ISBN: 978-972-8924-62-1 © 2008 IADIS

SOFTWARE X10-UPNP BRIDGE

M.J. Chico, J.M. Maestre, E.F. Camacho
Departamento de Ingeniería de Sistemas y Automática
Universidad de Sevilla
Camino de los Descubrimientos s/n, 41092, Sevilla, Spain

ABSTRACT

In a future of smart houses where multiple heterogeneous devices will be interconnected to provide new functionalities and to ease everyday tasks, several issues will challenge the research, including interoperability and interfaces between different systems. The work presented in this paper describes a software bridge that exposes X10 devices as UPnP devices to UPnP control points, banking on the networks integration and alternatives to create mixed installations that are optimized for ease, reliability and functionality.

KEYWORDS

Home Automation, X10, UPnP, interoperability, intercommunication.

1. INTRODUCTION

Home Automation offers the users the dream of an intelligent home that automatically controls the living environment and responds to individual preferences. However, high cost, limited capability, and a lack of standards have imposed major constraints on the market, keeping home automation more in the realm of imagination than practice.

To provide more comprehensive control functions for home use, a set of standards has been developed for a common command language and a series of software applications that would handle a variety of devices. The effort also defined communications methods for many media, including twisted-pair wiring, infrared, RF, and power-line signaling. One of these standards is UPnP (*Universal Plug and Play*).

With all the interest generated by the interoperability between networks, in this paper we'll focus on the design for an UPnP-X10. This bridge can both send and receive X-10 signals and offers a transparent link between both the communication protocols, in such a way that the X10 system is shown as a one more UPnP device in an UPnP environment.

The paper is organized as follows: sections 2.1 and 2.2 describe the major characteristics of the two protocols involved in the project. Section 2.3 describes relevant related works on bridging in domotic systems. Section 2.4 describes the basic principles and the assumptions on which the work is built. Section 2.5 details the architecture of the proposed application and the uses it can offer. Eventually section 3 drives conclusions and proposes some future investigations.

2. BRIDGE DEVELOPMENT

2.1 X10

The X10 is a power-line-signaling technology which Scotland's Pico Electronics first developed in 1975 designed for sending signals over 120 VAC wiring. X10 uses 120 kHz bursts timed with the power line zero-crossings to represent digital information. To be compatible with other X-10 receivers, the maximum delay from the zero-crossing to the beginning of the X-10 envelope should be about 300 μs.

The X10 control system sends data at 1 bit/8.33 msec, is limited to 16 commands, and can control a maximum of 256 devices in a single network. Despite these limitations, X10 products have enjoyed limited but continuous success in the market and are still available for consumer purchase and installation.

Regarding the protocol, X10 became a de-facto standard in 1997. A complete code transmission encompasses eleven cycles of the power line. The first two cycles represent a Start Code. The next four cycles represent the House Code and the last five cycles represent either the Number Code (1 to 16) or a Function Code (On, Off, etc.). This complete block, (Start Code, House Code, Key Code) should always be transmitted in groups of 2 with 3 power line cycles between each group of 2 codes. Bright and dim are exceptions to this rule and should be transmitted continuously (at least twice) with no gaps between codes.

HOUSECODE	DEVICE CODE	FUNCTION	BINARY	HEX
A	1	All lights off	0110	6
B	2	Status Off	1110	E
C	3	On	0010	2
D	4	Pre-set Dim(1)	1010	A
E	5	All Lights On	0001	1
F	6	Hail Acknowledge	1001	9
G	7	Bright	0101	5
H	8	Status On	1101	D
I	9	Extended code	0111	7
J	10	Status Request	1111	F
K	11	Off	0011	3
L	12	Pre-set Dim(2)	1011	B
M	13	All Units Off	0000	0
N	14	Hail request	1000	8
O	15	Dim	0100	4
P	16	Extended Data Transfer	1100	C

Figure 1. X10 Power Line Cycles and X10 House Codes and Key Codes

For further information about the X10 signals and the protocol, see the references [11] and [12].

2.2 UPnP

UPnP is an architecture for network connectivity of intelligent appliances, wireless devices, and PCs of all form factors. It is designed to bring easy-to-use, flexible, standards-based connectivity to unmanaged networks whether in the home, in a small business, public spaces, or attached to the Internet. UPnP is a distributed, open networking architecture that leverages TCP/IP and the Web technologies to enable seamless proximity networking in addition to control and data transfer among networked devices in the home, office, and public spaces. The UPnP™ Forum [7] is an industry initiative which consists of more than 800 vendors, including industry leaders, which work defining and publishing UPnP device and service descriptions.

UPnP is more than just a simple extension of the plug and play peripheral model. It is designed to support zero-configuration, "invisible" networking, and automatic discovery for a breadth of device categories from a wide range of vendors. This means a device can dynamically join a network, obtain an IP address, convey its capabilities, and learn about the presence and capabilities of other devices. Finally, a device can leave a network smoothly and automatically without leaving any unwanted state behind.

UPnP uses open, standard protocols such as TCP/IP, HTTP and XML. IP internetworking is a strong choice for UPnP because of its proven ability to span different physical media, to enable real world multiple-vendor interoperation. UPnP has been explicitly designed to accommodate these environments. In the same way, UPnP devices can be implemented using any programming language, and on any operating system.

The UPnP interaction process between devices and Control Points consists of five phases. Each device or Control Point compliant with the protocol must implement at least the four lower phases which are Discovery, Description, Control, Eventing, and Presentation.

2.3 Related Works

Nearly Thirty years ago, there was only one technology for home automation (X-10) and a very limited number of companies who made devices for this 'defacto' standard. Today, newer technologies (like Z-Wave, UPB and Insteon) have joined X-10 and there is a bewildering array of devices for the home owner to choose from. As with most emerging technologies, however, comes the issue of compatibility. Several new protocols have entered the market, giving home automation system users and enthusiasts a rich set of protocols to choose from. The theoretically "Best" protocol choice for a system is often, or even usually, to mix technologies rather than to implement a single protocol [4,5,8,9].This has resulted in the need to easily bridge between these alternatives to create mixed installations that are optimized for ease, reliability, affordability, and functionality.

At first glance, coming to a consensus on methods for ensuring easy interoperability of multiple formats seems to be the only way to make the converged home network a reality. Content providers, software writers, OEM's and ODM's need to work towards enabling such a vision. That's where organizations like the Digital Living Network Alliance (DLNA) [3], the Wi-Fi Alliance [10], and the Consumer Electronics Association (CEA) are working hard for interoperability [2]. Another chance to integrate system is being developed in the U.S. with SCP[6] (*Simple Control Protocol*).

In the same line of enquiry it has been developed a project PLC-based Home Automation System Completed [1] in which it's developed an UPnP X10 bridge by hardware. In this paper we do it by software, which offers more flexibility.

2.4 UPnP X10 Bridge Design and Implementation

As said above, this application note discusses the implementation of a bridge to offer the possibility to intercommunicate two home automation networks, on one hand X-10, and on the another UPnP.

From the point of view of an X10 user, this software enables to interpret the signals received by the CM11 – a computer interface intelligent controller- through the electrical installation and also to transmit electrical signals using the X-10 protocol. As well, the user will be able to program different macros, i.e. groups of events programmed by the users according to their preferences or needs. By UPnP enabling common X10 devices, the bridge expands the reach of UPnP technology into the pervasive world of non-IP-based everyday devices.

And as far as UPnP environment is concerned, the bridge will offer to any UPnP user the capability to interact with the X10 system and with all the possibilities and functions it offers.

All we need is a CM11, a computer interface connected and controlled by the serial port of that computer, and all the X10 electronical devices connected to the wiring, we will be manage to communicate and send orders to them. Thanks to the bridge, any UPnP control point will be able to control X10 devices (invoking their actions and checking the value of any state variable which it is interested in) and subscribe to all the events generated by the services of these devices, due to changes in any of its state variables.

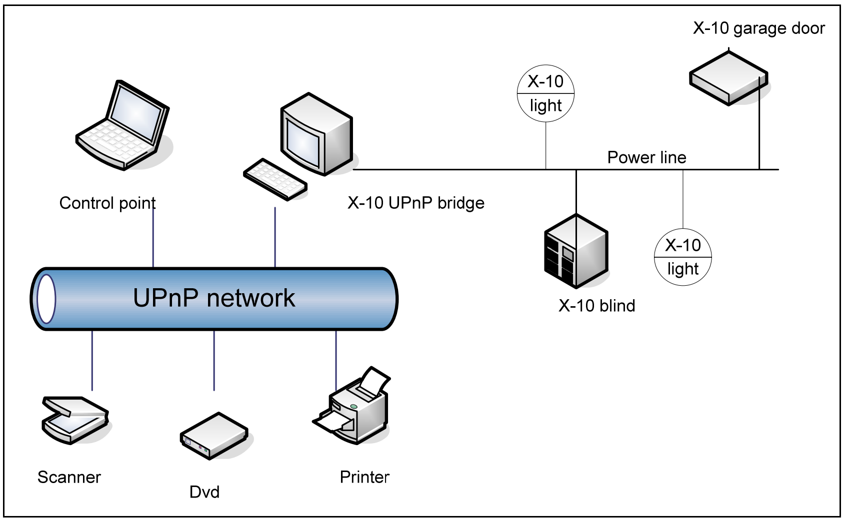

Figure 2. X10 UPnP Bridge

Taking this into account, the UPnP-X10 Bridge Bundle consists of the following services:

- Transparent link between the two networks, so that any UPnP control point can detect the X10 devices as UPnP ones.
- Send and receive commands to the X10 devices through the CM11 and the electrical line.
- Receive events from the wiring and report to the user, both X10 and UPnP.
- Translate the UPnP orders sent by an UPnP control point into X10 protocol, in such a way that can be sent through the wiring, using the house code and device code.
- When the state of an X10 device is modified –either by a X10 user or a remote UPnP one-, report the changes to the both networks.

The bridge creates some variables where the values of the house code, device code, function and dim value are stored. In that way, the bridge translates from one protocol to another to make the communication possible, both from the X-10 network to the UPnP one, and in the other way. The application has been implemented in .NET framework, in Visual C#, and is intended to be used for the Windows platform.

2.5 UPnP X10 Bridge Operation

When starting this application, we are shown a window which let us configure the serial port parameters, such us the port name, the baud rate and the parity, as illustrated in figure 3. Once the port is open, the user can send the electrical grid all the X10 orders he wants, using the house and device code, and he will hold a dialogue with the CM11.

Another great interesting possibility the application offers is the programming of macros. A macro is a series of X10 commands forwarded in sequence to other X10 modules (such us lamps and appliance modules). The sequence can include a time delay. For example: you are sitting at home, about to watch a video. By pressing one button on a hand-held remote, your stereo turns on, the overhead light turns off and the lamps at the back of the room fade down to a comfortable level - just right for watching a good movie. All that was accomplished with one X-10 command. You have just experienced a macro in action.

The CM11 only has 1Kb of EEPROM for storing macros and schedules, so you must be thrifty when deciding what to download. Macros are really quite simple to set up. The ‘Create Macro’ key shows a new window, as shown in the figure 4. There are two kinds of macros, as they can be triggered by either an external event every time the user wishes or by a programmed timer.

In the case of a timer, all the user has to do is select the time and the dates when macros have to be executed, and add the actions associated to it, to program timed schedules for your lights and appliances and also create X-10 macros. As illustrated in figure 5, the way to associate an event to a macro is very simple. Apart from the housecode, device code and function, the user has to decide the offset of time the actions are going to be separated. As well, it can be decided if the function dim or bright are absolute or relative, by marking or not the 'Brighten first' button.

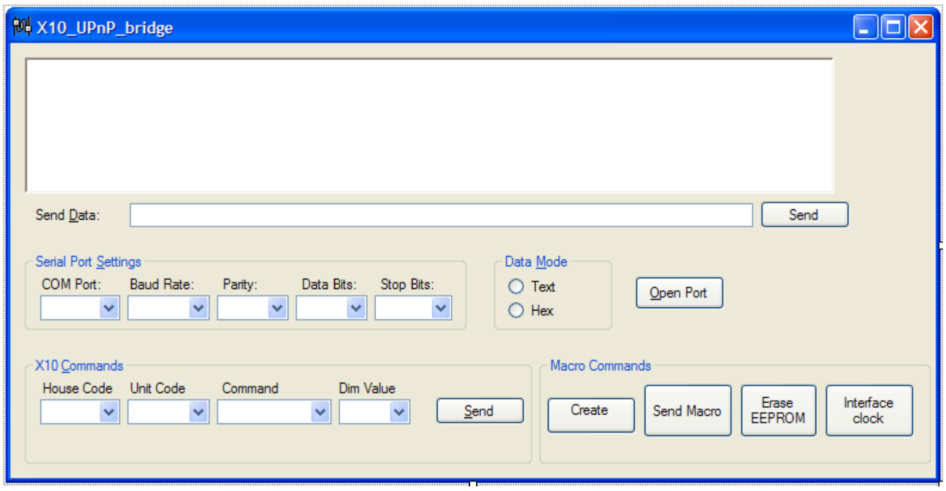

Figure 3. X10 UPnP bridge application

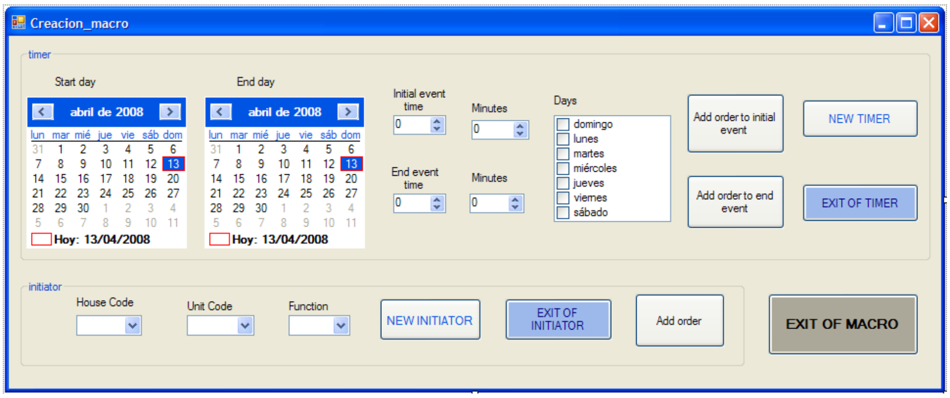

Figure 4. X10 UPnP bridge macro application

And in the other option, you can select a single X-10 command that can be used to trigger a sequence of X-10 commands to create different moods, as for example to create an environment to listen to music at home. In short, Macros simplify your schedule, save time, and give your home a "lived-in" look while you're away.

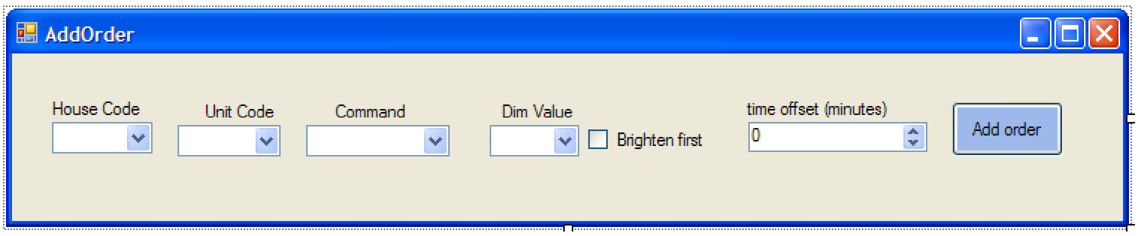

Figure 5. X10 UPnP bridge application

Let's see the appearance of the system from the point of view of the UPnP network. From the moment the application starts, an UPnP device is also generated. This device represents the X10 system. Once the UPnP device has been generated, it is announced within the local UPnP environment, so any control point can determine its presence and learn about its capabilities. Then, it is possible for a control point to invoke the permitted actions over that device. As a result, the same actions will be invoked over the correspondingly

X10 device via the transmitter, getting the physical device switched on or off. The functions the UPnP users can invoke are two:

-Sending simple commands to the wire, for example A3 DIM 20.

-Sending the programmed macros, in a hexadecimal format.

When the state of an X10 device is modified –either by an X10 user or a remote UPnP one-, report the changes to the both networks. In case the UPnP control point would be subscribe to events, it would be informed with the produced changes in the state variables of any X10 device, as house code, unit code, function and dimming value.

Another very useful application the bridge offers for informational purposes is the creation of XML files with user information about all the performed configuration (parameters of the serial port configuration and programmed macros).

3. CONCLUSION

This paper presented a control application that allows interoperability between two different technologies, X10 and UPnP. Several issues need further improvements, as the presented work is still under development, as for example, the creation of an UPnP presentation page for the bridge. The integration of PLC technology with UPnP environment allows for end-to-end remote monitoring, control and surveillance reaching places where no cabling exists other than the power wires. Besides, services bridges based solutions like the one presented in this document favor the integration of different technologies under a common interface based on IP.

REFERENCES

Antonio F. Gómez Skarmeta, 2004. *PLC-based Home Automation System Completed*

http://www.6power.org/open/6power_pu_d4_10_v1_4.pdf

CEA - Consumer Electronics Association

http://www.ce.org/

Digital Living Network Alliance.

www.dlna.org/

Henry Wiechman, December 2005. *Interoperability in the Multi-Format Home Network* http://www.hometoys.com/htinews/dec05/articles/ti/networkhome.htm

Richard A Quinnell,2007. Networking moves to home automation

http://www.edn.com/article/CA6455597.html

Simple Control Protocol

http://www.microsoft.com/japan/windows/scp/default.mspx

UPnP™ Forum.

http://www.upnp.org/

Warren Webb, 2006. *Smart-building systems converge*

http://www.edn.com/article/CA6360315.html

William Foard, October 2006. *Bridging to Home Automation's Future*

http://www.hometoys.com/article.php4?displayid=813

Wi-Fi Alliance

www.wi-fi.org

X10 protocol.

http://www.heyu.org/docs/protocol.txt

X-10 Technology Transmission Theory.

http://paginas.fe.up.pt/~ee99043/plm/x10tech.htm