School of Engineering & Design

Electronic and Computer Engineering

Power Line Network Automation Over IP

A thesis submitted for the degree of Master of Philosophy

Dimosthenis Trichakis

Brunel University

January 2011

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published work of others has been acknowledged in the text and a list of references is given.

…..................... …..................

(Signature) (Date)

Abstract

A smart home contains many kinds of devices, sensors and user interfaces. In order for these to communicate with each other and share information and processing power they need to be connected together. Thus, seamless networking and device interconnectivity is a fundamental requirement for a smart home.

Unfortunately, this requirement for compliance and compatibility still presents a difficult obstacle as there are numerous competing standards and technologies available today and some devices are completely without communication interface.

Current state-of-the-art smart homes are implemented with home based on Web servers in order to surpass the difficulties. Also, a more sophisticated implementation is based on middleware. These two methodologies are expensive to implement.

Our proposed system is cost-effective and does not require Web servers or any special hardware and has low complexity.

The novelty and the achievement of the thesis is the integration of homogeneous networks through IP. In addition, the system was built-in terms of software and hardware based on a new protocol.

Acknowledgements

First I wish to thank my supervisor Dr Marios Hadjinicolaou and I am grateful to him for his expert help and support.

I wish to give many thanks to Dr. Manolis Antonidakis for his support and recommendations.

I wish to thank Mr. Iraklis Rigakis for his support.

I also thank Mr. P.Stivaris, my friend Mr. Christos Chousidis and colleague Mr. Vasilis Stauropoulos for their help and support.

I am also thankful to my family and relatives for their enduring support, especially to my love, Apostolia, for being that special one for me.

Finally, I would like to thanks all the staff of the Department of Electronic & Computer Engineering at Brunel University for their kind help.

To The Man,

before He becomes a Slave or a Master...

Table of Contents

Abstract..II

Acknowledgements...III

Acronyms...XII

Chapter 1 Introduction...1

1.1 Overview...1

1.2 Scope of the Thesis...2

1.3 State of the Art...3

1.4 Contributions...4

1.5 Organization of the Thesis...6

Chapter 2 Internet Protocol Suite...7

2.1 Introduction...7

2.2 ISO/OSI Network Model...7

2.2.1 Physical Layer...8

2.2.2 Data Link Layer...8

2.2.3 Network Layer...8

2.2.4 Transport Layer...9

2.2.5 Session Layer..9

2.2.6 Presentation Layer...9

2.2.7 Application Layer...9

2.3 TCP/IP Network Model...9

2.3.1 IP...10

2.3.2 TCP..13

2.3.3 UDP...16

Chapter 3 Description of Open Power Line HAPs..................................18

3.1 Introduction..18

3.2 ISO/OSI Network Model in Smart Homes...18

3.3 European Regulation Basics in PLCs...19

3.4 Description of X10 Power Line Protocol...20

3.4.1 Media Layer..21

3.4.2 Physical Layer...21

3.4.3 Data Link Layer...22

3.5 Description of INSTEON..24
3.5.1 Media Layer..25
3.5.2 Physical Layer..26
3.5.3 Data Link Layer..27
3.6 Universal Powerline Bus(UPB)..30
3.6.1 Media Layer..30
3.6.2 Physical Layer..30
3.6.3 Data Link Layer..31
3.7 EIB/KNX..33
3.7.1 Media Layer..34
3.7.2 Physical Layer..34
3.7.3 Data Link Layer Type PL110 & Network Layer..36
3.7.4 Transport Layer..37
3.7.5 Application Layer..38
3.8 LonTalk..38
3.8.1 Media Layer..39
3.8.2 Physical Layer..39
3.8.3 Data Link Layer..40
3.8.4 Network Layer..41
3.8.5 Transport Layer..41
3.8.6 Session Layer..42
3.8.7 Application and Presentation Layers..42
Chapter 4 IP Convergence of PLN's HAPs..43
4.1 Introduction..43
4.2 The Transport Mechanism..43
4.3 HAPoIP Layer..44
4.3.1 KNXnet/IP and LON/IP..45
4.3.2 X10/IP, INSTEON/IP and UPB/IP..45
Chapter 5 Hardware Design of the Test Platform..............................50
5.1 Introduction..50
5.2 Power Line Modem..50
5.2.1 Carrier Band Pass Filter..51
5.2.2 120 kHz Frequency Decoder..51

5.2.3 Mains Zero Crossing Detector......53

5.2.4 X10 Transmission Amplifier and Surge Protection......55

5.2 MCU......58

5.4 UDP/IP Module......61

Chapter 6 Software Design of the Test Platform......66

6.1 Introduction......66

6.2 Platform Initialisation......67

6.3 X10/IP Message Reception Process......67

6.4 X10/IP Message Transmission Process......69

6.5 X10 Message Reception Process......71

6.6 X10 Message Transmission Process......73

6.7 PLN Access Protocol......74

Chapter 7 Test and Validation......75

7.1 Introduction......75

7.2 Software Tools and Devices......76

7.2.1 CM15Pro™......76

7.2.2 LM12™ and AM12™......76

7.2.3 ActiveHome Pro™......76

7.2.4 Ostinato......77

7.2.5 Wireshark......78

7.3 Validation of the Platform......79

7.3.1 X10 Interface Tests......79

7.3.2 X10/IP Layer Operation Tests......81

7.4 Platform Integration to IP Network......82

Chapter 8 Conclusions and Further Work......84

8.1 Conclusions......84

8.2 Further Work......84

References......86

APPENDIX A......88

APPENDIX B......100

APPENDIX C......101

List of Figures

Figure 1-1: A schematic overview of the smart home technical system infrastructure......1

Figure 1-2: Power Line Automation over IP..2

Figure 1-3: Web Server Solution...3

Figure 1-4: Example of Middleware Solution..4

Figure 1-5: PLN and IP planes...5

Figure 1-6: Far-End Communication..5

Figure 2-1: TCP/IP model versus OSI model...10

Figure 2-2: IP Datagram Encapsulation. 10

Figure 2-3: TCP Header..15

Figure 3-1: X10 Transmission (bits: 110)...21

Figure 3-2: X10 Standard Frame...22

Figure 3-3: X10 Standard Message and Standard Frame Transmission...........................23

Figure 3-4: Extended Message Format...24

Figure 3-5: INSTEON Network Implementation Example..25

Figure 3-6: INSTEON frames over 60 Hz PLN. 26

Figure 3-7: INSTEON's Messages..27

Figure 3-8: INSTEON messages Fragmentation...29

Figure 3-9: Start and Body Frame Structures...29

Figure 3-10: Frames Group Transmission...29

Figure 3-11: UPB Pulse Position(60 Hz)..31

Figure 3-12: UPB Frame...31

Figure 3-13: UPB Frame Header...32

Figure 3-14: PL110 Frame...36

Figure 3-15: LPDU in Standard Frame..37

Figure 3-16: LPDU in Extended Frame...37

Figure 3-17: Acknowledgment Frame...37

Figure 3-18: TPDU..38

Figure 3-19: LonTalk Protocol Stack...39

Figure 3-20: Power Line Frame...40

Figure 3-21: Encoded LPDU...40

Figure 3-22: LPDU..41

Figure 3-23: NPDU..41

Figure 4-1: HAPoIP Adaptation Layer.....44
Figure 4-2: HAPoIP Packet.....46
Figure 4-3: HAPoIP Encapsulation Method.....46
Figure 4-4: Stop and Wait ARQ Protocol Timeline for Four Different Scenarios.....47
Figure 4-5: Timeline for Stop and Wait Protocol with 1-bit Sequence Number.....48
Figure 4-6: End-to-End Delay and ARQ Implementation.....49
Figure 4-7: Packet Structure.....49
Figure 5-1: X10 to IP Adapter.....50
Figure 5-2: Mains Coupler.....51
Figure 5-3: LM567 Circuit.....51
Figure 5-4: Bandwidth vs Input Signal Amplitude.....52
Figure 5-5: LM567 Output.....53
Figure 5-6: Mains Zero Crossing Detector.....54
Figure 5-7: Tx Amplifier and Surge Protection.....56
Figure 5-8: Transmission Amplifier Output.....57
Figure 5-9: MCU Schematic Diagram.....61
Figure 5-10: Embedded Ethernet Solutions.....62
Figure 5-11: W5100 Block Diagram.....63
Figure 5-12: WIZ812MJ Module Connectivity.....64
Figure 6-1: Basic Microcontroller Operations.....66
Figure 6-2: X10/IP Message Reception Procedure.....68
Figure 6-3: X10/IP Message Transmission Procedure.....69
Figure 6-4: End-to-End Delay.....71
Figure 6-5: X10 Message Reception Procedure.....72
Figure 6-6: X10 Message Transmission Procedure.....73
Figure 6-7: FSM Diagram of the PLN Access Protocol.....74
Figure 7-1: ActiveHome Pro™.....76
Figure 7-2: Ostinato.....77
Figure 7-3: Wireshark Analyser.....78
Figure 7-4: Connectivity Diagram for Timing Test.....80
Figure 7-5: Transmission Amplifier Output.....80
Figure 7-6: Connectivity Diagram for CSMA Algorithm Test.....81
Figure 7-7: Connectivity Diagram for X10/IP Layer Operation Test.....82
Figure 7-8: Connectivity Diagram for Platform Integration to LAN.....82

Figure 7-9: Propagation Delay of X10 Signals between R and S Phase............................83
Figure 7-10: Connectivity Diagram for Platform Integration to Internet...........................83

List of Tables

Table 2-1: ISO/OSI Model Layers..7

Table 2-2: IPv4 Packet Structure..11

Table 2-3: UDP Header...17

Table 3-1: The Modified OSI model according to CENELEC..19

Table 3-2: Summary of maximum transmission levels (EN 50065-1)..............................20

Table 3-3: X10 Codes...23

Table 3-4: Flags Field of INSTEON Message..27

Table 3-5: General Requirements for Physical Layer PL110..35

Table 5-1: Hardware vs Software TCP/IP Stack...62

Table 6-1: Worldwide Ping Measurements...70

Acronyms

ACK	ACKnowledgment
ARP	Address Resolution Protocol
ARQ	Automatic Repeat reQuest
ATM	Asynchronous Transfer Mode
BPSK	Binary Phase-Shift Keying
BW	BandWidth
CENELEC	Comité Européen de Normalisation Électrotechnique
CISPR	Comité International Spécial des Perturbations Radioélectriques
CPU	Central Processing Unit
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTC	Clear Timer on Compare match mode
DHCP	Dynamic Host Configuration Protocol
DiffServ	Differentiated Services
DoD	United States Department of Defence
EoF	End of Frame
EHSA	European Home Systems Association
EHS	European Home System protocol
EIB	European Installation Bus protocol
FDDI	Fiber Distributed Data Interface
FSK	Frequency Shift Keying
FSM	Finite-State Machine

HAP	Home Automation Protocol
HAoIP	Home Automation protocols over IP
HSSI	High-Speed Serial Interface
HTTP	HyperText Transfer Protocol
HVAC	Heating, Ventilating and Air Conditioning
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IP	Internet Protocol
IPDV	IP Packet Delay Variation
IETF	Internet Engineering Task Force
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISO/OSI	International Standard Organization's Open System Interconnect
LAN	Local Area Network
LLC	Logical Link Control
LPDU	Link layer Protocol Data Unit
MAC	Media Access Control
MCU	Microcontroller Unit
MII	Medium Interdependent Interface
MIPS	Million Instructions Per Second
MSK	Minimum Shift Keying
NACK	Negative ACKnowledgment
NetBIOS	Network Basic Input/Output System
NPDU	Network layer Protocol Data Unit
NRZ	Non-Return to Zero code
OOK	On/Off Keying modulation

OS	Operating System
PC	Personal Computer
PDU	Protocol Data Unit
PLC	Power Line Communication or Power Line Carrier
PLN	Power Line Network
POP3	Post Office Protocol version 3
PPM	Pulse Position Modulation
RFC	Requests For Comments
SDU	Service Data Unit
SFSK	Spread Frequency Shift Keying
SIP	Session Initiation Protocol
SMB	Server Message Block
SPI	Serial Peripheral Interface
TCP	Transmission Control Protocol
TPDU	Transport layer Protocol Data Unit
TTL	Transistor-Transistor Logic
TtL	Time to Live
UDP	User Datagram Protocol
UPB	Universal Powerline Bus
WiFi	Wireless Fidelity
X10/IP	X10 power line protocol over IP

Chapter 1 Introduction

1.1 Overview

Personal computers, wireless networks, home gateways, Internet connections and device convergence have all opened up new possibilities and we can now perform computing in various locations and forms.

Smart homes are homes with integrated technological systems and appliances able to offer remotely and/or centralised controlled functionality and services (Figure 1-1).

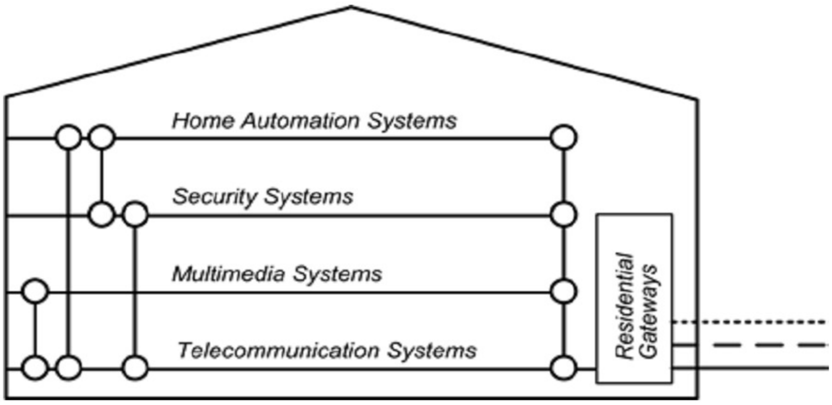


Figure 1-1: A schematic overview of the smart home technical system infrastructure [1].

The technological infrastructure is based upon:

- **Home automation systems**[1] including the control (on/off, open/close and regulation) of systems and functions for lighting, climate control (HVAC), blinds, doors, windows, locks, watering, white goods as well as the supply of water, gas and electricity.
- **Security systems** referring to alarms for intrusion (movement detectors, door and window detectors, glass break detectors), personal alarms (key fobs, wireless wristband transmitters and pendant transmitters), technical alarms (fire, smoke, gas, water leakage, and failure of electrical supply and telephone line) and remote video control for the protection of property, possessions and persons.

[1] Also called **domotics**.

- **Multimedia systems** including the capture, treatment and distribution of audio and video within, from and to the home, such as audio/video multi-room, home cinema and PC media servers.

- **Telecommunication systems** encompassing the distribution and sharing of files and data amongst machines and humans through cabled and wireless local area networks, IP telephony, telephony, broadband access, routers, etc.

- **Residential gateways or home gateways** having two main functions, one is to connect various home networks[2] together and/or to the Internet, while offering diagnostics, remote control facilities and user interfaces. The other is to be an enabling platform for new digital residential services [2, 3, 4 , 5].

All the above mentioned systems, except the home automation systems, have been already migrated to Internet protocol suite. Thus, the convergence of home automation protocols (HAPs) to Internet protocol (IP) will simplify not only the residential gateway tasks, but also the integration of different systems.

1.2 Scope of the Thesis

Traditionally, homes have been wired for electricity, telephony and television services only. Therefore, as the installation of a dedicated wired bus for automation usage is expensive, there are two cost effective possibilities for home automation media. One is the power line and the other is some type of radio signal (wireless).

The scope of this thesis is to propose a mechanism for IP convergence of the open automation protocols of power line network that have not already migrated to IP. As (Figure 1-2) shows a special node is introduced as gateway between the two networks.

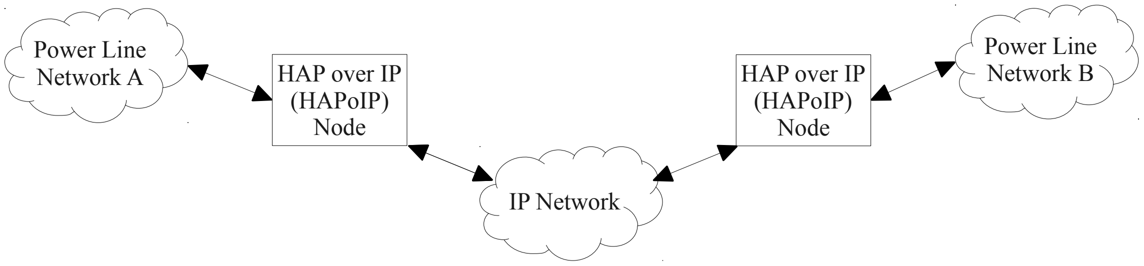


Figure 1-2: Power Line Automation over IP.

[2] Home networks can consist of many different kinds of networks and have incompatible physical interfaces, a gateway can bridge all these together.

1.3 State of the Art

There are different approaches for home automation systems convergence to IP network. In most cases the home gateway or a computer in the home's local area network (LAN), which is connected to the controller of the home automation system via a serial interface, accommodates a web server and the HTTP protocol is used for the communication with the other IP nodes and/or user interfaces (Figure 1-3) [2, 6].

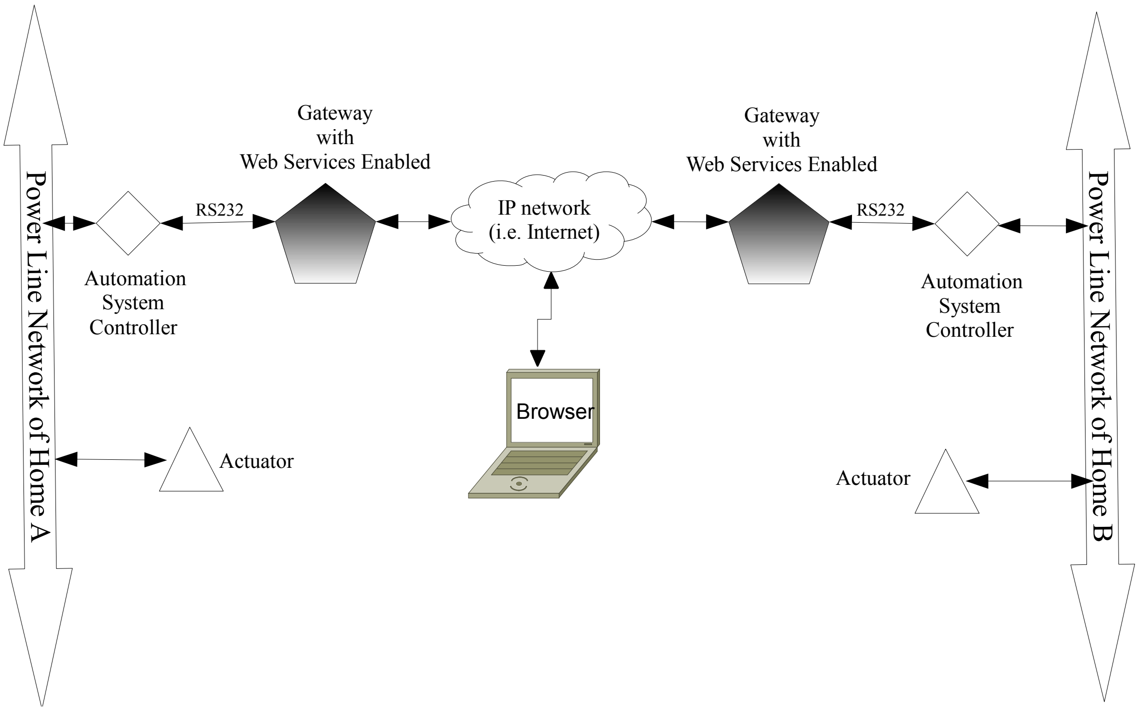

Power Line Network of Home A
Gateway with Web Services Enabled
RS232
Automation System Controller
Actuator
IP network (i.e. Internet)
Browser
Gateway with Web Services Enabled
RS232
Automation System Controller
Actuator
Power Line Network of Home B

Figure 1-3: Web Server Solution.

Another, more sophisticated, approach uses middleware that connects devices, network, software and services together in a smart environment. A home server with appropriate middleware could offer services such as timers, logging functions, preset modes with distributed user interfaces and group control (Figure 1-4) [3, 7, 8].

The disadvantages of the above mentioned approaches are the following:

- Not seamless solutions
- Complexity
- High cost

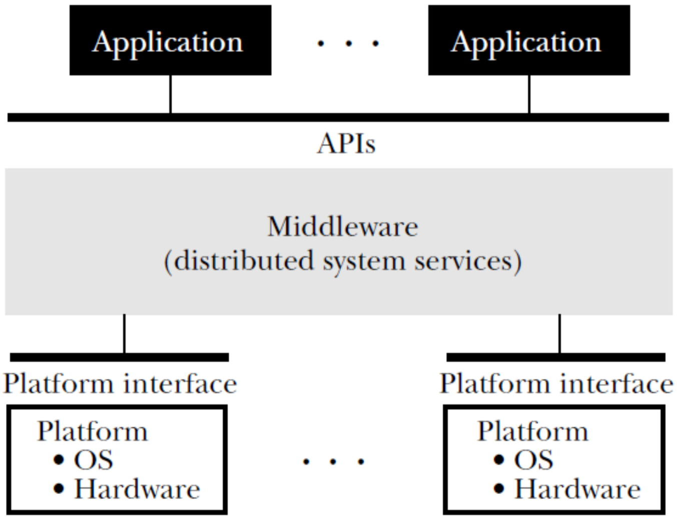


Figure 1-4: Example of Middleware Solution [9].

1.4 Contributions

This thesis will offer a wide range of contributions to the field of home automation systems as well as smart homes.

The main objective of this research was to design an application sub-layer in Internet protocol suite that gives the ability to open automation protocols of power line network to overcomes power line network barriers.

This achievement gives us the ability to connect transparently two or more separated and isolated power line networks via LAN and WAN with the help of HAPoIP protocol (Figure 1-5) and to have all the advantages of the services that have been developed in IP networks. Thus, widespread deployment of HAoIP solution presents an opportunity to expand building control communication beyond the local home power line bus providing:

- Remote configuration
- Remote operation
- Fast interface from LAN to power line protocol implementations and vice versa. The IP network acts as a fast backbone between HAPs nodes
- WAN connection between HAP networks

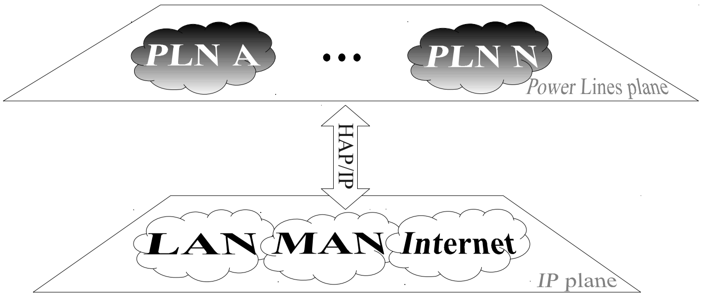


Figure 1-5: PLN and IP planes.

Therefore, we have the ability to control any distributed node (actuators, sensors and controllers) from any location around the world, and to have the control of the power line automation system in any place, in any time and in any way without the need of web servers and distributed controllers (Figure 1-3), (Figure 1-6).

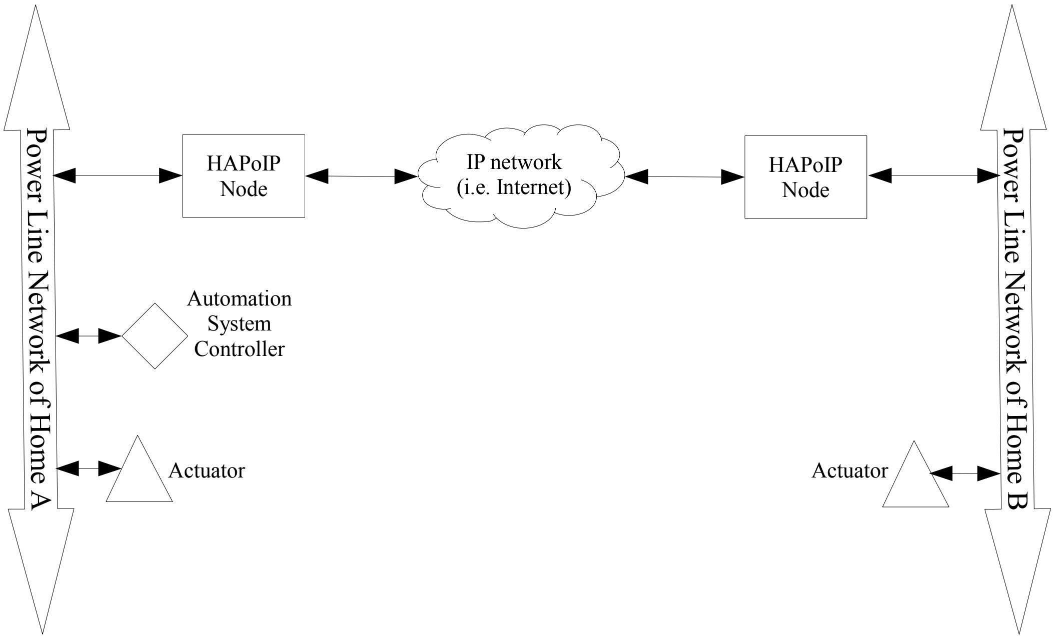

Power Line Network of Home A
HAPoIP Node
IP network (i.e. Internet)
HAPoIP Node
Power Line Network of Home B
Automation System Controller
Actuator
Actuator

Figure 1-6: Far-End Communication.

In addition, it will help in cost reduction of residential gateways as the Ethernet can be the only communication interface in use. Thus, as all power line communication (PLC)[3] data will be over IP, the gateway can serve only its basic role, which is the interconnection of two different networks, namely a network outside the home (WAN) with a network inside the home (LAN) [10]. All the other gateway services can be

[3] Also known as Mains Communication(MC) or Power Line Networking(PLN).

optional and in dependence to gateway platform hardware and software resources. Furthermore, the interaction and cooperation between different systems in smart home environment will be simple and robust.

Also, a low cost and easy to develop platform for HAPoIP node is presented, where common hardware parts and the powerful and portable C programming language gives us maximum flexibility in design and operation.

1.5 Organization of the Thesis

This thesis is organized into eight chapters.

Chapter two provides an overview on Internet protocol suite. It will be an overview of the OSI and TCP/IP architecture and protocols stack.

Chapter three presents the open power line automation protocols. This chapter is a brief description of X10, INSTEON, UPB, EIB/KNX and LonTalk protocols.

Chapter four analyses the IP convergence of the above mentioned protocols. This chapter describes how the HAPoIP layer is implemented in the protocols that have not converged to IP yet.

Chapter five presents the hardware implementation of X10/IP platform. As the UPB and INSTEON protocols have not developed for 50Hz power lines yet, our implementation took place in X10 protocol.

Chapter six analyses the developed software.

Chapter seven describes the tests that took place and experimental results are presented.

Finally, chapter eight draws the conclusions of the thesis and discusses possible directions for future research.

Chapter 2 Internet Protocol Suite

2.1 Introduction

The Internet protocol suite[4] is the set of communications protocols that implement the protocol stack on which the Internet runs.

A protocol stack is a complete set of protocol layers that work together to provide networking capabilities. It is called a stack because it is typically designed as a hierarchy of layers, each supporting the one above it and using those below it.

The purpose of this chapter is to provide a brief description of the International Standard Organization's Open System Interconnect (ISO/OSI) model and Internet protocol suite focusing on the three most important protocols in it:

I. Transmission control protocol (TCP)

II. User Datagram Protocol (UDP)

III. Internet protocol version 4

2.2 ISO/OSI Network Model

The ISO/OSI model is widely used for describing layered network communications and network protocols (Table 2-1).

Layer Number	Layer Name	Data Unit	Protocols
7	Application Layer	Data	HTTP, POP3, SIP, etc.
6	Presentation Layer		SMB, etc.
5	Session Layer		NetBIOS, etc.
4	Transport Layer	Segment	TCP, UDP, etc.
3	Network Layer	Packet	IP, ICMP, ARP, etc.
2	Data Link Layer	Frame	Ethernet, 802.11 (WiFi), etc.
1	Physical Layer	Bit	Wire, Radio Waves, etc.

Table 2-1: ISO/OSI Model Layers [11].

[4] Also called TCP/IP stack/model or DoD model.

The OSI model describes how data is moved between the independent layers and is divided into tasks that the seven layers perform. The purpose of a layer is to provide services to a layer above it and to receive services from the layer below it. Layers are transparent, so that the network layer of a device is able to communicate directly with the network layer of another device.

2.2.1 Physical Layer

The physical layer conveys the bit stream through the network at the electrical and mechanical level. The physical layer specifications also define characteristics, such as voltage levels, timing of impulses, physical data rates, maximum transmission distance, what modulation is used and physical connectors.

2.2.2 Data Link Layer

The data link layer defines the format of the network data (checksum, addresses etc.) and divides data received from the physical layer into frames. It is also responsible for flow control and checking for errors during the transmission.

Ethernet data link layer contains two sublayers, the Media Access Control (MAC), and the Logical Link Control (LLC).

The IEEE 802.3 specification defines MAC addresses, which enable multiple devices to uniquely identify each other at the data link layer. The MAC sublayer manages protocol access to the physical network medium. It controls how a network device gains access to, and permission to transmit data.

LLC manages communications between devices over a single link of a network. It controls frame synchronization, flow control and error checking.

2.2.3 Network Layer

The network layer is responsible for wrapping data received from the data link layer into packets and relaying them to the appropriate recipient. It also has to manage routing, i.e. choose the best path (shortest or the one with the smallest delay, for example) to the destination.

2.2.4 Transport Layer

The transport layer receives data from the session layer, divides it into blocks and ensures these arrive at their destination in the correct order. It also adjusts the transmission speed so that the recipient is not overwhelmed with data.

2.2.5 Session Layer

The session layer creates a session between two participants. It controls opening, maintaining and closing the session, and also resuming it after an interruption.

2.2.6 Presentation Layer

The presentation layer is not concerned with data transfer any more, its primary task is responsible for encryption/decryption and, as the name implies, the presentation of data (semantics, format, character sets etc.).

2.2.7 Application Layer

The uppermost layer is the application layer, which is the layer that is visible to the user. This provides different kinds of protocols for applications.

2.3 TCP/IP Network Model

With the growth of the Internet, the TCP/IP network model has taken ground as the dominant representation for network design. OSI model is actually more robust, but TCP/IP is the primary model used in common practice today.

Figure 2-1 depicts difference between the two models. Instead of OSI's seven layers TCP/IP deployment has only five layers. The TCP/IP model layers perform the same duties as in OSI model. However, the Application Layer in TCP/IP groups the functions of OSI's Application, Presentation and Session Layers. Therefore any process above the transport layer is called an Application in the TCP/IP architecture. In TCP/IP stack socket and port are used to describe the path over which applications communicate. Most application level protocols are associated with one or more port number.

In addition, the Data Link Layer and Physical Layer are grouped together to become the Network Access layer. TCP/IP architecture makes use of existing Data Link and Physical Layer standards, rather than defining its own. Many RFCs describe how IP utilizes and interfaces with the existing data link protocols such as Ethernet, Token Ring, FDDI, HSSI, and ATM.

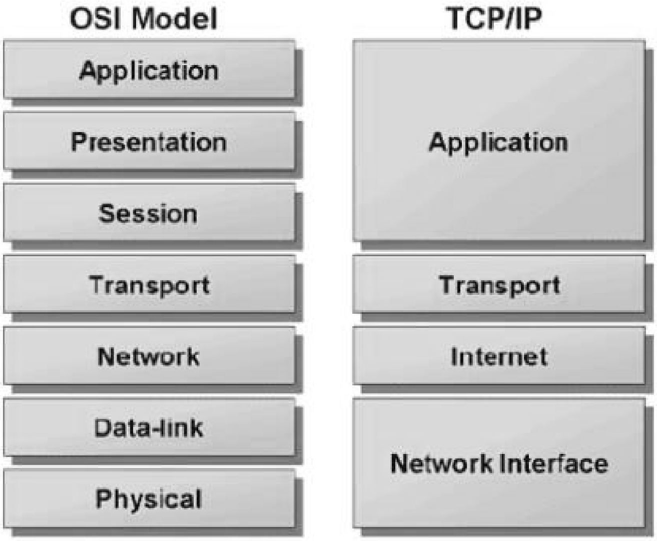


Figure 2-1: TCP/IP model versus OSI model.

In TCP/IP architecture, there are two Transport layer protocols the TCP and the UDP. Both protocols are useful for different applications.

The IP is the primary protocol in the TCP/IP Network/Internet Layer. All upper and lower layer communications must travel through IP as they are passed through the TCP/IP protocol stack. Moreover, there are many supporting protocols in the Internet Layer, such as ICMP, to facilitate and manage the routing process.

2.3.1 IP

As the previous section has already mentioned, IP is the primary protocol in the Internet Layer of the Internet Protocol Suite and has the task of delivering datagrams from the source host to the destination host solely based on their addresses. For this purpose, IP defines addressing methods and structures for datagram encapsulation (Figure 2-2).

IP Header	Upper Layer Data

Figure 2-2: IP Datagram Encapsulation.

The first major version of IP, now referred to as Internet Protocol Version 4 (IPv4) is the dominant protocol of the Internet, although the successor, Internet Protocol Version 6 (IPv6) is in active, growing deployment worldwide.

IPv4 is defined in RFC 791 (September 1981), which has been republished as STD 5. However, RFC 791 contained some vagaries that were clarified in RFC 1122. As such, IP implementations need to incorporate both RFC 791 and RFC 1122 in order to work reliably and consistently with other implementations [12].

IPv4 uses 32-bit (four-byte) addresses, which limits the address space to 4,294,967,296 possible unique addresses. However, some are reserved for special purposes such as private networks or multicast addresses. This reduces the number of addresses that can potentially be allocated for routing on the public Internet[5]. As addresses are being incrementally delegated to end users, an IPv4 address shortage has been developed.

The IPv4 packet header (Table 2-2) consists of 14 fields, of which the 14th field is optional (Options field).

Bit Offset	0 - 3th bit	4th - 7th bit	8th - 13th bit	14th - 15th bit	16th - 18th bit	19th - 31th bit
0 bit	Version	Internet Header Length	Differentiated Services Code Point	Explicit Congestion Notification	Total Length	
32th bit	Identification				Flags	Fragment Offset
64th bit	Time to Live		Protocol		Header Checksum	
96th bit	Source IP Address					
128th bit	Destination IP Address					
160th bit	Options(if Header Length > 5)					
160th or 192th+ bit	Data					

Table 2-2: IPv4 Packet Structure [13].

A brief description of IPv4 header's fields are the followings:

1. **Version:** For IPv4, this has a value of 4 (hence the name IPv4)
2. **Internet Header Length:** Since an IPv4 header may contain a variable number of options, this field specifies the size of the header

[5] Internet Assigned Numbers Authority (IANA) controls IP address allocation in Internet (public IPs).

3. **Differentiated Services Code Point:** Originally defined as the Type of Service field, this field is now defined by RFC 2474 for Differentiated Services (DiffServ)

4. **Explicit Congestion Notification:** Defined in RFC 3168 and allows end-to-end notification of network congestion without dropping packets. It is an optional feature that is only used when both endpoints support it and are willing to use it

5. **Total Length:** This 16-bit field defines the entire datagram size, including header and data, in bytes. Sometimes subnetworks impose restrictions on the size, in which case datagrams must be fragmented

6. **Identification:** This field is an identification field and is primarily used for uniquely identifying fragments of an original IP datagram

7. **Flags:** A three bit field follows and is used to control or identify fragments

8. **Fragment Offset:** The fragment offset field, measured in units of eight byte blocks, is 13 bits long and specifies the offset of a particular fragment relative to the beginning of the original unfragmented IP datagram

9. **Time to Live (TtL):** This field helps prevent datagrams from persisting (e.g. going in circles) on the Internet. This field limits a datagram's lifetime. Each router that a datagram crosses decrements the TtL field by one. When the TtL field hits zero, the packet is no longer forwarded and is discarded

10. **Protocol:** This field defines the protocol used in the data portion of the IP datagram. IANA maintains a list of IP protocol numbers, which was originally defined in RFC 790

11. **Header Checksum:** It is used for error-checking of the header. At each hop, the checksum of the header must be compared to the value of this field. If a header checksum is found to be mismatched, then the packet is discarded. The method used to compute the checksum is defined within RFC 1071

12. **Source address:** An IPv4 address is a group of four octets for a total of 32 bits. This address is the address of the sender of the packet

13. **Destination address:** Identical to the source address field but indicates the receiver of the packet

14. **Options:** Additional header fields may follow the destination address field, but these are not often used

IPv6 is a version of the Internet Protocol that is designed to succeed Internet protocol IPv4. It was developed by Internet Engineering Task Force (IETF), and is described in Internet standard document RFC 2460, published in December 1998.

The main driving force for the redesign of Internet protocol was the foreseeable IPv4 address exhaustion. Also, version 6 adapts in IP protocol some important features , such as QoS and network security and simplifies aspects of address assignment and network renumbering when changing Internet connectivity providers.

2.3.2 TCP

Transmission Control Protocol is defined in RFC 793, which has been republished as STD 7. However, RFC 793 contained some vagaries which were clarified in RFC 1122. In addition, RFC 2001 introduced a variety of congestion related elements to TCP, which have been included into the standard specification, although this RFC was superseded by RFC 2581. Finally, extensions are given in RFC 1323. As such, TCP implementations need to incorporate RFC 793, RFC 1122, RFC 1323 and RFC 2581 in order to work reliably and consistently with other implementations [12, 14].

TCP is a connection-oriented protocol that is responsible for reliable communication between two end processes. The unit of data transferred is called a stream, which is simply a sequence of bytes.

Among the services TCP provides are stream data transfer, reliability, efficient flow control, full-duplex operation, and multiplexing.

With stream data transfer, TCP delivers an unstructured stream of bytes identified by sequence numbers. This service benefits applications because the application does not have to chop data into blocks before handing it off to TCP. Also, TCP can group bytes into segments and pass them to IP for delivery.

By providing connection-oriented end-to-end reliable packet delivery, TCP offers reliability. It does so by sequencing bytes with a forwarding acknowledgment number that indicates to the destination the next byte the source expects to receive. Bytes not acknowledged within a specified time period are retransmitted. The reliability mechanism of TCP allows devices to deal with lost, delayed, duplicate, or misread packets. A time-out mechanism allows devices to detect lost packets and request retransmission.

TCP uses a sliding window flow control mechanism. In each TCP segment, the receiver specifies in the receive window field the amount of additional received data that is willing to buffer for the connection. The sending host can send only up to that amount of data before it must wait for an acknowledgment and window update from the receiving host [14].

Since many network applications may be running on the same machine, computers need something to make sure the correct software application on the destination computer gets the data packets from the source machine, and some way to make sure replies get routed to the correct application on the source computer. This is accomplished through the use of the TCP “port numbers”. The combination of IP address of a network node and its port number is known as a “socket” or an “endpoint”. TCP establishes connections or virtual circuits between two “endpoints” for multiplexing and reliable communications.

The port numbers are divided into three ranges[6]:

I. The *Well Known Ports* are those in the range 0 - 1023.

II. The *Registered Ports* are those in the range 1024 – 49151.

III. The *Private Ports* are those in the range 49152 – 65535.

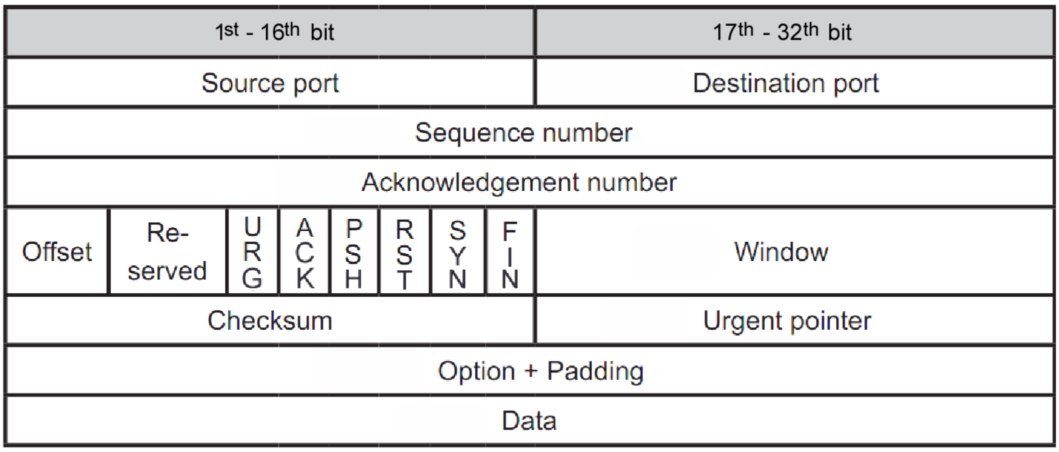
The TCP header (Figure 2-3) consists of 16 fields. The 16th field is optional (Option + Padding).

A brief description of TCP header's fields are the followings:

1. **Source Port:** Identifies the port number in use by the application that is sending the data

[6] Both Well Known and Registered Ports are assigned by the IANA for major protocols.

2. **Destination Port:** Identifies the 16 bits target port number of the application that is to receive this data



<table>
<tr><th colspan="8">1st - 16th bit</th><th>17th - 32th bit</th></tr>
<tr><td colspan="8">Source port</td><td>Destination port</td></tr>
<tr><td colspan="9">Sequence number</td></tr>
<tr><td colspan="9">Acknowledgement number</td></tr>
<tr><td>Offset</td><td>Re-served</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td>Window</td></tr>
<tr><td colspan="8">Checksum</td><td>Urgent pointer</td></tr>
<tr><td colspan="9">Option + Padding</td></tr>
<tr><td colspan="9">Data</td></tr>
</table>

Figure 2-3: TCP Header [15].

3. **Sequence number:** Each byte of data sent across a virtual circuit is assigned a somewhat unique number. The Sequence number field is used to identify the number associated with the first byte of data in this segment

4. **Acknowledgement number:** Identifies the next byte of data that a recipient is expecting to receive

5. **Offset:** Indicates where the data begins

6. **Reserved:** This field is for future use

7. **Control Flags:** Carry a variety of control information. The control bits may be:

   I. *URG:* Urgent pointer field in use

   II. *ACK:* Indicates whether frame contains acknowledgement

   III. *PSH:* Data has been "pushed". It should be delivered to application right away

   IV. *RST:* Indicates that the connection should be reset

   V. *SYN:* Synchronize sequence numbers

   VI. *FIN:* Used to release a connection

8. **Window:** Specifies the size of the sender's receive window, that is, the buffer space available in octets for incoming data

9. **Checksum:** Used to store a checksum of the entire TCP segment for error detection

10. **Urgent Pointer:** Communicates the current value of the urgent pointer as a positive offset from the sequence number in this segment. It points to the sequence number of the octet following the urgent data

11. **Option + Padding:** Provides a way to add extra facilities not covered by the regular header. The most important option is the one that allows each host to specify the maximum TCP payload it is willing to accept. A TCP segment's header length must be a multiple of 32 bits. If any options have been introduced to the header, the header must be padded

2.3.3 UDP

UDP is defined in RFC 768, which has been republished as STD 6. However, RFC 768 contained some vagaries that were clarified in RFC 1122. As such, UDP implementations need to incorporate both RFC 768 and RFC 1122 in order to work reliably and consistently with other implementations.

UDP is a connectionless, stateless transport layer protocol in the which provides a simple and unreliable message service for transaction oriented services. UDP is basically an interface between IP and upper layer processes. UDP protocol ports, which are common with TCP ports, distinguish multiple applications running on a single device from one another.

Unlike TCP, UDP adds no reliability, flow control, or error recovery functions to IP. Because of UDP's simplicity, UDP headers contain fewer bytes and consume less network overhead than TCP (Table 2-3). It is often used by applications that need multicast or broadcast delivery, services not offered by TCP.

Bit Offset	1st - 16th bit	17th - 32th bit
1st bit	Source Port	Destination Ports
33th bit	Length	Checksum
65th bit . . .	Data	

Table 2-3: UDP Header [16].

A brief description of UDP header's fields are the followings:

1. **Source Port:** It is an optional field. When used, it indicates the port of the sending process and may be assumed to be the port to which a reply should be addressed in the absence of any other information.
2. **Destination Port:** It has a meaning within the context of a particular Internet destination address.
3. **Length:** Specifies the size of the total UDP message, including both the header and data segments.
4. **Checksum:** Used to store a checksum of the entire UDP message for error detection.

Chapter 3 Description of Open Power Line HAPs

3.1 Introduction

This chapter presents the open HAPs that use the common power lines of a building as communication medium. We use the term open as in market there are a lot of product using proprietary power line communication techniques, which are out of the scope of this research ("IN ONE" by Legrand S.A., etc).

The advantages of using the already existed electrical wiring of a house is first of all the compatibility with legacy home infrastructure and the availability of transmission signal in any electrical outlet and switch throughout the home.

However, because power line cables are not designed for communication, they provide some challenges for home networking. Frequency selectivity and time variation of the channel frequency, high levels of noise and country-by-country regulatory issues are some of the main considerations [17, 18].

3.2 ISO/OSI Network Model in Smart Homes

A good way of describing where protocols lay in home automation systems is to use OSI model.

In addition, Comité Européen de Normalisation Électrotechnique (CENELEC)[7] has introduced two additional layers at the bottom of standard OSI model for the smart home implementation (Table 3-1). These new layers are "Media" and "Pathways and Spaces".

Pathways is a facility for the installation and placement of cables from the outlets and switches to the main distribution box/es. A pathway can be composed of several components including wall tube, cable tray, conduit, underfloor, access floor, noncontinuous fasteners, and perimeter systems, according to specific country regulations.

Space is an area used for housing the installation and termination of equipment, interconnections and cable, e.g., equipment room, closets, distributors and maintenance boxes/handholes.

[7] CENELEC is responsible for European Standardization in the area of electrical engineering.

Layer Number	Layer Name	Normative References for PLC
7	Application Layer	N/A
6	Presentation Layer	
5	Session Layer	
4	Transport Layer	EN 50090-4-2, etc.
3	Network Layer	
2	Link Layer	
1	Physical Layer	EN 50065-1, EN 50065-2-1, EN 50090-5-1, etc.
0	Media	
-1	Pathways and spaces	CWA 50487, EN 50173-4, ISO/IEC 15018

Table 3-1: The Modified OSI model according to CENELEC [19].

During the building or refurbishing phase of a home the installation of a cabling system causes such little trouble and cost that such a system always shall be installed. In case the use of cables is excluded for a reasonable number of years, at least, the pathways and spaces that would support such a cabling system shall be installed.

The Media layer accommodates a wide range of materials that can be used for signal transmission, e.g., copper cables and fibers. In this research we will examine the HAPs that use the power lines in Media layer.

3.3 European Regulation Basics in PLCs

The European PLC regulation norm is called EN 50065-1:"Signalling on low-voltage electrical installations in the frequency range 3 kHz to 148.5 kHz". It defines the allowed frequency ranges of power line communication, maximum signal amplitudes, as well as the limits of the interference to the surrounding frequency bands.

The defined sub-bands are the followings:

I. *3 kHz - 95 kHz: R*estricted to electricity suppliers and their licensees

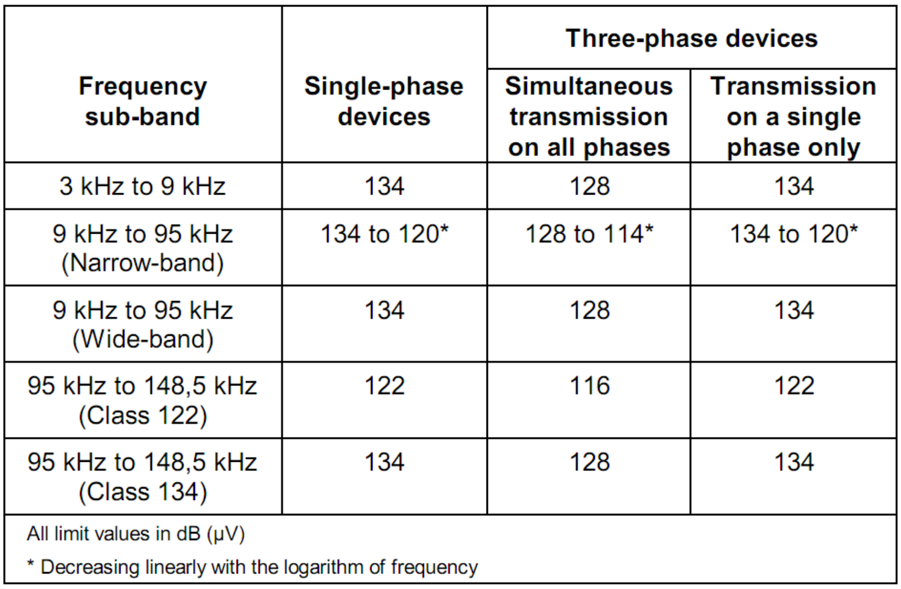
II. *95 kHz - 125 kHz*[8]*:* Available for consumers with no restriction

[8] Equipment for use in this frequency band is designated as either Class 122 or as Class 134 equipment.

III. *125 kHz - 140 kHz:* Available for consumers only with carrier-sense multiple access (CSMA) protocol

IV. *140 kHz - 148.5 kHz:* Available for consumers with no restriction

Maximum output levels (Vpeak) for all sub-bands shown in Table 3-2. Measurements have to be done according to the CISPR 16-1 publication.



Frequency sub-band	**Single-phase devices**	**Three-phase devices**	
		Simultaneous transmission on all phases	**Transmission on a single phase only**
3 kHz to 9 kHz	134	128	134
9 kHz to 95 kHz (Narrow-band)	134 to 120*	128 to 114*	134 to 120*
9 kHz to 95 kHz (Wide-band)	134	128	134
95 kHz to 148,5 kHz (Class 122)	122	116	122
95 kHz to 148,5 kHz (Class 134)	134	128	134

All limit values in dB (µV)

* Decreasing linearly with the logarithm of frequency

Table 3-2: Summary of maximum transmission levels (EN 50065-1).

The EN 50065-1 norm defines neither a data rate nor a modulation scheme, but because of the quite narrow frequency range of each band, only relatively low data rates are possible.

3.4 Description of X10 Power Line Protocol

X10 is an international industry standard that is used for communicating electronic appliance via electrical power network. A radio based transport is also defined.

X10 was developed in 1975 by Pico Electronics (PICO) of Glenrothes, Scotland, UK in order to allow remote control of home devices and appliances. The PICO engineers subsequently relocated to Hicksville, New York and continued with their efforts. It was their 10th project, hence the term X10, which has been used to describe the signalling technique. The X10 team subsequently named their new home automation company X10 Ltd., and relocated to Closter, New Jersey. Later X10 Wireless Technology Inc., a branch of the original company was established whose purpose was to broaden the market by wireless devices.

Today, several companies in addition to X10 Ltd produce X10 compatible devices for home automation. They are SmartLinc, Powerline Carrier Systems (PCS), Leviton, Monterey, Marmitek and others.

3.4.1 Media Layer

The X10 protocol uses mainly electrical power lines as transmission media but in order to allow the operation of wireless keypads, remote switches, and the like, a radio wave media is also defined.

3.4.2 Physical Layer

X10 transmissions are synchronised to the zero crossing point of the AC power line (Figure 3-1).

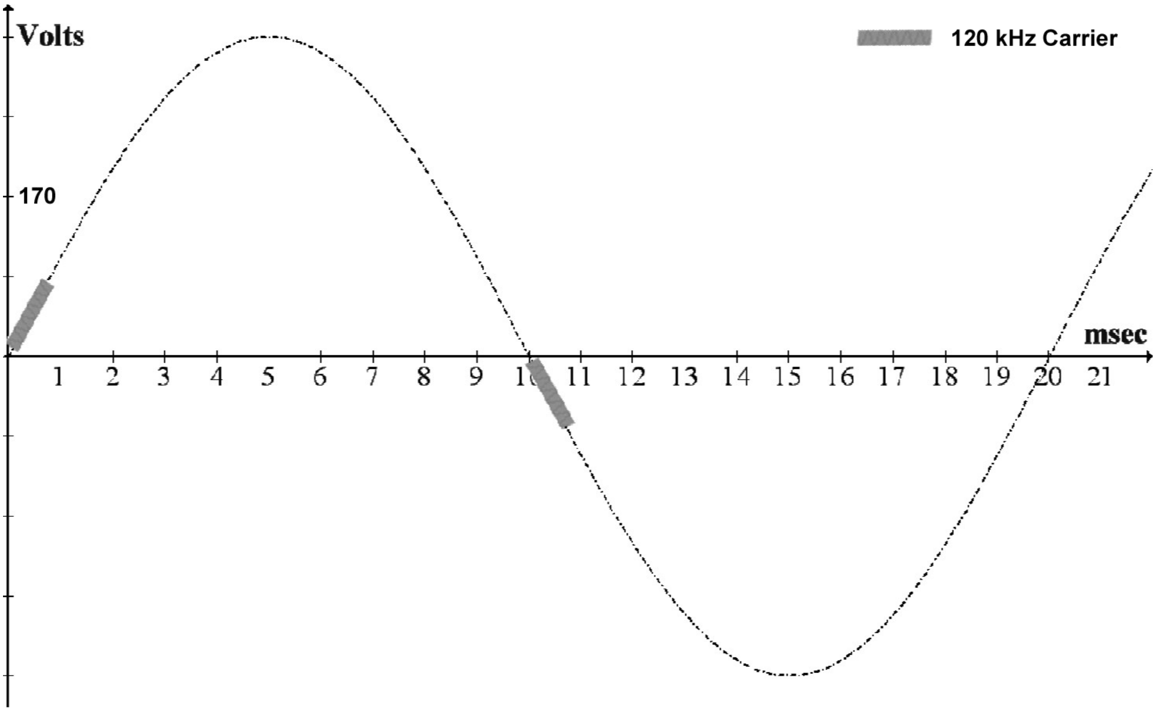


Figure 3-1: X10 Transmission (bits: 110).

X10 receivers detect the presence of 120 kHz signal in X10 information window just after each zero crossing point of the power line voltage. The X10 receiving window begins approximately 250 μs and ends 900 μs after a zero crossing. In this window 48 or more cycles of 120 kHz carrier are accepted as high data logic and fewer than 48 as a low data logic[9] [20]. Since two bits of X10 frame is transmitted per 50 Hz power line period, the protocol has a raw speed of 100 bps.

For most X10 receivers, the 120 kHz signal must have a minimum amplitude of 50 millivolts peak to peak to be detected reliably.

X10 wireless devices operate at a frequency of 433 MHz in Europe and a different frequency in the rest of the world. These wireless devices send X10 frames in the same way as in power lines (OOK modulation). An X10 controller then provides a bridge between radio frequency carrier and power line media.

3.4.3 Data Link Layer

In order to provide a predictable starting point, every X10 frame always begins with the alignment/start code "1110" (Figure 3-2). The start code nibble is unique and can be found only at the start of the X10 frame. Immediately after the start code, a house code (A–P) is sent and after the house code, comes a function code. Except for start code nibble the bits of both house and function code should be transmitted in true and complement form on alternate half cycles of the power line.

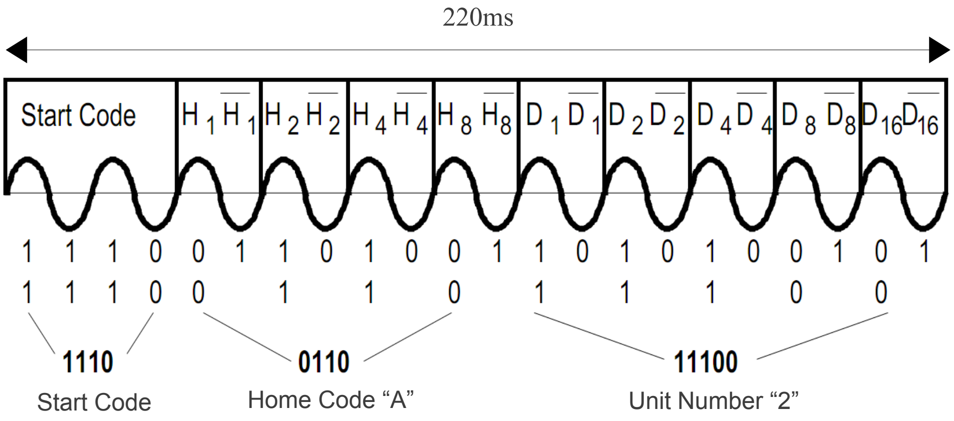

220ms
Start Code
H_1 $\overline{H}_1$ H_2 $\overline{H}_2$ H_4 $\overline{H}_4$ H_8 $\overline{H}_8$ D_1 $\overline{D}_1$ D_2 $\overline{D}_2$ D_4 $\overline{D}_4$ D_8 $\overline{D}_8$ D_{16} $\overline{D}_{16}$
1 1 1 0 0 1 1 0 1 0 0 1 1 0 1 0 1 0 0 1 0 1
1 1 1 0 0 1 1 0 1 1 1 0 0
1110
Start Code
0110
Home Code "A"
11100
Unit Number "2"

Figure 3-2: X10 Standard Frame [20].

Function codes may specify a unit number code (1–16) or an actual command code, the selection between the two modes being determined by the last bit, where zero means unit number and one means command (Table 3-3).

[9] On/Off Keying modulation(OOK).

House Code	H8	H4	H2	H1	Unit Number	D8	D4	D2	D1	D16	Command Code	D8	D4	D2	D1	D16
A	0	1	1	0	1	0	1	1	0	0	ON	0	0	1	0	1
B	1	1	1	0	2	1	1	1	0		OFF	0	0	1	1	
C	0	0	1	0	3	0	0	1	0		DIM	0	1	0	0	
D	1	0	1	0	4	1	0	1	0		BRIGHT	0	1	0	1	
E	0	0	0	1	5	0	0	0	1		ALL LIGHTS ON	0	0	0	1	
F	1	0	0	1	6	1	0	0	1		ALL UNITS OFF	0	0	0	0	
G	0	1	0	1	7	0	1	0	1		ALL LIGHTS OFF	0	1	1	0	
H	1	1	0	1	8	1	1	0	1		EXTENDED CODE 1	0	1	1	1	
I	0	1	1	1	9	0	1	1	1		HAIL REQUEST	1	0	0	0	
J	1	1	1	1	10	1	1	1	1		HAIL ACK.	1	0	0	1	
K	0	0	1	1	11	0	0	1	1		EXTENDED CODE 3	1	0	1	0	
L	1	0	1	1	12	1	0	1	1		UNUSED	1	0	1	1	
M	0	0	0	0	13	0	0	0	0		EXTENDED CODE 2	1	1	0	0	
N	1	0	0	0	14	1	0	0	0		STATUS "ON"	1	1	0	1	
O	0	1	0	0	15	0	1	0	0		STATUS "OFF"	1	1	1	0	
P	1	1	0	0	16	1	1	0	0		STATUS REQUEST	1	1	1	1	

Table 3-3: X10 Codes [20].

Every X10 frame is sent twice without a gap. After transmission of the group of twin frames, must be at least a tree power line cycles silence before the next frame transmission (Figure 3-3). The one exception to above rules is the frames that carry “BRIGHT” and DIM” command codes. These should be transmitted continuously (singles) with no gaps between frames.

4 bits	5 bits
House Code	Function Code

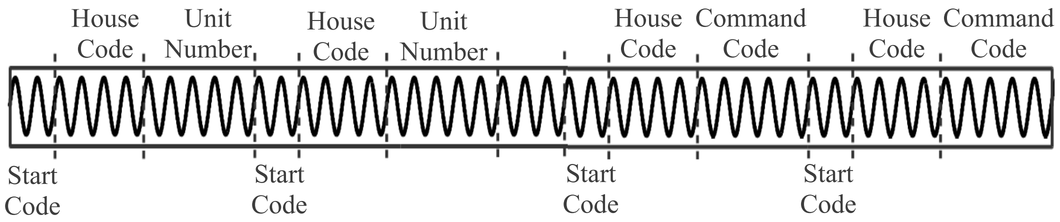


Figure 3-3: X10 Standard Message and Standard Frame Transmission [20].

X10 standard messages do not support more advanced control over the dimming speed, direct dim level setting and group control (scene settings). This is done via extended message set. However support for all extended messages is not mandatory, and a lot of modules implement only the standard message set.

In extended X10 frames, the extended message code is followed by extra bits which can represent sensors value, extra commands, etc. There should be no gaps between the extended message code and the actual data, and no gaps between data bytes (Figure 3-4).

29 Mains Cycles

4 bits	5 bits	4 bits	8 bits	8 bits
House Code	Extended Code	Unit Code	Data Byte	Type & Command

Figure 3-4: Extended Message Format [20].

As the power lines are the media for both transmission and reception, it is required that transmitters avoid message collisions mechanism where possible, and also that, when a collision does occur, it can be detected and the conflict resolved. In order to do this, the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. should be adopted.

When a transmitter has a message it wishes to transmit, it must wait for access to the power line for either 8, 9, or 10 half mains cycles during which the power line must have been continuously clear of 120 kHz carrier[10]. If a carrier is detected, it must restart its access timing and wait for another 8, 9, or 10 cycles.

After power line access has been achieved, the transmitter must check the line during the transmission of '0' bits (no carrier) to see that no other transmitter is transmitting. If a collision occurs, the transmitter must abort its transmission immediately and again go though the power line access procedure.

The choice of 8, 9, or 10 half cycles is chosen randomly for each power line access attempt.

3.5 Description of INSTEON

INSTEON is a cost effective, robust and reliable dual media home automation protocol. It is enabling peer-to-peer technology, meaning that any device can transmit, receive, or repeat other messages, without requiring a master controller or complex routing software.

INSTEON was designed in order to overcome the poor performance of X10 protocol keeping the cost of implementation as low as possible. INSTEON and X10 protocol can coexist with each other on power lines without mutual interference.

[10] All messages are assumed to have equal priority.

SmartLabs Technology, a division of SmartLabs, Inc. developed INSTEON since 2001. SmartLabs is organized into three companies: Smarthome Direct, which includes Smarthome.com, the SmartLabs Design, creators of home control products and the SmartLabs Technology, the pioneering architects of INSTEON.

Despite the ease of implementation, INSTEON has not been developed for 50 Hz power lines yet[11].

3.5.1 Media Layer

INSTEON's nodes communicates with each other using the INSTEON protocol over power lines or radio waves.

The Figure 3-5 shows INSTEON devices that communicate by power line only (PL), radio waves only (RF), and both (RP).

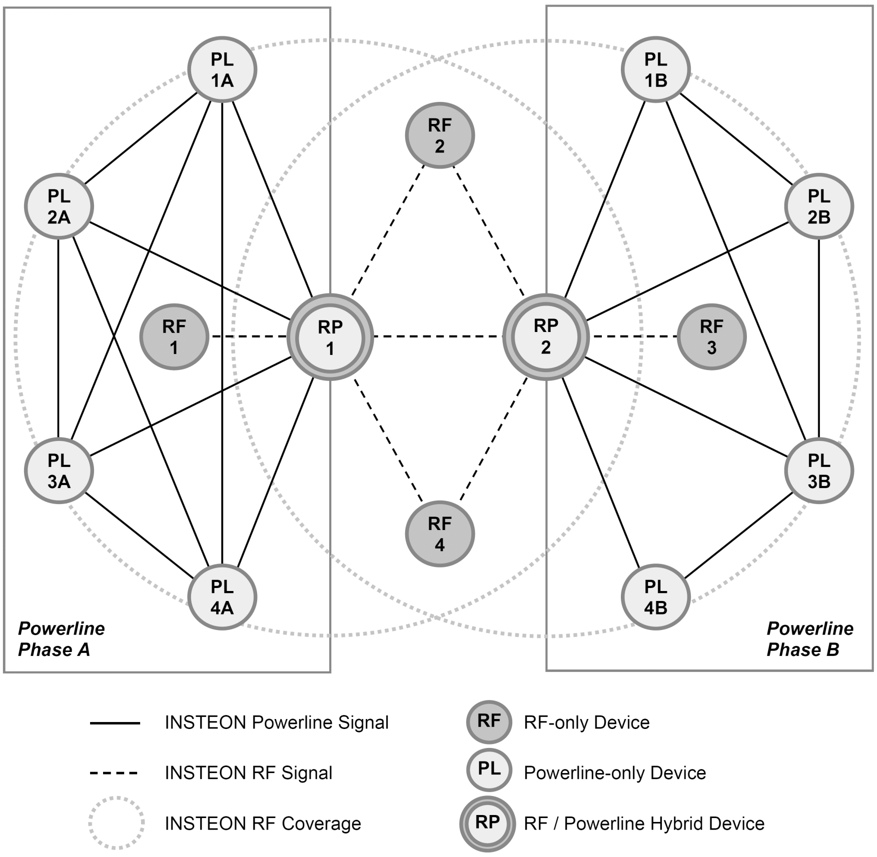


Figure 3-5: INSTEON Network Implementation Example [21].

[11] http://www.insteon.net/faq-home.html#12

3.5.2 Physical Layer

There is no specification for physical layer in both power line and radio waves for Europe.

In USA, INSTEON uses a Frequency-Shift Keying (FSK) modulated carrier in 900 Mhz band for wireless operation, which is permitted for unlicensed operation.

Also, for power line communication INSTEON uses a 131.65 kHz carrier signal with 4.64 Vp-p amplitude in 5 ohm load. The carrier is modulated with alternating Binary Phase-Shift Keying (BPSK) bit modulation and the bit-stream has encoded with Non-Return to Zero code (NRZ).

INSTEON frames are transmitted during the zero crossing quiet time and begin 800 microseconds before a zero crossing and last until 1023 microseconds after the zero crossing (Figure 3-6).

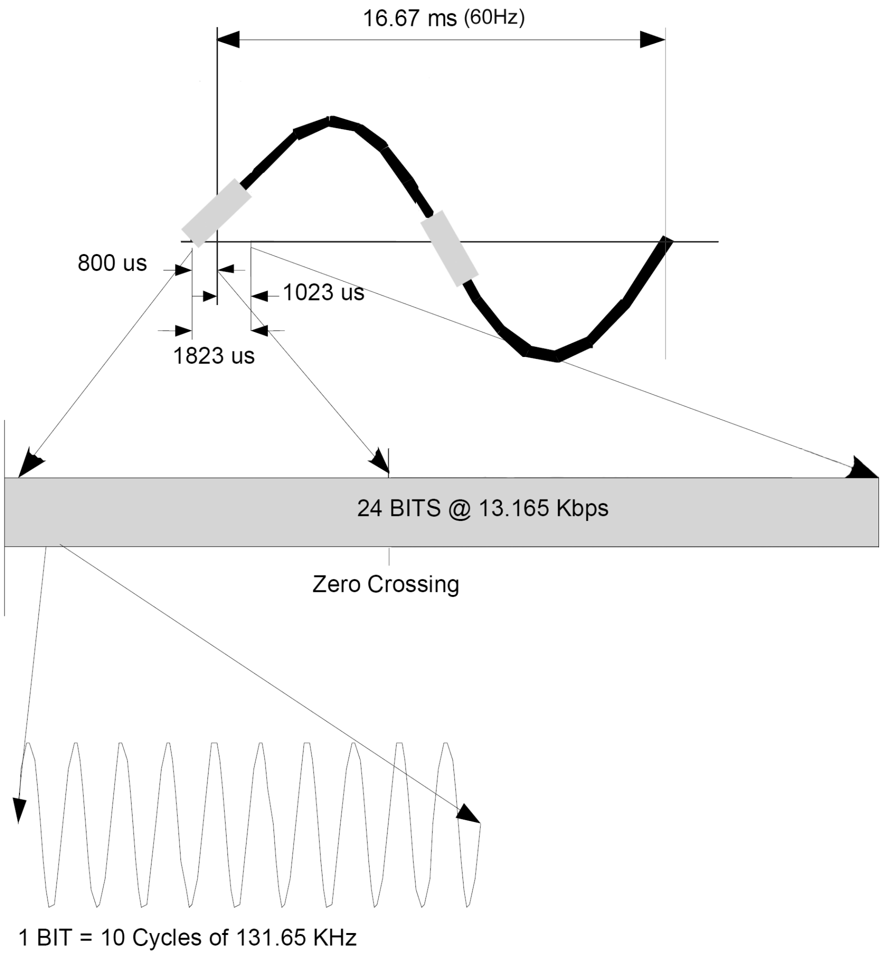


Figure 3-6: INSTEON frames over 60Hz PLN [21].

3.5.3 Data Link Layer

As X10 protocol, INSTEON has two types of messages, 10-byte standard length message and 24-byte extended length message (Figure 3-7).

INSTEON Standard Message				
3 Bytes	3 Bytes	1 Byte	2Bytes	1 Byte
Source Address	Destination Address	Flags	Commands 1, 2	CRC

INSTEON Extended Message					
3 Bytes	3 Bytes	1 Byte	2 Bytes	14 Bytes	1 Byte
Source Address	Destination Address	Flags	Commands 1, 2	User Data	CRC

Figure 3-7: INSTEON's Messages [21].

The only difference between the two is that the extended message contains 14 user data bytes not found in standard message.

The source address is the 24-bit number that uniquely identifies the INSTEON device originating the frame being sent. This is the address or ID code of the INSTEON device. During manufacture, this unique ID code is stored in each device in non volatile memory.

The Table 3-4 enumerates the meaning of the bit fields in the Flags byte.

Bit	Meaning
8	000 : Direct Message 001 : Acknowledgment(ACK) of Direct Message 101 : Negative ACK(NACK) of Direct Message 110 : Group Broadcast Message 011 : ACK of Group Cleanup Direct Message 111: NACK of Group Cleanup Direct Message 010 : Group Cleanup Direct Message 100 : Broadcast Message
7	
6	
5	0 : Standard Message 1 : Extended Message
3,4	00 – 11 : Number of Hops have been Left.
1,2	00 – 11 : Number of Allowed Hops

Table 3-4: Flags Field of INSTEON Message [21].

Broadcast messages contain general information with no specific destination and are not acknowledged.

Group Broadcast messages are directed to a group of devices. After sending a Group Broadcast message to a group of devices, the message originator then sends a direct Group Cleanup message to each member of the group individually, and waits for an acknowledgement back from each device.

The first four bits in Flags field of INSTEON message manage message repeating. As mentioned above, all INSTEON nodes are capable of repeating messages by receiving and retransmitting them. Without a mechanism for limiting the number of times a message can be retransmitted and an uncontrolled ‘data storm’ of endlessly repeated messages could saturate the network. To solve this problem, INSTEON message incorporates “Allowed Hops” and “Hops have been Left” fields. An “Allowed Hops” value of zero tells other devices within range not to retransmit the message. A no zero “Allowed Hops” value tells devices receiving the message to retransmit it depending on the “Hops have been Left” field. If the “Hops have been Left” value is no zero, the receiving device decrements the “Hops have been Left” value by one, then retransmits the message with the new “Hops have been Left” value. Devices that receive a message with a “Hops have been Left” value of zero will not retransmit that message. Also, a device that is the intended recipient of a message will not retransmit the message, no matter what the “Hops have been Left” value is.

The Commands field in INSTEON message made up of Commands 1 byte and Commands 2 byte. The usage of this sub fields depends on the message type (Direct, Broadcast, etc.). For example, in Direct messages the Command 1 value 0x11 meaning switch on and has a parameter in Command 2 ranging from 0x00 to 0xFF representing the on level [22].

INSTEON messages sent over the power lines are fragmented into frames, with each frame sent in conjunction with zero crossing of AC voltage on power line (Figure 3-6).

Standard Messages use a group of five frames and Extended Messages use a group of eleven frames, as Figure 3-8 shown.

A Start Frame (SF) is transmitted first in both the Standard and Extended Messages. The remaining message information is transmitted in Body Frames (BF).

Standard Message Frames Group				
SF	BF	BF	BF	BF

Extended Message Frames Group										
SF	BF	BF	BF	BF	BF	BF	BF	BF	BF	BF

Figure 3-8: INSTEON messages Fragmentation [21].

Both Start and Body frames consist of 24 bits and begin with a series of Sync Bits. Following the Sync Bits are four Start Code Bits. The remaining bits in a frame are INSTEON Message bits (Figure 3-9).

The total available space for INSTEON Message bits in a Standard frame group is 84, or 10½ bytes. So, as the standard message consist of 10 bytes the last four bits in a Standard frame group are ignored (stuffing).

Start Frame																							
1	0	1	0	1	0	1	0	1	0	0	1	x	x	x	x	x	x	x	x	x	x	x	x
8 Sync bits								4 Start Code bits				12 INSTEON Message bits											

Body Frame																							
1	0	1	0	0	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
2 Sync bits		4 Start Code bits				18 INSTEON Message bits																	

Figure 3-9: Start and Body Frame Structures [21].

INSTEON transmitters wait for at least one zero crossing after each Standard message and two zero crossings after each Extended message before sending another message (Figure 3-10).

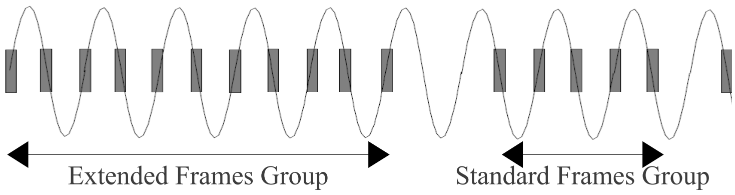


Figure 3-10: Frames Group Transmission [21].

These waiting times after sending power line messages are to allow sufficient time for INSTEON RF devices, if present, to retransmit a power line message.

If the originator of an INSTEON direct message does not receive an ACK from the intended recipient, the message originator will automatically try resending the message up to five times. In case a message did not get through because the number of "Allowed Hops" was set too low, each time the message originator retries a message, it also increases this number up to the limit of three. A larger number of Max Hops can achieve greater range for the message by allowing more devices to retransmit it.

3.6 Universal Powerline Bus (UPB)

UPB is a low speed peer-to-peer protocol designed by Power Control System Inc., of Northridge California. It has two patents issued, the 6,734,784, granted May 11, 2004, which is titled "Zero Crossing Based Powerline Pulse Position Modulated Communication System" and the 6,784,790, granted August 31, 2004, which is titled "Synchronization/Reference Pulse-Based Powerline Pulse Position Modulated Communication System".

UPB protocol can be used in the presence of the most power line protocol (X10, INSTEON, etc.) with no interference between either UPB or them.

UPB devices have not been developed for 50 Hz power lines yet.

3.6.1 Media Layer

The UPB protocol uses only electrical power lines as media transmission.

3.6.2 Physical Layer

The UPB communication technique consists of transmitting Pulse Position Modulated (PPM) frames over the electrical power line. During transmission, one pulse of 40V is superimposed on top of the normal AC power waveform (Figure 3-11). The generation of each UPB pulse is precisely timed to occur in one of four predefined positions before power line zero crossing. The position of each UPB pulse represents 2-bit encoded data (0, 1, 2, or 3 decimal value) [23].

UPB signalling window occurring in the 1080 microseconds preceding each zero crossing and each UPB pulse position is spaced 160 microseconds apart from each other with a drift of ± 40 microseconds.

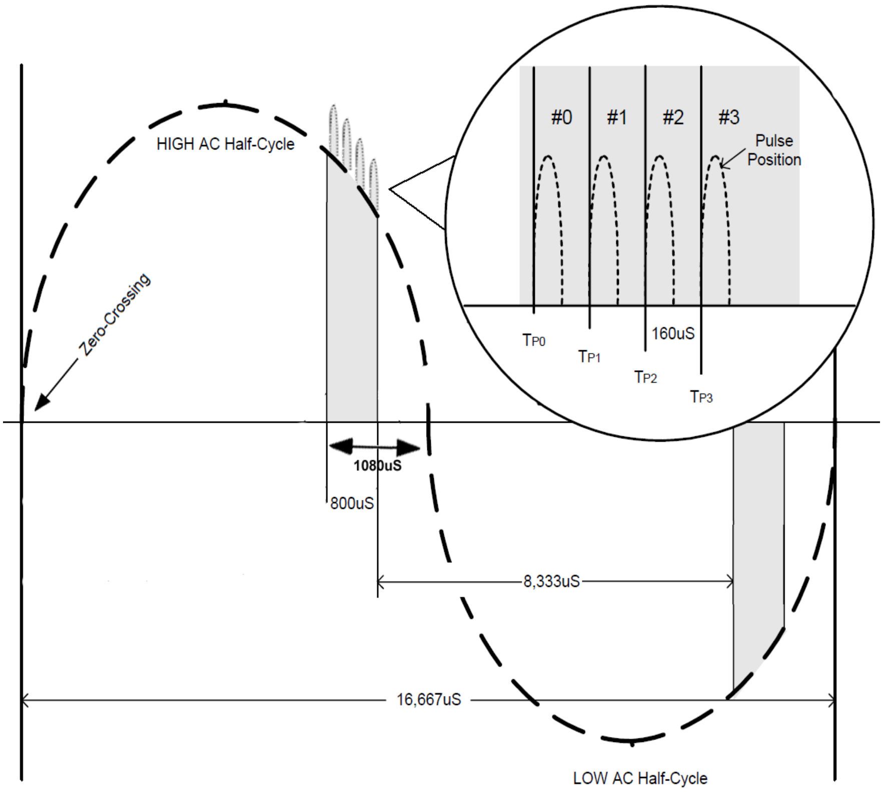


Figure 3-11: UPB Pulse Position (60 Hz)[23].

Since each UPB pulse can encode two bits of digital information, UPS protocol has a raw speed of 240 bps at 60 Hz power lines.

3.6.3 Data Link Layer

The UPB frame has variable length, containing from 7 to 25 bytes, as shown in Figure 3-12.

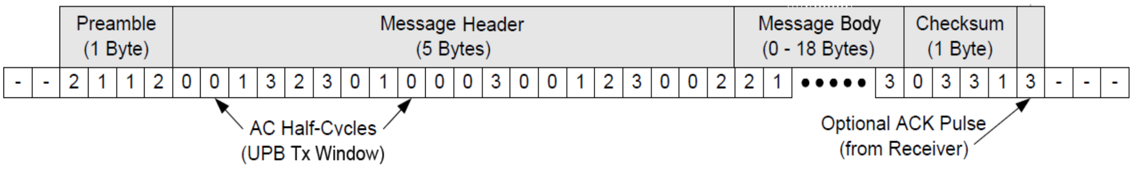


Figure 3-12: UPB Frame [23].

The frame begins with preamble byte that helps receivers to synchronize to the upcoming UPB pulses. After the preamble byte the frame contains a 5-byte header. The message header is used to indicate such information as the message's size, how the message is to be received, whom the frame is intended for, and who sent the frame. The header is broken up into four individual fields as shown in Figure 3-13.

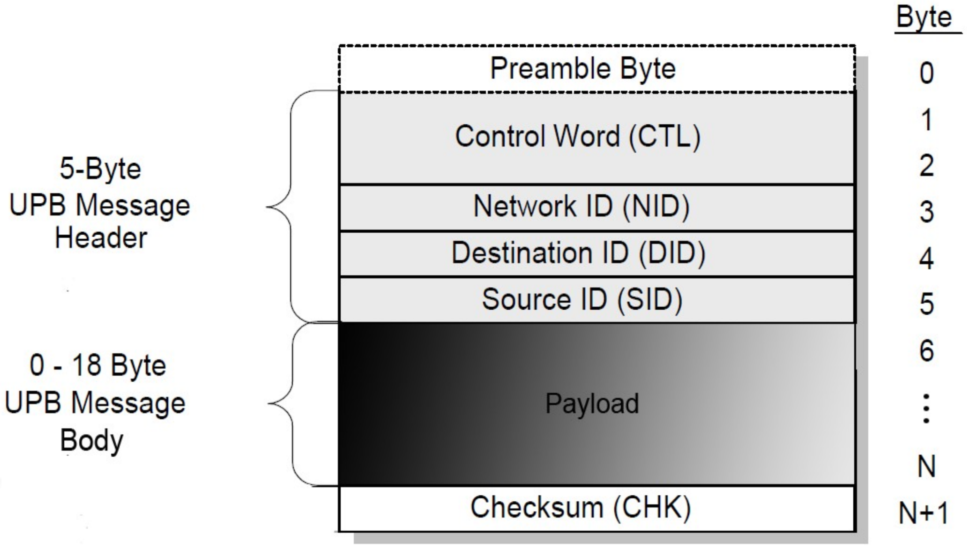


Figure 3-13: UPB Frame Header [23].

A brief description of UPB message header's fields is the following:

1. **Control Word:** Indicates the message's size, how the frame should be received and how it should be responded to, as well as counts the frame transmission sequence information
2. **Network ID:** Indicates which UPB network the frame is intended for. This field allows separate virtual networks to be formed on the same physical network (power line)
3. **Destination ID:** Indicates which individual node or group of node on a particular UPB network, the frame is intended for
4. **Source ID:** Identifies the frame originator
5. **UPB Message Body:** It is a variable length field that can range from 0 to 18 bytes and includes device control, extended message data, etc.
6. **Checksum:** Verifies the integrity of the received frame

At the end of every UPB frame (in the next transmission window) a single UPB pulse takes place, known as the ACK pulse. The ACK Pulse is generated by a UPB receiving node that is used to inform the transmitting node that the UPB frame was accepted. The ACK Pulse is generated in Position #3 of the UPB transmission window.

3.7 EIB/KNX

In June 1996, the European Home Systems Association (EHSA) started an initiative with the aim to converge the three already existed and standarised building automation protocols, BatiBUS, European Installation Bus (EIB) and European Home System (EHS). In April 14th 1999, the new and common organisation has been founded in Brussels as an organisation under Belgian law. The task of this new association is to build upon the existing competencies, technologies and resources of the three industry associations a common standard for homes and building communication. In the long run, the new organisation will replace the three existing associations and it will overtake their tasks. A proposal for a unique specification was presented mid-1999 in its version 1.0. By the end of 1999, the new standard was named “Konnex” (or in short “KNX”)[12].

KNX was ratified by CENELEC as the European Standard EN 50090 in December 2003. In 2006 a large section of the EN 50090 standard was approved for inclusion in the ISO/IEC 14543 International Standard, making KNX a worldwide open standard for home and building control [24].

The EN 50090 standard gives details of the configuration and features of a KNX system. It defines the topological rules for electrically connecting bus nodes and the protocols on how these nodes communicate with each other.

Contrary to already mentioned power line protocols, the KNX implements three more layers, network, transport and application layer. This provides a great deal of flexibility and expandability.

Unfortunately, KNX system implementation costs far more than X10, UPB or INSTEON system implementations. Generally, the investment is only worthwhile if several systems are to be connected with each other or if an installation needs to be flexible enough so that it can be quickly and effectively modified to meet any future

[12] http://www.knx.org/knx-standard/introduction/

changes in use or requirements.

KNX Association, as of June 2010, has over 200 members/manufacturers including Miele & Cie KG, ABB, Hager, Schneider Electric Industries S.A., Bosch, Siemens, etc.

3.7.1 Media Layer

Different types of transmission media can be used to transfer data from one KNX node to another and vice versa:

1. Twisted pair (KNX.TP)
2. Power line (KNX.PL)
3. Radio frequency (KNX.RF)

3.7.2 Physical Layer

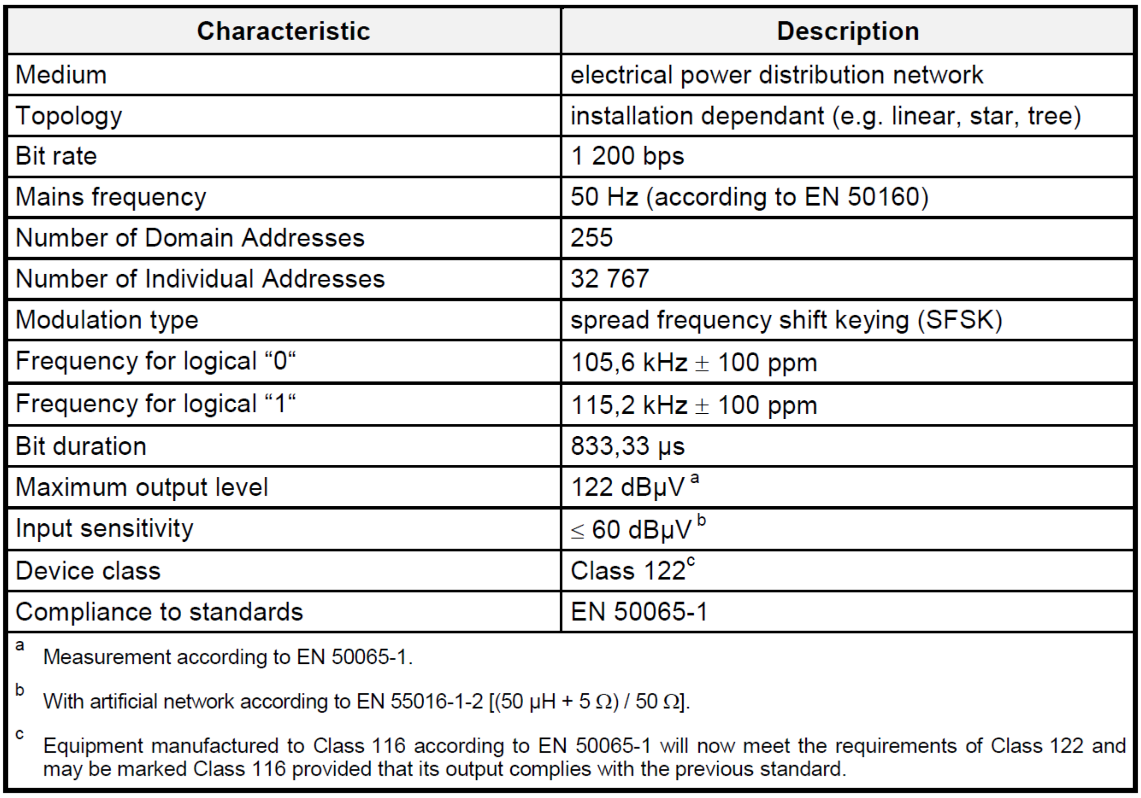
Several physical layers are already defined taking into account the variety of application requirements.

There are two different standards for transmission on twisted pair cable, TP0 and TP1 [25]. TP0 was taken over by Batibus and is popular in France. It is expected that this communication medium will disappear in the future since most manufacturers are switching to TP1. TP1 was introduced with the EIB and is used by most KNX products. KNX TP1 combines high quality transmission in various topologies with low cost hardware. Devices connected to the TP1 can be supplied over the bus. For the physical transfer of the data, balanced baseband signal encoding is used with a data rate of 9600 bps.

For wireless operation, KNX uses Frequency Shift Keying (FSK) with a centre frequency of 868.30 MHz and a typical deviation of 50 kHz for data modulation. With a data rate of 16384 baud, a similar number of frames can be transmitted as with TP1. KNX.RF is still a relatively new aspect of the KNX specification and will play a major role in the future. It lends itself well to the extension of existing wired KNX installations and for the renovation market where high costs prohibit rewiring.

In addition, for power line media there are two standards, PL110 and PL132 [26]. PL110 was taken over by EIB. A 110 kHz centre frequency is modulated with Spread

Frequency Shift Keying (SFSK) with a data rate of 1200 bps (Table 3-5). The transmission starts at the mains zero crossing with a maximum level of 122 dBuV[13]. The PL132 was inherited from EHS and is currently only being employed by a few manufacturers of white goods. It uses 132 kHz signal with Minimum Shift Keying (MSK) modulation to transmit KNX frames with data rate at 2400 bps.



Characteristic	Description
Medium	electrical power distribution network
Topology	installation dependant (e.g. linear, star, tree)
Bit rate	1 200 bps
Mains frequency	50 Hz (according to EN 50160)
Number of Domain Addresses	255
Number of Individual Addresses	32 767
Modulation type	spread frequency shift keying (SFSK)
Frequency for logical “0“	105,6 kHz ± 100 ppm
Frequency for logical “1“	115,2 kHz ± 100 ppm
Bit duration	833,33 µs
Maximum output level	122 dBµV [a]
Input sensitivity	≤ 60 dBµV [b]
Device class	Class 122[c]
Compliance to standards	EN 50065-1

[a] Measurement according to EN 50065-1.

[b] With artificial network according to EN 55016-1-2 [(50 µH + 5 Ω) / 50 Ω].

[c] Equipment manufactured to Class 116 according to EN 50065-1 will now meet the requirements of Class 122 and may be marked Class 116 provided that its output complies with the previous standard.

Table 3-5: General Requirements for Physical Layer PL110 [26].

Today only a few manufacturers support PL110 but they continue to offer a full range of devices for lighting, blinds and heating control. Since practically no products currently exist for PL132, this medium may disappear in the future.

As Figure 3-14 shows each PL110 frame starts with a 4 bits training sequence (0x5) and a 16 bits preamble (0xB0).

[13] It complies with EN 50065-1 and with either EN 50065-2-1 or EN 50065-2-1.

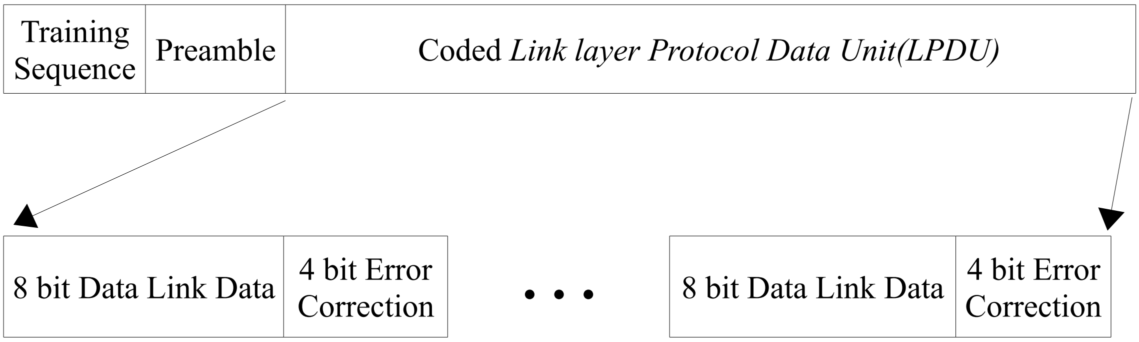


Figure 3-14: PL110 Frame [26].

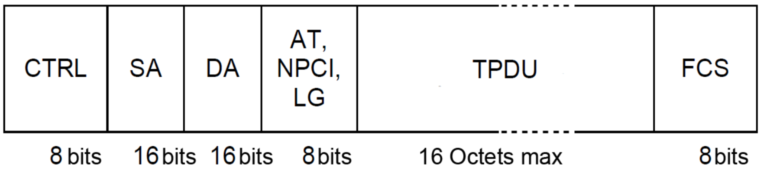
3.7.3 Data Link Layer Type PL110[14] & Network Layer

All frame information, except training sequence and preamble, is coded to a 12-bit character (Figure 3-14).

The LPDU contains the following fields (Figure 3-15, Figure 3-16):

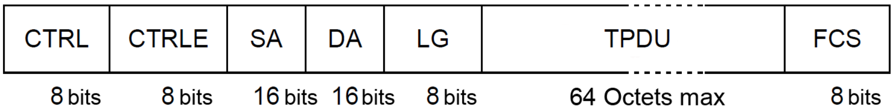
- **Control Field (CTRL):** Contains information about the data link service, its priority (alarm messages, etc.), frame type (standard or extended) and whether the LPDU is a repeated one
- **Source Address (SA):** The originator's unique address
- **Destination Address (DA):** The unique address of destination node or the destination address of a group of nodes (multicast)
- **Address Type (AT):** Defines if the destination address belongs to a single node or a group of nodes
- **Network layer Protocol Control Information (NPCI)**: Controlled by network layer and contains the hop count information for routing
- **TPDU (Transport layer Protocol Data Unit):** The payload from upper layer
- **Length (LG):** Defines TPDU length
- **Check Octet (FCS):** Helps ensure data consistency and reliable transmission
- **Extended Control Field (CTRLE):** Contain the extended frame format parameter and the hop count parameter (network layer information)

[14] All the layers above Data Link layer is media independent.



CTRL	SA	DA	AT, NPCI, LG	TPDU	FCS
8 bits	16 bits	16 bits	8 bits	16 Octets max	8 bits

Figure 3-15: LPDU in Standard Frame [26].



CTRL	CTRLE	SA	DA	LG	TPDU	FCS
8 bits	8 bits	16 bits	16 bits	8 bits	64 Octets max	8 bits

Figure 3-16: LPDU in Extended Frame [26].

KNX.PL110 protocol uses medium access control mechanisms to avoid collision and an ACK / NACK frame must be transmuted from receiver to inform frame originator for frame delivery.

The Acknowledgement frame consists of 20 bits training sequence (0x5) and preamble (0xB0), followed by a single character acknowledging or not acknowledging the received frame (Figure 3-17).

Training Sequence 4 bits	Preamble 16 bits	ACK / NACK 8 bits	Error Correction 4 bits

Figure 3-17: Acknowledgment Frame [26].

3.7.4 Transport Layer

The transport layer provides data transmission over different communication modes. These modes connect transport layer users with each other. The transport layer provides five different communication modes:

I. Point to Multipoint, connectionless (multicast)

II. Point to Domain, connectionless (broadcast)

III. Point to all Points, connectionless (system broadcast)

IV. Point to Point, connectionless

V. Point to Point, connection oriented

As Figure 3-18 shows TPDU frame consist of the transport control field and the Application layer information.

Transport Control Field 6 bits	Application layer information

Figure 3-18: TPDU [27].

3.7.5 Application Layer

Application layer offers a large “toolkit” variety of application services to the application process. These services are different depending on the type of communication used at transport layer. Services related to point to point communication and broadcast mainly serve to the network management, whereas services related to multicast are intended for runtime operation.

3.8 LonTalk[15]

LonTalk® or LON for short is an open protocol for building automation and control networks that was developed by the American company Echelon since 1986. It is designed in such a way that it can be used in centralized building automation controllers as well as in decentralized building control components [24].

LonTalk implements the eight upper layers of CENELEC's protocol stuck standard (Figure 3-19). It provides peer-to-peer communication for networked control and is suitable for implementing both peer-to-peer and master/slave control strategies [28]. This provides a great deal of flexibility and expandability. Small networks are not required to use all the services but as that network grows, the features are there for expansion without having to upgrade software or firmware.

The key to LON system is a microcontroller, the Neuron Chip, which was introduced in 1990 and initially produced by Toshiba and Motorola. Later, Cypress took over from Motorola as the second supplier. The Neuron chip is a complete LON node on a chip as it contains the entire LON protocol stack except physical layer[16].

LON is a global standardized protocol (ANSI/CEA-709.1-B and ISO/IEC DIS 14908) that enables intelligent devices to communicate with each other over a locally operated

[15] The LonTalk protocol, also known as the LONWORKS protocol.

[16] Different type of transceivers is used to connect Neuron Chip to transmission bus.

control network. LON stands for Local Operating Network.

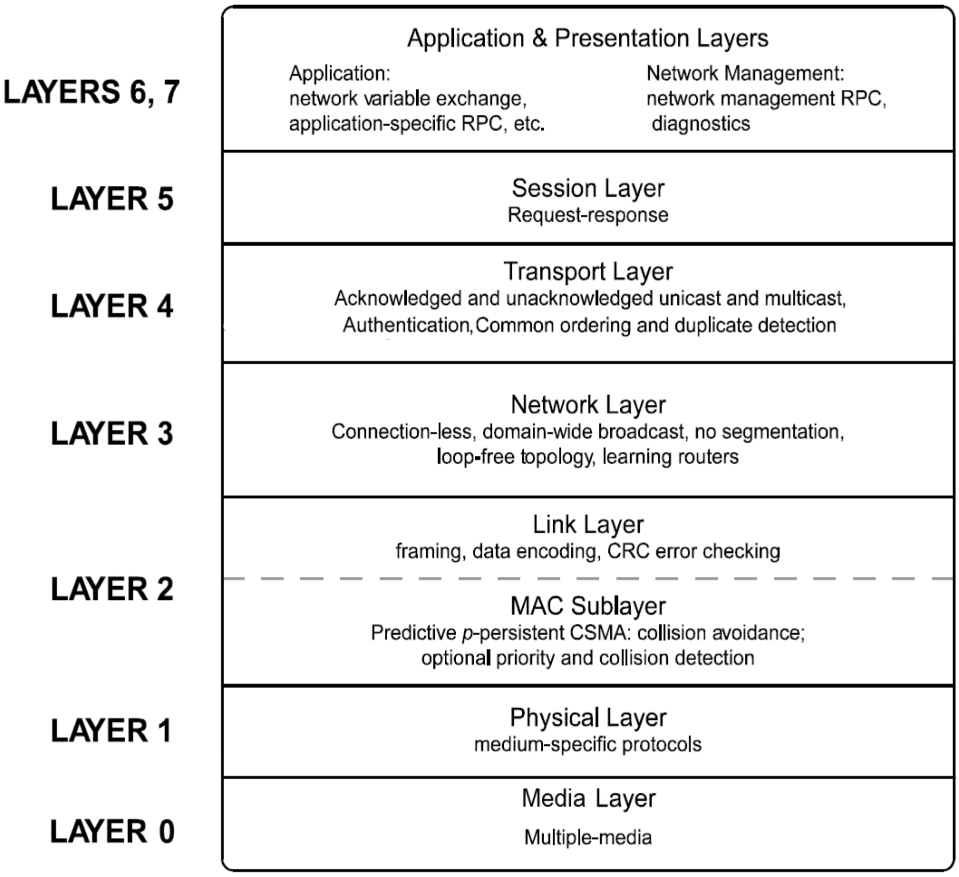

LAYERS 6, 7
Application & Presentation Layers
Application:
network variable exchange,
application-specific RPC, etc.
Network Management:
network management RPC,
diagnostics
LAYER 5
Session Layer
Request-response
LAYER 4
Transport Layer
Acknowledged and unacknowledged unicast and multicast,
Authentication,Common ordering and duplicate detection
LAYER 3
Network Layer
Connection-less, domain-wide broadcast, no segmentation,
loop-free topology, learning routers
LAYER 2
Link Layer
framing, data encoding, CRC error checking
MAC Sublayer
Predictive p-persistent CSMA: collision avoidance;
optional priority and collision detection
LAYER 1
Physical Layer
medium-specific protocols
LAYER 0
Media Layer
Multiple-media

Figure 3-19: LonTalk Protocol Stack [28].

Unfortunately, LON network implementation is very expensive[17]. Implementation of LON network is only worthwhile if several systems are to be connected with each other or if an installation needs to be flexible enough to meet any future changes.

3.8.1 Media Layer

LonWorks technology provides many different communication media options including twisted pairs, power lines, fiber optics, and RF transceivers.

3.8.2 Physical Layer

LON over twisted pairs supports raw data rates from 78.125 kbps to 1.25 Mbps between multiple nodes.

[17] LON implementation is two or three times more expensive than X10 similar implementation.

In addition, variety of RF transceivers are available for wireless communications in many different environments. Both licensed and non licensed versions are available in the 400 - 470 MHz and 900 MHz bands respectively with raw data rate up to 19.5 kbps.

LON protocol over power lines occupies bandwidth from 125 kHz to 140 kHz frequency band[18] (131,579 kHz carrier frequency) as a Binary Phase Shift Keyed (BPSK) modulated carrier with a NRZ coded bit-stream [29]. The raw bits rate is 4 kbps and the transmission maximum level is 122 dBuV[19].

The power line frame consists of a "bit sync" pattern, a "word sync", the upper layers information and two End of Frame (EOF) words (Figure 3-20). The "bit sync" pattern provides clock timing information and the "sync word" pattern provides bit polarity and data alignment information.

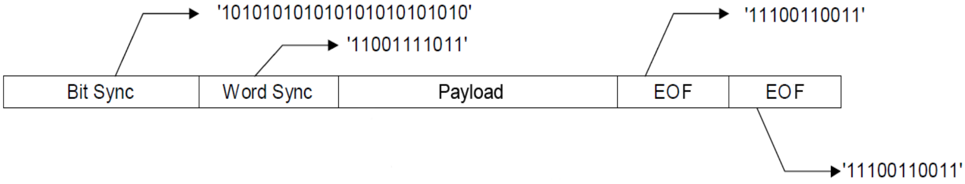


Figure 3-20: Power Line Frame [29].

3.8.3 Data Link Layer

As Figure 3-21 shows data link layer encodes each byte of MAC sublayer data into an 11-bit word.

Byte from MAC Sublayer	Parity bit	0	1	• • •	Byte from MAC Sublayer	Parity bit	0	1

Figure 3-21: Encoded LPDU [29].

At the bottom of the data link layer the MAC sublayer takes place, which employs a collision avoidance algorithm called predictive p-persistent CSMA. Above the MAC layer, for a number of reasons including simplicity and compatibility with the multicast protocol, the link layer supports a simple connectionless service. Its functions are limited

[18] It complies with EN 50065-1 and with either EN 50065-2-1 or EN 50065-2-1.
[19] Please see also Section 3.3.

to framing, frame encoding, and error detection, with no error recovery by retransmission.

LPDU consist of the following fields (Figure 3-22):

- *Header:* Specifies frame priority and logical channel
- *NPDU:* The payload
- *CRC:* For error detection

8 bits		16 bits
Header	NPDU	CRC

Figure 3-22: LPDU [28].

3.8.4 Network Layer

This layer specifies the destination of a message on the LON network. Services provided by this layer include:

- Contains the node address information
- Provides for routing of messages to segment and control network bandwidth usage

An NPDU carries and encapsulates the information of the upper layer. The header defines the protocol version, the payload format, the address, the domain and the length of the payload.

8 - 26 bits	
Header	Payload

Figure 3-23: NPDU [28].

3.8.5 Transport Layer

This layer establishes the type of services required for the node messages depending on the level of reliability required by the application. The services provide are:

- Broadcast addressing
- Unicast addressing

- Multicast addressing
- Repeated service
- Acknowledged service
- Unacknowledged service
- Duplicate packet detection
- Authentication

The level of service required by the application is established when each node is installed on the network. This is all handled by a network management installation tool and the node design.

3.8.6 Session Layer

The session layer provides a simple request-response mechanism for access to remote servers. This mechanism provides a platform upon which application specific remote procedure calls can be built. The network management protocol, for example, depends upon the request-response mechanism in the session layer.

3.8.7 Aplication and Presentation Layers

These layers include services to simplify development of application programs to interface to specific sensors, actuators, and external microprocessors.

Chapter 4 IP Convergence of PLN's HAPs

4.1 Introduction

IP networks are not like power line networks which are plain physical buses by nature. This implies that all nodes connected to mains power network will by default receive all information transmitted on the network. In addition when a new node is added to the network it is not necessary that other devices on the network become aware of it before they start exchanging information.

On the contrary, IP networks are not physical but logical in nature. Thus, before a HAPoIP node can transmit a packet to another node on an IP network, it should be aware of how to send a packet distinctly to that device (IP address, port and gateway).

Another significant difference between physical and logical networks is the case of typical physical networks where it is possible to calculate the fixed upper bounds of the time it needs a frame to traverse from one device to another, once the frame is released on the bus. This is not always possible for IP networks. The deviation of packet delivery times between HAPoIP nodes on an IP network, is much higher than those experienced with native HAP nodes.

Taking into consideration all the above issues, this chapter deals with the integration of power line network home automation protocols on top of IP protocol, called HAPoIP.

4.2 The Transport Mechanism

In order HAPs frames to travel throughout an IP network, an adaptation layer must be defined between power line home automation and Internet protocols.

Three most common mechanisms used to transport data above IP protocol are Raw IP, TCP and UDP (Figure 4-1).

Although it is possible to use Raw IP to transport the HAPoIP PDU, from the application point of view a lot of services that already run in/above TCP or UDP layer have either to be excluded from the whole implementation or to be implemented again in HAPoIP adaptation layer. For example, as both TCP and UDP protocols can detect errors in received packets from remote node, there is no necessity to implement an error detection mechanism in HAPoIP layer. Also, protocols as Dynamic Host Configuration Protocol (DHCP), Bootstrap Protocol (BOOTP), etc. that offer networking facilities to IP nodes can be easily implemented.

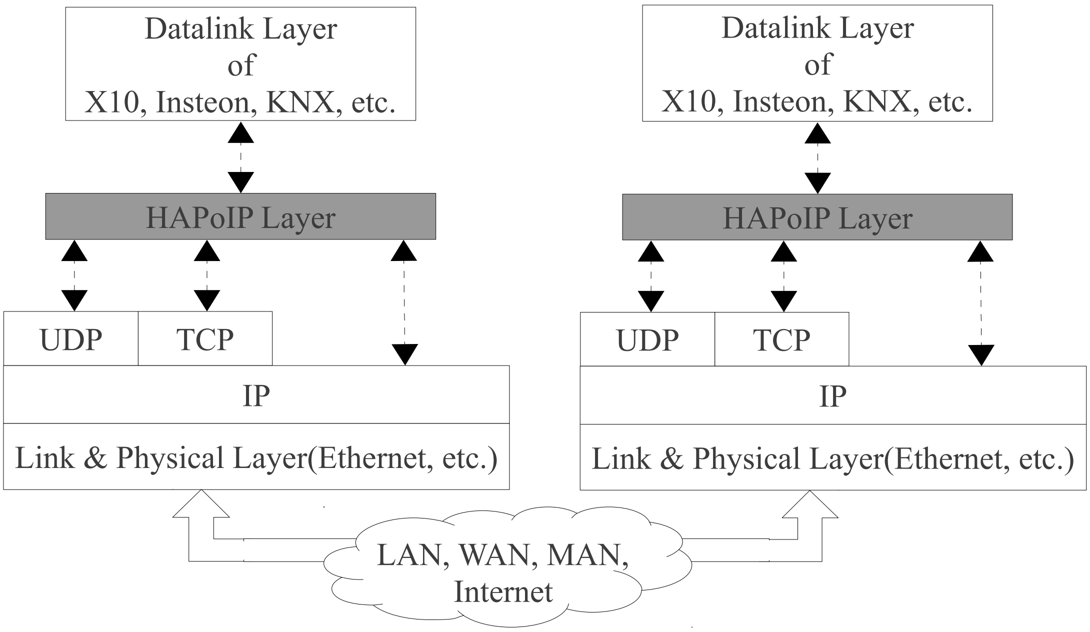

Datalink Layer of X10, Insteon, KNX, etc.
HAPoIP Layer
UDP
TCP
IP
Link & Physical Layer(Ethernet, etc.)
Datalink Layer of X10, Insteon, KNX, etc.
HAPoIP Layer
UDP
TCP
IP
Link & Physical Layer(Ethernet, etc.)
LAN, WAN, MAN, Internet

Figure 4-1: HAPoIP Adaptation Layer.

TCP has the advantages of reliable delivery service and hence will guarantee that the received packet order is preserved. On the other hand, it does not support multicast addressing and it is less bandwidth efficient than UDP (much more overhead). TCP also consumes much more resources of the HAPoIP node when implemented, instead of UDP.

What becomes apparent from all the above is that UDP is more efficient in carrying HAPoIP PDU and in addition, it supports multicasting for point to multipoint connections and IP facility protocols like DHCP and BOOTP, but because of the lack of a reliable delivery service, it will guarantee neither packet delivery nor packet order preservation.

4.3 HAPoIP Layer

The HAPoIP layer converges the already described in Chapter 3 power line home automation protocols to IP. In order to achieve this HAPoIP layer has to address the following issues that take place in an IP network:

- **Packet Loss:** Occurs when one or more packets of data travelling across an IP network fail to reach their destination because of a network node failure or channel congestion (packet drop), etc. In case that the HAP has a native mechanism to address message loss issues, it is not necessary for HAPoIP layer to implement a reliable delivery service. If for example HAP employs end-to-end acknowledgement it is not necessary for UDP to guarantee HAPoIP packets delivery
- **Packet Delay:** The time between a packet entering an ingress point of the IP network and its leaving in an egress point of the network [30]
- **Packet Misordering:** The delivery of data packets in a different order from which they were sent. Out of order delivery can be caused by packets following multiple paths through a network, or via parallel processing paths within network equipment that are not designed to ensure that packet ordering is preserved
- **Duplicate Packets:** Caused by IP equipment malfunctioning
- **Stale Packets:** Observed when ACK received after ACK waiting timeout. Thus, invalid packet retransmissions takes place

4.3.1 KNXnet/IP and LON/IP

In KNX and LON protocols the HAPoIP layer is already defined and standarised under the EN 13321-2 “KNXnet/IP Communication” and EN 14908-4 “IP Communication” standards respectively. Both KNXnet/IP and LON/IP use UDP and TCP support is optional. The IANA has assigned 1628 and 1629 ports in UDP and TCP for LON/IP usage and 3671 port in UDP and TCP for KNX/IP usage.

4.3.2 X10/IP, INSTEON/IP and UPB/IP

X10, INSTEON and UPB protocols will use UDP transport protocol for all the reasons mentioned in Section 4.3 . In addition, as IANA has not assigned a dedicated port number for each protocol, can be used 60000, 60001, 60002 private ports for X10, INSTEON and UPB protocol respectively[20].

[20] Please see also Section 2.3.2.

As Figure 4-2 shows HAPoIP layer has to encapsulate the data link layer of HAPs. Thus, the UDP payload (SDU) consists of a fixed header and the HAP's information.

7 bits	
Header	Payload

Figure 4-2: HAPoIP Packet.

In order for the encapsulation method to be as efficient as possible, only the HAP message (standard, acknowledgment, etc.) shall be captured (Figure 4-3). Thus, HAPoIP layer removes the static fields in HAP frame like start code and synch bits in transmission and adds it again in reception.

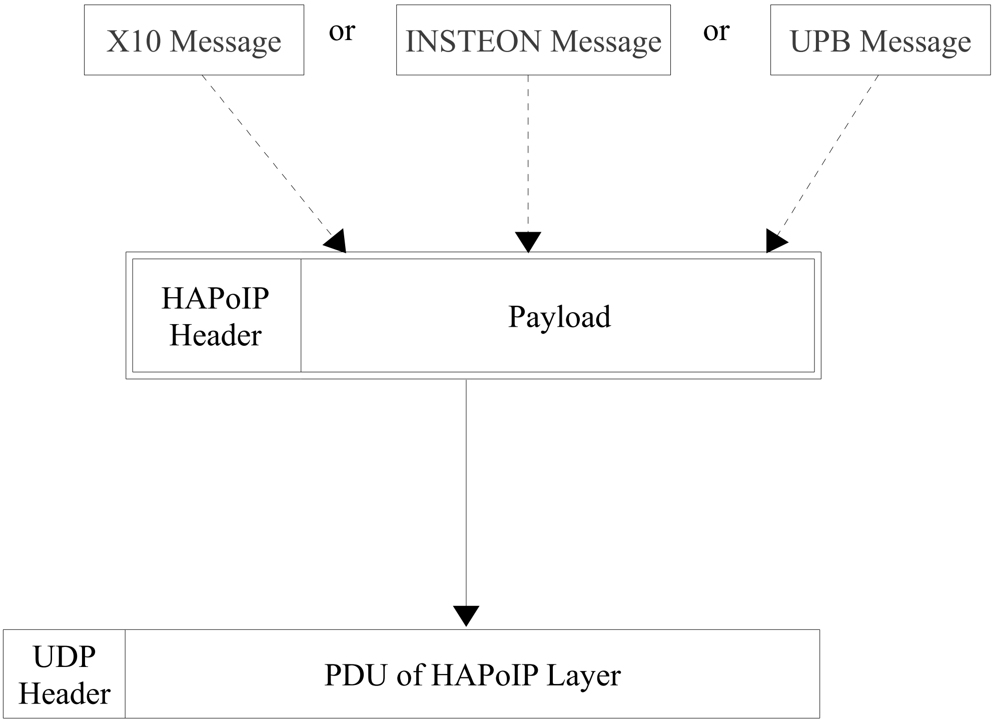


Figure 4-3: HAPoIP Encapsulation Method.

As X10 protocol has not any recovery mechanism, to restore lost or corrupted frames, it is necessary for a reliable IP communication between X10 nodes the HAPoIP layer to implement an Automatic Repeat reQuest (ARQ) service. The transmission rate of X10 protocol is very low[21] therefore, a simple Stop and Wait ARQ protocol will be quite

[21] A standard message needs 220ms to be transmitted over 50Hz power lines.

efficient for both IP transmission and platform implementation.

In stop and wait ARQ protocol the sender sends a single data packet and then waits for a positive ACK before it advances to the next data packet (Figure 4-4). The receiver only replies with an ACK if the data packet is correctly received. As either the data packet or the corresponding ACK may be lost/corrupted in transit, after the sender sends a data packet, it sets a retransmission timer to a predetermined value. If no ACK is received before the retransmission timer expires, the sender simply retransmits the data packet.

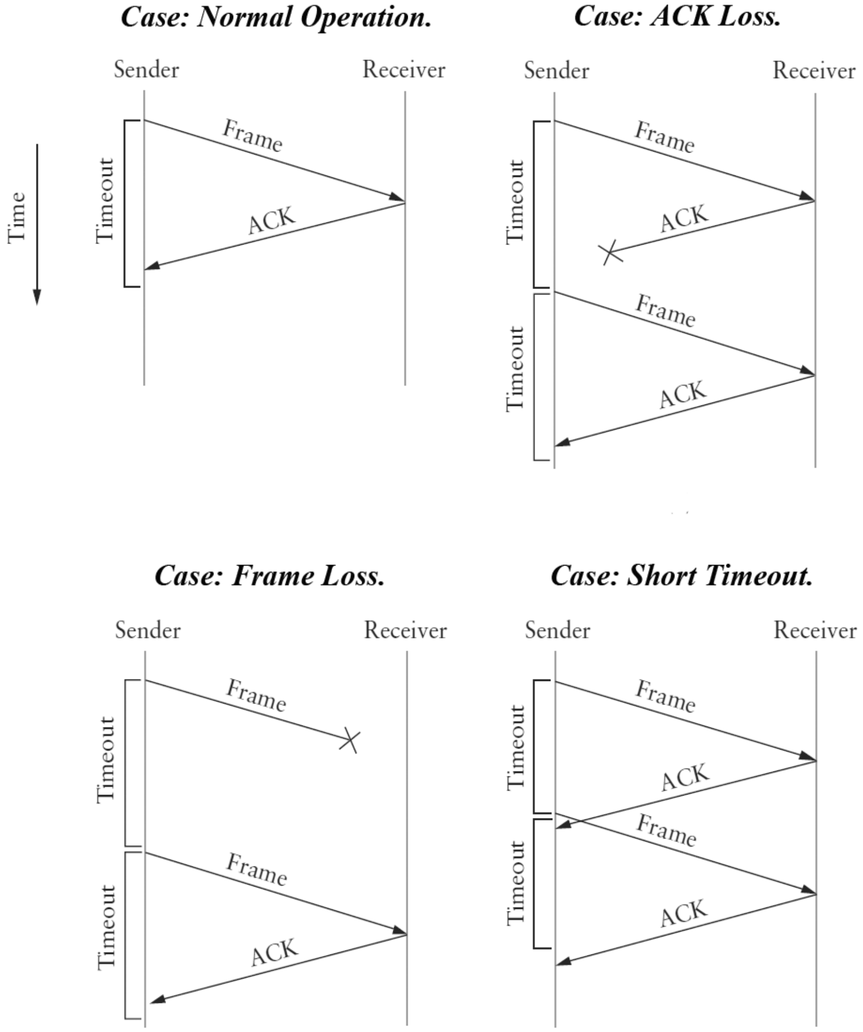


Figure 4-4: Stop and Wait ARQ Protocol Timeline for Four Different Scenarios [31].

For the receiver to distinguish between a data packet that is sent for the first time and a retransmission, a sequence number is included in the header of each data packet. For stop and wait ARQ, it is sufficient that the sequence number be 1-bit (i.e., either 0 or 1) because the only ambiguity is between a data packet and its immediate predecessor and successor, but not between the predecessor and successor themselves [31]. For similar reasons, each ACK should also contain a sequence number. In the common practice, the sequence number in the ACK is the sequence number of the next expected data packet rather than the sequence number of the data packet that has been recently received (Figure 4-5).

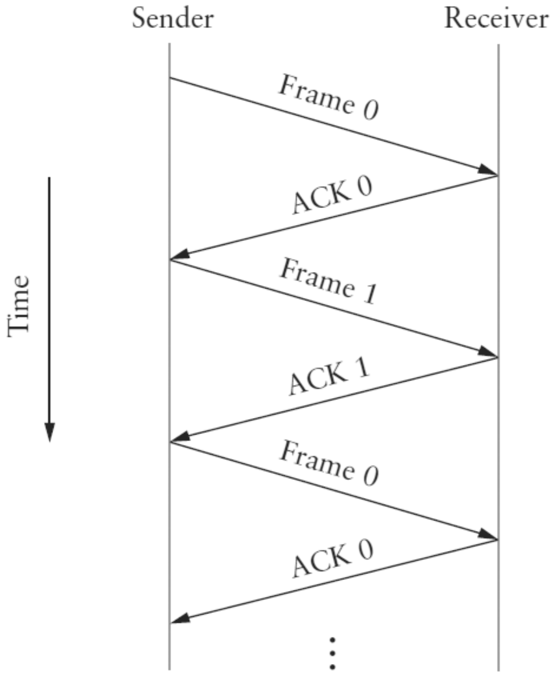


Figure 4-5: Timeline for Stop and Wait Protocol with 1-bit Sequence Number [31].

It should be mentioned that setting of the timeout interval at the sender is very important, and is a trade off between premature timeout and prolonged retransmission (stale packets). Stop and wait ARQ ensures that every data packet sent by the sender will eventually be received correctly by the receiver and that the receiver will get the data packets in order, i.e., it is an in order reliable unicast protocol.

INSTEON and UPB do not need any packet loss recovery service from HAPoIP layer as both protocols have a built-in end-to-end mechanism to ensure the correct delivery of their messages. But if the total round trip delay between message originator and message recipient exceeds the maximum ACK waiting time[22] the HAPoIP layer has to provide a proxy service in order to acknowledge the messages which pass through it. In this case the stop and wait ARQ protocol shall be implemented in HAPoIP layer towards IP network as already described above (Figure 4-6).

If the ARQ protocol is not adopted in HAPoIP layer the header of HAPoIP packet shall also have a sequence number in order to address the misorder, stale and duplicate packet issues.

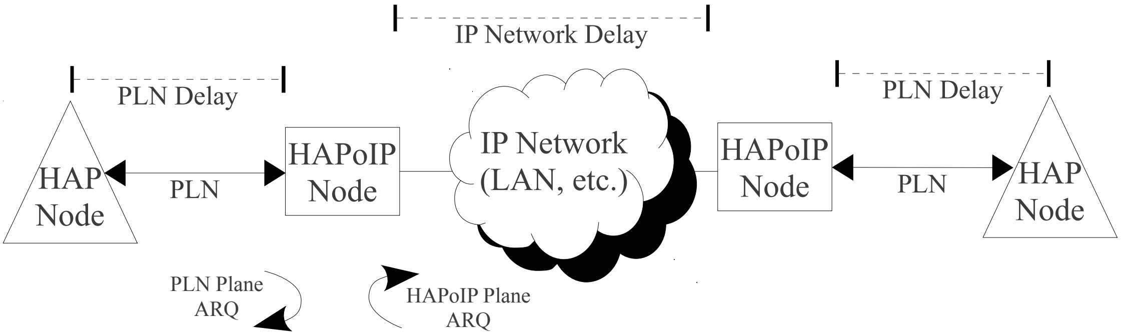


Figure 4-6: End-to-End Delay and ARQ Implementation.

Figure 4-7 shows the proposed packet structure and its header fields. The packet has a 7-bit fixed length header but its value can be changed if the version number increase. If Proxy service is used, both ACK and Seq. Num. fields accommodate the sequence numbers.

Protocol Version				ACK/ Seq.	Proxy	Seq. Num.	X10 or INSTEON or UPB Message
v	v	v	v	a	p	n	
Header							SDU

Figure 4-7: Packet Structure.

[22] Taking also into consideration the maximum allowed HAP's retransmission times.

Chapter 5 Hardware Design of the Test Platform

5.1 Introduction

This chapter presents the hardware design of a X10 to X10/IP converter. As Figure 5-1 shows, the converter consists of the following three major parts:

1. **Power Line Modem:** Responsible for X10 protocol's physical layer (frames transceiver)
2. **MCU:** The heart of the platform. It implements the data link layer of the X10 protocol and the HAPoIP layer. Also, MCU manages the UDP/IP module
3. **UDP/IP module:** Implements the UDP over IP stack with Fast Ethernet network interface layer

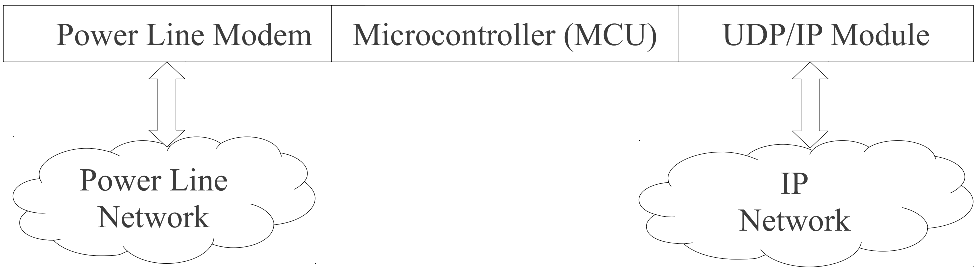


Figure 5-1: X10 to IP Adapter.

The effort is focus to build a simple, modular, flexible and reliable platform using, as much as possible, common market components instead of using proprietary hardware in order to reduce the cost of the whole system.

5.2 Power Line Modem

The power line modem consists of the following units that will be presented in the subsequent sections:

- Carrier band pass filter for X10 carrier capture/adaptation
- 120 kHz frequency decoder
- Mains zero crossing detector
- X10 transmission amplifier
- Surge protection

5.2.1 Carrier Band Pass Filter

Platform uses the Marmitek's FD10[23] 120 kHz band pass filter in order to couple and decouple the X10 carrier signal from/to power line network.

FD10 is also used as waveform shaper for the output signal of the X10 transmission amplifier. It removes the harmonics of the 120 kHz square wave signal, resulting in a sinusoidal 120 kHz signal.

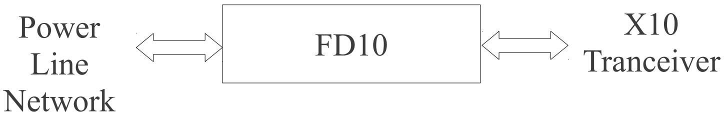


Figure 5-2: Mains Coupler.

5.2.2 120 kHz Frequency Decoder

The MCU needs the incoming frame voltage to be in TTL levels. This translation of the 120 kHz signal to TTL logic is obtained by demodulating the incoming 120 kHz sine wave. An appropriate, simple and low cost solution was found in LM567 frequency decoder integrated circuit. APPENDIX A shows the datasheets of the chip.

This chip consists of a phase lock loop and a voltage controlled oscillator, which can be adjusted through external components. The resulting circuit taken from the datasheets is shown in Figure 5-3.

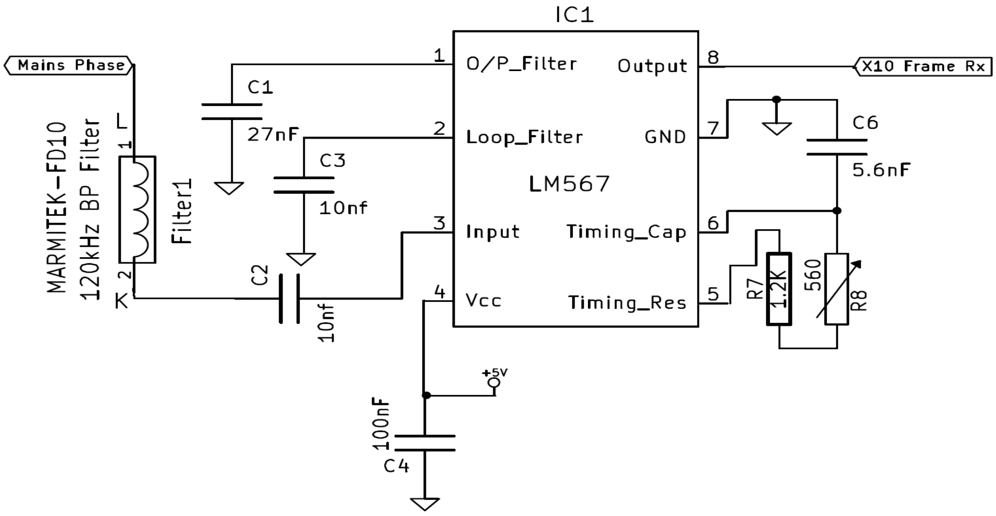


Figure 5-3: LM567 Circuit.

[23] http://www.marmitek.com/en/product-details/home-automation-security/x-10-home-automation/din-rail-modules/fd10.php

The center frequency of the decoder is given by the formula:

$$f_0 \simeq \frac{1}{1.1(R7+R8)*C6} \qquad (5.1)$$

where R7 and R8 values is in Ohms and C6 value in Farad.

Also, the bandwidth (BW) of the filter may be found from the Figure 5-10 and especially for input signal $V_{in} \leqslant 200\text{mV}_{rms}$ we can use the following approximation:

$$BW = 1070 * \sqrt{\frac{V_{in}}{f_0 * C3}} \quad \text{in \% of} \quad f_0 \qquad (5.2)$$

where V_{in} value in Volts RMS and C3 value in μF .

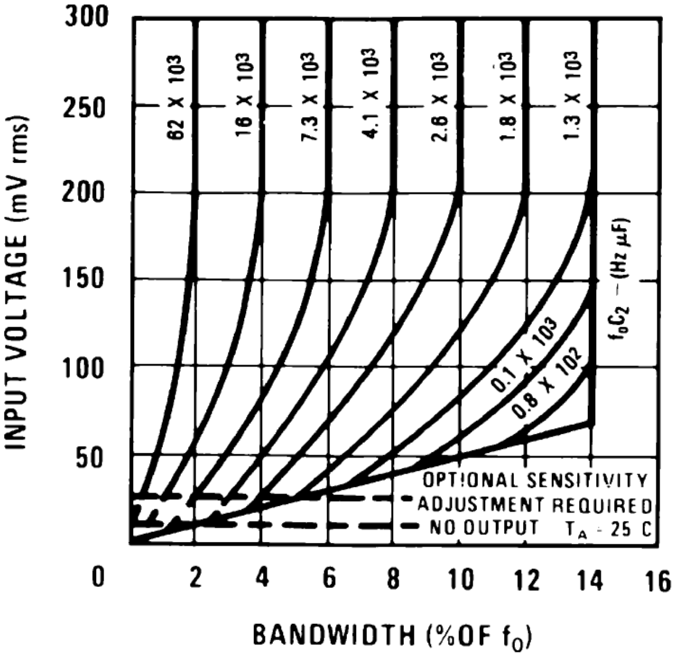


Figure 5-4: Bandwidth vs Input Signal Amplitude (APPENDIX A).

Free selecting C6 = 5.6 nF[24] we found R7 + R8 = 1.352 K[25] using the equation (5.1) for $f_0 = 120\,kHZ.$

The Marmitek's X10 devices, which have been used in the lab, have a range of input voltage from 0.0177 to 0.884 V_{rms} and a maximum frequency drift of ± 2 kHz. Thus, as according to equation (5.2) the BW of the LM567 frequency detector is minimum at low voltage input threshold, we have to guaranty a minimum 4 kHz BW for 17.7 mV_{rms}

[24] We use E12 and E24 series for capacitor and resistor values respectively.

[25] R8 is a variable component for fine tuning.

input. Taking into account all the above, we decide the value of 10nF for the C2 capacitor, which gives us a BW of 16.6 kHz with 0.2 V_{rms} or more and 4.9 kHz with 0.0177 V_{rms} input voltage.

For design uniformity reasons, a 5 V power supply voltage have chosen. Thus, as the datasheets mention, tone decoder will accept inputs between -10 V and 5.5 V without any problem.

LM567 is designed to provide a saturated transistor switch to ground when an input signal is present within the passband but no inversion of the signal is necessary since the MCU is engaged both for logic inversion and bias of the detector output with input pull-up mechanism[26] (Figure 5-5).

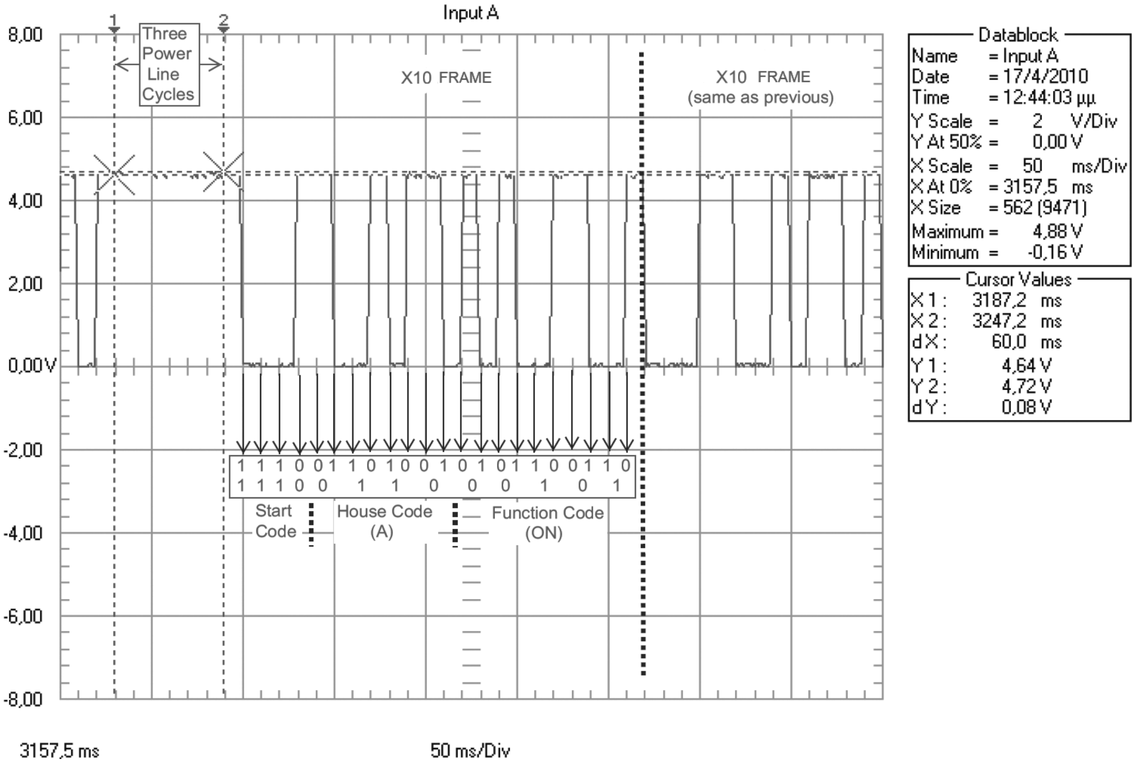


Figure 5-5: LM567 Output.

5.2.3 Mains Zero Crossing Detector

As Section 3.4.2 mentions, the X10 receiver detects the presence of 120 kHz carrier just after each zero crossing point of the power line voltage. Thus, MCU needs to know when zero crossing point of the AC power line takes place in order to sample the output

[26] MCU's port inputs can source current if the internal pull-up resistors are activated.

of the LM567.

The target is to design a circuit, which produces a square wave like output synchronisation signal, with an amplitude of approximately 5 V and frequency of 100 Hz, in phase with the power line voltage.

As Figure 5-6 shows, we use a step down transformer with cooperation of a full wave rectifier and a zener limiter to achieve the zero crossing synchronisation line.

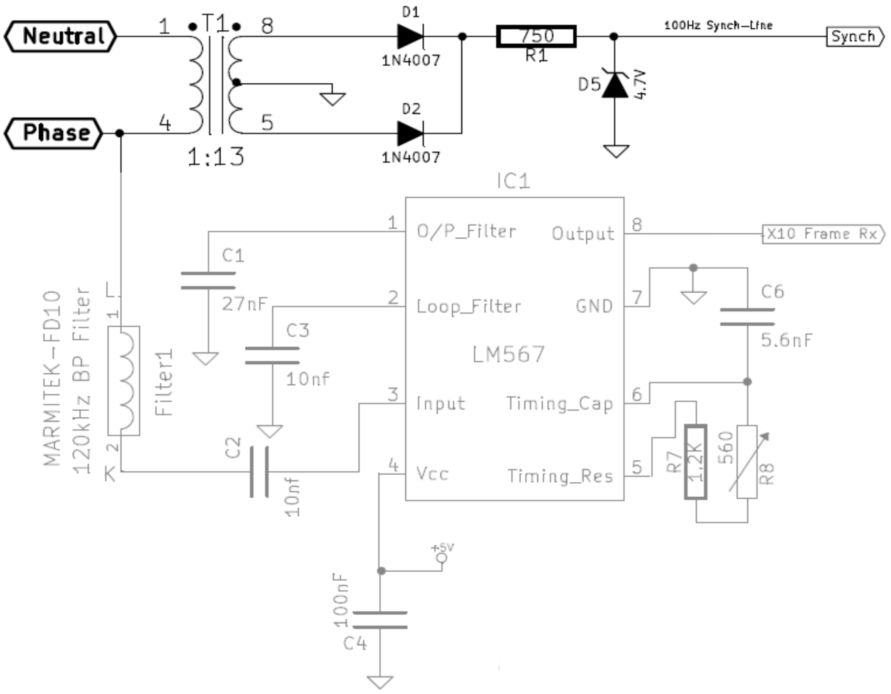


Figure 5-6: Mains Zero Crossing Detector.

The output of the transformer changes linearly, following its input. The output voltage of the transformer is given from the following equation:

$$\frac{V_{8,5}}{V_{1,4}} = \frac{n_{sec}}{n_{pri}} \qquad (5.3)$$

where:

- $V_{8,\,5}$ is the output voltage of the transformer
- $V_{1,\,4}$ is the input voltage of the transformer
- n_{pri} is the number of turns of primary winding

- n_{sec} is the number of turns of secondary winding

Free selecting $\frac{n_{sec}}{n_{pri}} = 13$ we have from equation (5.3):

$$V_{8,5} = \frac{230\text{V}_{rms} * \sqrt{2}}{13} = 25\text{V}_{peak} \qquad (5.4)$$

Because of the grounded center tap, each half of the secondary winding has a sinusoidal voltage with peak $\frac{V_{8,5}}{2} = 12.5\text{V}_{peak}$ and opposite phases.

The full wave rectifier D1 - D2 gives a half wave signal with frequency 100 Hz and V_{peak} = 11.8 V because of diode drop voltage (≈ 0.7 V).

Therefore, the voltage limiter circuit R1 - D5 has to handle a peak input voltage of 11.8 V. As the power supply voltage of the MCU is 5V we decide to use a 4.7 V zener in order for the output voltage of the limiter to be just below of the 5 V. Thus, the R1 value can be estimated from the following equation:

$$R1 = \frac{V_{rec\_out} - V_{zener}}{I_{R1}} \qquad (5.5)$$

where:

- V_{rec_out} is the output voltage of the full wave rectifier
- V_{zener} is the breakdown voltage of the zener diode
- I_{R1} is the maximum current through the R1 resistor, which is the sum of the maximum current through the zener (I_{zener}) and the maximum MCU's port input current (I_L)

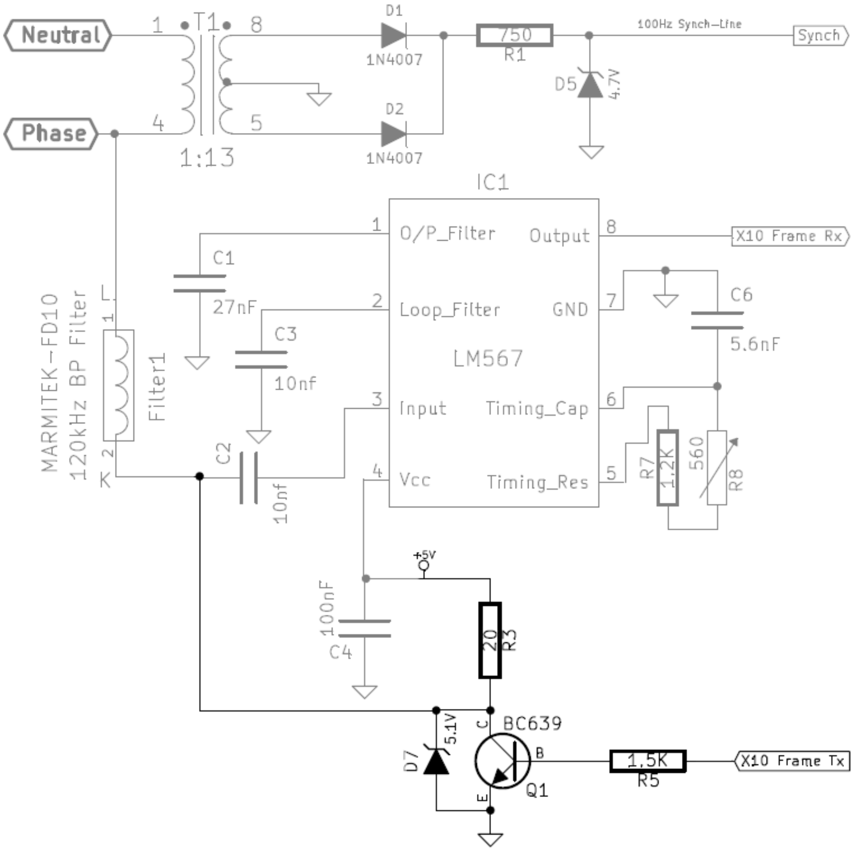
Free selecting I_{zener} = 10 mA and I_L << 1 mA, according to MCU's datasheets[27], we have from equation (5.5):

$$R1 = \frac{11.8\text{V} - 4.7\text{V}}{0.01\text{A}} = 710\Omega \rightarrow 750\Omega\,(according\text{ to }E24\ series).$$

[27] www.atmel.com/dyn/resources/prod_documents/doc2502.pdf

5.2.4 X10 Transmission Amplifier and Surge Protection.

According to datasheets, MCU's output ports current is limited to 40 mA, which is too low for X10 signal transmission to PLN. Thus, a transistor switch circuit is used as X10 transmission signal booster (Figure 5-7).



A transistor switch operates at either saturation or cutoff state. The value of resistor R3 is given from the following equation:

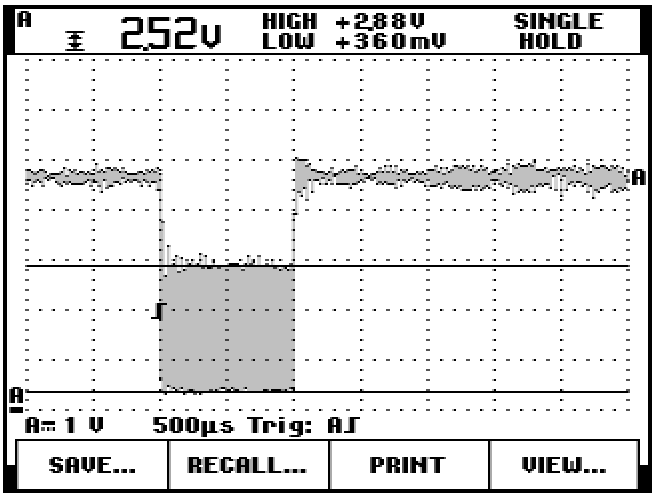
$$\frac{V_L}{V_{CC}} = \frac{R_L}{R_L + R3} \qquad (5.6)$$

where:

- R_L is the load impedance (FD10 filter)
- V_L is the peak voltage across the load
- V_{CC} is the power supply voltage

As the load for the transistor switch is the FD10 filter, which has minimum input impedance of 20 Ω at 120 kHz and the maximum load voltage must be 2.5 $V_{p\text{-}p}$[28], the value of resistor R3 is: $R3 = \frac{R_L * (V_{CC} - V_L)}{V_L} = 20\Omega$ (5.7)

Figure 5-8 shows a captured bit-one waveform in collector of BC639.



The base current needed for transistor saturation is given from the following equation:

$$I_{b(sat)} = \frac{I_{C(sat)}}{h_{FE}} \qquad (5.8)$$

where:

- h_{FE} is the dc current gain
- $I_{C(sat)}$ is the collector current during switch on

$I_{C(sat)}$ is given from the following equation:

$$I_{C(sat)} = \frac{V_{CC} - V_{CE(sat)}}{R3} \qquad (5.9)$$

Therefore, from equations (5.7) and (5.8) we have:

$$I_{b(sat)} = \frac{V_{CC} - V_{CE(sat)}}{h_{FE} * R3} \qquad (5.10)$$

[28] Please see Section 3.3.

To be sure of saturation, it is usual practice to increase this calculated value by 30%. Thus, the base current value for transistor saturation is estimated from the following equation:

$$I_{b(sat)} = 1.3 * \frac{V_{CC} - V_{CE(sat)}}{h_{FE} * R3} \qquad (5.11)$$

R5 is given from the following equation:

$$R5 = \frac{V_{X10\_Tx} - V_{BE(ON)}}{I_{b(sat)}} = \frac{V_{X10\_Tx} - V_{BE(ON)}}{1.3 * \frac{V_{CC} - V_{CE(sat)}}{h_{FE} * R3}} \qquad (5.12)$$

where:

- V_{X10_Tx} is the high voltage level of microcontroller output
- $V_{BE(ON)}$ is the base-emitter voltage during switch on

Therefore, R5 = 1470 Ω and rounded to 1.5 kΩ according to E24 series.

The D7 zener diode limits the incoming signal amplitude to a shape appropriate for the transceiver region (from -0.7 V to +5.1 V).

5.3 MCU

The microcontroller is the "heart" of the system, and the choice of it is very important for test platform design. The platform implementation has the following requirements:

1. Two 8bit timers/counters with at least one output compare pin for X10 frame transmission
2. One external interrupt port for X10 transceiver synchronisation with zero crossing of the mains
3. Two Input/Output ports for X10 bits capture input and UDP/IP module reset
4. One Serial Peripheral Interface (SPI) port for the UDP/IP module communication
5. 8K Bytes program memory and 256 Bytes of data memory
6. At least 12 million instructions per second (MIPS) computing performance

A general purpose low cost and high speed MCU the ATMEGA8535 from ATMEL is selected, which is a low power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture [32].

It executes almost all instructions in a single clock cycle and achiever throughput approaching 1 MIPS per MHz. It is designed according to Harvard architecture, which refers to the fact that the CPU unit co-works at the same time with a program memory and also a separate data memory.

The technical characteristics are listed below:

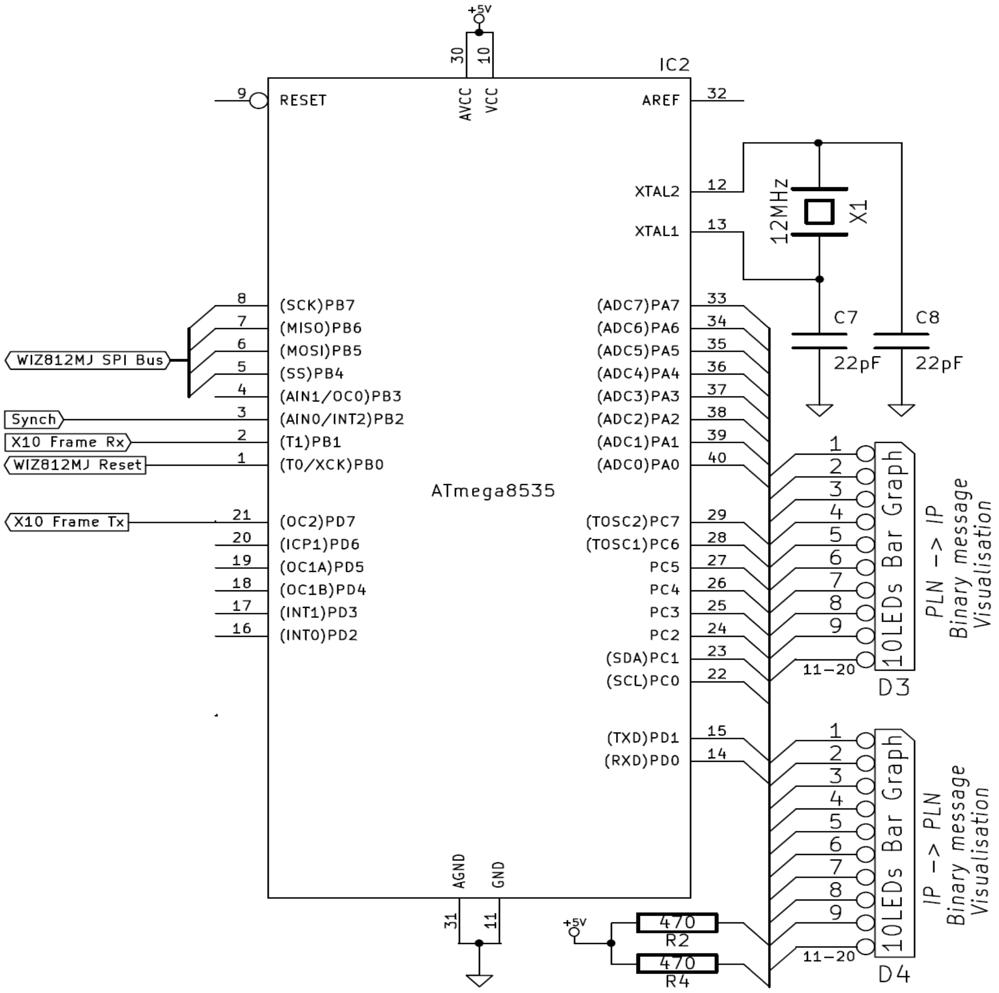
- High Performance and Low Power RISC Architecture:
    - 130 Powerful instructions – most single clock cycle execution
    - 32 x 8 General purpose working registers
    - Up to 16 MIPS throughput at 16 MHz
    - On-chip 2-cycle multiplier
    - 8K Bytes of in-system self-programmable flash
    - 512 Bytes EEPROM
    - 512 Bytes internal SRAM
    - In-system programming by on-chip boot program
    - Programming lock for software security
- Peripheral Features:
    - 32 Programmable Input/Output lines
    - Two 8-bit Timer/Counters with separate prescalers and compare modes
    - One 16-bit Timer/Counter with separate prescaler, compare mode, and capture mode
    - Real time counter with separate oscillator
    - Four PWM channels

  - 8-channel, 10-bit ADC
  - Byte oriented two wire serial interface
  - Programmable serial USART
  - Master/Slave SPI serial interface
  - Programmable Watchdog Timer with separate on-chip oscillator
  - On-chip analog comparator
- Special Microcontroller Features:
  - Power on reset and programmable brown-out detection
  - Internal calibrated RC oscillator
  - External and internal interrupt sources
  - Six sleep modes: idle, ADC noise reduction, power-save, Power-down, Standby and Extended Standby
- Operating Voltages and Speed:
  - V_{CC}: 4.5 – 5.5 V
  - 0 – 16 MHz

The reasons behind the choice of this particular microcontroller are the requirements of the platform mentioned above, the use of two additional 9-bit output parallel ports for X10 binary messages visualization (for debugging) (Figure 5-9), the chip availability in the lab and the low cost of implementation.

It must be mentioned that the initial platform design, which has served only the X10 standard message format, was developed with MCU's internal 1 MHz RC oscillator calibrated for 960 kHz clock frequency but as the 62-bit extended messages frame support has been added in the final version of the platform the need of an external 12 MHz oscillator was made mandatory to achieve the maximum X10 frame rate.

Figure 5-9 shows the final hardware design of the microcontroller module.



5.4 UDP/IP Module

As Ethernet is the most widely-installed LAN technology, it was decided as network interface protocol in our TCP/IP stack implementation.

Figure 5-10 depicts that every layer of TCP/IP stack can be implemented either in MCU by software (lwIP, uIP, etc.) or in a peripheral chip by hardware.

Running the whole TCP/IP stack in a peripheral chip which was designed to do this dedicated work, our system advantages summarised in Table 5-1.

In addition, it must be mentioned that a software TCP/IP stack platform that has been equipped with a public IP address can easily go out of order because of denial of service attacks.

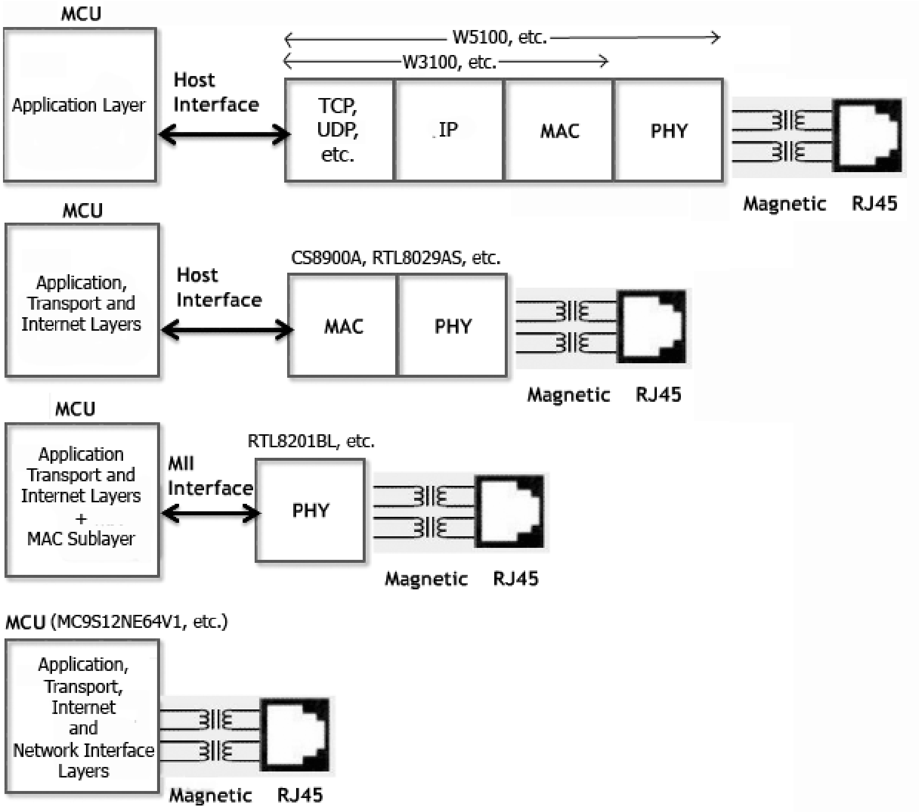


Figure 5-10: Embedded Ethernet Solutions.

ITEM	Software TCP/IP Stack	Hardware TCP/IP Stack
MCU Resource for TCP/IP	More than 40%	Under 10%
Stability	MCU Overload for TCP/IP (Less resource for applications)	Enough MCU Resources (whole system stability guaranteed)
Network Performance	Max 35Mbps	Max. 80Mbps
Tech Support	Just provide the code in Library format	Provide all driver and application reference codes
Time Resource	Need time for: • TCP/IP porting • Application Development • Debugging	Need time for: • Simple Socket Programming • Application Development
System Upgrade (MCU Change)	TCP/IP should be optimized according to MCU Type (Re-porting and debugging is required)	TCP/IP is independent from MCU

Table 5-1: Hardware vs Software TCP/IP Stack.

Hardware TCP/IP stack approach is more expensive by a few pounds but its benefits make it worth the cost.

WIZnet's W5100 chip has been selected, which is a full featured, single chip Internet enabled 10/100 Ethernet controller designed for embedded applications where ease of integration, stability, performance, area and system cost control are required.

The W5100 includes fully hardwired, market-proven TCP/IP stack and integrated Ethernet network interface layer (Figure 5-11). No need to consider the handling of the Ethernet Controller, but a simple socket programming is required.

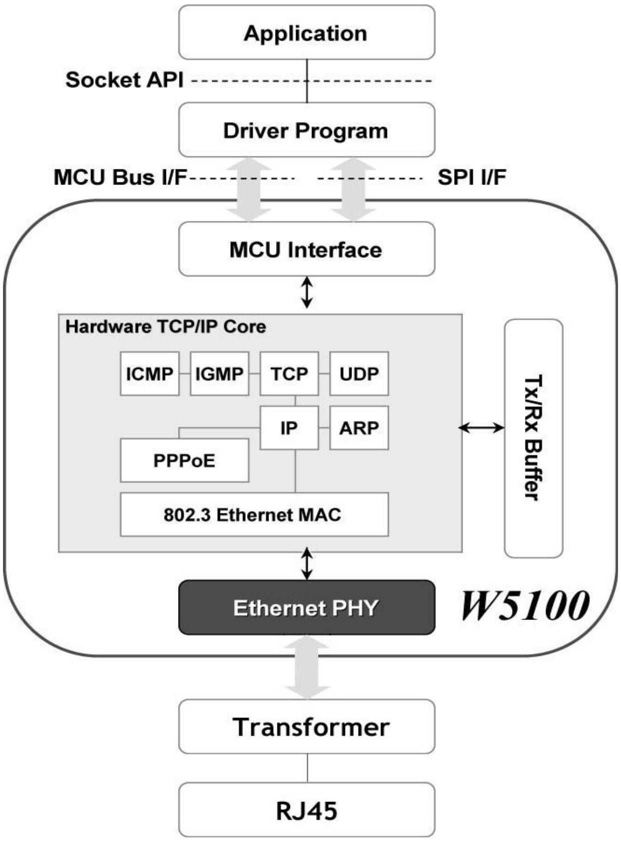


Figure 5-11: W5100 Block Diagram.

The main features of the W5100 are listed below:

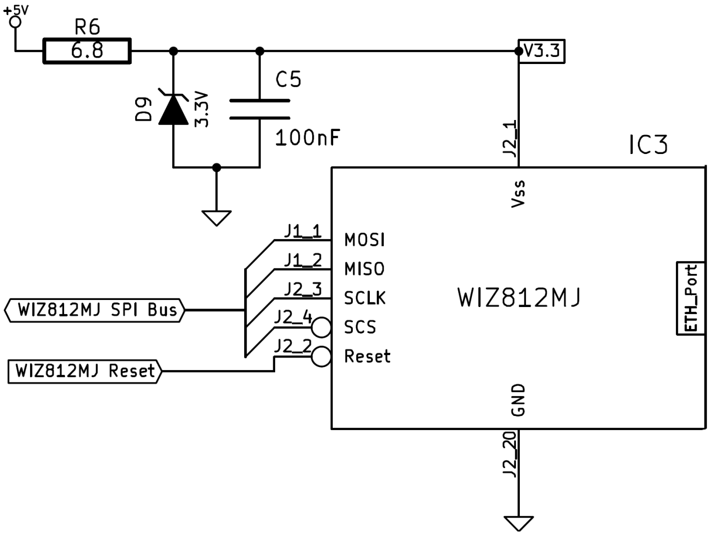
- Includes the Hardware Internet protocols: TCP, UDP, ICMP, IPv4 ARP, IGMP, PPPoE, Ethernet
- 10BaseT/100BaseTX Ethernet PHY embedded
- Auto Negotiation support (Full-duplex and half duplex)
- Auto MDI/MDIX support

- ADSL connection support (with support PPPoE Protocol with PAP/CHAP Authentication mode)
- Supports four independent sockets simultaneously
- Not support IP Fragmentation
- Internal 16Kbytes Memory for Tx/Rx Buffers
- 0.18 µm CMOS technology
- 3.3V operation with 5V I/O signal tolerance
- SPI interface support (SPI MODE 0, 3)

Regarding the hardware design convenience WIZ812MJ have been decided, which is a network module that includes W5100 chip, MAG-JACK (RJ45 with internal transformer) and all necessary devices for operation such as logic ICs, transistors, etc. APPENDIX B shows the schematic diagram of the WIZ812MJ module.

WIZ812MJ can be used as a component and no effort is required to interface W5100 and Transformer. Also, as the data throughput is low, MCU can access it through SPI interface, which reduces the circuit footprint.

WIZ812MJ specifications show that it operates in 3.3V with maximum consumption of 200 mA[29]. Thus, a zener regulation circuit has been implemented in order to step-down the power supply voltage (Figure 5-12).



[29] http://www.wiznet.co.kr/Sub_Modules/en/technical/FAQ.asp.

The value of R6 is given from the following equation:

$$R6=\frac{V_{CC}-V_{zener}}{I_{L(max)}}=6.8\Omega \qquad (5.13)$$

where:

- V_{CC} is the power supply voltage
- Vzener is the zener voltage
- IL(max) is the WIZ812MJ 's maximun consumption current

APPENDIX C shows the schematic diagram of the whole test platform.

Chapter 6 Software Design of the Test Platform

6.1 Introduction

This chapter describes the software design. The software was made in order ATmega8535 to receive/transmit X10 frames, process X10 messages and control the WIZ812MJ module.

In order to get software portability, C programming language was used for this program. Thus, the program with minor changes can be used for any microcontroller that meet the requirements of Section 5.3.

Programming and Debugging were accomplished through the use of GNU AVR GCC compiler and 'AVR Studio' software.

The programming of the microcontroller and the hardware tests were done by using Atmel's evaluation board STK500 [33]. The basic functions of the program are the following:

I. Platform initialisation

II. Receiving X10 Frames from the PLN

III. Transmitting X10 Frames to the PLN

IV. Receiving X10/IP messages from the IP network

V. Transmitting X10/IP messages to the IP network

Figure 6-1 shows a general flow chart of microcontroller processes.

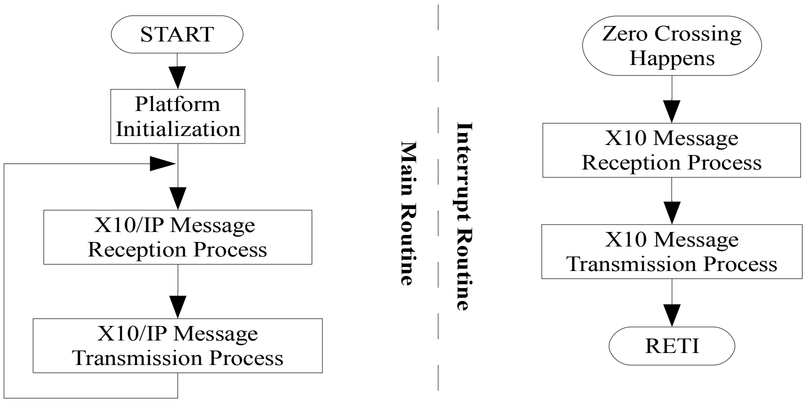

START
Platform Initialization
X10/IP Message Reception Process
X10/IP Message Transmission Process
Main Routine
Interrupt Routine
Zero Crossing Happens
X10 Message Reception Process
X10 Message Transmission Process
RETI

Figure 6-1: Basic Microcontroller Operations.

WIZ5100 default driver source code is based on ATmega128. Thus a lot of changes have been made in order to port it for Atmega8535 with minimum memory requirements.

6.2 Platform Initialisation

At the beginning the program defines appropriate variables for frames, messages and intermediate data storage/buffering and initialise both peripherals inside microcontroller and W5100 hardwired TCP/IP stack.

The internal peripherals that have been used are the following:

- *Parallel I/O ports A, C* and *0 & 1 data lines of port D* as outputs for X10 transmitted/received message indication (Figure 5-9)
- *Data line 1 of port B* as input for X10 frame reception
- *Timers 1 and 2* in Clear Timer on Compare match mode (CTC) for bit “1” signal duration definition (1 ms) and 120 kHz frequency square wave signal generation respectively
- *SPI and data line 0 of port B* for controlling and data exchange with W5100 chip

In addition, external interrupt 2 is enabled for X10 transceiver synchronization with mains zero crossing.

When the microcontroller initialisation has finished the microcontroller provides a reset signal to W5100 and afterwards all the necessary information for IP network connection (Section 4.1). Finally, microcontroller set up the socket 0 of the W5100 for UDP protocol and listening in 60000 port (Section 4.3.2).

6.3 X10/IP Message Reception Process

As Figure 6-1 depicts, after platform initialisation, the program enters in a continuous loop. During the loop microcontroller executes the X10/IP message reception procedure and afterwards the X10/IP message transmission procedure and so forth until an interrupt takes place (voltage zero crossing in PLN).

Figure 6-2 shows the flow chart of the X10/IP message reception procedure.

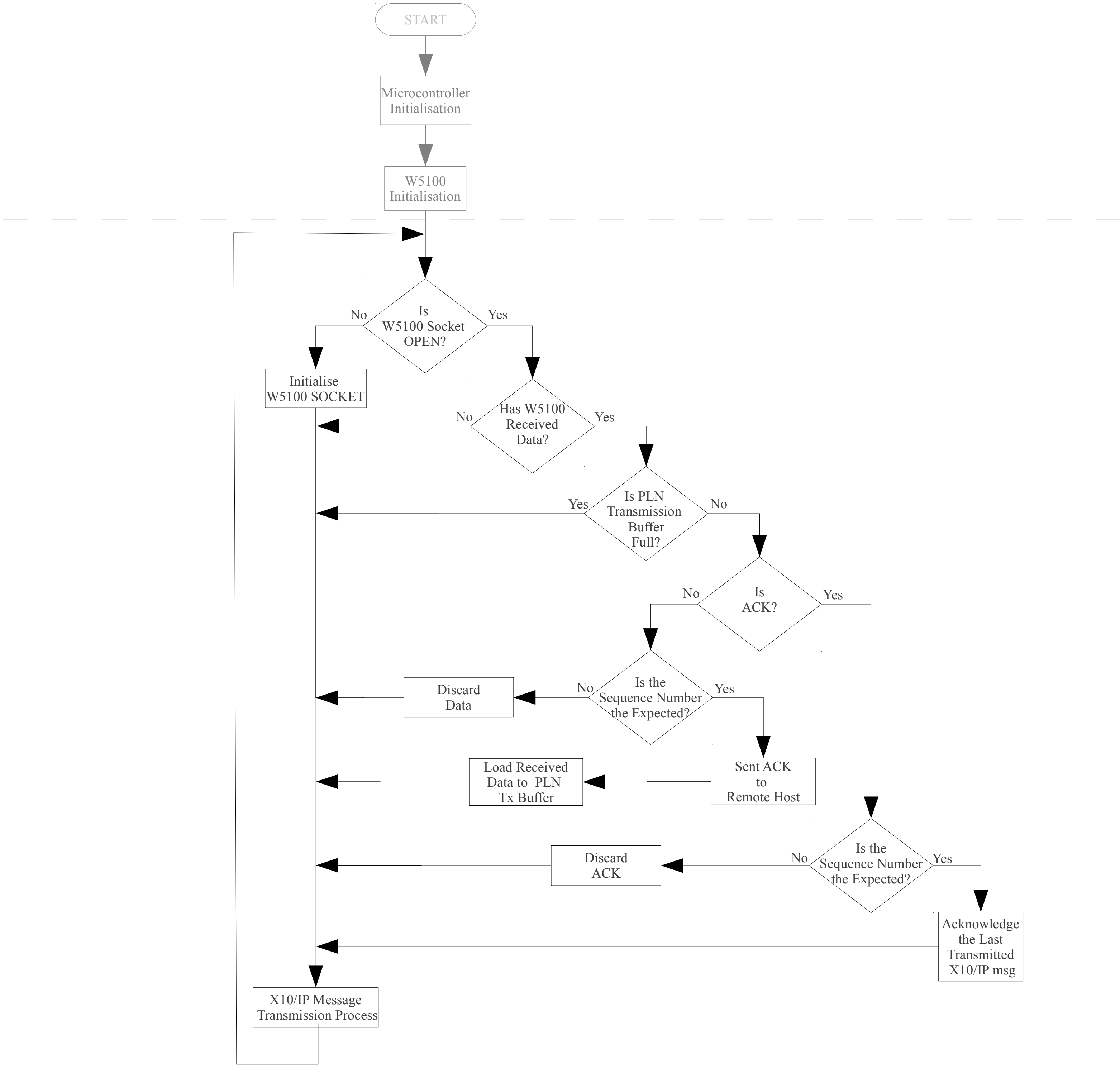

START
Microcontroller Initialisation
W5100 Initialisation
Is W5100 Socket OPEN?
No
Yes
Initialise W5100 SOCKET
Has W5100 Received Data?
No
Yes
Is PLN Transmission Buffer Full?
Yes
No
Is ACK?
No
Yes
Is the Sequence Number the Expected?
No
Yes
Discard Data
Sent ACK to Remote Host
Load Received Data to PLN Tx Buffer
Is the Sequence Number the Expected?
No
Yes
Discard ACK
Acknowledge the Last Transmitted X10/IP msg
X10/IP Message Transmission Process

Figure 6-2: X10/IP Message Reception Procedure.

Firstly, routine checks if the UDP/IP stack is listening in 60000 port and if for an unexpected reason the socket is closed MCU opens it again.

If the socket is open, there is received data from remote host and there is space to store the data, the routine clarifies if data is a data reception acknowledgement or new X10/IP message from remote host and if the data is an acknowledgement, the routine informs the X10/IP transmission process for the reception of the last transmitted data from the remote host. Otherwise, if the data is X10/IP message, routine sends acknowledgement to remote host and forwards data for transmission to PLN.

To avoid duplicate data to proceed as new data, always routine checks the validity of the sequence number.

6.4 X10/IP Message Transmission Process

Figure 6-3 shows the flow chart of the X10/IP message transmission procedure.

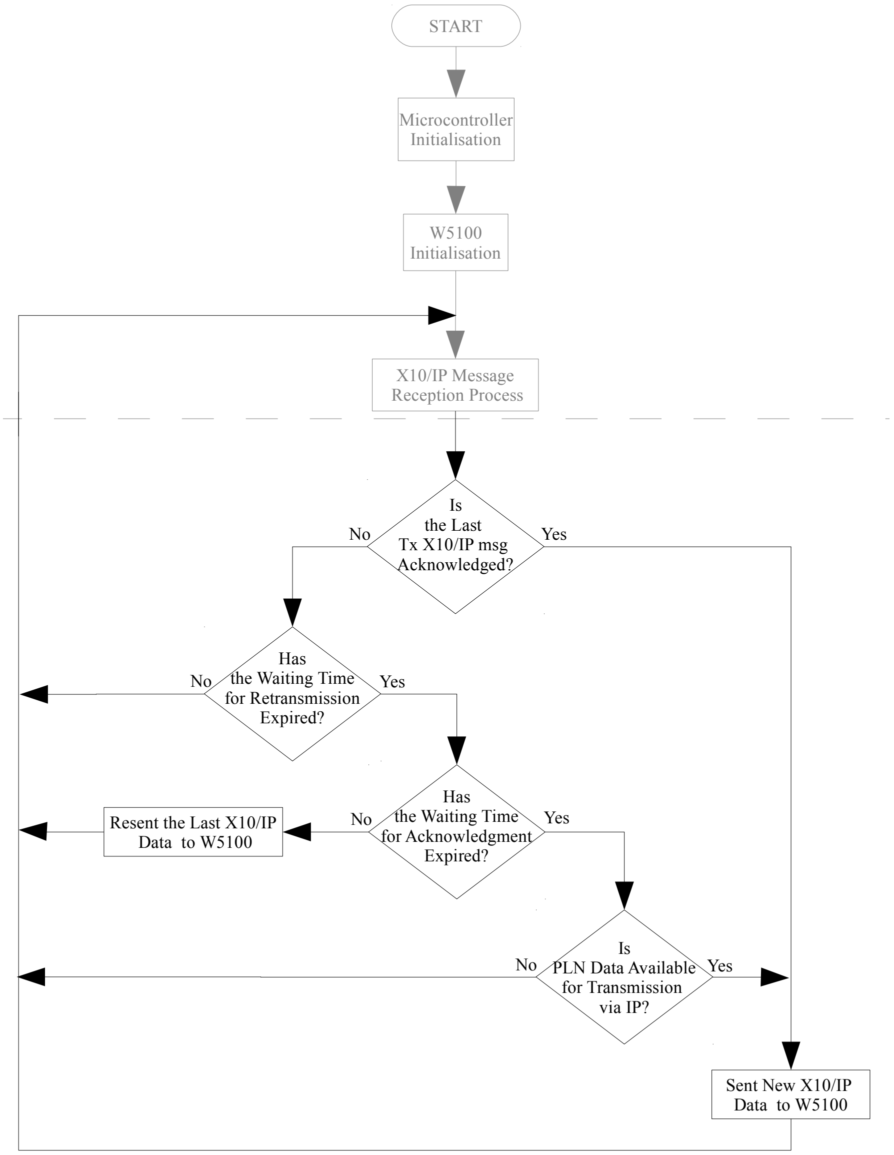

START
Microcontroller Initialisation
W5100 Initialisation
X10/IP Message Reception Process
Is the Last Tx X10/IP msg Acknowledged?
No
Yes
Has the Waiting Time for Retransmission Expired?
No
Yes
Has the Waiting Time for Acknowledgment Expired?
No
Yes
Resent the Last X10/IP Data to W5100
Is PLN Data Available for Transmission via IP?
No
Yes
Sent New X10/IP Data to W5100

Figure 6-3: X10/IP Message Transmission Procedure.

Before routine proceeds to new data transmission, it waits for acknowledgement from remote host for the last sent data. If a determined time passes without reception of acknowledgment the platform retransmits the last transmitted data. But retransmissions end when acknowledgment timeout occurs.

In order to determine both the retransmission times and acknowledgment timeout, data have been collected for Internet network round trip delay and packet loss from Verizon world-wide network statistics[30] and Internet Traffic Report site[31]. In addition, we have collected data from our location with the help of www.pingtest.net online tool (Table 6-1).

From the collected data, we have estimated that the host-to-host minimum and mean maximum round trip delay are 50 ms and 500 ms respectively. In addition the packet loss ratio is zero or near to zero.

Source IP	Destination Server Location	Round Trip Delay(ms)	Jitter(ms)	Packet Loss
134.83.1.242 (Brunel University Network)	London (UK)	49	0	0%
	Oslo (Norway)	106	1	0%
	Rome (Italy)	124	0	0%
	Heraklion (Greece)	152	1	0%
	Nuuk (Greenland)	168	1	0%
	Miami, Florida (USA)	188	0	0%
	Los Angeles, CA (USA)	206	1	0%
	Doha (Katar)	212	1	0%
	La Paz (Bolivia)	259	0	0%
	Nairobi (Kenya)	266	1	0%
	Villa Gobernador Galvez (Argentina)	310	2	0%
	Cape Town (South Africa)	317	1	0%
	Kuala Lumpur (Malaysia)	337	1	2%
	Sydney (Australia)	390	1	0%
	Canberra (Australia)	425	98	0%
	Seoul (South Korea)	524	30	0%

Table 6-1: Worldwide Ping Measurements.

[30] http://www.verizonbusiness.com/about/network/latency
[31] www.internettrafficreport.com

As Section 3.4.3 mentions, the extended message frame has the longest transmission time in PLN as needs 31 mains cycles (620 ms) for it. Thus as Figure 6-4 shows, in our implementation the mean maximum delay between action (push a button) in local node and reaction (light getting on) in remote node is equal to 620 ms + 250 ms + 620 ms = 1490 ms, under zero packet loss condition (no ARQ retransmissions).

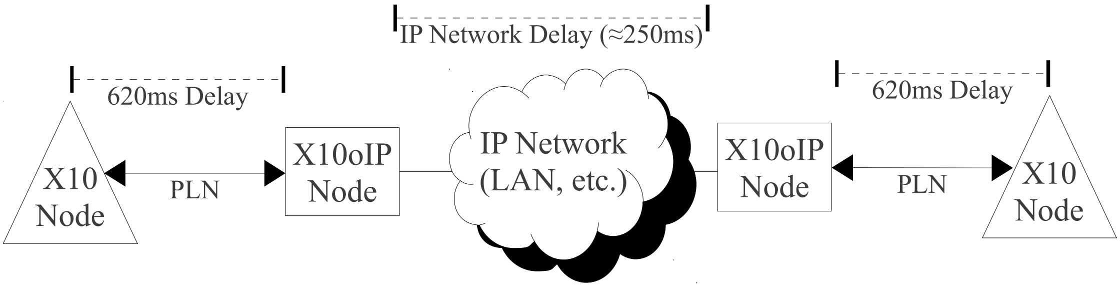


Figure 6-4: End-to-End Delay.

As less than two second delay between action (push a button) and reaction (light getting on) in building automation systems is affordable, we define acknowledgment timeout equal to 750 ms and retransmission time equal to 50 ms[32]. Therefore, we have the opportunity for at least one retransmission in the IP networks with big delay.

6.5 X10 Message Reception Process

Every time the mains voltage goes through zero the external interrupt routine (INT2) is enabled, which includes the PLN message reception and transmission procedures (Figure 6-1).

As Figure 6-5 depicts, in every mains zero crossing the MCU captures and buffers one possible bit of X10 frame. Afterwards if routine detects the start code of X10 frame in the buffer, it checks for the presence of extended frame. In case that the received frame is an extended frame, a flag is set on in order for the routine to indicate how many bytes are for transmission through UDP/IP stack.

In order for the program to avoid corrupted messages to reach the remote node, the routine checks the true and complement form of the X10 frame for errors.

Finally, routine adds the protocol and sequence number as Section 4.3.2 describes and the X10/IP data are stored in a FIFO waiting for the main routine to send it to W5100.

[32] Minimum measured round trip delay in Internet.

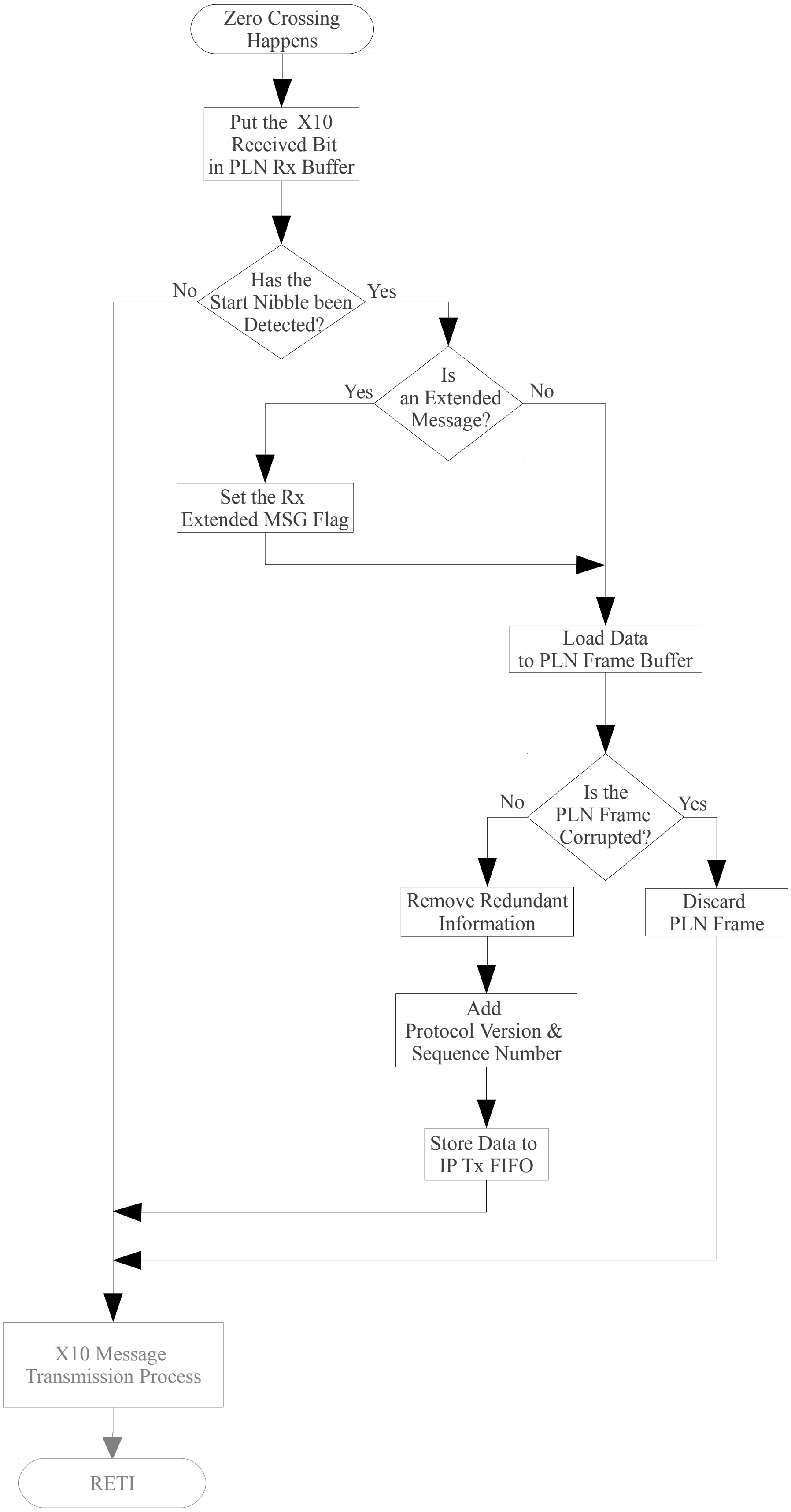

Zero Crossing Happens
Put the X10 Received Bit in PLN Rx Buffer
Has the Start Nibble been Detected?
No
Yes
Is an Extended Message?
Yes
No
Set the Rx Extended MSG Flag
Load Data to PLN Frame Buffer
Is the PLN Frame Corrupted?
No
Yes
Remove Redundant Information
Discard PLN Frame
Add Protocol Version & Sequence Number
Store Data to IP Tx FIFO
X10 Message Transmission Process
RETI

Figure 6-5: X10 Message Reception Procedure.

6.6 X10 Message Transmission Process

When the X10 message reception has finished, interrupt routine checks if all the bits of the last transmitted frame has been transmitted to PLN and if not, it sends the next frame's bit and then program flow returns to main routine (Figure 6-6).

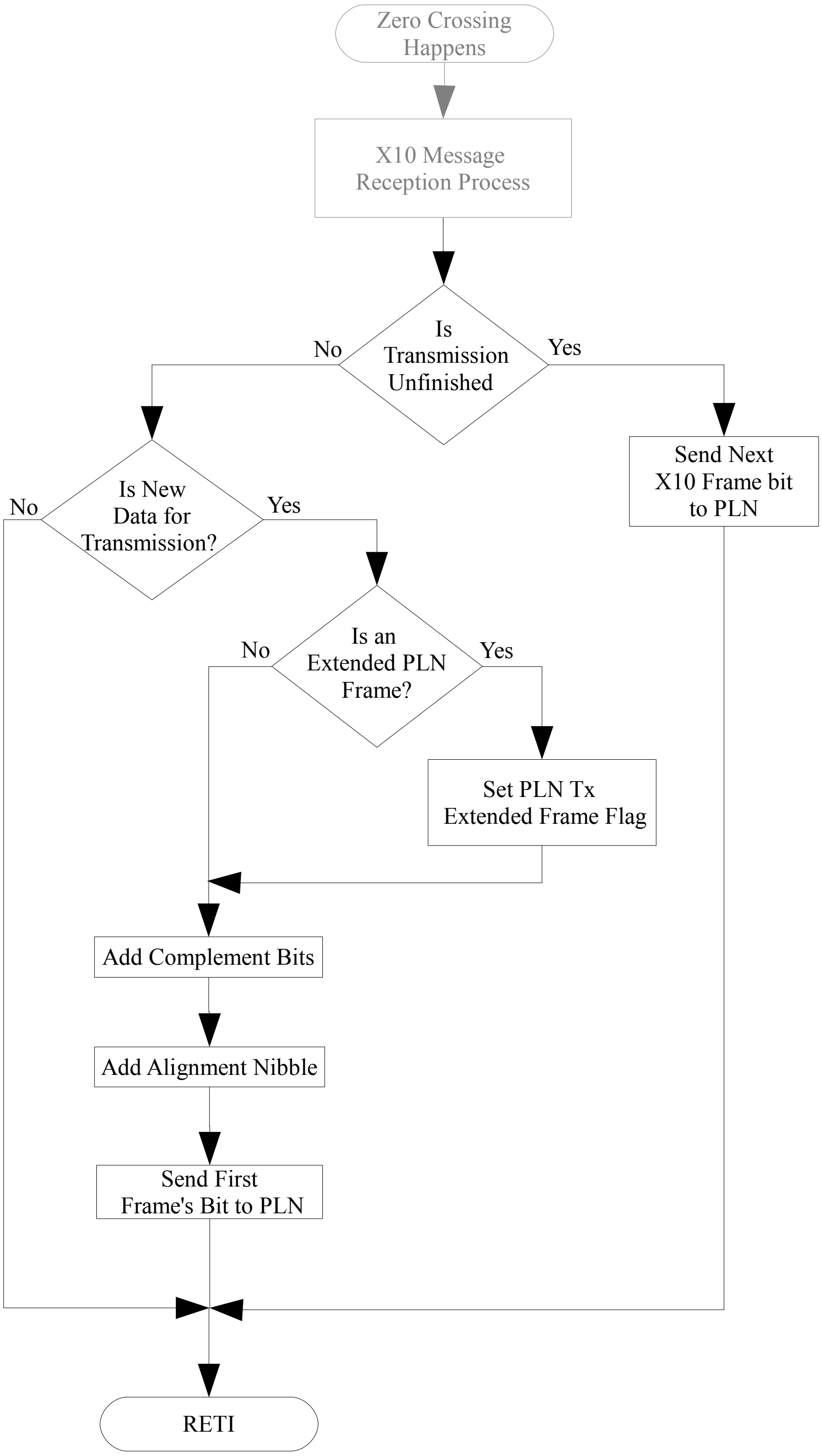

Zero Crossing Happens
X10 Message Reception Process
Is Transmission Unfinished
No
Yes
Send Next X10 Frame bit to PLN
Is New Data for Transmission?
No
Yes
Is an Extended PLN Frame?
No
Yes
Set PLN Tx Extended Frame Flag
Add Complement Bits
Add Alignment Nibble
Send First Frame's Bit to PLN
RETI

Figure 6-6: X10 Message Transmission Procedure.

If the last X10 frame transmission has finished and there is X10 message waiting for transmission to PLN, interrupt routine composes the X10 frame by adding complement bits and "1110" alignment nibble. In addition, if the message is an extended message, routine sets a flag on in order for its subroutines to know the length of the frame under transmission. Finally, the interrupt routine sends the first bit of the X10 frame and then the program flow returns to the main routine.

6.7 PLN Access Protocol

X10 protocol has adapted a CSMA algorithm in order X10 transmitters to avoid collisions [20].

Figure 6-7 depicts the Finite-State Machine (FSM) diagram of the algorithm that we have implemented in the platform.

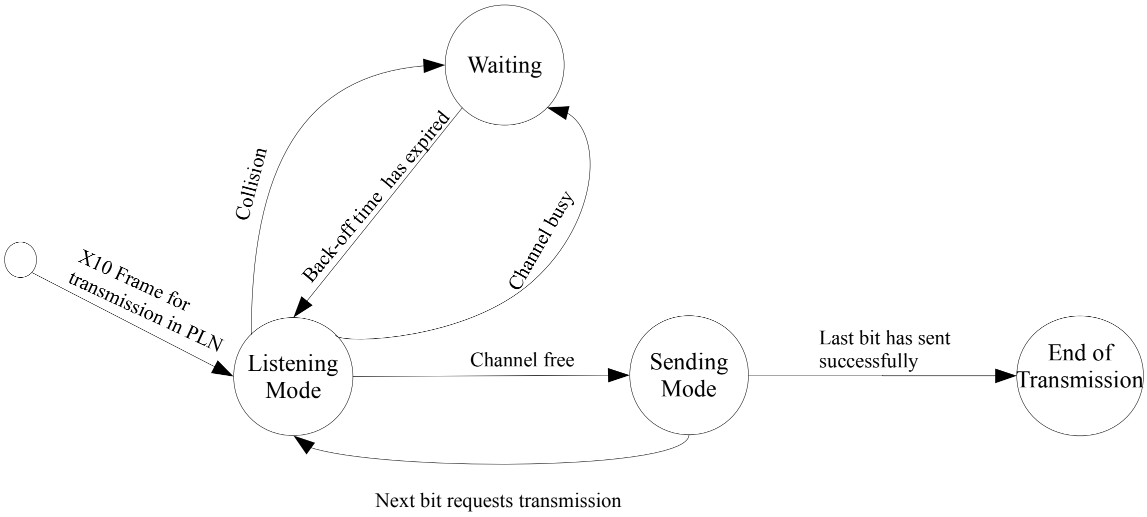

Waiting
Collision
Back-off time has expired
Channel busy
X10 Frame for transmission in PLN
Listening Mode
Channel free
Sending Mode
Last bit has sent successfully
End of Transmission
Next bit requests transmission

Figure 6-7: FSM Diagram of the PLN Access Protocol.

When MCU has a frame for transmission, it must wait for access to the PLN for 10 half mains cycles[33], during which the line must have been continuously clear of data '1' bits. If a '1' bit is detected, routine restarts its access timing and wait for another 10 half mains cycles.

After line access has been achieved, the transmitter must check the line during the transmission of a '0' bit (absence of 120 kHz carrier) to be sure that no other transmitter is transmitting. If a collision occurs, the transmitter must abort its transmission immediately and start-up the PLN access procedure again.

[33] In the X10 protocol the waiting time for access to PLN is 8, 9, or 10 half mains cycles.

Chapter 7 Test and Validation

7.1 Introduction

In this chapter the X10/IP layer and the related hardware designed in this project have tested, evaluated and validated.

A series of tests have been made in order to adjust all the necessary parameters for the correct functioning of the system. Additional tests, in order to evaluate the proper functioning of both hardware and software parts of the system in extreme conditions, were performed.

The devices and the instruments that have been used in this study are the following:

- X10 software and equipment from Marmitek company:
  - AM12™ Appliance Module
  - LM12™ Lamp Module
  - CM15Pro™ Programmable Computer Interface
  - ActiveHome Pro™ Software
- Ostinato, Network Traffic Generator
- Wireshark Network Protocol Analyser
- Fluke 192 SCOPEMETER 60 MHz Digital Oscilloscope and Multimeter

All the software tools were running in E4300 and DV4350EA Dell PCs with Linux (Lubuntu) operating system(OS).

Certain necessary tests and validations took place:

- *Validation of the platform* in order for it to work according to X10 and IP standards
- *Platform integration to IP network.* Platform has been tested as X10/IP node in both LAN and Internet networks

7.2 Software Tools and Devices

This chapter gives a brief description of the X10 equipment and software tools that have been used for the research.

7.2.1 CM15Pro™

The CM15Pro™ is a X10 protocol controller with wireless transceiver function[34]. CM15Pro™ in conjunction with the ActiveHome Pro™ software, are used for controlling X10 light and appliance modules.

7.2.2 LM12™ and AM12™

These equipments receive X10 frames from PLN and switch or dim the attached lamp or appliance.

7.2.3 ActiveHome Pro™

As already mentioned above, ActiveHome Pro™ software is connected through a serial port to CM15Pro™ X10 controller and gives us a virtual control panel for all X10 and X10 compatible products. ActiveHome Pro™ allows you to control your modules and place timers on them (Figure 7-1).

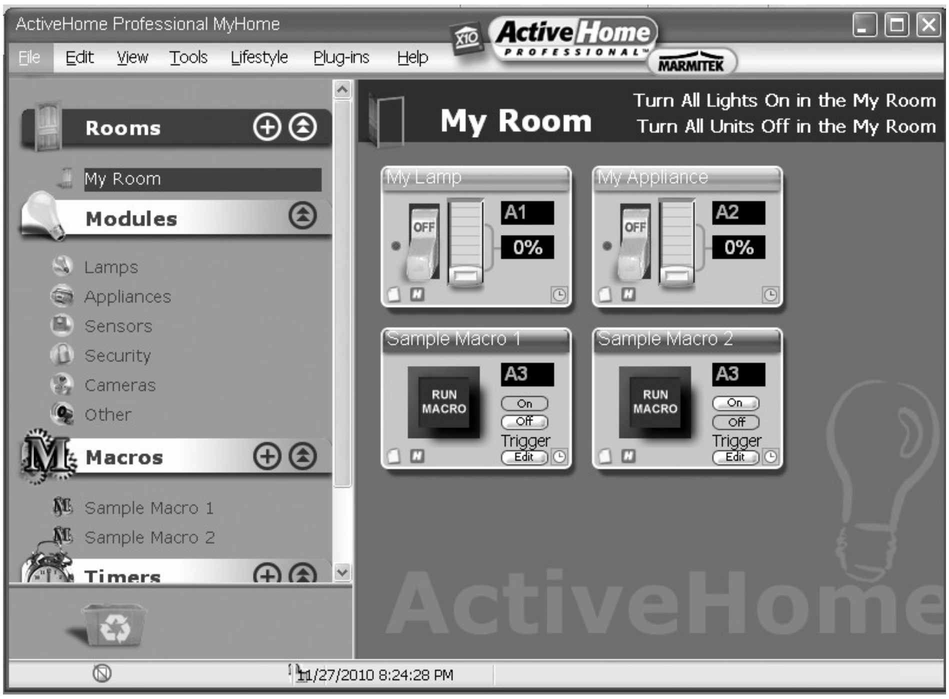


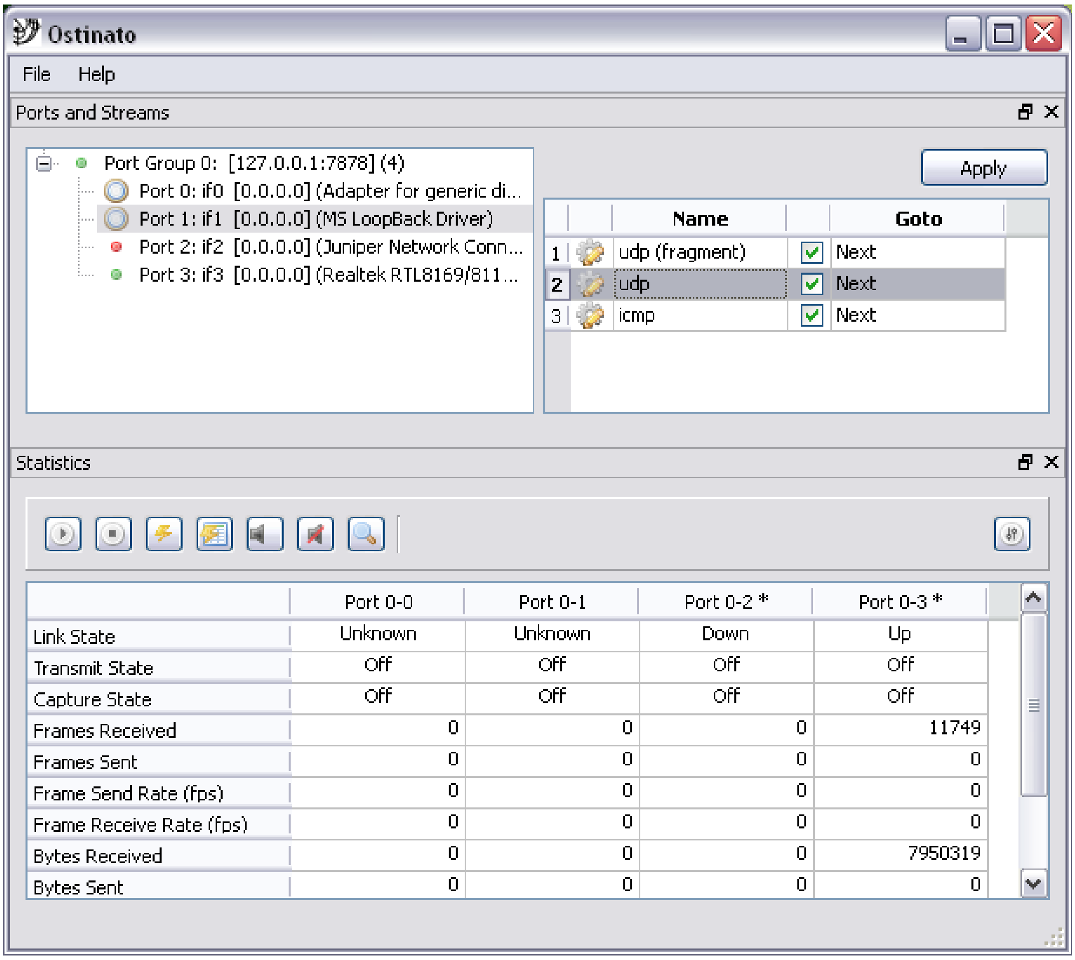
Figure 7-1: ActiveHome Pro™.

[34] When it receives a X10 message from a wireless remote control, it transmits the X10 message in PLN.

ActiveHome Pro™ software is a Windows OS application implemented by Marvitek company. In order to run Windows XP and ActiveHomePro™ software above Linux, Oracle's Virtual Box has been used.

7.2.4 Ostinato

Ostinato is an open source, cross platform network traffic generator (Figure 7-2). It composes and sends packets of several streams with different protocols at different rates.



- Port Group 0: [127.0.0.1:7878] (4)
  - Port 0: if0 [0.0.0.0] (Adapter for generic di...
  - Port 1: if1 [0.0.0.0] (MS LoopBack Driver)
  - Port 2: if2 [0.0.0.0] (Juniper Network Conn...
  - Port 3: if3 [0.0.0.0] (Realtek RTL8169/811...

		Name	Goto
1		udp (fragment)	☑ Next
2		udp	☑ Next
3		icmp	☑ Next

	Port 0-0	Port 0-1	Port 0-2 *	Port 0-3 *
Link State	Unknown	Unknown	Down	Up
Transmit State	Off	Off	Off	Off
Capture State	Off	Off	Off	Off
Frames Received	0	0	0	11749
Frames Sent	0	0	0	0
Frame Send Rate (fps)	0	0	0	0
Frame Receive Rate (fps)	0	0	0	0
Bytes Received	0	0	0	7950319
Bytes Sent	0	0	0	0

Figure 7-2: Ostinato.

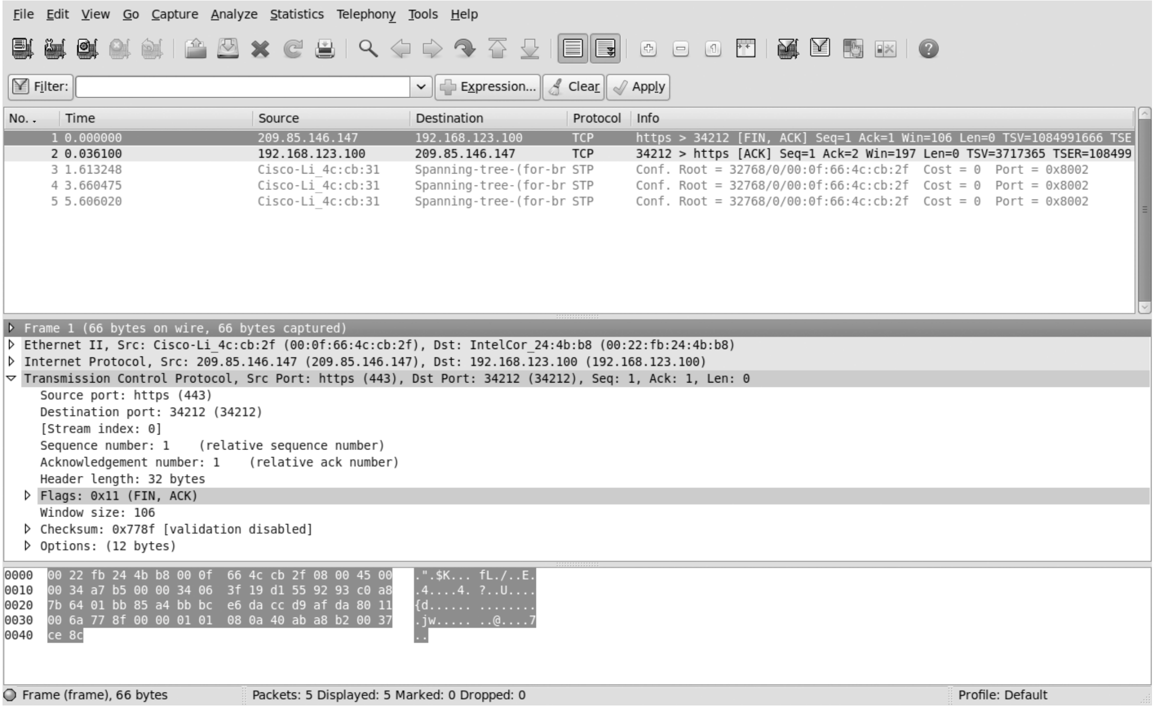
The major features of Ostinato are the following:

- Runs on Windows, Linux, BSD and Mac OS X
- Support for the most common standard protocols (Ethernet, VLAN, ARP, IPv4, Ipv6, IP Tunnelling, TCP, UDP, ICMPv4, ICMPv6, IGMP, HTTP, SIP, RTSP, NNTP etc.)

- Modify any field of the supported protocol
- User can provide Hex Dump - specifying some or all bytes in a packet
- Stack protocols in any arbitrary order
- Create and configure multiple streams
- Configure stream rates, bursts and number of transmitted packets

7.2.5 Wireshark

Wireshark is a cross platform and open source packet analyser. It is used for network troubleshooting, analysis, software and communications protocol development, and education (Figure 7-3). Originally named Ethereal, in May 2006 the project was renamed Wireshark due to trademark issues.



```
No. .  Time        Source               Destination          Protocol Info
     1 0.000000    209.85.146.147       192.168.123.100      TCP      https > 34212 [FIN, ACK] Seq=1 Ack=1 Win=106 Len=0 TSV=1084991666 TSE
     2 0.036100    192.168.123.100      209.85.146.147       TCP      34212 > https [ACK] Seq=1 Ack=2 Win=197 Len=0 TSV=3717365 TSER=108499
     3 1.613248    Cisco-Li_4c:cb:31    Spanning-tree-(for-br STP     Conf. Root = 32768/0/00:0f:66:4c:cb:2f  Cost = 0  Port = 0x8002
     4 3.660475    Cisco-Li_4c:cb:31    Spanning-tree-(for-br STP     Conf. Root = 32768/0/00:0f:66:4c:cb:2f  Cost = 0  Port = 0x8002
     5 5.606020    Cisco-Li_4c:cb:31    Spanning-tree-(for-br STP     Conf. Root = 32768/0/00:0f:66:4c:cb:2f  Cost = 0  Port = 0x8002
```

```
▷ Frame 1 (66 bytes on wire, 66 bytes captured)
▷ Ethernet II, Src: Cisco-Li_4c:cb:2f (00:0f:66:4c:cb:2f), Dst: IntelCor_24:4b:b8 (00:22:fb:24:4b:b8)
▷ Internet Protocol, Src: 209.85.146.147 (209.85.146.147), Dst: 192.168.123.100 (192.168.123.100)
▽ Transmission Control Protocol, Src Port: https (443), Dst Port: 34212 (34212), Seq: 1, Ack: 1, Len: 0
    Source port: https (443)
    Destination port: 34212 (34212)
    [Stream index: 0]
    Sequence number: 1    (relative sequence number)
    Acknowledgement number: 1    (relative ack number)
    Header length: 32 bytes
  ▷ Flags: 0x11 (FIN, ACK)
    Window size: 106
  ▷ Checksum: 0x778f [validation disabled]
  ▷ Options: (12 bytes)
```

```
0000  00 22 fb 24 4b b8 00 0f  66 4c cb 2f 08 00 45 00   .".$K... fL./..E.
0010  00 34 a7 b5 00 00 34 06  3f 19 d1 55 92 93 c0 a8   .4....4. ?..U....
0020  7b 64 01 bb 85 a4 bb bc  e6 da cc d9 af da 80 11   {d...... ........
0030  00 6a 77 8f 00 00 01 01  08 0a 40 ab a8 b2 00 37   .jw..... ..@....7
0040  ce 8c                                              ..
```

```
Frame (frame), 66 bytes    Packets: 5 Displayed: 5 Marked: 0 Dropped: 0    Profile: Default
```

Figure 7-3: Wireshark Analyser.

Wireshark is software that "understands" the structure of different networking protocols. Thus, it is able to display the encapsulation and the fields along with their meanings of different packets specified by different networking protocols. Wireshark uses pcap (packet capture software)[35] to capture packets, so it can only capture the

[35] http://en.wikipedia.org/wiki/Pcap.

packets on the types of networks that pcap supports.

The main features of the Wireshark are the following:

- Data can be captured "from the wire", from a live network connection or read from a file that recorded already-captured packets
- Live data can be read from a number of types of network, including Ethernet, IEEE 802.11, PPP, and loopback
- Captured network data can be browsed via a GUI, or via the terminal (command line) version of the utility, tshark
- Captured files can be programmatically edited or converted via command-line switches to the "editcap" program
- Data display can be refined using a display filter
- Plugins can be created for dissecting new protocols
- VoIP calls in the captured traffic can be detected

7.3 Validation of the Platform

In this initial test the ability of the platform to produce X10 frames and X10/IP messages, compatible with X10 and IP standards accordingly, is tested and evaluated. Tests were done in the following fields:

- Platform's X10 interface
- X10/IP layer operations

7.3.1 X10 Interface Tests

In order to test the X10 interface of the platform the followings tests took place:

- X10 transmission signal timing
- Measurements of data ‘1’ bit (120 kHz burst)
- Evaluation of CSMA algorithm
- Platform's cooperation with LM12™ and AM12™ modules

Figure 7-4 shows the connectivity of the platform and the equipment that we have used.

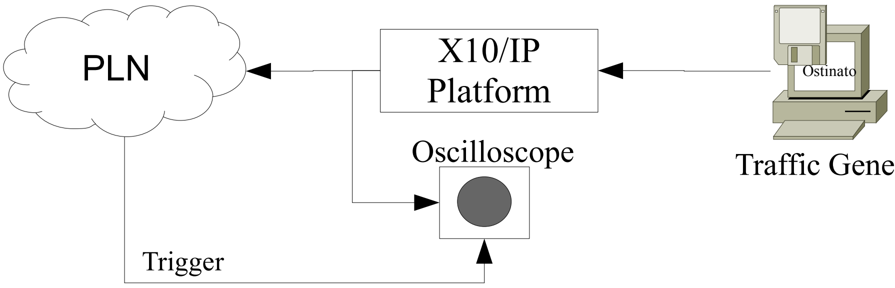


Figure 7-4: Connectivity Diagram for Timing Test.

For the test purposes we sent a X10/IP message to the platform with traffic generator (Ostinato) and with the oscilloscope we measured the X10 output signal of the platform (at BC639's collector). The oscilloscope was triggered from mains.
As Figure 7-5 shows, the drift that the bit '1' signal has in accordance to mains zero crossing is 120 μs. According to X10 specifications (Section 3.4.2) bit '1' signal needs 48 or more cycles of 120 kHz carrier in time window begins approximately 250 μs and ends 900 μs. In our case the bit '1' signal is transmitted from 120 μs to 1120 μs. Thus, the platform fulfills the condition of the specification.

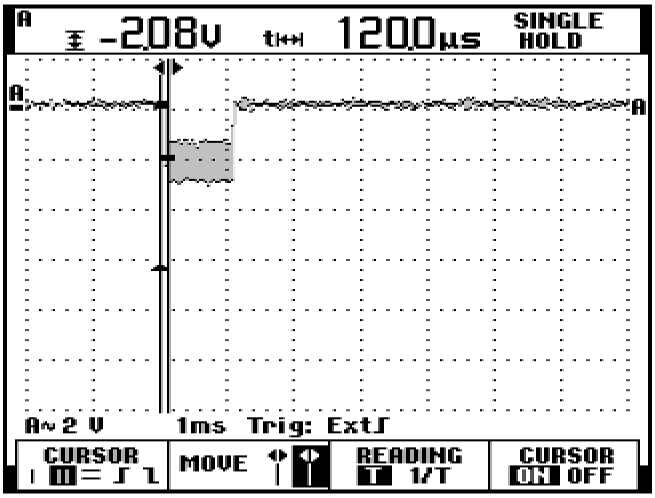


Figure 7-5: Transmission Amplifier Output.

In addition, as we have already shown in Figure 5-8 the X10 interface is in accordance to European regulations in PLC.

In order to evaluate the CSMA algorithm, we connected CM15Pro™, LM12™ and AM12™ devices in PLN and with the help of ActiveHome Pro™ we transmitted continuous X10 messages to LM12 (Figure 7-6). In the same time we sent a “switch on/off” X10/IP message to AM12™ from traffic generator. We observed that only when we stop to send X10 messages from CM15Pro™ the AM12™ reacts. Therefore, the CSMA algorithm prevents a X10 transmission when the PLN is busy.

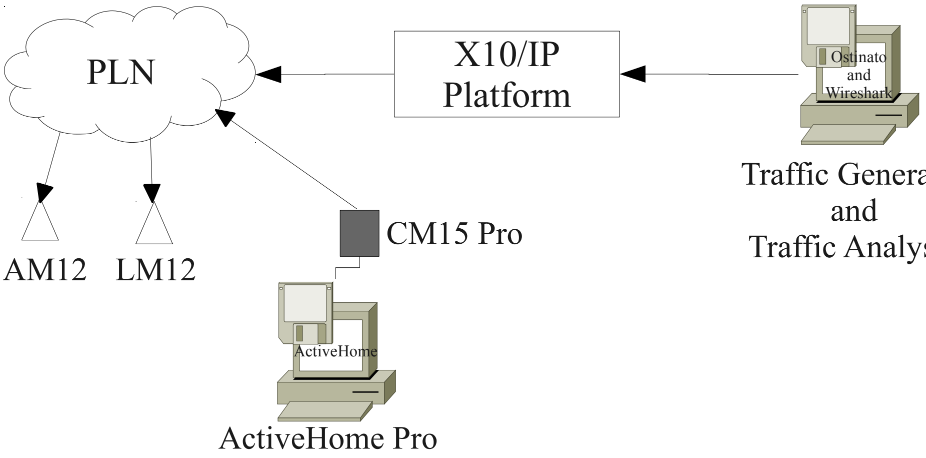


Figure 7-6: Connectivity Diagram for CSMA Algorithm Test.

In addition to the above test, a lot of tests (on, off, dim etc.) have been made in order to prove that the platform can control both AM12™ the LM12™ modules successfully.

7.3.2 X10/IP Layer Operation Tests

The following operations of the platform have been tested:

- The construction of the correct X10/IP message
- The operation of the ARQ algorithm
- Its ability to send extended messages to PLN

With the help of ActiveHome Pro™ and CM15Pro™ we sent various X10 frames to PLN and we observed the X10/IP messages in traffic analyser (Figure 7-7). All the X10/IP messages had the correct structure and format.

Furthermore, sending or missing the ACK message from traffic generator to the platform we approved that the ARQ algorithm is working as it is expected.

X10/IP Extended messages are sent from traffic generator and we observed the correspondent X10 messages in oscillator.

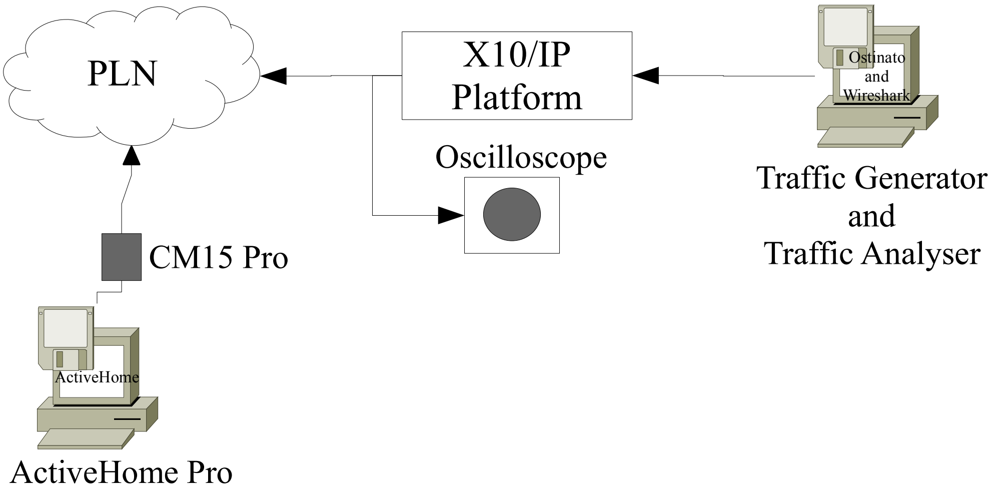


Figure 7-7: Connectivity Diagram for X10/IP Layer Operation Test.

7.4 Platform Integration to IP Network

This was the most important test for our project. Its objective was to check the proper functioning of the platform in both LAN and Internet networks.

Figure 7-8 shows the connectivity of the platform and the equipment that we have used for LAN tests.

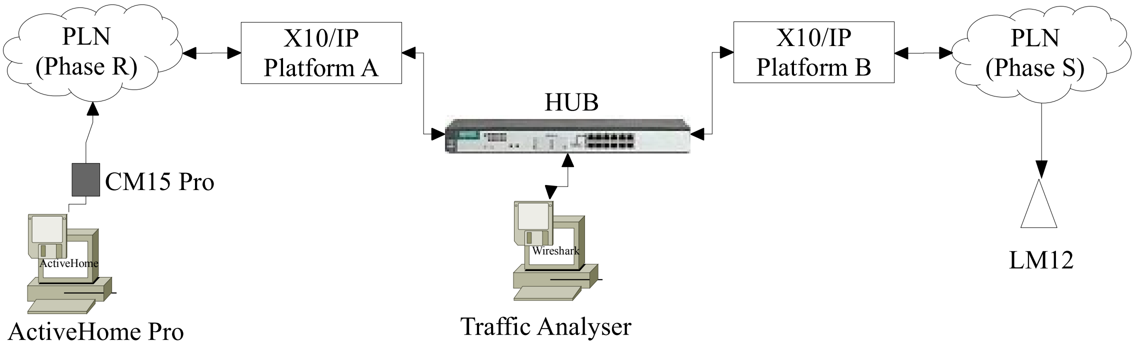


Figure 7-8: Connectivity Diagram for Platform Integration to LAN.

Two X10/IP platforms have been used, which were connected in different phases of the building's three phase PLN network. As the X10 controller (CM15) was in a different phase than the actuator they could not communicate thought power lines. The only way for X10 controller to was able to send X10 command to actuator was through X10/IP nodes. All the X10 messages have been delivered successfully and the measured delay for X10 Frame propagation between the two phases was 228 ms.

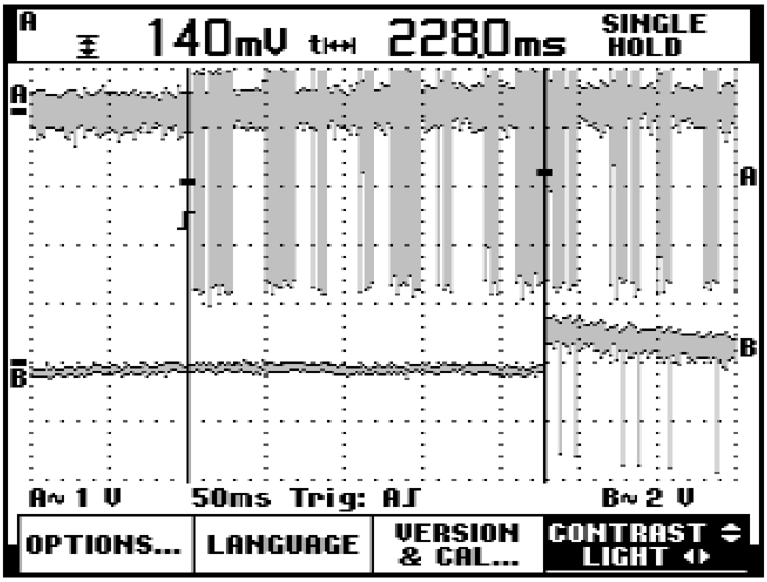


Figure 7-9: Propagation Delay of X10 Signals between R and S Phase.

Finally, as Figure 7-10 depicts, we have connected the X10/IP nodes to Internet.

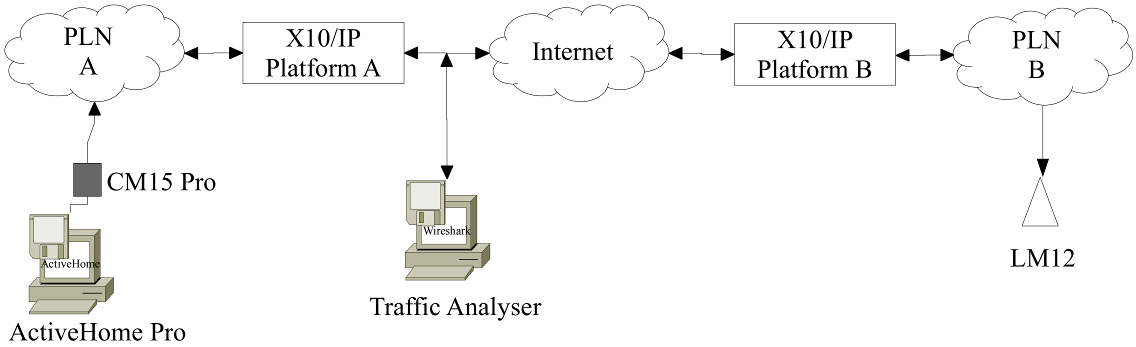


Figure 7-10: Connectivity Diagram for Platform Integration to Internet.

The delay between the controller command and the actuator response was at about one second (Internet delay ≈ 150 ms) and the flow of dimming and brightening messages was stable (very low jitter).

To sum up all the local and remote tests have been finished successfully.

Chapter 8 Conclusions and Further Work

8.1 Conclusions

The target of this research was to design and implement an application sub-layer in Internet protocol suite that gives the ability to open PLN protocols to convergence to IP.

The application sub-layer (HAPoIP) accommodates all the processes needed for PLN data preparation and transmission to IP network and vice versa. For maximum bandwidth utilization, the proposed protocol sends only the pure PLN messages through IP. The static part of the PLN frame is removed in local node and regenerated again in the remote node.

In addition, as multicast transmission reduce the traffic load in point to multipoint transmissions as much as possible, the UDP transport mechanism has been employed. Therefore, for reliability reasons, a ARQ mechanism must take place in case the PLN protocol has not its own or the IP network delay is not tolerable and a proxy function must be accommodated. According to the analysis of the PLN protocols and taking into account the Internet network characteristics like delay and packet loss the proposed mechanism engages a simple Stop and Wait ARQ protocol.

In order to verify the feasibility of the proposed convergence mechanism two test X10/IP nodes were implemented with particular attention to reduce the cost and maximise both hardware and software flexibility.

The X10/IP node consists of a power line modem, which is responsible for X10 protocol's physical layer, the 8-bit microcontroller unit, which implements both the data link layer of the X10 protocol and the HAPoIP layer, and the UTP/IP module, which is responsible for the data tranceiving via IP network. The whole node's software has been written in C programming language for maximum portability and after compilation does not exceed the 8K bytes program memory.

According to the results, very good performance of entire system has been achieved and the proposed mechanism can be an invaluable tool for the PLN automation industry, since two or more isolated power line networks can be logically merged and remotely controlled in a simple and reliable way.

8.2 Further Work

The following are suggested as areas of future work:

The simultaneous integration of different power line protocols in the same platform by mapping the messages of all the protocols in a common format. For example, platform implementations for 60 Hz power line network protocols could be implemented.

Also, a number of additional services can be added to the system in future improvements. Services like DHCP client or Internet Group Management Protocol (IGMP) support are welcome to our implementation.

Additionally, security services can be added to HAoIP nodes in order for only the authorised PLN nodes to be able to communicate with the remote PLN.

Finally, a simple user interface like a serial terminal port could give the possibility of node monitoring and systems errors and alarms reporting.

References

[1] S. Junestrand, “Being Private and Public at Home,” Ph.D. dissertation, School of Architecture, Royal Institute of Technology, Stockholm, Sweden, 2004.

[2] T. Satio, I. Tomoda, and Y. Takabatake, “Home gateway architecture and its implementation,” IEEE Transactions on Consumer Electronics, Vol. 46, No. 4, pp. 1161-1166, Jun. 2000.

[3] D. Valtchev and I. Frankov, "Service gateway architecture for a smart home, " in IEEE Communications Magazine, vol. 40, pp. 126-132, Apr. 2002.

[4] W. Zhang, J. Wang, and H. Ni, “Architecture Design of Embedded Home IMS Gateway Based on Convergence of IMS and Home Network,” International Conference on Embedded Software and Systems (ICESS '08), pp. 271-276, July 2008.

[5] D. Zhang, J. Xing, G. Chen, and K. Qin, “Collaborative Design of Heterogeneous Data Integration Communication Gataway Based on Embedded Platform,” International Conference on Embedded Software and Systems (ICESS '08), pp. 248, July 2008.

[6] V. Miorri, L. Tarrini, M. Manca and G. Tolomei, “An Innovative, Open-Standards Solution for Konnex Interoperability with other Domotic Middlewares,” IEEE Transactions on Consumer Electronics, Vol. 52, pp. 97-103, Feb. 2006.

[7] P. Dobrev, D. Fmolari, C. Kurzke, and B. Miller, “Device and Service Discovery in Home Networks with OSGi,” in IEEE Communications Magazine, Vol 40, No.8, pp. 86-92, Aug. 2002.

[8] K.Y. Cho, K.h. Choi, “A Novel Architecture of Home Gateway for Efficient Packet Process,” Proceedings of the IEEE Workshop on Knowledge Media Networking, pp. 63-67, July 2002.

[9] P. Bernstein, “Middleware: A Model for Distributed Services,” Communications of the ACM Magazine, Vol 39, pp. 86-97, Feb. 1996.

[10] K. Wacks, “The Successes and Failures of Standardization in Home Systems,” 2nd IEEE Conference on Standardization and Innovation in Information Technology, pp. 77, Oct. 2001.

[11] Basic Reference Model: *The Basic Model*, ISO/IEC Standard 7498-1-1994.

[12] E. Hall, *Internet Core Protocols: The Definitive Guide*. Sebastopol, CA: O'Reilly, 2000, ch. 2.

[13] *INTERNET PROTOCOL*, IETF Standard RFC 791-1981.

[14] A. Tanenbaum, *Computer Networks*. Upper Saddle River, New Jersey: Prentice Hall, 2003, ch. 6.

[15] *TRANSMISSION CONTROL PROTOCOL*, IETF Standard RFC 793-1981.

[16] *User Datagram Protocol*, IETF Standard RFC 768-1980.

[17] J. Jeon, “A Survey on Protocols for Home Networks based on Power Line Communication,” in Proc. 15th CISL Winter Workshop. Kushu, 2002, pp. 18.

[18] M. Tlich, H. Chaouche, A. Zeddam and P. Pagani, “Novel Approach for PLC Impulsive Noise Modelling,” IEEE International Symposium on Power Line Communications and Its Applications (ISPLC 2009), pp. 20, March 2009.

[19] *SmartHouse Code of Practice*, CENELEC Standard CWA 50487-2005.

[20] *"Standard" and "Extended" X-10 Code Formats*, X10 Wireless Technology Inc Standard XCT798-1993.

[21] *The Details*, 1st ed., Smarthome, Inc., Irvine, CA, 2005, pp. 10-39.

[22] *Developer's Guide*, 1st ed., Smarthome, Inc., Irvine, CA, 2007, pp. 60-67.

[23] *UPB Technology Description*, 4th ed., PCS, Inc,. Northridge, CA, 2007, pp. 4-19.

[24] H. Merz, T. Hansemann and C. Hübner, *Building Automation*. Berlin: Springer, 2009, ch. 3.

[25] *Media and media dependent layers-Network basedon HBES Class1, TwistedPair*, CENELEC Standard EN 50090-5-2-2004.

[26] *Media and media dependent layers-Power line for HBES Class 1*, CENELEC Standard EN 50090-5-1-2005.

[27] *Media independent layers–Transport layer, network layerand general parts of datalink layer for HBES Class 1*, CENELEC Standard EN 50090-4-2-2004.

[28] *Building network protocol-Part 1: Protocol Stack*, CENELEC Standard EN 14908-1-2004.

[29] *Control network protocol-Part 3: Power Line Channel Specification*, CENELEC Standard EN 14908-3-2006.

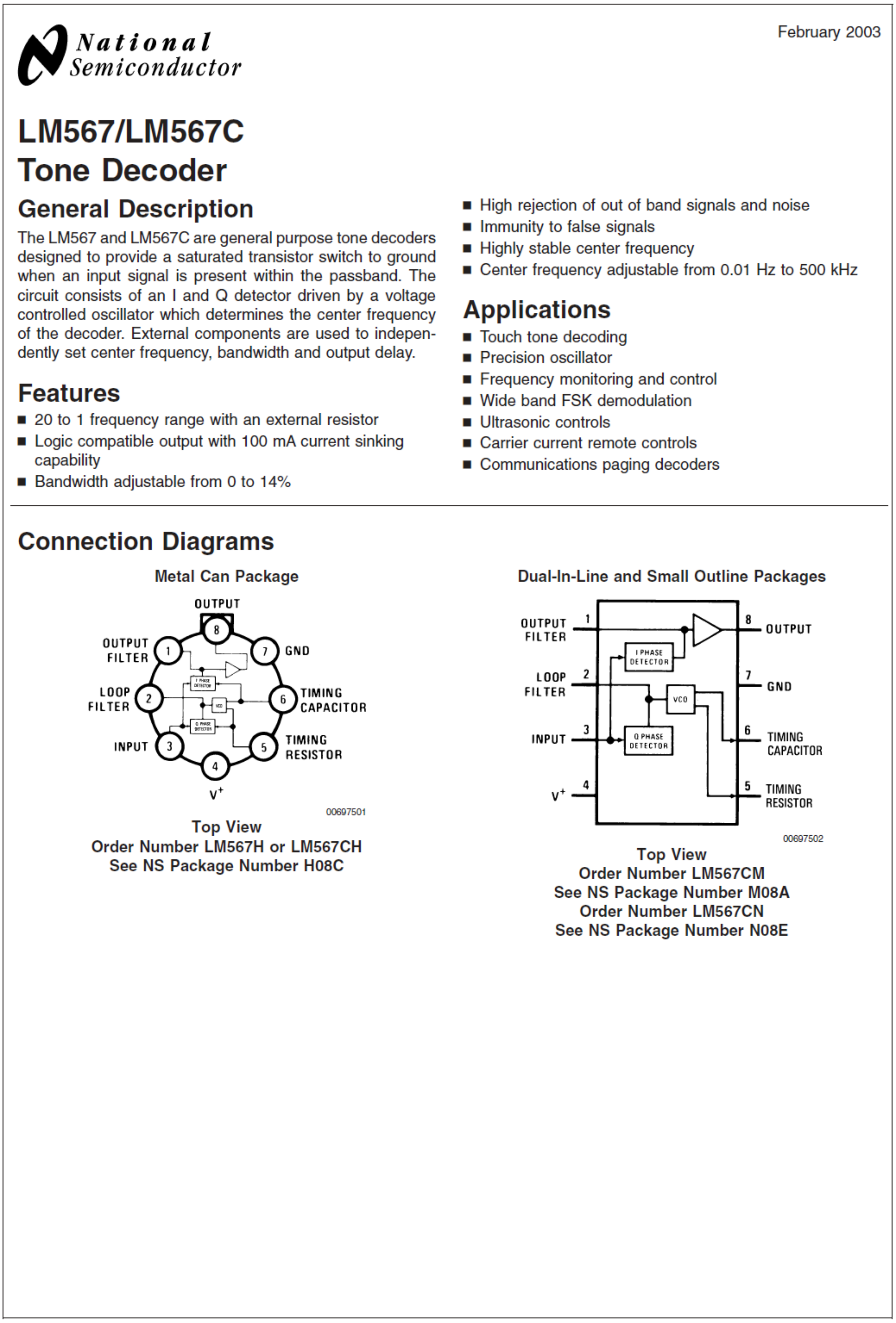
[30] K. Y. Cho, S. Moon, Z. L. Zhang, K. Papagiannaki and C. Diot, “Analysis of Point-To-Point Packet Delay In an Operational Network,” 23th IEEE Conference of Computer and Communications Societies (INFOCOM), pp. 1797, March 2004.

[31] L. L. Peterson and B. S. Davie, *Computer Networks*. San Francisco: Morgan Kaufmann Publishers, 2003, ch. 2.

[32] *ATmega8535 Datasheet*, Atmel Corporation, Oct. 2006.

[33] *AVR STK500 User Guide*, Atmel Corporation, Mar. 2003.

APPENDIX A



National Semiconductor

February 2003

LM567/LM567C Tone Decoder

General Description

The LM567 and LM567C are general purpose tone decoders designed to provide a saturated transistor switch to ground when an input signal is present within the passband. The circuit consists of an I and Q detector driven by a voltage controlled oscillator which determines the center frequency of the decoder. External components are used to independently set center frequency, bandwidth and output delay.

Features

- 20 to 1 frequency range with an external resistor
- Logic compatible output with 100 mA current sinking capability
- Bandwidth adjustable from 0 to 14%
- High rejection of out of band signals and noise
- Immunity to false signals
- Highly stable center frequency
- Center frequency adjustable from 0.01 Hz to 500 kHz

Applications

- Touch tone decoding
- Precision oscillator
- Frequency monitoring and control
- Wide band FSK demodulation
- Ultrasonic controls
- Carrier current remote controls
- Communications paging decoders

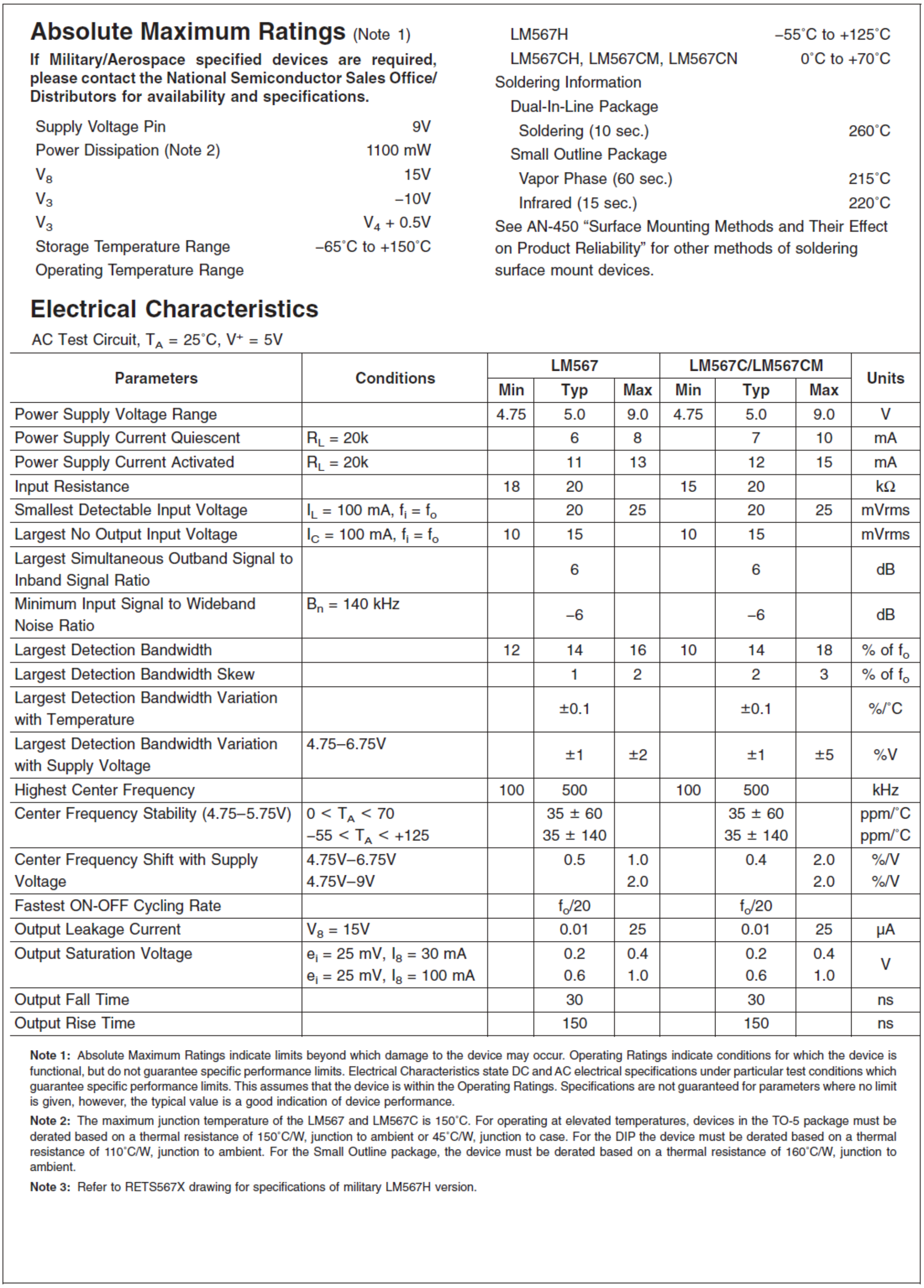
Connection Diagrams

Metal Can Package

Top View
Order Number LM567H or LM567CH
See NS Package Number H08C

Dual-In-Line and Small Outline Packages

Top View
Order Number LM567CM
See NS Package Number M08A
Order Number LM567CN
See NS Package Number N08E



Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage Pin	9V
Power Dissipation (Note 2)	1100 mW
V_8	15V
V_3	−10V
V_3	V_4 + 0.5V
Storage Temperature Range	−65°C to +150°C
Operating Temperature Range	
LM567H	−55°C to +125°C
LM567CH, LM567CM, LM567CN	0°C to +70°C
Soldering Information	
Dual-In-Line Package	
Soldering (10 sec.)	260°C
Small Outline Package	
Vapor Phase (60 sec.)	215°C
Infrared (15 sec.)	220°C

See AN-450 “Surface Mounting Methods and Their Effect on Product Reliability” for other methods of soldering surface mount devices.

Electrical Characteristics

AC Test Circuit, T_A = 25°C, V^+ = 5V

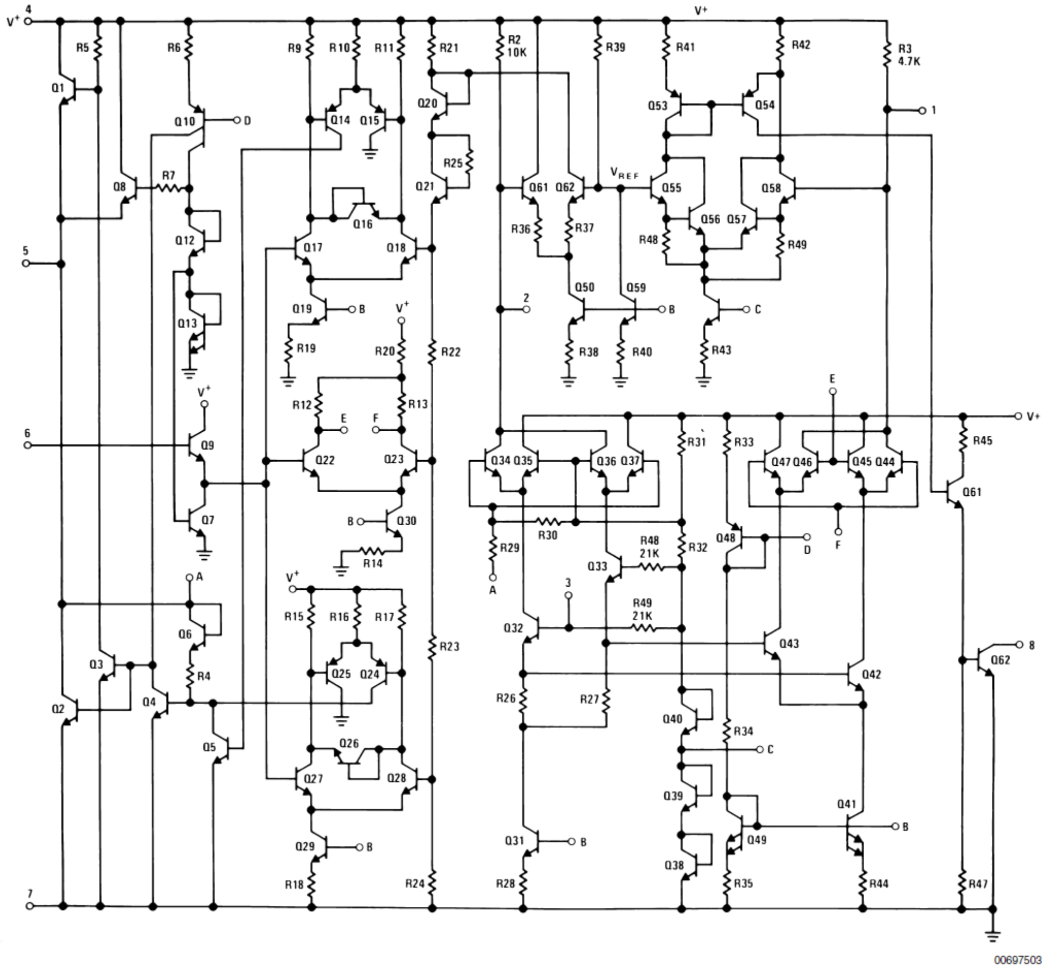
Parameters	Conditions	LM567			LM567C/LM567CM			Units
		Min	Typ	Max	Min	Typ	Max	
Power Supply Voltage Range		4.75	5.0	9.0	4.75	5.0	9.0	V
Power Supply Current Quiescent	R_L = 20k		6	8		7	10	mA
Power Supply Current Activated	R_L = 20k		11	13		12	15	mA
Input Resistance		18	20		15	20		kΩ
Smallest Detectable Input Voltage	I_L = 100 mA, $f_i = f_o$		20	25		20	25	mVrms
Largest No Output Input Voltage	I_C = 100 mA, $f_i = f_o$	10	15		10	15		mVrms
Largest Simultaneous Outband Signal to Inband Signal Ratio			6			6		dB
Minimum Input Signal to Wideband Noise Ratio	B_n = 140 kHz		−6			−6		dB
Largest Detection Bandwidth		12	14	16	10	14	18	% of f_o
Largest Detection Bandwidth Skew			1	2		2	3	% of f_o
Largest Detection Bandwidth Variation with Temperature			±0.1			±0.1		%/°C
Largest Detection Bandwidth Variation with Supply Voltage	4.75–6.75V		±1	±2		±1	±5	%V
Highest Center Frequency		100	500		100	500		kHz
Center Frequency Stability (4.75–5.75V)	$0 < T_A < 70$ $-55 < T_A < +125$		35 ± 60 35 ± 140			35 ± 60 35 ± 140		ppm/°C ppm/°C
Center Frequency Shift with Supply Voltage	4.75V–6.75V 4.75V–9V		0.5	1.0 2.0		0.4	2.0 2.0	%/V %/V
Fastest ON-OFF Cycling Rate			$f_o/20$			$f_o/20$		
Output Leakage Current	V_8 = 15V		0.01	25		0.01	25	µA
Output Saturation Voltage	e_i = 25 mV, I_8 = 30 mA e_i = 25 mV, I_8 = 100 mA		0.2 0.6	0.4 1.0		0.2 0.6	0.4 1.0	V
Output Fall Time			30			30		ns
Output Rise Time			150			150		ns

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is functional, but do not guarantee specific performance limits. Electrical Characteristics state DC and AC electrical specifications under particular test conditions which guarantee specific performance limits. This assumes that the device is within the Operating Ratings. Specifications are not guaranteed for parameters where no limit is given, however, the typical value is a good indication of device performance.

Note 2: The maximum junction temperature of the LM567 and LM567C is 150°C. For operating at elevated temperatures, devices in the TO-5 package must be derated based on a thermal resistance of 150°C/W, junction to ambient or 45°C/W, junction to case. For the DIP the device must be derated based on a thermal resistance of 110°C/W, junction to ambient. For the Small Outline package, the device must be derated based on a thermal resistance of 160°C/W, junction to ambient.

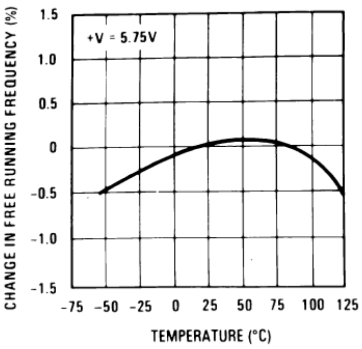
Note 3: Refer to RETS567X drawing for specifications of military LM567H version.

Schematic Diagram



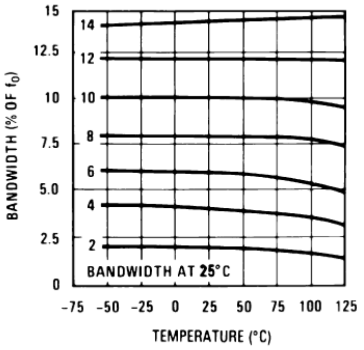
Typical Performance Characteristics

Typical Frequency Drift



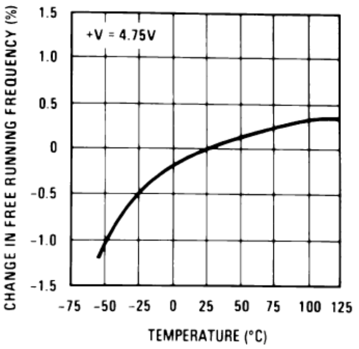
00697510

Typical Bandwidth Variation



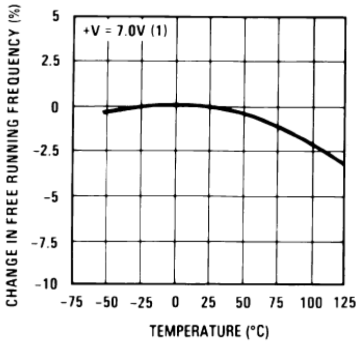
00697511

Typical Frequency Drift



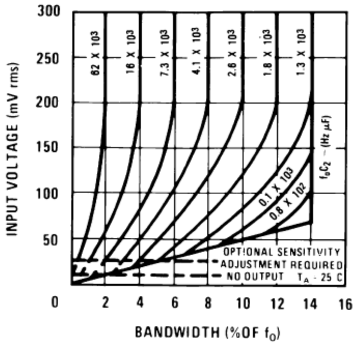
00697512

Typical Frequency Drift



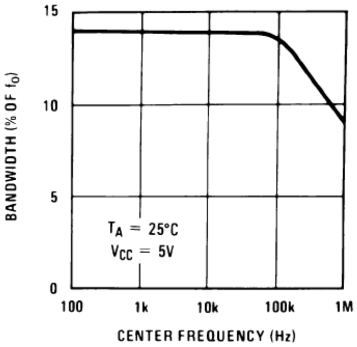
00697513

Bandwidth vs Input Signal Amplitude



00697514

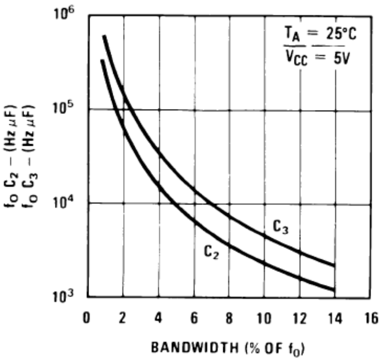
Largest Detection Bandwidth



00697515

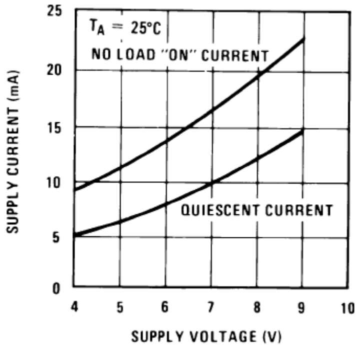
Typical Performance Characteristics (Continued)

Detection Bandwidth as a Function of C_2 and C_3



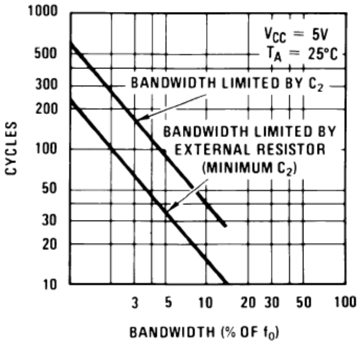
00697516

Typical Supply Current vs Supply Voltage



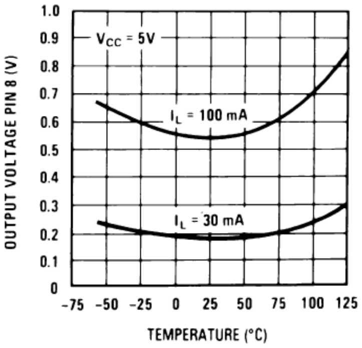
00697517

Greatest Number of Cycles Before Output



00697518

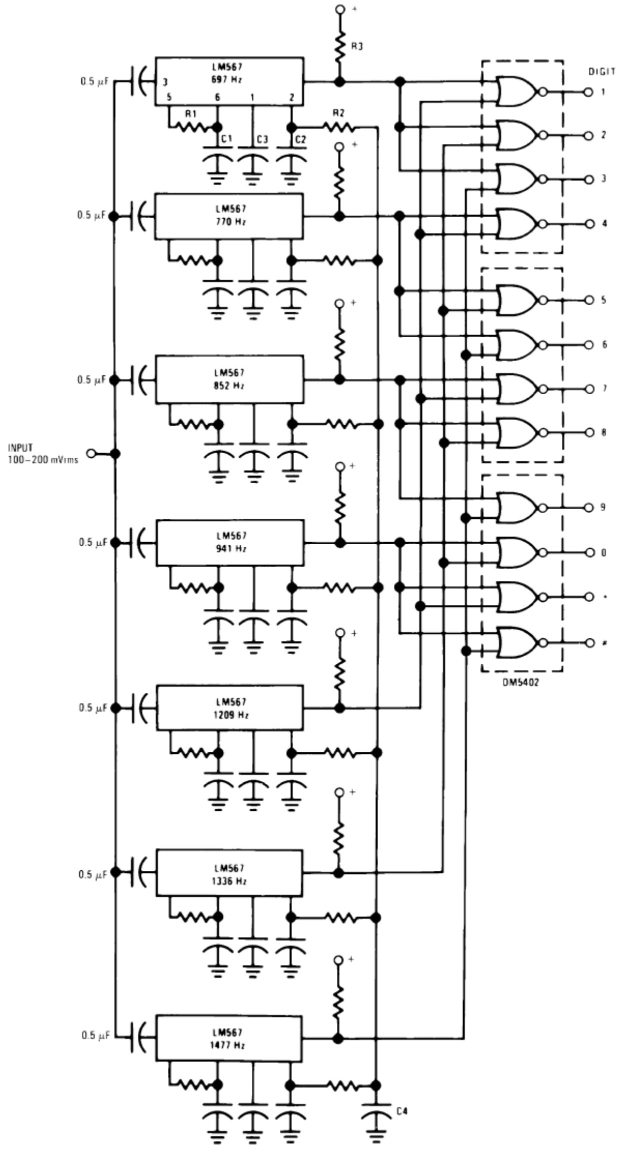
Typical Output Voltage vs Temperature



00697519

Typical Applications

Touch-Tone Decoder



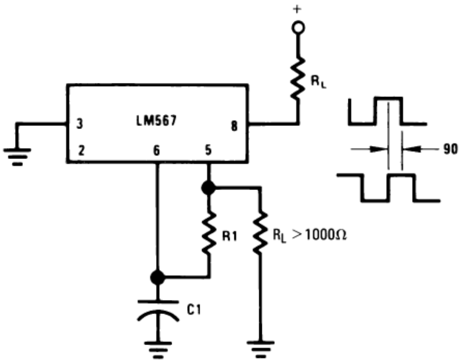
00697505

Component values (typ)

R1	6.8 to 15k
R2	4.7k
R3	20k
C1	0.10 mfd
C2	1.0 mfd 6V
C3	2.2 mfd 6V
C4	250 mfd 6V

Typical Applications (Continued)

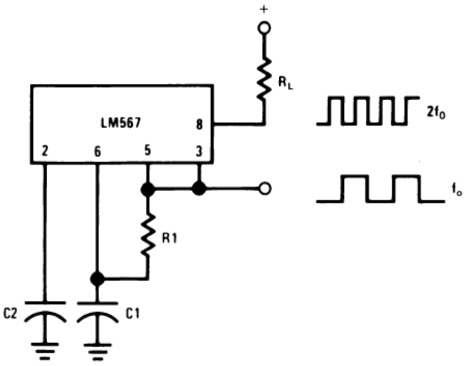
Oscillator with Quadrature Output



00697506

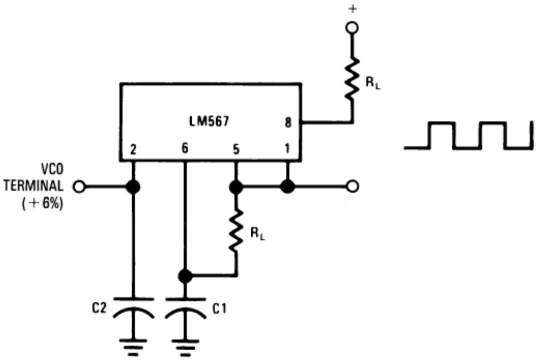
Connect Pin 3 to 2.8V to Invert Output

Oscillator with Double Frequency Output



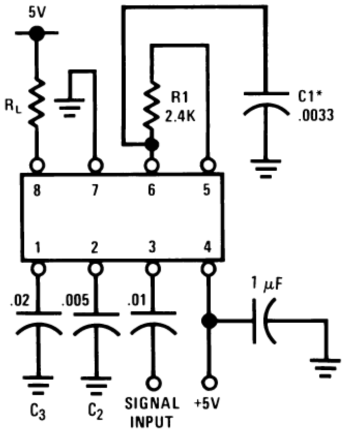
00697507

Precision Oscillator Drive 100 mA Loads



00697508

AC Test Circuit



00697509

f_i = 100 kHz + 5V

***Note:** Adjust for f_o = 100 kHz.

Applications Information

The center frequency of the tone decoder is equal to the free running frequency of the VCO. This is given by

$$f_o \cong \frac{1}{1.1\ R_1\ C_1}$$

The bandwidth of the filter may be found from the approximation

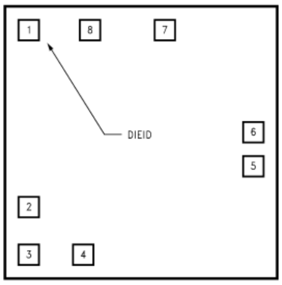
$$BW = 1070 \sqrt{\frac{V_i}{f_o C_2}} \text{ in \% of } f_o$$

Where:

V_i = Input voltage (volts rms), $V_i \leq 200mV$

C_2 = Capacitance at Pin 2(μF)

LM567C MDC MWC
TONE DECODER



00697523

Die Layout (C - Step)

DIE/WAFER CHARACTERISTICS

Fabrication Attributes		General Die Information	
Physical Die Identification	LM567C	Bond Pad Opening Size (min)	91µm x 91µm
Die Step	C	Bond Pad Metalization	0.5% COPPER_BAL. ALUMINUM
Physical Attributes		Passivation	VOM NITRIDE
Wafer Diameter	150mm	Back Side Metal	BARE BACK
Dise Size (Drawn)	1600µm x 1626µm 63.0mils x 64.0mils	Back Side Connection	Floating
Thickness	406µm Nominal		
Min Pitch	198µm Nominal		

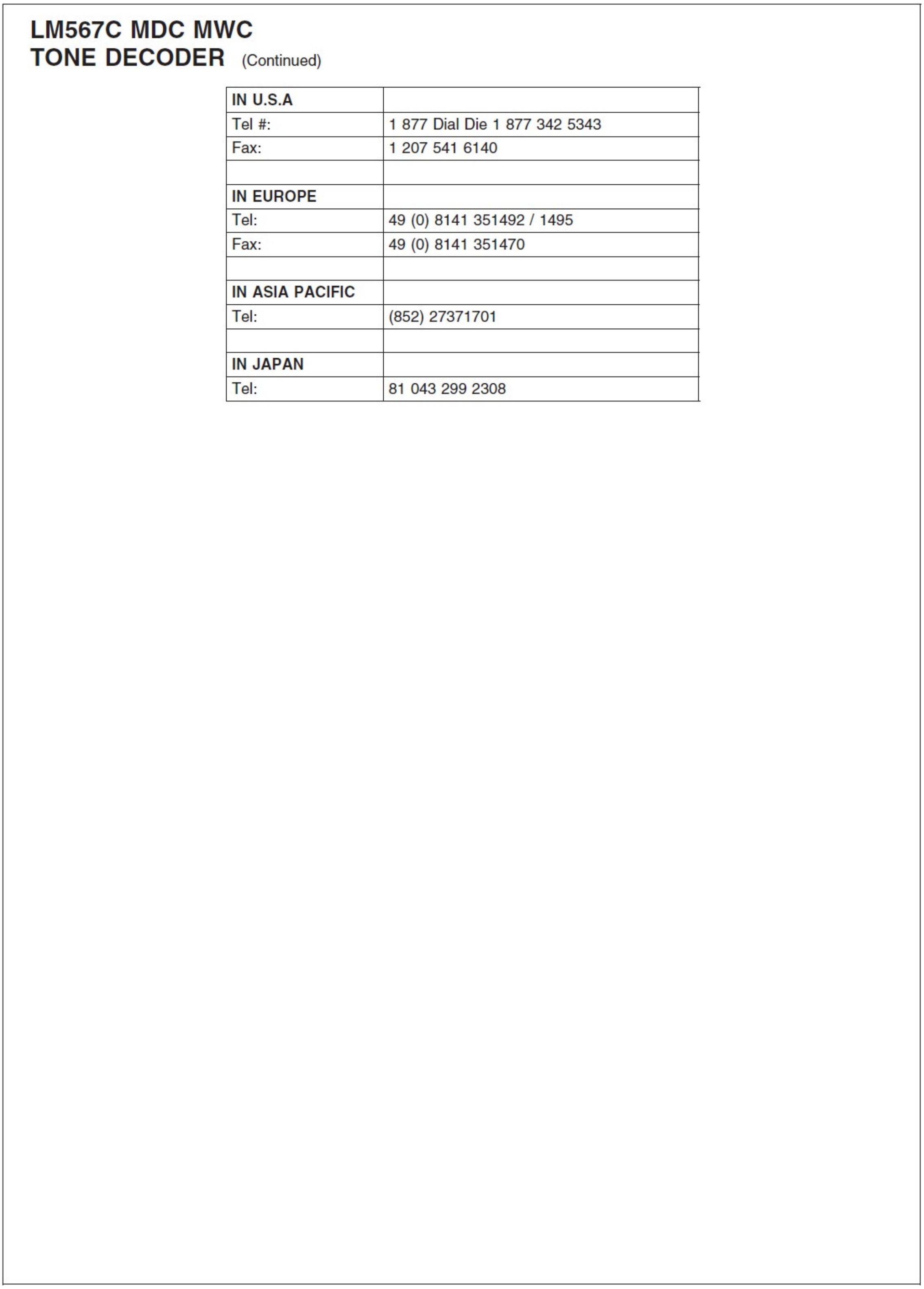
Special Assembly Requirements:

Note: Actual die size is rounded to the nearest micron.

Die Bond Pad Coordinate Locations (C - Step)

(Referenced to die center, coordinates in µm) NC = No Connection, N.U. = Not Used

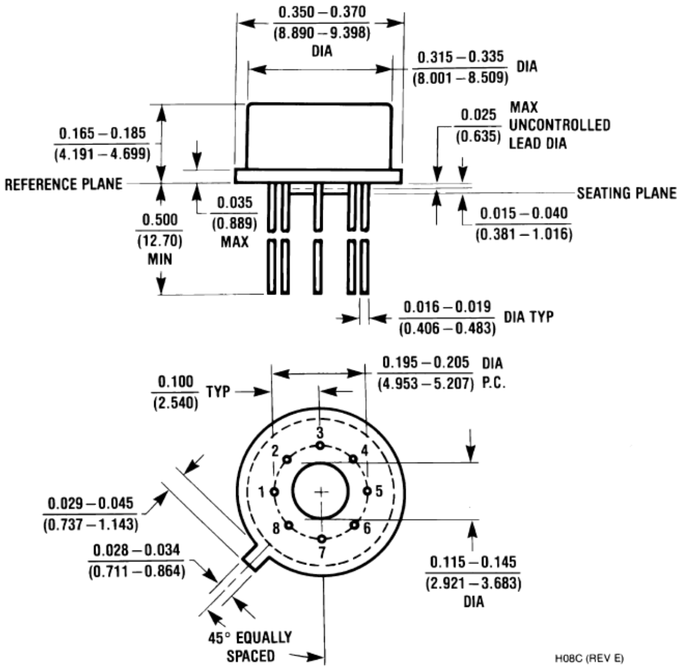
SIGNAL NAME	PAD# NUMBER	X/Y COORDINATES		PAD SIZE		
		X	Y	X		Y
OUTPUT FILTER	1	-673	686	91	x	91
LOOP FILTER	2	-673	-419	91	x	91
INPUT	3	-673	-686	91	x	91
V+	4	-356	-686	91	x	91
TIMING RES	5	673	-122	91	x	91
TIMING CAP	6	673	76	91	x	91
GND	7	178	686	117	x	91
OUTPUT	8	-318	679	117	x	104



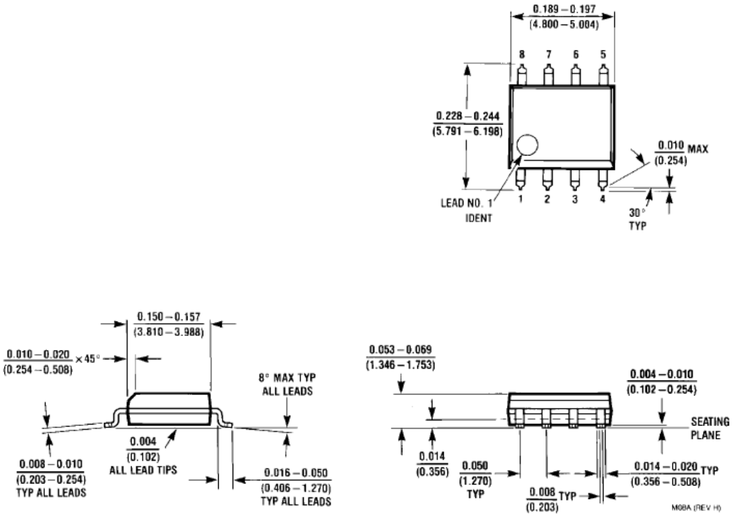
LM567C MDC MWC TONE DECODER (Continued)

IN U.S.A	
Tel #:	1 877 Dial Die 1 877 342 5343
Fax:	1 207 541 6140
IN EUROPE	
Tel:	49 (0) 8141 351492 / 1495
Fax:	49 (0) 8141 351470
IN ASIA PACIFIC	
Tel:	(852) 27371701
IN JAPAN	
Tel:	81 043 299 2308

Physical Dimensions inches (millimeters) unless otherwise noted

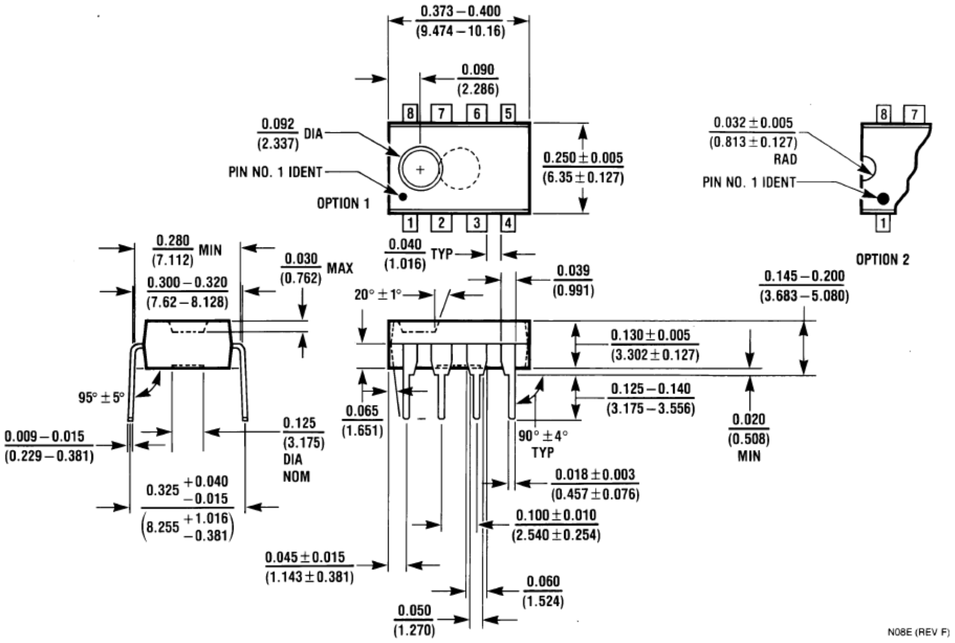


Metal Can Package (H)
Order Number LM567H or LM567CH
NS Package Number H08C



Small Outline Package (M)
Order Number LM567CM
NS Package Number M08A

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



Molded Dual-In-Line Package (N)
Order Number LM567CN
NS Package Number N08E

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.

For the most current product information visit us at www.national.com.

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

BANNED SUBSTANCE COMPLIANCE

National Semiconductor certifies that the products and packing materials meet the provisions of the Customer Products Stewardship Specification (CSP-9-111C2) and the Banned Substances and Materials of Interest Specification (CSP-9-111S2) and contain no "Banned Substances" as defined in CSP-9-111S2.

National Semiconductor
Americas Customer
Support Center
Email: new.feedback@nsc.com
Tel: 1-800-272-9959

www.national.com

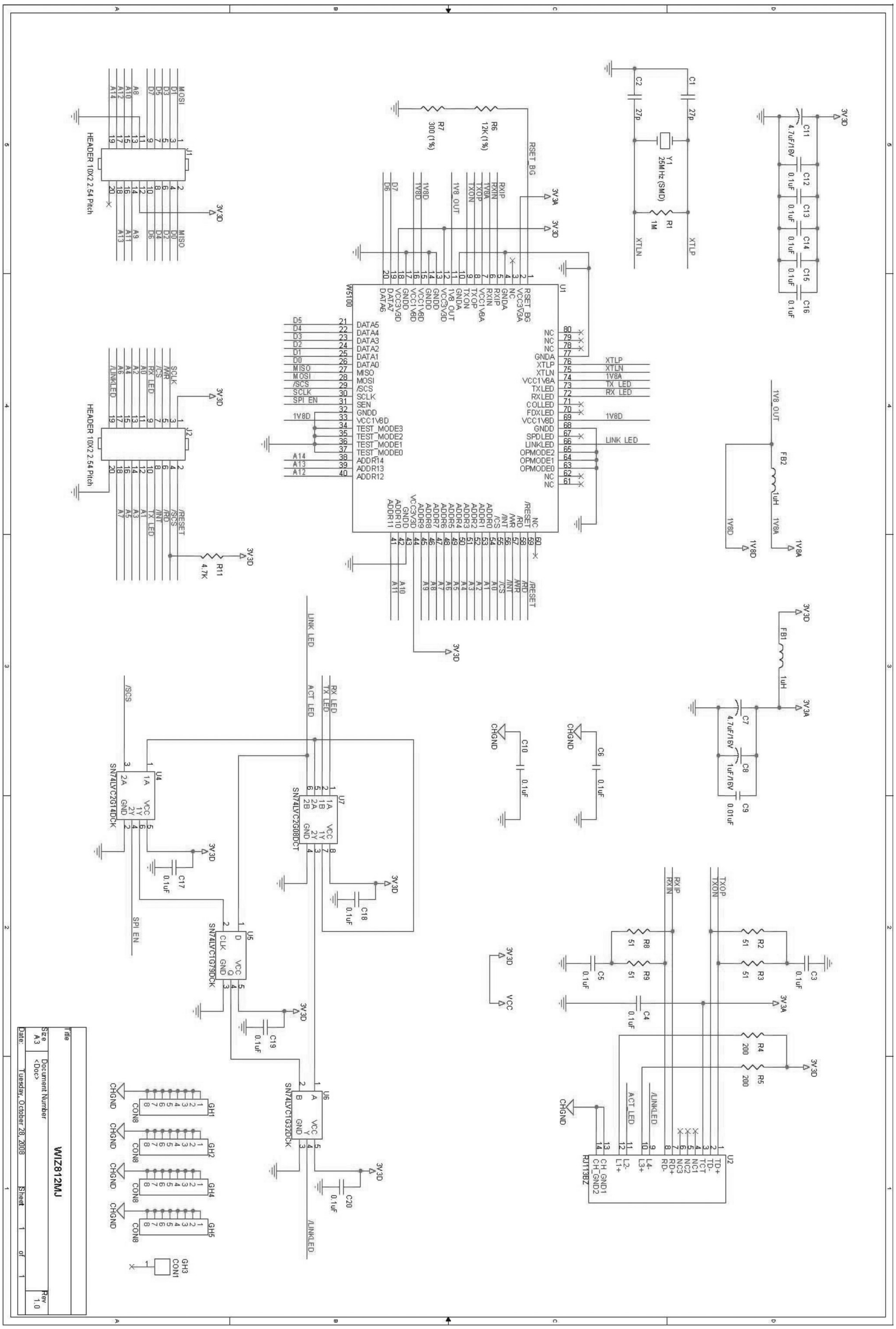
National Semiconductor
Europe Customer Support Center
Fax: +49 (0) 180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8790

National Semiconductor
Asia Pacific Customer
Support Center
Email: ap.support@nsc.com

National Semiconductor
Japan Customer Support Center
Fax: 81-3-5639-7507
Email: jpn.feedback@nsc.com
Tel: 81-3-5639-7560

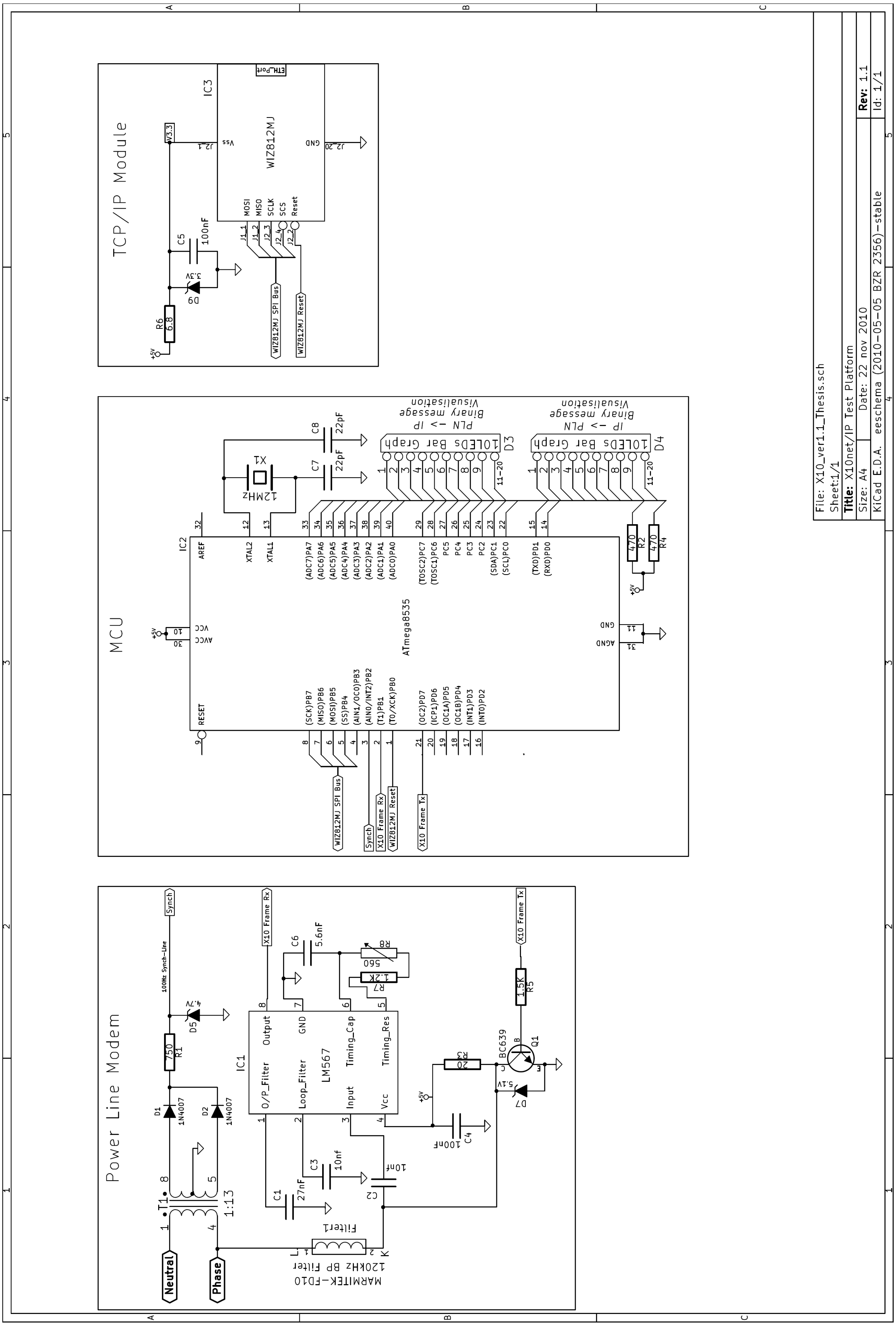
APPENDIX B

WIZ812MJ schematic diagram:



APPENDIX C

Test Platform schematic diagram:



TCP/IP Module
IC3
WIZ812MJ
ETH_Port
V3.3
Vss
J2_1
GND
J2_20
MOSI
MISO
SCLK
SCS
Reset
J1_1
J1_2
J2_3
J2_4
J2_2
C5
100nF
D9
3.3V
R6
6.8
+5V
WIZ812MJ SPI Bus
WIZ812MJ Reset
MCU
IC2
ATmega8535
AREF
XTAL2
XTAL1
12MHz
X1
C7
22pF
C8
22pF
(ADC7)PA7
(ADC6)PA6
(ADC5)PA5
(ADC4)PA4
(ADC3)PA3
(ADC2)PA2
(ADC1)PA1
(ADC0)PA0
(TOSC2)PC7
(TOSC1)PC6
PC5
PC4
PC3
PC2
(SDA)PC1
(SCL)PC0
(TXD)PD1
(RXD)PD0
VCC
AVCC
GND
AGND
RESET
(SCK)PB7
(MISO)PB6
(MOSI)PB5
(SS)PB4
(AIN1/OC0)PB3
(AIN0/INT2)PB2
(T1)PB1
(T0/XCK)PB0
(OC2)PD7
(ICP1)PD6
(OC1A)PD5
(OC1B)PD4
(INT1)PD3
(INT0)PD2
10LEDs Bar Graph
D3
D4
PLN -> IP
IP -> PLN
Binary message
Visualisation
R2
470
R4
470
WIZ812MJ SPI Bus
Synch
X10 Frame Rx
WIZ812MJ Reset
X10 Frame Tx
Power Line Modem
Neutral
Phase
T1
1:13
D1
1N4007
D2
1N4007
R1
750
D5
4.7V
100Hz Synch-Line
Synch
IC1
LM567
O/P_Filter
Loop_Filter
Input
Vcc
Output
GND
Timing_Cap
Timing_Res
X10 Frame Rx
C6
5.6nF
R8
560
R7
1.2K
C1
27nF
C3
10nf
C2
10nf
C4
100nF
R3
20
Q1
BC639
D7
5.1V
R5
1.5K
X10 Frame Tx
Filter1
120KHz BP Filter
MARMITEK-FD10
File: X10_ver1.1_Thesis.sch
Sheet:1/1
Title: X10net/IP Test Platform
Size: A4
Date: 22 nov 2010
Rev: 1.1
KiCad E.D.A. eeschema (2010-05-05 BZR 2356)-stable
Id: 1/1