Embedding a bridge Internet to X-10 and a web server in microcontrollers

Martin Perez, Pablo Agüero, Alejandro Uriz, Juan Carlos Bonadero, Juan Carlos Tulli and Esteban Gonzalez
Facultad de Ingeniería
Universidad Nacional de Mar del Plata
Mar del Plata, Argentina
Email: perezmartin@fi.mdp.edu.ar

***Abstract*—Domotics contributes to enhance life quality, making the house a more functional place where perform diverse activities. The proliferation of wired and wireless communications imposes a new direction for domotics: enabling remote home actions. X-10 is a communication protocol designed for sending signaling and control over 220 VAC wiring for plug-in modules available from various vendors, by using the AC wiring already installed within a home. This paper shows the implementation using Microchip's microcontroller of an interface between X-10 communication protocol and Ethernet protocol to let a person command his home using any device with TPC/IP protocol. In this way it will be possible to perform home actions even using a cell phone or a WIFI connection.**

***Index Terms*—Domotics, X-10, Ethernet, web management.**

I. Introduction

PEOPLE have always attemped to create a comfortable environment for themselves. As technology evolved this has become particularly easier to accomplish. The idea of "Intelligent Homes" is precisely one of such goals: a home that requires minimum interaction from people in order to perform some basic actions, and also, the possibility that these actions could be executed automatically or even remotely if needed [1].

Domotics contributes to enhance life quality, making the house a more functional place to carry out domestic, professional and fun activities. The control network of the domotic system may be integrated with the power line network, and may be coordinated with other networks, such as telephony, television, and new information technologies [2], such as internet and wireless devices.

This home automation concept, also known as smart homes or domotics, have had a great proliferation thanks to inexpensive equipments and standards. X-10 is an example, an international and open industry standard for communication among electronic devices used for home automation, also known as domotics [3].

X-10 is a communication protocol designed for sending signaling and control over 220 VAC wiring, using 120 kHz bursts timed with the power line zero-crossings to represent digital information. Plug-in modules available from various vendors enable users to create home automation systems by using the AC wiring already installed within a home.

The proliferation of wired and wireless communications imposes a new direction for domotics. Because it may be of interest that home actions could be remotely executed if the need arises, X-10 users may want remotely switch on heating in order to have a warm home on arrival from a long trip.

Commercially it is possible to find some solutions for X-10 remote management. For example, WebIO [4] is an Internet to X-10 control network gateway, providing both local and remote Internet control of lamps, appliances and other electrical devices in the home or office. Utilizing X-10 technology, the WebIO sends signals using existing AC power lines to control X-10 modules. The WebIO contains a built-in web server allowing device control by remote users with Internet access via a Smartphone or a computer running a web browser.

Fig. 1. WebIO module and smartphone web interface

IoBridge [5] is another commercially available option. With IO-204 is possible to control on a web page or mobile phone X-10 modules attached to appliances, outlets, and lights. In order to control X-10 modules over the Internet, are required X-10 modules, PSC04 controller, ioBridge X-10 Smart Board and ioBridge IO-204 Ethernet module.

Other examples of X-10 remote management solutions may be found on internet, such as myHouse Online (ActiveHome) [6] and Taiyito products [7].

This paper shows the implementation of an interface between X-10 communication protocol and Ethernet protocol to let a person control his home using any device with TPC/IP protocol. In this way it will be possible to perform home actions using a cell phone or a WIFI connection.

The paper is organized as follows. Section II briefly describes the X-10 protocol to provide the necessary background to understand Section III. The later explains the proposed approach for the management of X-10 using a web server embedded in a microcontroller. Finally, Section IV shows the conclusions and some future directions.

II. X-10 PROTOCOL

In 1970, a group of engineers started a company in Glenrothes, Scotland, called Pico Electronics. By 1975, the X-10 project was conceived. It was named so because it was the tenth project. In 1978, X-10 products started to appear in stores.

The protocol enables the data transmission through low tension power lines at very low speed (60bps in USA and 50bps in Europe), with very low costs. The use of already installed home power lines is an important advantage, because it is not necessary to install a new wired network to connect different devices. It is an important saving of installation time and hardware, and consequently, a reduction in costs.

Thanks to the maturity of this technological solution (more than 30 years in the market) and the applied technology in X-10 products, their prices are very competitive. X-10 is a leader in the market of homes and little enterprises. The installation may be performed by end users or electricians without any automatization knowledge.

In the X-10 protocol household electrical wiring (the same which powers lights and appliances) is used to send digital data between X-10 devices. Figure 2 shows the many devices that may be involved in a X-10 network, from lamps and wall switches, up to motion detectors or computers.

The before mentioned digital data is encoded onto a 120 kHz carrier which is transmitted as bursts during the relatively quiet zero crossings of the 50 or 60 Hz AC alternating current waveform. One bit is transmitted at each zero crossing.

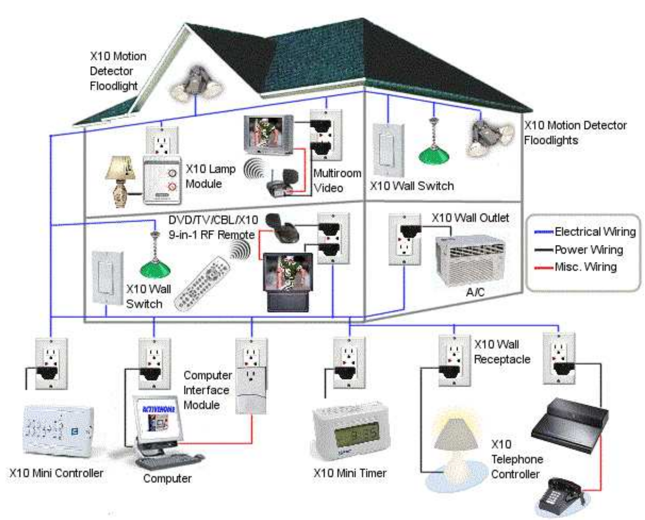

Fig. 2. X-10 House

X-10 transmitters may send signals according to the specifications of the X-10 protocol to up to 256 devices over the power line. Multiple transmitters may send signals to the same device. An example of a transmitter device is shown in Figure 3.

Fig. 3. X-10 transmitter device

X-10 receivers perform actions according to the signals sent by transmitter devices. Depending of the received signal, the X-10 devices may perform many actions, such as switching on, switching off, and many others.

These devices have two small dials, one with 16 letters and the other with 16 numbers, to assign one of the 256 posible addresses. In the same house, and even in the same phase, may receivers may be configured with the same address. In this way, all of them will perform the same predefined action when the transmitter sends a signal to that common address. An example of a transmitter device is shown in Figure 8.

Fig. 4. X-10 receiver device

A. X-10 protocol

Frames transmitted using the X-10 control protocol consist of a four bit house code followed by one or more four bit unit codes, finally followed by a four bit command. For the convenience of users that configure a system, the four bit house code is selected as a letter from A through P, while the four bit unit code is a number 1 through 16.

When the system is installed, each controlled device is configured to respond to one of the 256 possible addresses (16 house codes × 16 unit codes). Each device reacts to commands specifically addressed to it, or possibly to several broadcast commands. Available commands are shown in Figure 5.

	D1	D2	D4	D8	D16
1	0	1	1	0	0
2	1	1	1	0	0
3	0	0	1	0	0
4	1	0	1	0	0
5	0	0	0	1	0
6	1	0	0	1	0
7	0	1	0	1	0
8	1	1	0	1	0
9	0	1	1	1	0
10	1	1	1	1	0
11	0	0	1	1	0
12	1	0	1	0	0
13	0	0	0	0	0
14	1	0	0	0	0
15	0	1	0	0	0
16	1	1	0	0	0
All Units Off	0	0	0	1	1
All Lights On	0	0	0	0	1
On	0	0	1	1	1
Off	0	0	1	0	1
Dim	0	1	0	0	1
Bright	0	1	0	1	1
All Lights Off	0	1	1	0	1
Extended Code	0	1	1	1	1
Hail Request	0	0	0	0	1
Hail Acknowledge	1	0	0	1	1
Pre-Set Dim	1	0	1	1	1
Extended Data(analog)	1	1	0	1	1
Status = on	1	1	0	1	1
Status = off	1	1	1	0	1
Status Request	1	1	1	1	1

Fig. 5. X-10 codes

Because there are no restrictions (except possibly consideration of the neighbors) that prevents using more than one house code within a single house. The "all lights on" and "all units off" commands will only affect a single house code, so an installation using multiple house codes effectively has the devices divided into separate zones.

Inexpensive X-10 devices only receive commands and do not acknowledge their status to the rest of the network. Two-way controller devices allow for a more robust network but cost two to four times more and require two-way X10 devices.

In the 50/60 Hz AC power line, a bit value of one is represented by a 1 millisecond burst of 120 kHz at the zero crossing point (nominally 0, but within 200 microseconds of the zero crossing point), immediately followed by the absence of a pulse. A zero value is represented by the absence of 120 kHz at the zero crossing point (pulse), immediately followed by the presence of a pulse. All messages are sent twice to reduce false signaling. After allowing for retransmission, line control, etc., data rates are around 20 bps, making X-10 data transmission so slow that the technology is confined to turning devices on and off or other very simple operations.

In order to provide a predictable start point, every transmitted data frame (see Figure 6 [8]) always begins with a start code of 1110. Immediately after the start code, a house code (A-P) appears, and after that comes a function code. Function codes may specify a unit number code (1-16) or a command code, the selection between the two modes is determined by the last bit where 0 indicates a unit number and 1 a command. One start code, one letter code, and one function code is known as an X-10 frame and represents the minimum components of a valid X10 data packet.

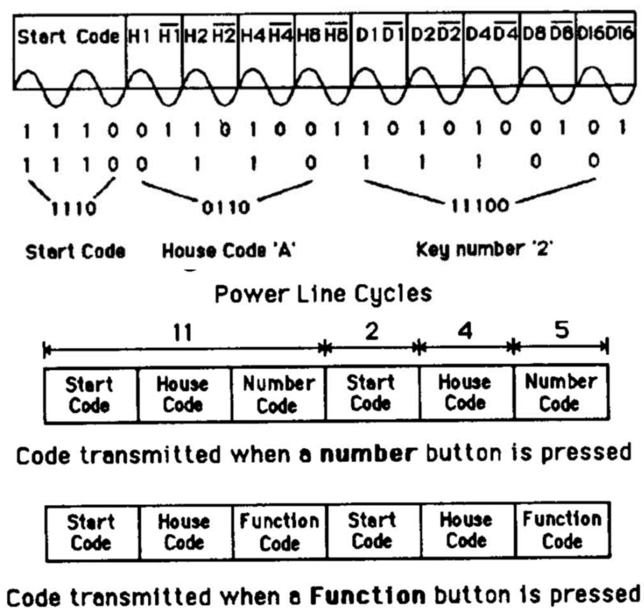

Fig. 6. X-10 frame

Each frame is sent twice in succession to make sure the receivers understand it over any power line noise for purposes of redundancy, reliability, and to accommodate line repeaters.

III. Managing X-10 using a embedded web server

As already mentioned in the introduction, the proliferation of wired and wireless communications imposes new possibilities for domotics. It is of interest that home actions could be executed remotely when the need arises. In this section it is explained the proposed X-10 interface with a web server embedded in a microcontroller. Figure 7 shows the prototype of the implemented system.

The embedded web server is running into a microcontroller device, and web pages are sent to the user to show all the available X-10 managing options. This web server may be connected to an ADSL router, and if a public IP address is

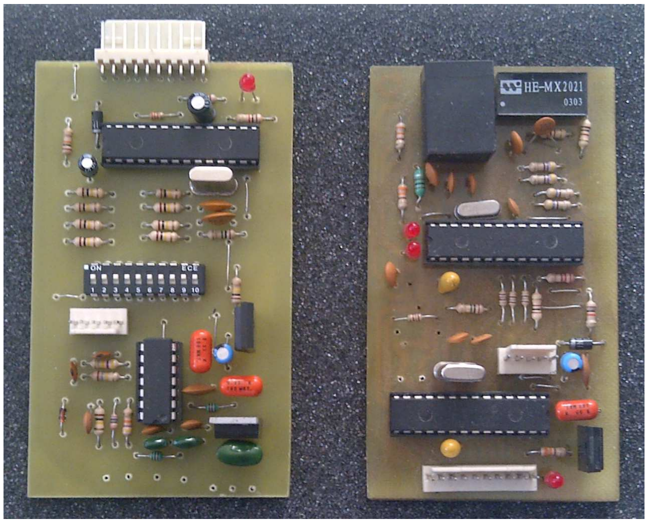

Fig. 7. X-10 transceiver (left) and embedded web server PCB (right)

available and no firewall restrictions are present, the access from any part of the world is guaranteed.

The microcontroller Ethernet connector may be also plugged to a wireless router, and the accessability via 802.11 protocol will be also allowed. In that situation, any portable device, such as a smartphone, could be used to manage the house.

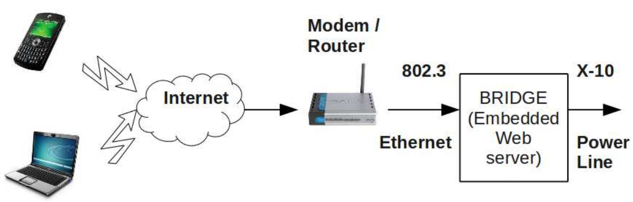

Fig. 8. Managing X-10 using a embedded web server

A. Proposed system

The proposed X-10 managing system using TCP/IP is shown in Figure 9. The system is attached to Ethernet using the RJ45 connector established by the standard. Microchip's ENC28J60 provides the Ethernet communication for the web server embedded in a microcontroller through the RJ45 connector. ENC28J60 is a stand-alone controller that is in charge of all Ethernet protocol tasks.

The web server in the microcontroller is accessed by the user using TCP/IP protocol over Ethernet. Any command or status request performed by the user is sent by the microcontroller to the X-10 transceiver using an UART. The transceiver is the module that performs the bridge between web page commands and X-10 network.

B. Microcontroller selection

An evaluation of the commercially available microcontrollers was made, and the choice was the family of Microchip's microcontrollers. The great advantage of these devices is the amount the available bibliography and information over internet about their products and design tools. These facts ease the tasks of the developer to get maximum benefits of the capabilities of the microcontroller in a relatively short time.

Fig. 9. Scheme of the proposed system

The chosen microcontroller has 16 bits and may be programmed with C30 compiler [9]. This tool offers many advantages, such as better optimization options, over the C18 compiler of 8 bits microcontroller families.

Microchip offers two families of PIC24 16 bit microcontrollers, PIC24F (low cost and medium performance) and PIC24H (high performance). These families have many common features, such as pinout compatibility, peripheral compatibility, same development tools, among others.

The proposed embedded web server is implemented using PIC24HJ128GP502 [10], due to its high speed (80Mhz,40Mips), easy to use 28-pin SDIP packaging, the high number of timers, the 128Kb of available memory, and the remappeable pins.

The main features are:

- 40 MIPS (Mega Instructions per second).
- 8MHz Internal oscillator.
- Internal PLL.
- Program memory: 128Kb.
- RAM memory: 8Kb.
- 2 UART, 2SPI and 1 I2C.

The speed and the amount of memory are well suited to perform the task of embedded web server.

C. Embedded web server

The Microchip TCP/IP Stack [11] shown in Figure 12 is a suite of programs that provides services to standard TCP/IP-based applications (HTTP Server, Mail Client, etc.), or can be used in a custom TCP/IP-based application.

The Microchip TCP/IP Stack is implemented in a modular fashion, with all of its services creating highly abstracted layers. Potential users do not need to know all the intricacies of the TCP/IP specifications to use it. In fact, those who are only interested in the accompanying HTTP Server application do not need any specific knowledge of TCP/IP.

The stack is written in the C programming language. Effective implementations can be accomplished in roughly 28-34 KB of code, depending on modules used, leaving plenty of code space on Microchip's cost effective, high-density microcontrollers for the user application.

The interface with the user of the proposed system consists of an embedded web page, stored in PIC memory using the MPFS format, using the HTTP server implemented by Microchip. The web page may be accessed using a cell phone, a WIFI connection or a computer to perform remote X-10

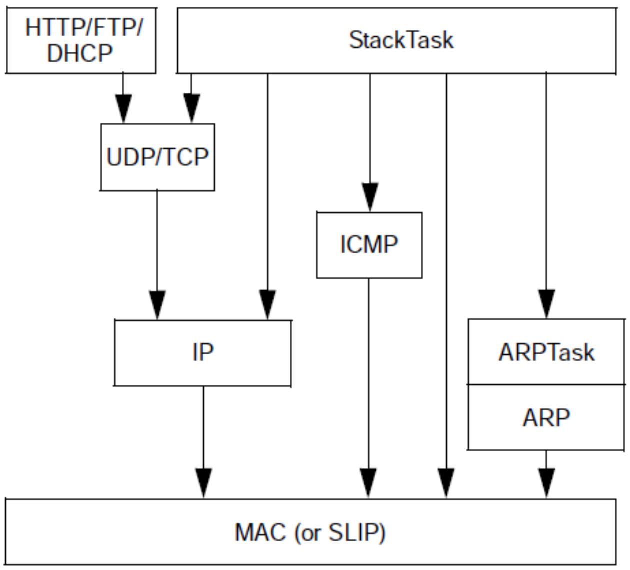

Fig. 10. Microchip's TCP/IP stack

actions. The web page has implemented security in order to prevent unwanted access from unauthorized people.

Fig. 11. Web page of the embedded web server

As shown in Figure 9, X-10 transceiver and HTTP server are implemented in separate microcontrollers due to several reasons, such as modularity to enable separate development of the two stages, and to preserve the timings of X-10 and TCP/IP tasks. TCP/IP task may be resource demanding, both memory and processing, and it is preferable to separate both tasks to avoid interaction problems.

The code of the sequence of tasks included in the HTTP server implementation has been modified. Source code was included for the communication with the X-10 transceiver via UART protocol whenever an action is performed in the web page.

The function HTTPExecuteGet was modified to include the analysis of the HTTP GET request. Each request produces a communication via UART with the X-10 transceiver, by means of the PrepareX10Request function. PrepareX10Request sets up the data to be sent via UART. Data is asynchronously sent in the end of the main infinite loop of Microchip TCP/IP stack routines, that begins with the subroutine StackTask and other operations.

```
  HTTP_IO_RESULT HTTPExecuteGet(void)
  {
   ptr = HTTPGetROMArg(curHTTP.data, (ROM
BYTE *)"luz1");
   if(strcmppgm2ram((char*)ptr, (ROM
char*)"on") == 0)
   {
   PrepareX10Request(0,HouseA,Unit1,On);
   }
  return HTTP_IO_DONE;
  }
```

In this way, the HTTP request is put in a queue, waiting to be sent by the UART transmission subroutine.

D. Ethernet interface

The interface of the PIC24H microcontroller with Ethernet is implemented using Microchip's ENC28J60. It is a 28-pin, 10BASE-T stand alone Ethernet Controller with on board MAC & PHY, 8 Kbytes of Buffer RAM and an SPI serial interface.

The ENC28J60 is a stand-alone Ethernet controller with an industry standard Serial Peripheral Interface (SPI). It is designed to serve as an Ethernet network interface for any controller equipped with SPI.

The ENC28J60 meets all of the IEEE 802.3 specifications. It incorporates a number of packet filtering schemes to limit incoming packets. It also provides an internal DMA (Direct Memory Access) module for fast data throughput and hardware assisted IP checksum calculations. Communication with the host controller is implemented via two interrupt pins and the SPI, with data rates of up to 10 Mb/s. Two dedicated pins are used for LED link and network activity indication.

This Ethernet controller communicates with the PIC24H through the SPI serial interface. In this way, many complicated task of IEEE 802.3 specifications are avoided by the microcontroller.

E. X-10 transceiver

The X-10 transceiver is also implemented in a PIC24H microcontroller. This transceiver makes the conversion of the orders sent by the embedded web server into X-10 messages.

The communication between the X-10 transceiver and the embedded web server is performed using a UART. This communication was chosen due to the low requirements in complexity and speed of the application.

Messages can be sent in both directions. The web server may send commands to perform actions. But, messages may also be sent by the X-10 transceiver to the web server due to device's status reports.

This microcontroller is also in charge of performing all X-10 communications with the electrical power lines. For example, it will receive the Hail Acknowledge messages sent by receptors, which indicate that X-10 transceiver messages were correctly received.

In order to perform transmitter and receiver tasks the transceiver must have a set of circuits to generate an adequate X-10 signal according to the standard specifications.

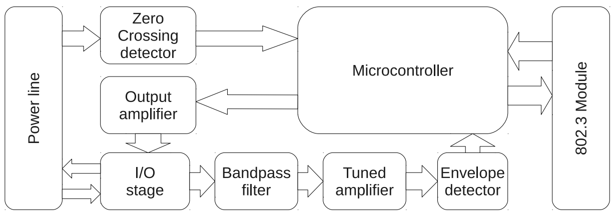

Fig. 12. X-10 transceiver

The Zero Crossing Detector (ZCD) is the synchronization circuit with the 220V signal. This circuit produces a squared signal with slopes that match zero crossings of the power line signal. This output squared signal is used to generate an interruption in the PIC24H microcontroller. The subroutine attached to the interruption transmits a bit of the queued output message if available. If nothing is prepared to be transmitted, that subroutine senses the 220V line to look for any data that must be received.

Message reception is performed using a state machine that progressively decodifies the received message. When the message is correctly received without any errors, it is sent to another subroutine to proceed with the corresponding action.

The incoming signal is sampled 500μs after the zero crossing and the incoming bit is stored. If the incoming bits match the start code, the state machine changes from state 0 (start code state) to state 1 (house code state). When the house code or the unit code (state 2) do not match the address of the transceiver module, it automatically proceeds to state 0 to look for any new start code sequence. State 3 corresponds to Command code.

Transmission and reception of X-10 signals are performed by Input/Output stage and other necessary circuits, such as Output Amplifier, Input Band Pass Filter, Input Tuned Amplifier, and Input Envelope Detector.

The input/output stage sends and receives the X-10 120 kHz carrier in the power line. It is made of a transformer with ferrite core and a high pass filter (30KHz cutoff frequency). The transformer isolates the module from power line, and it is also used to impedance matching.

Because of the noise present in the power line, it is necessary to further filter the received signal to get appropriate values of signal-to-noise ratio (SNR). For that purpose a Chebyshev passband filter with a central frequency of 120KHz was designed. The bandpass filter must have an adequate bandwidth to avoid slope deformations in the 120KHz bursts. The chosen bandwidth was 90KHz.

Tuned amplifiers are use to further amplify the input signal to get adequate levels for the envelope detector provides the digital signal input for the reception mode of the PIC24H transceiver microcontroller.

IV. CONCLUSIONS

This paper describes the proposed bridge between internet and X-10 protocol. The advances of wired and wireless communications imposes this new direction for domotics: wired and wireless X-10 management through web interfaces.

The proposed system is implemented with two PIC24H Microchip's microcontrollers. The speed, the amount of memory, SPI and UART interfaces of these devices are well suited to perform both tasks.

The web server included in one PIC24H microcontroller using the TPC/IP stack implementation of Microchip provides a user friendly interface to manage home both locally and remotelly, with wired and wireless devices. Another PIC24H performs the bridge between HTTP requests sent via UART, and X-10 network. X-10 replies are sent back to the web server also through UART.

The low cost and the simplicity of the proposed X-10 web management system makes it a suitable implementation for the domestic market of domotic solutions. Home automatization is an important solution to be applied to disabled people, to enhance their life quality.

REFERENCES

[1] J. Falquez, "X-10 protocol & power line communication," The George Washington University, Tech. Rep., 2009.
[2] M. Perez, "Domotics to enhance life quality of disabled people," University of Mar del Plata, Tech. Rep., 2012.
[3] http://www.x10.com/support/.
[4] http://www.webio.us/.
[5] http://iobridge.com/technology/x10-home-automation.html.
[6] http://www.x10.com/promotions/sw34a_cat_hm.html.
[7] http://factory.dhgate.com/remote-control/taiyito-tdxeseries.html.
[8] http://www.x10pro.com/pro/pdf/technote.pdf.
[9] http://ww1.microchip.com/downloads/en/DeviceDoc/MPLAB\%20C30\%20UG_DS-51284f.pdf.
[10] http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en534550.
[11] http://ww1.microchip.com/downloads/en/appnotes/00833b.pdf.