International Journal of Control and Automation
Vol. 10, No. 2 (2017), pp.89-102
http//dx.doi.org/10.14257/ijca.2017.10.2.08

Design of Environmental Monitoring system for Auxiliary Data Center using Lower Hardware Cost

Lionel Nkenyereye and Jong-Wook Jang

Department of Computer Engineering, Dong-Eui University, 176 Eomgwangro, Busanjin-Gu, Busan, 614-714, Korea
lionelnk82@gmail.com, jwjang@deu.ac.kr

Abstract

A large fraction of the consumed energy is spent on data center cooling, which has motivated a large amount of work on temperature management in data centers. Thus, environmental downtime is a significant cost to organizations and makes them unable to do business because what happens in the data center affects everyone. Installation of physical Information Technology and facilities related to environment on monitoring temperature, humidity, power, flood, smoke, air flow, and room entry is the most proactive way to reduce the unnecessary costs of expensive hardware replacement or unplanned downtime. For this reason, it is recommendable to implement an environmental monitoring system to monitor the temperature in data center with the option of sending an alert to the supervisors wherever they are. In this paper, we present environmental monitoring system for auxiliary data center implemented using open-source hardware platforms; Arduino, Raspberry Pi, and the Gobetwino. Wherein, Gobetwino transfers sensor data to the nearest host server in the same datacenter such as temperature and distance every time an object hits some object around the datacenter or a person comes in to the entrance. The objective of collecting temperature and humidity data allows monitoring server's health and gets alerts if things start to go wrong. When the temperature hits 50oC, the supervisors at remote headquarters would get a SMS, and then they would take appropriate actions that make sense to reduce electrical costs and preserve functionality of servers in auxiliary data centers.

Keywords: *temperature environmental monitoring; arduino; raspberry Pi; Gobetwino; auxiliary data center, secure copy, notification, system design,*

1. Introduction

There is a large number of prior efforts in building data center and equipped an organization with servers that are used to facilitate their employees to carry out their daily activity, store information, and data warehouse for reporting. As auxiliary data centers are located far from the primary data center, their computation tasks contribute on good functioning of the organization and monitoring them could prevent unexpected downtime of the primary data center.

Auxiliary data centers hold scalable Information and Technology (IT) equipment-servers to assure business continuity in many companies which operate in distributed manner for the reason of increasing quality-of-service of the system perceived by end-users. The amount of electrical energy consumed by data centers however, increases with the amount of computing power. In the same line, as the workload of computing grows in size, it seems that day after day a failure situation is close to happen, then increasing the energy waste even more and affecting the business continuity. Servers produce a lot of heat, figures vary but a rough figure for a large server is around 2500 British Thermal Unit (BTU) [1]. Normally, available ventilation is good enough to maintain a safe and reliable

ISSN: 2005-4297 IJCA

environment, but of that fails then overheating can be a problem. The overheating impresses IT heads to think what they should monitor.

A large fraction of the consumed energy is spent on data center cooling, which has motivated a large amount of work on temperature management in data centers. Many of the supervisors of data centers do not understand the aspect of temperature management. One of the important things is the controlling of the setpoint temperature at which to run a data center's cooling system [2]. For this reason, it is recommendable to implement an environmental monitoring system to monitor the temperature in the data center with the option of sending an alert to the supervisors to them wherever they are. This is done well not only for primary data center but also for auxiliary data that contributes to the great satisfaction of customers and companies which have opted for business continuity strategy in case of a disaster that might affect information and technology equipment.

Design and implementation of an environment monitoring and reporting strategy for metering the energy use and temperature of various area of IT equipment-server is essential for efficient maintenance, energy efficiency and planning further change. Therefore, installation of physical IT(Information Technology) and facilities environment monitors based on monitoring temperature, humidity, power, flood, smoke, air flow, room entry and other conditions is the most proactive way to reduce the unnecessary costs of expensive hardware replacement or unplanned downtime.

In this paper, we present datacenter environmental monitoring developed using open-source hardware platforms, Arduino [3], Raspberry Pi [4], and the Gobetwino [5]. Gobetwino is used to store sensor data to the nearest host server in the same datacenter such as temperature and distance every time an object hits some object around the datacenter or a person approaches the entrance. Such a design has the advantages of low cost, easy to build, and easy to maintain. The major obstacle is that Gobetwino is running on Windows only. Raspberry Pi lies in its reliability, flexibility, and scalability.

Approaches that have been investigated include, for example, methods to minimize air flow inefficiencies[6], load balancing and the incorporation of temperature awareness into workload placement in data centers [7], and power reduction features in individual servers[8,9]. Data centers typically operate in a temperature range between 20^{o}C and 220C, and some are as cold as 13^{0}C degree [10]. Due to lack of scientific data, these values are often chosen based on equipment manufacturers 'suggestions'. A regular monitoring of temperature and humidity is straightforward because new equipment such as servers can easily cope with energy efficiency by increasing the temperature and decreasing the minimum relative humidity in data centers. The advantage of the proposed environmental monitoring is a straightforward strategy to temperature management. The monitoring information should help to investigate the performance of the IT equipment such as servers, then limiting negative effects on system reliability. While the performance of servers, storage, and memory is increased, the possibility of saving energy is the next step which enables carbon emissions. Assuming that, it would mean that air conditioning would only be needed less as before such as 12 days a year in data centers for instance [10].

The rest of paper is organized as follows. Section2 provides related work and essential background on monitoring data center environmental conditions system. We describe hardware and software requirements in section 3. In section 4, we present the system design of the proposed system, and demonstrate deployment of our system in Section 5. Finally, conclusion and future research direction are described in Section 6

2. Related Work and Technologies

We begin our study by reviewing the related work and the definition of the Gobetwino, generic proxy" for Arduino.

2.1. Existing Works about Monitoring Data Center

An important number of research in area of monitoring data center from environmental conditions focuses on the implementation of such monitoring system based on the Wireless Sensor Networks (WSN) [11-14]. The architecture of deploying WSN in data center consists of the sensor in front and behind every server [14]. With the help of TelosB mote running on TinyOS 2.1, the collection of temperature data is made possible. However, one of the challenges of using that kind of TelosB is the power consumption because it uses battery. The power consumption is not the unique challenge, the lower performance of wireless sensor network stands as a challenge because WSN topology changes dynamically most of the time.

Recently, the trends towards design made a transition from 8 bit to 32 bit (ARM Cortex M3) explore the next release of Arduino in the form of cheap practical computer (like Raspberry Pi) or a 64-bit processing [4]. The idea for considering Arduino as a single board for practical industrial applications leads us to implement the remote monitoring of data center environmental conditions using lower hardware cost with Raspberry Pi. Using Arduino, the challenges of energy consumption of sensor implemented in WSN for collecting data such as heat, humidity, airflow, smoke are solved.

2.2. Gobetwino: Definition and features

Gobetwino is kind of a “generic proxy” for Arduino. It’s a program running on a PC (Windows only), that will act on behalf of Arduino and do some of the things that arduino can’t do on its own. So it works like a go between, hence the name [5]. Gobetwino is listening on the serial port, for “commands” coming from Arduino, and in response it will do something for Arduino and possibly return something to Arduino.

Using the defined command types you can create commands in Gobetwino that arduino can ask Gobetwino to execute. These commands can [5]:

- Start a program on the PC.
- Start a program, and wait until it finishes, and tell Arduino it finished.
- Send data to any windows program from Arduino, like it was typed on the keyboard.
- Send email, optionally with an attached file.
- Download a file from the internet.
- Read a file and return data to Arduino.
- Log data from Arduino to a file, with an optional timestamp.
- Periodically check a POP3 mailbox for incoming mails and send commands from the mail to Arduino.
- Get the time from the PC.
- Get the date from the PC.
- Ping a host or IP address.
- Copy a file on the PC.

3. Hardware and Software requirements.

These requirements are required in order to implement the data center environmental monitoring System [15].

3.1. Hardware requirements

Hardware requirements are listed as follows:

- Arduino Uno: microcontroller board based on the ATmega 328P with 14 digital input/output pins, 6 analog inputs. It has a 16 MHZ quartz crystal, a USB connection, a power jack, an iCSP header and a reset button
- LED, along with a 470 ohms resistor
- Temperature sensor(at the Adafruit shop), the Analog Devices TMP36(-40 to 150C) : this sensor has a very wide range and does not require a negative voltage to read sub-zero temperatures
- Digital output temperature and humidity sensor DHT11 basic temperature-humidity sensor: it uses a capacitive sensor and a thermistor to measure the surroundings air, and spits out a digital signal. Good for 20-80% humidity readings with 5% accuracy
- Sharp GP2YoA21YKOF IR Range sensor-10 cm to 80 cm: interfacing to most microcontrollers is straightforward: the single analog output can be connected to an analog-to-digital converter for taking distance measurements
- Mini Photocell Light sensor: photocell is basically little light sensor, which behave like resistors but the value of its resistor varies with the light level. To integrate it in our system, we will also need another resistor of 10K ohms resistor
- Raspberry Pi 3 : it includes an ARMv8 900MHz processor, videoCore IV GPU, and 1.2 GHz of RAM

3.2. Software Requirements

- Software requirements for this projects are listed as follows:
- The Arduino Integrated Development Environment (or Arduino Software (IDE): contains a text editor for writing code. It connects to the Arduino hardware to upload programs and communicate with it.
- Raspbian : It is the official supported Raspberry Pi Operating System based on Debian
- LAMP (Linux Apache MySQL PHP): components (Apache 2 , php 5 , mysql-client , mysql-server , tomcat 6) required to run a dynamic HTML webpage and these are the first steps to build a LAMP webserver on a Raspberry Pi
- Gobetwino Arduino: a kind of a “generic proxy” for Arduino. It is a program running on a personal computer (windows only), that will act on the behalf of Arduino[4]
- Secure copy: securely transferring computer files between a local host and a remote host. It is based on the secure shell.

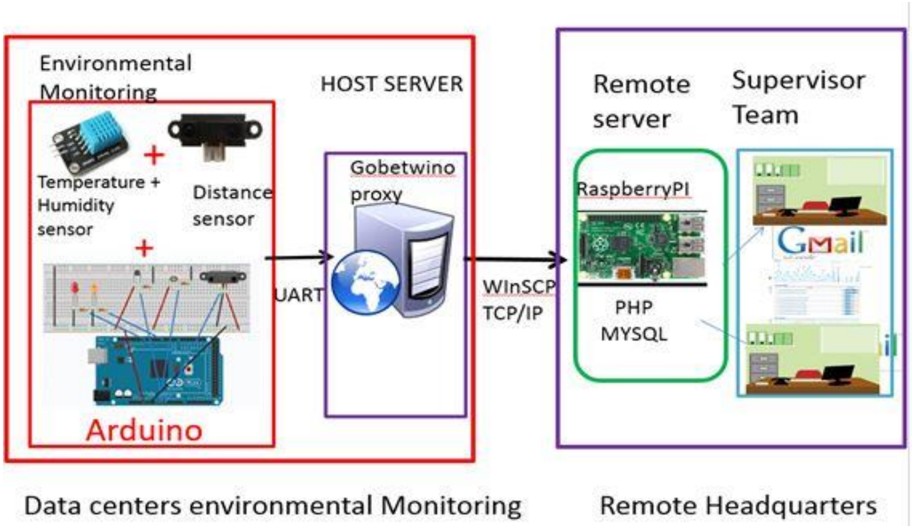


Figure 1. Overall System Architecture of the Environmental Monitoring system in Auxiliary data center with Arduino, Gobetwino and Raspberry PI.

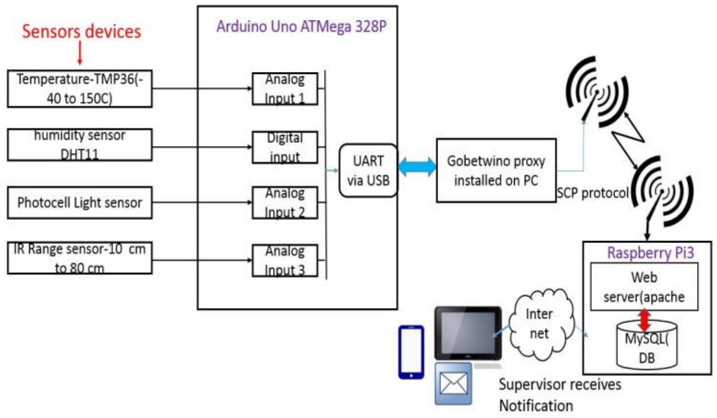


Figure 2. Block Diagram of the system

4. System Design

The system design is the process of transforming the user requirements and behavior into a purely programmatic oriented system by defining the architecture, components, modules, and data for a system in order to satisfy specified requirements [16]. The target system under implementation is settled into subsystems based on the analysis structure and the logical architecture.

4.1. System Architecture

Building an environmental monitoring requires development and integration of many hardware and software components. Figure 1 shows the overall system architecture of an environmental monitoring that we have developed. Gobetwino is listening on the serial port [5], for “commands” coming from Arduino, and in response it will do computation for arduino and possibly return computation to Arduino. WinSCP is an open source free SFTP client and FTP client for Windows. Legacy SCP protocol is also supported .Its main function is safe copying of files between a local and a remote computer. The host server that holds Gobetwino performs a transfer by copying the log sensor file to the remote server on raspberry Pi. The supervisor team should receive a notification email directly at primary working place.

Figure 2 represents the functional architecture of the system. Here, temperature-TMP 36, humidity sensor DHT11, photocell light sensor and infrared range sensor are the inputs for the system which will be used to generate the parameters which are the sensor data for the environmental monitoring system in the data center.

Gobetwino is kind of a “generic proxy” for Arduino. It’s a program running on a PC (Windows only), that will act on behalf of Arduino and do some of the things that Arduino can’t do on its own. So it works like a go between, hence the name [4]. Using the defined command types you can create commands in Gobetwino that Arduino can request Gobetwino to execute. These commands are used to start a program on the PC, start a program until it finishes, and send data to any windows program from Arduino, like it was typed on the keyboard. Gobetwino is used to send email, optionally with an attached file.

The implementation of the proposed environmental monitoring costs a lot less budget. A simple Arduino can offer more data for lot less money. For the budget conscious, you can get a cheap Arduino for less $ 5. To monitor the humidity, we use a DHT11 or DS18b20 [17] and a 4.7Ω resistor for $ 9.95. To read data from the sensor, we use an Arduino. The Arduino can be connected over a USB serial cable for the current value to be extracted and reported back. Gobetwino is installed on a simple computer with Windows as operating system to collect sensor data from Arduino connected to it.

The objective of collecting temperature and humidity data allows us to monitor the server’s health and gets alerts if things start to go wrong. When the temperature hits 50o C., the supervisor at remote headquarters would get a SMS, and then take appropriate actions that make sense to reduce electrical costs and preserve functionality of servers in data centers

4.2. System Flow Diagram

The readings of the sensors connected to the Arduino are programmed using the Arduino IDE. All Arduino sketches must have two functions: setup and loop. The setup function is executed only once when the Arduino board is first turned on, or when the reset button is pressed. The loop function is executed repeatedly (and indefinitely) after the setup function is finished. The process starts with initialization of inputs, outputs and configuration of header and footer part that the Gobetwino command uses to log readings data from sensors in csv form on the personal computer via Universal Asynchronous Receiver Transmitter (UART) serial communication. Global variables defined in the header are shared by all functions in the sketch. After initialization in setup function inputs and outputs of analog and digital corresponds to the sensors devices listed early in hardware requirements, the loop function (repeated continuously or body) execute the main tasks, waits one seconds and wait one more seconds.

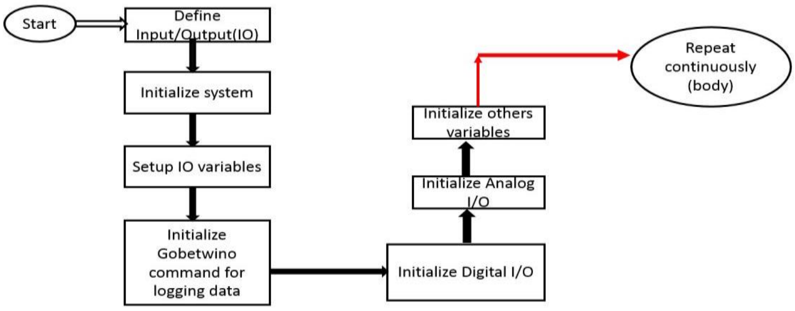


Figure 3. Flow Chart of the System

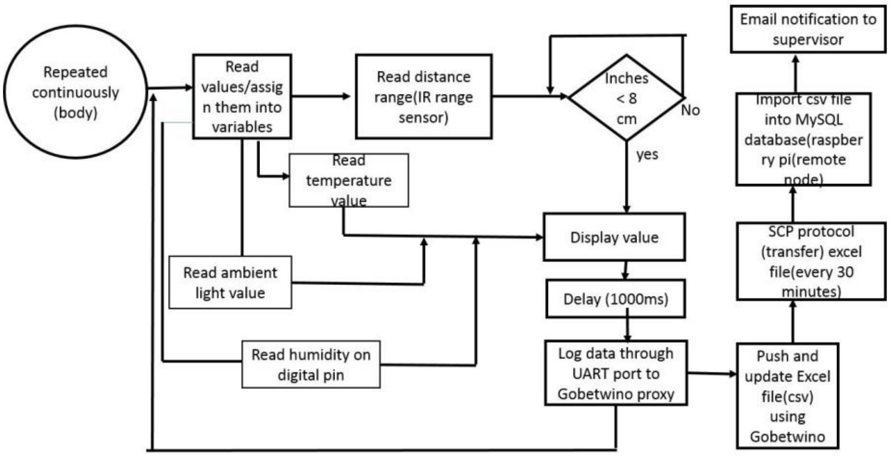

Repeated continuously (body)
Read values/assign them into variables
Read distance range(IR range sensor)
Inches < 8 cm
No
yes
Read temperature value
Read ambient light value
Read humidity on digital pin
Display value
Delay (1000ms)
Log data through UART port to Gobetwino proxy
Push and update Excel file(csv) using Gobetwino
SCP protocol (transfer) excel file(every 30 minutes)
Import csv file into MySQL database(raspberry pi(remote node)
Email notification to supervisor

Figure 4 .Flow Chart of the System (cont...)

Gobetwino is listening on the serial port [5], for "commands" coming from Arduino, and in response it performs something for Arduino and possibly sends command to Arduino. WinSCP is Secure Shell File Transfer Protocol (SFTP) client for Windows. Legacy Secure Copy (SCP) protocol is also supported [18]. Its main function is safe copying of files between a local and a remote computer. Figure 3 and 4 show the flow chart for implementing the environmental monitoring system. Building an environmental monitoring requires development and integration of many hardware and software components. In this paper, we develop a script that performs a transfer by copying the log sensor file to the remote server on Raspberry Pi. The supervisor team should receive a notification email directly at primary working place. We create one LGFIL type command with the name SENSORLOG that logs sensor readings from Arduino to a file

5. System Implementation

Implementation in this work denotes the realization of the application of the plan given in the design of the system architecture as designed on the Figure 5.

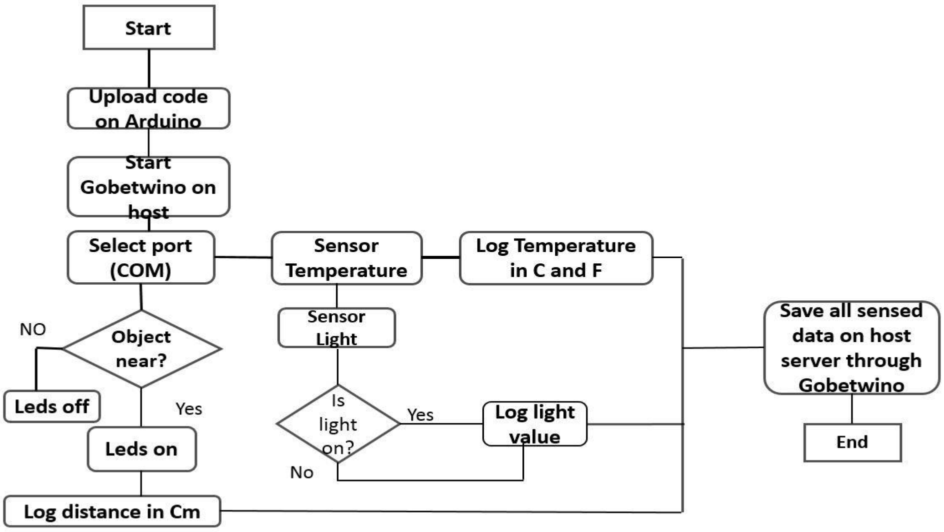


Figure 5. Flow Diagram of the System Implantation of the Temperature Management System in Auxiliary Data Center

The whole Implementation part of this project is divided into three different levels, those are:

- Design schematic circuit
- Configuration of Gobetwino
- Application development

5.1. Design Schematic Circuit

With the system definition in place and different sensors required, it is now time to start designing the schematic circuit using the Fritzing [19]. Fritzing is a free open-source design suite that works on Windows, Mac, and Linux. It has a simple and realistic interface that makes designing circuits intuitive. The idea is to allow us to just copy the breadboard design, using images of the used parts and then easily get to a Printed Circuit Board (PCB) design. Fritzing highly supports prototyping using the Arduino, actually the open hardware board is one of the first parts implemented. The Fritzing sketch on Figure 6 shows different sensors used to monitor temperature and light. The infrared proximity sensor is able to control the nearest object and performs some tasks according to the measurement of distance.

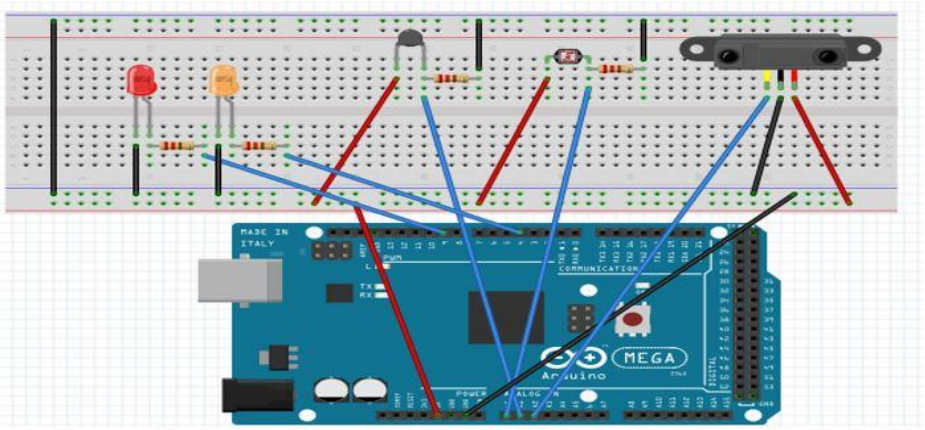
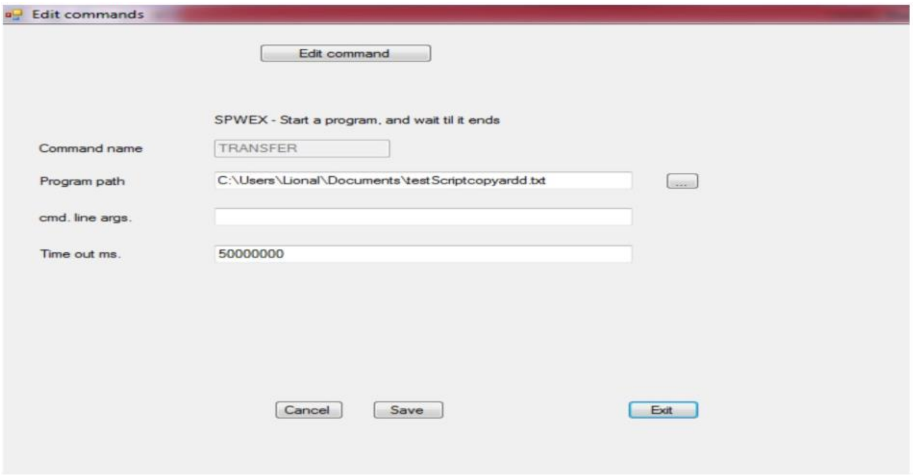


Figure 6 .Fritzing Sketch of the Design Circuit using Arduino as Microcontroller Board



SPWEX - Start a program, and wait til it ends

Command name: TRANSFER

Program path: C:\Users\Lional\Documents\testScriptcopyardd.txt

cmd. line args.:

Time out ms.: 50000000

Figure 7. Example of Gobetwino Command Allows Transferring Data from Arduino to Host Server

5.2. Configuration of Gobetwino

There are essential commands that should be programmed on Gobetwino in order to save the log monitoring on the host server. Three commands have been developed with Gobetwino. As stated earlier Gobetwino defines a set of command types that can be used as templates to create commands. Each command type has a name that is an abbreviation of its function. The difference between a command type and a command can best be explained with an example. Figure 7 shows the log to file command type in the project (TRANSFER) defines a command type that has parameters, program path, and timestamp. We create one LGFIL type command with the name SENSORLOG that logs sensor readings from Arduino to a file called “testlog.cvs.” So the command type defines an action and a set of parameters that each has a data type.

5.3. Application Development

We consider the following use-case scenario. The supervisor at the remote headquarters (primary data center) opts to receive an e-mail alert which includes an attached file of the sensor reading data at the auxiliary data centers. Sufficient temperature, humidity data are stored onto a database on the raspberry Pi on which LAMP (Linux Apache MySQL PHP) components are installed. We assume that the network is up and running. A web friendly portal is accessible by the supervisor either on his smartphone or browser on a desktop. This friendly portal plays a major role because it helps the supervisor to configure how and when an alert should be sent. The commands are stored on the database (installed on the raspberry Pi). It is through these commands of controls that will help to filter sensor data to be reported, the period time during of checking of abnormal situation at the auxiliary data center during a day.

The application code that will implement the functional requirements specified in the system flow diagram. The code must co-exist and log sensors data (temperature, humidity, ambient light value and distance range of an object) through Gobetwino installed on a host personal computer in the data center. To understand the code will need the understanding of how WinSCP operates and SSH (secure shell) protocol for operating data transfer over an unsecured network

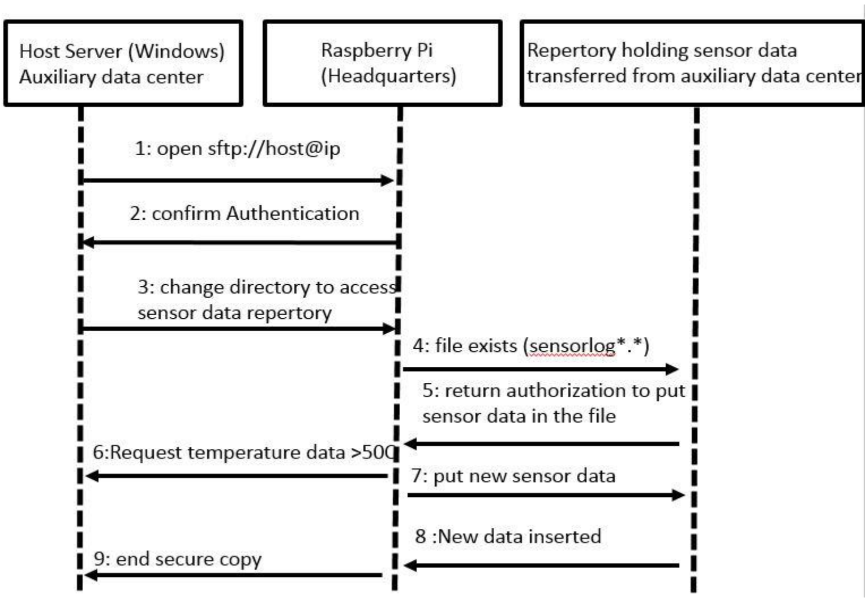

Host Server (Windows)
Auxiliary data center
Raspberry Pi
(Headquarters)
Repertory holding sensor data
transferred from auxiliary data center
1: open sftp://host@ip
2: confirm Authentication
3: change directory to access
sensor data repertory
4: file exists (sensorlog*.*)
5: return authorization to put
sensor data in the file
6:Request temperature data >50C
7: put new sensor data
8 :New data inserted
9: end secure copy

Figure 8. The Flow Execution of the Script Code Transferring Log Data from Host Server to a Raspberry PI Server at the Primary Data Center

5.3.1 Secure Transferring Log Data from Host Server to Remote Server Raspberry Pi

WinSCP is an open source free SFTP client and FTP client for Windows. Legacy SCP protocol is also supported [18]. Its main function is safe copying of files between a local and a remote computer. In this project, we develop a script that can set up as a task scheduled to run on a regular interval. The script performs a transfer by copying the log sensor file to the remote server on raspberry Pi. Figure 8 shows in detail the script code that transferring log data from host server to raspberry Pi where the intelligent controls must take place. For instance, controlling the limit temperature on which the supervisor of the data must be informed via an email notification.

5.3.2 Shell File to Load Data from Local Directory to MySQL Database

We design a database that will contain our static and dynamic data as follows: one table named “sensorupload “where we store the temperature logged classified by date in timestamp format. We write out a shell script that will be executed as a job every 5 minutes. The script consists to read the txt file from local path, loading the data into the table “sensorupload” after deleting the previous data inserted into the table [15].

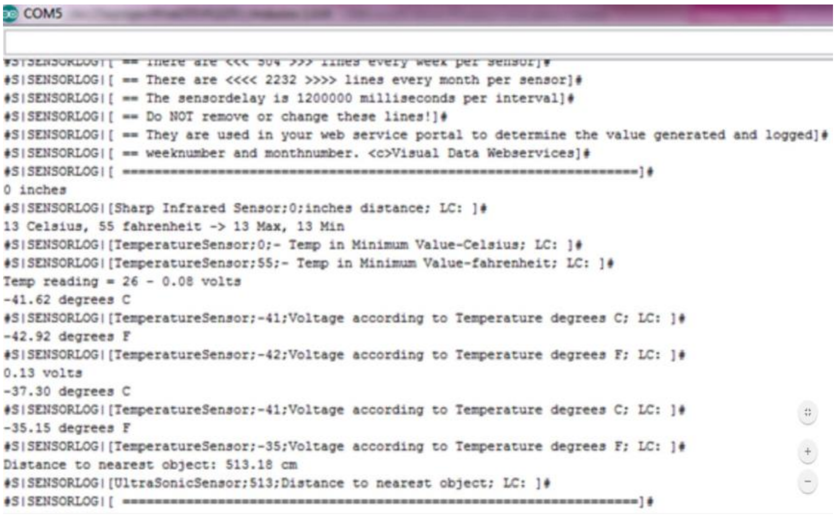
5.3.3 Design email Notification of Environmental Monitoring using Python

Python is a wonderful and powerful programming language that's easy to use and with Raspberry Pi lets you connect your project to the real world. The easiest introduction to Python is through IDLE. The detail of the script code consists of sending email within the attached log sensor data. This log data sensor data are transferring from the data center where the environmental monitoring system is deployed to remote server using secure tool WinSCP. Beside python script, we write out a shell script to execute automatically the script as job every 5 minutes using “crontab” [20]. When the temperature hits 50o C., the supervisor at remote headquarters would get a SMS, and then take appropriate actions that make sense to reduce electrical costs and preserve functionality of servers in auxiliary data centers.

5.4. Results of the Implementation of the Proposed Temperature Environmental Monitoring System for Auxiliary Data Center

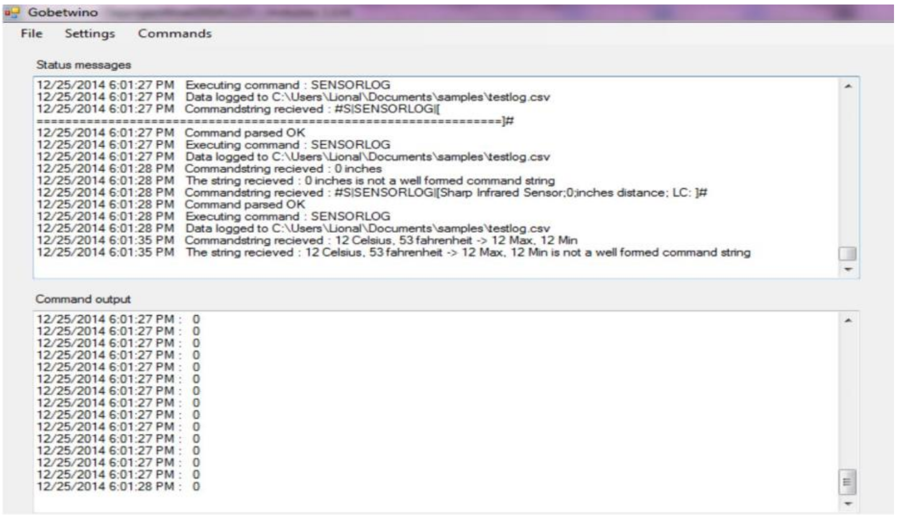
We have deployed the project at data center located at the Department of Computer Engineering for development and testing purposes. The deployment includes an Arduino board within temperature sensor, infrared, light sensor; and a host server on which we have installed Gobetwino software. The Arduino is connected to the host server and the port COM 5 is used to ensure communication between Arduino and host server through Gobetwino that acts on as a proxy. Figure 9 shows a real-time monitoring on Arduino board. The log data is saved on Host server in csv format. The overview of the file after being saved is on the specified path on the Gobetwino interface. On the host server, WinSCP command scheduled as a task is running. Figure 10 shows the log of Gobetwino commands executed on the Arduino when it is logging reading sensors data in csv format [15].

On the remote server which in this report is implemented on raspberry board, the LAMP (Linux Apache MySQL PHP) web service model. After job tasks uploading the csv file that holds monitoring data, the log data file is sent to e-mail of the supervisor team as shown on Figure 11.



```
#S|SENSORLOG|[ == There are <<< 504 >>> lines every week per sensor]#
#S|SENSORLOG|[ == There are <<<< 2232 >>>> lines every month per sensor]#
#S|SENSORLOG|[ == The sensordelay is 1200000 milliseconds per interval]#
#S|SENSORLOG|[ == Do NOT remove or change these lines!]#
#S|SENSORLOG|[ == They are used in your web service portal to determine the value generated and logged]#
#S|SENSORLOG|[ == weeknumber and monthnumber. <c>Visual Data Webservices]#
#S|SENSORLOG|[ ==============================================================]#
0 inches
#S|SENSORLOG|[Sharp Infrared Sensor;0;inches distance; LC: ]#
13 Celsius, 55 fahrenheit -> 13 Max, 13 Min
#S|SENSORLOG|[TemperatureSensor;0;- Temp in Minimum Value-Celsius; LC: ]#
#S|SENSORLOG|[TemperatureSensor;55;- Temp in Minimum Value-fahrenheit; LC: ]#
Temp reading = 26 - 0.08 volts
-41.62 degrees C
#S|SENSORLOG|[TemperatureSensor;-41;Voltage according to Temperature degrees C; LC: ]#
-42.92 degrees F
#S|SENSORLOG|[TemperatureSensor;-42;Voltage according to Temperature degrees F; LC: ]#
0.13 volts
-37.30 degrees C
#S|SENSORLOG|[TemperatureSensor;-41;Voltage according to Temperature degrees C; LC: ]#
-35.15 degrees F
#S|SENSORLOG|[TemperatureSensor;-35;Voltage according to Temperature degrees F; LC: ]#
Distance to nearest object: 513.18 cm
#S|SENSORLOG|[UltraSonicSensor;513;Distance to nearest object; LC: ]#
#S|SENSORLOG|[ ==============================================================]#
```

Figure 9. Sensor's Temperature Read from Arduino and Logged on the Computer which Hosts Gobetwino Application



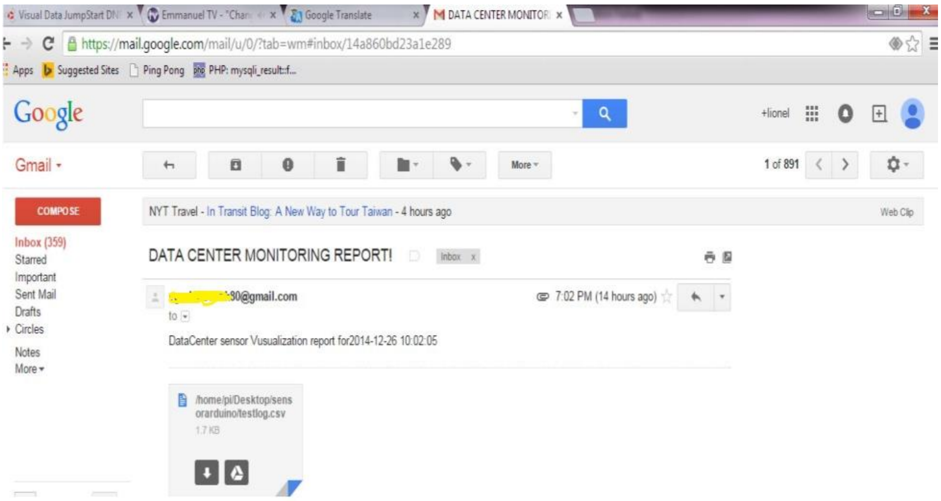
Status messages

```
12/25/2014 6:01:27 PM   Executing command : SENSORLOG
12/25/2014 6:01:27 PM   Data logged to C:\Users\Lional\Documents\samples\testlog.csv
12/25/2014 6:01:27 PM   Commandstring recieved : #S|SENSORLOG|[
==============================================================]#
12/25/2014 6:01:27 PM   Command parsed OK
12/25/2014 6:01:27 PM   Executing command : SENSORLOG
12/25/2014 6:01:27 PM   Data logged to C:\Users\Lional\Documents\samples\testlog.csv
12/25/2014 6:01:28 PM   Commandstring recieved : 0 inches
12/25/2014 6:01:28 PM   The string recieved : 0 inches is not a well formed command string
12/25/2014 6:01:28 PM   Commandstring recieved : #S|SENSORLOG|[Sharp Infrared Sensor;0;inches distance; LC: ]#
12/25/2014 6:01:28 PM   Command parsed OK
12/25/2014 6:01:28 PM   Executing command : SENSORLOG
12/25/2014 6:01:28 PM   Data logged to C:\Users\Lional\Documents\samples\testlog.csv
12/25/2014 6:01:35 PM   Commandstring recieved : 12 Celsius, 53 fahrenheit -> 12 Max, 12 Min
12/25/2014 6:01:35 PM   The string recieved : 12 Celsius, 53 fahrenheit -> 12 Max, 12 Min is not a well formed command string
```

Command output

```
12/25/2014 6:01:27 PM :   0
12/25/2014 6:01:27 PM :   0
12/25/2014 6:01:27 PM :   0
12/25/2014 6:01:27 PM :   0
12/25/2014 6:01:27 PM :   0
12/25/2014 6:01:27 PM :   0
12/25/2014 6:01:27 PM :   0
12/25/2014 6:01:27 PM :   0
12/25/2014 6:01:27 PM :   0
12/25/2014 6:01:27 PM :   0
12/25/2014 6:01:27 PM :   0
12/25/2014 6:01:27 PM :   0
12/25/2014 6:01:27 PM :   0
12/25/2014 6:01:27 PM :   0
12/25/2014 6:01:28 PM :   0
```

Figure 10. The Log of Gobetwino Commands Executed on the Arduino when it is Logging Reading Sensors Data in csv Format



NYT Travel - In Transit Blog: A New Way to Tour Taiwan - 4 hours ago

DATA CENTER MONITORING REPORT!

[redacted]30@gmail.com

7:02 PM (14 hours ago)

to

DataCenter sensor Vusualization report for2014-12-26 10:02:05

/home/pi/Desktop/sensorarduino/testlog.csv
1.7 KB

Figure 11. Detail email within Attached Log File Sent to Supervisor

6. Conclusions

The temperature environmental monitoring system for auxiliary data centers represents a step forward towards addressing awareness monitoring of data centers located far with ability to receive a notification by email. With the proposed monitoring data center, environmental organizations can effortlessly install the system with no additional wiring installation and maintenance cost. Additionally, our proposed system is implemented to view the system logs and status via Internet in real time. Simply monitoring this data is not enough. We need to make sure performing something to our environment monitoring spots problems that is whole point of monitoring. The implementation of the proposed environmental monitoring costs a lot less budget. A simple Arduino can offer more data for lot less money. For the budget conscious you can get a cheap Arduino for less $ 5. We have also presented the implantation of the temperature monitoring using Gobetwino, Arduino and Raspberry Pi. As future work, the system design presented in this project can be expanded in a number of different aspects. For instance, additional user friend web monitoring portal to allows supervisors to monitor sensor data in real time through charts which can be designed using HTML5.

Acknowledgments

This research was supported by the Brain Busan 21 Project (2016), Nurimaru R&BD project (Busan IT Industry Promotion Agency) (2016), and Dong-Eui University Research Institute (2016).

This paper is a revised and expanded version of a paper entitled, “Auxiliary Data center Environmental Monitoring system with Lower Hardware Cost”, presented at 12th International Conference on Multimedia Information Technology and Applications, Souphanoubong University, LuangPrabang, Laos, July 4-6, 2016.

References

[1] N. Rasmussen, “Calculating Total Cooling Requirements for Data Centers”. White Paper 25 Rev3, Schneider Electric., **(2003)**, pp.1-7.

[2] N.E. Sayed, I. Stefanovici, G. Amvrosiadis and A.A. Hwang. “Temperature Management in Data Centers: Why Some (Might) Like It Hot”, Proceedings of SIGMETRICS’12, England, UK, **(2012)** June 11-15.

[3] Arduino Uno, ww.arduino.cc/en/Main/ArduinoBoardUno

[4] Raspberry Pi, www.raspberrypi.org/

[5] Gobetwino, http://mikmo.dk/gobetwino.html.

[6] C.D. Patel, C.E. Bash, R. Sharma, and M. Beitelmal. “Smart cooling of data centers”. Proceedings of 2003 International Electronic Packaging Technical Conference and Exhibition, Maui, Hawaii, USA, **(2003),** July 6-11.

[7] K. Rajamani and C. Lefurgy. “On Evaluating Request-Distribution Schemes for Saving Energy In Server Clusters”. Proceedings of the 2003 IEEE International Symposium On Performance Analysis of Systems and Software**,(2003),**pp111-112.

[8] R.Sharma, C.Bash, C. Patel, R. Friedrich, and J.Chase.”Balance of power: Dynamic thermal management for Internet data centers.” IEEE Internet Computing 9(1), **(2005)**, pp.42-49.

[9] T. Evans, “Humidification Strategies for Data Centers and Network Rooms”. White Paper N0 58 Rev1, American Power Conversion by Schneider Electric., **(2008),** pp.1-14.

[10] J. Brandon.”Going Green In Data Center: Practical Steps For Your SME To Become More Environmentally Friendly”. Processor**,(2007)**

[11] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, " Wireless sensor Networks for Habitat Monitoring," in Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, New York. **(2002),** pp. 88-97.

[12] C. Gui, and P. Mohapatra, "Power Conservation and Quality of Surveillance in Target Tracking sensor Networks," Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, **(2004)**, pp. 129-143.

[13] T. He, S. Krishnamurthy, J.A. Stankovic, T. Abdelzaher, L. Luo, T. Yan, R. Stoleru, G. GU, J. Hui and B. Krogh, " Energy-efficient surveillance System using Wireless Sensor Networks," Proceedings of the 2nd of the 2nd International Conference on Mobile systems. **(2004),** pp. 270-283.

[14] S. Choochaisri, V. Niennattrakul, C. Intanagonwiwat and A.C. Ratanamahatana, "SENVM: Server Environment Monitoring and Controlling system for a small Data Center Using Wireless Sensor network," Proceedings of the 1st International Computer Science and Engineering Conference, (**2010)**, pp. 23-28.

[15] L. Nkenyereye, J.W. Jang. "Auxiliary Data center Environmental Monitoring system with Lower Hardware Cost". Proceedings of the 12th International Conference on Multimedia Information Technology and Applications, Luang Prabang, Laos**,(2016),** July 4-6.

[16] J. Waldo, "On System Design", In an Essay Series Published by Sun Labs, Perspectives 2006-6, pp1-40, 2006.

[17] Waterproof DS18B20 Digital temperature sensor + extras, https://www.adafruit.com/product/381

[18] WinSCP. Free SFTP, SCP, and FTP client for Windows, https://winscp.net/eng/docs/lang:fr

[19] Fritzing, http://fritzing.org/home/

[20] Crontab-Quick Reference, http://www.adminschoice.com/crontab-quick-reference