WIRELESS HOME AUTOMATION SYSTEM VIA RASPBERRY PI USING RUBY ON RAILS AND REACT-NATIVE

Rishendra Ravi , Divisha Chaturvedi , Nakul K. Warrier , Antra Mondal

Computer Science And Engineering Department, SRM University, Tamil Nadu, India

ABSTRACT

Today's fast growing technology is slowly drifting towards automation. The generation wants to control all machinery and its processes. Home automation technology provides with an efficient system to control everyday home appliances. Our project deals with Wi-Fi technology, since the range is better than Bluetooth and ZigBee technologies. This project uses Raspberry Pi module along with Rails web framework in collaboration with react native. The backend is a standard Rails app exposing some API endpoints with JSON output . The frontend is an Express app running on Node.js, which uses React and talks to the Rails API. Rails is a server-side web application framework written in Ruby under the MIT License. Rails is a model–view–controller (MVC) framework, providing default structures for a database, a web service, and web pages. The system is proposed to create a smart environment by switching on and off the target devices by Wireless-Fidelity (Wi-Fi). The aim of our project is to have remote access to home appliances and thereby save electric power and human energy. This project uses Internet of Things to control basic home functions automatically through a computer or mobile device. In this paper we present a Wireless Home Automation System(WHAS) built using Raspberry Pi, Rails supported backend and React supported User Interface to help set up wireless communication, to provide the user with control of various appliances within the home. Lights, air conditioners, electronic doors and fans are among the appliances that can be used in this system. The control can be done remotely or when connected to the same network as the appliances. This system increases the safeness of the house and electricity conservation by providing monitoring system and improve the quality of domestic life by using the control buttons of targeted device. This is a low cost system and is expandable allowing further development.

Keyword:- *Raspberry Pi, Rails, React, Wi-Fi, Android Application, Automation*

1. INTRODUCTION TO THE PROJECT

The proposed system will be implementing a wireless remote control solution for controlling the lights, fans and other electronic devices via Wi-Fi enabled device such as smartphone, using an Android Application, adding convenience and also encouraging electricity conservation. Since, the concept is not new, and we already have an existing system with more complexity, our prime motto is to reduce the complexity and price of the available system. In this project, appliances such as lights and fans that are connected to the main control unit, can be controlled remotely from the smartphone screen. This is implemented using a very simple Graphical User Interface of an Android Application, which can easily be understood by non technical users. This system is supported by Ruby on Rails, which is easy to implement with Raspberry Pi and a React supported Android Application, which is faster for implementation, than other coding options.

2. RELATED WORKS IN EXISTING SYSTEM

R. A. Ramlee[3] introduced a smart home system using Bluetooth technology which focussed on easing the daily activities of disabled people confirming safety and comfortability. ZigBee controlled home automation system via Internet or a remote control was introduced in a research by G. Khusvinder[4]. The system implementing Python Tkninter supported GUI with switch-on and switch-off icons to control appliances was introduced by M. Mahadi Abdul Jamil and M. Shukri Ahmad[1]. Mohd Nor Azni[2] introduced the use of Android Application supported with Python Django framework for enabling a home automation system.

3. SYSTEM OVERVIEW

Figure 1 shows the system architecture that consists of a Wi-Fi enabled server smartphone, connected to the main control unit consisting of Raspberry Pi, through an Android Application. Raspberry Pi contains all the necessary ports and features of a common PC or laptop embedded in it, such as 802.11n WiFi, Bluetooth 4.0, and a quad-core 64-bit ARM Cortex A53 running at 1.2 GHz, 40 input/output ports(GPIO), etc. The USB port is connected to a wi-fi dongle to add connectivity. Further, the GPIO are connected to the input output interface which contains the switching circuit and power ports.

First, the Ubuntu Mate OS is mounted on the memory card and it is loaded on Raspberry Pi and then connected to the screen using HDMI cable. The OS is installed and is setup with wi-fi using the dongle. Then the relay board is connected to GPIO of the device. Then, Ruby and Rails is installed on the device and backend is coded for the application. Figure 2 is a screenshot of OS after installation over Raspberry Pi.The backend is a standard Rails app exposing some API endpoints with JSON output . The frontend is an Express app running on Node.js, which uses React and talks to the Rails API. Rails is a server-side web application framework written in Ruby under the MIT License. Rails is a model–view–controller (MVC) framework, providing default structures for a database, a web service, and web pages. Figure 3 shows the MVC Web Framework.

The GUI is designed using React Native, which is easier and faster for Android Applications. The Android Application acts as a medium between the user and the main control unit of the system. The application can be operated over wi-fi and consists of switch on and off icons, which when tapped over, can operate the respective lights and fans.

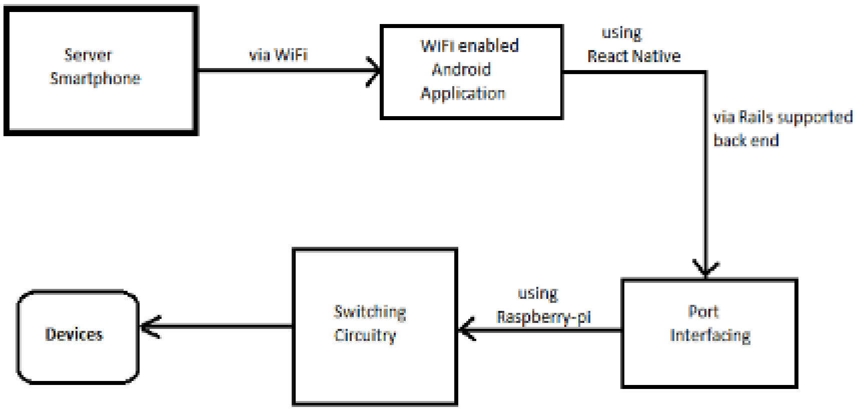


Figure.1 - System Architecture Diagram

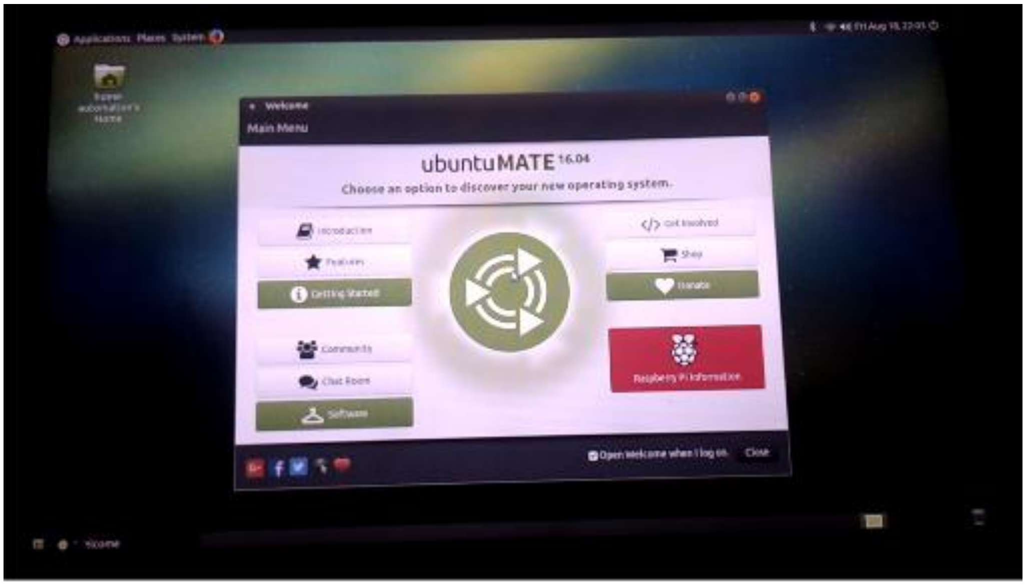


Figure.2 - Raspberry Pi after installing Ubuntu Mate OS

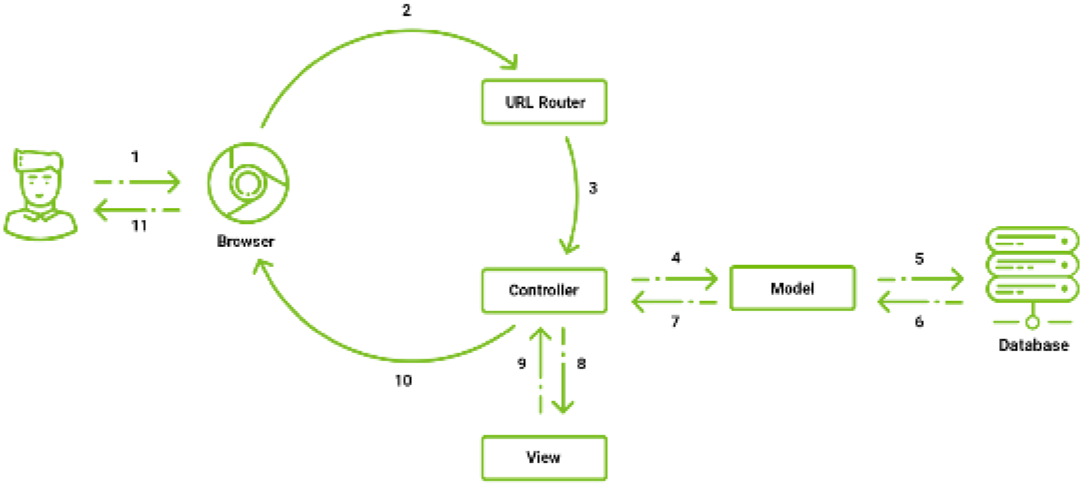


Figure.3 - MVC Web Framework

4. METHODOLOGY

Figure 5 shows the data flow diagram of the system programming for the process to turn on and off the output device using the GUI of Android application. The user accesses the Android application and taps on the required icon to command a signal through Wi-Fi module, which forwards an electromagnetic signal to be processed by the Router. The signal is further processed by Raspberry Pi which converts it to an electrical signal and the switching action is initiated. Thus, the appliances are controlled using a turn on and off mechanism.

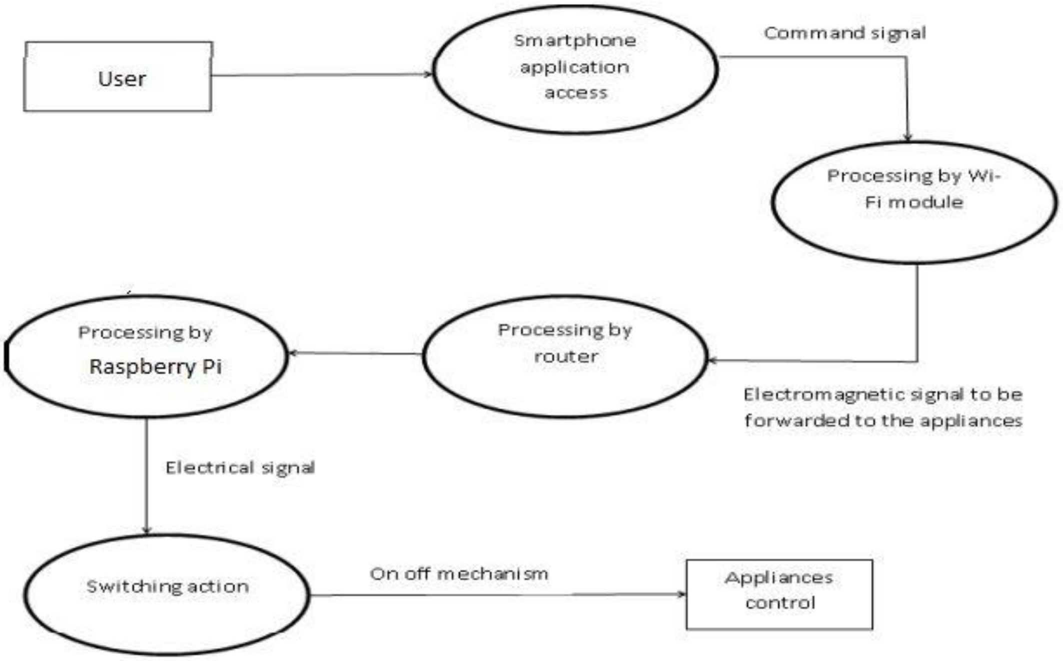

User
Smartphone application access
Command signal
Processing by Wi-Fi module
Processing by router
Electromagnetic signal to be forwarded to the appliances
Processing by Raspberry Pi
Electrical signal
Switching action
On off mechanism
Appliances control

Figure.4 - Data Flow Diagram

5. RESULT AND ANALYSIS

This section will discuss the results for the implemented system as well as analysis of the outcomes. The results of this project are divided into two parts, ie , the hardware result and the graphical user interface result.

5.1 Hardware Result

For testing the functionality of the system, the main control system of the prototype were connected to a light and a fan. The input output interface consists of 5V SPDT relay switching circuit to control the 240V output AC, along with Bc547 transistor, 1N4001 diode and a 9V battery, used for powering up the device and demonstration purposes. The GPIO pins are used to connect Raspberry Pi to the Input output Interface and the obtained results were recorded as given in Table 1.

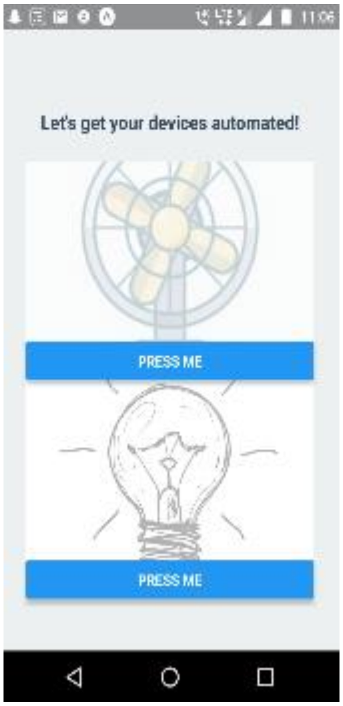
Input	**Button**	**Output**
Light	Press Me(Light icon)	The light changes its state from ON to OFF or ON to OFF, depending

		on its previous state.
Fan	Press Me(Fan icon)	Fan activates or deactivates, depending on its previous state.
Exit from GUI	Back button of the Smartphone	Deactivates the GUI Display

Table.1 - Input Output Interface Results

5.2 Graphical User Interface Result

Figure 5 shows the interface that will be displayed on the smartphone when the user accesses the Home Automation application. The frontend resides on the Raspberry Pi. The advantage of this is, regardless of what smartphone the user has, the GUI is always the same. The application has been coded using React Native, which when collabed with the system, operates faster. More than one user can access the application, at any given time. The devices can also be operated remotely using this application.



Let's get your devices automated!

Figure.5 - Implementation of UI design

6. DISCUSSION

The proposed solution for home automation, which is, allowing a user to control home appliances such as lights and fans through a smartphone application, having tap buttons for on and off controls, has been demonstrated to be functioning properly. With the growth of internet based and related technologies, and support of modern frameworks, the home automation system looks feasible to the current arena. Proper efforts in such direction will encourage the realization of a fully automated home, which will certainly prove a benefit for elderly and disabled people along with an encouragement for electricity conservation.

7. CONCLUSION

Our project demonstrates the possibility of implementing a faster and reliable system that helps the elderly and disabled people to access the control of lights and fans of their home using a smartphone application, by tapping an icon given in the application.

This system is faster and reliable, since the backend is Ruby and Rails supported. In all the home automation systems that are available for use, most of the raspberry pi device backend is coded with python language. Although python is the easier language to use, It's Django web framework has some flaws. In Django, the developer is expected to explicitly define routes, urls, and field values. This the reason we have gone with Rails web framework. It is much simpler and easier to use and handle. Also, it's fast and reliable.

Furthermore, addition of sensors, magnetic door locks and alarms will enhance the scope of this system even more. Finally, this project will provide a flexible and customizable design as well as low cost implementation for a number of appliances.

8. REFERENCES

[1]. M. Mahadi Abdul Jamil, M. Shukri Ahmad, "A Pilot Study: Development of Home Automation System via Raspberry Pi ", Biomedical Modelling & Simulation Research Group Universiti Tun Hussein Onn Malaysia, Johor, Malaysia.

[2]. Mohd Nor Azni, "Home Automation System with Android Application", School of Computer and Communication Engineering, Universiti Malaysia Perlis, Perlis, Malaysia.

[3]. R. A. Ramlee, D.H.Z. Tang dan M.M.Ismail(2012), "Smart Home System for Disabled People Via Wireless Bluetooth", IEEE International Conference on System Engineering and Technology, pp.978-983, 2012

[4]. G. Khusvinder, Y. Shuang-Hua, F. Yao and X. Lu, "A ZigBee-Based Home Automation System", IEEE International Conference on Consumer Electronics, pp 422-430, 2009

[5]."GUIwithRaspberryPiTkinterandGPIO",Retrievedon26October2013.Website:http://www.youtube.com/watch?v =xfxSH790rZ0