Buoy Documentation

for

Salish Sea Water Weather Station

Created by R. Lingg, S. VanZwol, M. Grappo, and K. Stewart  
*Advised by Dr. Erik Fretheim*

*Western Washington University*

Version 1.0

2019-10-30

# Table of Contents

[**Table of Contents**](#_mslvh2y1d8hf) **1**

[Revision History](#_6j27nldm1k7i) 2

[**1 Hardware**](#_smfx75kue2mz) **2**

[1.1 Parts](#_wakc85cac3wu) 2

[Physical Components](#_ci5pfqj8nmyk) 2

[Sensors](#_xd0tszafoneh) 3

[Temperature](#_uo67t7nd6l5) 3

[Turbidity](#_riwlad4djx3p) 3

[Salinity](#_g7fhrd9kt0u3) 3

[Light](#_gs48ornsxsnc) 3

[Possible Additions](#_mzvtojoj1eg) 3

[1.2 Cost Breakdown](#_ktze1eo5ngh7) 3

[1.3 Wiring](#_p2m2gzmd4i5y) 3

[2 Development Environment](#_h4aaaexpfndp) **4**

[2.1](#_hnu64ww17kiu) 4

[**3 Source Code**](#_3f86vtjkpbt) **4**

[3.1](#_yd1x7frw0wi2) 4

[**4 Communication Protocol**](#_9uk12so7q65p) **4**

[4.1 Procedure](#_fgbfpuy9tup5) 4

[Retrieve Collected Data](#_4tcjtemf0716) 5

[Start Data Collection](#_ktof2yipwxoa) 6

## 

## 

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Authors | Date | Description | Version |
| KS, SV | 2019-10-30 | Document Created | 1.0 |
|  |  |  |  |
|  |  |  |  |

## 

# 1 Hardware

## 1.1 Parts

### Physical Components

Updated from the [parts list](https://docs.google.com/spreadsheets/d/1D3wp2_h2FYYPjg3HTSsOnKXNpVOY5yYXva5ze-UkAWs/edit#gid=0) provided by the previous project group.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type | Name | Function | Approx. Cost ($/unit) | Comments |
| Container | Nalgene bottle | Contains and protects electrical components | ~$10 | <capacity> |
| Microcontroller | [SparkFun ESP32](https://www.sparkfun.com/products/13907) | Main controller of the data station | $21.95 | Wifi- and BLE-compatible |
| Sensor | [Turbidity](https://www.mouser.com/ProductDetail/Amphenol-Advanced-Sensors/TSD-10?qs=%2Fha2pyFaduhMb4Myu85WEoa%2FWkLiDTVQ8z7k2VF6ZtfHbZGZqD6ltA%3D%3D) | Detects water transparency (quality) |  | Currently not available (10/30/19) |
| Board Part | [Capacitor](https://www.mouser.com/ProductDetail/KEMET/C320C104M5U5TA?qs=sGAEpiMZZMt3KoXD5rJ2N%252BwgBl1a522xiJqI7eJgzUw%3D) |  | ~$0.15 |  |
| Sensor | [Photoresistor](https://www.amazon.com/goeasybuy-Sensitive-Resistor-Photoresistor-Optoresistor/dp/B01CGCNO34) | Detects light levels | ~$0.25 |  |
| Board Part | [Accessory resistor](https://www.mouser.com/ProductDetail/Vishay-Dale/CCF075K10GKE36?qs=sGAEpiMZZMtlubZbdhIBICS9vICa6AcHvyYUz9mXcs4%3D) |  | ~$0.04 |  |
| Sensor | [Salinity](https://www.ebay.com/itm/10pcs-Rain-Water-Liquid-Level-Sensor-Module-Depth-of-Detection-for-Arduino-UNO/152046028517?_trkparms=aid%3D555018%26algo%3DPL.SIM%26ao%3D1%26asc%3D20131003132420%26meid%3D7e4eed2caebc4ac6a3e35eb33995ba4f%26pid%3D100005%26rk%3D1%26rkt%3D11%26sd%3D232051955183%26itm%3D152046028517&_trksid=p2047675.c100005.m1851) | Detects water salinity | ~$0.30 |  |
| Sensor | [Temperature](https://www.mouser.com/ProductDetail/Analog-Devices/TMP36GT9Z?qs=sGAEpiMZZMv9Q1JI0Mo%2FtZYNPIqRJ81F) | Detects ambient temperature | ~$1.20 |  |

### Sensors

#### Temperature

#### Turbidity

#### Salinity

#### Light

#### Possible Additions

## 1.2 Cost Breakdown

Per the project’s validation criteria, the overall cost of the buoy’s components must not exceed the defined project budget. To that end,

## 1.3 Wiring

Sample

<https://docs.google.com/spreadsheets/d/19lzxTWsdkvWb_E3hUEWOfWlRzDsdCcBoqYzUNvuqZds/edit#gid=0>

<https://docs.google.com/spreadsheets/d/16PBMx9uLK1qTHvzmdbG0xfXh4A7sZzKRRO699uiOjMA/edit#gid=0>

<https://docs.google.com/document/d/1-gs1rVHc8KxXtjsdJf3wkLoyhVF1pIcRGSdBsArTynY/edit#heading=h.dqx40nea668p>

The H2O sensor repo has a file called Board\_Schematic.fzz that could contain a wiring schematic. It requires the program Fritzing to open which appears to be behind a paywall? Not sure.

# 2 Development Environment

## 2.1

The board software is compiled using the ESP-IDF library version 3.3.

<https://github.com/espressif/esp-idf>

# 3 Source Code

## 3.1 Initialization

The board initializes in ssds\_main.c, in the function app\_main(). This function calls the initialization functions of the storage, bluetooth, gpio, and timer code. As part of the bluetooth init, it registers some of ssds\_main.c’s functions as callbacks. Once it finishes all the initialization, this thread is killed.

Currently, app\_main will re-initialize if the cause for wakeup is undefined; we will likely need to review this behavior in future.

## 3.2 Timer

When initializing, ssds\_timer.c creates a data collection worker, and registers it to a timer provided by the rtos library. Each time the timer expires, the data collection worker will proceed to gather a data point via ssds\_gpio.c. It will then suspend again, waiting for the next timer interval.

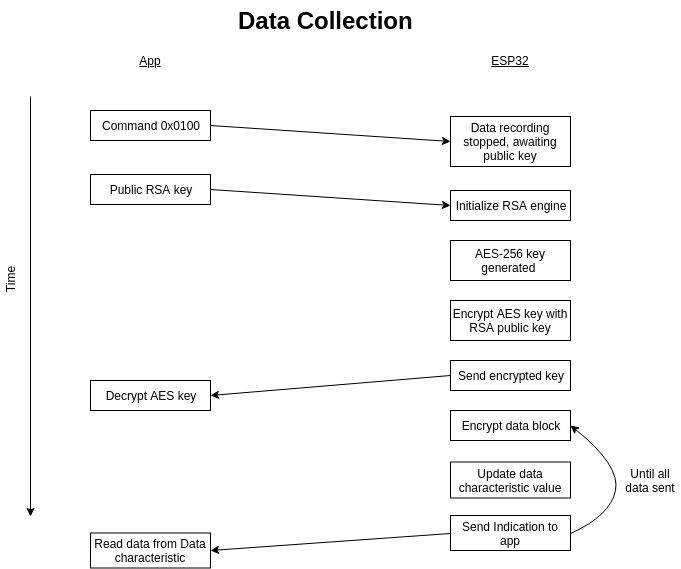
# 4 Communication Protocol

Adapted from the [communication protocol document](https://docs.google.com/document/d/14OYbrfvJ1-p8RKao8xCEPFe26fBFQeSXh2ftQ0SG2Ak/edit) created by the previous project group.

## 4.1 Procedure

The various functions of the buoy are initiated primarily by commands transmitted over BLE by a connected device using the mobile app. Under normal circumstances, the buoy collects and stores data at regular intervals autonomously. Upon receiving a command, the buoy triggers the associated event.

### Retrieve Collected Data



### Start Data Collection

