Raspberry Pi Web Application

1. User Requirements

* The application is a web application to which you can connect from any device such as laptop, phone, tablet.
* The application is a monitoring application which plots CPU and RAM usage.
* The application has a temperature and humidity sensor which will tell you that info from either inside or outside wherever you put the raspberry pi.
* The application is simple to use, it starts whenever you plug in the raspberry pi using a script to start the program.
* Because the application starts when the raspberry pi is plugged in, we can use the application remote.
* The application does refresh by itself with no need to refresh the page.

2. System Architecture

**Laptop Consumer**

**Web Application**

**Mobile Consumer**

**Web Server**

**(Raspberry Pi)**

**)**

**Sensor Extension**

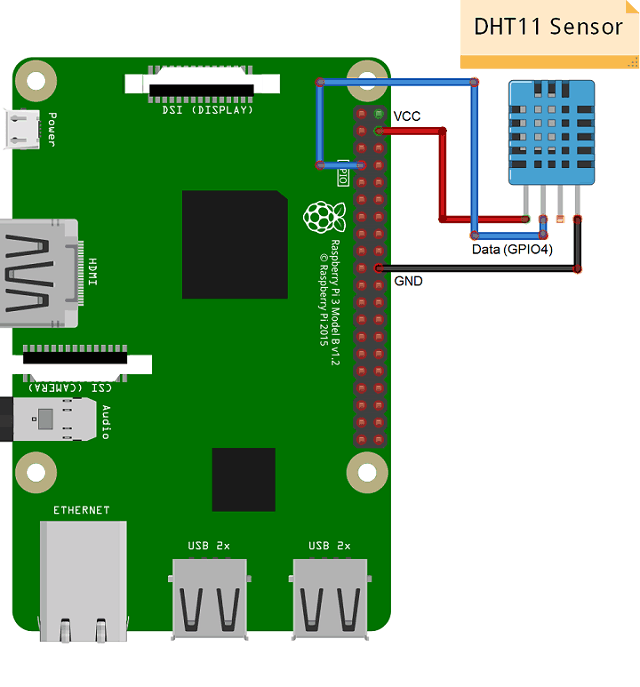
Sensor extension is, in our case the temperature and humidity sensor which has the sole purpose to acquire information from the sensor.

The Web Server is the server on raspberry pi to which we connect in order to see the results of our application.

The Web Application is the whole functionality of plotting the CPU and RAM usage and showing us the temperature and humidity on the webpage.

Laptop consumer and mobile consumers are two different consumers which access our application from two different devices.

3. Hardware overview



DHT11 sensor measures and provides humidity and temperature values serially over a single wire. The wiring is easily seen in the picture.

4. Software Overview

**Start-up Script**

**Python Modules**

**Web Application**

4.1 Start-up Script

The script is simple and it only points to the program so we don’t actually need to enter and start the application so it can be used remotely.

The script is:

/usr/bin/python3 /home/pi/Project/pymon.py

The script is a .sh script which is used in the autostart file from raspberry pi using the <terminal -e path, in our case the path is /home/pi/myscript.sh.

4.2 Python Modules

Matplotlib Library: Used to generate plots

Matplotlib.pyplot module: provides a MATLAB-interface for our simple plotting

Pandas library: it provides high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

Seaborn library: is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Flask library: Flask is a micro web framework written in Python.

Adafruit\_DHT22 library: it provides a quick implementation of one-wire interface communication with the sensor.

4.3 Web Application

Flask is used to serve a web application on the raspberry pi’s ip. We enter on the website with <ip:8000> because the socket is configured on port 8000.

We have index.html file which contains the auto refresh.

The GET request from the browser starts the App. Every subsequent GET plots new data.

5. Conclusion and further work

The current version of the project supports the following functionalities:

* reliable reading of the temperature and humidity
* current performance and measurements are depicted into a live plot using auto-refresh

List of possible extensions:

* any sensor added can be graphed to the application
* comparing two devices connected by storing data into a database

6. References

1. <https://www.electronicwings.com/raspberry-pi/dht11-interfacing-with-raspberry-pi?fbclid=IwAR0S537GNf4cMMPl1RDmjgVVX36oMcDDDz0JnS95S2kTsR9cd9CqiO3S5i8>

2. https://raspberrypi.stackexchange.com/questions/8734/execute-script-on-start-up