IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 09, Issue 3 (March. 2019), ||S (III) || PP 18-28

Design and Implementation of Web Based Smart Home Control System using Raspberry Pi

Dhiraj Sunehra
Department of Electronics & Communication Engineering, Jawaharlal Nehru Technological University Hyderabad, Telangana State, India
Corresponding Author: Dhiraj Sunehra

Abstract: The appliances used in our home operate using different control mechanisms. Some are controlled using an IR remote controller and some are operated manually using switches. To implement a smart home control system, this paper aims at development of a web based remote control to control the appliances in the home. The design involves decoding of IR remote patterns of the appliances. The Raspberry Pi3 microcontroller is used as a web server. When a particular button is pressed on the developed webpage, the relevant code is transmitted through a IR transmitter to perform the desired control operation on the device or home appliance. A media player remote is decoded here for experimentation. Lights and fans in the home can also be operated through the same webpage. To improve the home safety and security, a smoke sensor, passive infrared (PIR) sensor and a webcam are used. ZigBee is used for wireless communication. The webcam can be accessed from anywhere to check for any intrusion in the home. An email alert is sent to the user in case of intrusion. The software code is written in Python language. HTML and PHP are used for development of webpage. In the proposed smart home control system, home appliances can be controlled using an internet enabled device such as a smart phone, and can replace the individual remote controls of the appliances. The system facilitates elderly and physically disabled people to operate the appliances conveniently without actually going to the switch board or home appliance.

Keywords: Smart Home, Appliance Control, Intrusion detection, IR Remote, Raspberry Pi Web server, Python language

---

Date of Submission: 06-03-2019 Date of acceptance: 25-03-2019

I. INTRODUCTION

Smart home refers to the use of computers and information technology for control of home appliances. Its application varies from control of simple devices like light or fan to complex devices like TV, air conditioner, music player, etc. The smart home system uses wireless communication standards such as ZigBee, Bluetooth, Wi-Fi and GSM to communicate with various devices. In today's modern world, many appliances are used in home and offices for different purposes. These appliances operate using different control mechanisms. Some of the appliances may be operated using an IR remote control and the others operated manually using wired switches. With increasing number of appliances in the home, the number of remote controls may also increase. Due to this, we may sometimes find it difficult to locate the remote controllers. It will be easy and convenient, if we can control all the appliances through a webpage operated from any internet enabled device such as a smart phone. These systems are becoming popular worldwide and are referred to as home automation or smart home systems. Smart homes will become inevitable in the near future. Using the smart home automation, we can also control the devices when we are not at home. These systems are very useful for elderly people, as they can operate the lights, fans and other appliances without actually going to the switch boards.

II. LITERATURE REVIEW

Hsiao et al developed a webpage containing TV remote buttons to operate the TV using a smart phone. ZigBee and Wi-Fi protocols are used for wireless communication [1]. Ranga Sai et al developed a GUI based mobile application to control home appliances using a microcontroller, IR transmitter and Bluetooth [2]. Sunehra and Veena implemented an Email and Bluetooth based home automation system (HAS) using Raspberry Pi [3]. This system notifies the user about turning on or off the load in the subject of email. ElKamchouchi and ElShafee have implemented a HAS using SMS service for remote monitor and control of home appliances [4]. This system also includes security features to accept commands sent by pre-selected users. Tharanya and Sangeetha presented a speech recognition based HAS consisting of a web server application and

an android application [5]. The android application is used as an interface to transmit the user commands to the web server, which interprets and takes necessary action. Van Der Werff et al have used a java supported phone, a modem and a local server for home automation [6]. The user commands are given through mobile phone via cellular modem to the home server for appliances control.

III. SYSTEM DESCRIPTION

The block diagram of the proposed web based smart home control system is shown in Fig. 1 and Fig. 2. It consists of two units, *viz.* Raspberry Pi web server and In-Home device control unit. The web server includes Raspberry Pi, a web camera, a ZigBee transceiver module and the smart home control web page. By using any internet enabled device such as a smart phone or PC, user first selects the desired home appliance on the web page. Then he selects the button corresponding to the desired control operation. Whenever a button is pressed on the webpage, a particular pre-defined code assigned to it will be transmitted by the Raspberry Pi to the Arduino based In-Home device control and safety unit *via* ZigBee modules.

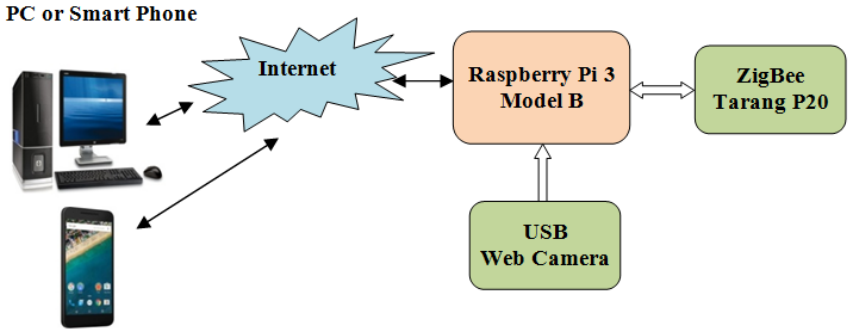

Figure 1: *Raspberry Pi Web Server Unit*

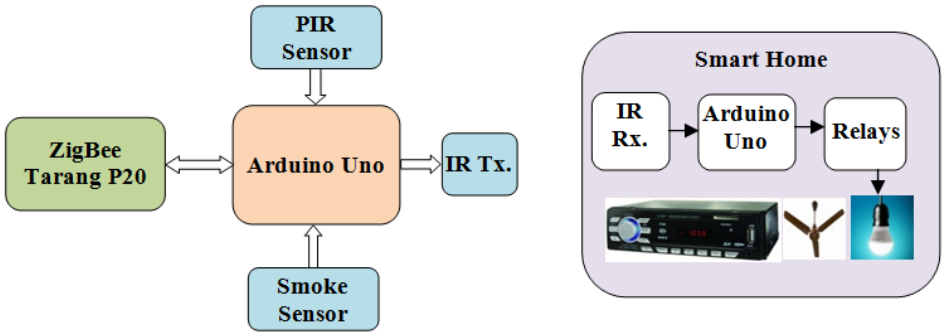

Figure 2: *In-Home Device Control Unit*

The In-Home device control unit consists of Arduino Uno board, IR transmitter and receiver, ZigBee transceiver module, relays and various loads. The Arduino receives the code from the ZigBee modules and then transmits the relevant code through the IR transmitter to perform the desired appliance control operation. In this experiment, a media player, bulb and fan are used for demonstration. To improve the security and safety standards of the home, additional sensors including PIR sensor and smoke sensor are incorporated in the In-Home device control unit. A PIR sensor is used to detect intrusion when there is nobody at home. A smoke sensor is used to detect occurrence of any fire accident in the home. The web page notifies whenever there is any occurrence of intrusion or fire accident. An email alert is also sent to the user through the Raspberry Pi web server. A web camera is interfaced with the Raspberry Pi to provide video streaming of the home environment in real time.

The salient features of various hardware modules used in the implementation of smart home control system are described here.

a. Salient Features of Raspberry Pi 3 Model B Board

Raspberry Pi3 is a low cost minicomputer. It consists of a Broadcom BCM2837 Cortex A53 processor operating at 1.2 GHz frequency (Fig. 3). It has a RAM of 1GB, a micro SD card slot, 40 pin header consisting of 26 GPIO pins, camera and display serial interfaces, a HDMI port, 100 Mbps Ethernet port and a 3.5 mm audio jack [7].

Figure 3: *Raspberry Pi3 Model B Board*

b. Salient Features of Arduino Uno Board

Arduino Uno is a small and widely used microcontroller board consisting of ATmega328P 8-bit reduced instruction set (RISC) controller (Fig. 4). It is an open source hardware and software. The Uno board consists of 14 digital pins used for I/O, six analog pins (A0-A5) with 10-bit ADC and a 16 MHz quartz crystal. It has 2KB of SRAM, 32KB of on-chip flash memory and 1 KB of EEPROM. The recommended supply voltage is 7-12 V [8].

Figure 4: *Arduino Uno Board*

c. MQ135 Smoke Sensor

The smoke sensor MQ-135 has four pins, *viz.* V_{cc}, GND, analog output (A_o) and digital output (D_o) (Fig. 5). The output from the smoke sensor provides both analog and digital voltage levels. As smoke level increases, the output voltage increases.

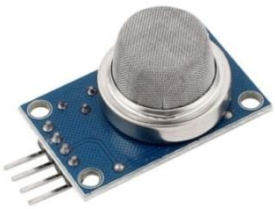

Figure 5: *Smoke Sensor MQ 135*

d. Passive Infrared Sensor

Passive Infrared (PIR) Sensor works on the heat sensing principle, and consists of two segments to detect the change in infrared radiation (Fig. 6). When a warm body such as a human being passes near by the sensor, and is within the range of about 10 m, ±15° view angle, it results in a differential change in the voltage pattern across the sensor and is used to detect the motion of human beings in the room [9].

Figure 6: *PIR Sensor*

e. IR Transmitter and Receiver

An IR LED is used as an IR transmitter. IR remote control is used in many appliances like TV, air conditioner, projector, etc. The RC-5 and NEC are the protocols used in the IR remote. TSOP 1738 series IR receiver is used here. The receiver converts the IR signal into electrical pulses. Fig. 7 and Fig. 8 shows the IR transmitter and IR receiver respectively.

Figure 7: *IR Transmitter*

Figure 8: *IR Receiver*

f. Tarang-P20 ZigBee Transceiver

ZigBee modules are less expensive and consume less power than Bluetooth and Wi-Fi. Tarang P20 ZigBee module is used which operates within ISM 2.4 - 2.4835 GHz frequency band (Fig. 9). The operating voltage of this module is 3.3 V and operating current is 40 mA [10]. Initially, pair of ZigBee modules are configured to communicate using AT commands in the 'HyperTerminal'.

Figure 9: *Tarang-P20 ZigBee Transceiver Module*

IV. SOFTWARE TOOLS

To implement the web based smart home control system, various software tools including Arduino IDE, Python, ExtraPuTTY, PHP and HTML are used.

a. Arduino Integrated Development Environment

The Arduino can be programmed by use of simple language like C/C++ by the use of Arduino IDE. The IDE tool helps in writing, compiling and uploading the code. It is the common tool for all the Arduino boards. The model of the board should be selected from the tools bar of IDE before programming.

b. ExtraPuTTY

ExtraPuTTY is the application which is helpful in communicating with the Raspberry Pi. The Pi is setup by using the ExtraPuTTY. We use SSH (i.e. secure shell communication) for communicating with the Pi. The terminal window of PuTTY is used to enter the commands to execute our task.

c. Python Language

The Raspberry Pi supports many languages. Among them python is chosen because of its extensive libraries and portability. It is a high level language. It is easy to learn and simple in nature. Python supports both structured and object-oriented programming [11].

d. HTML and PHP

Hyper Text Markup Language (HTML) is used for webpage design. HTML tags can be used for webpage format and display. Hypertext Preprocessor (PHP) is used to post the data from webpage to the server. It can perform various operations like open, read, write and close on the files of database [12].

V. IR REMOTE DECODING

For IR remote decoding, we need the original remote control of the appliance, IR receiver and Arduino board connected to PC. The IR receiver connected to the Arduino board is programmed to receive and display the IR code emitted by the remote control. The code is received at appropriate baud rate to accurately monitor the emitted signal. The IR remote control is operated at 38 KHz normally.

The received IR signal, generally follows a common pattern. It is noticed that radiation pattern includes various fields such as header, bit mark of constant length separating the code of the button pressed. The remote control of a media player is decoded in this work. The bit duration of each field of the IR signal pattern for this remote is given in Table 1. It may also consists of a check sum and stop bits. When the IR signal is emitted, higher nibble is transmitted first followed by the lower nibble.

Table 1: Typical IR Signal Pattern

S.No.	Bit Transmitted	Bit Duration (µs)
1.	Header ON	9088
2.	Header OFF	4528
3.	Mark Bit	600
4.	Zero Bit	480
5.	One Bit	1500

The decoded patterns for various device control operations corresponding to the buttons pressed on the webpage are shown in Table 2.

Table 2: Decoded Patterns for Various Device Control Operations

S.No	Various Buttons on Webpage for Appliance Control	Data posted by PHP onto Pi	Data Transmitted by Pi to Arduino	Code emitted by IR Transmitter for Appliance Control
1.	Device On/Off	ON	"0"	807F12ED
2.	Mode	MODE	"1"	807F1AE5
3.	Mute	MUTE	"2"	807F1EE1
4.	Play/Pause	PLAY	"3"	807F01FE
5.	Previous Song	PREV	"4"	807F02FD
6.	Next Song	NEXT	"5"	807F03FC
7.	Equalizer	EQ	"6"	807F04FB
8.	Volume –	VMINUS	"7"	807F05FA
9.	Volume +	VPLUS	"8"	807F0679
10.	Light On	ON1	"9'	807E1AEE
11.	Light Off	OFF1	"9"	807E1AEE
12.	Fan On	ON2	":"	807E1ADA
13.	Fan Off	OFF2	":"	807E1ADA

VI. SCHEMATIC DIAGRAM, EXPERIMENTAL SETUP AND FLOWCHART

a. Schematic Diagram

The block schematic diagram of the Raspberry Pi web server is shown in Fig.10. In this, ZigBee module is connected to the Raspberry Pi board using a USB to serial converter. The Vcc, ground (GND), Transmit (Tx) and Receive (Rx) pins of the converter are connected to respective pins of ZigBee module. Web camera is connected to Raspberry Pi using the USB port. The block schematic diagram of In-Home Device control unit is shown in Fig. 11. The PIR sensor pins Vcc, Data and GND are connected to 5V, pin 4 and GND of Arduino board respectively. The Smoke sensor pins, Vcc, A_o (analog output) and GND are connected to pins, Vcc, analog pin (A0) and GND of Arduino board respectively. The two terminals of IR transmitter are connected to Arduino's pin 3 and GND. ZigBee modules's GND, Vcc, Tx and Rx pins are connected to GND, 5V, pin 5 and pin 6 of Arduino respectively.

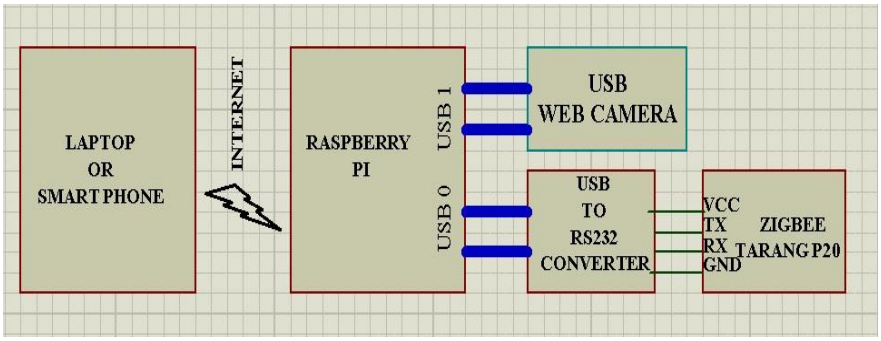

Figure 10: *Block Schematic Diagram of Raspberry Pi Web Server Unit*

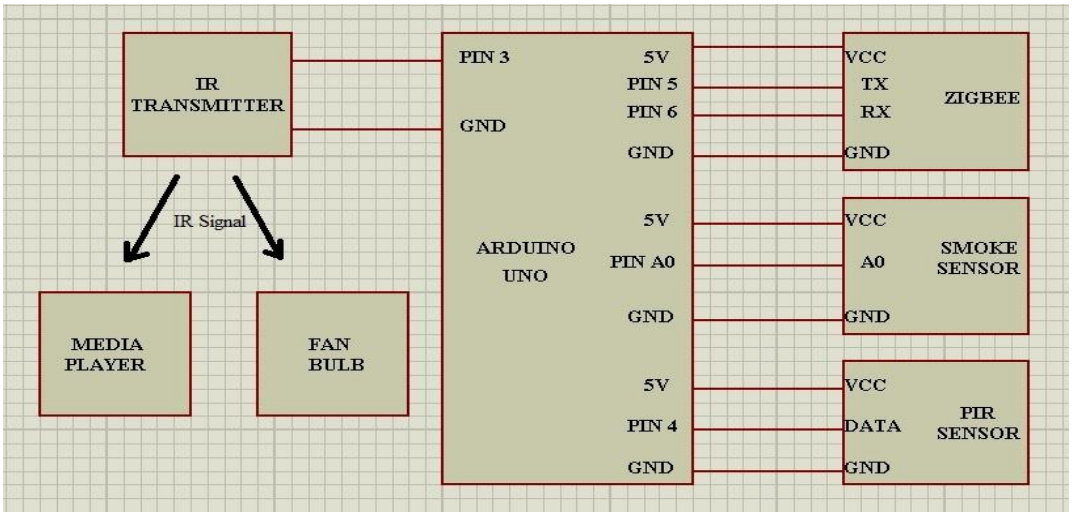

Figure 11: *Block Schematic of In-Home Device Control Unit*

b. Experimental Setup

Figure 12 shows the experimental setup of Raspberry Pi web server. It includes Wi-Fi router, Raspberry Pi, ZigBee module and web camera. The code to interface ZigBee and web camera is written in Python language. The SSID and password of the router needs to be saved in Raspberry Pi to access the Internet.

Figure 13 shows the experimental setup of In-Home Device control unit. The transmitter section consists of an Arduino ATmega 328P microcontroller board interfaced to various modules including ZigBee, PIR sensor, smoke sensor and IR transmitter. This unit transmits the necessary code to control the loads such as media player, light and fan. The load section consists of the IR receiver and an Arduino board for controlling the loads.

The code for interfacing the sensors and ZigBee module to Arduino is written in C. The sensor status is sent to the Raspberry Pi for display on the webpage.

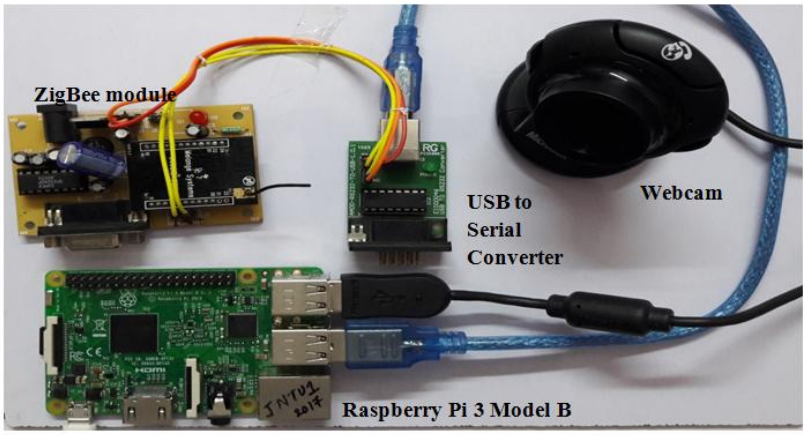

Figure 12: *Experimental Setup of Raspberry Pi Web Server Unit*

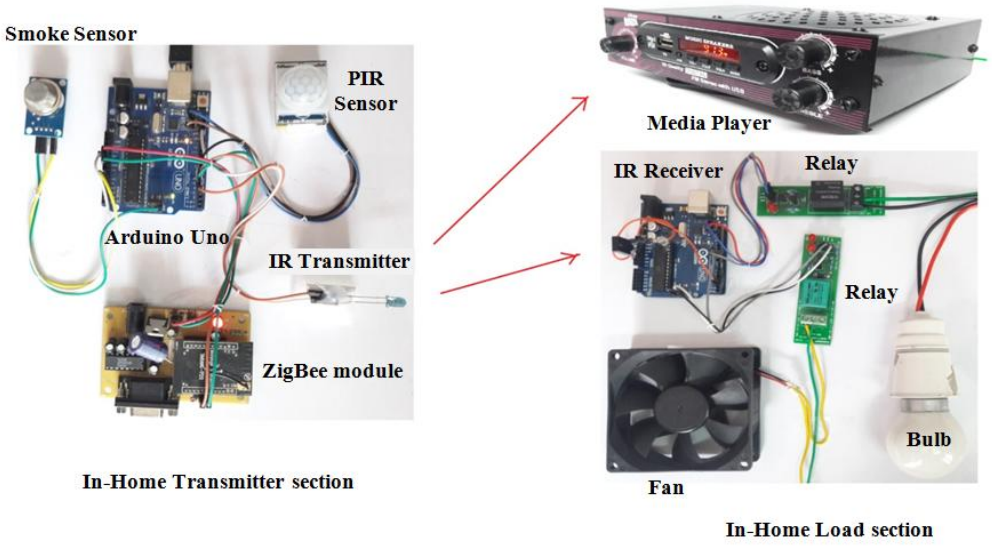

Figure 13: *Experimental Setup of In-Home Device Control Unit showing various Appliances*

c. Software Flowchart

The Smart Home Control System complete functioning is depicted in the flowcharts shown in Figs. 14 and 15.

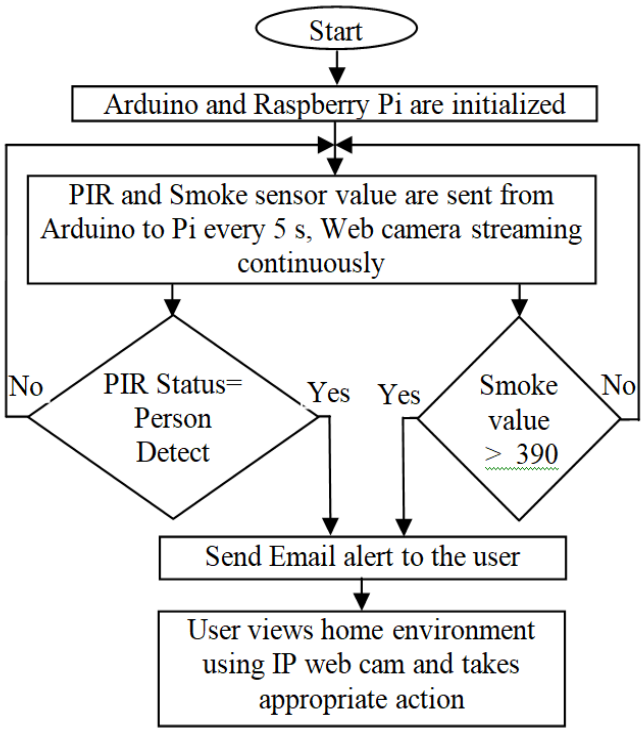

Figure 14: *Flowchart of events that occur in the Safety and Security block of In-Home Device control unit*

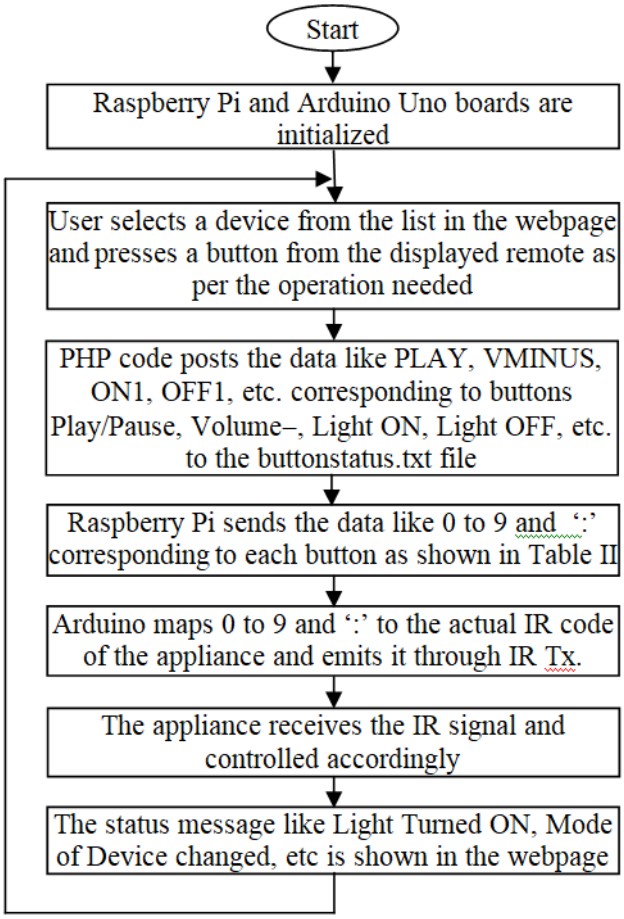

Figure 15: *Flowchart of various events that occur in the Smart Home Control System*

VII. RESULTS AND DISCUSSION

To operate the proposed web based smart home control system, user enters the IP address of the main page in the web browser as shown in Fig. 16. In the main page, the list of appliances to be controlled will be displayed. From the list of appliances, user can select any appliance. Then the corresponding appliance remote control will appear.

Figure 16: *Layout of Main page of Proposed Smart Home Control System*

Figure 17 shows the webpage view of the Media Player remote control. It appears when the Media player option is selected from main page.

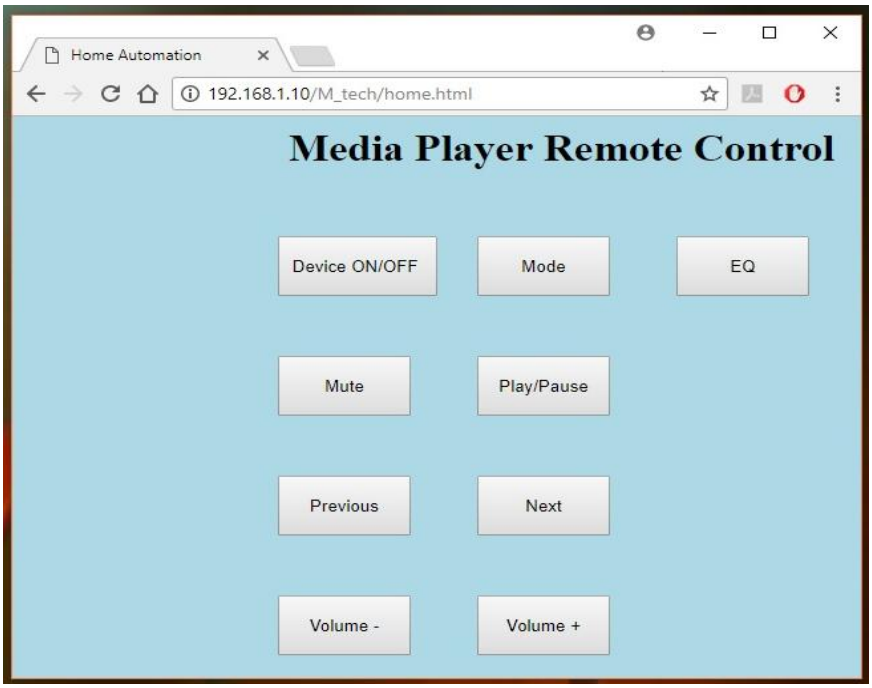

Figure 17: *Webpage View of Media Player Remote Control*

From the page of a browser, we can control the media player functions. The control operations like ON/OFF, Play/Pause, Mode, Volume+, etc can be performed. For button pressed in the web page, the PHP posts a pre-defined code in the buttonstatus.txt file. The buttonstatus.txt file is cleared after every operation. For the code posted in text file, the Raspberry Pi sends a corresponding single digit code to the Arduino via ZigBee. The Ardunio emits a particular IR signal for each single bit code. All the codes decoded for each button are given in Table 2. In the similar fashion, we can also control remaining devices. If the appliance does not have the IR control mechanism then we can control it by designing the IR receiver system. The IR receiver system design requires IR receiver (TSOP 1738), microcontroller (Arduino Uno) and relays (depending on type of load). This system can control both the high voltage and low voltage devices by use of relays. The web page for the Home Light and Fan control is shown in the Fig. 18. When the user presses the light ON (OFF) button, the light is turned ON (OFF) and the status is displayed on the webpage. Similarly, other loads in the home can be controlled.

Figure 18: *Webpage View showing Light and Fan Control Buttons*

The PIR status and smoke sensor data which is periodically sent from Arduino is stored in the status.txt file in Raspberry Pi. The received data is continuously added to the text file without erasing the previous data. We can see the previous status of the sensors, if needed. PHP and HTML code is written to display the data from status.txt file (i.e. sensor data) on the webpage.

If the smoke level is high, then the reading of the smoke sensor will be around 400 to 500. This indicates an emergency condition such as occurrence of a fire accident. Then an email alert about fire accident is

sent to the user. The user then views the home environment by accessing the IP web camera using an Internet enabled device and takes necessary action. Similarly, the PIR gives the status "No Person Present" when no one is present in the house. The snapshot of the Putty terminal window which is indicating the amount of smoke level and PIR status is shown in Fig. 19. When there is no smoke in the surrounding, the reading from smoke sensor will be around 100 to 200. This indicates the surrounding is safe. The PIR gives the status "Person Detect" when anyone is present in the house. An Email alert regarding the intrusion is sent to the user for further action.

When the user enters the address as shown in Fig. 20, the webpage displays the various sensor parameters, the status of PIR sensor along with date and time stamp. The Web camera can be used to stream the video of the house. The user can check the video footage to confirm any intrusion on receiving an email alert. The user can also take snapshot of the intruder, which is helpful in finding the culprit. The sample snapshot of the home as captured by the web cam is shown in Fig. 21.

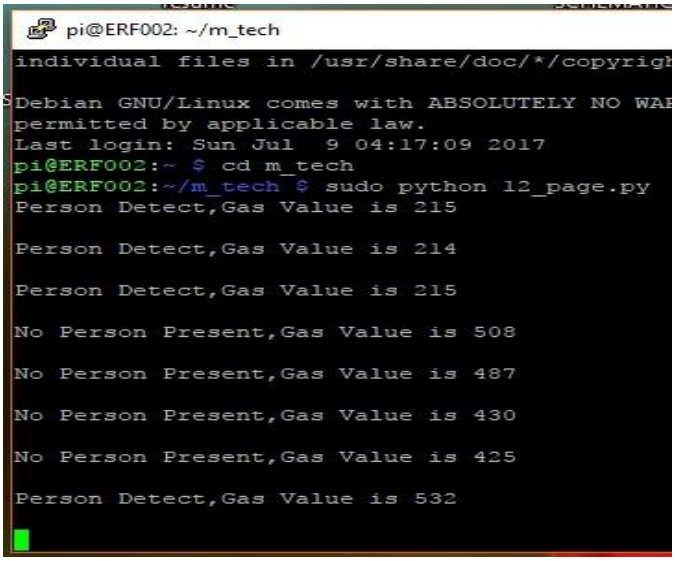

Figure 19: *PIR Sensor Status and Smoke Level*

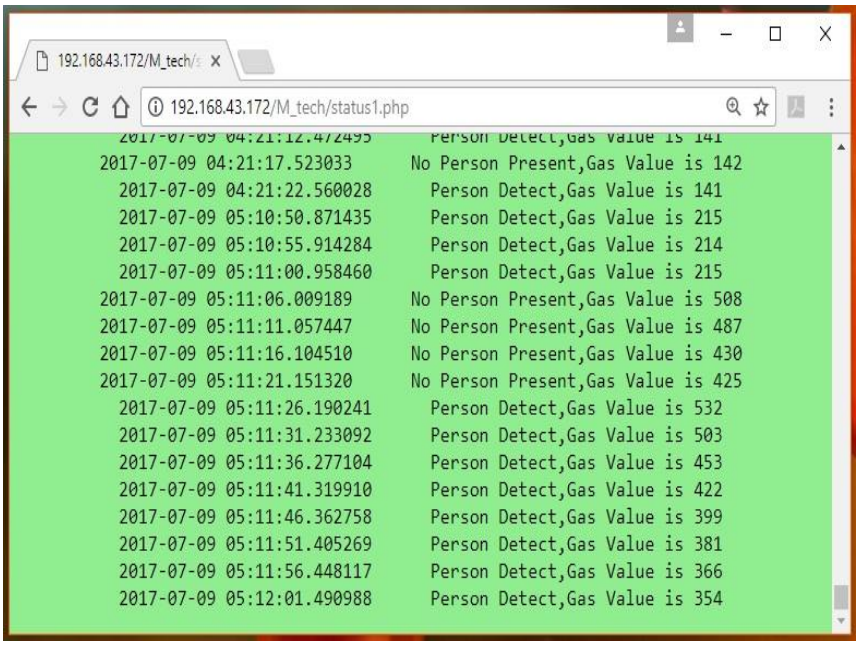

Figure 20: *Status of Smoke Sensor and PIR Sensor as displayed on Webpage*

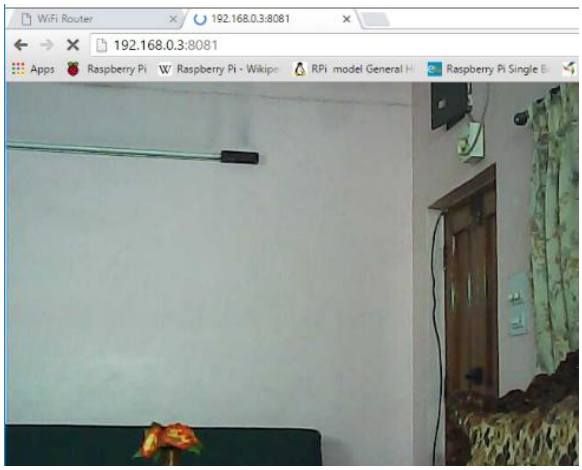

Figure 21: *Sample Webpage view of the Home environment as captured on Webcam*

VIII. CONCLUSIONS

A web based remote control for smart home applications is designed and implemented. The system is very simple and cost effective to use and can replace individual remote controls in the home. The developed web based smart home control system can be operated when the user is at home or when he is away from the home conveniently using any internet enabled device such as a smart phone. It is very useful for elderly and physically disabled people to operate the appliances without actually going to the appliance. Also, certain security and safety features are incorporated to enhance the functionality of the system.

REFERENCES

[1] Sung-Jung Hsiao, Kuang-Yow Lian and Wen-Tsai Sung (2016), "Employing Cross-Platform Smart Home Control System with IOT Technology Based", International Symposium on Computer, Consumer and Control (IS3C), Xi'an, China, 4-6 July, pp, 264-267, IEEE.

[2] Ch. Pandu Ranga Sai, V.Sameeka Datta and Sudheera (2016), "Design of a Smart Remote", International Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, India, 18-19 Mar., pp. 1-4, IEEE.

[3] Dhiraj Sunehra and Veena, M. (2015), "Implementation of Interactive Home Automation Systems Based on Email and Bluetooth Technologies", International Conference on Information Processing (ICIP), Pune, 16-19 Dec., pp.458-463, IEEE.

[4] ElKamchouchi, H. and Ahmed ElShafee (2012), "Design and Prototype Implementation of SMS Based Home Automation System", IEEE International Conference on Electronics Design, Systems and Applications (ICEDSA), Kuala Lumpur, Malaysia, 5-6 Nov., pp.162-167.

[5] M. Tharaniya Soundhari and S. Brilly Sangeetha (2015), "Intelligent Interface Based Speech Recognition for Home Automation using Android Application", International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 19-20 Mar., pp.1-11, IEEE.

[6] Van Der Werff, M., Gui, X. and Xu, W.L. (2005), "A Mobile-Based Home Automation System", 2nd International Conference on Mobile Technology, Applications and Systems, Guangzhou, China, 15-17 Nov., pp. 5, IEEE.

[7] https://www.raspberrypi.org/

[8] https://www.arduino.cc

[9] Adafruit Industries, "PIR Motion Sensor Manual", New York, 2016.

[10] Melange Systems Pvt. Ltd., "Tarang-P20 Product Manual", Bangalore, India.

[11] Richard L. Halterman, "Learning to Program with Python", 2011 (www.cs.uky.edu/).

[12] Larry Ullman, "PHP for the Web", Peachpit Press, San Francisco, 2016.

Dhiraj Sunehra. "Design and Implementation of Web Based Smart Home Control System using Raspberry Pi." IOSR Journal of Engineering (IOSRJEN), vol. 09, no. 3, 2019, pp. 18-28.