Phelipe Fernandes Arantes de Avila

Anthony Giordano

Project description: This project will consist in a system that monitors the temperature of water in 3 different depths. The system will be controlled by a Raspberry Pi model 2 B and will use a digital temperature sensor DS18B20. A DC Motor will move the sensor up and down to get the temperature in different depths. When web connection is established, the sensor Pi will connect to a hardwired server Pi to sync the databases, the server functions as an HTTP web hosting server that will display the data in an easily readable manner and update on a set interval.

2. Project goals (overall) + 3. Your project implementation plans.

1 - Develop a code to get the temperature from the sensor

-Python script using RPi libraries

2 - Develop a code to move the sensor

-Python Script

3 - Create a database file to save the measured temperatures

-SQLite3 Database

4 - Adjust timers and delays

-Intervals to be determined, scripting done

5 - Assemble the circuit

-Diagram below, waiting on some parts as of this progress report.

6 - Improve the system

-Potentially using solar power in addition to local power source

7 - Get the temperatures from the database

-Possibly implemented with Flask or jQuery, need to see which of the two will play nicely with the CanvasJS web graph

8 - Decide on an enclosure for the system

-Maybe an open source 3D printer file

9 – Create a visually appealing webpage for hosting the data

-Bootstrap framework used to ensure an all encapsulating web page for mobile and desktop devices, done in HTML, CSS and Javascript

10 – Create an interactive graph web application that will pull data from the database

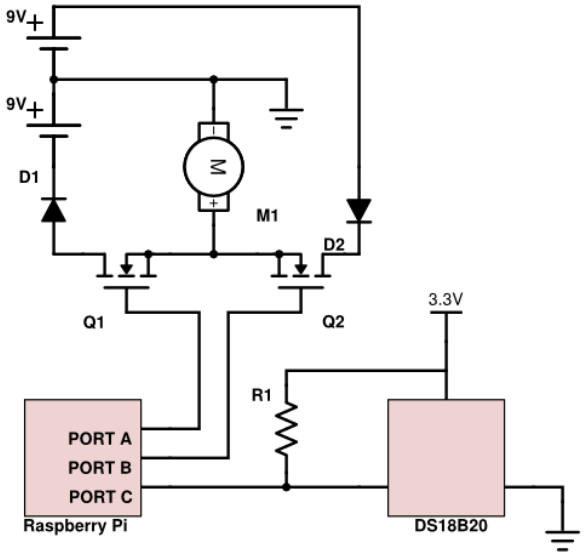
-CanvasJS used for the graph, but considering switching to Chart.js for built in SQL API

11 – Allow the server Pi to handle HTTP requests

-Flask, seemingly preferred over PHP for keeping the load on the Pi light

4. How you broke up the project into components – describe each component

Sensor side:
Components:



Raspberry pi: controls the motor, the sensor, save the temperatures, host the web interface
DS18B20: measure the temperatures
Q1: activates the motor clockwise
Q2: activates the motor counterclockwise
D1 and D2: protect the battery from reverse current

M1: moves the temperature sensor

Server Side:
-Flask (Python) for HTTP Requests (Also used for sensor -> server communication)
-HTML, CSS, JavaScript for webpage front-end
-JQuery or Ajax for graph
-SQLite3 for database

5. For each component above, describe how much work you have done, and what remains to be done. Include hardware and software used.

Sensor side:

By today (04/03), I am still waiting for the components, that were bought online. The only component that I have with me is the temperature sensor.

Server side:

Web page front-end is done and graph (Currently done with JCanvas) is able to be displayed but not yet pulling data from database. Considering switching to Chart.js for the graph as it seems more complete and has a built in SQL API. Flask is a work in progress. Long-term goal is Ajax to refresh the graph at a set interval without having to reload the page.

6. Summarize your project's “good” and “bad” aspects, “easy” and “tough” aspects, from the group's point of view

Sensor side:

This project is not difficult to build. The hardest part probably is going to be building the enclosure. The solution adopted to move the sensor is not the best solution, however, due to the short deadline and price of the components, we decided to use this solution.

Server side:

The web-page front end was not difficult, and the Javascript graph was actually remarkably easy and intuitive to implement (When downloading JSCanvas, one of the developers emailed me to ask if I was able to implement it in my project and to see if I needed assistance) but despite their kindness I am now leaning towards Chart.js for the final draft of the graph interface. Using Flask to handle HTTP requests for the HTML document seems straightforward, using it to sync data between sensor and server may prove challenging.

7. Describe your goals for the next 3-4 weeks

Have a working web-page that can pull data from a database and display it as well as a server that can handle all HTTP requests. Have a floatation method and power source, as well as an enclosure.

8. List group members, and each member's contributions.

Phelipe Fernandes Arantes de Avila: All sensor side + Database input

Anthony Giordano: All server side + Database data retrieval