**Networking Climate**

**Abstract**

Climate change is an integral part of our world and animals depend on this climate change for thriving. [5] My project will attempt to access weather data from openweathermap.org API and then feed this data to a raspberry pi. This raspberry pi will then have a temperature and humidity sensor attached along with a heating pad, which will be commanded to heat up or turn off according to weather data extracted. This project will most likely lay groundwork for a more in depth project dedicated imitating climates around the world. Success in imitating specific climates could allow and easier captivity maintenance of specific animals or meteorology research.

**Introduction**

My project was aimed to cultivate a regulation on climate control. Ideally this was a mechanism that would be used for tank-enclosed animals such as spiders, lizards, frogs, etc. I first proposed the environment of the rainforest by imposing the implementation rain through a water pump, which would have been very interesting. However my main components of concern were the temperature as this would be a very universal aspect of climate control. Along side the potential of the rain implementation the humidity would have shown to be of bigger importance in my project. Additionally all of this was going to be managed by a Raspberry Pi, a mini computer easily adaptable to many projects. Essentially the raspberry pi would gather the temperature data and accordingly activate another object to heat up (in this case a heating pad) to hopefully increase the temperature. Moreover, I would utilize an API to get access online weather data to hopefully ideally adjust the temperature according to this information. I will go into further detail of my project in the succeeding sections.

**Design**

My initial design for this project was to build the actual tank with the raspberry pi, sensors, water pump, and the lights all attached. However this did not happen due to time constraints and multiple issues throughout the implementation process. Thus my design for my project was constantly changing due to a balance between time management and implementation obstacles. I ended up discarding a lot of the original design plans like the water pump and the actual tank it’s self. I could not get my hands on a water pump in time and thus I was forced to leave it out of the project. However I did manage to get a heating pad (Electric Heating Pad) and the Humidity & Temperature sensor that became the focal point of my project. I also utilized LEDs as well however they were mostly used for testing purposes and due time I was unable to utilize the lights over Internet management. Ultimately, my project ended up being a mess attached to two breadboards. This of course was not the intentions but is what came out of the process of my first raspberry pi/electronic project.

**Implementation/Testing**

The implementation process was an unpredictable up and down roller coaster. I first began by booting up my very first raspberry pi and then taking a naïve approach to understanding electrical engineering. For weeks I was stuck understanding just how to manage electrical currents, what voltages specific components need, how to integrate the heating pad and the humidity/temperature sensor, and even simply utilizing the raspberry pi with different languages. [4] Probably the major issue that I had was allowing the heating pad access to 5volts from the raspberry pi. The GIPO pins on the pi only give off 3volts and thus from my beginner understanding, I needed a transistor to control the 5volts pin to the heating pad. [2] [3] However I was unable to get it to work in time. Once I got comfortable with all these concepts and had LEDs flashing, the heating pad heating, and the sensor sensing I began to look into the API service from openWeatherMap.org. [1] This website seems to hold the most popular weather API available and is no surprise as this website shares extensive API request functionality. I probably took about a week or less to figure out just how to use this API. I originally intended to access this API through a C++ program because the program managing the temperature sensor and the heating pad was written in C++. Although accessing API’s through C++ showed to be very cumbersome and I was starting to run out of time to I needed to figure out another way. Thus I decided to write a short python code that was very easy to access the API. In this program I request just the temperature of Corpus Christi (utilizing Corpus’s city code provided by openWeatherMap) and I write this temperature to an external text file simply called “file.txt”. The program running the sensor and the heating pad then reads from this file reading in the temperature written and retrieved by the python code. I understand this is a form of patching but it nonetheless worked. Also I would like to mention that in the C++ program there are two infinite loops, one for continuous sensing of the temperature and the other for grabbing the temperature retrieved from the python code. Additionally both infinite loops are running on their own separate threads allowing their iterations to execute at different speeds. This is vital, as you do not need to be fetching the weather data as fast as you’re sensing the surrounding temperature. After this stage in my project I began trying to turn on and off LEDs through the Internet via Apache2 with the possibility of using CGI files to implement C++ code and integrate the heating Pad and Sensor program with an Internet medium. However, again time constraints did not allow me to pursue this further.

**Future Work**

This project has a lot of improvements that can be built upon. Obviously finishing the ideal design with the actual tank, and integrating a water pump would be the first thing to accomplish. This is of course aside from some of the major bugs that need fixing like the electoral powering for the heating pad. Additionally, I think more heating pads should be involved, as this individual heating pad does not give off significant amount of heat. Stretching this idea further this can be utilized for watering plants or monitoring remote weather climates for whatever purposes.

**References**

[1] OpenWeatherMap. (2012). Retrieved from <https://openweathermap.org/api>

[2] Python. RPi.GPIO 0.6.3 (2016). Retrieved from <https://pypi.python.org/pypi/RPi.GPIO>

[3] Tutorialspoint. C++ Web Programming. (2016). Retrieved from <https://www.tutorialspoint.com/cplusplus/cpp_web_programming.htm>

[4] Wiring Pi. GPIO Interface library for the Raspberry Pi. (2012). Retrieved from <http://wiringpi.com>

[5] WWF Global. Wonder trees and plants on the world’s poorest soils. Retrieved from <http://wwf.panda.org/what_we_do/where_we_work/amazon/about_the_amazon/ecosystems_amazon/rainforests/>