Group 9
Vincent Carreon
Brandon Chau
6 May 2025

End-to-End IOT System Report

System architecture and approach summary:

When approaching an end-to-end Iot system, effectively it can be separated into three parts, a client-server connection, working devices that can send data, and a functional database that takes in the device metadata and sends it to the server which then sends the information to client with queried. To prevent users from sending an invalid query, we would build around a simple prompt to input either 1, 2, or 3 to test each query and 4 to end the server-client connection; otherwise, a message pops up to ask user to send a valid message. In addition, we created an uml structure to organize how objects were created in Dataniz to then base queries on encountered primary/foreign keys through grouping as the metadata payloads only provide values and respective keys, but we must ensure that they are uniquely identifiable.

Research findings on IoT sensors and data:

Ammeters (ACS712): Converted from ampheres, measure the energy, in kWh, used by the device.

- Operates on 5V
- (Initial value) Amp → (Final value)kWh
- kW = Amps × Volts / 1000

Moisture Meters (DHT11): Measures the moisture content, in relative humidity %, used by the device.
WaterFlow (YF-S201): Measures the amount of water used in a cycle.

- (Initial value) Liter per min (LPM) → (Final value) Gallons per min (GPM)
- GPM = LPM × 0.264172

Thermistor (Mlx90614): Measures the internal temperature of device

- (Initial value) Celsius → (Final value) Fahrenheit
- (C × 9/5) + 32 = F

Location: All devices were established to retrieve data in (36 longitude, -119 latitude)
Time: All sensors have been active since April 29th, with each sending data at minute intervals. Originally these values were in UTC, thereby we used the built-in sql function while querying to convert to PST

Dataniz Metadata usage (or lack thereof):
Effectively, the Dataniz metadata was successfully able to send data collected by the device sensors to a destination database in Neon. In effect, the metadata was crucial to organizing data collected from payloads, therefore providing accurate calculations.

Implementation Details (algorithms, calculations, unit conversions, etc.):

Our code uses the pyscopg2 python package to allow for a connection between the server and NeonDB database via connection link and can be accessed with a cursor object. Upon implementation in the code, the server program is fed data from the database which is then compiled into respective algorithms depending on requested query from client. For query 1, finding the average moisture is solved via creating a list of all accumulated moisture values in the given span of three hours and dividing it by the length of such list, providing a mean value. For query 2, like query 1, take all data from the dishwasher and divide

it by the total values collected. For query 3, to find which device used the most electricity can be solved by distributing each device their values for their ammeters in respective lists and comparing the sum of each list to each other.

Encountered Challenges:

Primary challenges that came with the project were the initial construction of code to connect it to the databases on Neon. As we had most of the framework for connecting the virtual machines together and have working devices sending data to NeonDB, most of our time was spent on building/testing the code. After doing research and implementing the psycopg2 package, creating a valid query to provide results for the 3 expected queries became easier.

Dataniz.com Feedback:

At its best Dataniz is an insightful tool to see a realized Iot system, the organized structure is initially challenging but becomes clearer with more experience and experimentation. However, there are a few suggestions that we believe could help maximize its potential and make it more intuitive for further development.

- Release Patch notes to update users of any difficulties encountered when fixing areas of issues
- Have a demo video to showcase features
- Change the layout of the devices and sensors, have a drop-down menu for a device and their sensors
- Implement a filter search to make looking for sensors easier and to prevent duplicating sensors in a device (while not many sensors were used, it can be overwhelming if more topics/devices/boards/sensors are added)
- Allow for min-max values on sensors to be inclusive to default values

Resources:

Time conversion

https://old.reddit.com/r/SQL/comments/te5rro/how_to_convert_from_utc_timestamps_to_pst_time/i0nv20c/

Python-Database connection

https://www.geeksforgeeks.org/executing-sql-query-with-psycopg2-in-python/