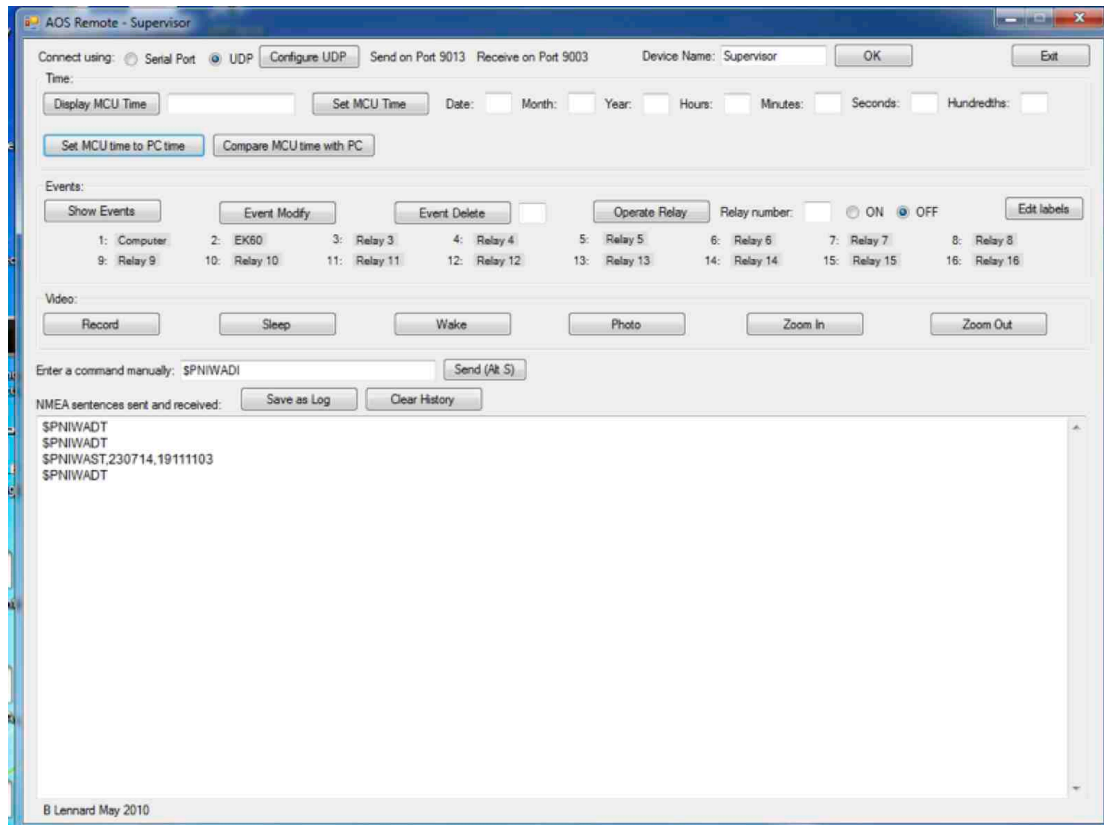


# A summary of some of my projects

## Applications:

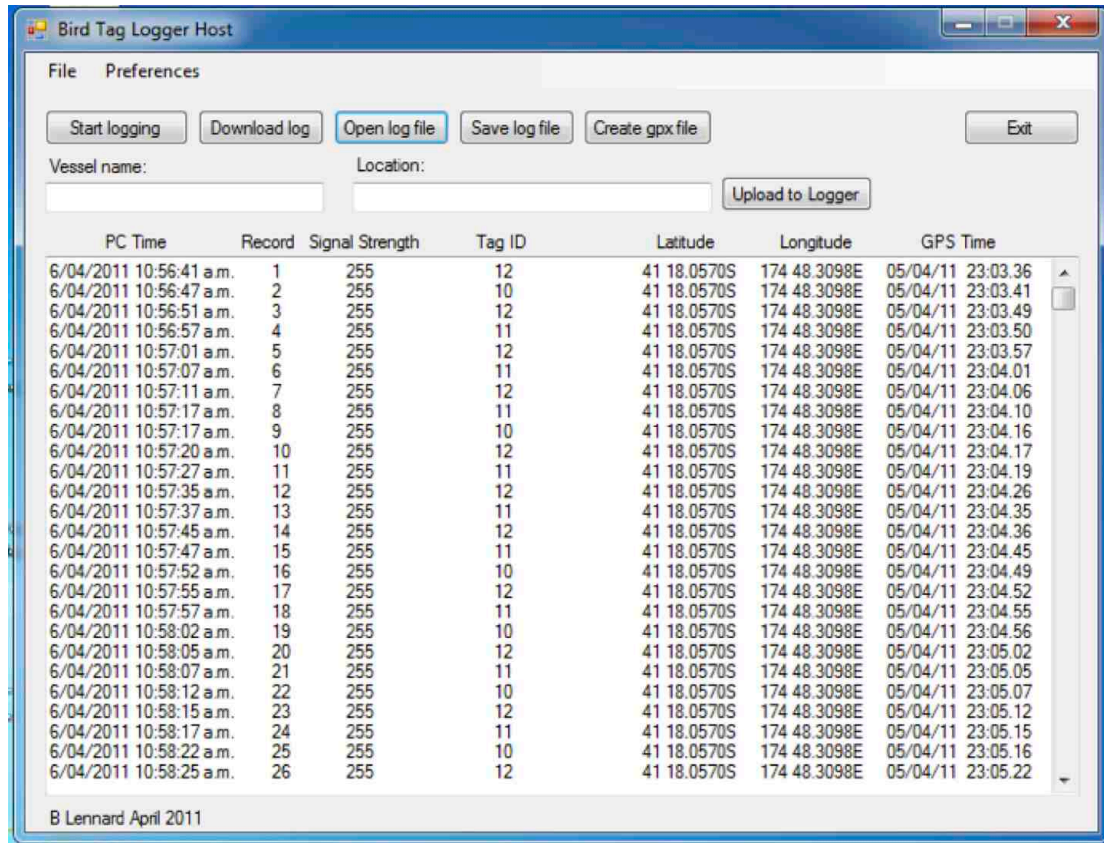
### AOS Remote

- A Windows application, written in C# (.NET).
- Purpose:
  - To set up Events, locally or remotely control circuits, and read telemetry for Niwa's "Acoustic Optical System".



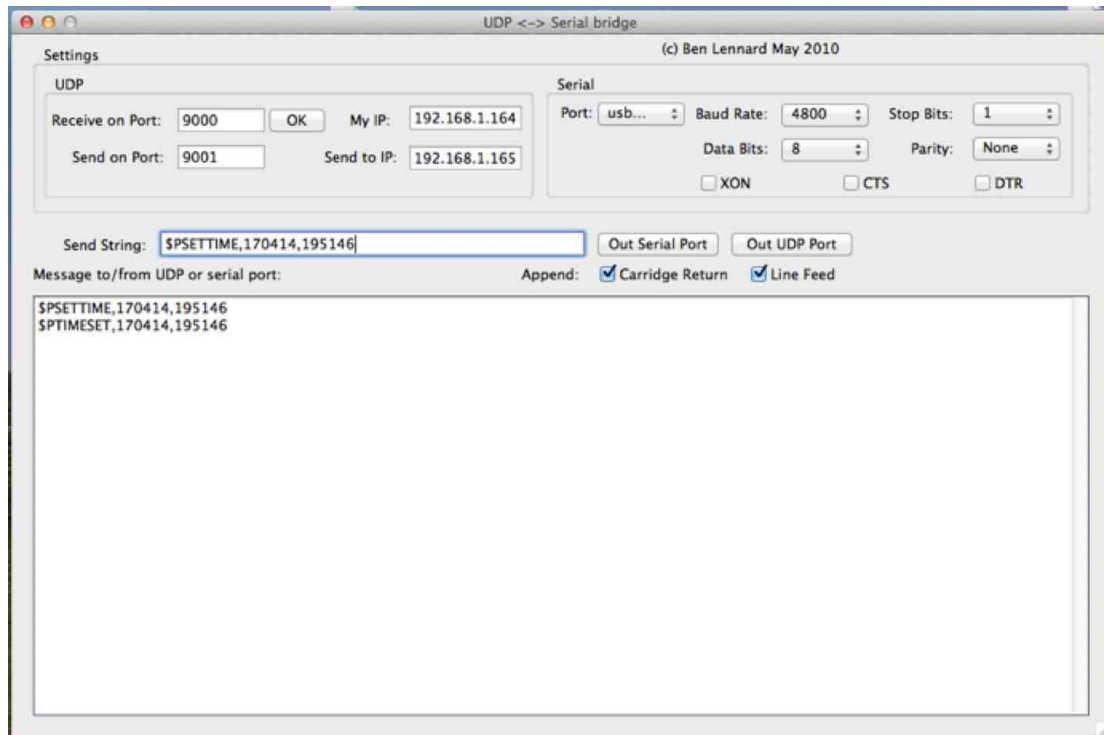
## Bird Tag Logger

- A Windows application, written in C# (.NET).
- Purpose:
  - To read logged data from an SD card and create gpx data so that a ship's track, along with logged tags, can be shown in Google Earth.



## UDP-Serial bridge

- A Macintosh and Windows application, written in Object Oriented Basic using Xojo.
- Purpose:
  - A software version of an Ethernet to Serial adapter. UDP data on an Ethernet port is sent out the serial port. Data on a serial port is sent out the Ethernet port as UDP packets.



## Wetlab Scale DAQ

- A Windows application, written in C# (.NET).
- Purpose:
  - Weigh catches of fish, with various scientific parameters, for stock assessment.

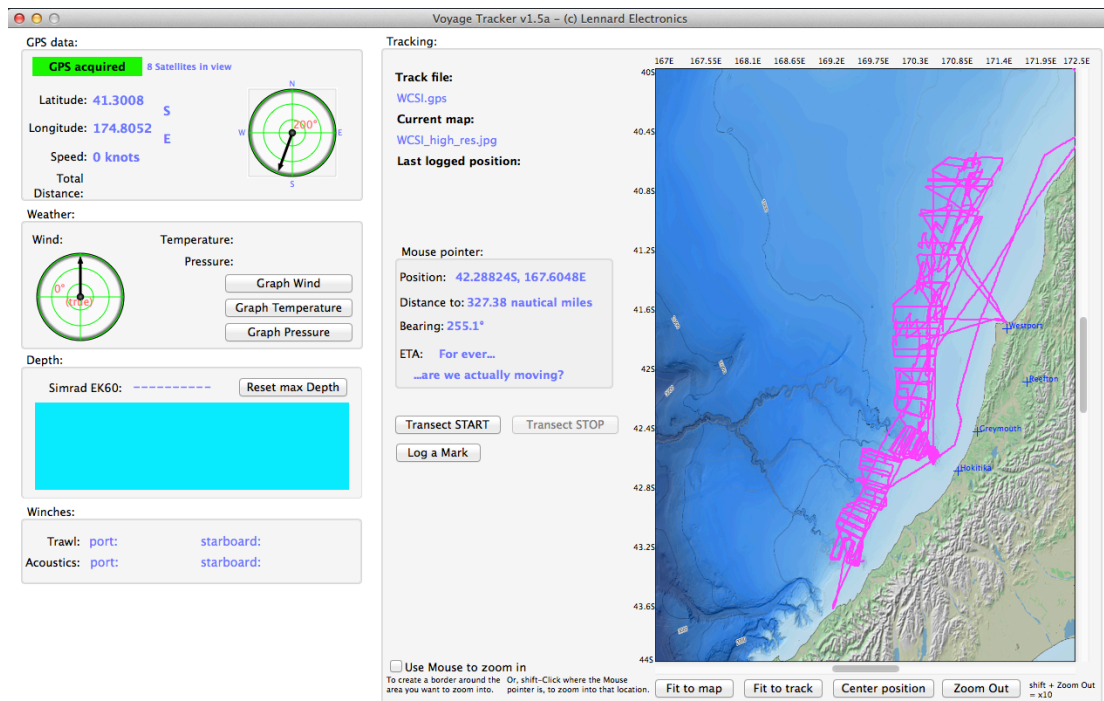
The screenshot shows the 'WetLab Scale Interface (Online)' application window. It features a menu bar with 'File', 'Scales', 'Catch Data', 'Serial Port', 'Help', and 'About'. The main interface is divided into several sections:

- WetLab Catch data:** Includes input fields for 'Trip code:', 'Work stn ID:', 'Station:', and 'User ID:'. To the right are buttons for 'Go Offline', 'Close WLCATCH and Exit WSI (Alt + X)', and 'Exit WSI (Alt + E)'.
- Online:** A green status indicator.
- History of records sent:** A table with columns: Record, Species, Sub Ch, LF, Biol, Weight, Oto, Kept, Catch Mthd, Number, Station, User ID, and Comments. The table is currently empty.
- Window 2 Current Record:** A form for entering data. Fields include 'Record:', 'Species:' (with 'GMU' entered), 'Sub ch:' (with '1' entered), 'LF:' (with '0' entered), 'Biol:' (with '0' entered), 'Weight kg:', 'Oto:' (with '0' entered), 'Kept:' (with '0' entered), 'Weigh method:' (with '1' entered), 'Number:', and 'Comments:'. A right-click option for 'Weigh method' is noted. An 'Enter Record (F3)' button is present.
- Fish Shortcuts:** A row of buttons labeled HOK, LIN, HAK, LDO, SPD, SWA, ORH, and BOE, corresponding to function keys F5 through F12. An 'Edit Keys (Alt+K)' button is also present.
- Scales:** Two sections for scale management. The left section, '100kg SeaWay Scale TARED.', has a 'Get Weight (F1)' button. The right section, 'Channel 2 Not Installed.', has buttons for 'Install Scale/s', 'Calibration Check', 'Zero Scale', and 'Tare', along with a 'Get Weight (F2)' button.

At the bottom, calibration dates are shown: 'Channel 1 last calibrated 3/01/2011 3:58:18 a.m.' and 'Channel 2 last calibrated: 17/11/2010 12:55:53 p.m.'.

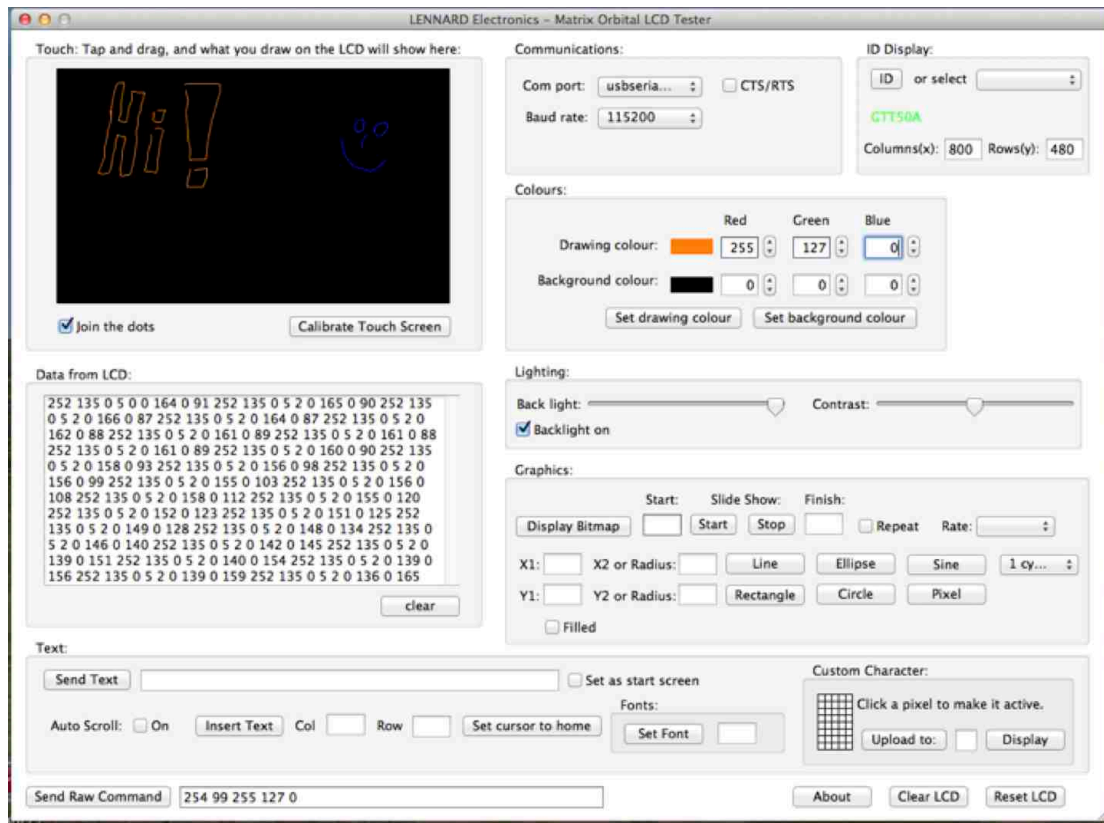
## Navigation software

- A Macintosh and Windows application, written in Object Oriented Basic using Xojo.
- Purpose:
  - Show, and log, current position, speed and direction.
  - Show, and log, current weather conditions, and other scientific data.



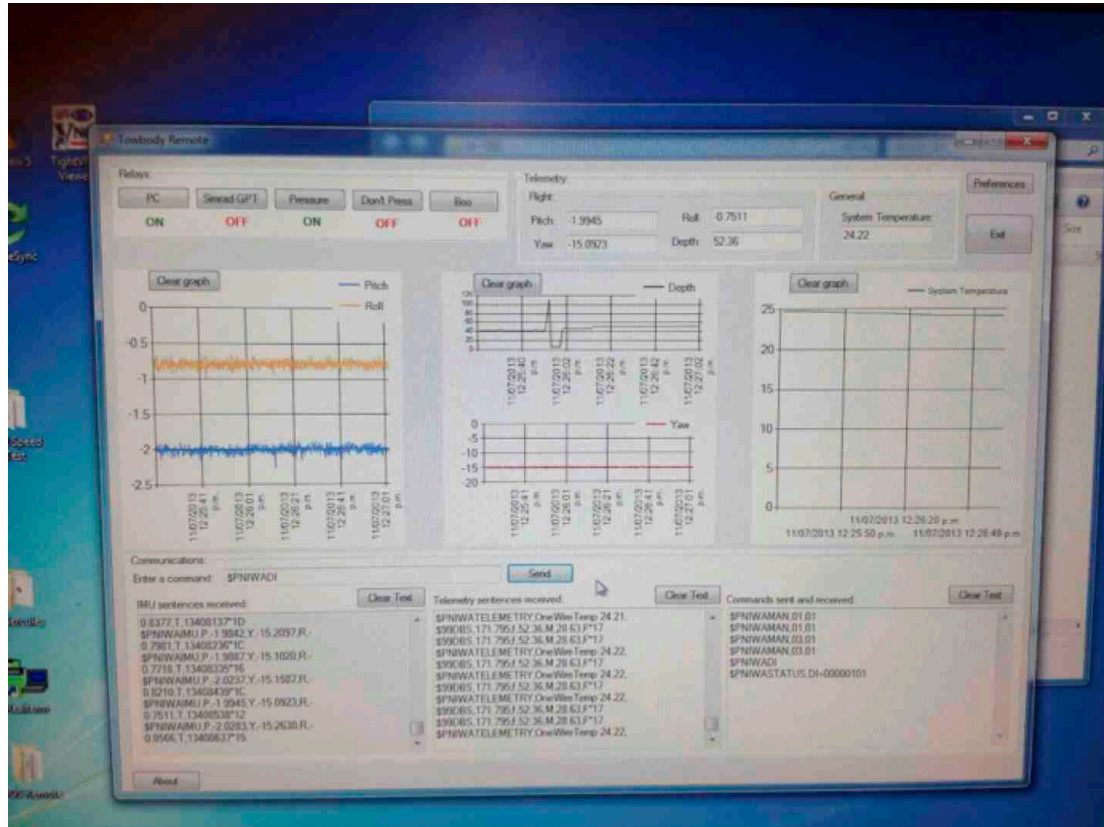
## Matrix Orbital LCD test software

- A Macintosh and Windows application, written in Object Oriented Basic using Xojo.
- Purpose:
  - Test the entire range of Matrix Orbital Displays.
  - Development tool for working with the Matrix Orbital range.



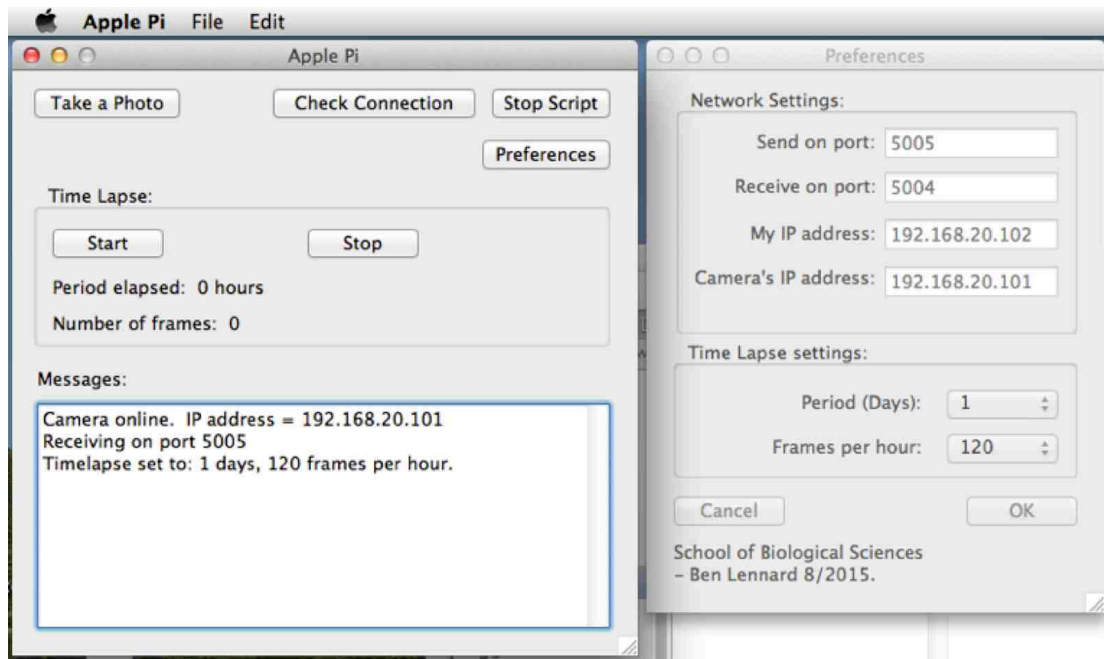
- A Windows application, written in C# (.NET).

- A Windows application, written in C# (.NET).
- Purpose:
  - To set up Events, locally or remotely control circuits, and read telemetry for Niwa's "Acoustic Towbody System".



## Time-lapse Photography

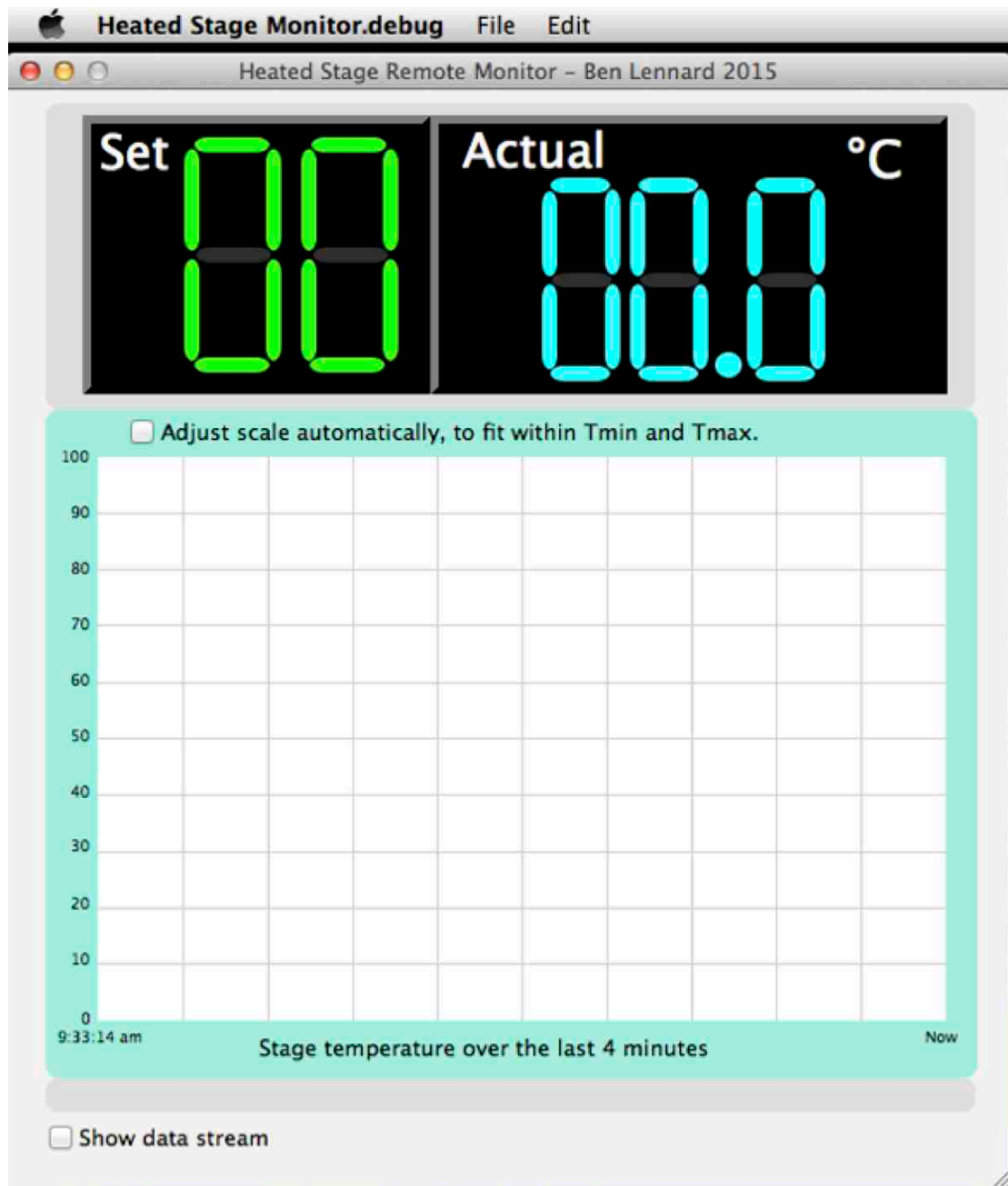
- A Macintosh and Windows application, written in Object Oriented Basic using Xojo.
- Purpose:
  - Remotely setup the camera's frames/hour, and period to run.
  - Start/Stop time-lapse.
  - Take a photo (separate to the time-lapse).
- The Time-lapse Camera was based on a Raspberry Pi (hence the Mac application is called "Apple Pi"), which automatically logged on and started a python script written by me.





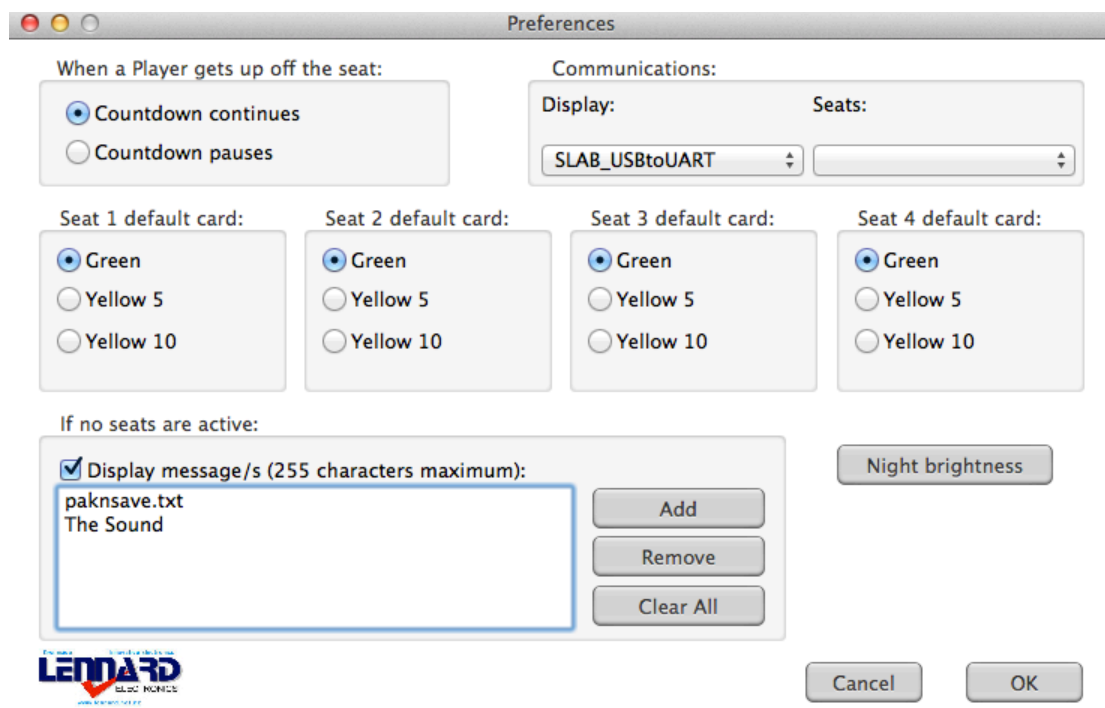
## Heated Microscope Stage Monitor

- A Macintosh and Windows application, written in Object Oriented Basic using Xojo.
- Purpose:
  - Remotely monitor the temperature of the heated stage
  - Provide a graph of the temperature over time
  - Provide data on individual temperature sensors, fan status, over temperature cutout relay, to aid in fault finding.



## Penalty Box Management system (aka “Sinbin system”)

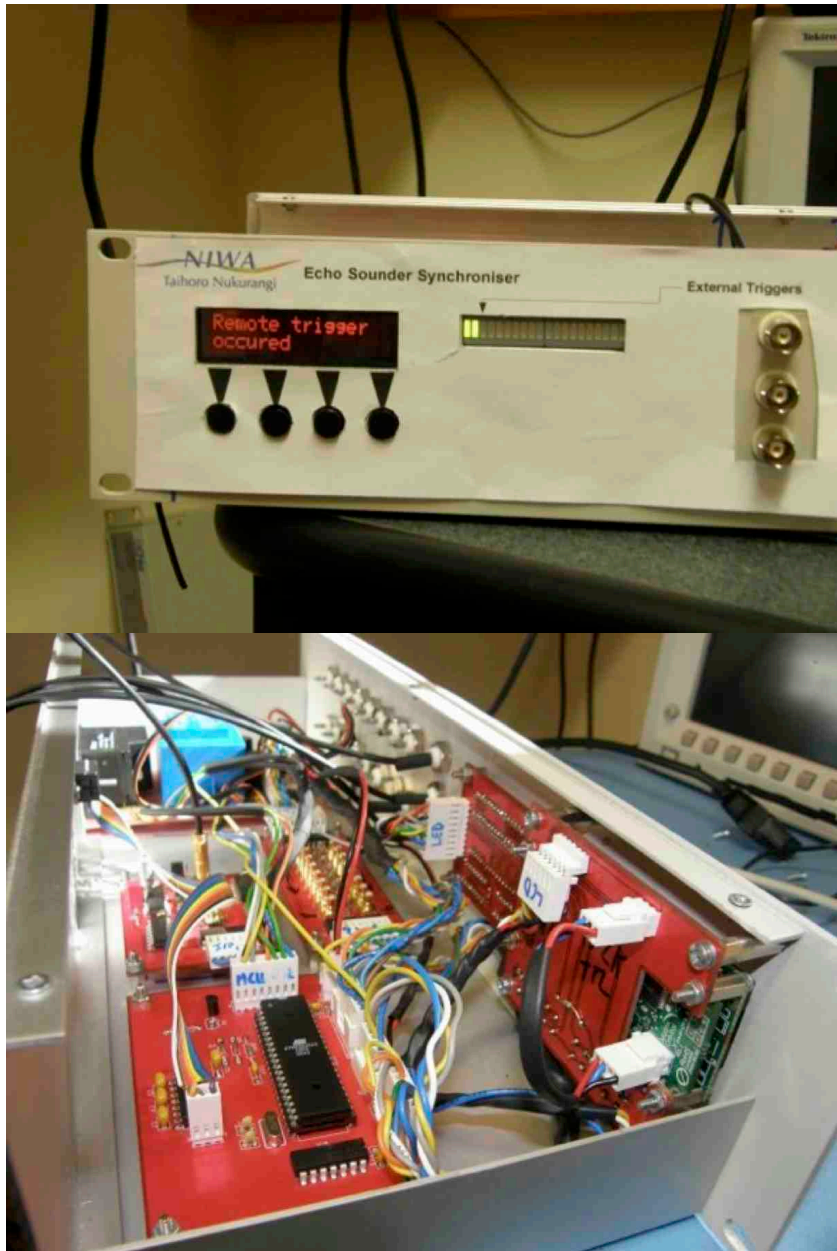
- A patented Macintosh, Windows, and Linux application, written in Object Oriented Basic using Xojo.
- Purpose:
  - Automatically manage players sent off during a game of Hockey
  - Allows Umpires and technical bench officials to concentrate on the game
  - Fully override-able by the technical bench
  - Displays advertising on an external display when no one is in the Sinbin



## Hardware

### **Echo Sounder Synchroniser**

- A system for synchronizing up to 16 different echo sounders on Niwa's research vessel RV Tangaroa. It can be setup via a keypad on the front panel, or remotely via a web based program.
- Main processor is an ATmega programmed in C.



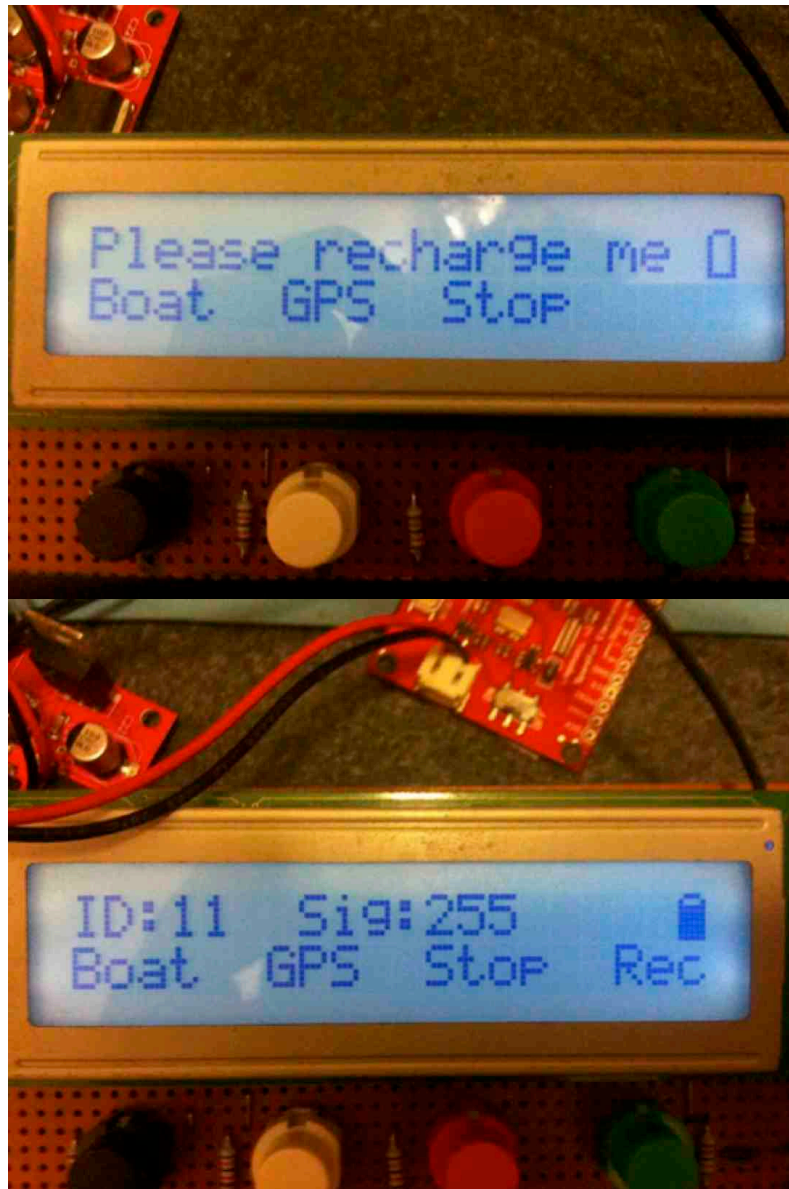
### Keyboard Emulator

- 8051 based, programmed in C.
- Purpose:
  - Takes x/y data from a digitizer board that uses an inductive stylus, and converts those coordinates into keyboard scan codes and sends the data out to a PS2 port (USB via a PS2-USB adapter).




### Bird Tag Logger

- 8051 based, programmed in C.
- Purpose:
  - Log bird tags, and a ship's GPS position, to study the impact of the Petral population due to the Snapper fishing industry.



# Bird Tag Logger

ID:      Sig:        
Boat GPS Start Log



← USB

**NIWA**

Taihoru Nukurangi

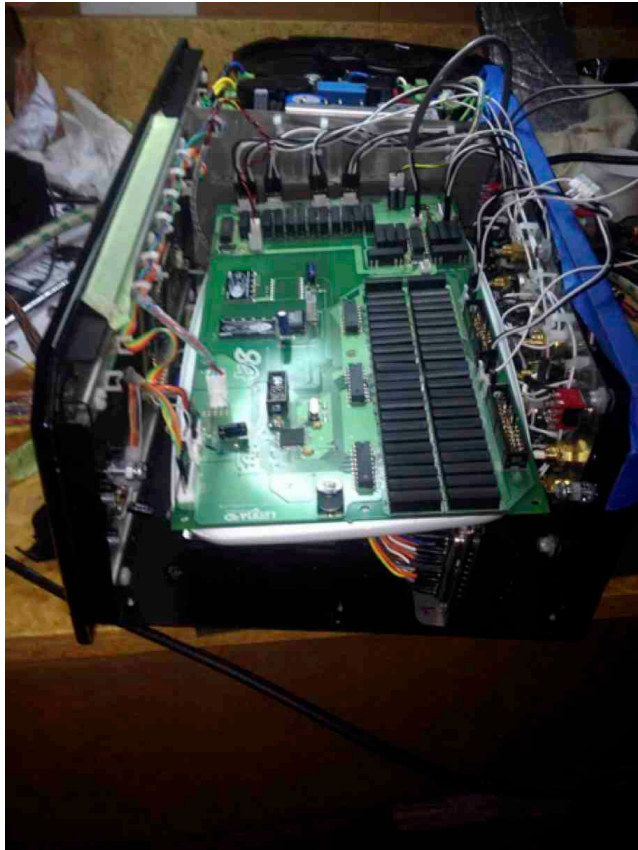
National Institute of Water & Atmospheric Research

Designed and built by the MDA Group



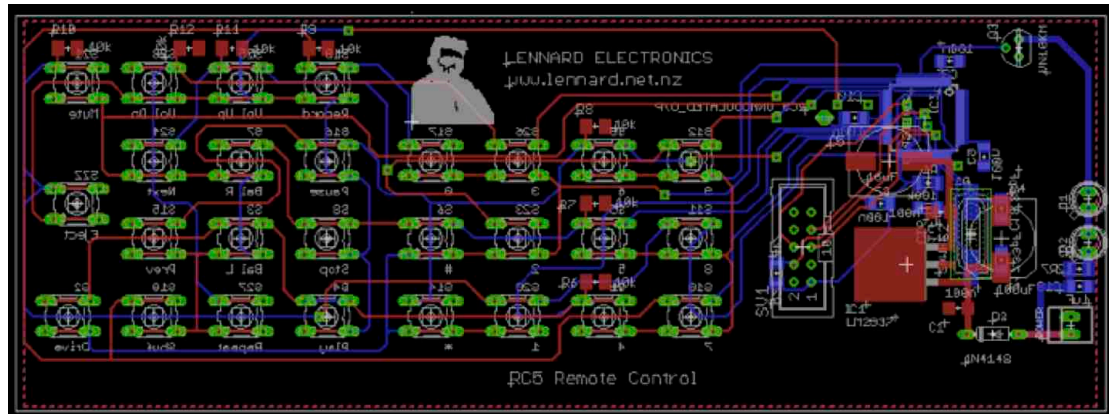
### **Amplifier Controller**

- 8051 based, programmed in C.
- Old school electronics, with a modern twist. The amplifier is a valve system, but all its functions are controlled via a touchscreen LCD, and remote control. Future versions will include iPhone control and USB audio support.

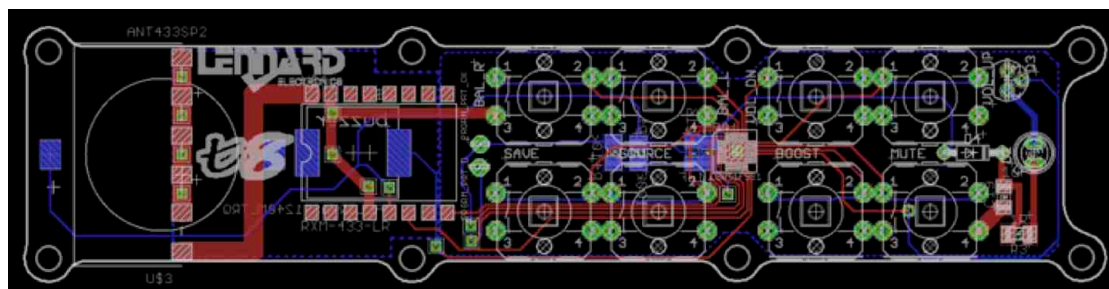


### RC5 compatible Remote Control

- 8051 based, programmed in C.
- Designed to use with a multi-deck CD player kit that did not have a remote control supplied.



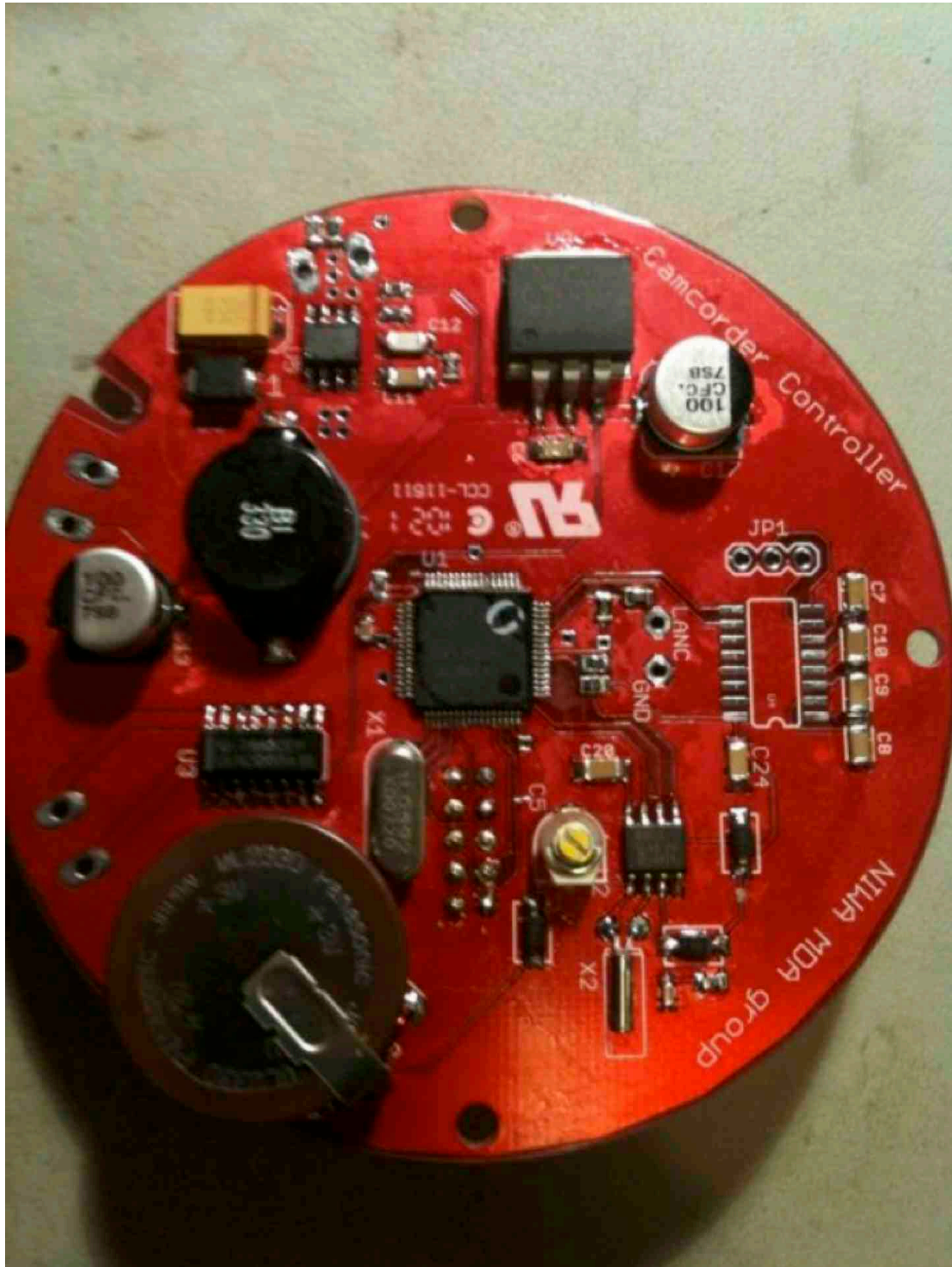
- A more compact and battery friendly version was designed for the Amplifier controller shown above.





### Camcorder controller

- 8051 based, code written by a colleague, hardware designed by me.
- Designed to work with the AOS Remote software shown above.
- Controls a Sony CamCorder or GoPro, in an underwater housing.



## System Supervisor

- 8051 based, code written by a colleague, hardware designed by me.
- Designed to work with the AOS Remote software shown above.
- Controls an Atom PC, Simrad EchoSounder, lights, and cameras, in an underwater housing.



### Heated Microscope Stage

- PID control of a stage to keep samples at the right temperature while being examined under a microscope.
- Arduino based microcontroller, and very precise temperature control up to 75 °C.
- Sends data out the USB port for external monitoring or fault finding.



### Penalty Box Management system (aka “Sinbin system”)

- Touch Screen Controller, based on a Raspberry Pi
- Four seats with sensors built into them, plug into the controller



- Controller displays sinbin information or advertising on an external display, via Bluetooth



