International Journal of Innovative Technologies
WWW.IJITECH.ORG
ISSN 2321-8665
Vol.03,Issue.01,
May-2015,
Pages:0054-0058

Home Automation using Cloud Network and Mobile Devices

DIPTI P. WALE[1], PROF. S. S. PATIL[2], DR. S. V. ANEKAR[3]

ABSTRACT: Today, we are entering post-PC era where mobile devices (e.g. iPads, Smartphones and Handheld tablets) are handling daily tasks that traditional desktop and laptop computers once handled. Several reports show that personal computers are no longer on the leading the edge of computing and the use of mobile devices are quickly taking over. Accompanying the shift from PCs to multi-touch mobile devices is the use and implementation of Cloud Networking. With the availability of products which integrate mobile devices and cloud networking rapidly increasing, many users can see how new technology can impact their everyday lives. In this paper we have developed a Home Automation system that employs the integration of multi-touch mobile devices, cloud networking, wireless communication, and power-line communication to provide the user with remote control of various lights and appliances within their home. This system uses a consolidation of a mobile phone application, handheld wireless remote, and PC based program to provide a means of user interface to the consumer. The home automation system differs from other systems by allowing the user to operate the system without the dependency of a mobile carrier or Internet connection via the in-home wireless remote. This system is designed to be low cost and expandable allowing a variety of devices to be controlled.

Keywords: Cloud Networking and Data Infrastructure, API (Application Programming Interface).

I. INTRODUCTION

Modern advances in electronics and communications technology have lead to the miniaturization and improvement of the performance of computers, sensors and networking. These changes have given rise to the development of several home automation technologies and systems [1,2]. According to [3], home automation can be useful to those who need to access home appliances while away from their home and can incredibly improve the lives of the disabled. Many of the home automation systems that are commercially available can be separated into two categories: locally controlled systems and remotely controlled systems. Locally controlled systems use an in-home controller to achieve home automation. This allows users complete use of their automation system from within their home via a stationary or wireless interface. Remotely controlled systems use an Internet connection or integration with an existing home security system to allow the user complete control of their system from their mobile device, personal computer, or via telephone from their home security provider. There are a number of issues involved when designing a home automation system. Piyare and Tazil [4] discussed that the system should be scalable so that new devices can easily be integrated into it. It should also provide a user- friendly interface on the host side, so that the devices can be easily setup, monitored, and controlled. Furthermore the overall system should be swift enough to realize the true power of wireless technology. Lastly the system should be cost effective in order to justify its application in home automation.

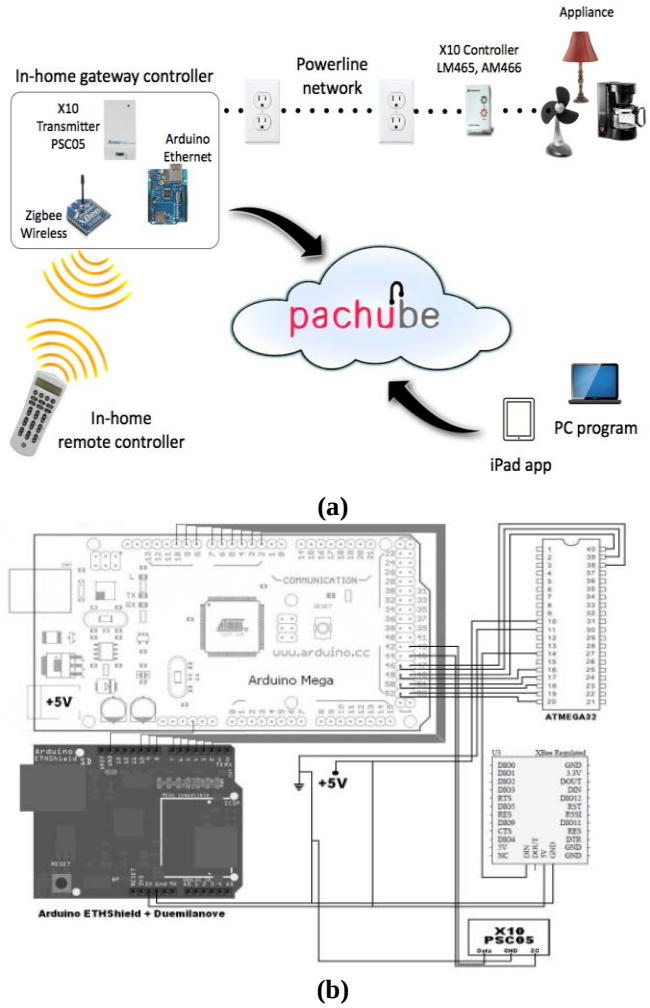


Fig.1. A system diagram of home automation using Cloud network and mobile devices: (a) communication between hardware components, (b) circuit diagram inside the in-home gateway controller.

To minimize the shortcomings of each system and to overcome the design issues previously mentioned, this project integrates locally and remotely controlled systems with the use of Cloud data network. This allows the system to operate without the dependence of a mobile provider, allows the system to be used with various mobile phone platforms, and allows the system to operate locally when phone or computer access is not available. Cloud networking and data infrastructure allow individuals to monitor, manage, and control their personal data points through the Internet [5-7]. One of the available services is Pachube [5]. It is a real time data infrastructure that allows management of millions of data points from thousands of individuals, organizations, and companies. It has physical-to virtual API which allows sophisticated applications and graphs to be built off of the user's data points. This paper will discuss how Pachube's API's were used to create home control user interfaces and monitoring graphs. Pachube allows its API's to be manipulated through different protocols such as: XML, CSV, JSON, and EEML. Pachube contains data streams which represent a single unit within the controlled house. Each data stream is given a unique feed identification number to differentiate itself from all other data streams on the Pachube network, therefore each data stream can be controlled and monitored separately at anytime.

The free service plan allows one month of storage memory, five data streams, five API requests per minute, and 500 import data points per day. The five units can be monitored up to one month history, which is ideal since most energy cycles are monthly. Pachube allows the user to control and monitor their home through PC, cellular, and Wi-Fi devices at anytime from anywhere in the world. This paper will discuss the development of a home automation system, as illustrated in Fig. 1a, that uses an integration of Pachube cloud networking, mobile devices, low-cost microcontrollers, X10 devices, and in-home user interfaces. From a mobile device, the user can run the mobile phone application and control the system via an Internet connection and cloud networking. From a personal computer, a Visual C# based program and cloud networking provides the user with an interface for control of the system. In home control is achieved via a hand held remote which uses Zigbee wireless communication to the in home controller which is integrated with the cloud network via an internet connection.

II. SYSTEM DESIGN

A. Proposed work

In this work a home automation system will be developed, as illustrated in Fig. 1, that uses an integration of Pachube[5] cloud networking, mobile devices, low-cost microcontrollers, and in-home user interfaces. From a mobile device, the user can run the mobile phone application and control the system via an Internet connection and cloud networking. From a personal computer, a Visual C# based program and cloud networking provides the user with an interface for control of the system. In home control is achieved via a hand held remote which uses Zigbee wireless communication to the inhome controller which is integrated with the cloud network via an internet connection.

B. Pachube Cloud Network

Pachube [5] is the networking cloud used in this design. Each data stream is classified by a feed identification number and secured by a forty-seven character password. Pachube treats each unit in the house as a data point. These data points are manipulated to control the unit inside the home automation system. When a unit is turned on, a value of one is given to that action, and zero for turning off action. When a unit is brightened or dimmed a value between the range of zero and nine is given to that action. After each unit is set to its desired action, the values are placed in a networking package and sent to the cloud. Pachube has an API (Application Programming Interface) editor that allows the user to create and design user interfaces and graphs for the user's need. The graphical interface can be designed and created through Pachube's API.

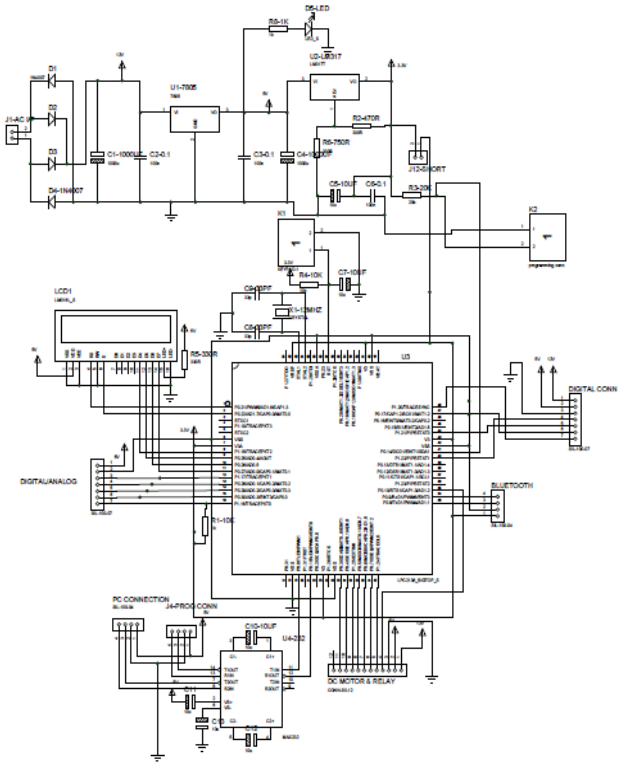


Fig.2. Circuit Diagram.

For example, switches and knobs can be used on the applications interface to simulate on/off and dimming respectively for the units in the Home Automation System, as shown in Fig. 3b. The API also allows the programmer to dictate the color, pixel dimension, title, axis, and time span of the graph. Whenever the user chooses the option to monitor a specific unit in the system, the software is programmed to extract the data points of that corresponding unit and plot a graph of it over the options of the past twenty-four hours or thirty days. To access Pachube outside the home, users can use either the iPad application or C# based PC application which were developed in this project. On the iPad app, UIWeb view class is mainly used to provide a simple way to display HTML content from Pachube inside an iOS application. Fig.3. Shows several screenshots of the iPad application that can control and monitor different appliances and circuit diagram is shown in Fig.2.

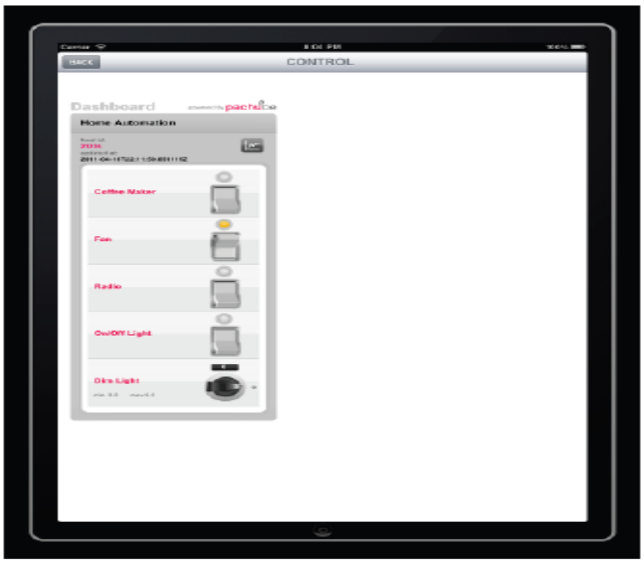
C. In-Home Gateway Controller

As shown in Fig. 1a, the gateway controller interfaces Pachube cloud with in-home equipment. The Arduino platform is used for the main communication between several components, i.e. internet connection, wireless and power line communication. Arduino is an open-source single board microcontroller which is designed to make the process of using electronics in multidisciplinary projects more accessible. The main hardware consists of a simple open hardware design for the Arduino board with an Atmel AVR processor and on-board I/O support, while the software side of the Arduino consists of a standard programming language and a boot-loader which runs on the board. Arduino hardware is programmed using a language similar to C++ with some modifications, and a Processing-based IDE [8]. An important feature of the Arduino family is the customary way that connectors are exposed, allowing the microprocessor board to be connected to a variety of interchangeable add-on modules (known as shields). As shown in Fig. 1b, the Arduino Ethernet Shield is connected, by stacking, to the Arduino Duemilanove through its metal pins. The Arduino Ethernet Shield is connected to the Pachube cloud by RJ-11 connector. The software of the Arduino Ethernet Shield is written to receive a networking package from the Pachube data infrastructure [9]. Using the server IP address for Pachube as 173.203.98.29, the firmware is written to request information from the Pachube data server from a certain feed identification. Within the firmware code, the forty-seven character password and the feed identification are used to access this networking package. These numbers are defined as follows.

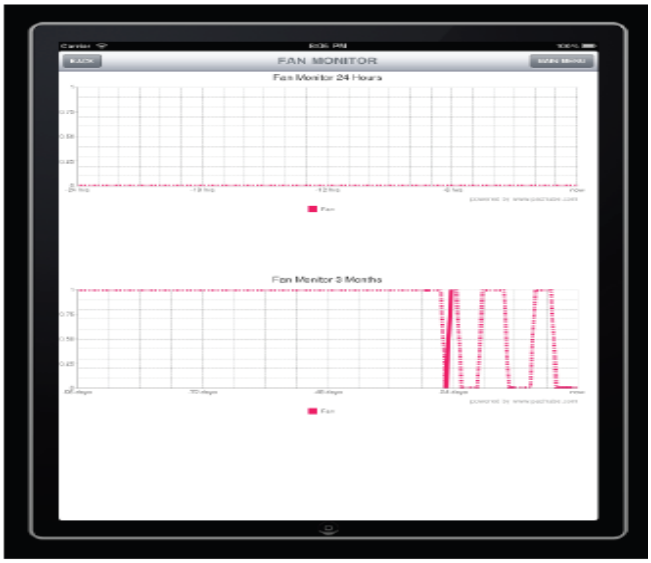
```
PACHUBE_FEED_ID 21204          PACHUBE_API_KEY
"JSFWp-nnxqdoh-tLkc2FA6gwJwhTV81zocgXvOI0Yl8"
```

The Arduino Ethernet shield takes the received networking package and manipulates it to extract the needed control values. After receiving the control values, the correlated units are adjusted in response of the values, such as turn on/off lamp number 1. The snippet of Arduino code communicating with Pachube is shown below.

(a)



(b)



(c)

Fig.3. Screenshots of iPad application for home automation using Pachube cloud; (a) the main menu where users can select to control or monitor the appliance, (b) the user interface for controlling appliances, and (c) the user interface for monitoring each appliance status.

The function starts with reading one ASCII character at the time into a buffer. The reading is limited to only 64 characters. Once the function detects the terminal string, it checks the string keyword, “200 OK”, to distinguish between valid and blank packets. If it is a valid packet, the numbers after the second keyword, “close” are retrieved and used to control the appliance afterward. The retrieved data will be compared to the previous packet, if the value in each appliance unit is different, the new command, based on the new value, is sent to X10 transmitter (PCS05) via stacked Arduino Duemilanove. The Arduino Mega is used for ZigBee wireless communication with an in home remote controller. ZigBee is a wireless technology developed as an open global standard to address the unique needs of low-cost, low-power, wireless sensor networks. The standard takes full advantage

of the IEEE 802.15.4 physical radio specification and operates in unlicensed bands worldwide at the various frequencies. It is designed around low-power consumption allowing batteries to essentially last forever. The low cost allows the technology to be widely deployed in wireless control and monitoring applications, the low power-usage allows longer life with smaller batteries, and the mesh networking provides high reliability and larger range.

```
void checkForResponse() {

    char c = client.read();
    buff[pointer] = c;
    //read up to 64 chars
    if (pointer < 64) pointer++;

    //distinguish between a blank or non-blank packet
    if (c == '\n') {
        //Keyword in Pachube string
        found = strstr(buff, "200 OK");

    //once there is a data packet, search for key information
    if ((!found_session_id) && (!found_control_values)){
        if (!found_d) {
            //search for everything after close, which are numbers.
            found = strstr(buff, "close");
            if (found != 0){
                //save Pachube data into a new array
                for (int i = 0; i < 15; i++)
                    digital_state[i] = (int) (client.read() - 48);
            }
        }
    }
}
```

D. X10 Communications

Of the many home automation systems that are currently available, two basic methods of data transmission are used among them. By using the existing electrical lines within the house to transmit a binary signal that is embedded in the zero crossing point of the sinusoidal AC power supply, power line communication can send commands throughout a house to control various electronic devices [10]. This method is widely known as X10 transmission and has been used to control devices for many years and is still commonly used in home automation systems. Other systems use either radio frequency signals or combinations of radio frequency signals and X10 transmission to communicate from the main transmitter to the individual controllable components. Each method has their strengths and weaknesses when examined for reliability and cost. Examining X10 transmission it is evident that components are widely available and very affordable, but there can be reliability issues related to using this method in certain settings. In reviewing home automation systems that use X10 transmission and radio frequency signals such as Insteon [11,12], it is evident that the method of transmission is more reliable thanks to the duality of how the signals are sent and received.

However, with the increased reliability of these systems comes increases in costs and decreases in component availability. While either of these systems could have been used in designing home automation, X10 was chosen in this project based on its reliability to cost ratio, ease of integration with available microcontrollers and overall availability of equipment. X10 communicates between transmitters and receivers by sending and receiving signals over the power line wiring. These signals involve short RF bursts which represent digital information [10]. A binary value of 1 is represented by a 1 millisecond burst of 120 kHz at the zero crossing point of the AC sine wave, and a binary value 0 by the absence of 120 kHz. The Gateway Modules modulate their inputs with 120 kHz, therefore only the 1 ms 120 kHz "packet" need be applied to their inputs.

E. iOS Mobile Application and PC Application

The new iOS application was created by the SDK 4.3. Most of the programs in this paper were implemented using Cocoa Touch and Objective-C code. To start creating an iOS application, there are four main parts. The first part is the XIB file which contains the graphical interface. The second part is the header files; this is where all the necessary libraries are included into the project. The third part is the main files where all the code and functions needed are written to execute required tasks. The last part needed is the resource folder which contains all graphics and sounds needed. Before the application can run, source code and interface builder must be linked properly. The created mobile home automation application, as shown in Fig. 3a, contains two settings: monitor and control. These two settings are created primarily from web view user interfaces. The control setting, as shown in Fig. 3b, allows the user to control any device in their home automation system from anywhere in the world with an internet connection. The monitor setting, as shown in Fig. 3c, allows the user to view how often each device has been used.

The monitor setting gives the user the option to view the usage of any device in the home automation system over the previous twenty-four hours or thirty days. The project design was created this way because twenty four hours gives the user the view of daily usage or thirty days which is the average billing cycle. The sample of iOS source code is shown below. For the PC application, Visual C# 2010 Express was used to create the application. C# is created for building different applications that run on the .NET Framework. To start creating the visual C# application, there are different levels. First the C# files, resources, and references communicate to the C# compiler. Next, the compiler creates the managed assembly. Next, C# uses .NET Framework and its class libraries to write the program. Finally, the project is converted to its native machine code for the operating system.

III. BLOCK DIAGRAM

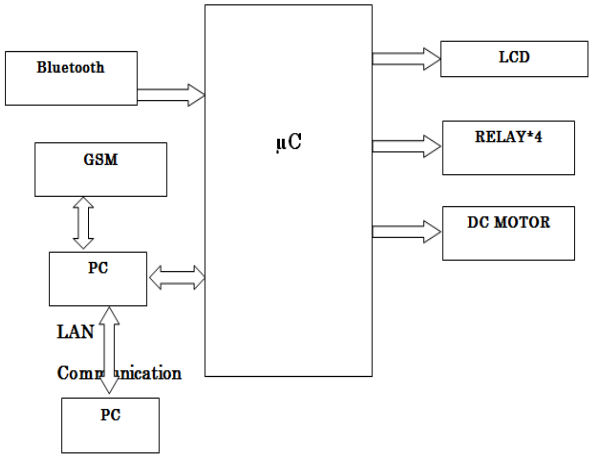


Fig.4. Block Diagram.

In Fig.4 block diagram here I turning On/OFF the appliances by various techniques

- LAN Communication
- GSM Communication
- Bluetooth Communication (Android)

A. LAN Communication

In this technique we are designing a client server model which communicate via RJ45 LAN cable using Socket programming. The server can monitor as well as control the appliances via LAN communication. For this we have an VB software on server as well as client. Firstly, the Client and server will set the IP address. If the IP address matches then the connection is established. After that the Domain name is confirmed. After that the data communication is done using Socket programming.

B. Bluetooth Communication

For Android Communication first do the Blue tooth pairing of Bluetooth modem connected to µC and the Bluetooth of Android Phone. After the pairing is enabled then the data can be transferred to mobile using Bluetooth medium. After sensing the data the µC will send the data on its serial port using serial port at 9600 baud rate. The data is then transferred to the blue tooth modem connected to µC. The Blue tooth modem will then transfer the data wirelessly to the other Bluetooth module in the mobile phone. Since they are paired, we can receive the data on Mobile phone using Bluetooth port. On Mobile an Android platform based Graphical User Interface (GUI) to monitor the Vehicle data on a Graphical user interface. Also we can add a control window through which can monitor the parameters.

C. GSM Communication

Here the user can control ON/OFF the electrical Appliance via SMS using "AT" commands. The µC is connected to GSM Modem which is first initialized using AT commands via RS 232 Protocol. After that the SMS that is sent by user is received By GSM Modem. The µC will receive the SMS and store the SMS in internal RAM and display the frame on LCD. Depending on frame the µC can turn ON/OFF the home appliances.

IV. RESULTS

The project was completed and is working as described in this paper. The system allows the user to control appliances and lights in their home from an iPad and PC from anywhere in the world through an internet connection. It also allows the user to control their units within their home from a wireless remote. The wireless remote has primary control over the system; therefore if the remote is active neither the iPad nor PC will be able to control the units in the home. This design prevents from the iPad, PC, and wireless remote all trying to control the system at the same time. The project was tested to turn appliances on and off such as: radio, fan, coffee maker, and television. It also was tested to brighten and dim various light structures. The system refreshes on the iPad and PC every time the user chooses an option to control or monitor a specific unit. The in-home remote is updated on the LCD monitor every time the system receives a command. The project did run into a memory problem. After research, we found that the ArduinoDuemilanove's flash drive does not operate well with the Arduino Ethernet Shield connected. Therefore all the incoming data from Pachube had to be saved on the Arduino Duemilanove's EEPROM data storage. This posed a problem as the EEPROM only allows the user to rewrite over data for a certain number of times. An external Arduino flash drive is considered for the option for further work.

V. CONCLUSION

By integrating multi-touch mobile devices, cloud networking, wireless communication, and power-line communication, we were able to design and build a fully functional home automation system. It allows the user to control various appliances and lights within their home from any location in the world through Pachube cloud network using 1) mobile devices, 2) PCs, or 3) in-home remote controller. Using this system as framework, the system can be expanded to include various other options which could include home security feature such as open-door and motion detection, energy monitoring, or weather stations.

VI. REFERENCES

[1] Chan, M., Campo, E., Esteve, D., Fourniols, J.Y., "Smart homes-current features and future perspectives," Maturitas, vol. 64, issue 2, pp. 90-97, 2009.

[2] Das, S.R., Chita, S., Peterson, N., Shirazi, B.A., Bhadkamkar, M., "Home automation and security for mobile devices," IEEE PERCOM Workshops, pp. 141-146, 2011.

[3] Laur, I., "Microcontroller based home automation system with security," International Journal of Advanced Computer Science and Applications, vol. 1, no. 6, pp. 60-65, 2010.

[4] Piyare, R., Tazil, M., "Bluetooth based home automation system using cell phone," IEEE ISCE, pp. 192-195, 2011.

[5] https://pachube.com/.

[6] http://www.arkessa.com.

[7] http://www.idigi.com/.

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

[9] http://community.pachube.com/arduino/ethernet.

[10] http://www.x10.com/support/technology1.htm.

[11] Gomez, C.,Paradells, J.," Wireless home automation networks: A survey of architectures and technologies," IEEE Communications Magazine, vol. 48, issue 6, pp. 92 – 101, 2010.

[12] http://www.insteon.net/.

```
- (void)viewDidLoad {

    NSString *urlControlAddress = @"http://apps.pachube.com/dashboard/21204/l8QDkT5xkIdQRKTpjEE9_wDN-ZTY2q4gMRbT9T8ecBw";

    NSURL *url = [NSURL URLWithString:urlControlAddress];              //create a url object
    NSURLRequest *requestObj = [NSURLRequest requestWithURL:url];      //URL request Object
    [webControl loadRequest:requestObj];                               //Load the request in the UIWebView
    [super viewDidLoad];
}
```