JACOBS UNIVERSITY

MEGI002-210214 – Geoinformatics Lab

Project Report

Integrated Water monitoring system

Topic Description:

Dashboard for Sensor and particulate camera data.

Group members:

Vijaya Nawale &

Vaibhav Apraj

May 24, 2020

Table of Contents

1. Abstract

This paper is about how we built a reporting dashboard using analytical web applications based Python framework **dash** with graphs. We describe organised data presentation of the data collected from GPS, particulate camera and sensors as temperature, pH and EC.

2. Introduction

Data from water quality sensors are used to make decisions on a range of management issues. In this project there are 3 sensors namely Temperature, pH, EC and particulate camera used to place in the water to convert water quality parameters into measurable values to give input to the controller.

The aim of this report is described in the following section as background gives ideas about hardware devices used to integrate systems to design the buoy. Methodology describes data collection and technologies used for the Dashboard development. Data visualization includes python framework description. Lastly in the results and conclusion we concluded about our understanding from collected data and information from the visualization.

As a team we worked together in every task and equally responsible for overall project work. To differentiate with more focus, Vaibhav Apraj is involved little more in data collection and tool selection and Vijaya Nawale in python development for dashboard and data visualization.

3. Background

The integrated system of Monitoring of the water quality via sensors. The system is built using Raspberry Pi, Arduino, sensors, particulate camera and GPS.
Devices used:

1. Raspberry PI 3 b+ with Raspbian installed.
2. Wi-Fi Hydroponics Meter Kit.
3. Particulate Camera
4. GPS Puck G-Star IV.
5. Power Source.
6. Raspberry Pi Camera Board v1.3 (5MP, 1080p).
7. Sensors
    a. Temperature
    b. pH
    c. EC

Software Versions:

1. Python 3.8.2 for scripts and Visualization
2. Python framework Dash, pandas, plotly.express

4. Methodology

4.1 Data Collection

The data collection system consists of multiparameter sensors and wireless devices for sensor data transmission to controller.
Data is collected with following steps:

1. 3 sensor's (Temperature, EC, pH) data pushed to the serial port.
2. On Raspberry Pi run Python script to get data from serial port to Excel file format
3. Image data collection from Particulate camera.
4. Date, time and position data from GPS.

4.2 Dashboard Development & Data Visualization

After collecting data from sensors in measurable format some sensible information should be extracted from that data and there is the requirement of a dashboard to visualize data in readable forms.

We learned about python's open source library *Dash* while searching for the technology to develop dashboard applications which found useful to create reactive dashboard applications with *Plotly.* It's super simple with Dash to design and develop GUI. We decided to use python framework *Dash* for creating an interface to view the visualizations made with *Plotly*. Plotly's Dash framework provides various components to layout the graphs, images, textareas etc. [1].

The purpose of visualization is to display sensor data with graphical representation. This way to get some useful information from the collected data. We chose Plotly to construct an information visualization in pure Python [1].

To design and develop the dashboard we used html integration in the python. Various packages used are imported in python as:

```
import dash
import dash_html_components as html
import dash_core_components as dcc
```

```
import datetime
from datetime import datetime as dt, datetime
import pandas as pd
import plotly.express as px
```

Following are the purpose of using the above imported libraries.

dash_html_components - Has all html tags used to layout and style the graph and other tools.
dash_dcc_components - Provides an interface to view graph visualization.
pandas - Used to read data from *csv* file.
plotly.express - Makes available a variety of graphs like line, scatter, box etc.

Data format used is csv file, to read data from file *pandas* package and different graphs from *plotly.express* package are used [2].

To plot in line graph *.line()* function is used which takes attributes as
(data, x_axes_value, y_axes_value, title)

```
fig = px.line(df, x='EC', y='TIME', title='EC sensor data')

app.layout = html.Div(children=[
        dcc.Graph(
                id='example-graph-2',
                figure=fig
        )

], style={'columnCount': 2})
```

To plot scatter graph *.scatter()* function is used with attributes as
(data, x_axes_value, y_axes_value)

```
fig = px.scatter(df, x="EC", y="TIME")

app.layout = html.Div(children=[
                dcc.Graph(
                        id='example-graph-2',
```

```
                figure=fig
            )
    ], style={'columnCount': 2})
```

Below figures show data visualization results. In all the graphs below Y axes values are time values and X axes shows different sensor's values.

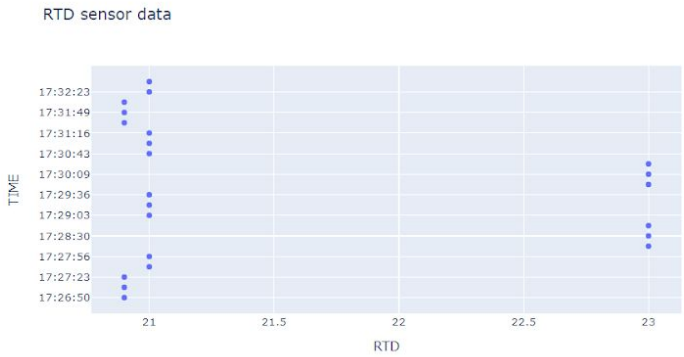

Figure 1. Scatter graph of temperature sensor data.

Y axes values are time and X axes shows temperature sensor's values. Graph shows the temperature sensor data(temperature of the drinking water) according to time.

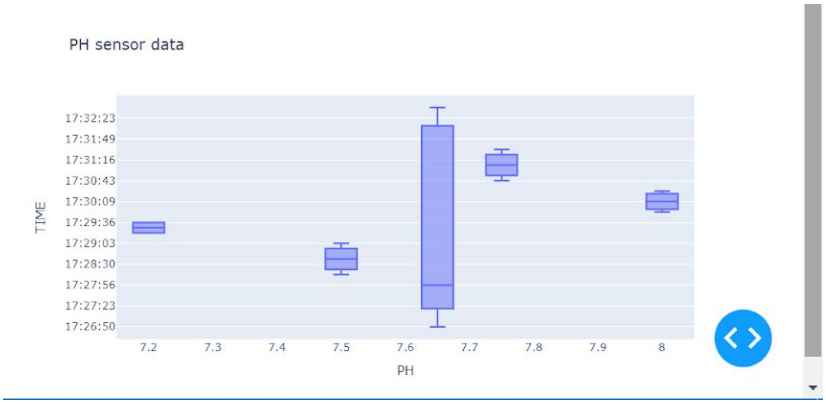

Figure 2. Box plot representation of pH sensor data.

Y axes values are time and X axes shows pHsensor's values. Graph shows the pH sensor data(pH value of the water) according to time.

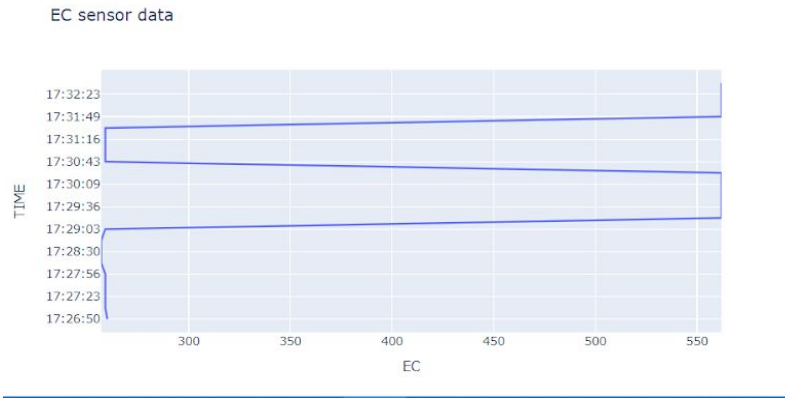

Figure 3. Line graph EC sensor data.

Y axes values are time and X axes shows EC sensor's values. Graph shows the EC sensor data(electrical conductivity of the water) according to time.

5. Results and Conclusion

Combined all visualizations in one frame to design a user friendly dashboard.

Figure 4. Dashboard.

Plotly's Dash application runs in the web browser with url generated after executing .py file in the terminal. Every time Pi restarts we just need to refresh the browser page to see least recent values captured from the sensors.

Data varies with the variety of the water like ocean water, tap water etc.
As due to the current situation we could not test in ocean water or elsewhere outside. We tested the sensor data with drinking water therefore, measured result values represented in the visualization are as per the drinking water quality.

1. Temperature sensor - Measures temperature in the range of -5**°C to** 50**°C** In water chemistry temperature is a crucial factor. Temperature impacts the dissolved levels of oxygen in water, photosynthesis rates etc. For drinking water temperature sensor values vary between 20**°C** to 23**°C.**
2. pH sensor - Used for water test to indicate acidity and base present in water. Express data in 0-14 numeric scale. pH 7 is neutral. pH sensor data values for drinking water are in range of 6.5-8.

3. EC sensor - Measures electrical conductivity of the water. Used to trace water's hydrological and chemical processes. This shows the ability of the water to conduct electric current. Tap water EC sensor values are in range 50 - 800 uS.

4. Particulate camera image - With image data from a particulate camera underwater to measure settling rate of the particles.

5. GPS Date time and location - GPS provided datetime and location data used to display with sensor data. Used a slider to change sensor data values depending on the datetime.

Appendix A Code Repository

All the code implemented in this report and also the output files are available at the Dropbox - https://www.dropbox.com/s/rqpflkq2r57q8hx/GeoLab2020_Dashboard.py?dl=0

References

[1] https://plotly.com/dash/
[2] https://medium.com/plotly/introducing-plotly-express-808df010143d