

Design Doc of Water Weather Station

Senior Project 2

By: Bryan Pham and Caleb Ponce

Table of Contents

Context and Scope	3
Background.....	3
Business Opportunity and Objectives	3
Market and Customer Needs	3
Vision for this Project	3
Major Features	3
Scope	4
Scope after the Initial Scope	4
Limitation	4
Business Context.....	4
Stakeholder	4
SEA Discovery Center	4
SEA Discovery Center Students	5
General Public.....	5
Project Priorities.....	5
Buoy Parts	5
Components	5
Sensor Information.....	5
Buoy Build	8
Board Assembly.....	8
Buoy Development Environment.....	11
ESP-IDF	11
Diagram of Connection	12
Data Collection Diagram	12

Context and Scope

Background

The Weather Water Stations project is an ongoing project with the SEA Discovery Center located in Poulsbo, WA. The project goal is to educate youth campers on cyber-security, computer science, and electrical engineering. Students take a camp at the SEA Discovery Center and can take home a Water Weather Station in a small form factor to take readings of multiple water conditions.

Business Opportunity and Objectives

The project offers an educational tool for youth campers that will attend the camp at the SEA Discovery Center. This will give youth an opportunity to learn more about how a system involving hardware, networking, and security to capture sensor data to create meaningful information about the current water conditions. This information is valuable, as it can be used to analyze changes in water conditions in many areas across the region. The final system where the campers can build their weather station, bring it home, make a secure connection to it, and upload that data to an online database. Campers will use the knowledge they learned at camp about secure connections to connect to the device. Then using the water weather and GPS location create useful data that can be used to track weather changes over the greater Pacific Northwest area.

Market and Customer Needs

We will create a system for students to practice their cyber-security knowledge and create useful data for The SEA Discovery Center. The Water Weather Stations is a system of hardware and software that allows for the secure collection of water weather data by students from the SEA Discovery Center. Unlike an online or local hardware connection our product is a combination of both that gives students the ability to make physical secure connections and globally useful data.

Vision for this Project

We will create a system for students to practice their cyber-security knowledge and create useful data for The SEA Discovery Center. The Water Weather Stations is a system of hardware and software that allows for the secure collection of water weather data by students from the SEA Discovery Center. Unlike an online or local hardware connection our product is a combination of both that gives students the ability to make physical secure connections and globally useful data.

Major Features

The project will consist of three major code bases: the water weather station hardware code, the Android application used to create a Bluetooth connection to the weather station,

and the web interface for displaying collected data in a relevant way. Alongside the web interface there will be a database that can store and organize the collected data.

Scope

This is a continuation of the on-going project with work from a previous team. For the station, we will finish implementing the current sensors to the station. The station firmware will be updated to log the data, and to send it to the Android app. The Android app will need to be within 30 ft of the buoy to read and send the data logs to the database with an encrypted connection. For the web app, we will continue to finish and complete the web app to get it up and running live, with a simple, visual representation of the data logs from the many water weather stations. For the Android app, we will implement communication⁴ between the android app and the database which will be storing all the station data logs and making sure the data is transmitted reliably and securely using an encrypted connection.

Scope after the Initial Scope

As development progresses, we would like to add a visual guide on the Android app for the campers when they first receive the station and install the Android app. The guide will guide the user on how to assemble the station together and how to connect and retrieve data from the station. For the web app, we want to add filters and sorting options for the visual representation of the station data. For the station, we would like to improve the power usage of the station, through more battery capacity, more efficient and optimized code, and testing various times between sleep and wake of the station.

Limitation

We have a couple limiting factors, the funding for the stations. As these stations will be used at the SEA Discovery Center's Cyber-Security Summer Camp, the material costs for creating the devices will be funded by sponsors of the camp. We are currently limited to \$50 per station with the current housing and internal hardware components. We will not be able to modify the housing design of the current station since that will change the layout and circuitry of the station and there is not enough available time to fulfill such a redesign of the hardware.

Business Context

Stakeholder

SEA Discovery Center

SEA Discovery Center will benefit by being able to engage students after they have left their campus. This will allow for greater retention and aid in their goal of youth education on cyber-security.

SEA Discovery Center Students

SEA Discovery Center students will be able to create meaningful data using what they learned at the cyber-security camp. This allows for a deeper understanding of why cybersecurity is important and has a meaningful impact on the student's education.

General Public

The General public will be able to access the water weather data collected by students and be able to get copies of this information for free through the web application.

Project Priorities

The most key features will be the hardware, hardware code and Android app. These three things must work together to allow students to operate them freely. Then the ability for the Android app to connect to the database and store the collected information remotely. Finally, the web-app must make the stored data relevant, organized, and available to the public.

Buoy Parts

Components

Components - Model	Quantity
Breadboard	1
Microcontroller – SparkFun ESP32 Thing	1
Photodiode – SFH 2505-Z	1
Temperature Sensor – TMP36GT9Z	3
Turbidity Sensor – TSD-10	1
Potentiometer – 100 Kohm	1
M-to-F Jumper Wires	3
Prototyping Wire	As much as you need

Sensor Information

Temperature:



Sensor Model: TMP36GT9Z
Manufacturer: Analog Devices INC.
Input Channel: TEMP1_CHANNEL, TEMP2_CHANNEL, TEMP3_CHANNEL
Attenuation Value: ADC_ATTEN_DB_6
Unit Conversion Formula: Degrees Celsius = (Voltage(mV) - 500) / 10
Implementation State: Implemented and Tested

Light:



Sensor model: GM5539 Photoresistor
Manufacturer: Wodeyijia6
Input channel: LIGHT_CHANNEL
Attenuation value: ADC_ATTEN_DB_11
Unit conversion formula: $Watts/M^2 = 0.005167 * Voltage (mV) + 0.7315$
Implementation state: This sensor is **not** calibrated against sunlight, so this conversion formula only works for LED light. It represents an example of the final equation and is not implemented into the board. For the sensor to work with sunlight, the reference resistance must be increased, and the sensor must be calibrated against sunlight.

Salinity (Water):



Sensor model: Water Level Sensor
Manufacturer: ARCELI
Input channel: SALINITY_CHANNEL
Attenuation value: ADC_ATTEN_DB_6
Unit conversion formulas: none
Implementation state: Unimplemented.

Turbidity:



Sensor model: TSD-10
Manufacturer: Amphenol Advanced Sensors
Input channel: TURBD_CHANNEL

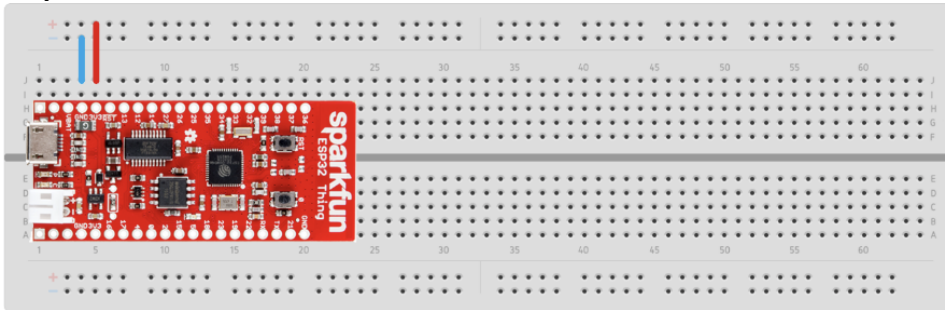
Attenuation value: ADC_ATTEN_DB_6

Unit conversion formula: Turbidity (NTU) = $-1120.4 (\text{Voltage (mV)}^2) + 5742.3(\text{Voltage(mv)}) - 4352.9$

Buoy Build

Board Assembly

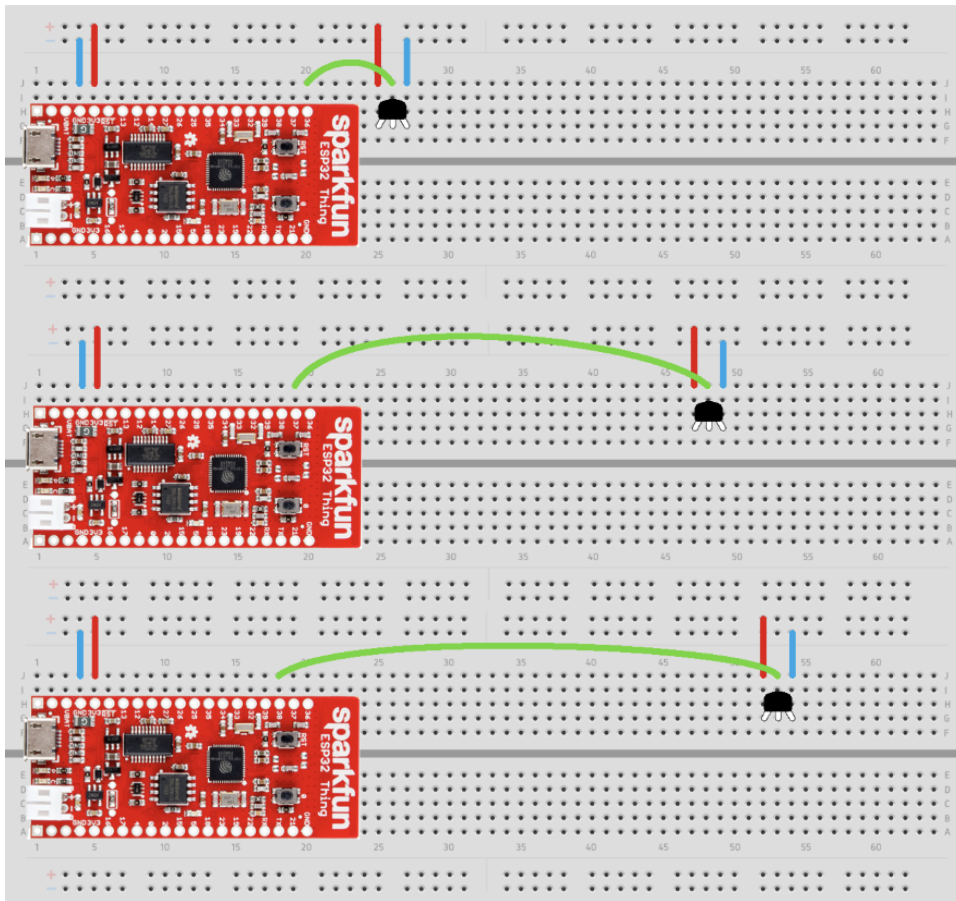
Step 1 – Power:



First, make sure that the ESP32 board is properly seated. The bottom left pin of the board should be connected to the 1st column in row A. The board should take up columns 1 through 20, and should cover rows A through H.

Next, connect the wire from column 4 to the negative rail and column 5 to the positive rail. This will make sure that ESP32 board is getting power to it.

Step 2 – Temperature

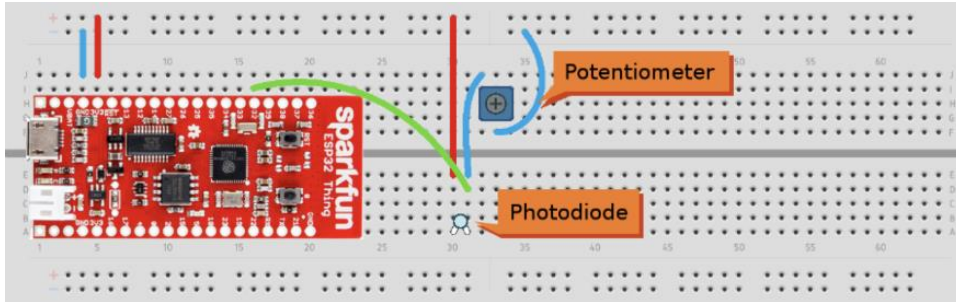


Next, we need to assemble our three-temperature sensor. Insert the temperature sensor to columns (25, 26, and 27) (first temperature), (47,48, and 49) (second temperature), and (52, 53, and 54) (third temperature) so the flat side of the sensor faces the middle of the board.

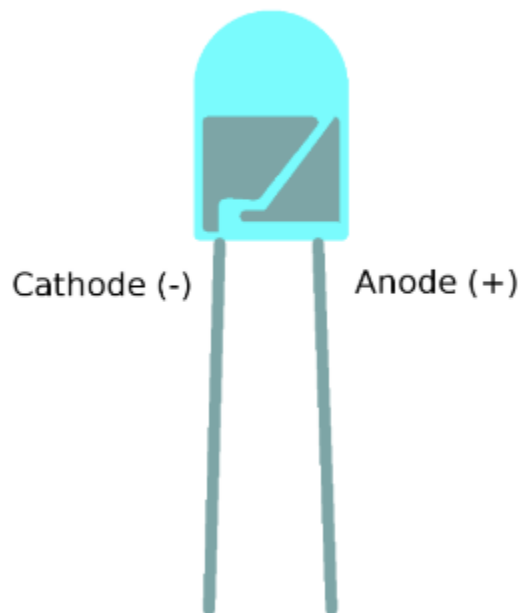
Then connect the first temperature sensor to pin 36 of the ESP32 second temperature sensor to pin 37 on ESP32, and third temperature sensor to pin 38 on ESP32, which should be connected to the row “i” which is next to the ESP32 pin.

Then on the other side of the wire connect to row “j” in the same column as the middle pin of the temperature sensor.

Step 3 – Light



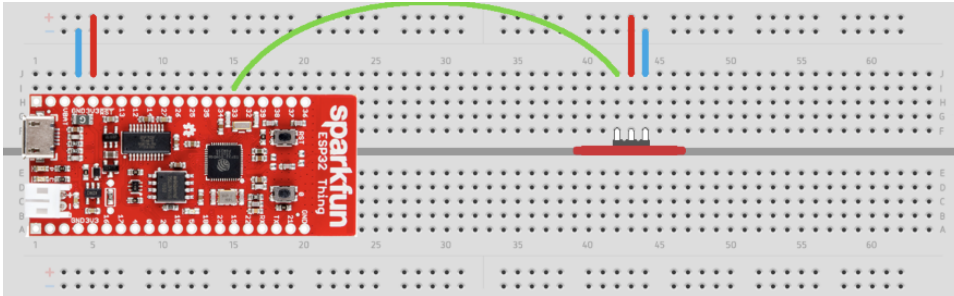
Next, we will be adding the light sensor, The photodiode should be far away from the other components to not be obstructed by wires and to allow for a diffuser to be placed over it. Insert the photodiode into columns 30, 31, and pay close attention to the orientation of the diode.



The photodiode is running in reverse bias, which means the cathode (the negative pin) is connected to the positive power rail, the anode (positive pin) is connected to the rest of the circuit. Looking closely at the photodiode, you will see two small metal components on the inside of the cell; the larger of the two is on the negative side of the device.

The positive pin of the photodiode (column 31) connects first to our output, which is read by pin 32 of ESP32 (column 16). After that, as the diagram shown above, make sure that it is connected correctly.

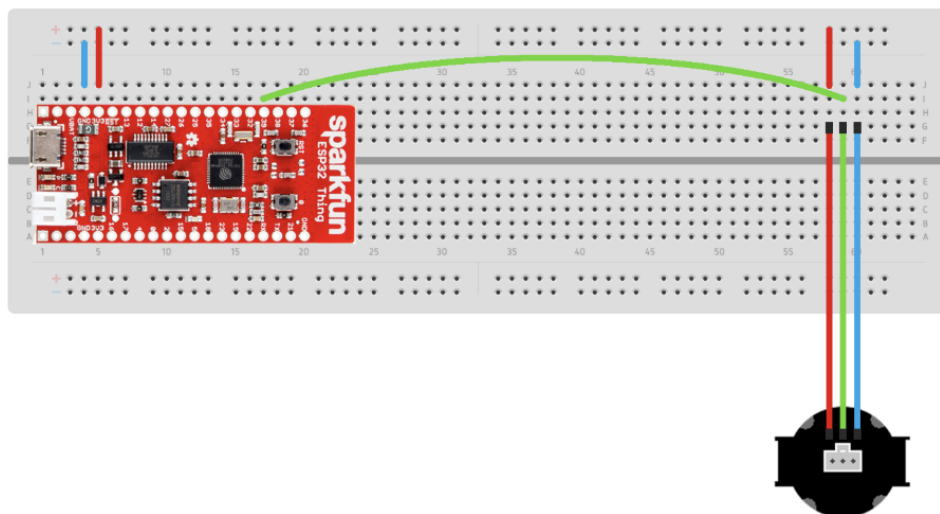
Step 4 – Salinity



Now we will be putting in the water sensor slots into the center groove of the breadboard. Insert the sensor into columns 42, 43, and 44, with the pins facing the top of the board. In this orientation, the middle pin is power, right pin is ground, and the left pin is data.

Then connect the data pin to column 41 to pin 33 of the ESP32 in column 15.

Step 5 – Turbidity



The turbidity sensor is connected to the board by a set of M-to-F jumper wires. Orienting the sensor as shown, the power pin is on the left, the data pin is in the middle, and the ground pin is on the right. Insert the jumper wires into columns 58, 59, and 60, and connect each to the appropriate sensor pin. Then connect column 59 to pin 39 of the ESP32 in column 17.

Buoy Development Environment

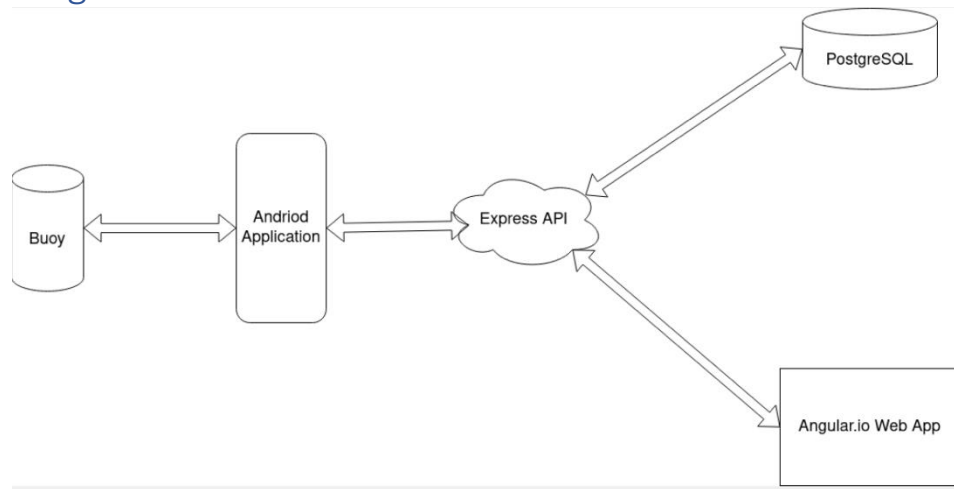
ESP-IDF

The Sparkfun ESP32 uses the ESP-IDF development environment by Espressif. Their comprehensive documentation page, which includes a full getting started guide, can be found here: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>

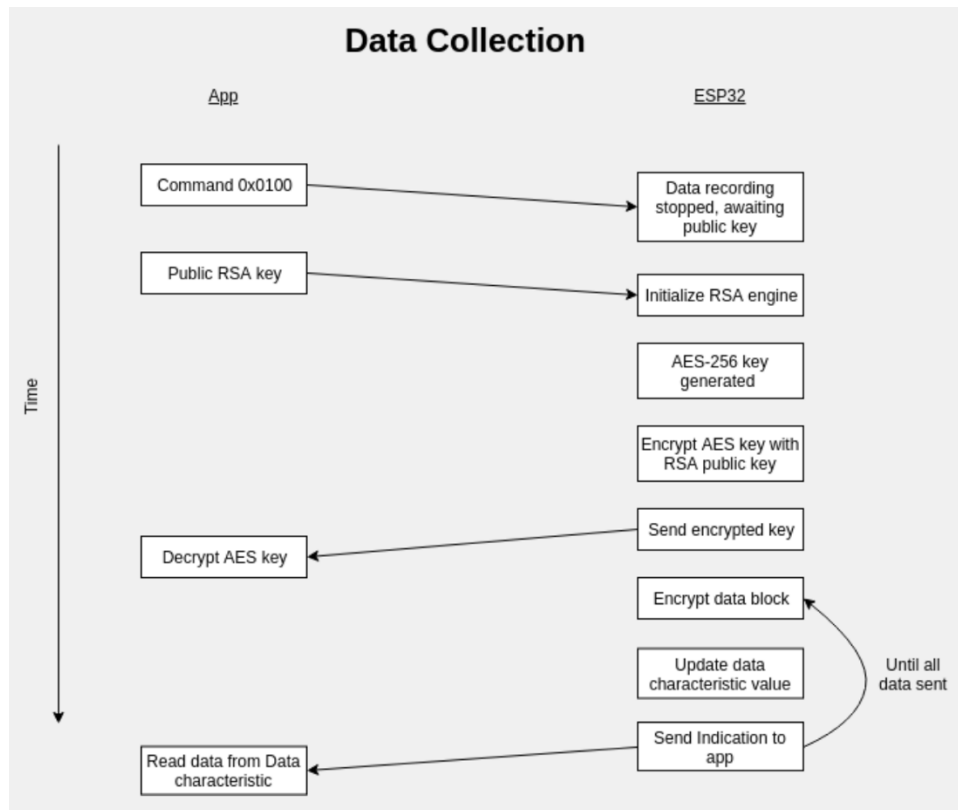
Mobile App

The data stored on the buoy will be transferred to our database via our app. The app will be made with the language dart using flutter/dart, a coding language that will allow the app to run on both Android and IOS. The functionality of this app will be to take the data off the buoy and into the database while also labeling the location that the data was received. The diagram below shows a model of how the data is transferred and connected through the project.

Diagram of Connection



Data Collection Diagram



The app will also be password protected through firebase allowing users to see the data attached to their personal account.

Flutter

Flutter is a state-based language that allows for various Widgets to hold information and perform tasks like they do on a smartphone's home screen. The app that we have made consists of ~4 screens all holding widgets that allow the app to communicate with the buoy.

Website

The website for this project will be password protected requiring the user to register an account before gaining access. The accounts will be made and maintained by a third-party program called firebase that allows users to make accounts that store and remember data specific to that account. Once registered and verified, the user can see the data points collected by the buoys and sent via the app. The website will have a simple and easy to read format that allows the user to see where the data was collected from as well as all five types of data collected by the buoys various sensors. The website is made with a combination of html, JavaScript, and

Transport Stream code. The site's API (Application Programming Interface) is handled through Docker desktop