CAWTHRON INSTITUTE

30 July 2014

Paul Sheldon
NCC Acting Envirolink Co-ordintor
Tasman District Council
189 Queen Street, Private Bag 4
Richmond, Nelson, 7050
NEW ZEALAND

Dear Paul

ENVIROLINK NLCC75 DEVELOPMENT OF A LOW COST OPEN SOURCE BUOY CONTROLLER

As discussed, please accept this letter as a summary and report for the work carried out under Nelson City Council's (NCC's) Envirolink request (NLCC75). The purpose of the Envirolink was to develop an inexpensive prototype open source controller for water quality monitoring buoys or similar remote sensing applications.

Recently, Cawthron has invested a great deal of time and effort into developing water quality buoys that can be used as 'turn-key' platforms by Regional Councils for helping to fulfil, among other things, their state of the environment monitoring objectives for coastal systems. However, councils have consistently indicated cost as a barrier to expanding their use of monitoring buoys and the purpose of this project has been to investigate the efficacy of developing a low-cost controller for these systems using an off-the-shelf open source single board computer platform (the Beaglebone©).

This letter summarises the outputs from the design and fabrication of the prototype controller developed and offers some comments and recommendations on the lessons learned and future direction for these types of controllers.

Prototype controller at a glance:

The following table outlines the specs of the proto-controller but in summary, the controller consists of an embedded linux computer (the Beaglebone©) with a custom designed and fabricated printed circuit board (PCB) that the computer plugs into and gives access to various computer components and peripherals necessary for controlling a buoy. These include RS232 serial ports, analogue inputs, switches for powering different instruments and a real time clock.

Item	Title	Description
1	Main Controller	Beaglebone© Black with Debian linux operating system
2	Serial Ports	Four (4x) on-board serial ports each with a RS232 level shifter
3	Power Ports	Four (4x) diode protected power ports each controlled via a solid state relay. Port power can be either 5 VDC or 12 VDC depending on instrument
4	Analogue inputs	Four 12 bit analogue to digital inputs
5	Real Time Clock	Real time clock with coin cell backup and alarm output to wake main controller
6	DC-DC module	DC-DC step down to accept 12 VDC input but power computer at 5 VDC.
7	Physical size	54.6 mm wide x 203.2 mm long. Includes header to plug Beaglebone into PCB

An Adobe Acrobat PDF of the prototype controller schematics, PCB and bill of materials are attached to this letter. Many of the components used (*e.g.* solid state relays, RS232 level shifter *etc.*) in the design of the controller are ones that are presently used in Cawthron's buoy programme so the efficacy and reliability have already been tested. At this stage it is envisioned that Cawthron will keep possession of the three prototype controllers built under this project and continue the development process, but we are happy to share the designs and hardware with other end users.

Lessons Learned and Where to From Here:

These single board computers like the Beaglebone© offer excellent computing power, a small footprint, and are very cost-effective but they are still largely aimed at a desktop environment where power consumption is not an issue. Buoys and other embedded systems need to be very frugal with power use and the biggest hurdle to date has been addressing this issue. Our solution has been to incorporate a real time clock on the PCB that can be used to re-power the unit after a set period of time and to use on-board software to put the main controller to sleep (*i.e.* power down). Hopefully future versions of the Beaglebone will have better power management and this can all be done via software control.

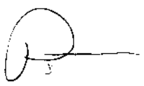
Other minor hurdles have been getting access to all the serial ports, ensuring the GPIO pins being used aren't required by the main processor for other tasks and making decisions on what to include/exclude in order to maximise functionality but keep the general design as simple as possible

Finally, although not a specific output for this project, the choice of software and data storage has also been investigated. At this stage, we have selected Python© scripts and SQL-lite for testing the functionality of the hardware. This will involve the main steps required for a buoy controller. That is, powering a serial instrument (*e.g.* a GPS enabled met station), parsing the NMEA input, and writing the data to SQL tables.

Future development will continue and it's likely that second generation boards might have minor modifications, but the output from this project has demonstrated that this type of controller is worth pursuing and will hopefully become part of the standard toolbox for future buoys.

It is my understanding that the Coastal SIG may be interested in getting some information on the outputs from this Envirolink. I think this is an excellent avenue for disseminating the lessons learned on the project and we would me more than happy to send along some hardware or even attend a meeting to discuss the outputs. Feel free to contact me should you have any questions or comments on this letter,

Yours sincerely



Paul Barter
Senior Marine Scientist - Coastal and Freshwater Group