Volume 5, Issue 12, December 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Wireless Ambient Monitoring Sensors using Power Efficient Thread Protocol and PSoC Microcontrollers

Yogesh Dhanawade[1,2], Ganesh Gangane[1,2], Pratik Bhosale[1,2], Harshad Bhore[1,2], Rama Gaikwad[3,2]
[1]Student, [3]Assistant Professor,
[2] Department of Computer Engineering, ABMSP's APCOER, Pune, Maharashtra, India

Abstract— This paper describes implementation of a compact self-charging and reliable ambient monitoring system using existing micro controllers and sensors for the sake of power saving, cost reduction and live monitoring. The proposed system monitors ambient light intensity, humidity, temperature, carbon dioxide level and absolute pressure of indoor spaces and provides feedback to control use of light and reduce energy consumption of air conditioners and other comfort making electronic devices. It uses any preferred wireless communication media and most power efficient and low cost Thread protocol for remotely controlling and gathering sensed data. This paper covers some of the drawbacks of the existing system and tries to enlighten some extra features into it.

Keywords— IoT; Wireless Sensors Network; Thread Protocol; Beagle Bone Black; Arduino Duemilanove (Atmega 328); IPv6 for Low Power Personal Ad-hoc Network (6LoWPAN).

I. INTRODUCTION

There are many ambient monitoring systems developed and existing from last decade, but they are more specific to their application, no ambient monitoring system is generalized neither it is commercially available for customized applications. Air quality monitoring is the major part of ambient monitoring and plays a vital role in maintaining comfort, health and nature friendly environment in the industrial infrastructure. According to the United States Environmental Protection Agency, the Department of (AQS) Air Quality System defines six different kinds of common air pollutants, namely, carbon monoxide, nitrogen oxides, ground level ozone, lead, sulphur oxides and particulate matter [3]. These are the major so called criteria pollutants and thus are required to be tracked on, to ensure the quality of air we breathe [4], [5].

The major issues arising from such unhealthy air are categorized under 'Sick Building Syndrome'. [1] These include a set of symptoms like headaches, dizziness, and stress in concentration. The suggested system is a step towards solving such problems by providing necessary data samples for analysis and decision making.

The present paper [1] focuses on similar system but it has certain drawbacks. These flaws and future implementations are covered in the proposed system. In addition to this we have made the communication reliable and faster with the use of Thread protocol. We have also concerned several other solutions and currently available products which are mentioned in the references.

II. RELATED WORK

In earlier studies, it was observed that embedded and limited ambient sensors were incorporated to recognize environment parameters and assess them either for automation of electrical loads or census estimation by environment related government agencies and NGOs [3], [4]. The major drawback of previous study is that researchers focused on limited number of sensors and applications. As a result proper analysis can be achieved if more number of sensors are considered in an uniform generalized system structure. This is possible, only if we have a sensor independent node which can communicate on most widely accepted existing wireless technologies. In other findings as in [16], we come across various ambient sensing systems which have made efforts towards increasing the battery lifetime of the sensor node [6]. But it finally has to be replaced or recharged manually at the choice of time and cost. The solutions for this, like self-charging or light harvesting sensor nodes were discussed in [2]. Some solutions for battery lifetime enhancement and using low-power sensors are discussed in [6] and [7]. Furthermore, the major constraints for these types sensors are low bandwidth and delay tolerance, mainly for the applications like monitoring of real-time environmental parameters [21]. In [22], lighting systems were evaluated using an experimental testbed under different control schemes. In addition to this, an implicit assumption in these lighting control schemes was that the sensors sample the environment fast enough, the control adaptation rate is high and that the luminaires can be driven at a high rate so that occupants do not experience visible light fluctuation during actuation. As we have seen that sensor nodes have limited storage capacity, the use of Storage Area Network can be more feasible to store and collect the daily monitoring data [15], [17]. A similar system was proposed in [18] and [19] but it had a limited scope for Internet of Things using only Wi-Fi and RFID. Most ambient sensors were custom built and had to adhere with their manufactured wireless communication mediums. Moreover majority of the ambient WSNs used traditional protocols, widely UDP and 6LoWPAN just for the sake of faster and light weight data transfer which is often vulnerable and more prone to data loss while in transit.

III. HARDWARE ARCHITECTURE

A. General Overview

The designed sensor node is a combination of an Arduino Duemilanove microcontroller and respective sensors. This device captures various sensor values and transmits them to an IP address as and when required. The defined IP-address is preset in the Arduino's code and is destined for the cluster head mainly the Beagle Bone Black or the gateway interface device. The Beagle Bone Black which acts as a gateway cum router will collect the sensor node's transmitted data store them on the local SAN storage or global cloud storage. All this logic is implemented in the Beagle Bone Black as a Python program and is initiated at the time of setup.

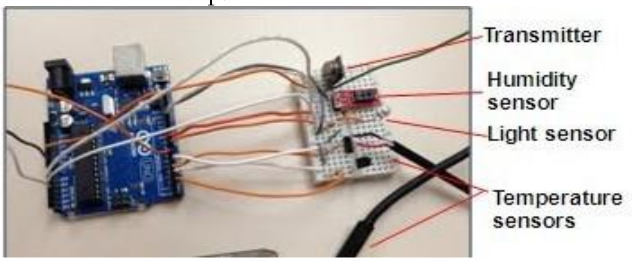

Fig. 1 Arduino setup as Wireless Sensor Node.

B. Internal Structure

The central control unit is represented by an open source Programmable System on Chip micro-controller, the Arduino Duemilanove board. This initiates the main functioning and operation of the ambient sensors. The system can be categorized into the four major functional blocks: (1) The Input/Sensing Unit, (2) The CPU and Storage Unit, (3) The Communication Unit and (4)The Backbone Power Supply Unit.

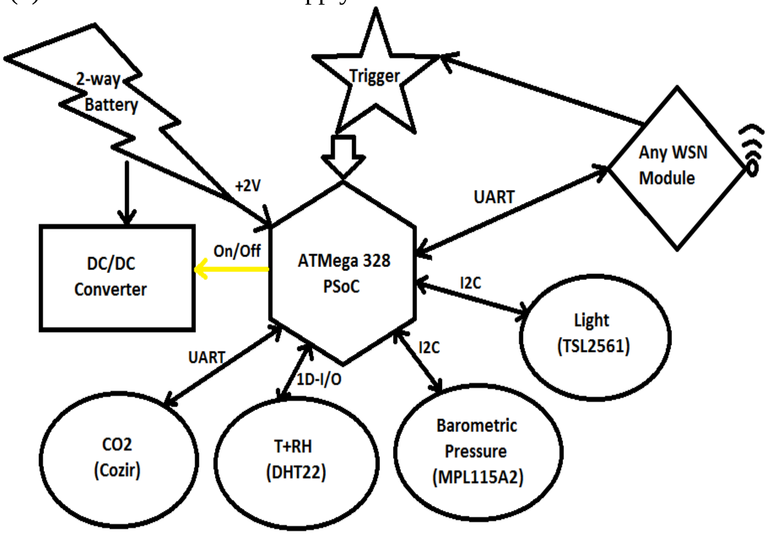

Fig. 2 Proposed Hardware architecture.

The Input/Sensing Unit, this block consists of the sensors like temperature, humidity, light, CozirTM CO2 Ambient Sensor, and a barometer which provide input for the system. These five senses are analogous to human sensing organs. These were chosen as their popularity, availability in local electronic stores, cost and most probably their accuracy and precision. This makes the system more feasible and survive with the other competitive counterparts. The CPU and Storage is the core micro-controller. On other hand the Communication Unit is implemented using Bluetooth module HC-05 and Ethernet shield for some nodes. As shown in the Figure. 2, the system has a 2-way Power Supply of a battery and a Photo Voltaic Cell shown in Figure.3. The overall integration makes the system to sense different parameters simultaneously.

C. The Arduino Duemilanove (PSoC)

The Programmable System on Chip, used as an acronym PSoC, is defined here as an Arduino, an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.[8] You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so we use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery. Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-

24, Phidgets, MIT's Handyboard, and many others offer similar functionality [8]. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems: Inexpensive, Cross-platform, Simple, clear programming environment, Open source and extensible software, Open source and extensible hardware [8].

D. Beagle Bone Black (for Sink Node/Cluster Head)

We have used Beagle Bone Black as part of this WSN because of its potential to offer enough memory for overall implementation and its processing capabilities. It uses an ARM processor for computation and has an inbuilt manufactured version of Linux Debian Operating System [9]. So it much feasible to develop and implement various protocols and configure the in an infrastructure. It also supports various connectivity options like Bluetooth, X-bee and 802.11 Wi-Fi Ethernet. It has ability to support external extended storage by means of MicroSD card and USB storage. This helped in replacing the traditional gateways, access points and routers.

E. Thread Protocol

This is the new feature which makes the prototype more secure and reliable [10]. Google's newly introduced protocol is light weight and specifically designed for IoT. Here it is implemented in the Beagle Bone Black as a Linux library for Sink Node (Beta Version). While the Arduino holds the client implementations for transmitting data. It is available as open source so can be implemented easily in any Linux system. Currently it is used for communication between sensor and the sink node only. It can be later enhanced over the entire network after its official release. The major advantages of Thread protocol are discussed in the Software Implementation section.

F. Overview of Sensors Used

The type and specifications of sensors used here is identical to the ones used in paper [1]. The only enhancement made in sensors is that we have replaced the Light sensor TSL2561 with a photo voltaic cell 3WK16473 and a charging circuit, in order to increase the power efficiency. The same can be found in Figure.3

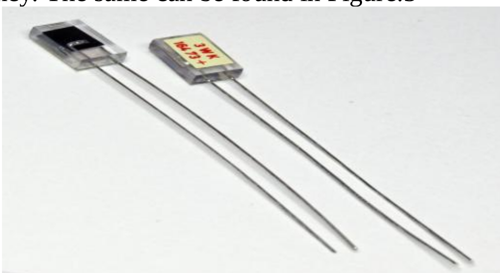

Fig. 3 Photo voltaic cell 3WK16473 used as Light Sensor.

G. 2-way Power Supply and Reverse Battery Protection

The proposed system was designed with a 2 -way power supply which includes a 2V, 1000mAh rechargeable Li-ion battery and a photo voltaic cell with a charging circuit. This photo voltaic cell acts both way, as a light sensor in active mode and a recharging cell in sleep mode. The photo voltaic cell has following specifications:

TABLE I PHOTO VOLTAIC CELL SPECIFICATION

Parameters	Specification
Photo-diode Type	3WK16473
Efficiency	High
Reverse voltage	> 1.5V
Operating Temperature	-25˚C to +85˚C
Storage Temperature	-25˚C to +85˚C

Its charging circuit has a reverse protection logic designed by Robosoft.

IV. SOFTWARE IMPLEMENTATION

The code is loaded as a firmware into the Arduino's flash memory, which was developed and cross compiled using Arduino IDE 1.0. The code has provisions for two independent modules: The sensor interaction and calibration module, which interfaces the sensors and network. And the Communication module which configures the network related parameters, such as protocols and server credentials like (SSID, IP-address, Authorization passwords, etc.). All this takes place via a serial interface.

The application normally works in the default mode. It wakes up at regular predefined intervals, assesses and records sensor values, and transmits them to the cluster head. And again sleeps to retain power and charge after meeting a threshold value of battery meter. In case the server or cluster head requests data during sleep cycle, the triggering from the network takes place with the help of a preamble which is continuous sequence of 1's of eight octets. This is coded in the sink node or cluster head (i.e. Beagle Bone Black).

Using Beagle Bone Black as an access point or sink node will reduce the network traffic and also solve the Thread protocol related issues discussed later in this paper. As Beagle Bone Black contains Linux Debian Operating System, it is configured with the Thread protocols beta version, along with this it also allows data aggregation, proxy server, and to cache most recently synced sensor data to reduce sensor interaction and power consumption.

The communication takes place with the help of Thread protocol which is the newly introduced specialized protocol for Internet of Things by Google's Nest Lab. [10] and has following advantages over others:

1) IPv6 based,
2) Lightweight and low latency,
3) Not a whole new standard,
4) Collection of existing IEEE and IETF standards,
5) Runs on existing 802.15.4 based products. [10]

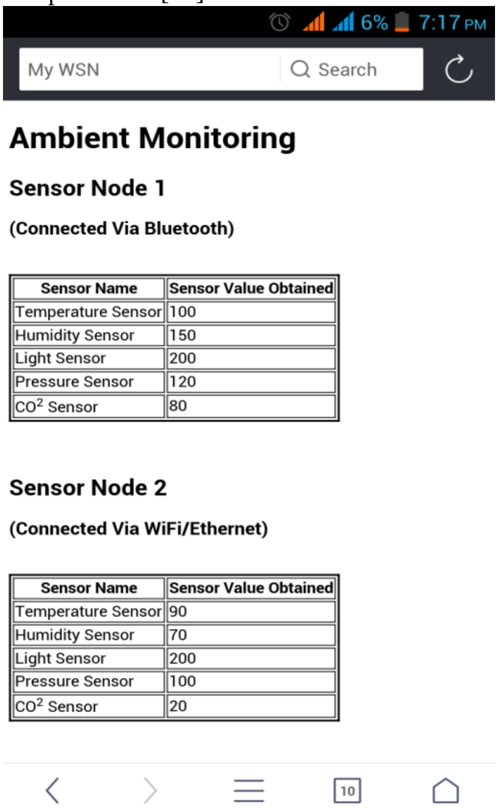

Fig. 4 Screen-shot of Sensor Data Displayed Locally on Android Phone

But in spite of so many pros, the thread protocol is not feasible with some old existing networking devices. These compatibility issues are solved in the latest official release of Thread protocol which may be an add-on for the future implementations.

A. Real-time Scenarios

As we have discussed earlier the Arduino sensor node is programmed and made to sleep and wake-up at regular time intervals. This can be a predefined schedule or a manual trigger from the network. On the hand Beagle Bone Black will act as a bridge between multiple sensor nodes so as to interface all Wi-Fi, Bluetooth and X-bee mediums. The collected sink data can be analysed, aggregated and sent to the server as per requests. The server may request data in either UDP or TCP formats.

As the concept of IoT evolved from the efforts towards environment care and monitoring, we must attain a level of maturity and standards in order to encourage use of sensors in other domains not limited to health, home and commercial applications [11],[12]. These can be noticed from the discrete areas discussed in [13] and [14]. By satisfying the requirements for use in WSNs, namely low cost, low power consumption, multi-functionality, small dimensions and wireless communication capabilities, the sensor presented in this paper represents a wireless sensor node [1].(Fig.1).

Recently the new context aware systems have been introduced as a part of Ubiquitous and Pervasive computing which use ambient sensors for implicit Human to Computer Interaction (iHCI). This technology is used for controlling electrical loads and lighting devices by a smart system named as context aware for its intelligence and ability to sense the surrounding environment. They are from an emerging field of Artificial Intelligence.

V. LIFETIME OF THE SENSOR NODE

To estimate the lifetime of a sensor node we have to make some assumptions that there should be insufficient light to charge the battery with a photo voltaic cell and there is no variation in illuminance. At this instance the sensor node with sleep modes will operate on battery backup. The capacity of battery may be calculated as [2]

E = 500mAh X 2.0V = 1000 mWh

By design, in default mode, the wireless sensor node has a power consumption of $P_{stand\ by}$ = 0.020mW. Thus, at 50% energy charge by photo voltaic cell, the lifetime is given by [2]

$$T_{life} = \frac{50\% \times E}{P_{standby}}$$

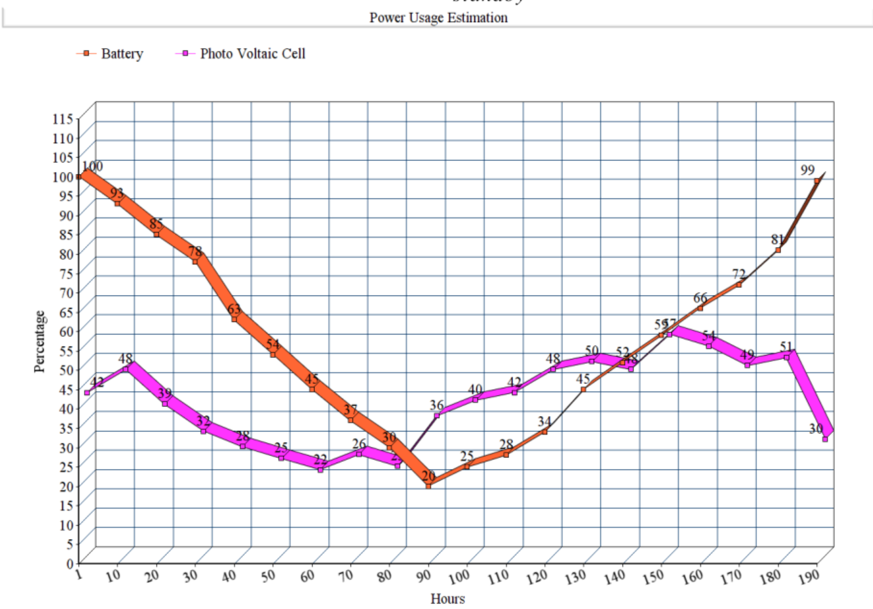

Fig. 5 Power Consumption Estimation

This graph in Figure.5 shows the power usage estimation of both battery and photo voltaic cell with charging circuit. The plotted data highlights that the battery charging activates only when its threshold falls below 20%, and the power from the accumulator gets drained to the battery.

TABLE II BATTERY LIFETIME IN DARK ENVIRONMENT

Measurement Cycle in Minutes	Days	Months
1	27	-
2	45	1.5
10	228	7.6
30	618	20.6
60	1159	38.6
100	1986	66.2

The above table was drawn with the observation taken from the system implemented and deployed in our computer laboratory. This sensor node was using only the battery and no charging circuit was embedded in it.

VI. CONCLUSIONS

The correct operation of the system is guaranteed if the power source delivers at least 500mA at 2 V, but it can of course work with higher values, thus allowing high flexibility in the choice of the energy source. The depleting speeds of battery energy of sensor nodes and use of 2-way power will significantly affect the network lifetime of a WSN. This approach can not only relieve the burden of the access points, but can also integrate the energy-aware routing to enhance the performance of the prolonging network lifetime for ambient monitoring sensors. The future implementations can be a smart machine learning algorithm for mining frequent patterns as well as allow fault and intrusion detection by finding unusual behavior of sensor nodes, because wireless security is something that we need to still work on. Making the system context aware can also be future work and is under development. By estimating the cost and accuracy of occupancy sensors we have developed a power efficient sensor cluster of ambient monitoring.

ACKNOWLEDGMENT

A heartfelt gratitude to my parents, college faculty members, friends for supporting this research.

REFERENCES

[1] Silviu C. Folea, Member, IEEE, and George Mois, Member, IEEE *"A Low-Power Wireless Sensor for Online Ambient Monitoring"* IEEE SENSORS JOURNAL, VOL. 15, NO. 2, FEBRUARY 2015.

[2] Ashish Pandharipande and Shuai Li, *"Light-Harvesting Wireless Sensors for Indoor Lighting Control,"* IEEE SENSORS JOURNAL, VOL. 13, NO. 12, DECEMBER 2013.

[3] *The Six Common Air Pollutants,"* http://www.epa.gov/air/urbanair/
[4] *"Criteria from Pollutants Survey,"* http://www.epa.gov/airquality/montring.html#criteria
[5] *"Clean Air Act,"* http://www.epa.gov/air/caa/
[6] S. Folea, G. Mois, L. Miclea, and D. Ursutiu, *"Battery lifetime testing using LabVIEW,"* in Proc. 9th Int. Conf. Remote Eng. Virtual Instrum.(REV), Jul. 2012, pp. 1–6.
[7] Mihai Hulea, George Mois, Silviu Folea, Liviu Miclea, *"Wi-Sensors: a Low Power Wi-Fi Solution for Temperature and Humidity Measurement,"* 978-1-4799-0224-8/13/$31.00 ©2013 IEEE.
[8] *"Arduino Introduction, Installation Setup and Specification,"* https://www.arduino.cc/en/Guide/Introduction.
[9] *"Beagle Bone Black Development,"* http://beagleboard.org/BLACK.
[10] *"Thread Protocol Implementation,"* threadgroup.org/Technology.aspx
[11] D. Larios, J. Barbancho, G. Rodríguez, J. Sevillano, F. Molina, and C. León, *"Energy efficient wireless sensor network communications based on computational intelligent data fusion for environmental monitoring,"* IET Commun., vol. 6, no. 14, pp. 2189–2197, Sep. 2012.
[12] M. Gorlatova, A. Wallwater, and G. Zussman, *"Networking low-power energy harvesting devices: Measurements and algorithms,"* in Proc. IEEE INFOCOM, Apr. 2011, pp. 1602–1610.
[13] J. Ko, C. Lu, M. B. Srivastava, J. A. Stankovic, A. Terzis, and M. Welsh, *"Wireless sensor networks for healthcare,"* Proc. IEEE, vol. 98, no. 11, pp. 1947–1960, Nov. 2010.
[14] C. H. See, K. V. Horoshenkov, R. A. Abd-Alhameed, Y. F. Hu, and S. Tait, *"A low power wireless sensor network for gully pot monitoring in urban catchments,"* IEEE Sensors J., vol. 12, no. 5, pp. 1545–1553, May 2012.
[15] Ariefianto, T., Purwanto, Y., Wiratama, H. *"Storage Area Network based-on Internet Small Computer Standard Interface optimization using Internet Protocol Multipathing,"* IEEE, ISBN: 978-1-4673-4990-1 pp. 303 – 307, March 2013
[16] Srinivas Devarakonda, Parveen Sevusu, Hongzhang Liu, Ruilin Liu, Liviu Iftode, Badri Nath, *"Real-time Air Quality Monitoring Through Mobile Sensing in Metropolitan Areas,"* UrbComp'13, August 11–14, 2013, Chicago, Illinois, USA.
[17] Tate, John. (dkk).2006. *"Introduction of Storage Area Network."*(4th Ed). USA: New York. IBM Corp
[18] C.S. Lee, D.H. Kim, and J.D. Kim, *"An energy efficient active RFID protocol to avoid overhearing problem,"* IEEE Sensors J., vol. 14, no. 1, pp. 15–24, Jan. 2014.
[19] S. Tozlu, M. Senel, W. Mao, and A. Keshavarzian, *"Wi-Fi enabled sensors for Internet of Things: A practical approach,"* IEEE Communications Magazine, vol. 50, no. 6, pp. 134 – 143, June 2012.
[20] J. Haase, J. M. Molina, and D. Dietrich, *"Power-aware system design of wireless sensor networks: Power estimation and power profiling strategies,"* IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, vol. 7, no. 4, pp. 601 – 613, Nov. 2011.
[21] V. Potdar, A. Sharif, and E. Chang, *"Wireless sensor networks: A survey,"* in 2009 International Conference on Advanced Information Networking and Applications Workshops, 2009, pp. 636 – 641.
[22] D. H. W. Li and J. C. Lam, *"Evaluation of lighting performance in office buildings with daylighting controls,"* Energy Build., vol. 33, no. 8, pp. 793–803, 2001.

