Journal of Basic and Applied Engineering Research
p-ISSN: 2350-0077; e-ISSN: 2350-0255; Volume 4, Issue 4; April-June, 2017, pp. 307-312

http://www.krishisanskriti.org/Publication.html

Raspberry Pi Based Real Time Data Acquisition Node for Environmental Data Collection

Davinder Pal Sharma*, Keide Samuel, Kris Ramoutar, Takim Lowe and Isaiah David

Department of Physics The University of the West Indies St. Augustine, Trinidad and Tobago
*E-mail: *davinder.sharma@sta.uwi.edu*

Abstract—*A data acquisition node/system uses various sensors, dedicated hardware and respective software to measure and store physical signals like voltage, temperature, pressure etc. Present paper describes the design of a real-time data acquisition system, using a low cost, credit card sized computer known as a Raspberry Pi to collect various environmental data related to wind, solar, rain, temperature and humidity, which can be used for further analysis and model development. Wind direction and wind speed was obtained using a wind-vane and an anemometer by Davis. To collect rain data, a rain gauge with an ultrasonic distance module was used and Kipp & Zonen CM-11 Pyranometer was used to collect data related to solar irradiance. The temperature and humidity related data was collected using a DHT-11 sensor. Data from all the sensors was collected by Raspberry Pi using python scripts for each respective sensor and then collected data were logged and stored on a web server. Environmental data collected by our system was verified and found satisfactory.*

Keywords: Raspberry-Pi, real-time systems, data acquisition, weather station.

1. INTRODUCTION

Data acquisition is the ability of an individual to take analog data or physical signals such as temperature, humidity, wind speed, wind direction, solar radiation and rain fall. The data is then converted into digital form to feed into a computer so that the data can be read and logged. The devices used for data logging are known as data acquisition systems. These data acquisition systems are mainly used by scientist and engineers in fields such as industry control systems, research laboratories and various sectors which require the measurement and testing of equipment. Information gathered from these systems can analyzed for forecasting, modelling and simulations purposes. The data gathered can ensure reliability or to make certain that machinery is operating safely [1].

Fig. 1 shows a basic data acquisition system (DAS), which incorporates the use of common hardware relevant to the collection of data. A DAS comprises of three parts: an input or output sub system, the host computer and controlling software. Major components of DAS are the sensors, which convert physical parameters into electrical signals, analog to digital converters, which can convert analog signals to digital signals and signal conditioning circuitry which can be used to convert sensor signals into a different form that can be changed into digital values. Software programs are usually used to control acquisition systems and are developed by using different programming languages such as assembly, Basic, C/C++, Java etc. However, some systems such as the Power DNA Cube, which runs a standalone Linux application combines the Input/ Output Sub – system and the host CPU in the same package. Standard data acquisition systems have a separate host computer and Input/ Output System, which requires a method to communicate with each other. The most common interfaces for the data acquisition system are the PCI bus, PXI bus, Ethernet, USB and Serial Input/ Output.

Most DASs can be categorized into two types: an external box configuration or an internal plug – in board configuration. External system configurations can be classified as Ethernet and USB, while internal configurations can be classified as PCI and PCI Express. For an external box approach, a major advantage would be that the actual system can often be located closer to the sensors and the actuator, where it interfaces with. This allows for a reduction in the length of wires required to connect the Input/ Output to the data acquisition system, which in turn reduces the noise level in the field wiring. Also, the proximity of the Input/output system to the desired physical parameter is crucial as the increased proximity allows for a reduction in both the time and resources required for connecting the field wiring. With the internal plug in board configuration, it has the advantage of closer connectivity to the controlling CPU.

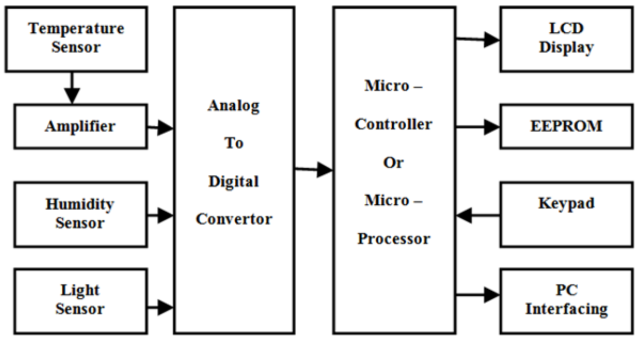


Fig. 1: Basic data acquisition system

This type of data acquisition system allows for the system to function faster and collect data at higher rates. The simplicity of the system is that everything can be in one box, however, over the years as technology advances and the intelligence of the external box system grows more, the difference between the two becomes much more obvious [2].

The Raspberry Pi, which is a low-cost, credit card size computer, can be used in various applications, ranging from toys to industrial circuits to home automation. One can use Raspberry Pi for entertainment, for instance in the creation of a media centre using television, creating a versatile game station by creating own game and allow one to wirelessly control one's stereo. Another category that can be discussed is its use for children, where it can be used for Kano Kit, as an educational tool to show children how to program in a funny and interactive way. Using various hardware components along with software, one can create or design various forms of technology, which include a smart fridge, smart phone, Raspberry Pi unmanned aerial vehicle (UAV), robots, wireless extender and even make one's home fully secured & automated [3-5].

In the present study, the Raspberry Pi (Model B+) was programmed to be used as a real-time data acquisition node so that environmental data i.e. solar irradiance, wind speed, wind direction, rain level, temperature and humidity can be obtained. Various sensors were used to acquire the data for each of the parameters.

2. SYSTEM DETAILS

Block diagram of data acquisition node designed around Raspberry Pi to collect environmental data is shown in Fig. 2. Environmental data related to wind speed/direction, rain, temperature/humidity and solar radiation was collected through anemometer, ultrasonic module, DHT-11 sensor

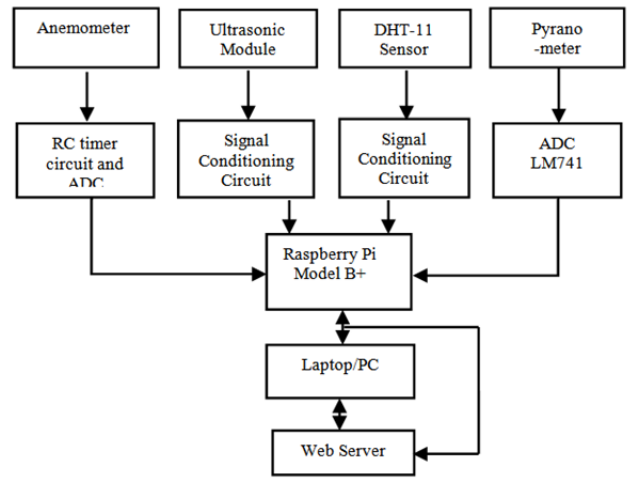


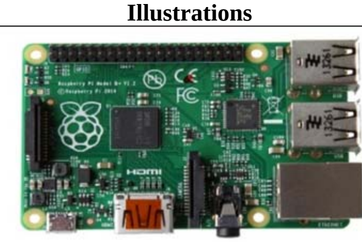
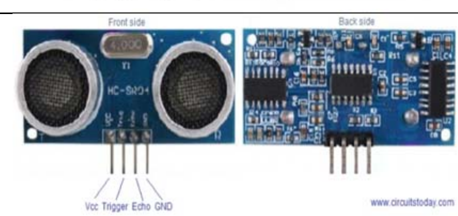
Fig. 2: Block diagram of Raspberry Pi based DAS

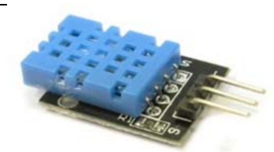
and pyranometer respectively and then was sent to Raspberry Pi using python scripts through signal conditioning circuits and ADCs. Collected environmental data was then stored on the web server for real time access using PC or any mobile device. Further detail of the designed node is given below:

3. HARDWARE

The main components used in the present study are listed in Table 1. The heart of present DAS is Raspberry Pi (Model B+), which is a 6 OZ circuit board with 700MHz CPU and 512MB SDRAM. It has an HDMI port, a RJ45 Ethernet port, 4 USB 2.0 ports, a micro USB port for power, a SD card slot, a 4-pole A/V jack, 2 Camera ports and 40 pins for input and output [6].

TABLE 1: List of Main Components

Components	Illustrations
Raspberry Pi B+	
HC-SR04 Ultrasonic Distance Module	

DHT-11 Sensor	
Davis Anemometer Pro II	
Kipp & Zonen CM-11 Pyranometer	

The Davis anemometer was used to detect wind speed with accuracy of ±5% and wind direction with accuracy of ±3^{o}C using wind cups and wind vane respectively with operating temperature - 40°C to +65°C [7].

The HC-SR04 ultrasonic module, used for measurement of rain level, includes an ultrasonic transmitter, receiver and control circuit. The accuracy of the module can reach upto a maximum of 3mm. Module has a VCC pin for a 5V DC power supply, a TRIG pin for transmission of signals, an ECHO pin to create an output waveform from the module and a GND pin [8].

The DHT-11 is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and produces a digital signal on the data pin. It’s simple to use, but requires careful timing to obtain data. It has humidity range of 20-80% with ±5% accuracy and temperature range of 0-50ºC with accuracy of ±2ºC [9].

The CM-11 pyranometer from Kipp & Zonen is a highly accurate instrument, which was used for the measurement of solar radiation on a plane or level surface. It has sensitivity of 4-6µV/Wm^{-2}, electrical impedance of 700-1500Ω, operating temperature of - 40 to 80ºC and irradiance range from 0-1400W/m^{2} [10].

4. SOFTWARE

The Raspberry Pi runs on a free operating systems (OS) known as the Linux-kernel operating systems. The operating system used for the Raspberry Pi in the present study is the Raspbian. One of the advantages of this OS is that it offers a text-based menu on boot. The systems can be configured if necessary, enable SSH and boot automatically into the user-friendly LXDE Graphic User Interface (GUI). The recommended operating system, for the beginner is Raspbian, followed by RISC OS and Arch as they become more familiar with Linux and its workings [11].

Python was used as the coding language to send instructions to the Raspberry Pi, which is a high-level, interpretive, interactive and object-oriented scripting language. Furthermore, it is designed to be highly readable, uses English keywords frequently where as other languages use punctuation and has fewer syntactical constructions than other languages. The Python allow Raspberry Pi to collect data in real time[12].

PHP is a programming language which was used to build custom web content to serve as a browser, to communicate with a database, evaluate form data sent from a browser and receive cookies. PHPMyAdmin tool was used in PHP to allow the administration of MySQL using a web browser. This tool allow for the creation and editing of databases and tables, which can be imported onto a web server or webpage through importing data onto the databases and linking with the PHP scripts which one would use [12].

5. METHEDOLOGY

Fig. 3 illustrates the method used for the collection of data for our DAS using a Raspberry Pi. The free Linux base

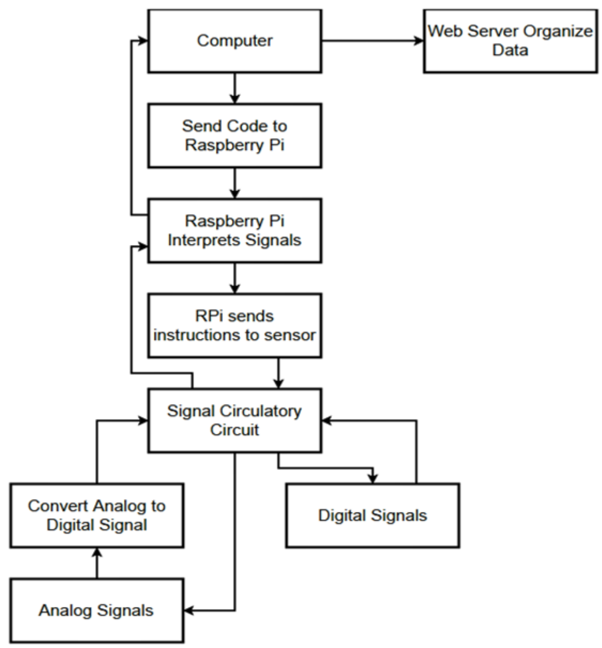


Fig. 3: Data acquisition method

operating system called Raspbian for the Raspberry Pi was downloaded from the Raspbian website, which is known as NOOBS and placed onto the Raspberry Pi via an 8GB SD card. The operating system was then installed and updated using an external monitor, keyboard, mouse and internet connection. The Raspberry Pi was then programmed to operate from a computer by sending and receiving data from an Ethernet cable. This operating process is known as headless where no external components are needed besides an Ethernet cable and a computer. The sensors described in Section II were then connected to Raspberry Pi using the GPIO pins where the data for wind, rain, solar radiation, temperature and humidity was obtained.

6. RESULTS AND DISCUSSION

Wind Data

The wind speed was obtained from the anemometer, which is an analog sensor that requires an ADC to convert the analog signals from the anemometer to digital signals so that the Raspberry Pi can interpret the information being received. The data obtained was in terms of voltage and converted to speed using an equation developed from researching the specs and limits of the anemometer. The data received was then placed in a format where it can be read and analysed. The wind direction was obtained using the wind vane, which is a digital sensor known as the potentiometer that requires a RC timer circuit to log readings directly to the computer. The capacitor in the RC timer circuit stores the voltage in accordance to the direction of the position of wind vane and those voltage values were used to determine the direction.

The codes were then developed using python for each of the sensor so that data can be received for the different parameters. The codes were created using a program known

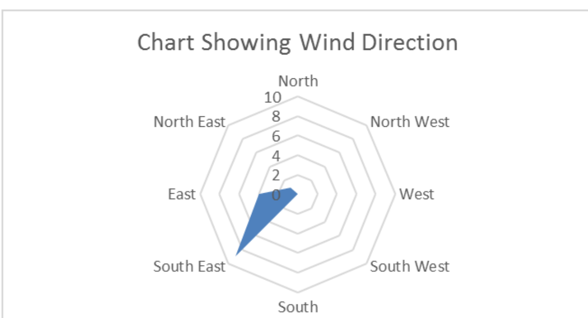


Fig. 4: Consistency of the direction of the wind

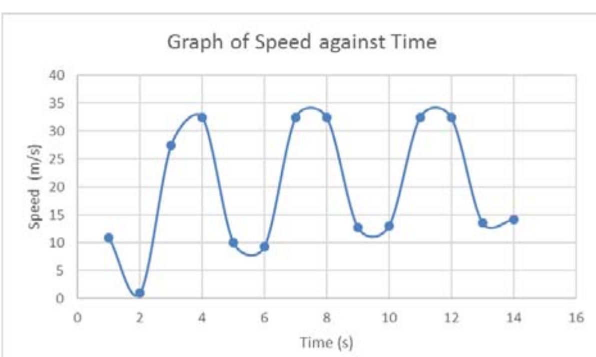


Fig. 5: Speed of the wind over a period

as Notepad++ and it was sent to the Raspberry Pi using Filezilla. The python file was then ran so that data can be received from each of the sensors for both wind speed and wind direction. Plot of the data collected for the wind speed and wind direction are shown in Figs. 4 and 5 respectively. The data was then transferred from the Raspberry Pi onto the web-server as shown in Fig. 6, where it was stored for further analysis.

7. RAIN DATA

To collect data for rainfall, the ultrasonic distance module was used. Module was placed over one (1) Litre measuring cup filled with some water to obtain rain data at one (1) minute intervals as set within python code. Using a water bottle, at each interval, the water was added to the measuring cup until all ten readings were recorded and stored accordingly. The collected data was recorded as a 'csv' file onto the Raspberry Pi itself into the WiringPi folder where the python code itself for the sensor was stored.

Fig. 7 shows variation of distance with time as obtained from ultrasonic module. It can be seen that the distance from the sensor to the surface of the water is inversely proportional to the increase in the amount of water in the cup. There are fluctuations in the reading, which are caused because as the distance between the sensor and the surface

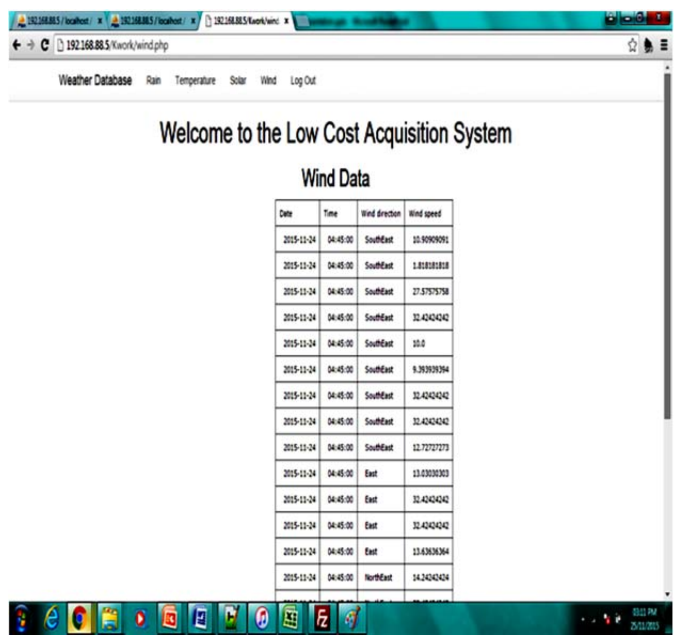


Fig. 6: Wind data stored on a web-server

of the water reaches below the minimum range of the sensor, the accuracy of the sensor drops. This is the same for if the sensor exceeds its maximum range.

8. TEMPERATURE AND HUMIDITY DATA

The temperature and humidity data as shown in Fig. 8 and Fig. 9 was obtained using the DHT-11 module. DHT-11 allowed for a precise temperature reading of ±2ºC and a humidity accuracy of ±5% per bit parsing for every one (1) second relay interval. The python script was written to read and display the temperature and humidity values to precisions up to three decimal places for set interval, which would progress indefinitely, constantly producing a reading as each interval elapse. It was observed that it is more efficient to have it running indefinitely so as to maintain the rigorous timing necessary to read the very sensitive sensor. Thus, a better average of read outs can be obtained.

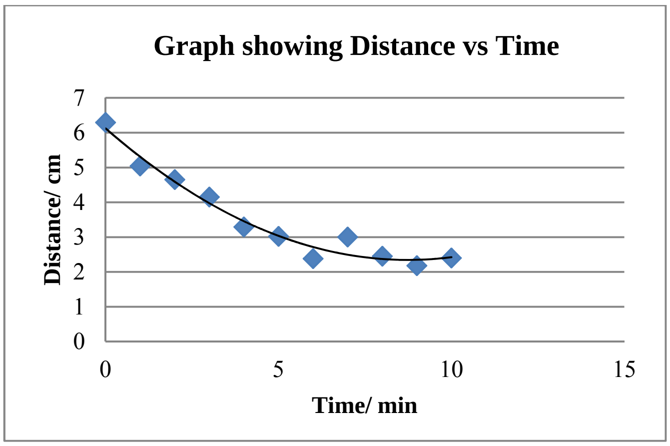


Fig. 7: Distance vs. Time during the collection of rain water

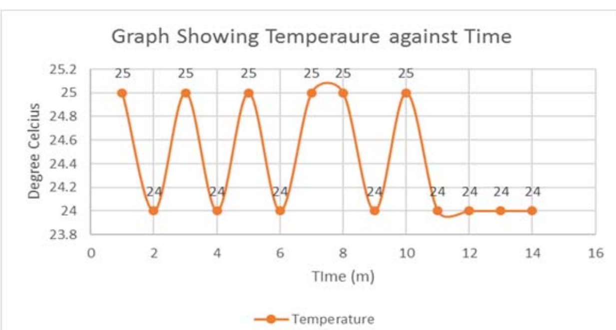


Fig. 8: Graph showing Temperature in degrees over a period of time

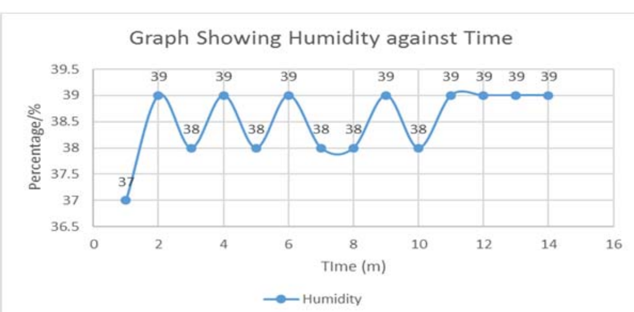


Fig. 9: Graph showing Humidity percentage over a period of time

9. SOLAR DATA

Solar intensity data was acquired with the use of the Kipp & Zonen CM-11 Pyranometer. The low voltage output was boosted up with an appropriate voltage amplifier. The amplified analog signal was then transferred to an ADC, the output of which was sent to the Raspberry Pi. Python script was written to convert the voltage input into numerical readings of solar intensity using the formulae provided by Kipp & Zonen and data was recorded every second to ensure real-time data acquisition. Snapshot of solar data obtained from CM-11 is given in Fig. 10.

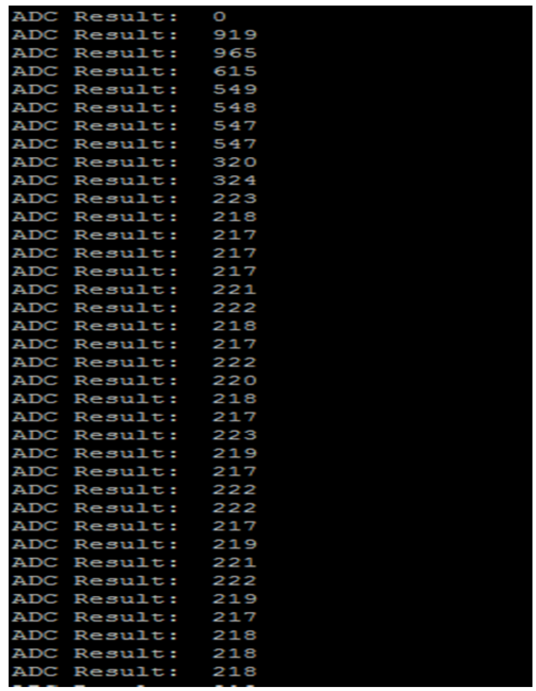


Fig. 10: The data collected for solar irradiance

10. CONCLUSION

Raspberry Pi based real-time data acquisition node to collect environmental data was presented in this paper. Environmental data related to wind, solar, rain, temperature and humidity were collected with the help of appropriate anemometer, pyranometer, ultrasonic and DH-11 sensors. Collected data was logged on to the web-server using PHP for anywhere anytime access through any PC or mobile device. The Raspberry Pi B+ was found very suitable for implementation of weather station, which was able to collect data in real time. Collected data can be used to develop models to understand intermittent behaviour of wind and solar systems that can help further in studying their integration challenges with electrical grid.

The data acquisition system designed here is relatively cheaper than most weather stations. Other weather station's computer interface and software are sold separately from the hardware, which makes them more expensive. The software used in present system is more users friendly and easier to acquire and store data as compared to most weather stations. However, the data from our data acquisition system has a higher error percentage due to the systems error margins as well as sensor errors margins while other systems only has the sensor errors to consider.

11. ACKNOWLEDGMENT

Authors are thankful to UWI T&T RDI Fund to provide necessary financial support to carry out present work through project on “Capacity Building and Research on Smart Grid Technology in the Caribbean Region”.

REFERENCES

[1] R. Tuna Balkan, Data Acquisition Handbook, 3rd ed, Measurement Computing Corporation, 2012, Chap. 1.
[2] S. Dave, Data Acquisition System and Data Loggers, Slideshare, 2011.
[3] M. Cook , J. Evans and C. Brock, Raspberry Pi Projects for Dummies, John Wiley & Sons Inc., NJ, 2015.
[4] D. P. Sharma, A. Bladeo and C. Phillip (2015), “Raspberry Pi based smart home for deployment in the smart grid ,” Int. J. of Comp. Apps, vol. 119, pp. 6-10, June 2015.
[5] R. Dudas, V. B. Christopher, B. Alex, Z. A. Syed, and T. O. Matthew, “Inexpensive telecytology solutions that use the Raspberry Pi and the iPhone,” Journal of the American Society of Cytopathology, vol. 3, no. 1, pp. 49-55, 2014..
[6] L. Ada, “Introducing the Raspberry Pi Model B+,” Adafruit Industries. 2015.
[7] User Manual, “Integrated Sensor Suite for Vantage Pro2 and Vantage Pro2 Plus ,” Davis Instruments, CA, 2015.
[8] Data Sheet “Ultrasonic Ranging Module HC - SR04,” ElecFreeks, 2017.
[9] Data Sheet, “ DHT11 Humidity & Temperature Sensor,”D. Robotics, UK, 2010.
[10] D. Brooks, “Measuring Sunlight at Earth’s Surface: Build Your Own Pyranometer,”2007. http://www.instesre.org/construction/pyranometer/pyranometer.htm, (Accessed on March 20, 2017).
[11] M. Richardson, and S. Wallace, “Getting Started with Raspberry Pi,” O’Reilly , CA, 2013, pp. 32-42.
[12] M. Maclean, “Raspberry Pi: Measure, Record, Explore,” Leanpub, 2015.