Online Remote Control of a Wireless Home Automation Network

John Wade, University of Ulster, UK

Jose Santos, University of Ulster, UK

Noel Evans, University of Ulster, UK

ABSTRACT

Embedded systems within home appliances are not usually manufactured to operate in a networked environment; connecting supplementary hardware/software systems through a wireless, PC-controlled medium is necessary to enable full, efficient control of their functions from a remote location. Access to the home's central PC may be gained via a local web server, giving Internet-based control from almost anywhere in the world. The proposed system constitutes a significant improvement over those discussed in the literature to date, and reviewed here. It enables complex-appliance control in a secure and reliable portable-wireless environment, and was developed using ASP.Net. The system was assessed for Received Signal Strength (RSS) in an environment more radio-hostile than that found in a typical household. The minimum RF level found at a transfer rate of 9.6 kbps was 8 dB above the receiver's quoted sensitivity of -103 dBm; this fading margin will increase in a normal household environment. [Article copies are available for purchase from InfoSci-on-Demand.com]

Keywords: Home Automation Network; Microcontrollers; Wireless Sensor Network

INTRODUCTION

As the cost of electronics components and microprocessors decrease, they are embedded increasingly into every day household items, e.g. microwave ovens, toasters, televisions and washing machines. Unfortunately it is still

not possible to communicate with these appliances via a network, be that from within or outside the home. Although the appliances have embedded microprocessors to help control their onboard systems, they do not have the capability to communicate with a network of any kind. This is because there are no communication devices pre-installed and as yet no clear leader exists in home automation network protocols. Until this happens, true home automation networks cannot be fully realised. This, however, does not limit the ability to remotely control appliances in the home. To do this, domestic appliances can be grouped as:

1. Appliances that can only be turned on and off (Group 1).
2. Appliances that can be controlled in a more functional manner, such as televisions, DVD players and HiFi equipment (Group 2).

There are many reasons why the ability to remotely control the home is an advantage. In modern times, security and time management have become important issues. Houses may give an impression of occupation through controlled lighting and have their alarm system monitored; kitchen tasks may be started remotely from a work location, saving time after the daily commute. A control network could also give the elderly and disabled a more independent way of life.

In section two, a background review of previous research into this topic is presented. Section three discusses the hardware design for the system investigated. Section four explores the software design for both the Internet interface and the firmware for the microprocessors. Section five discusses the testing protocol and results obtained from the system. Finally, section six presents conclusions and suggestions for future work.

COMPUTER CONTROLLED HOME AUTOMATION

There have been several approaches to computer controlled home automation: what follows is a review of the most significant and recent contributions.

Al-Ali A.R. & Al-Rousan M. (2004) developed a system built from a Java based user interface that was accessible through the World Wide Web. The system allowed the user to turn a cooker, light bulb or a fan on and off. This system used direct wiring from the computer to the appliances to exercise control over them: this approach is very costly due to the amount of cable needed to connect all the appliances found in a modern house. Also, appliances that fall into Group 2 could not be implemented in this system and introduction of new appliances into the household will result in further integration costs.

Sriskanthan N., (2002), proposed a system that controlled home appliances from a PC using Bluetooth as the communications technology. As before, there was no consideration for

those appliances that fall into Group 2 and could not be controlled over the Internet. By using Bluetooth there is also a limit to the number of appliances that can be controlled. According to Stallings (2002), up to eight devices can communicate in a small network called a piconet, and up to ten piconets can coexist in the same coverage range of a Bluetooth radio. This permits up to 80 appliances to be connected. Although a significant number at present, in the future this may become a limitation on the system. Another disadvantage of Bluetooth is that it allows ad-hoc networking. This enables an appliance that is Bluetooth enabled to establish an instant connection with another Bluetooth enabled appliance in the network when it comes into range. The main implication of this statement is that as new appliances are introduced into the home, the PC software has to be able to identify the appliance and either accept it or reject it automatically.

Bigio P. & Cucos A. & Corcoran P & Chahil C & Lusted K. (1999) designed and implemented a non-standard means of wireless networking suitable for home automation. Their system consisted of a low-power RF network broadcasting on 433 MHz, which again is license-free as it falls in the Industrial Scientific & Medical (ISM) band according to the European Radio Commission Committee (2005).

The system is based on a master/slave topology and is composed of a base station, relay units, and slave units. The base distinguishes between application requests and functions: application requests remain transparent to the radio system and therefore only the function of the appliance is broadcast. These broadcasts consist of frames/packets which have headers, body, trailer and a block/frame check for error checking. The relay units are used to extend the range of the base station, which has a range of 150 m. This range is sufficient in most homes and therefore the relay units are not normally needed. The slave units are used to relay the information provided by the base station to their appliance. The network manager software controls all communications of the Radio Frequency (RF) network; it runs as a background service on the PC and is also responsible for granting or denying access to recently arrived slave units, depending on the unit being a legal part of the network. This system, whilst providing clear advantages over those previously proposed, still does not provide support for control over the Internet. However, it does show how useful and powerful using a non-standard RF network for the communications medium can be, by allowing the designer full flexibility over the communications protocol.

Thomas R. (2000) designed a system that could decode signals from remote control units using the RC5 protocol, which is used by Philips, and the SIR C protocol used by Sony. This work highlights the problems that arise when trying to capture, decode and recreate remote control protocols. This system only dealt with two of the more

common protocols: there are numerous others from a variety of manufacturers. Trying to produce a system that can recreate all protocols is not feasible. for two reasons. Firstly, from an economic point of view. a very wide range of remote controls from various manufacturers would need to be available so that signals may be captured and recreated. Secondly, the time needed to do this accurately is problematic; other workers dedicate much effort in this area [ref: Innotech Systems]. Therefore, it was considered more practical here to adopt a pre-designed universal remote control in the design.

Guan R. & Pruehsner WR & Enderle JD (2000) designed a system which used a GUI (Graphical User Interface) running on a PC to address the control problem presented by appliances that fall into Group 2. User input from a graphical representation of a remote control in the GUI sent data wirelessly to a PIC microcontroller via the serial port of the PC. This data was then transformed into special control commands forwarded to a PP4001 universal remote control IC: this is preprogrammed to control most brands of appliances that have Infra Red (IR) remote controls. Although this approach negates the need to design a custom decoder, again there was no facility for control over the Internet and the PP4001 is expensive.

By extending the elements discussed above, it became possible to implement a system that is capable of controlling both groups of appliances over the Internet, thus providing a reliable and secure connection to all appliances within a wireless home network from almost anywhere in the world.

## HOME AUTOMATION NETWORK HARDWARE IMPLEMENTATION

The Home Automation Network is broken into 3 different sub-systems: the control unit (Base Unit), the Group 1 receiver units (GP1 Units), and the Group 2 receiver units (GP2 Units).

Base Unit

As shown in Figure 1, the Base unit is the main control center for the system; it communicates with the host PC (home

Figure 1. Base unit block diagram
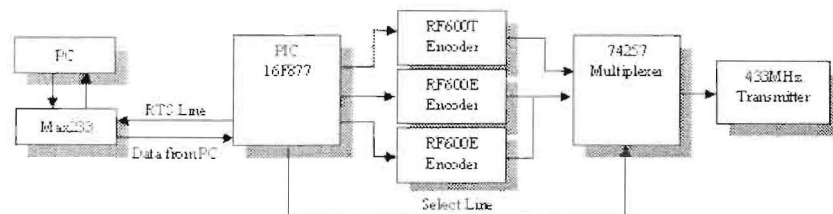
PC) via a serial port connection. The main computational element that controls communications is a PIC 16F877 microcontroller, which has a USART (Universal Synchronous Asynchronous Receiver Transmitter) built in. This receives data from the PC and is also responsible for RTS (Request to Send) handshaking between the Base unit and the PC. This ensures that no data is lost during communications due to buffer over-runs. All communications between the microcontroller and the PC use a MAX233 as an intermediate signal level converter.

Once data has been received by the microcontroller, it verifies that a full packet is present and the data is correct; data and packet structure will be discussed further in section 4. Once deemed correct, the microcontroller decides which group of appliances the data is intended for and processes it accordingly, either passing it to the Group 1 encoders or sending it through the USART to the Group 2 encoder, and on to the transmitter.

If the microcontroller decides that the data is intended for the Group 1 appliances, it firstly selects the appropriate input line of a digital multiplexer (Mux); this ensures that only data from the Group 1 encoders is sent to the transmitter. Then the data is output on ports A and C of the microcontroller, which are connected to the Group1 Manchester encoders (RF600E). These communicate with 4 decoders (RF600D), one on each GP1 unit. Matching the encoders to the decoders ensures that no other

appliances outside of the system can be activated unintentionally and aids system security. The new encoded data is then sent via the Mux to the RTFQ1 (433 MHz FSK transmitter) [rfsolutions]; this has an estimated radiated power of 5 dBm when using a $\lambda/4$ antenna. The companion receiver has a sensitivity of -103 dBm for 9.6 kbs operation. This implies that a maximum path loss of 108 db may be tolerated.

If the microcontroller decides that the data is intended for the Group 2 appliances, then once again the correct input line to the Mux is chosen and the data is relayed via the USART to a serial encoder (RF600T). This is a serial version of the previous encoder which also uses Manchester encoding. The Encoder encapsulates the data into its own transmission packet, which consists of a 76 bit preamble, a 4-bit sync pattern, 1 start bit, 8 command bits, the data bits (size depends on the data sent from the microcontroller), and 8 checksum bits. This ensures that all data is received correctly by the decoder (RF600T) on the GP2 unit. This new encoded data is then sent to the FSK transmitter.

GP1 Units

The GP1 unit block diagram is shown in Figure 2. Each of the units is responsible for receiving control information from the Base unit and controlling the Group 1 appliances connected to them. Each GP1 unit can control 1 appliance.

When data is received by the FSK receiver the RF600D decodes it; if the

Figure 2. GP1 unit block diagram
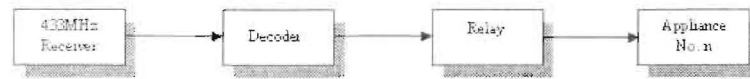
data is intended for that unit then the decoder activates or deactivates an isolating relay, which in turn controls the relevant AC mains-connected appliance. The decoder has two modes of operation, momentary and latched: here the latter mode is used. Each time data is received and verified for that specific decoder the output of the decoder is switched to the opposite state, i.e. if it was 'on' then it will be switched to 'off', and vice-versa. This implies that the same data is transmitted for on and off. This element is part of the software design and will be discussed in section 4.

GP2 Units

Figure 3 shows a GP2 unit, responsible for receiving control information from the Base and controlling the appropriate Group 2 appliance remote control. Each GP2 unit can control one remote.

When data is received by the FSK receiver the decoder (RF600T) ensures steering to that particular unit, and on to a PIC 16F877 microcontroller for further verification that it originated from the registered Base. If this is so, the PIC outputs the data on ports B and D. These are connected to 4 separate 4016 quad bilateral switch banks. These switches are in turn hardwired to a Philips RU252 Universal Remote Control, which has the ability to control both televisions and VHS/DVD players/recorders from a wide number of manufacturers: up to 16 buttons on the remote may be controlled. In the prototype 15 were used for proof of concept. All 32 remote switches could be controlled with a few changes to the software and the addition of another 4 4016 devices.

HOME AUTOMATION NETWORK SOFTWARE IMPLEMENTATION

The software required to control the system was broken down into three different programs: to enable communications to the home server via a website, to control the microcontroller on the Base

Figure 3. GP2 unit block diagram
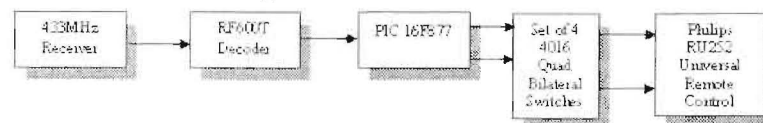
unit, and to control the microcontroller on the GP2 units.

Website Software

The website used to access the system from a remote location is programmed in ASP.Net which forms part of the Visual Studio.Net package and the .Net framework. It allows the creation of Web applications and services that run under IIS (Internet Information services). IIS host Web applications on the Windows server, which in this case was based on the home PC. It manages the application, passes requests from clients to the application, and returns the applications responses to the client. ASP .Net is one of the most complete platforms for developing Web applications, making creation, debugging and deployment straightforward, as noted in Webb J. (2002).

When the user first accesses the website the Login screen is presented. Here the user is required to enter a user name and password; authorized user information is stored in a configuration file on the server. If the user is verified he / she is redirected to the home page, shown in Figure 4. When the Home page is loaded it accesses a database on the server that contains the current state of the Group 1 appliances on the system and displays this information under the Current State check boxes. If a check box is ticked then the appliance is "on" and vice-versa. The user can then update the state of the appliances by making adjustments to the New State check boxes. Once the user is satisfied with the changes the Update Appliances button is pressed; this sends the New State check boxes information back to the server, which then calculates two sets of data from this information.

Figure 4. System home page
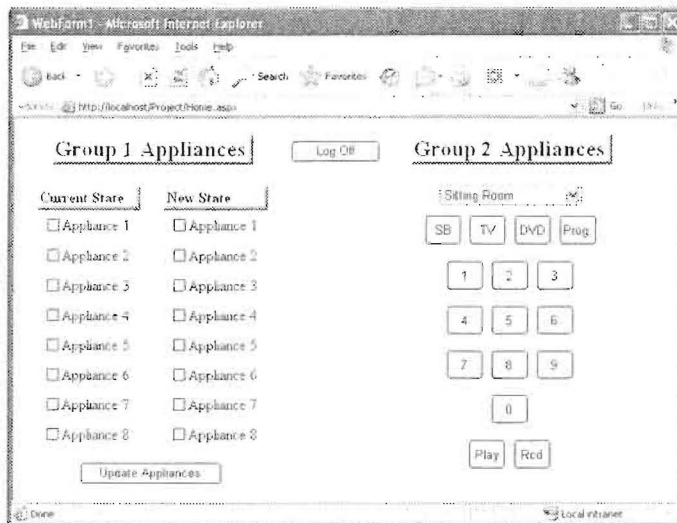
Firstly, the New State information needs to be converted into two numbers, each set ranging from 0 to 16 i.e. a 4-bit nibble. This is carried out as follows: appliances 1 to 4 represent the first set of data where appliance 1 is the least significant bit and appliance 4 is the most significant bit. If an appliance is set to "on" then the corresponding bit is set, otherwise it is cleared. The same procedure is carried out for appliances 5 to 8. Once this set of data has been generated it is the necessary to produce another two numbers from the Current State check boxes. The two sets of data are then XOR'ed with one another and the result stored as Function bytes 0 and 1. This ensures that if an appliance changes state then the corresponding bit for output will be set, allowing the Base unit to drive the encoders correctly, as discussed in section 3.1.

This data is then sent to the serial port where it is added to the transmission packet and transmitted to the base. This packet is made up of a start byte = 0, address byte = 1 and the two function bytes as shown in Table 1. The database is updated with the New State checkbox information and display in the Current Status check boxes on the refreshed page.

The user also has the option of controlling a Group 2 appliance via the onscreen remote control, by firstly selecting the area where the remote control is located from a dropdown list box. The address of the remote is defined by adding two to the index of the area selected in the list and is stored. When the user presses a button the address is sent to the serial port and forms part of the transmission packet. Each button has its own function defined for controlling the remote control and the two function bytes are exclusively defined within these functions. The Function 0 byte controls the first 8 switches and the Function 1 byte controls the next 8 switches. The screen is then refreshed. The transmission packet is similar to the transmission packet for the Group one appliances. Table 2 shows an example of the sitting-room remote control being accessed with the number 1 button pressed. Figure 5 shows the two function bytes and the corresponding switches that each bit controls.

If the Program button is pressed then buttons 1 and 3 are simultaneously accessed on the remote control; this allows the remote to enter program mode where a new device number may

Table 1. Transmission packet from server to base unit

Start Byte	00000000
Address Byte	00000001
Function Byte 0	0000 to 1111
Function Byte 1	0000 to 1111

Table 2. Transmission packet for GP2 Unit

Start Byte	00000000
Address Byte	00000010
Function Byte 0	00010000
Function Byte 1	00000000

ption of
iance via
by firstly
e remote
down list
is defined
f the area
ed. When
address is
ns part of
ch button
l for con-
d the two
y defined
unction 0
tches and
the next 8
refreshed.
ilar to the
iroup one
example
ontrol be-
r 1 button
o function
switches

s pressed
imultane-
e control;
r program
mber may

t for GP2

Figure 5. Function bytes and their corresponding switches

	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Function Byte 0	SB	TV	DVD	0	1	2	3	4
Function Byte 1	5	6	7	8	9	Play	Record	NA

be added and hence allow the remote to control a new appliance. On completion, 'Log off' permits a return to the Login screen.

Base Unit Microcontroller Firmware

When the microcontroller is started it clears the Request to Send (RTS) line leaving it ready for reception and waits until a byte has been received. The RTS line is then set, stopping all transmission from the PC. If this byte is 0x00 then it stores it as the Start byte, clears the RTS line and waits for the next byte, otherwise the byte is ignored, the RTS line cleared and the next byte checked for 0x00. The process continues with the next three bytes stored as Address, Function byte 0, and Function byte 1 respectively. When the complete transmission packet has been received the program then checks which group of hardware has been addressed by subtracting 1 from the Address byte. If the result is 0 then Group 1 has been addressed, otherwise the system is addressing a Group 2 appliance.

If Group 1 has been addressed then Function byte 0 is passed to Port A which controls the first encoder, and the Mux line is cleared. A short delay of 0.5 s is called to allow the transmission of the

signal, after which the port is cleared. Function byte 1 is then output on Port C and another delay of 0.5 s is called before the port is cleared.

If Group 2 has been addressed then the Mux line is set, the Start byte, Address byte, System Identification Number (SID) and the two function bytes are transmitted via the onboard USART to the serial encoder chip and then onto the RF transmitter. The SID number is pre-programmed into the microcontroller and consists of 4 bytes that provides up to 2^{32} different combinations. It is used as a unique system identifier for all the Group two remote controls. This ensures that only GP2 units registered with the same SID number as the base unit can be controlled by that base unit, thus preventing another system from accessing the GP2 units.

GP2 Unit Microcontroller Firmware

The start of the GP2 microcontroller program is similar to that in the Base unit, but with 4 extra bytes of data to be received before a complete transmission packet is collected: these are stored as SID 0 to 3. When the full packet has been received the microcontroller verifies that it has been addressed by subtracting the pre-programmed address from

the address received. If the result is 0 its carries on, otherwise the packet is ignored and the microprocessor resets, ready for another transmission.

The next step is to verify that the SID number received matches the pre-programmed SID in memory: if so it continues, otherwise it again resets ready for another transmission. Once it has verified that the data is intended for the GP2 unit it checks if the program button was pressed on the website. This is done by subtracting the function 0 byte from 0x50. The result of pressing the program button is that Function byte 0 = 0x50. If the result is 0 then it enters the Program routine. Here Function byte 0 is passed to Port B and a delay of 3 s is called before clearing the port. This activates buttons 1 and 3 on the remote control and makes it enter programming mode. If the result is not 0 then "Function 0" byte is passed to Port B for 0.5 s. and the port cleared. This is followed with Function byte 1 being passed to port D for 0.5 s before the port is cleared. Ports B and D are connected to quad bilateral switches that are hardwired to the remote control. In total. 253 GP2 units can be addressed by the Base unit.

SYSTEM TESTING

Hardware tests were carried out using a program written in VB6 that allowed the Start, Address, and Function bytes to be explicitly entered as numbers before accessing the serial port. Transmission range in a hostile radio envonment was evaluated and software tests were carried out on the website.

Transmission Characteristics Test

Transmission range was tested by locating the transmitter and receiver units along different paths with a variety of obstacles, giving a significantly more radio-hostile environment than that found in a typical household, and measuring the Received Signal Strength (RSS) in each case. Figure 6 shows the location of the transmitter and receivers, along with the relevant obstacles in the path. The walls and columns are made of reinforced concrete with metal platting. The floor and ceilings are made of concrete. the overhang ceiling is made of mineral fiber tiles, the stud wall is made of a wooden frame covered in gypsum plaster board, and the elevator shaft wall is a red-brick construction.

As noted in Table 3, the received input level never fell below -94.2 dBm, giving an 8.8 dB margin above the manufacturer's quoted sensitivity of -103 dBm, for a transfer rate of 9.6 kbps. The fading margin in a normal domestic situation (with thinner concrete absorbers) is expected to be significantly greater.

Hardware Test & Software Tests

To test the hardware, a prototype system was constructed. Three GP1 units were implemented using two encoders; the

onment was
ts were car-

eristics

; tested by
nd receiver
with a vari-
significantly
nment than
isehold, and
;nal Strength
6 shows the
nd receivers,
stacles in the
ins are made
h metal plat-
; are made of
ng is made of
wall is made
d in gypsum
tor shaft wall
ı.

the received
ıelow -94.2
ıargin above
d sensitivity
er rate of 9.6
a normal do-
ıner concrete
: significantly

ware Tests

otype system
P1 units were
encoders; the

Figure 6. Transmitter and receiver locations for RSS evaluations
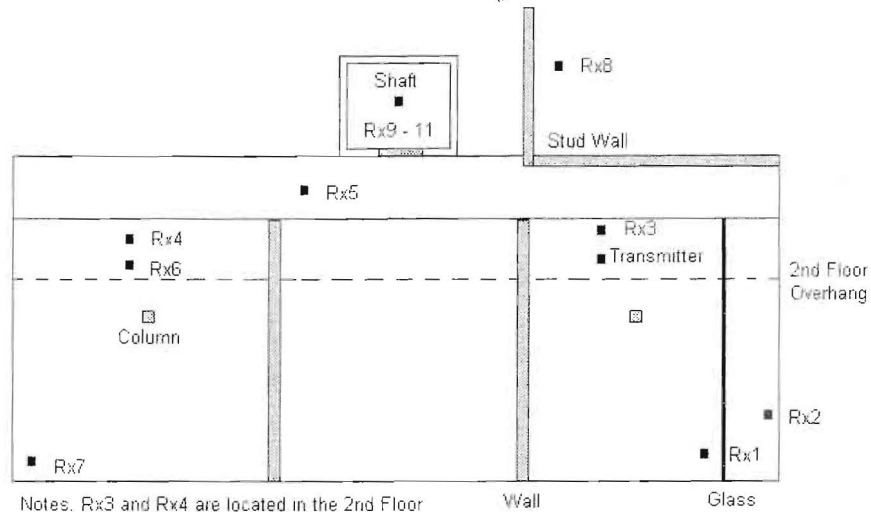
Notes. Rx3 and Rx4 are located in the 2nd Floor
Rx9, Rx10 and Rx11 were located inside an
elevator with the doors closed and located at
level, 1 and 2 floors down with respect to the Tx

Table 3. Received signal strength measurements

R x No	RSS (dBm)	Propagation Path	Direct Path Length
1	-57.52	Around Square Column (x m²)	9 m
2	-52.75	Through Glass Panel (direct Line of Sight)	10 m
3	-53.32	Through Overhang Ceiling	3 m
4	-88.09	Overhang Ceiling and 2 Walls	17 m
5	-65.15	Through Wall	11.5 m
6	-72.89	Through 2 Walls	19 m
7	-85.75	Through 2 Walls	23 m
8	-66.80	Through 2 Stud Walls	10 m
9	-93.30	Elevator Shaft	13.5 m
10	-93.38	Elevator Shaft – 1 Floor Down	14.2 m
11	-94.20	Elevator Shaft – 2 Floors Down	16.2 m

first encoder addressed 2 units, and the
second addressed one unit. The intention
was to prove that more than 1 decoder
could be addressed by the same encoder
and operated independently, and that
the signal from the two encoders were
separate, i.e.: if the first appliance was
operated by the first encoder then the
first appliance on the second encoder
should not be operated at the same time
as the signals should be different. The
GP1 units were operated individually
and then simultaneously by sending

the appropriate commands from the test program.

During testing, the only problem that occurred was that if two sets of packets were sent one immediately after the other, one of the two encoders would lock up; this resulted in the base unit needing to be reset to correct the problem. However this will not represent a problem when the system is controlled through the website as it provides adequate time between packet transmissions due to page loading and refreshing.

The prototype also included a single GP2 unit. Correct operation was confirmed using a test program and the unit was also tested to verify that it would not respond to an address that was different to that stored in memory and that it would not respond to a packet with the incorrect SID number. This last test was achieved by replacing the microcontroller in the base unit with another one with a different SID number.

The web-based application program was first tested in a stand-alone mode for access security and for correct updating of the database with the "New States" when the "update" button was pressed on the application; then the prototype was fully connected to the host-PC and full control was achieved from the web-server.

CONCLUSION

The system developed is a very powerful home automation network that can be accessed from remote locations through the World Wide Web and operates in a wireless environment within the home. It consists of a website which is located on a home server and associated hardware, also connected to the server. The hardware is comprised of a Base unit that receives signals from the website and relays them on to the receiver units which are connected to the appliances. At present the system can control 8 appliances that can be turned on and off (GP1) and 253 appliances that are accessed via a remote control (GP2). Only minor software changes are needed to permit significant expansion. The system can operate in radio-hostile environments while maintaining a data rate of 9.6 kbps.

Although the system as described is not a fully-definitive home automation solution, it does serve as useful prototype. There are many improvements that could be made e.g.: the implementation of duplex communications between the Base and receiver units. The receivers could then inform the Base, and ultimately the user via the Web, that commands have been received and executed. This negates the need for the database as the receiver units could relay their status when the website is accessed. A related improvement is to grant users local control of the Group 1 appliances, with duplex links updating their Current State resulting from local switch changes within the home. The website program might also be enhanced to include a plan/layout of the home where each controlled appliance

locations
and oper-
ent within
site which
nd associ-
ted to the
iprised of
nals from
on to the
inected to
ystem can
be turned
ippliances
te control
hanges are
xpansion.
lio-hostile
iing a data

escribed is
itomation
ful proto-
ments that
mentation
tween the
ie receiv-
Base, and
Web, that
ived and
need for
nits could
website is
ment is to
he Group
iks updat-
ting from
the home.
t also be
out of the
appliance

is shown in its proper physical location; this decreases search time on user access by eliminating the need to search through check boxes and dropdown lists. Finally, the system developed can save time, help improve home security, and in general bring a better way of life to the end user.

REFERENCES

Al–Ali, A.R., & Al-Rousan, M. (2004). Java-Based Home Automation System. *IEEE Transactions on Consumer Electronics, 50*(2), 498-504.

Bigioi, P., Cucos, A., Corcoran, P., Chahil, C., & Lusted, K. (1999, August). Transparent, Dynamically Configurable RF Network Suitable For Home Automation Applications. *IEEE Transactions on Consumer Electronics, 45*(3), 474 - 480.

European Radio Commissions Committee (2005) Frequencies and Standards. (online) http://www.ero.dk/doc98/official/ pdf/ REP083.PDF#433050000. Accessed 5th July 2005.

Guan, R., Pruehsner, W.R., & Enderle J.D. (2000, April). The Computerized Environmental Remote Control. *Proceedings of the IEEE 26th Annual Northeast Bioengineering Conference* (pp. 147–148).

RFSolutions. (2006) *RTFQ1 Data Sheet.* (on-line) http://www.rfsolutions.co.uk/ acatalog/DS069-7.pdf

Sriskanthan, N., & Tan, K. (2002). Bluetooth Based Home Automation Systems. *Journal of Microprocessors and Microsystems, 26,* 281-289.

Stallings, W. (2002). *Wireless Communications and Networking.* New Jersey USA: Prentice-Hall.

Thomas, R. (2000, September) Remote Control IR Decoder. *Everyday Practical Electronics, 29*(9), 698–701.

Webb, J. (2002). *Developing Web Applications With Microsoft Visual Basic .Net and Microsoft Visual C#.Net.* Washington USA: Microsoft Press

ilobai

John J Wade graduated from the North West Institute of Further and Higher Education (N.W.I.F.H.E.) in 2001 with a BTec HND in electrical and electronic engineering. Wade is the holder of a first class honours degree in electronics and computing, with diploma in industrial studies (DIS), which he obtained from the University of Ulster in 2004. In 2005 he obtained an Msc in computing and intelligent systems from the University of Ulster. Wade is currently undergoing a PhD in intelligent systems at the University of Ulster's Intelligent Systems Research Centre (ISRC) where he is focusing on developing a biologically inspired training algorithm for spiking neural networks. As well as neural networks Wade's main areas of interest are, wireless communications, and intelligent systems.

Jose A. Santos obtained his electronic engineering degree from the Universidad Simon Bolivar in Caracas, Venezuela in 1998, his PhD in electronic engineering from the University of Ulster, Northern Ireland, UK in 2002, and a PgCert in higher education teaching from the University of Ulster in 2003. He is a lecturer in the School of Computing and Intelligent Systems at the Magee Campus of the University of Ulster since 2002. He is also a member of the Ambient Intelligence Research Group which is part of the Intelligent Systems Research Centre (ISRC) at the University of Ulster. His research interests lien in the area of ambient intelligence, RF and wireless systems, sensor networks, biotelemetry and biomedical systems.

Noel E. Evans received the BSc degree in 1973 and the MSc degree in 1974 from the Queen's University of Belfast, Northern Ireland, UK. He obtained a PhD degree from Queen's in 1977 for work on programmable transversal filters built using intracell charge-coupled devices. Following further research into narrow-gate CCDs at Queen's, he taught analogue electronics and communications at the University of Ulster from 1980-2006. Here he also developed research programmes into biomedical instrumentation, human and animal physiological signal acquisition using radio frequency techniques, and electromagnetic wave propagation at frequencies extending across the HF-UHF bands. He now works as an independent consultant in communications and telemetry.