Humidity Control System

CS 426 Final Project Report

Qicheng Xiong

qxiong1@binghamton.edu

Binghamton University

ABSTRACT

Extreme low indoor humidity is a huge problem in the Binghamton area during wintertime, especially to those who have dry skins. However, if you buy a humidifier, you need to turn on and off the humidifier yourself, which many people will forget easily.

What if there exists a humidifier that can adjust itself, based on its surrounding environment? This humidity control system turns on the humidifier when humidity reading is below a certain level defined by user, and it turns off the humidifier when humidity reading is above that level.

1 INTRODUCTION

Being automatically controlled, humidifier gains the ability to protect us from powering itself up when humidity is too low, and powering down when humidity is too high, since both extreme high and low humidity can harm our body.

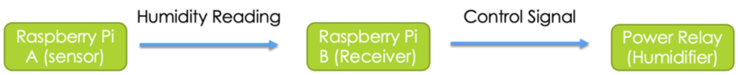
2 MOTIVATION

Every morning I wake up during winter, my throat is exhausted by the long night dryness, and sometimes my skin turns red on certain areas. It’s because the indoor humidity is too low. Along with other factors like the dry AC air and water absorbent wood furniture, humidity reaches record low in winter.

I do have a humidifier, but when it turns on for too long and the indoor humidity already turns up to 70%, it still won’t stop. I can’t wake up at night to turn it off, so I need to design a automated system to help me.

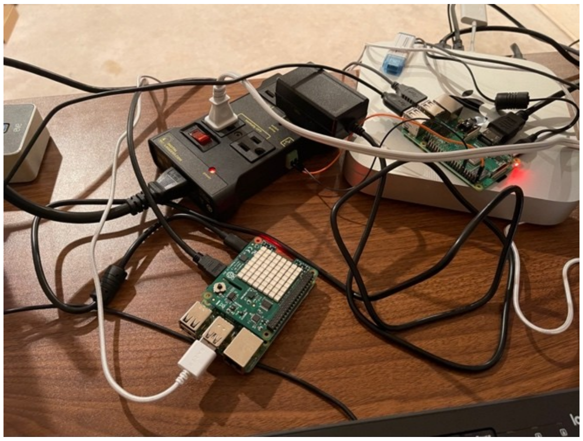
3 SYSTEM DESIGN

To build the system, I need two Raspberry Pi, one functioning as a sensor and sending humidity readings to another Pi, which acting as an actuator to control the humidifier.



4 IMPLEMENTATION

On the first Raspberry Pi (Sensor), I write a Python program to get the humidity reading from the SensorHat attached on top of it, and then write the reading onto a text file. This text file is sent to another Raspberry Pi (Actuator) for every 0.1 second repeatedly, with command SCP (for transferring file) and WATCH (for repetition).



One the second Raspberry Pi (Actuator), the text file sent from the first Pi will be scanned every 0.5 second to ensure the responsiveness. User can define a value, if the actual humidity reading is below that value, the second Pi will turn on the Power relay and therefore the humidifier connected to the relay will work. If humidity reading is higher than that value, humidifier will stop working.

5 CHALLENGES

There are many implementations that need improvement. For example, using SCP and WATCH method is not a perfect resolution for people who want an easy setup. It requires ssh and ssh-keygen, it needs to know the IP addresses of each devices. All of these procedures can be set up by an automated software with user-friendly GUI, but I have no time nor enough knowledge to do that.

6 HARDWARES

Hardware that are procured purely for this project: Honeywell humidifier, Raspberry Pi 3B+, Raspberry Pi 4, IoT Relay, several GPIO wires, SensorHat.

7 CONCLUSIONS

This humidity control system is designed to help you automatically turn on or off the humidifier, but only as long as you don’t bother filling water into the humidifier. I imagined the fully automated system before, which will even fill the water for you. But that would include the valve and water pressure issue. That’s too risky to run since water pipes has a really good chance to be frozen and explode in Binghamton area during winter.