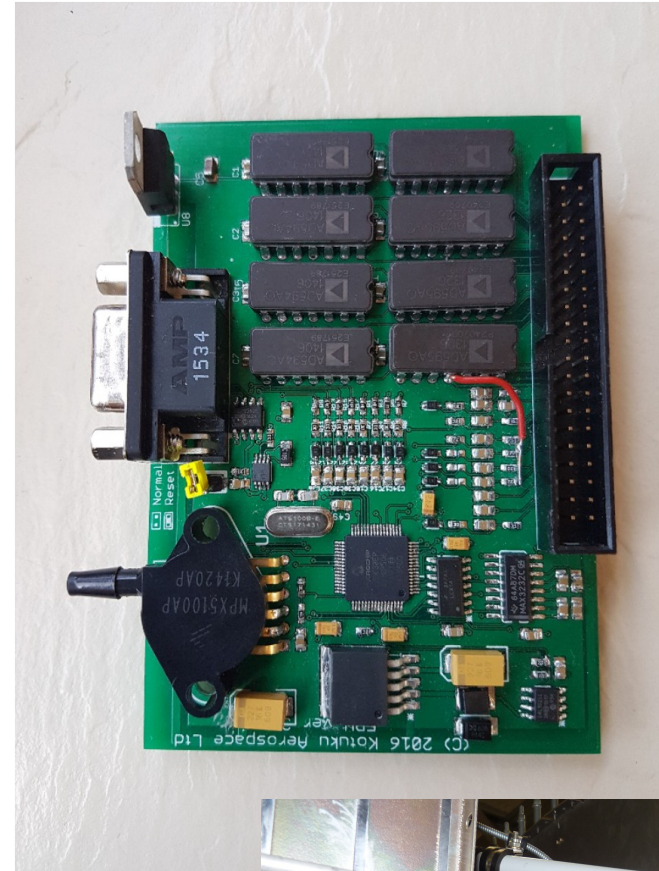
- Altitude, Direction and Position system
- 16 Channel GNSS GPS
- Altitude, airspeed sensors
- 3 axis accel, gyro, mag
- Autopilot commander functionality
- Reference design available in open-source repository.





# Lycoming Engine Data Unit

- Complete EDU for Lycoming generating CANAerospace Data items.
- Designed to mount on firewall behind engine (85 °C/185 °f) ambient air).
- 4 EGT and CHT.
- Oil Temp, Oil Pressure.
- Fuel Pressure, Fuel Flow.
- Volts and Amps.
- 2 P-Mag serial channels for diagnostics.
- 2 tachometer channels.
- Alarms
- Open source repository includes design and source code.

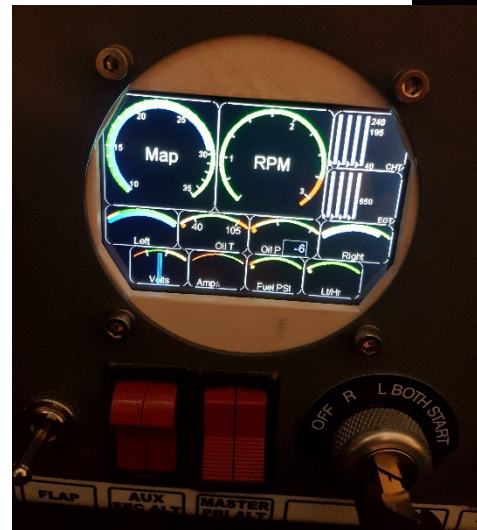
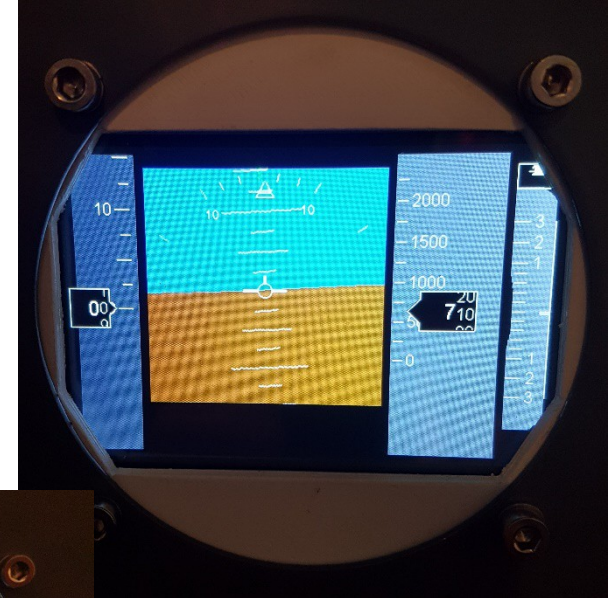


[www.kotuku.aero](http://www.kotuku.aero)

<https://github.com/kotuku-aero/diy-efis>

# Raspberry Pi CAN HAT

- Modified version of SLCAN with the footprint designed to fit on a Raspberry-Pi
- Lowest cost display system. Linux based.
- Version already flying.
- Has a 3D printed display enclosure for a 480x320 display.



[www.kotuku.aero](http://www.kotuku.aero)

<https://github.com/kotuku-aero/diy-efis>



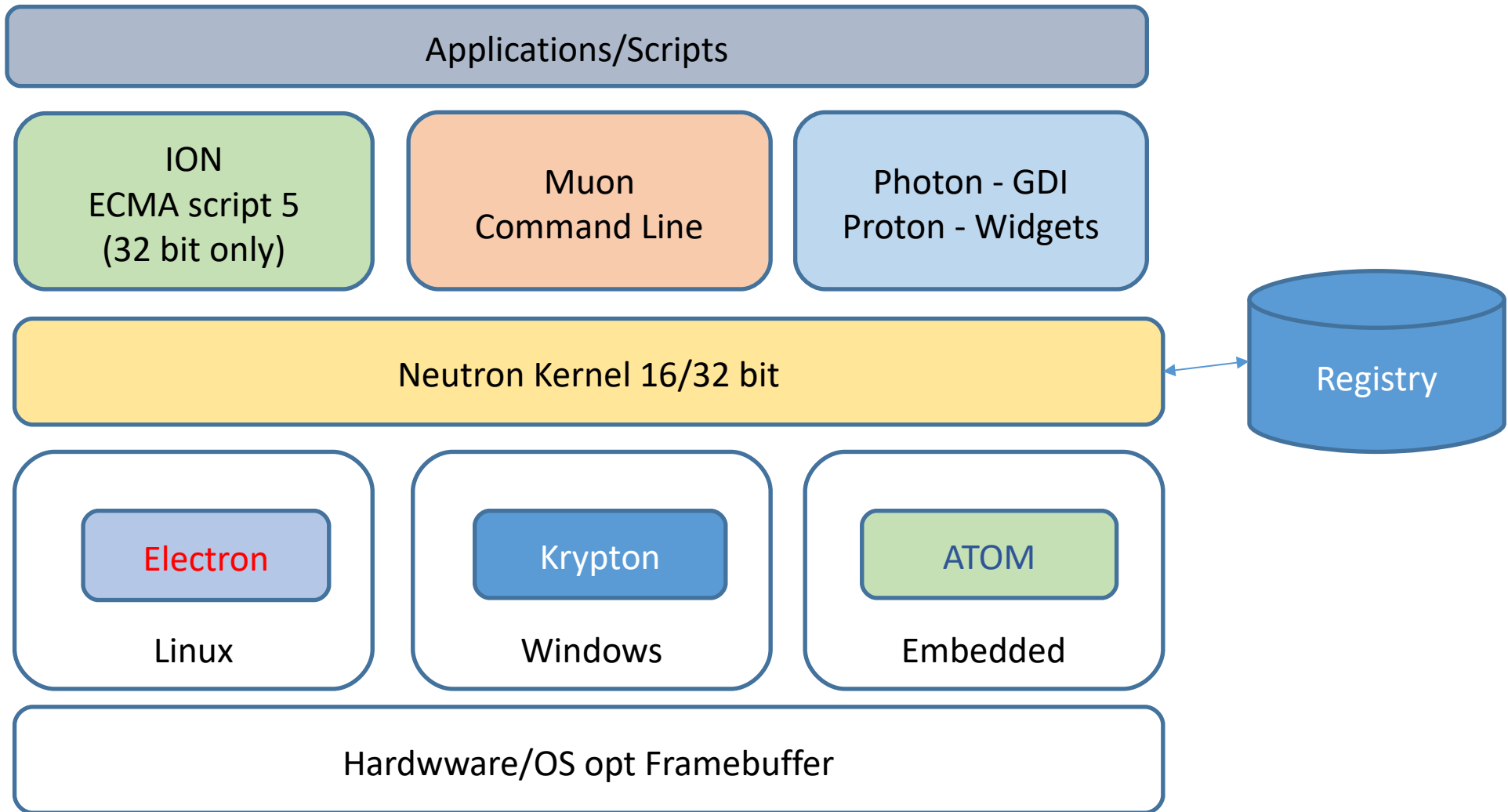
# 2019 version updates.

- 32 bit kernel released. Atom is ported to PIC32MZxxxxEFH for the 32bit peripherals (AHRS and EDU).
- Atom also ported to pic33EP for 16 bit devices used in the serial adapter and diy-efis interfaces.
- Design for CM3 version of Raspberry PI
- Javascript widgets supported for display.
- Menu system is running on displays.
- New CanBus -> Ethernet converter. Higher speed and less problematic than I2C or Serial interfaces.

# Architecture

- Written to be a 32bit platform in pure 'C'.
- Composed of libraries:
  - Neutron – Microkernel and Registry
  - Muon – Command shell
  - Ion – Javascript Engine
- Ported using platform libraries:
  - Electron – Linux using pthreads and framebuffer
  - Krypton – Windows, uses framebuffer emulator for testing
  - Atom – Embedded Microchip 16/32 bits
  - Tachyon – TCP/IP drivers

# Block Diagram



# Neutron Kernel

- Small multi-tasking kernel that is portable. Co-operative multitasking. 255 priority levels.
- Semaphores, and queues and Arrays are implemented as primitives.
- Registry allows for storage of 16/32 bit ints, Floats, strings, streams, XYZ, LLA, QTN and Matrix types. Designed for 128k-1mb EEproms.
- Ported to Linux, Windows, Pic16 and Pic32MZ
- Complete GDI with a Framebuffer.

# Hardware adaptation

- DIY-EFIS includes hardware drivers for 16bit PIC, 32/64 bit Windows and linux.
- All adaptations include framebuffer drivers.
- Framebuffer emulator is available for Windows and Linux under X11. Raspberry-PI uses /dev/fb0 in native mode.
- Linux / Windows systems don't have embedded features such as real-time kernel or hardware drivers.



# Muon CLI

- Command line with predictive input. CLI compiler is provided as part of code.
- Windows/Linux shell to allow for testing and development of scripts to be uploaded to devices.
- Built in Wordstar like VT-100 text editor.
- Has add-on modules for ION and Neutron on top of basic editor.
- Proton GDI widget is now available. Allows for building widgets using Javascript.

# ION Javascript

- Port of DUCKTAPE Open Source Javascript interpreter.
- Adds methods for calling CanFly support routines in Neutron as well as registry functions.
- Architecture is to provide scriplets to handle CanBus events published by the CanFly publisher in Neutron
- Muon includes an exec function() to test and develop.
- Debugger will come (not sure when!)

# Questions



July-2019

[www.kotuku.aero](http://www.kotuku.aero)  
<https://github.com/kotuku-aero/diy-efis>