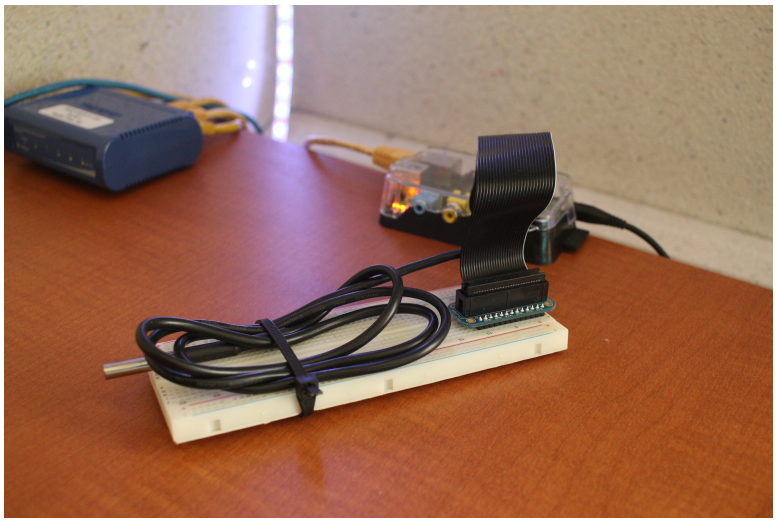
Temperature Data Aggregation in Bliss Hall at SUNY New Paltz

Embedded Linux (CPS342)

April 25, 2016



Brendan Lowe
Cesar Done
Heidi Fritz
Jabari Dash
Roberto Milanese
Victoria Bottali

Contents

1 Introduction

This report documents the process that several students in Dr. Chirakkal Easwaran's Spring 2016 Embedded Linux class took to complete their final class project. The course is designed to introduce students to the fundamentals of Linux programming - particularly embedded Linux. Dr. Easwaran's course is oriented around the Raspberry Pi and the Raspbian (Debian) Linux distribution to give students this fundamental practice. Students throughout the course learn basic terminal commands, how to read temperature sensors, how write to a SQLite3 database, and more. Midway through the semester, students are separated into groups, and assigned a final project. Students Brendan Lowe, Cesar Done, Heidi Fritz, Jabari Dash, Roberto Milanese, and Victoria Bottali were tasked with collecting temperature data throughout Bliss Hall at the SUNY New Paltz campus and representing it graphically for later analysis by the Sustainability Office.

2 Individual Student Responsibilities

Brendan handled server-side programming, networking, deployment

Cesar handled front-end programming, GUI design

Heidi handled client-side programming, front-end programming

Jabari handled documentation, logistics, GUI design

Roberto handled front-end programming, GUI design

Victoria handled client-side programming

3 Project Goals

3.1 Student Learning Outcome Goals

For this project students will be provided with first hand experience with software engineering. Being a Software Engineer / Developer is different than simply being a Coder or Programmer. A Programmer is someone who knows a set of programming languages, and knows how to write programs as assigned in those languages - the same for a coder. A Software Developer differs in that they analyze a problem, gather a team, and design and implement a solution. The goal of this project is just that. The task of this project is to familiarize students with processes and resources that Software Engineers use when creating solutions.

For this given project, students will use resources such as GitHub as a repository for their code. Using GitHub will give students the opportunity to work collaboratively, stage, and commit code. They will write software using Python, JavaScript, PHP, SQLite3, MySQL, HTML, and more. They will also obtain some of the soft skills required such as team work, communication, and basic negotiation skills (in terms of decision making).

3.2 Project Output Goals

To have several weeks worth of temperature data on Bliss Hall accessible in a user friendly web page that the Sustainability Officer will be able to view and later analyze.

6 Raspberry Pis at the pre-deployment stage

4 Materials & Programming Languages

4.1 Hardware

- Raspberry Pi Single Board Computer (8)
- Adafruit Raspberry Pi Enclosure(8)
- MicroUSB + Wall Adapter Kit (8)
- 4GB+ SD Card (8)
- Cobbler Cable + Pi Header Kit (8)
- DS18B20 Digital Temperature Sensor (8)
- Breadboards (8)
- 4.7k Ohm Resistor (8)
- Ethernet Cables (9)
- Networking Switch (1)
- Wire Tie (8)
- Duct Tape

4.2 Software

4.2.1 Client-Side Programming

- Python
- SQLite3

4.2.2 Server-Side Programming

- JavaScript
- PHP
- HTML
- MySQL

4.2.3 Other

- Crontab
- GitHub
- LATEX

5 Implementation

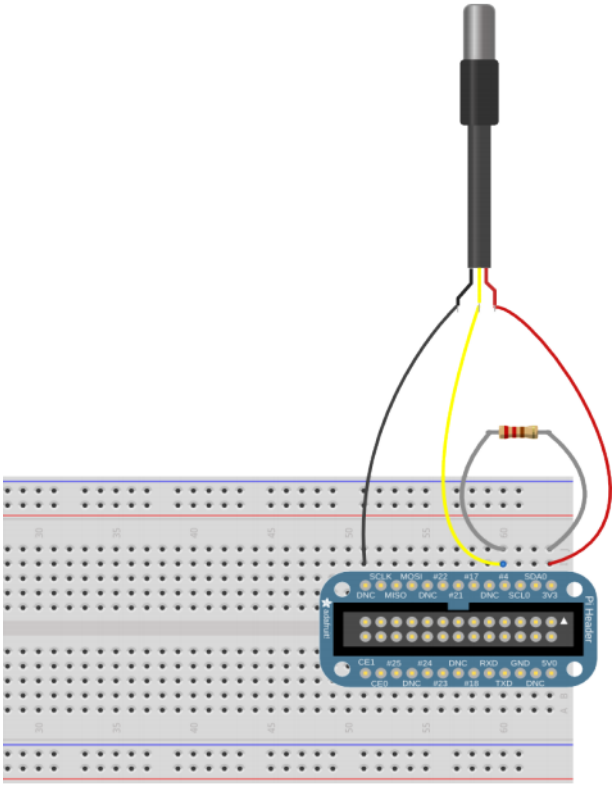
5.1 Overall Structure

Students distributed several internet connected Raspberry Pi computers throughout Bliss Hall in residents' rooms where each Pi periodically reads the current ambient temperature in the room and writes it to a local database. A Linux Apache MySQL and PHP (LAMP) server running on SUNY New Paltz servers periodically performs a pull request on each Pi, and compiles all of the temperature data in a MySQL database.

Google Maps screen-shot of rear of Bliss that demonstrates Raspberry Pi / Sensor distribution

5.2 Temperature Sensor Setup

The temperature sensor is connected to the Raspberry Pi's General Purpose Input Output (GPIO) pins via breadboard, and the 24-pin cobbler cable. The DS18B20 has three cables: ground (GND), 3.0 volt - 5.0 volt power line, and a data line. We connect the GND to and GND pin on the Raspberry Pi, the power line to the 3.3v rail, and the data line to GPIO #4. We then set up a pull-up resistor that connects the data line (pin 4) to the 3.3v line.



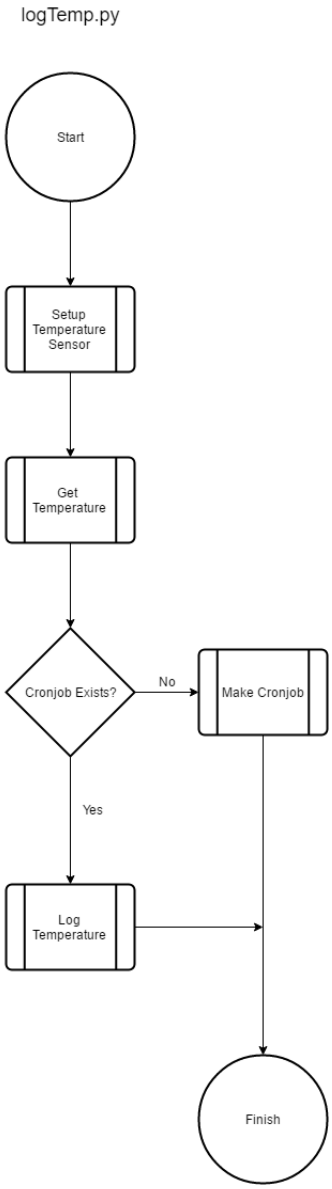
DS18B20 Digital Temperature Sensor connected to the breadboard using a 4.7k pull-up resistor

5.3 Client-Side Software (Raspberry Pi)

Each Pi will have 3 Python scripts that allow the Pi to collect data, convert it to JavaScript Object Notation (JSON) format, and return the JSON to the LAMP server.

5.3.1 Python Scripts

- tempLog.py: Creates a Cronjob (scheduled task) the first time tempLog.py executes. The Cronjob executes tempLog.py every 10 minutes. The script reads the ambient temperature in celcius and writes it to a SQLite3 database file called climate_info.db



- index.py: When passed a given start and stop date, index.py sets up a Flask server on the Pi and converts the data from the SQLite database into JSON objects. The JSON is put onto the Flask server in preparation to be sent to the LAMP server.
- json_push.py: Creates the JSON post to push to data to the LAMP server.

5.3.2 SQLite3 Database Schema

5.4 Server-Side Software (LAMP Server)

INSERT OVERVIEW HERE

5.4.1 PHP Scripts

5.4.2 MySQL Database Schema

5.4.3 Graphical User Interface

6 Student Analysis

6.1 As of April 25, 2016

- Brendan Lowe:
- Cesar Done: The project thus far has been running very smoothly. We are lucky to have access to several rooms here on campus as well as several network settings that students dont regularly have. The Pis havent given us any problems and the actual set up and maintenance of them have been smooth as well. This project can become a even bigger, funded project by the school if we manage to provide accurate as well as beneficial data to the school. Also if we can create a simple interface that school can use to access the data, they would be more inclined to support and provide their services for the project.
- Heidi Fritz: This project had simple tasks that we planned out and I took on the back end operations that the raspberry pi would carry out to collect the temperature and humidity data. To do this I connected my own temperature sensor to a personal pi and wrote python script which consisted of four functions. A function to get the temperature from the device, log the temperature and time into a sqlite database, create a cronjob to repeat data collection, and a main function to start the process. One struggle I came across was opening the device file so it would work on every pi. I imported glob to find the correct path name within the pi. The python script was easily integrated with the other scripts to work on one pi so we could make copies of the sd card. In the end, I met the team goals and expectations.
- Jabari Dash: The project so far is going well in that the data is being collected correctly. The beginning was challenging in that the group was lost because we were unsure of how to implement this project - particularly the intercommunication between the Raspberry Pis and the server. We wanted to used static IP addresses on the school network, but this was not permitted. Fortunately, Brendan's skill in networking, and more importantly, his position as the Networking Manager allowed us to have a subnet on the school network, with static IP addresses. This was a "quick and dirty" solution that allowed us to focus more on other elements of the projects, but also reduces the scalability and portability of our implementation.
- Roberto Milanese:
- Victoria Bottali: My goal for the end of the project is to have a working Flask app that is able to graph the temperature from the back-end based on different sets of parameters passed via user input. More specifically, I would like the app to be able to dynamically plot data given a broad spectrum of options,including different periods of time and whether or not the

user would like to see multiple data sets plotted together for comparison. I think the project turned out well, though we needed to rely on a a few "quick" fixes here and there in the interest of time.

6.2 As of May 1, 2016

- Brendan Lowe:
- Cesar Done:
- Heidi Fritz:
- Jabari Dash:
- Roberto Milanese:
- Victoria Bottali:

7 Conclusion