David Bisgrove, Steven Selementi, John Padoginog

May 11, 2017

Embedded Linux

Chirakkal Easwaran

# Project Report

For our project we were tasked with creating a device that, once plugged in would start collecting temperature and humidity data at regular intervals then store the data both locally and on a remote site of our design. The site, that we were to create, would require the user to login then it would display the temperature and humidity data of each deployed device. It would also let the logged in user adjust the time frame of the device such as how long the device runs for and how frequently it takes measurements.

Our strategy for completing this project was to split the overall project down into several goals and handle each of these goals one at a time as a group. While this strategy may have been tedious and redundant at times it proved to be useful due to everyone gaining a strong understanding of how the project functioned. While this strategy may not be practical in a work environment, where the goal is to get the project working properly as quickly as possible, it flourished in an educational environment, where the goal is to have everyone understand how all the different aspects of the project work together.

Goals:

1. The first goal was to get the raspberry pi to measure both temperature and humidity and store the data into a data table .We figured that this was the most important objective and wanted to get it done as early as possible.
2. Our second goal was to create an admin site that could display the data table created by the pi. The site would also let the user set the frequency of measurements along with the duration.
3. The last goal was to make the whole system “pretty”. This included working out bugs, making it easier to use, and etc.

Implementation/components:

For the first goal, we wrote a program that first connects to an SQL database and creates a data table called “temperatureTable”. It then asks the user for two inputs, “timeframe” and “timepulse”. Using these two inputs, the program proceeds to try and read data from a DHT22 sensor located at pin 4 for every occurrence of “timepulse”, immediately storing the data into “temperatureTable”. The program does this for a duration equal to “timeframe” before finally stopping.

For the second goal we created a basic website hosted on one of our wyvern servers. This is also where we started to run into a serious problem. After creating a basic website that should have been able to display our data table we ran into an error that we couldn’t solve. Every time we tried to open the site instead of getting our data table displayed we got an error message stating that root users permission was denied.

In an attempt to solve this we tried changing the user's permissions, changing the user's password, manually going into the the configuration file and editing it, and many other possible solutions we found on the internet but none of them seemed to have worked. In the end we were unable to get the website to connect to the sql database.

Important Note: to get as far as we did we had to install several other softwares such as sql, phpmyadmin, apache2, and the DHT 22 sensor library. The tutorials listed below include how to do that.

Summary:

Overall the project didn't work due to an error that we couldn't solve. Besides that we were able to create a data table stored in a sql database and store information into this data table with a program that used a sensor to measure temperature and humidity. We were also able to create a webserver and design a webpage for it that should have been able to display our data table, but due to a complex error it could not.

Each member of our group (Steven, John, and David) put in an equal amount of effort. This is due to us only working on the project as a group with the idea being that everyone would gain a strong understanding of how the project works.

John kept track of the components we were using and it was his raspberry pi we did most of the work on. He was also the primary programmer with heavy oversight from both Steven and David. We kept this setup because John was the only non-computer science major in the group and he wanted to practice his programming.

Steven and David acted as backup programmers and researchers. They would find solutions to more difficult problems that we came across and take over programming

every now and then to better implement these solutions. These situations could range from finding out how to set up a web server to finding out why a certain line of code isn't working as how it was supposed to.

Useful tutorials:

http://www.instructables.com/id/Turning-your-Raspberry-Pi-into-a-personal-web-serv/

http://www.instructables.com/id/Raspberry-PI-and-DHT22-temperature-and-humidity-lo/