

SurfTrak

Live Surf Report Display //CS530 Project

Created by:
Logan F

[https://github.com/Thanks
Logan/SurfTrak](https://github.com/ThanksLogan/SurfTrak)

Project Description

Initiative, Background, and Motivation

Why did I choose to make this project?

My motivation this project lies with my number one hobby when I am not busy with school: surfing. I love the sport and culture of surfing so much that it influenced my decision making in creating a project for this class: CS530.

A background for this project and some inspiration comes from an existing app: Surfline. This app shows live data coming from all over the world regarding surf conditions, forecasts, and articles about surfing. My idea about creating a live marquee that could display the surf report in real time seemed like a great idea because I wouldn't have to log into Surfline.

Objective of my Project

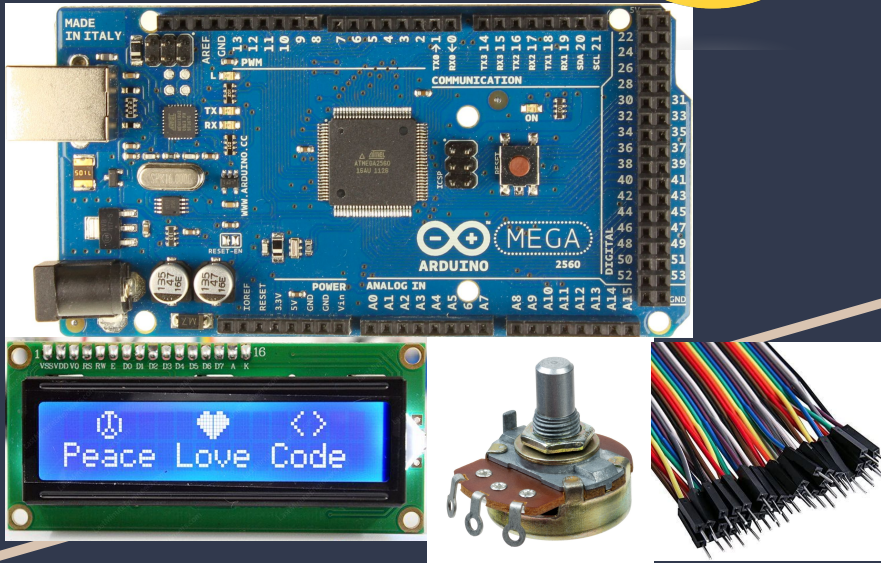
What is my main objective with creating it?

My primary objective of creating my live surf report display is to showcase a live report of the surf for a given location which my database will update to.

What problem does it solve?

Surfers love to know the conditions before they drive all the way to the beach, but cannot always rely on the word of mouth, or the conditions from the last time they surfed, since the ocean is always changing. The solution my surf report display brings is that it will display the live conditions of the surf spot with moderate accuracy so that the surfer can decide for himself whether it's worth it to go or not.

Hardware & Software



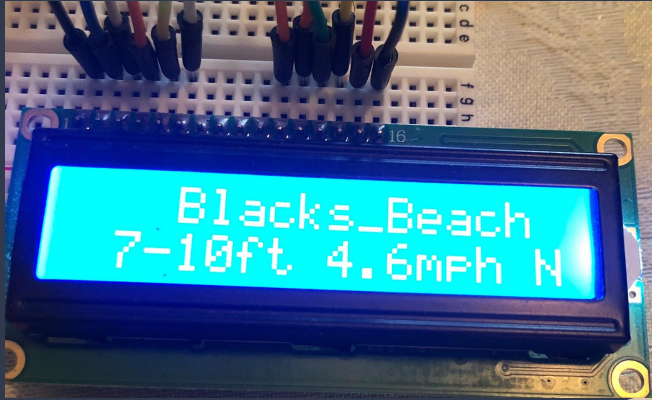
Hardware:

- Arduino Mega 2560
- Breadboard and Wires (Elegoo kit)
- USB connection to Mac
- LCD 1602 Display (Elegoo kit)

Software:

- Python: libraries/modules included Serial.tools, time, requests, re, math, BeautifulSoup
- Arduino IDE: Uses C/C++, with <LiquidCrsal.h>

Breakdown of Functions



Python Functions: In a python script, I utilize the serial.tools module to communicate with the Arduino software. This is done by sharing the USB port as a means of communication, then sending data via a Serial.write command. The python script itself consists of a web scraper I designed which utilizes the “BeautifulSoup” library and “re” module to fetch the HTML source code of the buoy website I needed. Another function used that scraped data to form a wave height prediction using a formula found from Surfline.com All of this data is written into a string where it can be easily transmitted to the Arduino’s source code which runs on the Arduino IDE

Arduino IDE Functions: Once the Arduino receives the string from python via the Serial port, it begins formatting some of the string itself to separate the surf spot location from its data, which includes wave height (ft) and wind speed (mph), and direction (N,W,S,E).

Arduino MEGA 2560: The Arduino microcontroller receives instructions from its source code which transmit through the data input and power ports via wires.

Data Dictionary:

Welcome to WaveTrak!

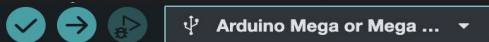
Spots to choose from:

- '1' - San Diego CA, (Black's Beach)
- '2' - Orange County, CA(Trestle's)
- '3' - North Shore of Oahu, HI
- '4' - Cocoa Beach, FL

Type the number of a surf spot, or type 'exit' to end the program:
-> 1

...Connecting to Arduino...

Blacks_Beach 7-10ft 4.6mph N



surftrak_arduinoketch.ino

```
8   int x;
9   String msg;
10  int lineNumber = 1;
11  int cursor = 0;
12
13  void setup() {
14      Serial.begin(115200); // Connects to python with baud rate of 115200
15      lcd.begin(16,2); // This sets up the LCD screen format of 16*2
16  }
17
18
19  // Loop function will receive communication from python script
20  // data is brought in one long string so that its easy
21  // for the arduino for it to read.
22
23  // The C++ code can also parse some of it for purposes of
24  // formatting onto the LCD screen
25
```

```
def getHTMLdocument(url):
```

```
def waveHeightFunc(H,P):
```

```
def parseSwellSite(urlNum):
```

```
def parseWindSite(urlNum):
```

```
def summarizeData(spotID):
```

```
# ARDUINO COMMUNICATION SETUP
```

```
ports =
```

```
serial.tools.list_ports.comports()
```

```
serialInst = serial.Serial()
```

```
serialInst.baudrate = 115200
```

```
serialInst.port =
```

```
"/dev/cu.usbmodem1101"
```

```
serialInst.open()
```

Competitors

Who are our competitors?

The only similar product which is given by Surfline forecasters and similar websites are the forecast of the surf for a given spot.

However, there does not yet exist a forecast display like mine which can show the live surf report on a display.

Project Management

Breakdown of Tasks, and Distribution of Tasks.

Design, Implementation, Testing:

I will be responsible for all facets of the project, including design, implementation, and testing since I will be working alone.

Project Milestones

9/15: Purchased all Materials (Arduino Mega Set, LED Displays, Breadboards, wires, Arduino IDE)

9/20: Began assembling all wiring materials included with the arduino in order to set up LCD display.

9/21: Successfully assembled.

9/22: Began to test programming features on the Arduino to LCD display connection.

9/23: Began working on code scraper to inspect element of forecasting websites

11/30: Finished code scraper and began testing with display

12/3: Finished testing with display, ready to present

Waterfall Method

Purchase all Pieces ->

Assemble all pieces ->

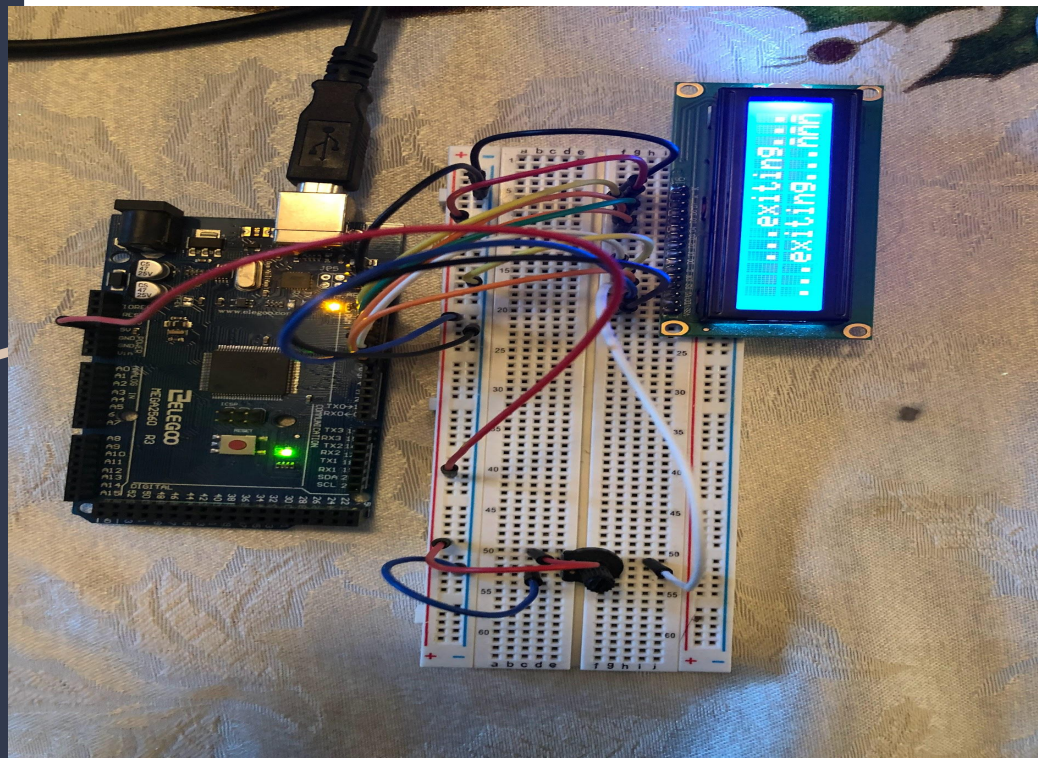
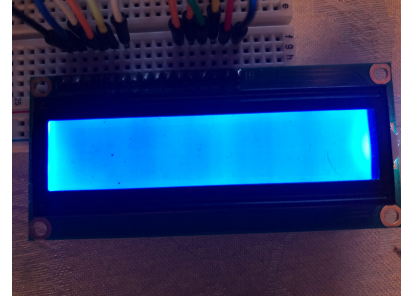
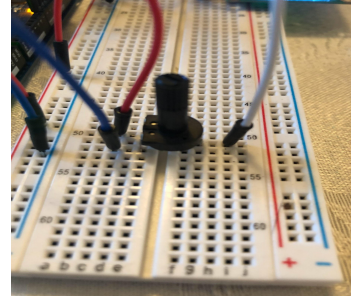
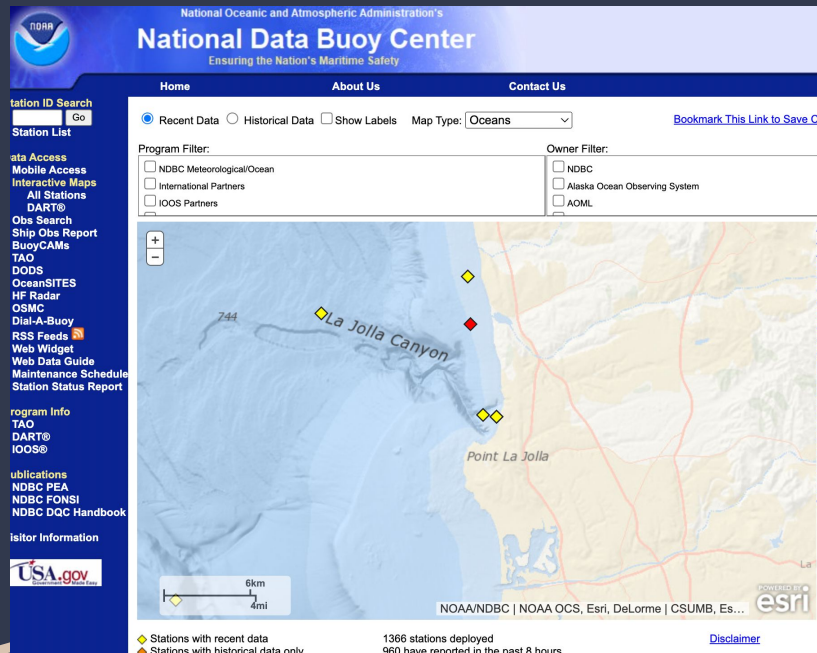
Connect pieces on computer ->

Begin implementing on computer ->

Test on computer ->

Repeat ->

Project Photos



References

Surfline:

<https://www.surfline.com/surf-report/blacks/5842041f4e65fad6a770883b>

NDBC:

https://www.ndbc.noaa.gov/station_page.php?station=46275

WillyWind:

<https://wind.willyweather.com/ca/san-diego-county/tristes.html>

Formula:

<https://icce-ojs-tamu.tdl.org/icce/article/view/2763/2427>

Github: <https://github.com/ThanksLogan/SurfTrak>