Monitoring Temperature and Relative Humidity (RH) of Server Rooms and Data Centers Using Arduino Board and DHT22 Sensor

Kirtikumar Solanki, IRS, ONGC and Naveen Kumar Matturthi, IRS, ONGC*

kirti_solanki@ongc.co.in

Keywords

Server Room, Data Center, Temperature Monitoring, Relative Humidity Monitoring, Arduino Board, DHT22

Summary

Monitoring the environment conditions in a computer room or data center is critical to ensuring uptime and system reliability. Server Rooms and Data Centre are often equipped with Air Conditioning system that does allow monitoring of temperature and relative humidity from a different location over the existing network. Moreover it does not allow generating alerts using a software code using application program interface. Present work describes how the monitoring of temperature and relative humidity over the network is achieved using a single-board microcontroller Arduino Uno R3 and a DHT22 temperature and relative humidity sensor. Objective is to achieve a complete functional system in terms of software and hardware; it includes programming Arduino board and writing interfacing software for reading and uploading data to a webserver.

Introduction

Maintaining and monitoring of desired temperature and humidity is an essential parameter for protecting digital assets such as servers, network switches, storage media like tapes etc. It is quite common that server and network switches are provided UPS backup, while air conditioning system for them are on raw power supplied by the utility company. Failure of raw power causes air conditioning unit to stop functioning. In this case servers and other equipment remain in running condition which leads to heating in data/server room. It may cause undesired effect on life of the equipment and data.

Data of temperatures and humidity are collected through the Arduino Uno R3 microcontroller board. Using an Arduino board has following advantages.

a) Open Hardware – Design of Arduino is open source and anyone can use it to make their own microcontroller board.
b) Programmability – It is easy to program this micro-controller board; Integrated Development Environment (IDE) is available for Windows, Mac OS and Linux platform as free open source software.
c) Cost – It is an easily available at very cheap and competitive price.

Theory and Method

Arduino Uno board is connected with a computer running on Linux platform through the USB cable. USB connection provides 5V power to the Arduino board and is also used for collection of temperature and humidity data. DHT22 sensor is connected with Arduino board. Data collected from the Arduino board is uploaded to the web server. Block diagram of the system is represented in the figure-1 below.

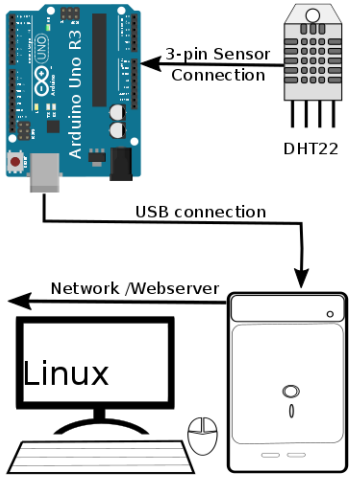

Figure 1: Block Diagram of the System

Monitoring Temperature and Relative Humidity of Server Rooms and Data Centers

Design of the System is divided in following parts

a) Programming of an Arduino Uno board using Arduino IDE for reading data from DHT22 sensor.
b) Writing interface software for Linux system to read the temperature and humidity data from the Arduino Uno board.
c) Creating REST (Representational state transfer) web services in the webserver for the temperature and humidity data.
d) Writing shell script and using curl tool for uploading data of temperature and humidity data at regular interval.

Arduino board is the central part in the block diagram of the system. Arduino Uno R3 development board is based on the ATmega328P which is 8-bit, low power microcontroller based on enhanced RISC architecture.

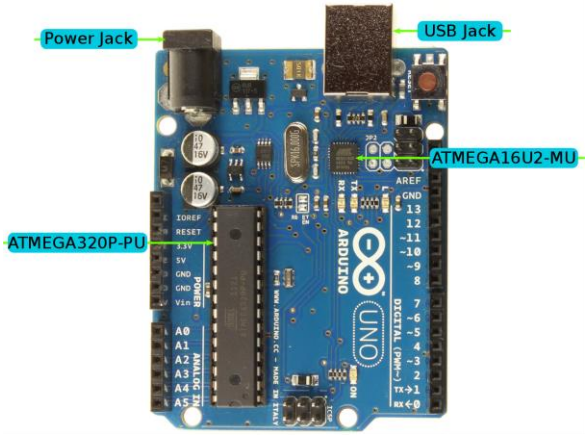

Figure 2: Arduino Uno R3 Pin-out

Arduino Uno R3 board can be powered through power jack or using a USB connection from a host machine which is the case here. It has 14 digital I/O pins (out of which 6 pins provide Pulse Width Modulation output) and 6 Analog input pins. An ATmega16U2 controller on the Arduino Uno board provides serial communication over USB and it appears as a virtual com port to interfacing software on the computer when connected. The ATmega16U2 firmware uses the standard USB COM driver, and no external driver files are needed for Linux and MacOS. However, on Windows platform driver files are required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board and it help uploading code to the Arduino board.

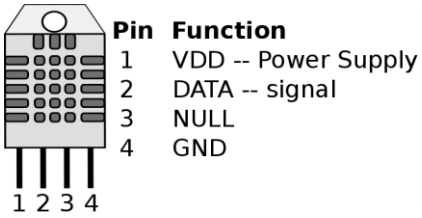

Figure 3: Pin-out of DHT22 sensor

DHT22 is a digital temperature and relative humidity (RH) sensor. Choice of the digital sensor over analog sensor is made for higher accuracy. Figure-3 describes the pin-out of the DHT22 sensor. The working temperature of DHT22 is -40°C ... + 80°C and the humidity range is from 0-100%. It has temperature accuracy of 0.5° C and the humidity accuracy of 2%. Pin-2 of the sensor is connected to the pin-8 digital pin of the Arduino Uno board and sensor is powered through the Arduino board only. Sensor sends higher data bits firstly. It sends data in following format.

DATA=8 bit integral RH data+8 bit decimal RH data+8 bit integral Temperature data+8 bit decimal Temperature data+8 bit checksum

Data from the sensor can be read only after the delay of 2 seconds of the first reading. Code for sending data at every 5 second on the serial out of the Arduino board is written and uploaded on the board using the Arduino IDE software. Communication parameters used are 9600 baud rate, no parity and no hardware flow control. Software sensor library for DHT22 from adafruit, has been used to program the sensor.

When an Arduino board is connected through the USB cable to the Linux computer, it is automatically detected by the Linux computer. It can be seen using a ‘dmesg’ system command in the Linux. As seen in figure-4 Arduino board is detected as ‘/dev/ttyACM0’ virtual COM port.

Monitoring Temperature and Relative Humidity of Server Rooms and Data Centers

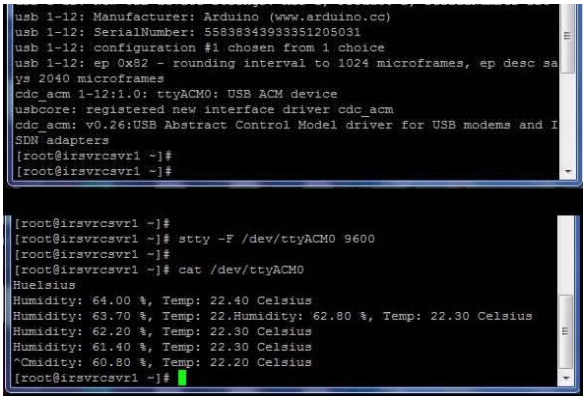

Figure 4: Detection of Arduino board and data reading from virtual COM port

It is possible to verify the working of Arduino board and sensor without writing any code using 'cat' command after setting baud rate of 9600 by 'stty' command. The same can also be verified from the Arduino IDE COM port monitor.

Capturing and Uploading of Data to webserver

Figure-5 describes the flow of the temperature and relative humidity data. Temperature and relative humidity data are captured on computer by a software code written using GNU gcc compiler. Two data length are read and sent to the standard output while errors are sent to the standard error. Reason for reading two data length is that data can start from any byte, if read asynchronously; we can have at least one complete reading of the data.

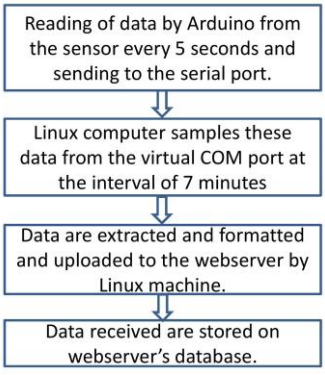

Figure 5: Data flow diagram of the system

Single data record will look like

$ Humidity: 58.60 %, Temp: 20.60 Celsius

While random read with two records may look like

$ .60 %, Temp: 20.60 Celsius Humidity: 58.60 %, Temp: 20.60 Celsius Humidity: 58

Data of two records are stored in a shell variable, where it is easy to get one complete record. Using a shell script's regular expression and pattern matching features, data of temperature and relative humidity are extracted and fed to 'curl' command to send it to the webserver. If multiple data collectors are used than it can be identified by 'devie_id' in curl POST request. Following is the example of the 'curl' command.

$ curl -v -request POST http://10.203.xx.xx/oct/api/v1/iot/feed/data/ -d "data={temp:21,humidity:53.4}&device_id=1"

Using 'curl' we can use http command GET request to read the data stored on the webserver.

OctoberCMS is used over the apache webserver as a content management system (CMS) along with MySQL as database. REST (Representational state transfer) web services are created for storing of temperature and relative humidity data. Each http POST request stores the temperature and relative humidity data in the MySQL database with timestamp. Data stored on the database can further be used for analysis and charting.

Conclusion

Work presented here is deployed in the Virtual Reality Server room at IRS. All software and hardware components used are open source. It is possible to extend this work by adding system e-mail alerts/SMS alerts using GSM module. It is also possible to use wireless sensor with the help of LoRa (Long Range) wide area network module with Arduino board.

References

1. Getting Started with Arduino 3rd Edition, Maker Media Publishing, ISBN: 978-1-4493-6333-8, (2014)
2. https://www.arduino.cc
3. https://learn.adafruit.com/dht
4. Data Sheet of ATmega328P and DHT22

Acknowledgements

The authors are indebted to Shri O N Gyani, GGM-Head IRS, ONGC and Shri K Raghu, CGM(Prog), IRS, ONGC for providing conducive environment, guidance and support. This paper benefited from the fellow colleagues for providing the technical input for writing this paper. We thank ONGC management for allowing us to submit this paper in SPG-2020 conference.