Department of Computer Science

Temperature and Humidity Sensor Box

By

Waleed Binselim
Hoang Nguyen
Stephanie Fernandez

Submitted: 05/12/17

Project Description:

For this project, the raspberry pi had to be used to create a building's heat and humidity profile. This was to be done by using sensors to collect temperature and humidity data and then saving it to a local database. A robust, attention-free, easily deployable system had be created that integrates data from different sources and allows a non-technical user to access the building data at any specified time. A website had to be created to save the data from multiple pi's to a remote server that authorized persons could log into. The pi's had to be stored inside of a 3D printed box, and the data from the pi's had to go into an sqlite3 database. From the database, the data had to be transferred into a website, where the three group members can log onto with different login information. A graph for the individual data profile and stacked curves for the cumulative view needed to be made.

Project Goals:

- The pi's should be inside a box and be easy to use
- The data from the raspberry pi's should go onto a database
- The website should be able to access the database in order to get data
- The website should be able to receive data from multiple pi's
- The website should only be accessed by authorized users
- New users should be able to create a username and password to access the website
- The raspberry pi's should automatically sync data recorded to the website created
- The user should be able to search for temperature and humidity data by date
- The website should automatically create individual data profile graph based on searched data

Component Break Up:

- *Collecting data*: The pi collected the data from the sensor by running a python file
    - Hardware: Aosong DHT22 Temp/Hum sensor, Raspberry Pi 3,
    - Software: Python IDLE, Terminal
- *Data into the database*: The data was put into a database by running a python file.
    - Hardware: Raspberry Pi 3
    - Software: SQLite3, Python IDLE, Terminal
- *Login/create user*: The user could log into the website or create user if no username previously existed.
    - Hardware: Macbook, Raspberry Pi 3
    - Software: Google Chrome, Chromium, Atom, Terminal

- *Database to website*: The website read from the database and displayed data by searched date on the display page.
    - Hardware: Macbook, Raspberry Pi 3
    - Software: Google Chrome, Chromium, Atom, Terminal
- *Data to graph*: After the website displays the data for the searched date(s), a graph is automatically generated with that data.
    - Hardware: Macbook, Raspberry Pi 3
    - Software: Google Chrome, Chromium, Atom, Terminal
- *3D printed box*: Casing for the pi and breadboard.
    - Hardware: MakerBot Replicator 3D Printer
    - Software: Solidworks, Makerware

Implementation:

The pi was hooked up to the Aosong DHT22 Temp/Hum sensor using a breadboard and a 10 kOhm resistor. This allowed the pi to read temperature and humidity data from the sensor by running a python file. In the same file, it stores the data every time it reads from the sensor into an SQLite3 database. The file was run for nine hours and it recorded and stored data every half an hour; this was done three times. The data then needed to be put in a website.

The website was made using Flask, which is a Python web framework, Jinja2 template engine, which came with Flask, and Bootstrap, which uses HTML elements and CSS properties. The website requires a user to log in to display the data that was logged over time. The user can also register a new username and password in order to access the data. The username and the password are both stored in the SQLite3 database so when the user tries to login, the website will check the database for the username/password combo. If it is incorrect, or does not exist, the website will display an error page.

When the user logs in with authorized information, the website automatically takes the user to the display page, where the data could be found by searching by the date. When the user searches a date, a table of the date/time, temperature, and relative humidity is generated using the same database that the pi uses.

A button was made for plotting the temperature vs time. The user can search for the date then click the graph button. The graph is made by using plotly, a python library for creating interactive graphs.

A flowchart of the processes in this project are shown in Figure 1 below.

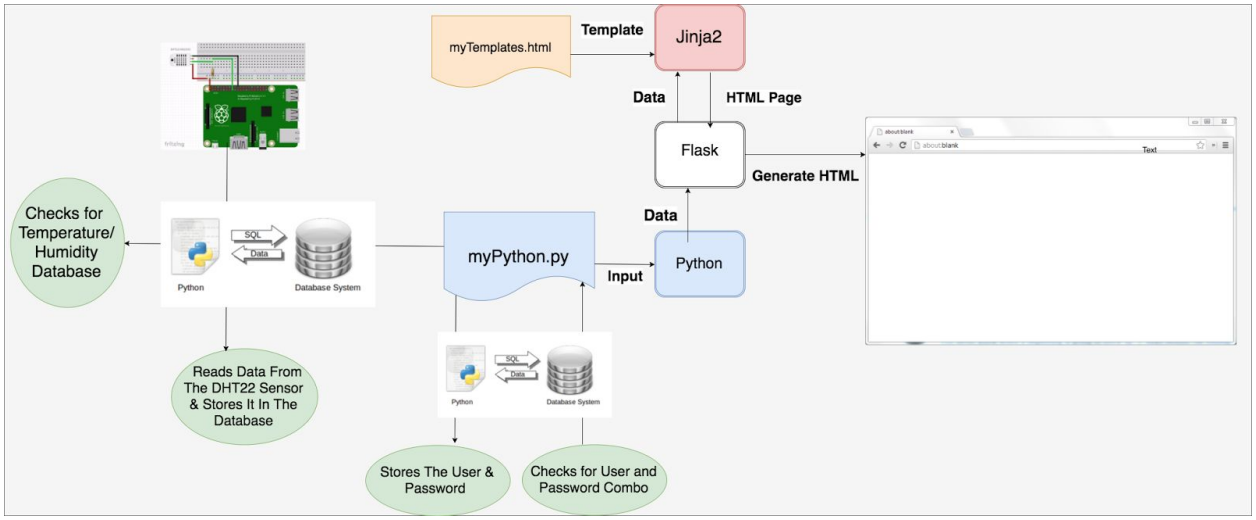

Figure 1: Flowchart of the implementation process.

Achievements and Challenges:

Good Aspects: The pi collects data using the Aosong DHT22 Temp/Hum sensor. The data goes into a database, which then gets automatically uploaded on a website. An authorized user can access the data and search by date. If a user does not exist, they can create a username and password. When data is searched for, a button links to a graph of the data.

Bad Aspects: Graphs of data per day were not able to be automatically generated by the website. The graphs had to be made beforehand and uploaded to the website with a button to show them.

Easy: It was simple to connect the sensor to the pi and collect temperature and humidity data. It was easy enough to create the website and a simple login section.

Tough: It was tough to create a "create user" part of the website and to automatically generate a graph from the searched data. It was not difficult to create the graph but it was difficult to make it automatically generate on the website based on the searched data by the user.

Achieved: The pi successfully collects data using the temperature and humidity sensor. The data that the pi collects can be loaded onto the website automatically and the user can search for the data by date, as well as graph data by date. The user can successfully log into the website or create a username and password, if needed.

Not Achieved: Reading data from multiple pi's was not achieved. Automatically generating a graph of data was not achieved. A stacked curve graph was not achieved.

Group Member Contribution:

Stephanie Fernandez designed and 3D printed the box that the pi's had to go inside of. It first started with sketches and then it moved onto Solidworks once the project was half underway. The set up had to be finalized in order to make a box for the pi and the breadboard. Once the setup was set and will not be changed, the dimensions were taken of the pi and the breadboard. Finally, a 6x6x6 inch box was made with two sides cut out: the top lid and the side lid. The top lid was designed with a 3.5x0.6 inch rectangular hole for the breadboard so that the sensor would be open to the atmosphere. The breadboard would be connected to the Raspberry Pi inside the box. The side lid was then designed with a 1x1 inch rectangular hole at the bottom so that the power cable and HDMI cable could both come out of the box, easily. After consulting with the 3D printing lab, however, the box could not be printed because it needed to be sized down. The box was then completely redesigned and made to be a 5x5x5 inch box with the same hole dimensions for the two lids.

Hoang Nguyen started off by trying to host a website with the Raspberry Pi by installing Apache which is a popular web server application that allows the pi to serve web pages. With this software, the website can be edited, such as adding data, creating users, or drawing a graph. To access the website, the IP address of the pi is needed. The issue with this method is that if the IP address of the pi changes, the entire website will go down. This was learned when a connection was attempted outside of the university's Wi-Fi; the website could no longer be accessed. Therefore, another method had to be used. In addition to this, the plan was to leave the Raspberry Pi on during the presentation so this led to another problem. The pi would not be strong enough to host a server so it would take too long to get to the website. Therefore, the plan changed to using Flask. Hoang assisted in the website creation and in the attempt of automating the graph for the data. After much effort, graphs was then made with the x-axis being the time and the y-axis being the temperature or relative humidity. These graphs were made for each day of data and the graphs were uploaded to the website with a button to show the graphs depending on the date searched.

Waleed Binselim started off by connecting the Aosong DHT22 Temp/Hum sensor to the pi by using a breadboard. The data read by the pi was then put into an SQLite3 database. The website was then made using Flask, and Flask-Bootstrap for the layout. In the website, a login and "create user" were made using Flask-SQLAlchemy, an extension of Flask that adds support to the python library SQLAlchemy which helped to store the username and password into a

database. After logging in, the user could type in the date, then a table would be outputted from that data by accessing the SQLite3 database. The user can also generate a temperature vs time graph by typing in the data in the textbox . The graph is made by using Plotly, which is a python graphing library that makes interactive, publication-quality graphs online.

Resources

[1] "Welcome." *Welcome | Flask (A Python Microframework)*. N.p., n.d. Web. 12 May 2017. <http://flask.pocoo.org/>.

[2] "Welcome to Jinja2." *Welcome to Jinja2 — Jinja2 Documentation (2.9)*. N.p., n.d. Web. 12 May 2017. <http://jinja.pocoo.org/docs/2.9/>.

[3] Adafruit. "Adafruit/Adafruit_Python_DHT." *GitHub*. N.p., 28 Feb. 2017. Web. 12 May 2017. <https://github.com/adafruit/Adafruit_Python_DHT>.

[4] "Flask-SQLAlchemy." *Flask-SQLAlchemy — Flask-SQLAlchemy Documentation (2.1)*. N.p., n.d. Web. 12 May 2017. <http://flask-sqlalchemy.pocoo.org/2.1/>.

[5] "11.13. Sqlite3 — DB-API 2.0 Interface for SQLite Databases." *11.13. Sqlite3 - DB-API 2.0 Interface for SQLite Databases — Python 2.7.13 Documentation*. N.p., n.d. Web. 12 May 2017. <https://docs.python.org/2/library/sqlite3.html>.

[6] "Pressure in Flask | Scatter Chart Made by Bylee | Plotly." *Plotly · Make Charts and Dashboards Online*. N.p., n.d. Web. 12 May 2017. <https://plot.ly/~bylee/35/pressure-in-flask/#plot>.