Server Temperature and Humidity
Internet of Things (IoT) Project

Proof of Concept

January 2015

Arduino Yun microcontroller

Temperature Humidity Sensor

Purpose:

Our server room at the Lynnwood, Wa. Office was having issues with the humidifier that is installed in the ceiling. We wanted to know what the humidity and temperature was doing since there were many alarms going off from the AC unit dedicated to the server room. We were getting many low humidity level warnings over time. Finally, we decided to turn the humidifier off for the server room as it is not really necessary. Thus, we wanted to figure out what was happening to the temperature and humidity in the server room.

That's when this project came about.

What if there is a way to monitor the temperature and humidity in the room on continual basis and report that information back to a server to display in a chart on a web page….

How:

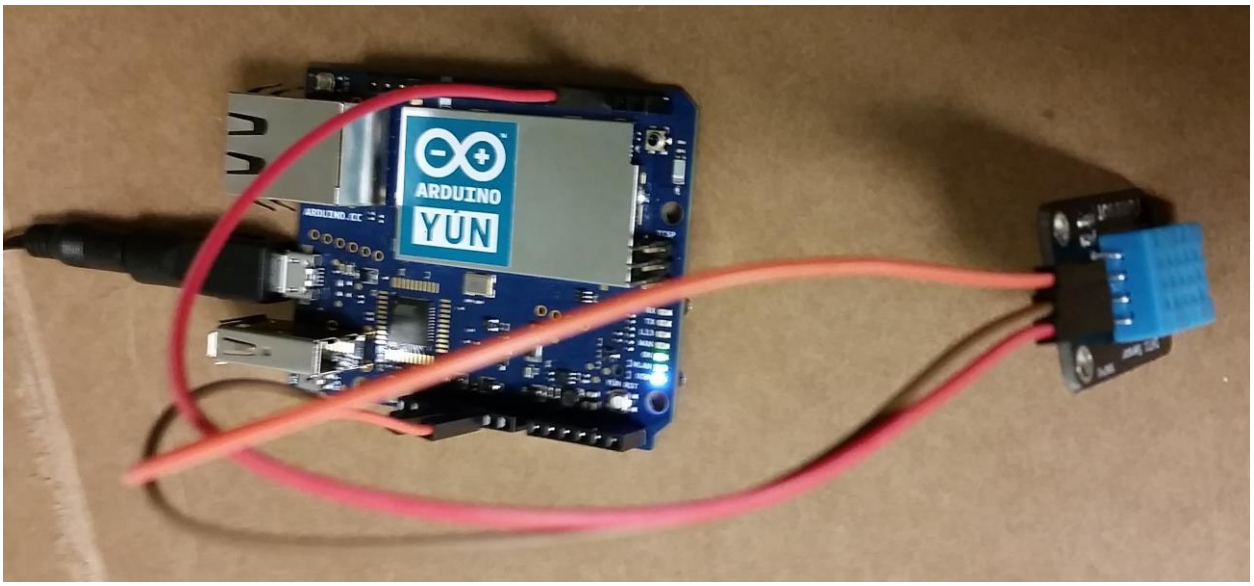
How does one do such a thing to sense information and push it to a web page for reporting?

One way is to put a sensor hooked up to a device that can talk to a web server to pull and push the temperature and humidity information to the web server. The web server can collect the information and store it. A web page can be designed to read the stored temperature and humidity information and display in a chart for reports.

What:

Enter the Arduino Yun microcontroller (pictured above) and a digital temperature and humidity sensor (pictured above). These devices can be connected to together to share information and then passed up

to the web server for storage and report. The Arduino Yun is a microcontroller device that interface with the temperature humidity sensor and send the information up to a web server for storage. The Arduino Yun is able to talk to a web server since it has its own web server embedded on the board. This microcontroller can be programmed to do what we need it to do. In this case is to read the data from the temperature humidity sensor and send that information up to a URL for storage and charting.



Here is the Arduino Yun connected to the temperature humidity sensor.

Pictured above is configuration setup that is put in the server room. This device is what senses the temperature and humidity in the server room and is connected to our WiFi. It then communicates to the web server sending the information up.

The Arduino Yun microcontroller has WiFi built in to its board. This device can also be connected via an Ethernet RJ45 cable.

The Arduino Yun microcontroller can be programmed to sample the temperature and humidity on any determined time interval. In this case every 10 minutes.

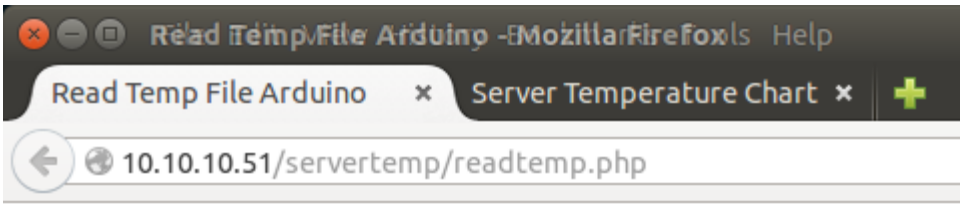
So, each ten minutes the microcontroller would poll the temperature and humidity from the sensor and send the information to the URL for the web server.

The web server would collect the data and store it in a file on the web server.

A web page could be accessed to show a chart of the values stored.

Results:

With that collected data it looks like this:



Temp Data Content

file found
DateTime,temp,hum,HeatIndex
2015/01/30 17:19:48,73.40,32.00,76.69
2015/01/30 17:19:58,73.40,32.00,76.69
2015/01/30 17:20:09,73.40,32.00,76.69
2015/01/30 17:20:19,73.40,32.00,76.69
2015/01/30 17:20:29,73.40,32.00,76.69
2015/01/30 17:20:40,73.40,32.00,76.69
2015/01/30 17:20:50,73.40,32.00,76.69
2015/01/30 17:21:01,73.40,32.00,76.69
2015/01/30 17:21:11,73.40,32.00,76.69
2015/01/30 17:21:22,73.40,32.00,76.69
2015/01/30 17:21:32,73.40,32.00,76.69
2015/01/30 17:21:43,73.40,32.00,76.69
2015/01/30 17:21:53,73.40,32.00,76.69
2015/01/30 17:22:04,73.40,32.00,76.69
2015/01/30 17:22:14,73.40,32.00,76.69
2015/01/30 17:22:25,73.40,32.00,76.69
2015/01/30 17:22:35,73.40,32.00,76.69
2015/01/30 17:22:45,73.40,32.00,76.69
2015/01/30 17:22:56,73.40,32.00,76.69
2015/01/30 17:23:06,73.40,32.00,76.69
2015/01/30 17:23:17,73.40,32.00,76.69
2015/01/30 17:23:27,73.40,32.00,76.69
2015/01/30 17:23:38,73.40,32.00,76.69
2015/01/30 17:23:48,73.40,32.00,76.69
2015/01/30 17:23:59,73.40,32.00,76.69
2015/01/30 17:24:09,73.40,32.00,76.69
2015/01/30 17:24:19,73.40,32.00,76.69
2015/01/30 17:24:30,73.40,32.00,76.69
2015/01/30 17:24:40,73.40,32.00,76.69
2015/01/30 17:24:51,73.40,32.00,76.69
2015/01/30 17:25:01,73.40,32.00,76.69
2015/01/30 17:25:12,73.40,32.00,76.69

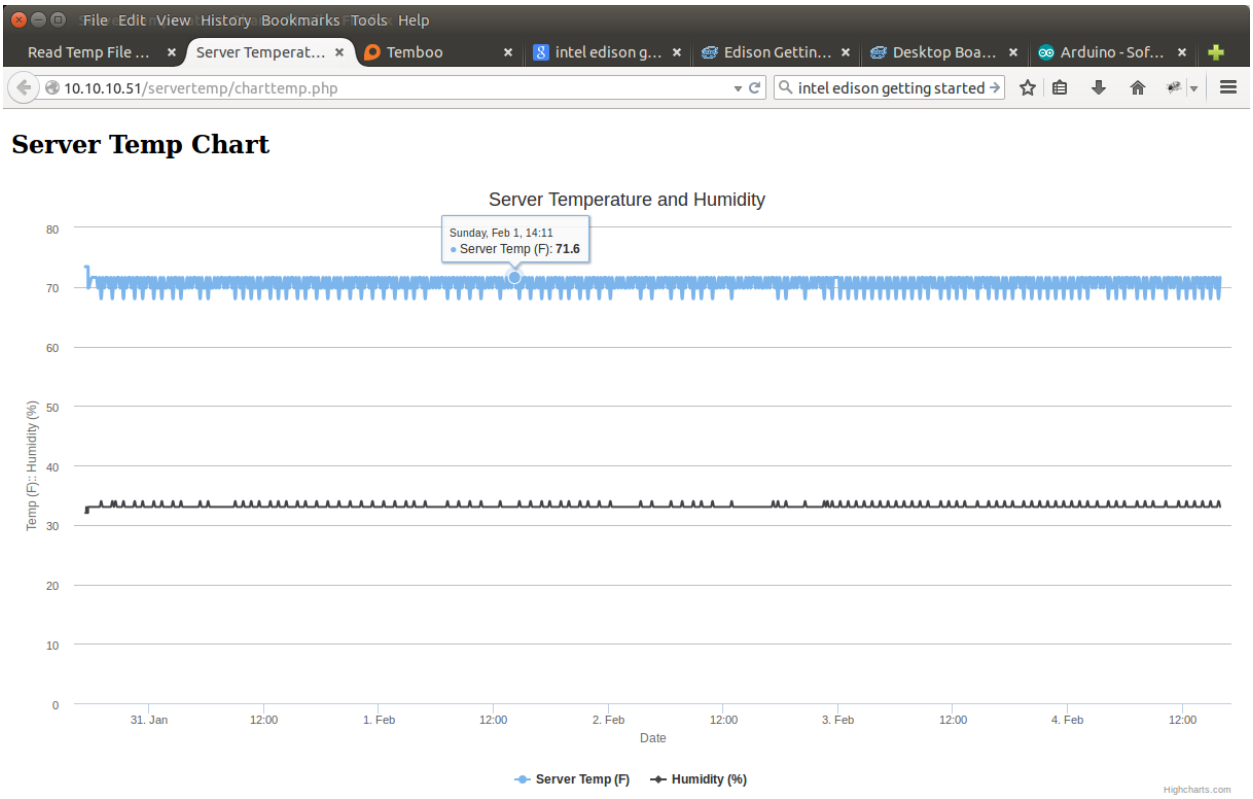
Temperature and Humidity data structure stored on web server (web page).

Data Format:

The data is composed of a set of comma separated values (CSV) in a file on the web server.

The data structure shown above is the date time stamp (UTC), temperature, humidity, and heat index. The head index is calculated using the temperature and humidity values. This Head Index is calculated on the Arduino Yun at sample time.

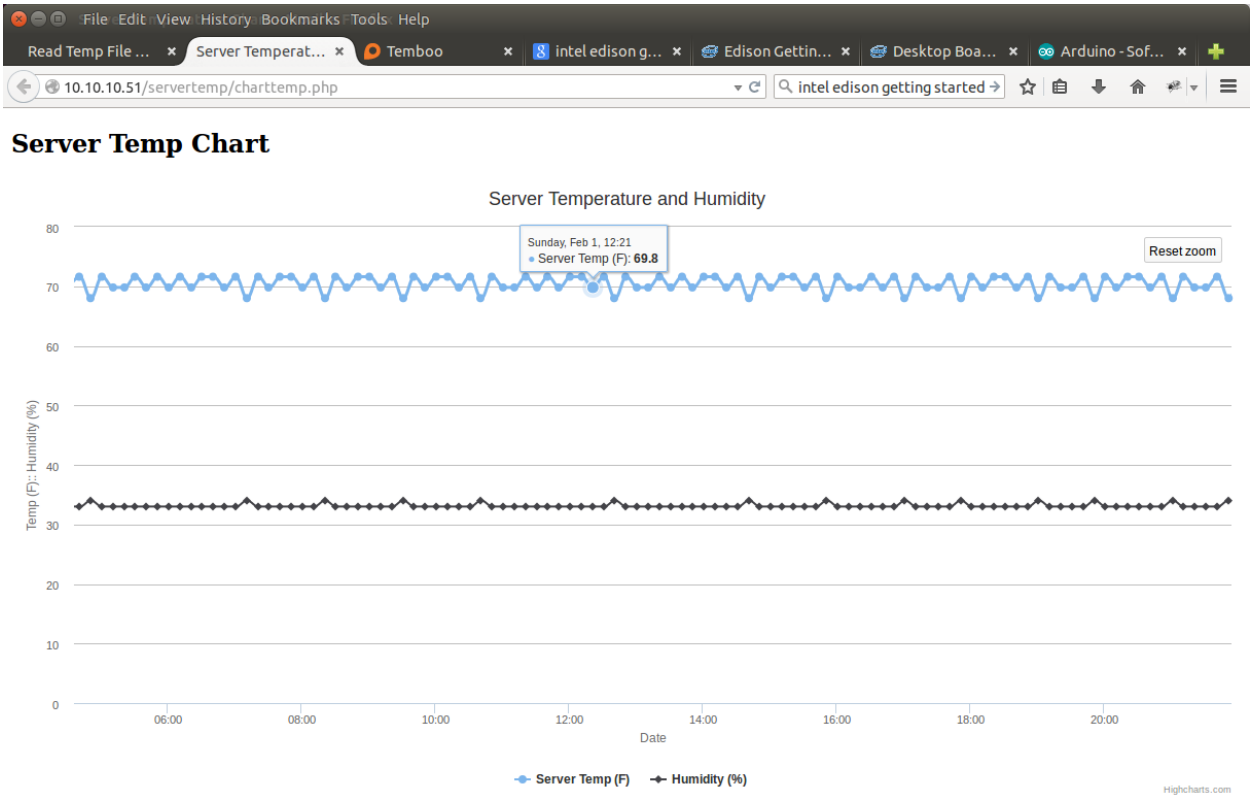
Charts:



Server Temperature and Humidity chart report (all days collected)

Above is the chart representing the temperature and humidity from the server room collected over a period of 5 days.

The chart has the capability to zoom into any period of time on the chart. An example follows:



Zoom in Temperature and Humidity chart.

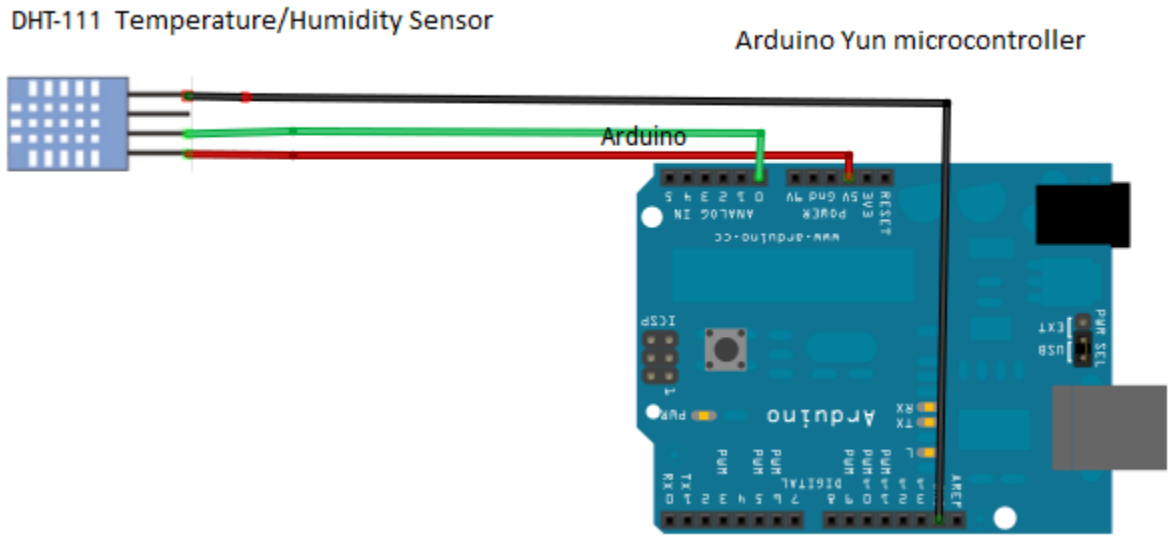
The blue line on top is the temperature value fluctuating in the server room. This is due to the servers running generating heat and the AC unit cooling down the room in a constant cycle. It is well behaved within tolerances.

The bottom line is the humidity line showing the humidity levels over this period of time. One can see that there is very little fluctuation in this humidity value. Due to this information turning off the humidifier was a wise decision as it was not necessary. Since the humidity stays rather constant in the room throughout time.

Summary:

The microcontroller allowed us to monitor and measure the temperature and humidity values over a period of time and make an educated decision on how to deal with the issues dealing with the server room. This information proved to be valuable to make a decision on whether to invest in repairing the humidifier or not. In this case it was a wise decision not to spend funds on repairing the unit.

Wiring Diagram:



Wiring diagram for Arduino Yun and DHT-11 sensor.

Above is the wiring diagram for the connecting the temperature/humidity sensor to the Arduino Yun. The data pin the sensor is to be connected with the Digital #2 pin for the Arduino code to work.

Server Detail:

This project was spawned at home to start out with connecting the Arduino Yun to the temperature and humidity sensor as well as connecting to a web server at home. This web server I decided to use a Linux distribution of Ubuntu 14.04 LTS. This decision was made that Ubuntu has been used a lot in my home and was easy to setup as a web server. It was also easy to create the web pages that collected the information form the Arduino Yun/sensor set and store it in a file on the server. The time stamp was stored at save time. Thus PHP was used on the server side to process the information on the web server. Another web page was created to read the data out of the temperature humidity file and translate it into the JSON code needed for the chart. The chart was downloaded of the net for free from Highcharts. Highcharts allows for single users (non-commercial) to use there charts free of charge. The fact that a Linux server was using in this project was for ease of use and setup. A Microsoft .NET web server project could be used just the same.

Paul Saletzki (February 4, 2015)