A COST EFFECTIVE RESIDENTIAL TELEMETRY NETWORK

Sean Byland, Craig Clarke, Matt Gegg, Ryan Schumacher, Chris Strehl
with Dr. Kurt Kosbar (Advisor)
Missouri University of Science and Technology
Department of Electrical and Computer Engineering
Rolla, MO 65409

ABSTRACT

As cost and power consumption of wireless devices decreases, it becomes increasingly practical to use wireless communications and control in residential settings. These networks share some of the same challenges and constraints as conventional telemetry networks. This particular project focused on using a commercial, off-the-shelf router to implement a residential automation system using Z-Wave wireless devices. The router can communicate status, and accept commands over a conventional 802.11 network, but does not require a remote host to operate the network. The router was reprogrammed using open source software, so it could issue commands, collect data, and monitor the Z-Wave network.

KEY WORDS

Home automation, network, wireless, control

INTRODUCTION

Telemetry networks have been in use for a few decades now to automate a variety of aspects of the home. Primarily, these networks have served to create a comfortable atmosphere with some applications focusing on efficiency of devices throughout the home. A significant amount of progress has been made over the past decade as powerful computers continue to decrease in size and cost and communications technology has become easy to integrate. However, even with this progress, home automation networks have failed to penetrate much of the consumer market outside of now commonplace, programmable thermostats.

Though several software systems exist that allow a user's PC to control their system of controllers via RS-232 or a wireless standard; this requires a PC to be powered on for the system to be truly automated. Using a PC like this for total control of an automation process presents the problems of lacking real-time control while the system is off and negating energy efficiency while the system is on. A more suitable alternative that has been created to this PC only system is a hybrid plug-in system that utilizes a PC interface for programming and a separate low-power control module to carry out the scheduled commands. Hawking Technology provides such a model in its HomeRemote Gateway that utilizes Z-Wave technology [1]. This type of system model is what formed the basis for the design in this paper.

The residential telemetry network described in this paper is centered around a standard wireless router that has been modified through hardware and software to provide the flexibilities of a PC controller with the energy efficiency of a dedicated controller. A reduced version of Linux was installed onto the router to create an easily usable file system and to allow bash script programming for creation of automation commands. TTL outputs on the router were modified for RS-232 serial communication so the router could receive and transmit commands through a wireless transceiver chip.

DESIGN

A PC interfaced home automation transceiver was developed by modifying a wireless router to communicate to Z-Wave radio technology over a RS-232 connection. A Linksys WRT54GL wireless router satisfied cost and efficiency requirements as this standard router costs around $40-60 and uses a 12 Watt power supply as opposed to the 500-1000 Watt power supplies used in PCs today. Cost of building a device with similar capabilities to the finished design would be approximately $100-120. Full desired operation of the transceiver was defined as being able to control a multi-state controller wirelessly, receive status/sensor data wirelessly, and store data to memory within the transceiver.

I. ROUTER SELECTION AND MODIFICATION

Several manufacturers, including Asus, La Fonera, Buffalo, and Linksys all produce models that are capable of being used to create a residential telemetry network. Hardware features on certain models may make them more or less desirable for this purpose, and in some cases the router may require no hardware modification at all. A Linksys WRT54GL wireless router was chosen in this case due to its low cost, the team's experience with this router, and the large community of users and hobbyists. A number of modifications have been well documented by users and this was used to add two RS-232 ports to the router for communication with a transceiver [2]. These serial ports were added to the router by using a common MAX232 IC chip circuit to ensure a conversion of TTL voltage levels to RS-232 levels.

II. COMMUNICATION STANDARD

A number of commercial residential telemetry solutions and standards are currently available and are competing for dominance in the marketplace. The most common standards that are currently being used in home automation are X-10, Zigbee, and Z-Wave. X-10 and Z-Wave standards currently offer some of the least costly sensors and control modules [3]. Zigbee and Z-Wave are both wireless technologies that communicate in the 900 MHz ISM band in the United States while X-10 primarily communicates over existing power wiring in the home [4], [5]. Ideally, a transceiver from any of these standards should be capable of being controlled if the commands are set specifically for the standard being used. However, it should be noted that a number of Zigbee devices operate in the 2.4 GHz range and present a possibility of interfering with 802.11 if proper precautions are not taken [6]. A similar home automation design has been created using X-10 and was found after completion of the work described here [7]. Z-Wave was chosen for use in this project mainly for its focus on low power and the large availability of affordable controllers and sensors [8].

III. FIRMWARE AND SOFTWARE

Turning the wireless router into a flexible Linux computer was at the heart of making this project work. A number of embedded Linux distributions are available that suit this task, including two designed specifically for routers: OpenWRT and DD-WRT [9], [2]. Initially DD-WRT was used in hopes that its graphical web interface could be modified to display a visual layout of the user's home with controls and sensor values. Unfortunately, this version lacked software needed to adjust serial port baud rate and proved difficult to add missing commands to. This prevented communication efforts with the transceiver as a higher baud rate of 115200 was required by the transceiver. Though DD-WRT still had significant capabilities for this project, it was exchanged for OpenWRT due to a shortage of time.

OpenWRT - Kamikazee distribution, consists of a simple Linux command line interface and has software and several built-in commands that allow for programming in ASH script on the router. In addition to this, OpenWRT contains an 'ipkg' program that allows for selective installation of Linux commands that have been left out of its reduced set. Using 'ipkg', software was added that allowed for full BASH scripting and routing of serial data.

The team decided to approach design of the interface from both C and Linux script concurrently. Eventually, due to the limitations of time and the scope of the project's goals, the team decided to discard programming a GUI in C and focused on creating a simpler interface in Linux script. Though the interface would be slightly more difficult to use; functionality would be maintained. Linux scripting was created to control on/off, level set, status retrieval, and logging through the use of Z-Wave development kit documentation and serial communication capture.

SYSTEM OVERVIEW

In this system, the router remains powered on and continues to operate as a regular wireless router while maintaining necessary logging and control operations from either direct or scheduled commands. A block diagram shown below in Figure 1 provides a description of the physical layout of the router's communications. Commands are executed by sending a specific string of hex code to the Z-Wave transceiver via RS-232. This hex string contains the action to be performed and information on the destination controller or sensor as well as a specific identification number for the users home. All controllers or sensors that are not of the desired destination or of the user's home network will ignore the command or forward it until the command reaches the desired Z-Wave module. An update of the status of the module can be sent back and logged as a string of hex with date and time for diagnostic or efficiency uses on the router's flash memory, an external flash memory card, external hard drive, or possibly a web server.

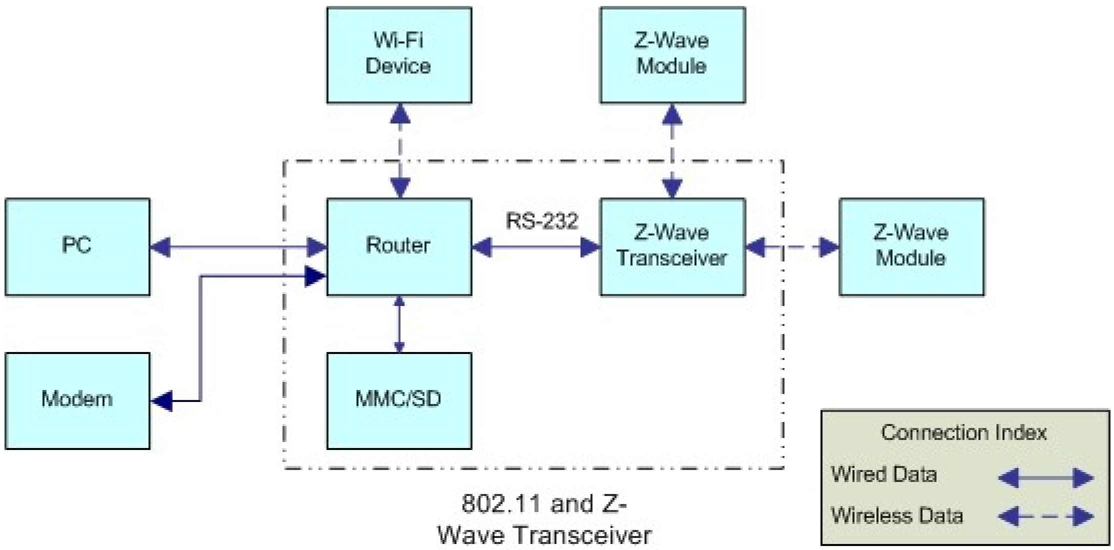

Figure 1. Router Communication

Communication to and from the transceiver is done by using a number of Linux commands through scripting to direct serial input and output. Commands can be issued directly to the router by connecting to the router through a PuTTY client [10]. Upon connecting to the router a menu can be brought up on the command line by typing at the command line “/example”. A step by step numbered menu is brought up that allows logging control, 0-100 value setting of controllers, retrieval of Z-Wave module status, and timer settings for on and off times of a module. Each command was tested and confirmed to be working using a variable intensity Z-Wave light module with results being logged to memory on the router. Though no noticeable effects were seen on the router's normal operation, more testing will need to be done to determine if use of the router as a controller for the telemetry network is detrimental to the Ethernet network's integrity and speed.

Only tests for a single Z-Wave module were tested at this time due to budget limitations, but some problems were found that will need to be resolved for a full multiple sensor network. It was found that a system or algorithm needs to be created to update the controller identification numbers if they are changed. In testing, if the controller's identification changed, communication would be lost and code would need to be changed to restore it.

CONCLUSION

As energy use and prices continue to increase so will the need for energy efficiency in the home. An open source residential telemetry network, such as the one described here can become a significant contribution by making home automation systems more affordable and commonplace. Creation of an affordable device that could be made to work with multiple standards would allow customers to use savings for more automation modules throughout their home, therefore increasing overall energy efficiency and providing greater convenience.

REFERENCES

[1]Hawking Technology Website http://www.hawkingtech.com/homeremote/features.html
[2]DD-WRT Firmware Site http://www.dd-wrt.com/wiki/index.php?title=DD-WRT_Docu_(EN)
[3]Smarthome Automation Sales Site http://www.smarthome.com/_/index.aspx
[4]Zigbee Alliance Website http://www.zigbee.org
[5]X-10 Website http://www.x10.com
[6]Zibee Alliance Website Documentation http://www.zigbee.org/imwp/idms/popups/pop_download.asp?contentID=9565
[7]Omwoyo, Justin Kerosi, "Embedded Monitoring Server," Lappeenranta University of Technology https://oa.doria.fi/handle/10024/29686
[8]Z-Wave Alliance Website http://www.z-wave.com/modules/Z-Wave-Start/
[9]OpenWRT Firmware Main Website http://openwrt.org/
[10]PuTTY Client Website http://www.putty.org/