Design and Implementation of an Internet of Things Communications System for Legacy Device Control and Management

Martin Saint[1,2](✉), Aminata A. Garba[1], Audace Byishimo[1], and Rodrigue Gasore[1]

[1] Department of Electrical and Computer Engineering, Carnegie Mellon University, Kigali, Rwanda
{msaint,aminata,abyishim,rgasore}@andrew.cmu.edu

[2] Interdisciplinary Telecommunications Program, College of Engineering and Applied Science, University of Colorado Boulder, Boulder, CO 80309-0530, USA
martin.saint@colorado.edu

Abstract. Applying the capabilities of the Internet of Things holds particular promise for advancing certain of the United Nations Sustainable Development Goals. For instance, citizens in many developing countries lack access to energy, and suffer from a lack of energy independence, scarcity, and high energy costs. Efforts are underway to make energy generation, transmission, and consumption more efficient via the application of modern communications and control. In this paper we focus on a communications and demand-side control application for legacy appliances. Instead of replacing existing devices with new "smart" devices with better communications capability, we integrate legacy devices into a centralized control system. We designed and implemented a communications system that allows control of infrared-enabled appliances over a network using wired or wireless communications. We designed a web application and user interface that can function remotely over the Internet, and use microcontrollers and infrared transmitters to communicate with infrared-enabled devices. We demonstrate a prototype of the system to control air conditioning units in a commercial building for energy management. The system uses open source hardware, software, and protocols. While we use energy management as an example, we focus on the general communications and control architecture, which is easily scalable and applicable to other devices and applications beyond those demonstrated here.

Keywords: Internet of Things (IoT) · Telecommunications · Wi-Fi · Internet · Pulse Width Modulation (PWM) · Infrared · Machine-to-machine · m2m · Energy management system · Smart building · Building automation system · eInfrastructure

1 Introduction

In most African countries electricity is scarce and expensive. In March 2016 the cost of electricity in Rwanda was US $0.22 per kilowatt-hour [1], compared to

T.F. Bissyande and O. Sie (Eds.): AFRICOMM 2016, LNICST 208, pp. 145–154, 2018.
https://doi.org/10.1007/978-3-319-66742-3_14

an average cost in the US of US $0.126 for residential customers and US $0.101 for commercial use [2]. The United Nations Sustainable Development Goals call for affordable and clean energy for all, which serves as a foundation for other infrastructure and development [3]. Given limited resources in fossil fuels, the best solution to addressing global energy demand lies in renewable energy sources such as solar and wind, as well as various demand side management techniques that reduce energy consumption and increase efficiency [4]. Energy systems will benefit from applications of the Internet of Things to monitor, control, and manage generation, transmission, and energy demand [5].

In a small home it is easy to monitor and manage energy usage, and on a continent where the average income is not large, families are well aware of the financial impact of electricity usage. However, in large commercial buildings which lack any form of centralized energy management system, we observed ample opportunities for energy savings. Like similar buildings, our building was constructed to take advantage of natural light and ventilation where possible. During the day lighting was, in fact, seldom necessary. Ventilation, however, did not provide adequate cooling for most of the building, and therefore air conditioning units had been installed. On our campus, air conditioning was the largest consumer of electricity, and for this reason, we decided to focus our demand-side energy management efforts there.

In developed countries it is common for buildings to have central air conditioning, where cool air is delivered via ducts; or cold water or other refrigerant is delivered to individual fan-coil units distributed throughout the building. These systems depend on large centralized chillers or air conditioning units, and have electric or pneumatic controls which manage temperature and are often controllable from a remote central location. As is common in Africa, our building utilized a so-called *mini-split* or *ductless* system, which does not share any centralized components or control, and only serves one room per system. Mini-split systems have an air conditioning compressor and heat exchanger which is mounted on an exterior wall or balcony. The compressor serves a single fan-coil unit mounted in the room to be cooled, which is the component most users identify as the air conditioner, visible in Fig. 5. The compressor and fan coil unit are connected by refrigerant and electrical lines. The user can control functions such as temperature, fan speed, cooling mode, and blower vane direction via a keypad on the in-room unit, or with an infrared remote control. Our objective was to build an inexpensive communications system that can control infrared devices and centralize energy management using open source hardware and software. Our prototype focused on controlling air conditioning, but the approach is general and could apply to any other appliance which is infrared-enabled, such as floor fans or television monitors. It is also straightforward to retrofit infrared-enabled switches on lighting for a more comprehensive energy management system, but for this work we focus on air conditioning.

1.1 Other Work

Remotely monitored fire alarm systems date back to the 1870s, an example of an early building automation system [6]. Other building systems, such as elevators and heating/cooling, have had control systems for decades [7]. While not necessarily sophisticated or interconnected, they did provide a degree of automation and control for large buildings. The Internet of Things (IoT) provides new capabilities for smart buildings, and even smart homes. The International Telecommunication Union defines the Internet of Things as "a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies" [8]. The IoT enables better synergy between three functions that have traditionally been difficult to integrate in building automation: monitoring, communications, and computation [9]. The IoT also enables connecting a larger variety of devices at lower cost, more monitoring and control options, and offers the possibility of using open source hardware, software, and communication protocols.

With the continuing development of the Internet of Things, a larger number of "smart" appliances and devices are being manufactured. These include devices with embedded sensors, radio frequency identification (RFID) tags, microcontrollers, and wireless transmitters and receivers. This enables the devices to integrate into home or building networks, and allows users to control them even over the Internet. We wanted to achieve similar functionality, but with existing non-smart devices. Undertaking work similar to ours, applications have been developed for controlling devices in buildings and the home using a variety of universal input devices. The Remote Commander was an early application which enabled users to control a personal computer using a handheld device [10,11]. This work is cited by Nguyen et al. as an influence in their research developing appliance control systems using infrared and powerline communications [12]. Powerline communications were also used for controlling smart appliances via a network by Kim et al. [13] and Khan et al. [14]. Ballagas et al. were pioneers in using smart phones as input and control devices, and developed a number of user interface guidelines [15].

The user input devices, whether smartphones, PCs, 2000s-era personal digital assistants (PDAs), or proprietary controllers, must ultimately be able to communicate with the devices they are intended to control. This typically requires an adapter, also called a proxy or bridge. The input device communicates to the adapter, which either controls the device state directly, or translates between communication protocols in the input device and the device or appliance it is intended to control. One of the first widely used protocol and adapter families was X10, primarily for home automation applications. It uses a proprietary command module, the X10 powerline communications protocol, and various control modules such as switches and timers. It can be used to control devices such as lights and appliances, and later featured control software which could run on a PC. Device control is also possible via the IEEE 1394 protocol, which until recently was widely used in personal computers for a wired connection to peripheral devices, but also had wireless, coaxial, and fiber optic versions. Universal Plug and Play (UPnP) performs similar device-to-device communications for home use, but does not scale well to commercial applications. Commercial building automation is frequently accomplished via proprietary interface devices and protocols such as Modbus, LonWorks, and BACnet. For our application we wanted to use open source hardware, software, and protocols to the extent possible, and we wanted the system to be wireless. We wanted hardware which was inexpensive and readily available, and we wanted to build a system which was scalable and available from any user input device. In keeping with the *appropriate technology* for developing countries theme proposed by Schumacher [16], we wanted to create a solution which solved our problem and met our design goals as simply as possible, and was simple for the end user to understand and operate. This precluded the use of proprietary modules, protocols, and interfaces, and we needed a system which did not depend upon smart appliances. By utilizing the existing infrared control capability of the legacy air conditioners in our building we were able to design and implement a communications and control system for central energy management. The system uses infrared communications to control the air conditioners, and Wi-Fi and the Internet to permit the user to send commands from ordinary devices like smart phones or laptops from anywhere.

2 System Model and Design

The system architecture is shown in Fig. 1, which illustrates the web-based communications system designed for infrared appliance control. Details of the system are as follows.

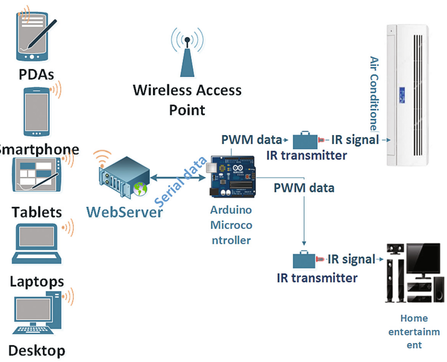

Fig. 1. The architectural view of the system.

2.1 System Components

The system is composed of:

1. A web server hosting a responsive web application. This provides the user with interface buttons rendered in software which control the same functions as the buttons available on the original factory remote control or on the air conditioner itself.

2. An Internet-enabled user interface device, such as a smart phone, tablet, laptop, or desktop computer. This permits access to the application on the web server.
3. An Arduino microcontroller and infrared control library, used to connect the web server hosting the control application to an infrared transmitter. The communication between the web server and the microcontroller can be via a wired or wireless connection.
4. Infrared transmitters, used to send control signals to the appliances as directed by the infrared control library on the Arduino.
5. Infrared-enabled target devices, such as building or home systems. In this case, air conditioners in individual rooms.

2.2 System Operation

To control the designated building or home devices, the application is opened by an authorized user on the desired user interface device, such as a smart phone. An Internet connection is established between the user interface device and the web server hosting the interface application. The user selects the device to be controlled, and is presented with a menu of control functions. For an air conditioner these consist of setting the cooling mode, temperature, the direction of airflow, or turning the device on or off. After the user selects the desired control function the web server sends an American Standard Code for Information Interchange (ASCII) text code corresponding to the instruction to the microcontroller. The microcontroller maps the ASCII code into an appropriate pulse width modulated (PWM) control command, and the PWM code is sent to the infrared transmitter(s). The infrared transmitter then sends the appropriate control command to the device being controlled, such as the air conditioner. In the next section we discuss the design considerations which were necessary to make the system operational.

2.3 User Interface Application

The user interface is a web application built using the HyperText Markup Language (HTML). To accommodate a variety of user interface devices we use responsive web design. As the user switches from one interface device to another, the interface application automatically adapts to the different device capabilities, such as for resolution, image size, and scripting language. This eliminates the need for the user to resize the application window, pan, or scroll in order to view the website on different devices. The user interface is composed of (1) the list of appliances to be controlled; and (2) the soft control keys that map to the control functions available for each device to be controlled. Each end device to be controlled must be preconfigured in the system so that it can be uniquely identified by the user and web application, and so that control functions appropriate for that device will appear in the user interface. For the system detailed in this paper we customized the interface application to control several Carrier brand air conditioner units in different rooms on our campus.

2.4 Web Server

The web server application was developed using the Node.js runtime environment, which is designed to build network applications with modules written in the JavaScript programming language [17]. It is an open source, cross-platform runtime for server-side programming that can run on all modern operating systems. As Node.js is non-blocking and utilizes JavaScript, an interpreted language, it responds to input in real-time and enables real-time communication [18]. After receiving user input, the web server sends an ASCII code that is specific for every instruction to an Arduino microcontroller.

2.5 Microcontroller

We used an Arduino UNO microcontroller for transmission of the signal from the web server to the end device to be controlled [19]. An Arduino program, or sketch, was written to transmit a signal corresponding to the user input command to the end appliance via an infrared transmitter connected to the board. Communication between the web server, microcontroller, and IR transmitter can be wired or wireless. With a wired connection cables are used between the web server, the microcontroller, and the IR transmitters. This option has the advantage of minimizing interference with other communications. However, to install the system for operation in multiple rooms would require installing cables to connect all devices, and likely a microcontroller per room. With a wireless connection, infrared modules are wirelessly connected to the microcontroller. This solution is easier to scale.

2.6 Infrared Transmitter

The infrared transmitter must be located within line of sight to the device it is intended to control and has a typical range of 10 to 15 m. The transmission pattern is relatively narrow, so for practical reasons this typically means one infrared transmitter for each end device to be controlled.

2.7 Control Commands

Every manufacturer and device model has its own set of control commands. Some manufacturers make these available on their official website. Another method of determining the appropriate commands is by capturing them from an existing infrared remote control transmitter supplied with the device to be controlled. We implemented a microcontroller program that listens to an infrared receiver connected to the Arduino and records the incoming set of commands from the manufacturer remote control. These commands are used later by a second programmed sketch that listens on a serial port for ASCII commands from the web server and communicates to the infrared transmitter in our system. The signal exchanged between the microcontroller/infrared transmitter and the infrared receivers is modulated using pulse width modulation (PWM), which is a technique of encoding a message into a pulsing signal [20].

3 Prototype and System Demonstration

We used our infrared receiver to capture control signals from the legacy remote control for the Carrier brand air conditioners used in our building, see Fig. 2. To assist with code capture we used the open-source IRremote Arduino library from Shirrif [21]. We modified the library to support the control commands required for our demonstration on different Carrier air conditioners. The commands captured by our infrared receiver are decoded by the library, and the corresponding voltages can be observed and saved into a text file, see Fig. 3. These commands are copied to the Arduino microcontroller sketch and can be transmitted using the infrared library. We designed and implemented the web server application and configured it to control the target Carrier air conditioners. Figure 4 shows a screenshot of the application running on a laptop PC. Via the desired user interface device and interface program we were able to successfully:

Fig. 2. Signal code capture from a legacy remote control via an infrared receiver.

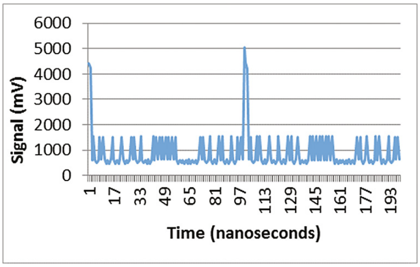

Fig. 3. Voltage over time representation of the pulse width modulated *on* command for a Carrier air conditioner.

- Select the desired room or location.
- Select the desired device to control.
- Toggle the power on and off using the power control button.
- Change the cooling mode from the available options.
- Change the desired temperature set point within the range permitted by the manufacturer.
- Change the air direction and swing mode using the appropriate button.

The infrared transmitter and air conditioner being controlled are shown in Fig. 5. The air conditioner has a factory infrared receiver designed to accept commands from the factory remote control.

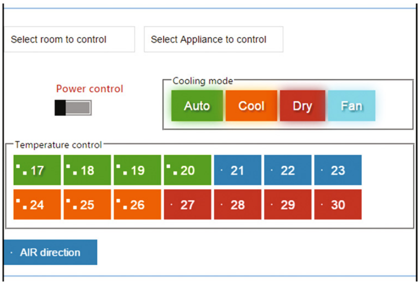

Fig. 4. Screenshot of the user interface.

Fig. 5. Infrared transmitter and air conditioner.

4 Discussion

Additional considerations for scaling and extending the project follow.

4.1 Communication Protocols

In our demo, the system operation was limited to the local area network (LAN), although the design is capable of operating over the Internet. This requires allocating a public IP address to the web server. The web server could also be hosted in the cloud, which is an increasingly practical option. The microcontroller unit is then connected to the server over the Internet, and wireless modules are mounted on the microcontroller for local connectivity. In addition to Wi-Fi, other communication protocols such as cellular wireless (GSM communication, 3G, 4G) can be used. The system could also be modified to incorporate other communication protocols for some functions under appropriate conditions, such as Bluetooth or ZigBee.

4.2 Device and Function Scalability

The proposed system is scalable to as many devices as desired as long as the initial commands are available from the manufacturer or can be captured using the legacy infrared remote control. Each time a new device is added, the web application is updated to support it and to present the device in the user interface. An interesting possibility is the idea of scaling the system to add new control functions beyond the ones initially programmed by the vendor. New functions can readily be created and communicated between the user interface device, interface application, web server, microcontroller, and infrared transmitter. Scaling the functionalities of a device becomes a question of envisioning the new functions and being able to successfully modify the device to support the new functions. Many devices already contain microprocessors, so they are limited only by their (current) inability to communicate.

4.3 Machine to Machine Communications

In the system demonstrated, integration of the infrared-enabled devices with the IoT is accomplished using a web application and a user interface. This human-to-machine communication model is extensible to enable machine-to-machine communication and control. For example, with smart grid communications, commands can be sent from a smart meter, energy controller, or an alternative energy system to appliances. The devices can be instructed to turn on or off, or to run with full or reduced functionality based on the availability or cost of energy. In such a case the proposed system runs an additional algorithm which takes as inputs the building energy information and control parameters and automatically sends the appropriate commands to the appliances. Similarly, the web application can send information to the home energy controller about the state of the appliances and their energy usage. This information could be utilized to construct a machine learning algorithm for forecasting and predictive control of energy generation and loads, enhancing capabilities for demand-side management.

5 Conclusion

In this paper we presented the design of a communications system for control of infrared-enabled appliances using a custom responsive web application. We demonstrated a working prototype of the system which controls air conditioners located in different rooms of a building. We showed that the proposed system does not require a line of sight between the user and the controlled appliance, and that it can work remotely. The communications and control system enables the integration of legacy infrared-enabled electrical and electronic devices into a building automation network using the Internet of Things. Significantly, it enables convenience and energy-savings without replacing existing appliances.

References

1. Economic Regulation Unit: Key statistics in electricity sub-sector as of March of the year 2016. Rwanda Utilities Regulatory Authority, Kigali, Rwanda, Report, March 2016
2. U.S. Energy Information Administration: Electric power monthly with data for. U.S. Department of Energy, Washington, DC, Report, March 2016
3. General Assembly: Transforming our world: the 2030 Agenda for sustainable development, United Nations, New York, Resolution A/RES/70/1, 25 September 2015
4. Suberu, M.Y., Mustafa, M.W., Bashir, N., Muhamad, N.A., Mokhtar, A.S.: Power sector renewable energy integration for expanding access to electricity in sub-Saharan Africa. Renew. Sustain. Energy Rev. **25**, 630–642 (2013)
5. Siano, P.: Demand response and smart grids-a survey. Renew. Sustain. Energy Rev. **30**, 461–478 (2014)
6. Moore, W.D.: Fire alarm system research-where it's been and where it's going, White Paper. Hughes Associates, Warwick, RI (2006)

7. Kastner, W., Neugschwandtner, G., Soucek, S., Newman, H.M.: Communication systems for building automation and control. Proc. IEEE **93**(6), 1178–1203 (2005)
8. Telecommunication Standardization Sector: Series Y: Global information infrastructure, Internet protocol aspects and next-generation networks-next generation networks-frameworks and functional architecture models-overview of the Internet of things. International Telecommunication Union, Geneva, Switzerland, Recommendation ITU-T Y.2060, June 2012
9. Dietrich, D., Bruckner, D., Zucker, G., Palensky, P.: Communication and computation in buildings: a short introduction and overview. IEEE Trans. Ind. Electron. **57**(11), 3577–3584 (2010)
10. Myers, B.A.: Using handhelds and PCs together. Commun. ACM **44**(11), 34–41 (2001)
11. Nichols, J., Myers, B.A.: Studying the use of handhelds to control smart appliances. In: Proceedings of 23rd International Conference on Distributed Computing System Workshops, pp. 274–279. IEEE, May 2003
12. Nguyen, T.V., Lee, D.G., Seol, Y.H., Yu, M.H., Choi, D.: Ubiquitous access to home appliance control system using infrared ray and power line communication. In: 3rd IEEE/IFIP International Conference in Central Asia on Internet (ICI 2007), Tashkent, pp. 1–4. IEEE, September 2007
13. Kim, D., Jun, T., Kwon, W.H.: Home network systems for networked appliances using power-line communication. In: 30th Annual Conference on IEEE Industrial Electronics Society (IECON), vol. 3, pp. 2394–2399. IEEE, November 2004
14. Khan, S., Islam, R., Khalifa, O.O., Omar, J., Hassan, A., Adam, I.: Communication system for controlling smart appliances using power line communication. In: 2nd International Conference on Information and Communication Technology, Damascus, vol. 2, pp. 2595–2600. IEEE (2006)
15. Ballagas, R., Borchers, J., Rohs, M., Sheridan, J.G.: The smart phone: a ubiquitous input device. IEEE Pervasive Comput. **5**(1), 70–77 (2006)
16. Schumacher, E.F.: Small is Beautiful: A Study of Economics as if People Mattered. HarperCollins, Scranton (1973)
17. Node.js Foundation: Node.js (2016). https://nodejs.org
18. Widman, J.: Overview of blocking vs non-blocking, GitHub Repository, March 2016. https://github.com/nodejs/node/blob/master/doc/topics/blocking-vs-non-blocking.md
19. Arduino.cc: Arduino UNO & Genuino UNO, n.d. https://www.arduino.cc/en/Main/ArduinoBoardUno
20. Hirzel, T.: PWM, Arduino Forum, n.d. http://www.arduino.cc/en/Tutorial/PWM
21. Shirriff, K.: IRremote Arduino library, GitHub Repository (2016). https://github.com/shirriff/Arduino-IRremote