**1930 SW Fourth Ave, Portland, Oregon 97201 Rev. 1 1/14/15**

Team Members

Chelsea Throop

Saida Akhter

Phu Nguyen

Trevor Conant

**Winter**

15

Capstone Project Proposal

08

**Fall**

# 

# Project proposal

Concept: The NEAR-lab sea glider is currently implementing a prototype acoustic device that uses two hydrophones to gather acoustic data for oceanographic research. This current recording device, however, lacks some desired features. Our goal is to design additional hardware and a software interface to improve the usability of the current device.

Improved Feature Goals:

* Must be able to interface with TASCAM buttons
* Ability to access multiple SD cards, must be capable of up to 4 days of audio recording
* Must be capable of reasonable power consumption to provide at least four days of continuous recording 30 days sleep mode using alkaline batteries
* Must develop battery mounting harness
* Must develop hardware brackets to mount completed recording device
  + This will depend on available space measured in the glider
* Completed device must weigh less than 0.5 kg
* Must allow for complex, user selectable recording schedule
* Must be able to modify sample rate
* Time stamps must be able to be synchronized between recording device and laptop
* Data stored on SD cards must be downloadable to laptop
* Must be able to hear or display (spectrogram) recorded data

Hardware:

* Microcontroller will be used for button interface, must include enough GPIO pins for buttons, power, etc
* MUX will be used to select SD card for multiple SD use
* Power budget must be estimated to determine amount of battery supply needed for 4 days of continuous recording/ 30 days of sleep
* Hardware should implement sleep mode to conserve power when not recording
* Memory must be non-volatile, no data loss while device is powered down
* RTC chip for accurate time stamp labeling of recording, used as interrupt to wake recorder for scheduled recordings
* Completed hardware must be designed for testability

Software:

* User interface command line, C#, C?
* User interface should be simple, though allow for complex scheduling, may use I2C for RTC
* Interface for programming schedule must be determined (USB, Ethernet, etc)
* Development of spectrogram may be included in software
* Software must be thoroughly tested
* Documentation and user guide must be completed with software