



Weather IOT Device Project

By: Dawson Burgess, Alphonse Crittenden, and Lucien Lee

Project Description



Project Overview

- The IOT weather device will utilize various sensors and web API's to get the relevant data for the device and display it to the user
 - This device will be built with FreeRTOS
 - Utilize OpenWeatherMap API
 - Utilize GeoAPIIfy
 - Utilize pool.ntp.org
 - Gives time in UTC time by default
- The IOT device will use the following sensors in harmony with the various APIs:
 - ESP32 Feather
 - GPS Module
 - Light Sensor
 - Pixels
 - Temperature and Humidity Sensor
 - OLED Display Screen
 - Buttons for controlling the Device

Open Weather Map API

This API provides the device with the weather location based off of the current latitude and longitude.

```
{
  "coord": {
    "lon": -117,
    "lat": 46.73
  },
  "weather": [
    {
      "id": 804,
      "main": "Clouds",
      "description": "overcast clouds",
      "icon": "04d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 283.65,
    "feels_like": 282.51,
    "temp_min": 283.65,
    "temp_max": 283.65,
    "pressure": 1013,
    "humidity": 67,
    "sea_level": 1013,
    "grnd_level": 919
  },
  "visibility": 10000,
  "wind": {
    "speed": 3.75,
    "deg": 273,
    "gust": 3.98
  },
  "clouds": {
    "all": 100
  },
  "dt": 1714249103,
  "sys": {
    "country": "US",
    "sunrise": 1714221505,
    "sunset": 1714272744
  },
  "timezone": -25200,
  "id": 5601538,
  "name": "Moscow",
  "cod": 200
}
```



OpenWeather

- Free open source API
- Allows for up to 60 calls an hour for free (One a minute maximum)
- Has 16 different weather type returns
- Able to operate without sensors if needed
- In order to query:
 - Requires Longitude
 - Requires Latitude
 - Requires API key
- Returns JSON string



GeoAPIfy - Time API

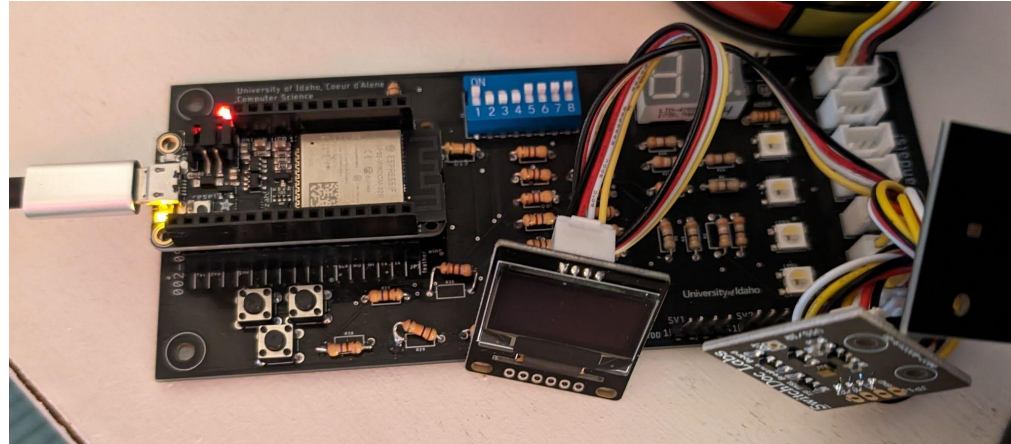
- This API returns the correct day and time to the device based on the current gps location.
- This will return the proper numbers to set the offsets so the time and day from the pool.ntp.org is able to be set correctly

```
1  {"type": "FeatureCollection",
2  "features": [{"type": "Feature",
3    "properties": {"datasource":
4      {"sourcename": "openstreetmap",
5       "attribution": "@ OpenStreetMap contributors",
6       "license": "Open Database License",
7       "url": "https://www.openstreetmap.org/copyright"},
8      "country": "United States",
9      "country_code": "us",
10     "state": "Idaho",
11     "county": "Latah County",
12     "city": "Moscow",
13     "postcode": "83843",
14     "street": "South Washington Street",
15     "houenumber": "604",
16     "lon": -116.9997655505071,
17     "lat": 46.7299978,
18     "state_code": "ID",
19     "distance": 11.029992083869553,
20     "result_type": "building",
21     "formatted": "604 South Washington Street, Moscow, ID 83843, United States of America",
22     "address_line1": "604 South Washington Street",
23     "address_line2": "Moscow, ID 83843, United States of America",
24     "category": "building",
25     "timezone": {"name": "America/Los_Angeles",
26       "offset_STD": "-08:00",
27       "offset_STD_seconds": -28800,
28       "offset_DST": "-07:00",
29       "offset_DST_seconds": -25200,
30       "abbreviation_STD": "PST",
31       "abbreviation_DST": "PDT"},
32     "plus_code": "85R5P2H2+X3",
33     "plus_code_short": "H2+X3, 83843 Moscow, United States",
34     "rank": {"importance": 0.0000099999999995449,
35       "popularity": 1.0931301584485722},
36     "place_id": "511cc6a528fc3f5dc05973936291705d4740f00102f901894b212a00000000c00203"},
37     "geometry": {"type": "Point",
38       "coordinates": [-116.9997655505071, 46.7299978]},
39     "bbox": [-116.9998558, 46.7298701, -116.9996758, 46.7300472]}]},
40  "query": {"lat": 46.73,
41    "lon": -117,
42    "plus_code": "85R5P2H2+X2"}}
```

Device Implementation

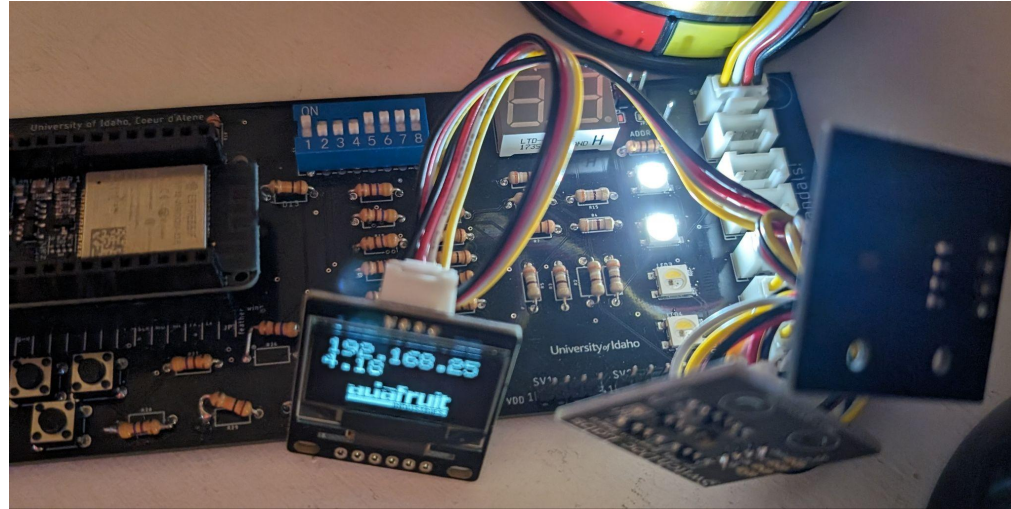
Setting up the Vandalino Board

- The proper configuration for this device to function with the Vandalino board is seen in the picture.
- The GPS sensor will be connected to a GPIO grove connector and the rest of the sensors will be connected to the onboard I2C grove connectors



Starting the Device

- On device startup, the leds will turn on to let the user know the board is connected to WiFi and the GPS signal has been found
- Additionally, the local IP address for the hosted website will be displayed on the OLED screen



Accessing the Website

- If the user is able to access the website, then the board is fully up and functional!

Weather Data Server - RTOS Final Project
By: Dawson Burgess, Alphonse Crittenden, and Lucien Lee



Latitude: 46.731117
Longitude: -117.004585

Date/Time: Monday, April 29 2024 20:50:37

Light Sensor Data: 7.12

Room Temperature: 71.10

Additional Information: Loading...

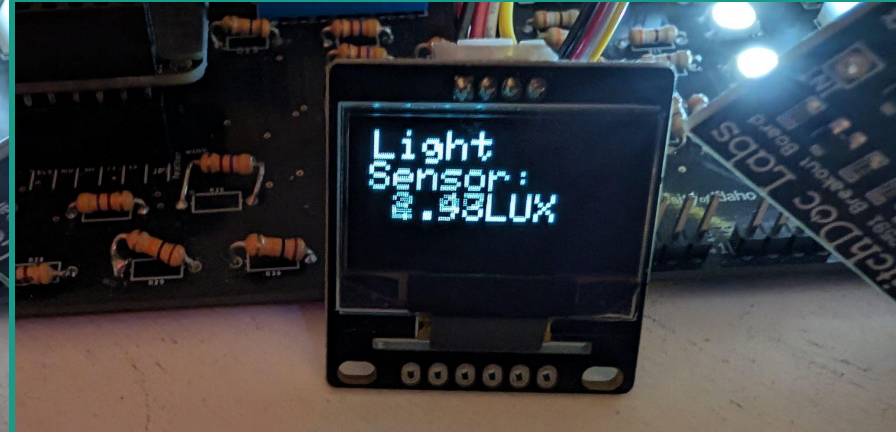
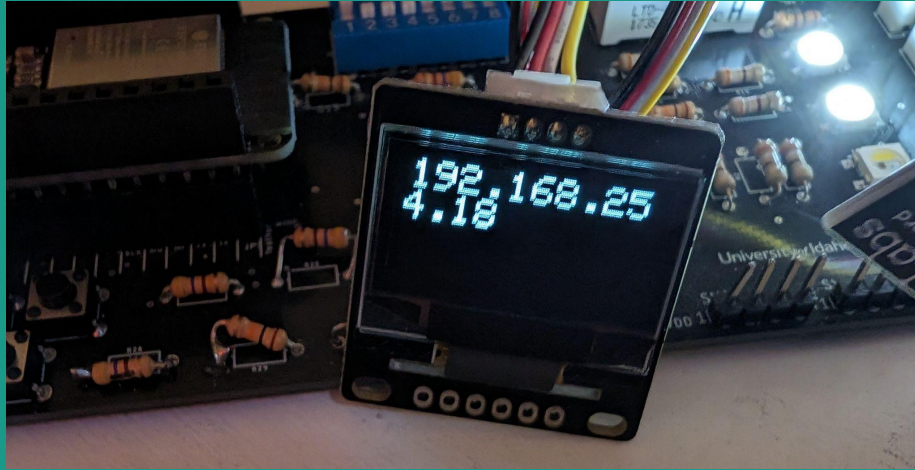


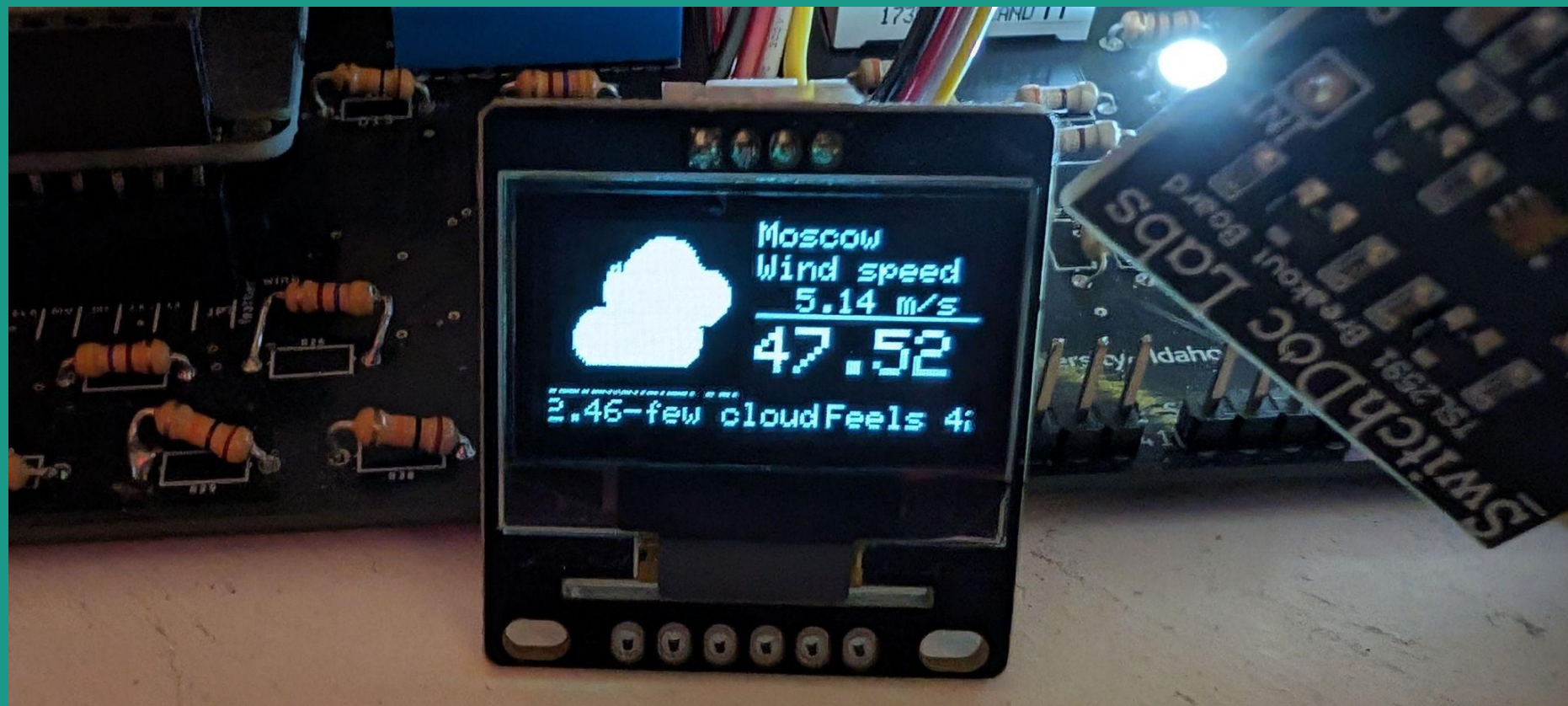
Using the Buttons

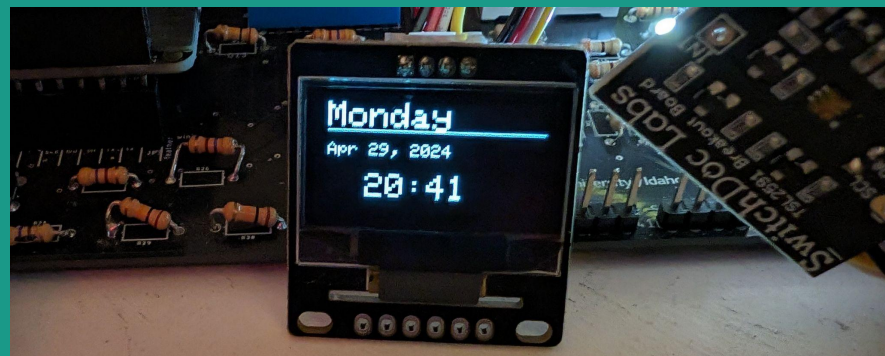
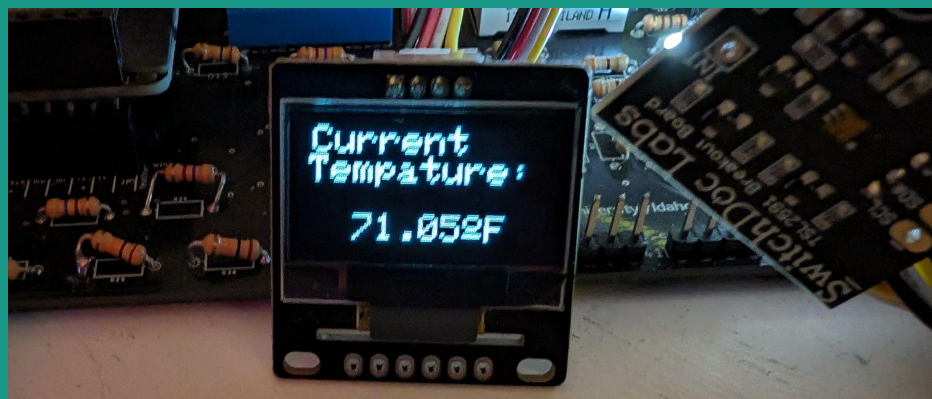
The on-board buttons allow the user control over several device functions

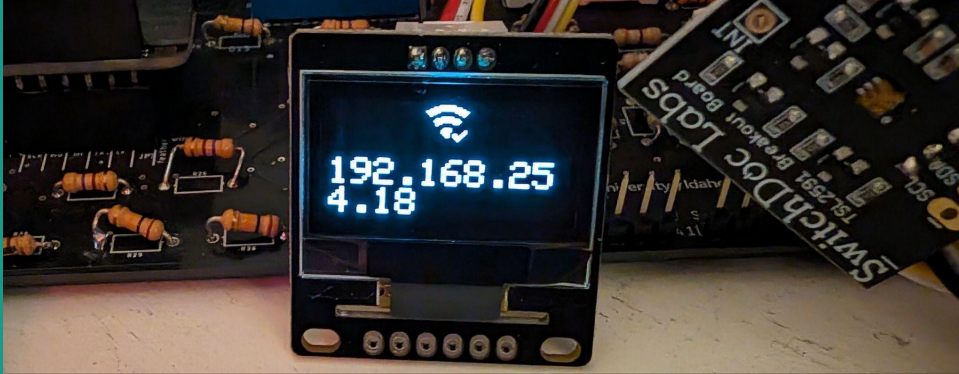
- The on-board buttons accomplish the following things:
 - Button 1 (left most button)
 - This button acts as a toggle for the LEDs.
 - If the LEDs are on, and the user wants them off then the first button can be pressed
 - Button 2 (right most button)
 - This button acts as a reset button for the program.
 - If the user would like to reset the device, then they should hit this button and it will restart
 - Button 3 (bottom button)
 - This button will allow the user to cycle through what is displayed on the screen
 - These are all on a loop, and can be accessed over and over again

Possible OLED Displays









Thank you! Any Questions?
