

Power Efficient Automated Lighting System using Wireless Sensor Network Approach

Dhiraj Sunehra

Abstract— In the present times, electric power has become a basic necessity in everybody's life. The conservation of electric power has become a prime issue throughout the world. Due to urbanization, several buildings and malls are being constructed every year which increases the power requirements in the future. Hence, an efficient power conservation system is required as an additional support for the power conservation techniques which are being currently used. In this paper, a prototype of an efficient automated lighting system for power conservation based on Wireless Sensor Network (WSN) and Internet of Things (IoT) approach is implemented. A Passive Infra Red (PIR) sensor is used to sense the human movement, whose output is fed to a Arduino processor to control the relay that operates the load (lights). Two ZigBee transceivers are used here for communication between the WSN and a Raspberry Pi based web server. The load connected to each sensor node can be controlled through the Raspberry Pi web server. The status of light on or off is displayed on the webpage. The prototype can be extended to shopping malls by increasing the number of sensor nodes in WSN model.

Index Terms— Power conservation, Efficient lighting system, Wireless Sensor Network (WSN), Passive Infra Red (PIR) sensor, Raspberry Pi, Arduino Uno, ZigBee communication

1 INTRODUCTION

THE shopping malls in the cities are the new generation windows to the country's commercial activity. The big buildings, commercial complexes and malls require a lot of electric power due to continuous operation and their number may become larger in the future. Precautionary measures could be taken to reduce power wastage to some extent, but an efficient lighting system for power conservation is needed to limit the power consumption [1].

This paper presents the implementation details of a prototype automated lighting system using Raspberry Pi based web server that enables the operation of lights in a building from a remote area through internet (Fig.1). This system makes use of Wireless Sensor Network (WSN) and IoT approach. Figure 2 shows the architecture of the WSN. It consists of a number of nodes where each sensor node usually have the network components of low computing power, memory and energy. Each sensor node has a wireless transceiver, a microcontroller and a source of energy. The task of a sensor node is the detection process and transmission of data. It can participate either as a sensing node or a sink node [2]. The WSN communicates with a Local Area Network (LAN) or Wide Area Network (WAN) through a gateway, in many applications.

The Internet of Things (IoT) is an upcoming technology which is a network of physical objects that allows objects to be sensed and controlled remotely across existing network infrastructure. By giving the on/off status on the webpage, the Raspberry Pi which acts as a gateway transmits to the sensor node through ZigBee transceivers. At the sensor node, the Arduino Uno microcontroller operates the light according to the status through a relay. The nodes are equipped with a PIR

sensor, which detects motion and automatically turns on the lights in that area. With a predefined time, we can switch them off, if no activity is detected [3]. The status of load (light on or off) along with time and date is also displayed so that the lighting incharge or a suitable person can check the status and control the lighting even when he is away. This lighting system can save a lot of power. Various authors have proposed similar systems for security and energy management [4, 5].

2 HARDWARE IMPLEMENTATION

2.1 Salient Features of Raspberry Pi Model B+ Board

Raspberry Pi is a small sized single board computer. It is based on the Broadcom BCM2835 system on chip (SoC), and includes an ARM1176JZF-S 700 MHz processor, VideoCore IV graphics processing unit (GPU) and 512MB SDRAM memory [6]. There is no built-in hard disk drive, instead a micro SD card for booting and persistent storage. It has 40 GPIO pins with 26 pin out, four USB 2.0 ports, push-push micro SD version card socket, a 10/100 Ethernet port and High-Definition Multimedia Interface (HDMI) port, a digital replacement with analog standards. Although it has a Linux-based operating system, there is a flexibility to choose the programming languages and installation software.

2.2 Salient Features of Arduino Microcontroller Board

The Arduino board provides an open-source platform with simple hardware and software. The Arduino Uno Board consists of a ATmega328P microcontroller [7]. The Atmel 8-bit AVR Reduced Instruction Set Computer (RISC) microcontroller includes 32 KB of flash memory, 1 KB EEPROM, 2 KB SRAM, 32 general purpose registers, 23 general purpose I/O lines, three timer/counters, interrupts, serial programmable USART, watch dog timer with internal oscillator and a 6 channel 10-bit A/D converter.

• Dr. Dhiraj Sunehra is currently an Associate Professor of Electronics and Communication Engineering in Jawaharlal Nehru Technological University Hyderabad working at JNTUH College of Engineering, Jagtial, Karimnagar, Telangana State, India, PH-+917893278314, E-mail: dhirajsunehra@yahoo.co.in

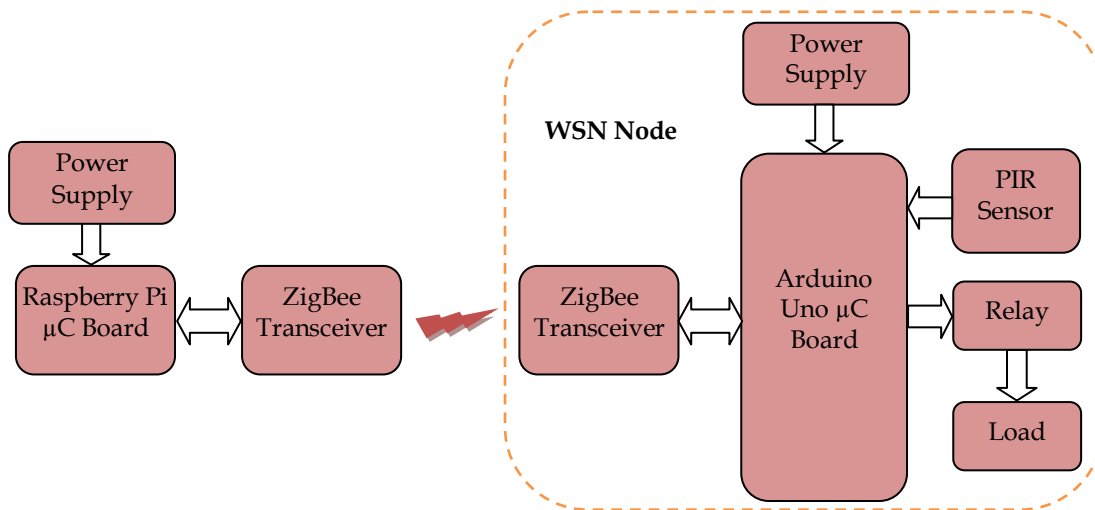


Fig. 1 Block Diagram of a Prototype Power Efficient Automated Lighting System using a single WSN Node

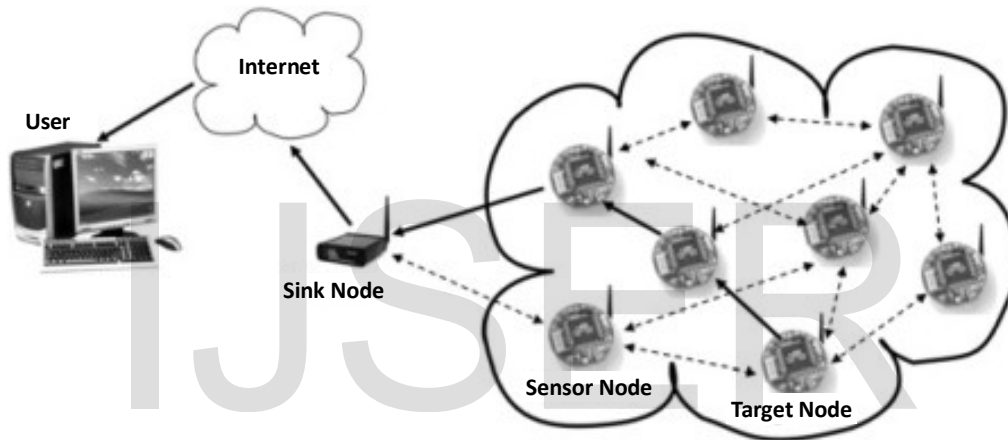


Fig. 2 Architecture of a Wireless Sensor Network

It connects to the computer via USB. Programs are written inside the Arduino integrated development environment (IDE) tool, compiled and are uploaded to the board.

2.3 ZigBee Communication

ZigBee specifications are built around the IEEE 802.15.4 wireless networking standards. ZigBee protocols are used in applications that require low power consumption and tolerable for low data rates. It operates in the range of 2.4 GHz with data rate of 250kbts/sec and a range of 10-100 meters. Routers can be used with ZigBee networks to allow more nodes to interconnect, thereby eliminating the usage of physical data buses like USB and Ethernet cables [8]. Two ZigBee transceivers are used in this prototype implementation for communication between the Raspberry web server and sensor nodes. The Pi acts both as a server and a gateway for wide area networks.

2.4 Passive Infra Red Sensor

PIR sensor is a sensitive pyroelectric device that detects motion by sensing infrared fluctuations from surrounding objects. Its elements are prepared using a crystalline material which generates a surface electric charge on exposure to heat in the form of infrared. The changes in the amount of infrared

striking the crystal change the voltages generated, which are measured by an on-board amplifier. The PIR sensor is small in size, inexpensive, and easy to interface with. It can be used to sense the movement of human or vehicle within the sensor range. Its range is approximately up to 20 feet with tentative environmental changes [4].

2.5 Relay

A relay is an electromechanical device which is actuated by an electrical current. It is an electrical switch which is switched on with the shift of electromagnet present in it. Even highly sophisticated devices which require high power can be controlled by using relays which require small amount of power to switch on. Relays are relatively simple, highly reliable and long-lasting devices.

3 SOFTWARE USED

The various software that are used in the implementation of the automated lighting system are Python programming language, Arduino IDE and the PuTTY software.

3.1 Python Language

Python is an interpreted, interactive, object oriented programming language. Its design focuses on code readability and a clearly defined syntax allows programmer to express in fewer lines of code than C++ or Java. It is portable and compatible with UNIX, Windows and Macintosh. The Python commands are used to design the webpage in order to operate the load through webpage and display status, date, time information [9, 10].

3.2 Arduino IDE

A program written with the Arduino IDE is known as a sketch. The programming code, sketch written with the IDE is in C/ C++ for which source program is easily available. The sketch is compiled and linked with the GNU tool chain, included with the IDE distribution program to convert executable code into a text file in hex. code which is loaded into the Arduino board. The code written using this software operates

the load based on PIR sensor detection signals.

3.3 PuTTY Software

PuTTY is an open emulator, serial console and network file transfer application. It can also connect devices through a serial port. It is a maintain tool but does not process the commands typed into it [11]. PuTTY is used to operate and run the files in the Raspberry Pi placed away from the lighting system.

3.4 Proteus Software

Proteus can be used to draw the circuit for a microcontroller based system and then test it interactively. Meanwhile, ISIS is developed for circuit simulation to make the schematic capture of a microcontroller-based system.

Figure 3 shows the flowchart showing sequence of events that take place in the operation of the prototype automated lighting system for power conservation.

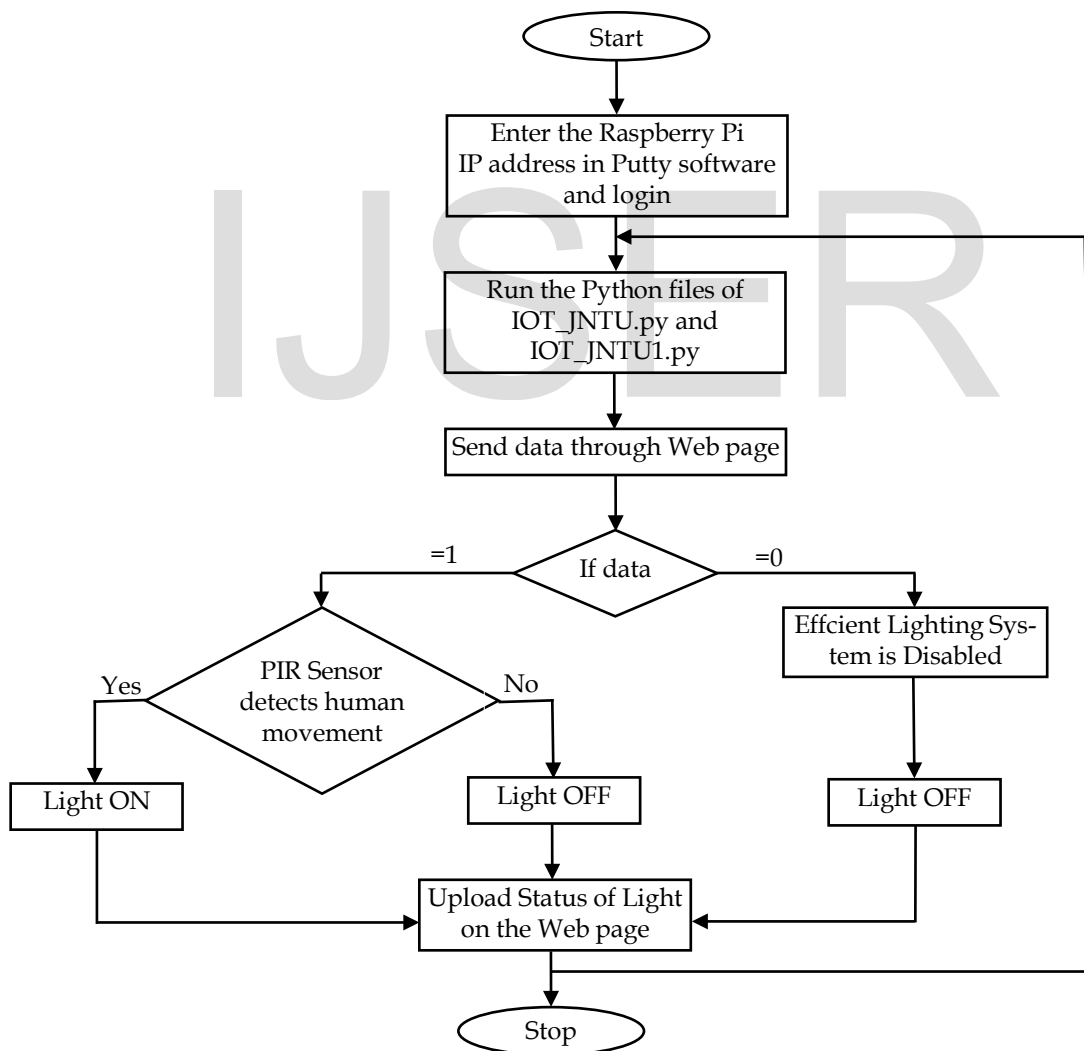


Fig. 3 Flowchart showing sequence of events in the operation of prototype automated lighting system

4 RESULTS AND DISCUSSION

4.1 Schematic Diagram and Experimental Setup

The schematic diagram of the power efficient automated lighting system is developed using Proteus. Only one WSN node is considered in the prototype system for demonstration (Fig. 4). The output of the PIR sensor is connected to digital pin 4 of the Arduino board, which senses the movement of the person. Pin 6 of the Arduino board is connected to the control pin of the relay. The relay is connected to the load (light) to turn it on/off. Transmit (Tx) and receive (Rx) pins of the Arduino are connected to the receive (Rx) and transmit (Tx) pins of the USB

to serial converter, which is connected to ZigBee module. The status of the light is transmitted to Raspberry Pi using ZigBee modules. The Ethernet port of Raspberry Pi is connected to the router.

Figure 5 shows the experimental setup of the power efficient lighting system prototype, which consists of Raspberry Pi, router, two ZigBee transceivers, Arduino board, PIR sensor, a relay, and the load (electric bulb).

The PIR sensor output is fed to the Arduino, which then activates the relay to turn on/off of the load. The status of the light is sent to Raspberry Pi through the ZigBee transceivers, which is uploaded on the webpage along with date and time. The load is enabled through the Raspberry Pi webpage.

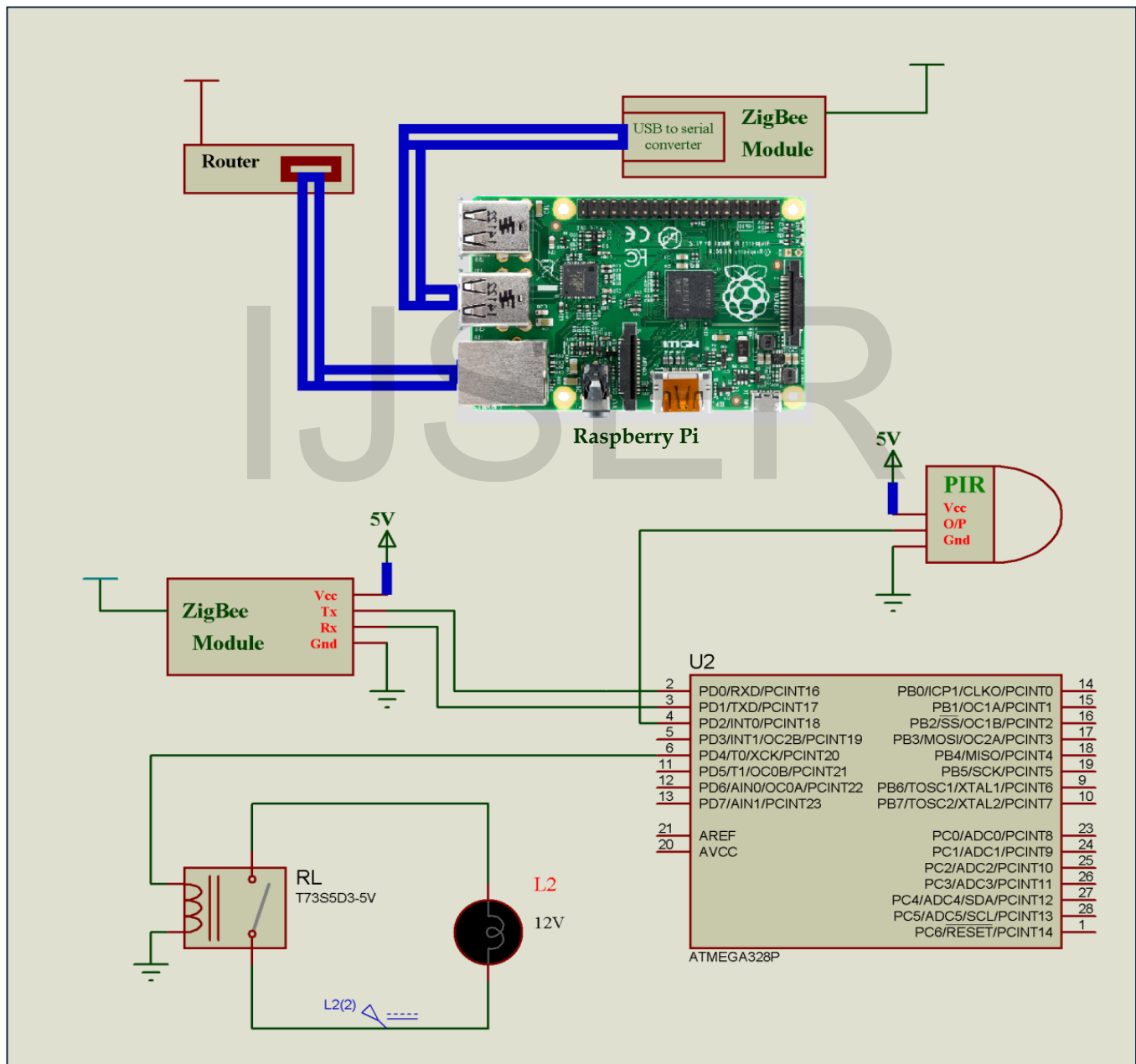


Fig. 4 Schematic Diagram of the Prototype Power Efficient Automated Lighting System

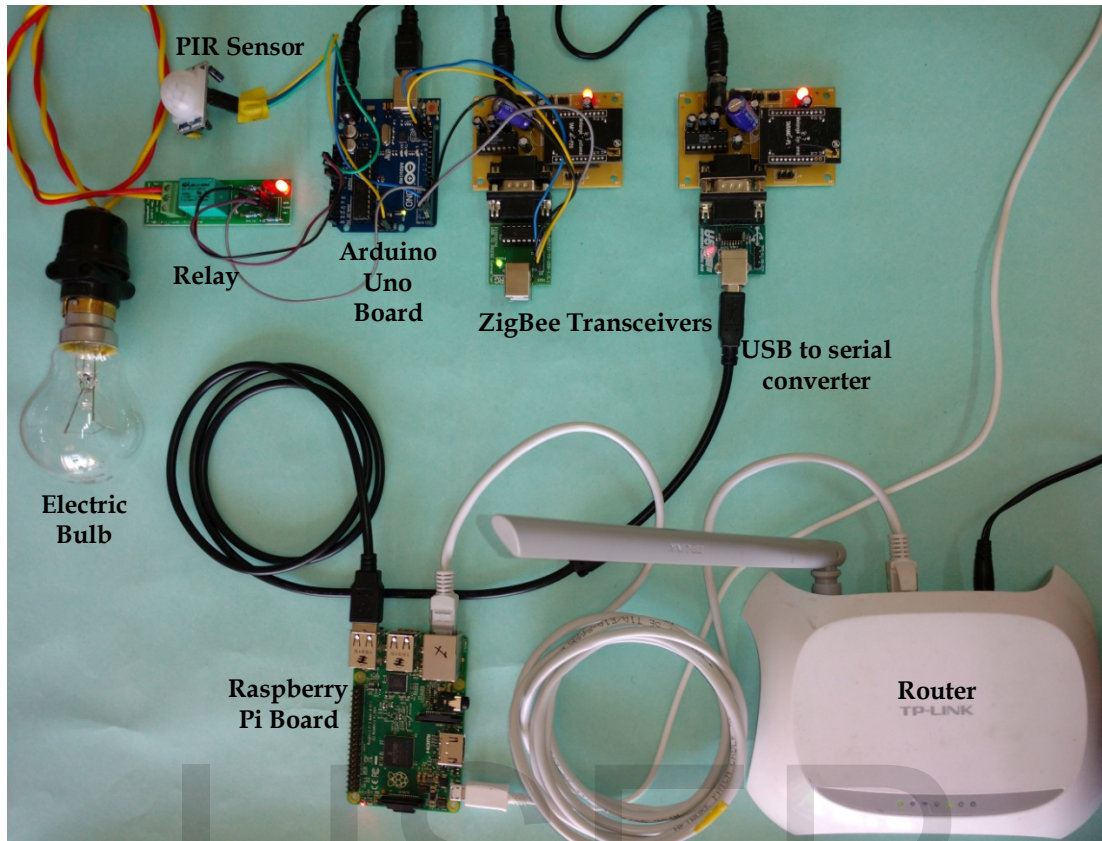


Fig. 5 Experimental Setup of the Prototype Power Efficient Automated Lighting System

For each sensor node added in the WSN model of Fig. 2, additional hardware needed is the Arduino board, a ZigBee transceiver, PIR Sensor, relay and the corresponding load.

Following snapshots depict login process and the outputs:

1. The initialization of Raspberry Pi is done using Putty software by giving the IP address of the Pi (Fig. 6).

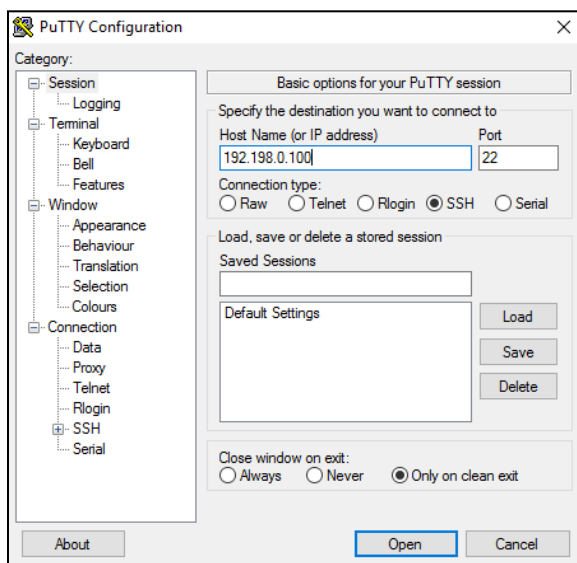


Fig. 6 Initializing Raspberry Pi using PuTTY Software

2. Login to Raspberry Pi contains login id as pi and pass-

word (Fig. 7). After logging in, redirect to the folder where files are stored in Raspberry Pi and open the files by using ls command in Linux. After finding the file, to execute it use nano command nano <file_name>.

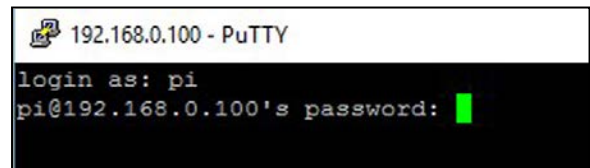


Fig. 7 Logging in to the Raspberry Pi

3. On executing or running python file to enable the intelligent lighting system, the data on webpage is refreshed automatically. The status of the system i.e., whether the system is permitted to turn on or off is displayed on the putty window as shown in Fig. 8

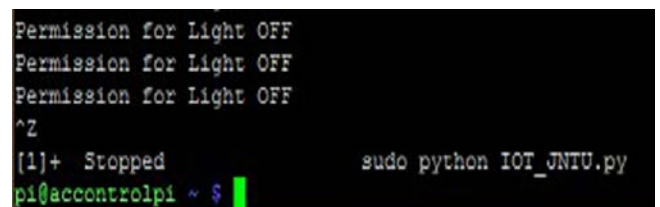


Fig. 8 Snapshot showing running of the python file for permission to turn on or off the light

4. Simultaneously on running python file for uploading the status of light, the status permitted i.e., 'LIGHT ON' or 'LIGHT OFF' can be seen on the putty window as shown in the Fig. 9.



Fig. 9 Snapshot showing running of the python file for uploading the status of light on Web page

5. Simultaneously on running python file for uploading the status of light, the status permitted i.e., 'LIGHT ON' or 'The light is ON when status is '1', and OFF when status is '0' for controlling through web page. The HTML file is run to get web page through which the light can be controlled. After submitting the data 1 or 0 on the web based device control window (Fig. 10), it is stored in a text file and is seen as output of PHP (Fig. 11).

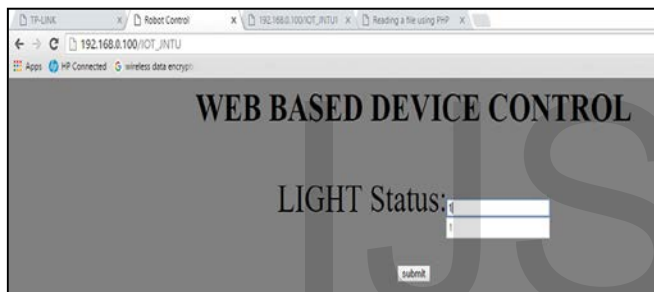


Fig. 10 Snapshot showing the data entered as '1' in web based device control window

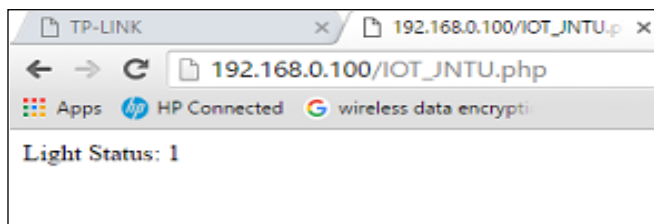


Fig. 11 Snapshot of web page displaying light status as '1'

6. According to the data, the light will be turned ON or OFF and that information is uploaded on the web page along with data and time as shown in the Fig. 12.

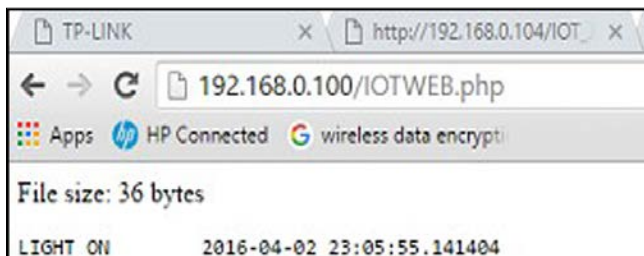


Fig. 12 Snapshot of web page showing uploaded status of light, date, and time

5 CONCLUSIONS

A prototype of an efficient automated lighting system for power conservation is implemented using wireless sensor network approach. The controlling of the system is done remotely through a Raspberry Pi based web server. A PIR sensor used with the node senses the human movement to turn the light on or off. With this approach, there can be a good amount of power saving, hence the costs towards power consumption will fall within affordable cost limits.

REFERENCES

- [1] A.A.Nippun Kumar, G. Kiran, TSB Sudarshan, "Intelligent Lighting System using Wireless Sensor Networks", International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC) Vol.1, No.4, December 2010, pp.17-27.
- [2] <http://www.redcad.org/members/benhalima/azem/toknow.html>
- [3] Vipul Singhvi, Andreas Krause, Carlos Guestrin, James H. Garrett Jr., H. Scott Matthews, "Intelligent Light Control using Sensor Networks" Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys 2005), San Diego, California, USA, Nov. 2-4 2005.
- [4] Z.I. Chowdhury, M.H. Imtiaz, M.M. Azam, M.R.A. Sumi, and N.S. Nur, "Design and Implementation of Pyroelectric Infrared Sensor Based Security System using Microcontroller", Procs. of 2011 IEEE Students' Technology Symposium (TechSym), pp. 1-5, 14-16 January 2011, IIT Kharagpur.
- [5] Hanne Grindvoll, Ovidiu Vermesan, Tracey Crosbie, Roy Bahr, Nashwan Dawood, Gian Marco Revel, "A Wireless Sensor Network for Intelligent Building Energy Management based on Multi Communication Standards - A Case Study", Journal of Information Technology in Construction (ITcon), Vol. 17, 2012, pp. 43-62.
- [6] Eben Upton and Gareth Halfacree, "Raspberry Pi User Guide", Wiley, Sept. 2012.
- [7] Atmel, "ATmega328P Datasheet", Nov. 2015 (http://www.atmel.com/pt/br/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Summary.pdf).
- [8] Shahin Farahani, "ZigBee Wireless Networks and Transceivers", Newnes, First edition, 2008.
- [9] Richard L. Halterman, "Learning to Program with Python", 2011 (<http://www.cs.uky.edu/~keen/115/Haltermanpythonbook.pdf>).
- [10] Jivan S. Parab, Vinod G. Shelake, Rajanish K. Kamat, Gourish M. Naik, "Exploring C for Microcontrollers - A Hands on Approach", Springer, 2007.
- [11] <https://en.wikipedia.org/wiki/PuTTY>.