

# **Adding functionality to X10 networks with 802.15.4. Using 802.15.4 to communicate and add functionality to X10 wired networks.**

William Wallace, Jose I. Moreno, Ruben Hidalgo

Universidad Carlos III de Madrid, Dpto. Ing. Telemática  
Avda. Universidad 30,  
28911 Leganés, Madrid, Spain  
{williamdaniel.wallace, joseignacio.moreno, ruben.hidalgo}@uc3m.es

**Abstract.** Although several newer technologies exist in the domotics market, one of the oldest remains extremely popular. Despite of its very limited functionality and performance, X10 is still a very common technology to manage simple Home Networks. Even if other wired solutions such as KNX/EIB or LonWorks out perform X10 in every single aspect, they might still be a bit of an overkill when addressing simple needs in a modest Home Network. However, X10 lacks support for wireless networks that could add a lot of functionality to its wired network. A good example would be how sensor networks could communicate with already existing X10 modules to allow full home automation in, for example, light and shutter control.

This paper will focus on the use of 802.15.4 to add functionality to simple X10 networks, in particular how 802.15.4 could allow the use of remote controls (or switches) acting over regular X10 modules, and how X10 switches can act over wireless modules using 802.15.4 as the transport layer. We will also see how 802.15.4 doesn't add complexity in the deployment and management of X10 Home Networks, which is one of the reasons such an old technology remains popular.

**Keywords:** 802.15.4, X10, Java, ZigBee, Domotics.

## **1 Introduction**

X10 [1] is one of the oldest, yet still fairly popular, domotics technologies. Developed in 1975, it remains to date one of the references in home networking by using the Power Line as the Physical mean to communicate different modules. X10 has a very modest performance, close to just one command per second, and isn't reliable, this means there's no way to know if a command has reached destination or not. But still, X10 is a pretty cost effective solution and its functionality is enough for the modest needs of a small home: control over lights, shutters, alarms and even climate conditions.

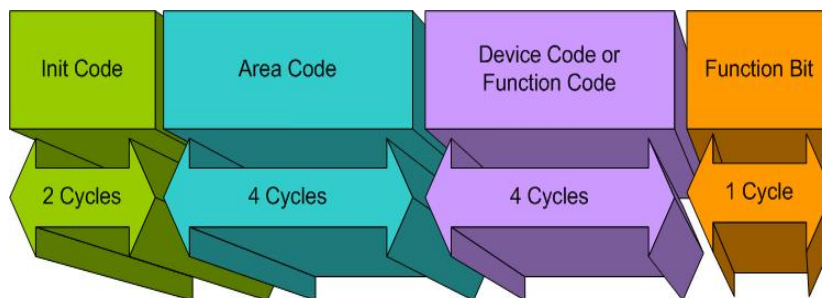
However, X10 lacks a good wireless companion. There's no real standard for wireless communication using X10, and most solutions don't really offer a true

wireless network solution, but more the ability to act over wireless modules separately. Companies such as Insteon [2] do offer wireless compatibility, but still lack the strength of a real standard behind them. IEEE 802.15.4 [3] provides a cost effective, standard solution for this. 802.15.4 is a low cost, with extremely low power consumption and low data rate technology released in 2004, and has the strength of a standard defined by the IEEE organization. It's designed to work very well in sensor networks, which is an asset X10 lacks, because even if it is compatible with sensor networks, its limitations don't really allow it to manage complex sensor networks that could really improve the domotics service it provides [4]. Other technologies offer much better performance, or even lower cost, if deployed from scratch, but still many X10 networks already exist that could be greatly improved by using a simple and cost effective wireless solution.

## 2 X10 Overview

X10 is one of the oldest domotics technologies. Through the Power Line network, it allows the communication of up to 256 modules, divided in 16 Areas with 16 devices per Area. Therefore, a module's address is composed of an area code (ranging from A to P) and a device code (ranging from 1 to 16).

X10 isn't reliable, this means there's no way to know if a command has been correctly received or not. In order to avoid undesired effects (for example, a shutter going down when it shouldn't), each command is sent twice, and in case the copies don't match nothing is done. A command is divided in two parts before being sent. The first part addresses one to several modules and tells them to listen for the command to be sent afterwards. Then, the function these devices have to execute is sent. Since 3 cycles have to pass before the second part is sent, each is sent twice to avoid errors and 3 cycles have to go by before another command is sent, each command takes about 50 cycles.



**Fig. 1.** X10 telegram structure

X10 has a very low performance by today's standards, since it sends "bits" of information on the zero crossing of the power line's AC signal. Since the AC signal's frequency in Europe is 50 Hz, and each simple command needs about 50 cycles, the performance is at best one command per second.

On the other hand, X10 is very inexpensive and simple, which is the reason many users still choose it as a solution for their home automation needs. In fact, X10 only supports up to 16 commands, of which 3 of them are extended commands (one for security, another one for sensor networks and advanced functionality and the last one is reserved for the manufacturer). However, only 13 commands are normally used:

- ON: Turns a device On.
- OFF: Turns a device Off.
- DIM: Dims a light or lowers the shutter by 5%.
- BRIGHT: Brightens a light or raises a shutter by 5%.
- ALL LIGHTS OFF: Turns off all light modules of an area code.
- ALL LIGHT ON: Turns on all light modules of an area code.
- ALL UNITS OFF: Turns off all units in an area code.
- STATUS ON.
- STATUS OFF.
- STATUS REQUEST.
- HAIL REQUEST.
- HAIL ACKNOWLEDGMENT.

### **3 IEEE 802.15.4 Overview**

Released in 2004, the IEEE 802.15.4 standard defines a low cost, low data rate (up to 256Kbps) and very low power consumption communications technology. It's aimed at small devices, powered by batteries which should last up to several months or even years. Because of this, data communication must be brought to a minimum if batteries must last, so even if the data rate may be enough for the transmission of multimedia content, it's strongly recommended not to use 802.15.4 networks for this purpose, since the relatively high data rate is useful to bring the time the 802.15.4 antenna is powered to a minimum, thus greatly reducing power consumption.

802.15.4 allows up to 16 channels on the 2.4 GHz band, which is useful to avoid interference with widespread WiFi and Bluetooth / WiBree networks [5], which also use the same band to communicate.

Although 802.15.4 is specially useful in sensor networks, its application in other scenarios is also interesting. Take for example a 802.15.4 light switch, light switches are not activated extremely often, maybe just a dozen times a day or so. By using 802.15.4, a light switch may be placed wherever we want, at a very small cost (around 3 euros per 802.15.4 module, plus the cost of the switch itself). Compare this to the cost of placing a regular, power line connected, light switch. It's fairly obvious that by the use of this standard, not only new functionality may be achieved in domotics networks, but could also lower the cost of common home networking needs.

802.15.4 allows for two types of devices, Full Function Devices (FFDs), used as network coordinators and Bridges to other networks, and Reduced Function Devices (RFDs), that can only communicate with their assigned FFD. In a domotics environment, it is clear a star topology might be the best idea.



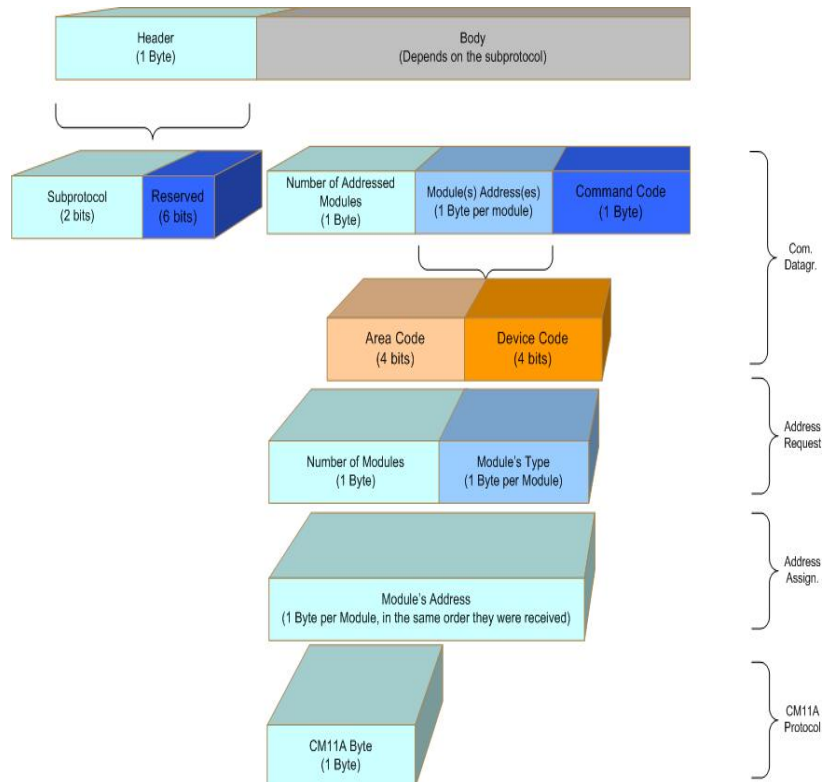
The X10 protocol will be implemented directly over 802.15.4's MAC layer. This layer provides device discovery functionality, so all the upper X10 layer has to provide is the necessary logic in order to identify different modules connected to the same device. Since the protocol architecture is very simple, so will be the necessary software to be developed on 802.15.4 devices, and the data-grams to be sent are kept as small as possible in order to profit from 802.15.4 low power consumption.

Several protocols are set in place:

- Address assignment and device pairing protocol: it's used to add devices to the wireless network. The 802.15.4 device sends a message to its FFD asking it to assign an address to each of the modules connected to it. Modules connected to the device fall in one of two categories. First of all, they can need an address to be assigned to them in order to be able to receive commands from other X10 modules. Other modules might need to be paired to another device, such as light switches for example, and therefore need the PAN coordinator to tell them the address of the module they must send their commands to. A module type field is therefore included in the datagram in order to identify the type of module connected to the device. If the PAN coordinator isn't able to satisfy this request, it will forward this message to the corresponding X10 network coordinator.

- A CM11A [6] direct transport protocol. This is useful to establish a direct link between the PC and the Gateway (in our case, a CM11A module) in case they aren't directly connected through a RS-232 or USB link, but rather through the 802.15.4 network. In fact, in the chosen scenario, since the PC is connected through a RS-232 link to a 802.15.4 device, it will really "think" it is directly connected to the CM11A device.

- The X10 - 802.15.4 protocol. This protocol has the same functionality as the wired X10 protocol. For the sake of simplicity, and because the elements of the network we studied don't support extended commands, these are yet to be implemented.



**Fig. 4.** Complete X10 - 802.15.4 Protocol

In this scenario, only two types of module will be used, however, by using a whole byte to identify the module's type we can differentiate up to 256 different types of modules, which should be more than enough in simple Home Networks. However, it's obvious that more bytes could be eventually used in case this was needed.

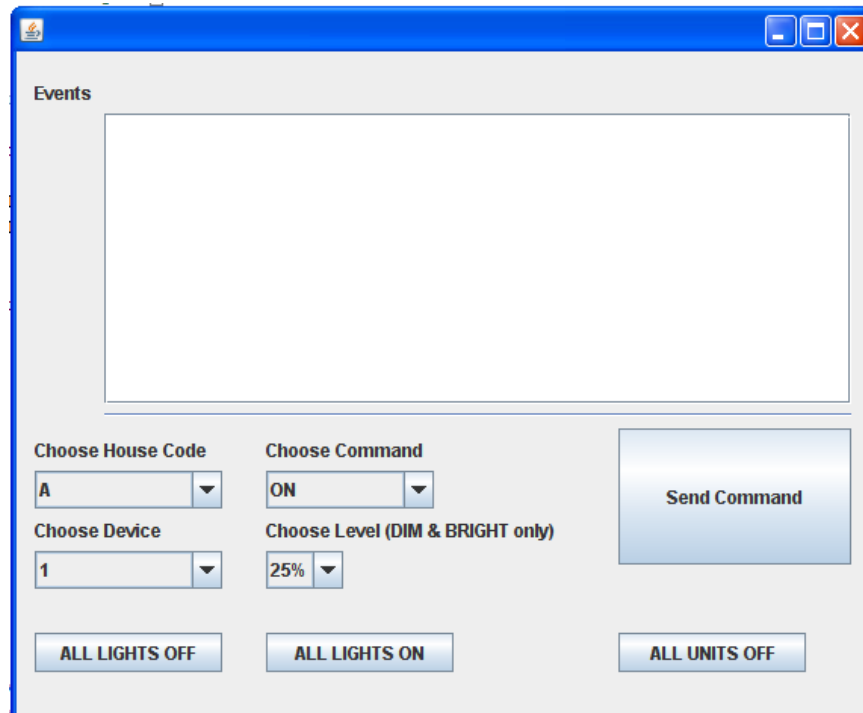
## 6 Management and Deployment of Hybrid Networks

As we've already said, this network structure is to be used only in scenarios with humble needs, in which control over shutters, lamps and simple appliances is enough. Deploying an X10 wired network is fairly easy, no software is needed to do this and plenty documentation is ready on the net that describes the process of running and installing regular X10 networks. In short, X10 modules usually come equipped with two dials. These dials are used to manually assign it an address (Area and Device Codes). This address is either the module's address, or its target's.

There are two options when adding 802.15.4 controlled modules to the domotic network. Either these modules can be "programmed" as regular X10 modules are, that is, their address (or destination address, in the case of switches for example) can be manually set, or they can use the protocol defined above. This protocol allows the

dynamic pairing of different modules, but comes at a cost, you need a complex device (such as a PC) to manage the network.

To solve this issue, a set of software tools has been developed. A JAVA API allows the development of applications that need to communicate with the X10 network (wired and wireless devices alike). This software allows the control and monitoring of wired as well as hybrid networks alike, without adding too much complexity. Through the serial port, an 802.15.4 gateway device or a CM11A module may be connected, giving access to either the Hybrid or the wired network. The API allows the application to monitor X10 Commands as well as sending them, in order to act over a certain device (for example, turning off all lights in the living room). It also allows to create sets of devices that can be addressed at the same time (for example, the lights in a room, or the shutters in an-other one).



**Fig. 5.** Software interface to monitor and act over the devices of the X10 network.

The software also allows to assign addresses to different wireless modules that ask for one. This assignment may be done manually (by asking for direct input by the user), or automatically by using a database (in case the device lost power and needs to be reprogrammed, for example). This is specially useful in scenarios where some devices may act over different modules depending on the circumstances, such as, for example, a switch that could control shutters at day time, and lights at night.

The CM11A protocol over 802.15.4 allows the API to ignore if it's really directly connected to a CM11A gateway device or not, so except for the casual in-put request to pair devices or assign an address to a module, the management and monitoring of a

Hybrid network should appear to the user as almost equivalent to a regular wired X10 network. The CM11A protocol is not described in this document, although more information about may be found on the References section. Suffice to say, this protocol allows a computer, or any other device for that matter, to send commands to the X10 network as well as monitoring the commands other modules send through it. However, in some scenarios it might not be always possible or desirable to physically connect the computer to the network, and using a wireless connection might be more useful, for example, if the user wishes to monitor and control the network from different or mobile devices such as PDAs, lap-tops, etcetera.

Finally, the JAVA API also allows remote devices to access the Hybrid network as if they were directly connected to it, through an Internet connection.

## 7 Application Scenario

The network is as shown in figure 2, where device B controls four switches and appliances (LEDs). In this section, we'll not cover how the wired X10 network is set in place, since it's very simple and widely explained in many documents over the Internet. We will, however, focus on how the Hybrid network is configured using an Application developed over the Java API described in the previous Section.

The following Hardware was used:



**AD10D IN Switch by Marmitek [7].** Controlled by simple on and off commands. Two of these modules are used to control two LEDs. The modules are connected to the regular power line installation and are controlled by two switches.

**LD11 DIM Dimmer by Marmitek.** Allows the control of lighting devices, by turning them on, off, dimming or brightening them. Two switches control these two modules, which are connected to two light bulbs.

**TMD4 Micro module by Marmitek.** Converts up to four switches into X10 modules capable of transmitting on, off, dim and bright commands. Four switches are used in this network to control two light bulbs and two simple appliances (that just turn on or off).

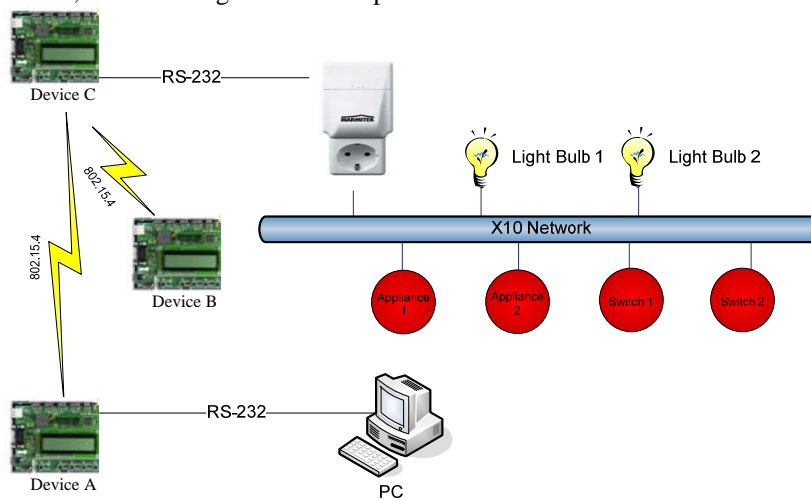




**CM11A interface by Marmitek.** Through a serial or USB port, this device allows the management and monitoring of X10 networks, as well as other advanced functions not covered in this paper.

**1321XNSK-BDM by Freescale [8].** Allows 802.15.4 communications and has 4 LEDs, as well as four switches and an LCD panel. Programming of these devices was done using Metrowerks' CodeWarrior.

Therefore, the following network setup is used:



**Fig. 6.** Application Scenario network structure.

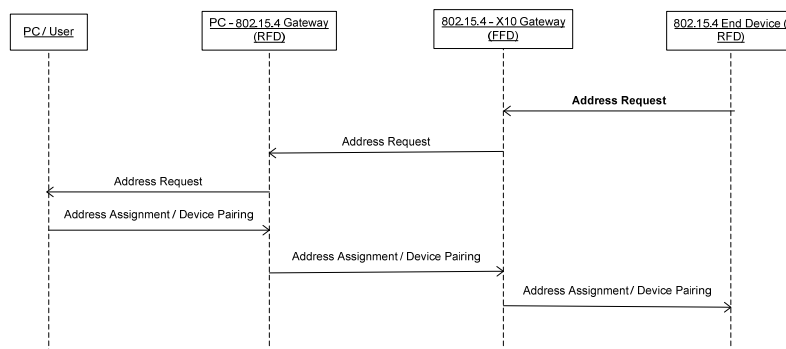
Device A allows the PC to communicate to the Hybrid network through a serial port. Device C allows the CM11A gateway module to communicate with 802.15.4 devices, and Device B will act as our only 802.15.4 device, which allows 4 switches and 4 leds to communicate with the rest of the network.

Once Device B is added to the Network, using the common 802.15.4 device discovery procedure, the following steps are taken:

1. Device B sends a Message requesting an address for each of the modules connected to it. The first four are announced as switches (so the received address will be the one of the module they should control) and the other four as simple appliances, which will only answer to ON, OFF and ALL UNITS OFF commands. Since the PAN coordinator isn't able to answer this request, it will forward the message to device A, which is connected to the network manager (the computer).
2. The PC - 802.15.4 gateway receives the message and asks the PC for in-pu through the JAVA application.

3. The user inputs the information or the application automatically answers the request.
4. The PC - 802.15.4 gateway answers the request.
5. Device C receives the message and forwards it to Device B.
6. Device B stores the received data, modules are effectively paired.

The following Sequence Diagram sums up all of this process.



## 8 Conclusions and Future Work

As we've seen, adding functionality to X10 wired networks through 802.15.4 is very simple, yet extremely cheap, from an economic point of view as well as from a complexity point of view. Almost no complexity is added to the management and deployment of regular X10 networks, still, the ability to place a switch wherever we want, as well as controlling simple appliances not connected to the main power line (such as a turning on a sprinkler for example) at a very low cost proves that not only 802.15.4 is capable of bringing X10 networks to a whole new level, but that 802.15.4 or ZigBee's [9] Home Networking Profile could provide a perfect replacement for wired technologies such as this one at a very low cost.

Still, many X10 networks exist already, and many owners don't feel the need to upgrade them by throwing away their whole domotics installation. In those cases, the direct use of 802.15.4 may be a perfect solution to upgrade those old networks and still get a glimpse of what 802.15.4 can offer.

Future work will focus on bringing full sensor network functionality to these Hybrid Networks, by adding support for extended commands. Also, since security is a key concern, specially in wireless networks where physical access is very hard to forbid, data encryption is a must [10]. The next logical step will be to bring the same encryption tools ZigBee uses to Hybrid networks. This will probably increase power consumption and cost, but will provide the necessary security a domotics environment must have.

Because of the extremely limited functionality of X10 networks, which allow a small amount of modules very limited in functionality, large or even medium size

projects (such as a big house, for example) can't profit from such a low cost solution. In that case, other technologies such as KNX/EIB [11] or ZigBee are more suited. Future work will also focus on hybrid KNX/EIB ZigBee networks, since KNX/EIB RF doesn't allow data encryption and ZigBee's Home Networking Profile would be a perfect substitute for it.

**Acknowledgments.** This work was partly funded by Spanish Minister of Education (MEC) under CASERTEL-NGN project (TSI2005-07306-C02-02) and Madrid National Research Program e-Magerit (S-0505/TIC/000251).

## References

1. X10 protocol description: <ftp://ftp.x10.com/pub/manuals/xtcode.pdf>.
2. Insteon's website: <http://www.insteon.net/>
3. 802.15.4 Standard Definition:  
<http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf>
4. Ed Callaway, "Home Networking with IEEE 802.15.4: A developing standard for low rate wireless personal area networks", Aug 2002, IEEE Communications magazine.
5. Khaled Shuaib, Maryam Alnuami, Mohamed Boulmalf, Imad Jawhar, Farag Sallabi and Abderrahmane Lakas, "Performance Evaluation of IEEE 802.15.4: Experimental and Simulation Results", Journal of Communications, Vol. 2, No. 4, June 2007.
6. CM11A protocol description:  
<http://mywebpages.comcast.net/ncherry/common/protocol.html>
7. Marmitek's website: <http://www.marmitek.com/>
8. Freescale's web site: <http://freescale.com/>
9. ZigBee Alliance's web site: <http://www.zigbee.org>
10. Naveen Sastry, David Wagner, "Security Considerations for IEEE 802.15.4 Networks", University of California, Berkeley.
11. Konnex Association web site: <http://www.knx.org/>
12. Workgroup's site: <http://panal.it.uc3m.es/zigbee/>