Energy Preserving Remote Supervised (EPRS) Room

Aastha Alica Das[1], Dr. Neetu Sharma[2]

1M. Tech. Scholar Computer Science Department, Ganga Institute of Technology and Management, (M.D.U.), India

2H.O.D., Computer Science Department, Ganga Institute of Technology and Management, (M.D.U.), India

Abstract- Smart environment represents intelligent automation as the next level of basic automation. The home, traffic, industrial and building automation are called smart when the automation is done by processing contiguous sensor data. Smart environment, like any other sentient system, relies on sensor data as the prime requirement. This sensor data is originated by heterogeneous sensors deployed in the real world. The main challenge of the smart environment is the availability of a contiguous data in the implemented area. This research work proposes a Room Automation System (RAS) named as Energy Preserving Remote Supervised (EPRS) Room, to provide the automated and manual control on electric appliances of the room. The proposed EPRS is implemented in a real environment to save electricity and provide a remote control on the room environment.

Keywords- Energy preserving remote supervised room, Room Automation System, Internet of Things, Wireless Sensor Network, Arduino Uno, Adhoc On-Demand Distance Vector, Open source Home Automation.

I. INTRODUCTION

There is a huge consumption of electricity worldwide, this consumption is increasing every day roughly in exponent pattern. Countless energy efficient systems are reducing energy consumption by applying automation. Among these energy efficient systems, IoT offers some of the best solutions like smart city, smart building, smart home, smart industry etc. The smart environment is the automation of objects in a particular area; automation for improved comfort, for increased security and for energy efficiency. Numerous automation systems are designed and developed for energy efficiency in the home and building environment. Most of these automation systems provide remote control to the home/building through the Local Area Network (LAN) or Personal Area Network (PAN). Related works on home automation system are saving electricity and improving the security of the home as well as improving comfort. Room Automation System (RAS) is a part of the smart home that is responsible for creating the smart environment in a room [1]. A similar system based on room automation is proposed in this thesis. The proposed concept of EPRS Room (Energy Preserving Remote Supervised Room) provides the automation environment to the room in order to improve comfort and energy efficiency [2]. The EPRS provides automation of room appliances according to the real environment needs and serves a GUI based web application to give the control of the room at the geographically remote location [3]. The system is saving electricity by 45% as the result of implementation in the room and provides the remote accessibility to the user. The comparison between EPRS and other Room automation system concludes that the EPRS Room is a better RAS in some aspects such as providing remote access, controlling appliances, the frequency of updating data on the cloud, and saving the electricity in the room [4]. For the need of contiguous data from distributed sensor nodes, Wireless Sensor Network (WSN) is formed. WSN receives the data from all the sensor nodes and further process this data for analytics and applications, according to the requirement [5]. The smart environment is an implementation of Internet of Things (IoT), IoT is a networked interconnection of quotidian objects. IoT is an emerging technology, reforming the living standard of people and decreasing operational expenses [6]. IoT reduces the operational expenses by the automation of real-world objects such as electric appliances for the reduction in electricity consumption. There are many applications of IoT such as smart city, smart home, smart traffic, health monitoring, etc. Smart home basically focuses on the reduction of in-home energy consumption [7]. Similarly, various automation systems are designed to minimize the energy consumption in a room. The EPRS consists of WSN, a microcomputer and a web application [8]. Status of the room environment condition is required for the realization of room automation, WSN is responsible for the generation and transmission of this data [9]. The microcomputer processes the data provided by the WSN and also responsible for the automation of room. The web application has a user interface, providing a graphical representation of room environment and the remote control to the administrator [10]. The EPRS reduced electricity consumption by 45% in room and provides remote access to visualize the Real-time sensor data and switch control [11]. Moreover, a comparison with other similar RASs manifests that the EPRS is better RAS in terms of providing remote access and also it is better in reducing electricity consumption.

The commitments of this paper are:
1) Design the architecture for EPRS Room Automation System.
2) Design the system to be cost-efficient and simple.
3) Providing control over the room from the remote location.
4) Emulation of the room automation system with available resources.
5) Comparison of proposed work with existing systems.

II. DESIGN AND ARCHITECTURE

For the detection and prevention of possible errors and faults before implementation of EPRS, the prototype is designed. The same methodology is used for the prototyping of EPRS in small-scale. Limited hardware and software resources are used to develop the prototype to minimize the loss in case of any failure occurred. The EPRS Room prototype is equipped with a microcomputer, a relay board, and two sensor nodes.

2.1. Hardware Components

EPRS consists of three levels of hardware input, processing and output. Input is provided by the sensors, microcomputer processes the input data, and the output is realized by the relay in real environment as shown in Figure 1.

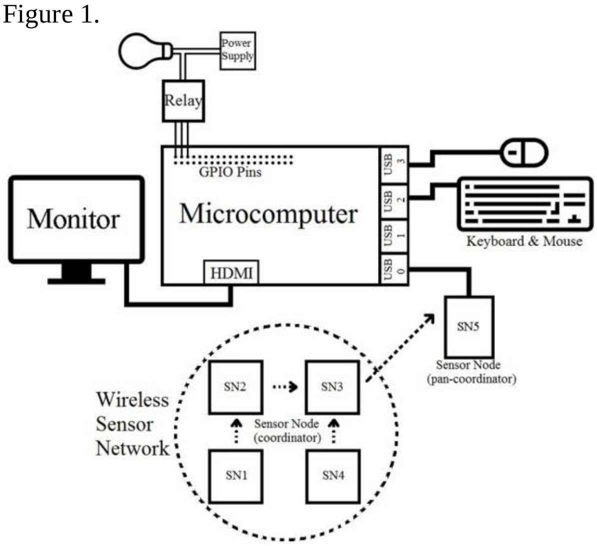

Fig.1: Hardware used in EPRS Room

2.1.1. Sensor and Microcontroller: The sensor is a tiny node which is used to detect and record the changes in environmental conditions. Arduino Uno module is embedded with sensors and a microcontroller. Arduino Uno is a stacked module of sensors, gateway, and a microcontroller. This module supports IEEE 802.15.4 XBee that makes it a better option for WSN because of the less energy consumption [12].

2.1.2. Microcomputer: A small sized, fully functional and inexpensive computer became popular worldwide as MicroGear NetPie control [13]. It is used in EPRS as it can handle 28 relays at a time and the requirement here is to control 20 appliances [14].

2.1.3. Relay Board: Relay is an electromagnetic device works as a switch to electric appliances. Microcomputer gives signal to relay and on the other side it is connected to the electric appliance that is controlled according to the signal provided by the microcomputer [15].

2.2. Software Components

2.2.1. C Language: C is considered as the most suitable programming language for embedded systems. In EPRS C language is used to program the microcontroller attached with the sensors insures the working of sensors and the data transmission in WSN [16]. The functions used in C program are as follows:

2.2.1.1. relay_action(): This function carries the relay control

2.2.1.2. save_data(): This function is used to save eeprom.

2.2.1.3. setup(): This function set up all the activities related to led.

2.2.1.4. blinkcode (): Simply loop a few times flashing the status light (this is used during boot up).

2.2.1.5. fastblinkcode (): Simply loop a few more times flashing the status light (this is used during boot up).

2.2.2. PHP: Hypertext Preprocessor is the most used open source general scripting language that is basically suited for web development and can be embedded into HTML.

2.2.3. SQLite: It is a very simple RDBMS since it doesn't need to run any separate server. It is also possible to make a prototype using SQLite and then porting the code to a bigger database like Oracle or PostgreSQL [18].

2.3. Architecture

There are three layers in the architecture of EPRS. Perception layer consists of the hardware installed in real environment such as sensors, microcomputer and relays. Network layer consists of communication mediums such as XBee, Wi-Fi, LAN and Internet. Application layer consists of the output generated by the working of whole system such as automation and web application.

2.4. Prototype of EPRS

Before developing the EPRS a prototype of this system is designed to detect and prevent possible errors in small-scale. The prototype contains two sensors one for sensing and transmitting the data, another for sending this data to the serial port. This system has a small-scale database and a simple web interface for Real-time data display and switch control. The prototype includes all the technologies used in the actual model of EPRS. The prototype has a

WSN, serial port communication and GPIO pin control same as the EPRS.

III. METHODOLOGY

After the successful implementation of the prototype, it's testing and debugging the actual EPRS is developed. This system is more complex than that of the prototype. The development involves a circuit system of connection between more than 20 hardware, complex programming of the WSN consisting AODV routing protocol and a web application with better User Interface.

3.1. Hardware Setup

EPRS is equipped with very complex hardware circuit design including five nodes in WSN wirelessly connected to each other using XBee, pan-coordinator being head of the WSN is receiving data from all the coordinator nodes and it is associated with the microcomputer through the USB. A microcomputer as a central hub takes control over everything and behaves like a central processing unit for EPRS. There is one relay boards of size four relays that are separately connected to the AC switches of the room. The circuit diagram of EPRS is shown in figure 2.

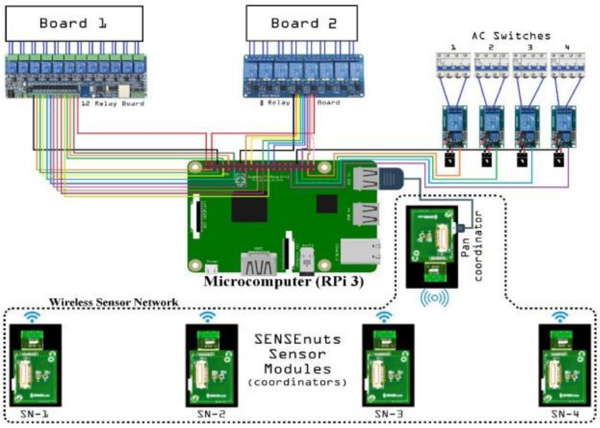

Fig.2: Circuit Diagram of EPRS Room

3.2. Software Setup

Software setup includes programming WSN, serial port communication, web application design and GPIO pin control.

3.2.1. Wireless Sensor Network: In WSN the coordinator and pan-coordinator nodes are programmed.

3.2.2. Serial Port Communication: The pan-coordinator node forwards the sensor data to the microcomputer. This communication between USB gateway of sensor module and the microcomputer is performed by the serial port.

3.2.3. Switch Control by GPIO: The microcomputer used in EPRS has inbuilt GPIO pins. The GPIO pin holds the binary signal as 1 or 0. The relay is switched on/off by sending the signal to the associated GPIO pin. Below pseudo code shows how the program is controlling the pins in order to control the appliances.

```
#include <LiquidCrystal.h>
#include <EEPROM.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
//Initialize Device
void relay_action()
{
//Relay control
 if(EEPROM.read(11)==1)
   {
    digitalWrite(R1,LOW);
    lcd.setCursor(0, 2);
    lcd.print("ON  ");
   }
  else if(EEPROM.read(11)==0)
   {
    digitalWrite(R1,HIGH);
    lcd.setCursor(0, 2);
    lcd.print("OFF ");
   }
  if(EEPROM.read(12)==1)
   {
    digitalWrite(R2,LOW);
    lcd.setCursor(4, 2);
    lcd.print("ON  ");
   }
  else if(EEPROM.read(12)==0)
   {
    digitalWrite(R2,HIGH);
    lcd.setCursor(4, 2);
    lcd.print("OFF ");
   }
  if(EEPROM.read(13)==1)
   {
    digitalWrite(R3,LOW);
    lcd.setCursor(8, 2);
    lcd.print("ON  ");
   }
  else if(EEPROM.read(13)==0)
   {
    digitalWrite(R3,HIGH);
    lcd.setCursor(8, 2);
    lcd.print("OFF ");
   }
  if(EEPROM.read(14)==1)
   {
    digitalWrite(R4,LOW);
    lcd.setCursor(12, 2);
    lcd.print("ON  ");
   }
  else if(EEPROM.read(14)==0)
   {
    digitalWrite(R4,HIGH);
    lcd.setCursor(12, 2);
    lcd.print("OFF ");
```

```
  }
}
```

3.2.4. Database: The database is created using SQLite. Table in the database is aastha_relay_control. The aastha_relay_control stores the switch data from the web application of EPRS and stores the sensor data with sensor id and timestamp. The communication flow in the EPRS room from the database is shown in figure 3.

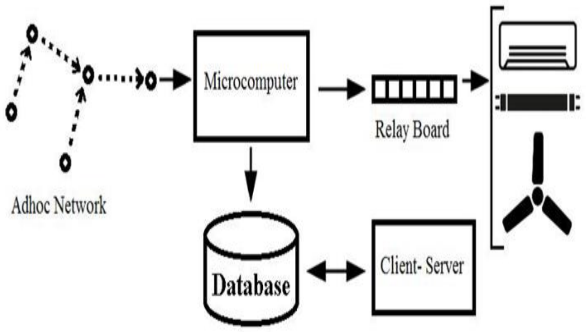

Fig.3: Communication flow in the EPRS Room

IV. RESULT AND DISCUSSION

The main goal of the EPRS Room was to save electricity in public sector buildings and provide remote control to the administrator. After its installation in the room, it started controlling electric appliances smartly, which results in saving in the wastage of electricity. After the calculations on existing data, it is concluded that the EPRS Room is saving electricity as well as the administration cost for everyday use of this room. The room has 10 fluorescent tube lights, 3 fans and 2 Air conditioners are sucking electricity uselessly even when there is no need of light and ACs at some corners of the room.

4.1. Remote control

The web application of EPRS provides a user interface as shown in Figure 4. This web application displays the Real-time sensor data and control switches to handle the room environment.

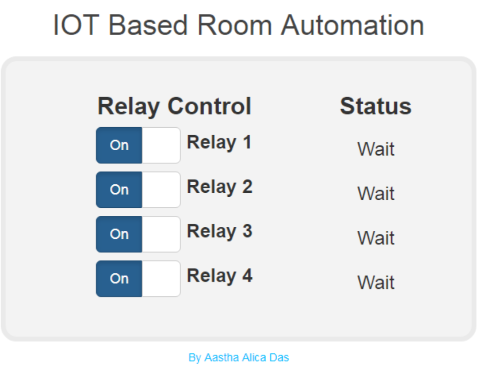

Fig.4: User Interface of the EPRS Room Web Application

4.2. Power Consumption

A comparison of electricity usage is also performed in which the total usage of electricity in the room before and after the implementation of automation is calculated. According to Table 1, it is concluded that a total of approximately 12500 units of electricity is saved annually by the implementation of EPRS Room in one room. The electricity usage is reduced to about 45% from the normal room, when fully automated. By minimizing electricity consumption EPRS Room is indirectly reducing CO2 emission by 2.8 tons annually.

Table 1 Electricity saving in room

Number of Appliances	Normal Room usage (per week)	EPRS Room usage (per week)	Total Saving (per week)
3 Fans (70 W each)	11 Units	7 Units	4 Units
10 Tube-lights (80 W each)	36 Units	20 Units	16 Units
2 Air Conditioners (3.9 KW each)	491 Units	273 Units	218 Units
Total Electricity usage	538 Units	300 Units	238 Units
Charges (₹7 per KWh)	₹3766 ≈ $56	₹2100 ≈ $31	₹1666 ≈ $25

V. CONCLUSION

This paper presented the EPRS Room architecture as a small-scale RAS, the role of IoT for energy efficiency, and the comparison with existing RASs. Various technologies and hardware such as C programming, PHP, microcomputer, sensors, and relays are used in the development of EPRS. Being a basic automation system EPRS Room has the possibility of many more improvements. The work can further be improved by using PIR Motion sensors and GPS positioning. Since these sensors are of very small memory and processing capacity the security is a very important aspect of this automation. Security of whole automation system can be improved. Another work that is possible after this project is the use of machine learning which can more smartly control the appliances. Since the data of a long time is available in the database that data can further be processed to identify a pattern in the usage of the room, this pattern will help to improve the system. Open source Home Automation can also be developed so that a user can design his own type of Smart Home. Since now every time a Smart Home is designed from bottom to top, open sourcing will make it use just like a plug and play device with a little use of programming. In short, the EPRS has many possibilities for future research and project works.

More possible research works on EPRS Room are shown in Figure 5.

VI. ACKNOWLEDGEMENT

I thank GOD almighty for guiding me throughout this research. I would like to thank our Head of the Department, Dr. Neetu Sharma for all help, support and cooperation extended to me. I would like to thank my parents without whose blessings, I would have not been able to accomplish my goal. I would like to thank all those who have contributed to completion of this research work.

Security	Communication	Management	Architecture	Implementation
*Improvement in security of WSN *Authentication in web app	Replacement of serial port communication	Providing Administrator window for upper level control	Improvement in existing architecture of Smart Lab	Open sourcing of smart lab

Fig.5: Suggestions for future research work

VII. REFERENCES

[1]. Vibhuti and S. L. Shimi, "Implementation of smart class room using WAGO PLC," *2018 2nd Int. Conf. Inven. Syst. Control*, no. ICISC, pp. 807–812, 2018.

[2]. M. Singh and S. L. Shimi, "Implementation of Room Automation with Cloud Based Monitoring System," *2018 2nd Int. Conf. Inven. Syst. Control*, no. Icisc, pp. 717–813, 2018.

[3]. A. Dementjev and K. Kabitzsch, "A CONSULTING MODULE IN ROOM AUTOMATION Alexander Dementjev, Klaus Kabitzsch," 2001.

[4]. A. C. Oezluek, H. Dibowski, and K. Kabitzsch, "Automated Design of Room Automation Systems by using an Evolutionary Optimization Method," 2009.

[5]. C. Sarkar, A. U. Akshay, R. V. Prasad, A. Rahim, R. Neisse, and G. Baldini, "DIAT: A scalable distributed architecture for IoT," *IEEE Internet Things J.*, vol. 2, no. 3, pp. 230–239, 2015.

[6]. P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, "Convergence of MANET and WSN in IoT urban scenarios," *IEEE Sens. J.*, vol. 13, no. 10, pp. 3558–3567, 2013.

[7]. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[8]. "Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020." [Online]. Available: https://www.gartner.com/newsroom/id/2636073.

[9]. J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, 2014.

[10]. S. Agrawal, "A survey on Internet of Things," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347--2376, 2015.

[11]. A. Science, "Integration of Cyber-Physical Systems for Smart Homes," no. December, 2014.

[12]. E. Technologies, "SENSEnuts Hardware Guide." Sensenuts University.

[13]. "Raspberry Pi 3 Model B+" https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf

[14]. "Usage - Raspberry Pi Documentation." http://www.raspberrypi.org/documentation/usage/

[15]. Georgi Dalakov, "The electromechanical relay of Joseph Henry." [Online]. Available: http://history-computer.com/ModernComputer/Basis/relay.html.

[16]. E. Technologies, "API Reference Guide." Sensenuts University.

[17]. B. Simon Monk and B. Simon, "Programming the Raspberry Pi: Getting Started with Python," **(2012)**.

[18]. M. Owens, "Introducing SQLite," in The Definitive Guide to SQLite, Apress, **(2006)**, pp. 1–16.