Data transmission over power lines and applications

Jad Allam 200300105 jfa05@aub.edu.lb
Camille Eid 200300308 cae03@aub.edu.lb
Joseph Raad 200300301 jfr01@aub.edu.lb
Abstract

“Data transmission over power lines” a technology that sends data through existing electric cables alongside electrical current, is set to turn the largest existing network in the world, the electricity distribution grid, into a data transmission network. We study the applicability of data transmission over power lines (DTOPL) techniques to home automation. We conducted this study after noticing the lack of gathered information on the subject, and this caused by the relatively new research and development in this area. In our project, we will present both a broad overview and a technical reference on data transmission over power lines with a comprehensive presentation dealing with the different aspects of this technology. We designed a home automation system using DTOPL and in a first phase, run the performance analysis of the used power line technology, and secondly, build the system that best represents the real life application dealing with the security issues.
Acknowledgements

We wish to express our deep gratitude to Dr. Sami Karaki for his continuous support and ideas that helped persevere in our project.

We would like also to thank Mr. Francis Raveneau, business development manager in Ariane Controls Corporation, Quebec, Canada, for the technical support that he provided us with about the company’s products.

We would like also to thank the electrical and computer engineering department for funding a large part of our project.
Table of contents:

1 Introduction ...................................................................................................................................... 8  
1.1 General overview .................................................................................................................. 8  
1.2 Project specifications .......................................................................................................... 10  

2 Review of DTOPL technologies ............................................................................................ 12  
2.1 Applications of Data transmission over Power lines ...................................................... 12  
2.1.1 Low voltage or in-house ............................................................................................. 12  
2.1.2 Medium Voltage and Low Voltage ......................................................................... 13  
2.2 Comparison and evaluation of different protocols ......................................................... 14  
2.2.1 X-10 ......................................................................................................................... 14  
2.2.2 CeBus technology ...................................................................................................... 16  
2.2.3 LonWorks technology .............................................................................................. 18  
2.3 Standards and regulations ................................................................................................. 20  
2.3.1 CENELEC .................................................................................................................. 20  
2.3.2 FCC .......................................................................................................................... 21  
2.3.3 IEEE ......................................................................................................................... 21  
2.3.4 IEC ........................................................................................................................... 21  
2.4 Challenges in the power line home network .................................................................... 22  
2.4.1 Noise and disturbance characteristics ................................................................. 22  
2.4.2 Impedance and transfer function ............................................................................ 27  
2.4.3 Signal attenuation ...................................................................................................... 28  
2.4.4 Security Issues and Problems on the line ............................................................ 28  
2.5 Comparison with other technologies .............................................................................. 31  
2.5.1 X-10 ......................................................................................................................... 31  
2.5.2 CEBus ....................................................................................................................... 33  
2.5.3 LonWorks ................................................................................................................ 37  

3 Design of the system .................................................................................................................. 40  
3.1 Description of the project block diagrams ...................................................................... 40  
3.1.1 Choice of the chip and diagram of the system ....................................................... 40  
3.1.2 Ariane AC Transceiver ............................................................................................ 43  
3.1.3 Microcontroller ........................................................................................................ 47  
3.1.4 PC ............................................................................................................................. 48  
3.1.5 Relay ......................................................................................................................... 49  
3.1.6 Controlled system .................................................................................................... 50  
3.2 Project application .............................................................................................................. 51  
3.2.1 Performance analysis ............................................................................................... 51  
3.2.2 Encryption ................................................................................................................ 54  
3.3 Time and cost planning of Spring semester .................................................................... 56  
3.3.1 Time table division ................................................................................................. 56  
3.3.2 Task Description ..................................................................................................... 57  
3.3.3 Tentative projected cost ......................................................................................... 58  

4 Implementation of the design .................................................................................................. 59  
4.1 DTOPL Management Software ....................................................................................... 59  
4.1.1 Overview ................................................................................................................... 59  
4.1.2 Implementation of the software objectives ........................................................... 59  
4.1.3 Detailed description of the communication protocol ........................................... 63  
4.1.4 Snapshots from the software interface ................................................................... 70
List of figures:

Figure 1: Representation of the X-10 signal ................................................................. 15
Figure 2: Two packets of the X-10 protocol ................................................................. 16
Figure 3: Table of the different channels that support the LonTalk protocol .......... 19
Figure 4: Comparison table from Xilinx ...................................................................... 19
Figure 5: Measured amplitude and duration characteristics of noise [11] ............... 26
Figure 6: Measured impedance models of common electric apparatus [11] ............ 27
Figure 7: Simplified diagram of the final circuit ......................................................... 41
Figure 8: AC Transceiver diagram ............................................................................ 43
Figure 9: Coupling unit of the model [15] ................................................................. 44
Figure 10: Dataflow diagram .................................................................................... 46
Figure 11: Relay diagram ......................................................................................... 50
Figure 12: Column graph showing the reliability of Ariane [14] .............................. 52
Figure 13: Table describing the sources of noise in the Ariane test [14] ................. 53
Figure 14: Steps for the encryption process ............................................................. 55
Figure 15: Data encryption diagram ......................................................................... 55
Figure 16: Tasks planned for the Spring semester .................................................... 56
Figure 17: Time span for each task of the previous table ......................................... 56
Figure 18: Product list and Costs ............................................................................. 58
Figure 19: Flow diagram of the protocol implementation in the software ............. 61
Figure 20: Web Interface ......................................................................................... 63
Figure 21: General Packet Format .......................................................................... 64
Figure 22: Single Data Packet .................................................................................. 65
Figure 23: Multi-Packet Data Start .......................................................................... 65
Figure 24: Multi-Packet Data Continuation .............................................................. 66
Figure 25: Multi-Packet Data End ............................................................................ 66
Figure 26: Ping ......................................................................................................... 66
Figure 27: Ping Reply .............................................................................................. 66
Figure 28: Command ON ....................................................................................... 67
Figure 29: Command OFF ...................................................................................... 67
Figure 30: Advertise ON Power .............................................................................. 67
Figure 31: Advertise OFF Power ............................................................................ 67
Figure 32: Assign Address................................................................................................ 68
Figure 33: Advertise Address ........................................................................................... 68
Figure 34: Assign Keys..................................................................................................... 68
Figure 35: Advertise Keys ............................................................................................... 69
Figure 36: Microcontroller/Relays..................................................................................... 69
Figure 37: Graphical User Interface ............................................................................... 70
Figure 38: Example of a “single data” packet being transmitted first encrypted (Hello) and then in plain text (Goodbye) .............................................................................. 71
Figure 39: Example of a “multi-packet data” being transmitted
   (abcdefghijklmnopqrstuvwxyz) ................................................................................ 72
Figure 40: Example of home automation opening relays 2, 3 and 4.  ............................... 73
Figure 41: Example of ping/reply showing that a device is connected. ............................ 74
Figure 42: Picture of the controlling circuit...................................................................... 75
Figure 43: Table of possible cases for the controlling circuit........................................... 78
Figure 44: Diagram of the experiments procedure ........................................................... 79
Figure 45: Transmission phase from PC1......................................................................... 80
Figure 46: Reception phase on PC2.................................................................................. 81
Figure 47: Different load noise sources ............................................................................ 85
Figure 48: Type 2 experiment results ............................................................................... 86
Figure 49: Example of a form submission where R0R1R2R3=0101 ............................... 89
Figure 50: Diagram showing the AMR system .............................................................. 92
1 Introduction

For the past years, power lines have been used for the transmission of electricity; but nowadays with the emergence of modern networking technologies and the need for information spreading, data transmission over power lines has seen a really big growth. The technologies already used for spreading information such as telephone wiring, Ethernet cabling, fiber optic and wireless have each its limitations in cost and reliability. In this project we are going to discuss the advantages of using power lines as a communication medium and the wide range of applications that this technology can provide, in addition to the implementation of a home automation system using the data transmission over power lines.

1.1 General overview

Home automation is the solution for the future household. Today’s technology has evolved houses to be as smart as simulation games on PCs and PS2s. Smart houses can be literally programmed to execute pre-defined and real time instructions on a specific time of the day, the week, or even the month. The garden sparklers can turn on in the morning automatically; the music can be dimmed in all around the house to provide a comfortable environment for work, study, or even entertainment. Moreover, a smart house is a safe house. Safety can be considered both from the inside and the outside. Safety is first to make the house theoretically impossible to rob, and practically more difficult only. Different schemes could be adopted in order to attain this privilege. One could implement a safe perimeter by installing cameras, infrared sensors, or just
traditional safety equipment. All these instruments require electricity for operation. The second type of safety is inside the house. The most important issue is to monitor the living persons and animals inside the house, especially children. Baby monitors are nowadays the easiest way to keep an ear at the baby while performing everyday tasks. A pacemaker in the heart of an elderly person can suddenly turn off and lead to a heart attack without anyone near that person to hear his call for help. An automatic feeding system for cats could be installed inside the kitchen and will save the house residents from the burden of “remembering to feed the cat”!! All three of these equipments need electricity to function.

When using Data Transmission Over Power Lines (or DTOPL), any household can afford having a small scale smart house, and most importantly a safe one. No need to run new wires inside the walls and under the floor in order to get this network. Regular, standard power lines are the best alternative to be able to control, from a single computer, a set of equipments and tools by sending commands to them, either on a pre-defined schedule, or simply when getting a response from these instruments.
1.2 Project specifications

Our project consists of sending data and commands over regular power lines in order to implement home automation. First of all, we had to make a choice whether to build and implement a new protocol or use an existing one in order to achieve this task. Having seen the amount of alternatives, we finally decided that building our own would be reinventing the wheel and would not be as performant as one that has been tested, designed and commercialized by world wide companies. Hence, we shifted the largest weight of our project on the application and its benefits and on improving well-established protocols rather than starting from scratch. In order to achieve that, we had to review the available emerging technologies, compare them and choose the one that would suit our purposes the best. Having achieved the literature review, our choice fell upon Ariane Controls. In fact, among the candidate protocols, Ariane offered the fastest, the most reliable and the most flexible of them all.

Once the protocol choice has been made, we started designing our project, keeping several objectives in mind:

- Low design price
- Simplicity in design
- Efficiency and reliability
- General purpose (be adaptable to several applications, configurations and environments)
- Security
- Ergonomic and aesthetical design
The design mainly consists, at the transmitter side, of a central command unit (PC) that will provide an interface through which a user can issue orders that will be modulated by a power line modem. Then, the modulated message is transported over a power line used as a channel. At the receiver side, another modem will demodulate the message and send it to a local command unit that, based on the message, will issue orders to appliances and equipments. However, in our strive for improvement, we pushed the design further by adding several functionalities such as security, Web access, real time network discovery and real time status update.
2 Review of DTOPL technologies

2.1 Applications of Data transmission over Power lines

The applications of DTOPL are very wide and we can divide them into two categories: the Medium Voltage or access technology mainly used by the utility authority and which is behind the scope of our project, the Low Voltage or in home which cover the area of sending data over power lines within the consumer’s side and extends to all the electrical outlets within the home.

2.1.1 Low voltage or in-house

2.1.1.1 Home automation

Many years ago control of appliances in the home used to call for the establishment of new cable wiring in the home. With the DTOPL technology automation of a building can be done using power lines only. Hence we can control home appliances, light switches, wall outlets, thermostats, Heat Ventilation and Air Conditioning systems (HVAC), sensors, alarm and security. One of our project goals is to implement the home automation.

2.1.1.2 Street lights monitoring

The use of DTOPL to monitor street lights leads to big savings in the electricity bill of the government by introducing selective dimming or selective turn-off features. This application can increase energy savings by 25%.
2.1.3 Low cost inter-device peer-to-peer networking

Power lines may be used to create a network that links devices together on the power grid. Since such a network makes use of the existing infrastructure, installation time and cost are virtually non-existent. Also since every outlet or junction box becomes a point where a device may be connected, the device can be moved around numerous times. An example of such a network can be to replace the RS232 wiring required to set audio and video inputs on various systems in a house or a building.

2.1.2 Medium Voltage and Low Voltage

2.1.2.1 Utility

a) Automatic Meter Reading is a technology that uses the power line to send information to the utility directly. Meters can be linked to concentrators to allow suppliers to have remote access to each individual meter, to read or write information such as rates, prepaid amounts, current and cumulative counts, tampering detection, etc. Meters and/or concentrators can also be used along with AC Remote LCD devices to replicate and distribute their information to one or more points located anywhere on the electrical network.

b) Load shedding: this is done when we need to reduce power given to loads when we have peak demands hours. As an example incandescent lights, with the help of load control circuit will receive less power when the utility notices that the demand for electricity is at its peak in certain periods.
2.1.2.2 Broadband data transmission

This developing technology which is at its testing level (Italy, USA) nowadays enables broadband internet to be provided to your home using the electrical grid. This feature is behind the scope of our project however we will give a brief overview of what this technology is about:

An example of a company developing this technology is ABB Medium Voltage Power Solution [1]. This system certified for use up to 24 KV provides data transfer rates of up to 10 Mbps and hence challenging the xDSL and the broadband cable technology. The main problem in this technology is the connection between the MV and the LV grid which is done using optical fibers, copper pairs or wireless.

2.2 Comparison and evaluation of different protocols

Research and study in the technology domain supporting DTOPL led to the emergence of different protocols describing the modulation scheme used, the data transmission rate, the limitations, the drawbacks, the relative cost and the target applications.

2.2.1 X-10

It is the most ancient communication protocol used in home networking since 1978 [2] developed by X-10 US Corporation. It is used to allow compatible devices to communicate with each other over 110V AC wiring.
2.2.1.1 Protocol description

X-10 simply provides the technical specifications of how a device should place a signal onto the power line. The X-10 technology transmits binary data using the amplitude modulation technique. In order to differentiate the data symbols the carrier uses the zero-voltage crossing point of the 60 Hz on the negative or positive cycle.

Hence for synchronization, the presence of a 120 kHz signal burst at the zero crossing indicates the transmission of a binary one, whilst the absence of the 120 kHz signal indicates a binary zero. (fig.)

![Figure 1: Representation of the X-10 signal](image)

X-10 contains a detailed addressing scheme to prevent device clash. Devices contain two addresses - a house (dwelling) address, and then an individual device address. A typical X-10 transmission would include a start code, house address, device address, and then function code (such as ON, OFF, etc.). The X-10 system is limited in that it does not easily provide for two-way communications, and is very slow, although adequate for simple home automation tasks.

Every bit requires a full 60 Hertz cycle and thus the X-10 transmission rate is limited to only 60 bps. Usually a complete X-10 command consists of two packets with a 3-cycle
gap between each packet. Each packet contains two identical messages of 11 bits (or 11 cycles) each. A complete X-10 command consumes 47 cycles that yield a transmission time of about 0.8.

![Figure 2: Two packets of the X-10 protocol](image)

### 2.2.1.2 Disadvantages

The X-10 technology would not fit our project design for the main fact that it has limited potential in speed and intelligence terms. Its low data rate and undeveloped functionality permit to use the X-10 technology in limited applications. In addition to its unreliability of amplitude modulation and error correction, X-10 operates on 110V AC, which is a major drawback for its use.

### 2.2.2 CeBus technology

CEBus, or Consumer Electronics Bus [3], a standard proposed by the Electronic Industries Association, is based on the concept of Local Area Networks (LAN’s), for the home. CEBus based products consist mainly of two components: a transceiver which
implements spread spectrum technology along with a controller to run the protocol. The given protocol standards are for radio frequency, twisted pair, power line communication and a number of other home networking methods. The CEBus DTOPL standard specifies that a binary digit is represented by how long a frequency burst is applied to the channel. For example, a binary ‘1’ is represented by a 100 microsecond burst, whilst a binary ‘0’ is represented by a 200 microsecond burst. Consequently, the CEBus transmission rate varies with how many ‘0’ characters, and how many ‘1’ characters are transmitted. The CEBus standard specifies a language of object oriented controls including commands for volume up/down, temperature up one degree, etc. Due to the high noise level of power line channels, data should be transmitted via short frames, which is assured by the use of the spread spectrum technology. CEBus protocol uses a Carrier Sense Multiple Access/Collision Detection and Resolution (CSMA/CDCR) protocol to avoid data collisions. CEBus is a commercially owned protocol, and thus attracts registration fees.
2.2.3 LonWorks technology

This technology has been developed by Echelon Corporation [4]. It is essentially structured as an automatic control system that consists of sensors, actuators, application programs, communication networks, human-machine interface and network management tools.

LonWorks (Local Operation Networks) technology is an important new solution for control networks developed by Echelon® Corporation. A control network is any group of devices working in a peer-to-peer fashion to monitor the different components cited above. In some ways, a LONWORKS control network resembles LAN. It can control and link factory conveyor belts, product inventory, and distribution systems for optimum efficiency and flexibility. Smart office buildings can turn lights on and off, open and lock doors, start and stop elevators, and connect all functions to a central security system. In the same manner, homeowners can program a vast array of products and conveniences, from sprinkler systems to VCRs, with a touch tone phone from any remote location.

The LonTalk communications protocol is a layered, packet-based, serial peer-to-peer communications protocol. This protocol is designed for the requirements of control systems, rather than data processing systems. Also, this protocol is media-independent, which allows the system to communicate over any physical transport media. LonTalk has been approved as an open industry standard by the American National Standards Institute (ANSI)-EIA 709.1. [5]
Power lines are a possible medium that LonWorks devices could be attached to. The data rate will then be 5 kbps and, obviously, there is no limit to the number of devices that could be connected to the system.

Some real life applications of LonWorks are automated supermarket pricing, avionics instrument integration, circuit board diagnostics, electronic locks, intelligent industrial I/O irrigation, management, lighting control, power supply management, and research experiment monitoring.

<table>
<thead>
<tr>
<th>Channel Type</th>
<th>Medium</th>
<th>Data Rate</th>
<th>Max. Devices</th>
<th>Max. Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP/XF-1250</td>
<td>Twisted pair, bus</td>
<td>1.25 Mbps</td>
<td>64</td>
<td>125m (bus length)</td>
</tr>
<tr>
<td>TP/XF-78</td>
<td>Twisted pair, bus</td>
<td>78 kbps</td>
<td>64</td>
<td>1330m (bus length)</td>
</tr>
<tr>
<td>TP/FT-10</td>
<td>Twisted pair, flexible topology</td>
<td>78 kbps</td>
<td>64 (upto 128 if link-powered)</td>
<td>500m (node to node)</td>
</tr>
<tr>
<td>PL-20</td>
<td>Power line</td>
<td>5 kbps</td>
<td>No Limit</td>
<td>Determined by attenuation</td>
</tr>
</tbody>
</table>

**Figure 3: Table of the different channels that support the LonTalk protocol**

Power lines are a possible medium that LonWorks devices could be attached to. The data rate will then be 5 kbps and, obviously, there is no limit to the number of devices that could be connected to the system.

Some real life applications of LonWorks are automated supermarket pricing, avionics instrument integration, circuit board diagnostics, electronic locks, intelligent industrial I/O irrigation, management, lighting control, power supply management, and research experiment monitoring.
2.3 Standards and regulations

When using the powerline as a channel certain standards and regulations must be followed in order to avoid interference between the frequencies transmitted with any other frequencies already existing. Hence the bandwidth in the DTOPL environment is not limited by physical capabilities of the line. Rather, regulatory authorities in the developed countries limit the available bandwidth in order to prevent radio interference, other devices interference or military bandwidth interference.

We will discuss in the following section the limiting standards CENELEC, FCC, IEEE and IEC governing data transmission over power lines.

2.3.1 CENELEC

For Western Europe CENELEC’s standard EN50065 [6] “Low voltage mains signaling” gives regulations on key parameters such as frequency range, signal power and so on. The standard allows signals to operate in the frequency band 3-148.5kHz, avoiding interference with ripple control systems at the lower boundary, and interference with long wave (LW) and medium wave (MW) radio broadcasts by posting the upper boundary.

CENELEC then divide this band into further categories:

The band from 3-95 kHz, or A-Band, is allocated for electrical utility use, for such things as automated meter reading and customer load control. The range from 95-148.5 kHz, comprising the B, C, and D bands is reserved for end-user applications. These three bands are primarily differentiated by regulations in protocols for each band. B band, from 95-125 kHz requires no use of access protocol for establishing communications. This band is designed for use in applications such as baby-monitors and intercoms. The C
band from 125 to 140 KHz is mainly used for intra-building computer communication. Finally, the D band from 140 to 148.5 KHz is limited to energy customers' providers.

Lastly, EN50065 specifies such things as maximum signal attenuation allowed due to multiple, filter specifications for carrier removal, and specifies the maximum transmitted power for transmission over power lines should not exceed 500 mW.

2.3.2 FCC

For North America the Federal Communications Commission (FCC) [7] regulates transmitted power and bandwidth. The frequency range in this standard is from 100 to 450 KHz which is higher than the CENELEC. Moreover Part 15 of the American FCC’s rule allows transmission over power lines outside the AM frequency band (535 to 1705 KHz).

2.3.3 IEEE

The Institute of Electrical and Electronics Engineers [8] have published a set of recommendations and standards pertaining to the power line communication available at the reference: http://standards.ieee.org/catalog/olis/psystcomm.html.

2.3.4 IEC

The International Electrotechnical Commission (IEC) [9] has standardized the distribution line communications through technical committee #57 working group 9. In this standard, IEC TC57/WG9 uses frequencies below 150 KHz.
2.4 Challenges in the power line home network

Power lines are a hostile environment that makes the accurate propagation of communication signals difficult and are not designed for data transmission use. Noise levels are often excessive, and cable attenuation at the frequencies of interest is often very large. Important channel parameters such as impedance and attenuation are time varying in unpredictable ways. Following is a detailed description of the challenges encountered in the power line channel.

2.4.1 Noise and disturbance characteristics

Noise on electrical power networks on the Medium Voltage side is usually caused by corona discharge, lightning, power factor correction banks and circuit breaker operation. In our project we are dealing with a low voltage network, and hence much of this noise is filtered by medium/low voltage transformers, therefore the most common interference in low voltage domestic networks can be attributed to the various household devices and office equipment connected to the network. We can generally classify the disturbances into two categories and the noise into four categories.

2.4.1.1 Disturbances

Waveshape disturbances

These include:

- *Over and under voltages*, both persistent (>2 seconds) or surges (<2 seconds) which are of little effect on the network due to the robustness of the transceivers.

- *Outages* where there will be no transmission of information naturally.

- *Frequency variations* can cause major problems as many simple systems rely on the mains carrier (50 or 60Hz sine wave) for synchronization between transmitter and
receiver. Frequency variation in this wave will cause transmission error. Modern systems overcome this obstacle by avoiding reliance on the mains carrier for synchronization.

- **Harmonic distortions** are a major source of disturbance, yet these occur at frequencies below those designated in the standards regulating DTOPL.

**Superimposed disturbances**

These include:

a. Persistent oscillations, either coherent or random.

b. Transient disturbances, both impulse and damped oscillations.

On the medium voltage network, the superimposed disturbances are attributed to large factories with extensive plant or machinery, and industrial users with poorly filtered appliances. On the low voltage network, a number of household appliances are most often responsible for superimposed disturbances.

### 2.4.1.2 Noise

The four types of noise according to Vines [11] are:

A. Noise having line components synchronous with the power system frequency;

B. Noise with a smooth spectrum;

C. Single event impulse noise, and;

D. Non synchronous noise.
A. Noise having line components synchronous with power system frequency.

The usual source of this called type A noise are triacs or silicon controlled rectifiers (SCR’s), found domestically, for example, in light dimmers or photocopiers. The spectrum of this noise consists of a series of harmonics of the mains frequency (50Hz).

There are three ways to combat this kind of noise: [12]

• As the frequency spectrum of class A noise is regular, successful communication may be possible with modulation schemes that avoid, or have nulls, at these frequencies.

• Filter these noise components out using accurate notch filtering.

• Considering the time domain representation of class A noise, a noise pulse can be expected at equal intervals. Using fairly simple time division multiplexing schemes and error correction, unwanted effects can be minimized.

B) Noise with a smooth spectrum.

Type B noise with a smooth spectrum is generally caused by universal motors. A result of the commutation process in motor powered appliances such as blenders and vacuum cleaners, this noise has a flat spectrum in the frequency ranges used by DTOPL systems. Thus it can be modeled as band limited white noise.

A characteristic of many of the appliances that contain universal motors is that they are often used for a short period of time. Thus, we can avoid this noise by operating the DTOPL system when the noise is absent. Conversely, real time systems must be able to cope with type B noise.
**C) Single event impulse noise.**

Single event impulse noise (type C noise) is primarily due to switching phenomena—lightning, the closing of contacts, etc. Type C noise disturbs the whole frequency band for a short amount of time, and is often modeled as an impulse disturbance due to the relatively short times involved. Experience with impulse noise in other communications environments shows that type C noise can be overcome by error correcting codes.

**D) Non synchronous noise.**

Non synchronous noise (type D noise) is characterized by periodic components that occur at frequencies other than harmonics of the mains frequency. Major sources of type D noise include television and computer monitors. The scanning and synchronization signals in such appliances cause noise components at known frequencies— for example, interference from a PAL system television set is at 15734kHz and associated harmonics. Different standards of television and computer scanning have different radiated noise components. The solution to minimizing such interference is to avoid data transmission at 15734 kHz and associated harmonics, and to use a modulation scheme that is frequency diverse, thus avoiding potential type D noise at any unforeseen frequencies.
Following are quantitative values of noise for various common appliances used in any home:

<table>
<thead>
<tr>
<th>Electric Apparatus</th>
<th>Amplitude (mV)</th>
<th>Duration (μs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Single pulse)</td>
<td>Average Value</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>Electric Oven</td>
<td>329.2</td>
<td>431.2</td>
</tr>
<tr>
<td>Iron</td>
<td>369.3</td>
<td>583.8</td>
</tr>
<tr>
<td>(Periodic pulses)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Television monitor</td>
<td>197.2</td>
<td>311.3</td>
</tr>
<tr>
<td>Light dimmer</td>
<td>670.8</td>
<td>1199.3</td>
</tr>
<tr>
<td>(Continuous pulses)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vacuum cleaner</td>
<td>1457.5</td>
<td>2155.5</td>
</tr>
<tr>
<td>Dryer</td>
<td>87.9</td>
<td>119.7</td>
</tr>
</tbody>
</table>

**Figure 5: Measured amplitude and duration characteristics of noise [11]**

With an understanding of the noise inherent on domestic power networks, various suggestions can be made for the development of a PLC communications system:

- **Appropriate error correcting codes should be implemented to cope with noise types A, B and C.**
- **To avoid type D noise, television line frequency and harmonics should be avoided when modulating the signal onto the channel- no signal information should be transmitted at these frequencies.**
- **Some kind of frequency diversity (for example frequency hopping) should be implemented to cope with interference at unknown frequencies.**
2.4.2 Impedance and transfer function

The characteristic impedance of an *unloaded* power cable can be obtained by a standard distributed parameter model, and given by

\[ Z = \sqrt{\frac{R + j\omega L}{G + j\omega C}} \]

At the frequencies of interest for DTOPL, this approximates to

\[ Z = \sqrt{\frac{L}{C}} \]

where L and C are the line impedance and capacitance per length. Bergen and Vittal [13] list the characteristic impedance of the cables used for power transmission as ranging from 70-100 Ohms.

The power line channel impedance is a strongly fluctuating variable that is difficult to predict due to the number of appliances (loads) connected to the network.

<table>
<thead>
<tr>
<th>Apparatus</th>
<th>Impedance model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refrigerator</td>
<td>![Refrigerator Diagram]</td>
</tr>
<tr>
<td>Incandescent lamp</td>
<td>![Incandescent lamp Diagram]</td>
</tr>
<tr>
<td>Foot warmer</td>
<td>![Foot warmer Diagram]</td>
</tr>
</tbody>
</table>

**Figure 6: Measured impedance models of common electric apparatus [11]**

In summary the overall impedance of a low voltage network results from:
• Impedances of the devices connected to the network.
• Characteristic impedance of the cable used.
• Impedance of the distribution transformer.

Maximum power transfer theory states that the transmitter and channel impedance must be matched for maximum power transfer. With strongly varying channel impedance, this is very difficult. The system designer must suffice with designing a transmitter and receiver with sufficiently low output/input impedance (respectively) to approximately match channel impedance in the majority of expected situations.

2.4.3 Signal attenuation

The signal attenuation for low voltage networks is around 100 dB/Km which leads to a need for repeaters to be plugged in at desirables weak areas. The main factors leading to signal attenuation are:

- Time dependence.
- Frequency dependence.
- Distance dependence.
- Difference between phases

2.4.4 Security Issues and Problems on the line

2.4.4.1 Problem

Data transmission over power lines happens over a public/shared medium (the power line) thus, the information that is being exchanged is available anywhere in that medium within a certain distance and to everyone. This is similar to networking that also works in
a promiscuous mode. From that, we can logically deduce that this type of communication can be under threat of hacking especially from eavesdropping and also this can be a source of confusion and problems on the network. In fact, the data transmission over the power line is valid up to a certain distance and within that distance, the data is available at any point and to anyone be it someone in the house or someone in another neighboring house.

2.4.4.2 Examples

First of all, eavesdropping (listening on the line) is possible since as we said the data is available at any point within the distance and in Ariane the data is not encrypted. So, if a neighbor B on the network has a PLM-1 system he can listen to the transmission that is being sent in the house A and if there is sensitive information being transferred it can be compromised. Also, in another case, a potential high-tech thief can eavesdrop on the house from a plug in the garden for example and get the codes to disarm the alarm. Next, the fact that the data is transmitted to anyone and anywhere within the distance, suppose a command to turn off all the lights is given in house A, if the addressing is not well implemented, all the lights in neighbor B’s house will also go off. Finally, if the neighbors are using different protocols on their separate home networks, their might be conflicts, interference and cancellations because some protocols work on the same ranges and some work on very large ranges.

2.4.4.3 Solution

The solutions can be numerous and the ones we came up with were: we might remove any plug found outside the house to disallow outside access but this creates
inconvenience if for example we want to use the lawn mower we’ll have to add an extension from inside the house. Another solution would we to install a frequency filter that would filter out the high pass frequencies at the point where house A and house B’s circuits meet. However, these 2 solutions include additional costs: new equipments and professional labor. Thus, we thought that the best and cheapest solution would be using encryption and we opted for the RSA algorithm. In fact, with encryption, we didn’t care if the neighbor or the thief would get the encrypted data as long as the key was large since that would require a huge amount of time, computing power and resources to be cracked.
2.5 Comparison with other technologies

2.5.1 X-10

*Modulation:*
The PLM-1 chip uses Frequency Shift Keying (FSK) modulation. FSK systems always modulate signals, offering a much better signal to noise ratio. X-10 uses ASK modulation which is noise sensitive: when a silence is transmitted, any in-band noise or tone, even at small amplitudes, can transform a 0 symbol into a 1.

*Narrow-band:*
PLM-1 uses high efficiency filters to transmit and receive signals. These filters are highly selective, eliminating unwanted noise and signals at the receiver input and filtering out unwanted harmonics at the transmitter amplifier. X-10 isn't a narrow band filter. It uses a simple, second-order LC filter, tuned at 120 kHz. X-10 generates a lot of harmonics while transmitting.

*Carrier:*
PLM-1 has a programmable carrier that can be programmed on the 50 kHz to 500 kHz range while X-10 has a fixed carrier at about 120 kHz.
**Bit-rate:**

PLM-1 has a programmable bit rate ranging from 0.100 to 30 kbps. Typical applications use Baud rates on the 1 to 10 kbps range. The X-10 Baud rate is power line synchronous: it is fixed at 0.050 or 0.060 kbps.

**Medium access:**

The PLM-1 chip offers an efficient Medium Access Control logic. There is no Medium Access Control with X-10. If two nodes transmit simultaneously, even in independent dwellings connected to a common power line, collisions are not detected and messages are lost.

**Priority:**

The PLM-1 chip manages packet priority and queuing while there is no priority management and no queuing function in the X-10 protocol.

**Error handling:**

PLM-1 corrects up to two consecutive erroneous bits per transmitted data byte using Forward Error Correction (or FEC). If more than two bits per byte are erroneous, the chip implements the CRC-16 error detection algorithm and thus avoids interpreting erroneous packets. Neither error correction, nor CRC error detection are present in X-10.
**Bi-directional:**

The PLM-1 technology is fully bidirectional, but also accepts unidirectional receive-only nodes on the network. X-10 offers very limited bi-directional capabilities.

**Transaction management:**

The PLM-1 chip manages UNACK and ACK/IACK transactions whereas X-10 offers no transaction management.

**Timer:**

The PLM-1 chip offers a timer function that X-10 does not support.

**Data encoding:**

PLM-1 encodes data at the nibble level. X-10 encodes data at the bit level. A 1 bit is represented by a one symbol (a 120 kHz carrier burst) followed by a zero symbol (a silence); a 0 bit is represented by a zero symbol followed by a one symbol. Each symbol is transmitted for about 1 ms at the zero-crossing of the power line frequency and is repeated twice, 120 degrees apart from the power line.

---

**2.5.2 CEBus**

**Narrow-band:**

The PLM-1 is a narrowband FSK-modulated system. Millions of current products use FSK. PLM-1 uses a high-efficiency filter. This filter offers high noise rejection, typically better than 20 dB at a few kilohertz outside its operating frequency. If a high-power tone were to enter directly into the operating frequency of any given system, it would impede its functioning. Wideband systems such as CEBus are vulnerable to most noise sources.
existing on the PLC medium while a narrowband system such as the PLM-1 filters out most of that noise.

CEBus is a swept-carrier system, vulnerable to any noise in the 100 to 400 kHz range. CEBus is not a true Spread Spectrum system, because it does not process incoming signal according to Spread Spectrum theory and consequently does not show the processing gain normally associated with real Spread Spectrum systems. For example, a true Spread Spectrum system (100-400 kHz carrier, 10 kHz modulation rate) would offer about -15 dB Signal to Noise ratio, i.e., noise 5 times the signal. CEBus is specified at 3 dB, i.e., noise only 0.7 times that of the signal. For noise under 100 kHz or over 400 kHz, a CEBus system typically uses a simple second-order band-pass filter, that leads to poor protection against noise outside its main band.

*Carrier:*

The PLM-1 carrier frequency is programmable on the 50 to 500 kHz range, allowing different PLC networks to share the same power-line medium, a method called "channeling." For example, a 50 kHz PLM-1 system may be used on 25 kV distribution power lines, while a 262 kHz PLM-1 system is used on 120/240 V distribution power lines. By using various carrier frequencies, the narrowband PLM-1 system shares the power-line medium with other systems at various frequencies.

CEBus uses only one carrier type, a swept carrier in the 100 to 400 kHz band, and exclusively occupies almost all the available bandwidth in the 10 to 500 kHz band. In most situations, CEBus is unable to share the power-line medium with other common PLC technologies. CEBus does not tolerate their presence well, while properly designed narrowband systems can work alongside CEBus and suffer little interference, if at all.
**Bit rate:**

You can program the PLM-1 to operate at Baud rates ranging from 0.100 to 30 kbps. There are mathematical relationships between Baud rates and noise tolerance. You can select the exact combination for your specific applications. For Home Automation applications, typical Baud rates range from 1 to 10 kbps.

CEBus speed is fixed and limited at 10,000 symbols per second. Because of the technique used to encode data, the effective CEBus Baud rate comes down to about 6.67 kbps. Typical RF or Twisted Pair (TP) communication technologies offer transmission rates of 1 Mbits per second or more. The CEBus standard specifies 10k symbols, or about 6.67 kbps, on RF, TP, PL or coax channels.

**Error correction:**

The PLM-1 implements a high efficiency Forward Error Correction (FEC) algorithm. Before transmitting, the chip automatically adds redundant data so that the receiver can detect and correct bits received erroneously. The FEC algorithm corrects up to two consecutive erroneous bits per transmitted data byte. If more than two bits per byte are erroneous, the chip implements the CRC-16 error detection algorithm and thus avoids interpreting erroneous packets. To further increase the probability of successful transmission, applications typically repeat packet transmission two or three times. Long lines or high attenuation cases may use repeaters.

CEBus does not implement any forward error-correction algorithm. If a single bit (out of a typical 120-bit packet) is erroneous, the whole CEBus packet is lost. Many noise sources in residential environments (typically 50/60 Hz) can destroy one or two bits
every half cycle. Noise generating devices typically include halogen lamps, computers, compact fluorescent lights, etc. Attempting to solve the problem by repeating the packets is totally inefficient against cyclic noise. In the face of such error bursts, only an error-correction algorithm can deliver the unaltered data packets. CEBus does not implement any forward error-correction algorithm and only offers a CRC-16 error-detection algorithm.

Protocol:
The PLM-1 chip is protocol-neutral. It is a general data transportation mean. You can use several protocols simultaneously on a single PLM-1 network. A PLM-1 system can, on a single power line or network, transport data from up to 43 standard protocols and up to 768 user-defined protocols. The PLM-1 chip is a general transceiver, implementing high-level network functions such as: CSMA/CD medium access, data encoding-decoding, error correction and error detection, packet priority management, packet queueing, UNACK or ACK/IACK transactions, etc. No single protocol has ever provided a universal solution for all applications. Electrical outlets or wall switches cannot, in an economical fashion, implement the intricacies of the high level protocols required for complex applications. Conversely, a simple protocol with a few commands (used to economically implement simple control functions) cannot be used in complex applications. The PLM-1 system allows systems using different protocols to not only share a common medium, but also to "inter-operate."

CEBus is a single protocol in and of itself. A CEBus network cannot operate in a multi-protocol environment. The CEBus protocol does have some provisions to transport
foreign data packets within the CEBus protocol packet, but CEBus does not identify the protocol in use.

### 2.5.3 LonWorks

**Narrow-Band:**

Thanks to its high efficiency filtering, PLM-1 offers high selectivity. Moreover, this filter is a rugged, passive component and allows the receiver front-end to sustain the high-energy noise pulses found in the power line environment.

LonWorks offers different transceivers, some using narrowband signaling. However they do not offer the high selectivity found in PLM-1 systems. Their selectivity is compliant with that specified under CENELEC standards.

**CPU design:**

The PLM-1 design is based on elementary gates and flip-flops, without any ROM, RAM, nor microprocessor core. The PLM-1 contains about 22,000 gates. A gate-level design offers very high-level performance, but its development cost is accordingly high. However, a 22,000 gate design (a relatively low gate-count for today's ASICs) is low cost. For applications requiring a microcontroller, users may select the exact microcontroller they need, with the combination of A/D, D/A, display drivers, ROM/RAM space, etc. as needed. For volume production, the PLM-1 can be integrated onto almost any microcontroller you selected.

In LonWorks, a typical PL node includes one transceiver module (for signal processing) and one Neuron chip. The Neuron chip itself contains three CPUs. The Neuron chip does not include functions like D/A converters, LCD display drivers, etc. Applications using
such building blocks may require that a fourth processor be added to the design. We can say that this design is microprocessor intensive.

**Bit rate:**
As seen before, the PLM-1 chip has both programmable frequency and Baud rate, while LonWorks, uses fixed frequency and bit rate.

**Timer:**
LonWorks offers no timer like the PLM-1 chip does.

**Error correction:**
The PLM-1 implements a high efficiency Forward Error Correction (FEC) algorithm. Before transmitting, the chip automatically adds redundant data so that the receiver can detect and correct bits received erroneously. The FEC algorithm corrects up to two consecutive erroneous bits per transmitted data byte. If more than two bits per byte are erroneous, the chip implements the CRC-16 error detection algorithm and thus avoids interpreting erroneous packets. To further increase the probability of successful transmission, applications typically repeat packet transmission two or three times. Long lines or high attenuation cases may use repeaters.

The ANSI/EIA 709.2 standard, provides a CRC-based error detection algorithm, but does not show any Forward Error Correction algorithm. ANSI/EIA 709.2 compliant nodes do not implement a Forward Error Correction algorithm.
**Protocol:**

PLM-1 is a data transportation mean that is open to evolution, can support any protocol, and supports many different protocols simultaneously on a single power line network.

The LonWorks Control Network Protocol Specification ANSI/EIA 709.1 has provision to encapsulate a foreign packet within one of its own; although it supports 4 protocol versions, it is essentially a single protocol system. It does not identify the protocol in use.

**Cost:**

Thanks to its gate level design at the CPU level, PLM-1 is low cost. Typical applications may use an external, low-performance micro-controller, with the required functions (A/D converters, LCD driver, or others, according to the application requirements). Since the PLM-1 technology resolves all tedious and complex networking issues, the micro-controller does not have to offer high performance and can easily implement the higher levels of the application. The total cost of a PLM-1 node is low. For volume production, the PLM-1 can be integrated into most microcontrollers ICs.

Receive only nodes may exist on a PLM-1 network, for less demanding applications (e.g. lighting control dimmers). These nodes are very low-cost.

In LonWorks, Typical Power Line nodes comprise one transceiver and one Neuron chip, along with the Power Line coupling components. If ROM/RAM space offered by the Neuron is not adequate, or if the application requires some specific functions that are not available in the Neuron*, a supplemental micro-controller (or external devices) could be required.
3 Design of the system

3.1 Description of the project block diagrams

3.1.1 Choice of the chip and diagram of the system

3.1.1.1 Choice of Ariane PLM-1 chip

In the previous section, we drew the comparison between the Ariane PLM-1 chip and the other different protocols available in the market: X-10, CEBus and LonWorks. We decided to work with the PLM-1 chip for several main reasons. First, the PLM-1 chip uses high efficiency filters to transmit and receive signals while X-10 isn't a narrow band filter and uses ASK modulation which is noise sensitive, CEBus is a swept-carrier system, vulnerable to any noise in the 100 to 400 kHz range and LonWorks do not offer the high selectivity found in PLM-1 systems. Second, typical applications use Baud rates on the 1 to 10 kbps range. PLM-1 has a programmable bit rate ranging from 0.100 to 30 kbps while X-10 Baud rate is fixed at 0.050 or 0.060 kbps, CEBus Baud rate comes down to about 6.67 kbps and LonWorks uses fixed frequency and bit rate. Third, PLM-1 corrects data using Forward Error Correction (or FEC) and detects errors using CRC-16 error detection algorithm, whereas neither error correction, nor CRC error detection are present in X-10 and both CEBus and LonWorks do not implement any FEC algorithm and only offer a CRC-16 error-detection algorithm. Fourth, you can use several protocols simultaneously on a single PLM-1 network while a CEBus network cannot operate in a multi-protocol environment and LonWorks supports 4 protocol versions but does not identify the protocol in use. Finally, the PLM-1 chip is low cost and can be integrated
into most microcontrollers ICs whilst LonWorks may require a supplemental microcontroller, which increases its cost significantly.

3.1.1.2 Diagram

Following is a diagram of the entire circuit. No details are shown for simplicity. The PC is the main controller of the entire system.

Figure 7: Simplified diagram of the final circuit
3.1.1.3 Diagram Description

1- Command given by user to turn on the Air Condition through the internet using Web interface program developed.

2- Command arrives at the Host System which is my PC installed at home.

3- The software developed in Delphi and installed on the home PC has the following tasks to achieve:
   - Analyse the command given through the internet and put it in a form that can be readable by the Ariane transmitter module.
   - Encrypt the command through the RSA algorithm in order to ensure that the data that will be transmitted via power lines can’t be read by any other module than the Air Condition system.
   - Send the encrypted command to the Ariane RS-232 transmitter module which is linked to the computer through the RS-232 serial port.
   - The Ariane RS-232 transmitter module receives the data and modulates it in order to transmit it over the power lines at the 262 KHz frequency.
   - The Ariane RS-232 receiver module receives the data from the power line and demodulates it in order to send it to the PIC 16F84 microcontroller.
   - The PIC 16F84 analyses the command, decrypts it through the assembly language and then sends it respectively to the corresponding relay.
   - The relay will send the command to the appropriate controlled system which is in our case the Air Condition system.
3.1.2 Ariane AC Transceiver

Figure 8: AC Transceiver diagram

3.1.2.1 Description

The AC transceiver [14] is a complete system that is made up of several modules:

- AC-PLM-1 Modem: implements the modulation/demodulation schemes using the FSK technique. Also, it implements several known error detecting and correcting schemes like Cyclic Redundancy Check (CRC-16) and Forward Error Correction (FEC). On the other hand, it also implements several networking features such as CSMA/CD collision detection and resolution, packeting, packets priorities and packet queuing and Media Access Control (MAC) addressing that can be disabled and implemented by a host (PC or microcontroller for better control).
• External Oscillator: used for synchronization purposes and is directly connected to the AC-PLM-1 clock generator.

• Analog Front End: it handles the analog signal that's coming out of the modulator

• Interface: enables it to be connected to a host (PC or microcontroller) via a parallel, serial or USB cable.

• Power Supply: as its name suggests, supplies power to the AC-PLM-1, the interface and the Analog Front End.

• Coupling Unit [15]

![Coupling unit of the model][15]

The coupling unit has to effectively couple the carrier signal with the power lines. When transmitting, it inserts the carrier signal on the power line, and while receiving, it removes the signal from the power line. Also, the coupling unit provides isolation and protection of the low voltage circuit from the high-voltage surges that can occur on the power line. It comprises two parts: the coupling capacitor and the isolation transformer.

a) Coupling capacitor

The coupling capacitor has to block the large A.C. mains voltage, while passing the high-frequency communication signals. The value of the capacitor is chosen
so that its impedance at the communication frequency is low, while its impedance at the power line frequency (50Hz or 60Hz) is high.

b) Isolation transformer

In order to provide isolation between the low-voltage section and any hazardous part of the system, a high-frequency transformer is included in the coupling unit. And, the transformer can act as an amplifier for the signal leading to a better performance.

3.1.2.2 Operation

- Data Input: Data is fed as bits to the system from a Host (PC or Microcontroller).
- Data Encapsulation (on Send): Once the Data Input has been achieved, the data is encapsulated in packets along with addresses and error correction.
- Modulation (on Send): Using FSK the packets are modulated and sent on the line as an analog signal.
- Demodulation (on Receive): Once the signal arrives on the line, it is demodulated and transformed back into bits.
- Data Retrieval (on Receive): Once the Data Input has been achieved, the address is checked to make sure that this chip is the correct destination and once this is done the data is retrieved from the packets and it is checked for errors and corrected if need arises.
- Data Output (on Receive): once the data has been retrieved, it can now be sent as bits to the Host (PC or Microcontroller)
Figure 10: Dataflow diagram
3.1.3 Microcontroller

3.1.3.1 Description

The microcontroller is made up of two main components:

- Processor
- Memory

3.1.3.2 Operation

- Data I/O: the microcontroller is responsible for data inputs and outputs. In fact, it is used to send data to the transceiver whenever we want to send a command on the line and receive data from the transceiver whenever we want to receive a command.

- Addressing: If we disable the MAC addressing from the transceiver, it can be implemented in the Microprocessor.

- Computations and Decision Making: Having the ability to do calculations, logic and store data, the Microprocessor is used to make some decisions i.e. whenever the temperature is above 25 turn on the AC.

- Encryption: In an effort to improve security, the Microprocessor will be used to implement RSA encryption algorithm. (Each Microprocessor will be assigned a public key and a private key)

- Additional features: The flexibility of the Microprocessor makes it ideal for the implementation of any additional feature such as a simple ping that we’ll use to know the status of the device that the microcontroller commands.
3.1.4 PC

3.1.4.1 Description

For our purposes, the PC will be mainly used as a microcontroller. However, we’ll consider it as a “super-microcontroller” and we’ll use it as our main command and control unit and as server.

3.1.4.2 Operation

The PC will enable the user to take control of the whole system via a USB connection to the AC Transceiver and allow him to perform the following tasks through a custom made interface:

- Controlling devices on the network: giving ON/OFF commands …

- Scheduling devices operation: This will allow the user to enter a certain schedule for the different components that are on the line and specifying the time each one should go on or off. Example: The user sets the time at which the external lights should be turned on and when they should go off.

- Conditional operation: This will allow the user to enter a certain logical expression based on conditions for example if a component is on a second one should be off.

- Holding information about the network (what’s connected on the line): This is very important so that the PC will know at any time what is connected and what is not to prevent errors and giving commands to non-existing devices. Also, this should allow the network to be updated whenever there is a new device on the line. This is like the Networking Address table and should be possible through the use of a custom made ping function.
• Acting as a web server for the web interface and several other operations: The PC holds a website that the user can access through the internet from anywhere in the world and gain control of his home network. Example: the user wants to take a hot shower as soon as he gets back from work so he accesses the website from work and gives an on command to a water heater before he leaves work.

• Allowing the transmission of encrypted data: The PC will implement the RSA encryption algorithm in order to allow secure data transmission over the public medium (power line).

3.1.5 Relay

3.1.5.1 Description

The relay is a very simple device that acts as a low voltage operated switch for high currents i.e. it needs a small voltage to be operated and it allows a high current to pass through it.

3.1.5.2 Operation

For our purposes, the relay acts as a digitally operated switch that allows analog currents through it. In fact, it is commanded by the Host (PC or Microcontroller) receiving a 1 (5V) or a 0 (0V) to switch it on or off. When it is switched on it’ll allow the current to pass through it and when it’s off it’ll block it. Example: A lamp (Load) is connected to the power line (Power source) through a relay. The relay is initially off and no current passes it and goes to the load thus the Lamp is off. The user decides to turn on the light so he gives an on command to the Lamp which in reality means that the host will transmit a 1 to the relay and turn it on allowing the current to pass and the Lamp to go on.
3.1.6 Controlled system

3.1.6.1 Description

It is any electric appliance that will be controlled by the Host (PC or Microcontroller) through the relays. In our FYP we used fans (Air Condition) as the controlled system.

3.1.6.2 Operation

Most if not all of these controlled systems will be easy to manipulate just requiring on/off commands. The fans use 12 V in order to operate.
3.2 Project application

3.2.1 Performance analysis

3.2.1.1 Claims

Most communications systems that are widely in use today have a dedicated medium such as twisted pairs cable, coaxial cable, fiber optic cable… Data transmission over power lines don’t and this is the major drawback and advantage of using power lines as a medium. In fact, the power lines may be used to power several equipments at different ratings and different times thus making them subject to noise levels that can be very high, very variable and in no way predictable. Ariane controls implemented several ways to make sure that the data being transmitted would be delivered reliably by implementing Cyclic Redundancy Check (CRC) and Forward Error Correction (FEC). On its website and in its datasheets Ariane control provides data about the performance (speed and reliability) of its PLM-1 chip. The following is the data about reliability:

- Data about Speed:
  
  The PLM-1 can be programmed to operate at Baud rates ranging from 0.100 to 30 kbps, however there are mathematical relationships between Baud rates and noise tolerance the exact combination can be selected according to the specific applications. For Home Automation applications, typical Baud rates range from 1 to 10 kbps.
3.2.1.2 Experimental Method

The robustness of protocols was quantified by the level of communications success in the presence of noise in the electric network. Now that we have seen the claims made by Ariane, we have checked them to see how accurate they are and how they are affected in various setups.

In fact, the performance analysis experimental approach is presented in section ---- of the report.
### Sources of noise

<table>
<thead>
<tr>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 lamps+0.22µF</td>
<td>Two 50W halogen lamps without magnetic transformers connected in parallel. A 0.22µF capacitor is connected at a fixed point in the set-up.</td>
</tr>
<tr>
<td>LampNum. 1</td>
<td>50W halogen lamp without magnetic transformer.</td>
</tr>
<tr>
<td>Lamp Num. 2</td>
<td>50W halogen lamp without magnetic transformer.</td>
</tr>
<tr>
<td>Intercom A</td>
<td>200kHz Frequency modulation Intercom over power line carrier.</td>
</tr>
<tr>
<td>Intercom B</td>
<td>230kHz Frequency modulation Intercom over power line carrier.</td>
</tr>
<tr>
<td>Intercom C</td>
<td>260kHz Frequency modulation Intercom over power line carrier.</td>
</tr>
<tr>
<td>2 computers</td>
<td>Two 486 type personal computers connected in parallel, no screens.</td>
</tr>
<tr>
<td>Computer ASI</td>
<td>486 type personal computer, no screen.</td>
</tr>
<tr>
<td>X-10+ alarm clock</td>
<td>Alarm Clock continuously transmitting the command X-10 “DIM” on the 120kHz carrier.</td>
</tr>
<tr>
<td>Alarm Clock</td>
<td>Alarm Clock with electronic circuitry and X-10 captor.</td>
</tr>
</tbody>
</table>

Figure 13: Table describing the sources of noise in the Ariane test [14]
3.2.2 Encryption

RSA algorithm [16] is a public key encryption algorithm. In RSA algorithm, encryption and decryption are of following form, for some plain text \( M \) and cipher text \( C \):

\[
C = M^e \mod n
\]

\[
M = C^d \mod n
\]

In fact, both the sender and the receiver must know the public key \( n \). Also, the sender knows the value of \( e \) whereas only the receiver knows the value of the secret key \( d \). Thus, this is a public-key encryption algorithm with a public key of Public Key = \( \{e, n\} \) and private key of Private Key = \( \{d, n\} \). For the algorithm to work the following should be true:

1. It is possible to find values of \( e, d, n \) such that \( M^{ed} = M \mod n \) for all \( M < n \).
2. It is relatively easy to calculate \( M^e \) and \( C^d \) for all values of \( M < n \).
3. It is infeasible to determine \( d \) given \( e \) and \( n \).
The key generation process is as follows

1. Select 2 prime numbers p and q
2. Calculate n = p x q
3. Calculate $\Phi(n) = (p-1) \times (q-1)$
4. Select integer e such that $\gcd(\Phi(n), e) = 1$; $1 < e < \Phi(n)$; e and $\Phi(n)$ are relative prime
5. Calculate d such that $e \cdot d = 1 \mod \Phi(n)$
6. Public key = {e,n}
7. Private key = {d,n}

<table>
<thead>
<tr>
<th>Encryption process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plaintext M&lt;N</td>
</tr>
<tr>
<td>Ciphertext C = M^e (mod n)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decryption process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plaintext C</td>
</tr>
<tr>
<td>Ciphertext M = C^d (mod n)</td>
</tr>
</tbody>
</table>

Figure 14: Steps for the encryption process

Figure 15: Data encryption diagram
3.3 Time and cost planning of Spring semester

3.3.1 Time table division

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Software interface</td>
<td>30 days</td>
<td>Mon 2/6/06</td>
<td>Tue 3/7/06</td>
</tr>
<tr>
<td>2 Web interface</td>
<td>15 days</td>
<td>Tue 2/14/06</td>
<td>Tue 2/28/06</td>
</tr>
<tr>
<td>3 Product specification paper</td>
<td>10 days</td>
<td>Wed 3/1/06</td>
<td>Fri 3/10/06</td>
</tr>
<tr>
<td>4 Equipment installation</td>
<td>10 days</td>
<td>Wed 3/15/06</td>
<td>Fri 3/24/06</td>
</tr>
<tr>
<td>5 Performance analysis</td>
<td>9 days</td>
<td>Mon 3/27/06</td>
<td>Tue 4/4/06</td>
</tr>
<tr>
<td>6 Special features paper</td>
<td>7 days</td>
<td>Thu 4/6/06</td>
<td>Wed 4/12/06</td>
</tr>
<tr>
<td>7 Final report</td>
<td>10 days</td>
<td>Sat 4/15/06</td>
<td>Mon 4/24/06</td>
</tr>
<tr>
<td>8 Poster</td>
<td>5 days</td>
<td>Mon 5/1/06</td>
<td>Fri 5/5/06</td>
</tr>
<tr>
<td>9 Final presentation</td>
<td>6 days</td>
<td>Mon 5/15/06</td>
<td>Sat 5/20/06</td>
</tr>
</tbody>
</table>

Figure 16: Tasks planned for the Spring semester

Figure 17: Time span for each task of the previous table
3.3.2 Task Description

a) Software interface: this consists of building custom software that would form the command center for the system. It will handle the communication between the PC and the transceiver and will allow the user to send commands in real time or in pre-defined time. In fact, it will allow ON/OFF switching, scheduling, network discovery (the connected devices and their status), conditional operation and data encryption.

b) Web interface: this consists of building a Web site that would be hosted on the PC that is connected to the power line network. This interface is similar to the software one in the sense that it provides a command center, however in that case commands can be issued from anywhere in the world where an Internet connection is available.

c) Product specification paper: we will finalize our project specification taking into consideration the hurdles that we will be confronted with and any optimization that might come along the way.

d) Equipment installation: we will implement several configurations to be able to achieve several experiments and put to the test our design.

e) Performance analysis: as mentioned above, we will measure several performance criteria (BER, Baud rate …) using different configurations in order to verify the data provided in the literature.
f) Special features: in order to achieve all our planned tasks, there are several features that we need to design.

### 3.3.3 Tentative projected cost

<table>
<thead>
<tr>
<th>Product</th>
<th>Quantity</th>
<th>Unit Cost</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC-MIO RS-232Powerline Transceiver</td>
<td>2</td>
<td>200$</td>
<td>400$</td>
</tr>
<tr>
<td>Microcontroller PIC16F84</td>
<td>2</td>
<td>1.3$</td>
<td>2.6$</td>
</tr>
<tr>
<td>Bread Board</td>
<td>2</td>
<td>4$</td>
<td>8$</td>
</tr>
<tr>
<td>Relay</td>
<td>4</td>
<td>0.33$</td>
<td>1.33$</td>
</tr>
<tr>
<td>DB-9 female connector</td>
<td>1</td>
<td>1$</td>
<td>1$</td>
</tr>
<tr>
<td>Fan</td>
<td>4</td>
<td>0.5$</td>
<td>2$</td>
</tr>
<tr>
<td>Capacitor 10 pF</td>
<td>2</td>
<td>0.0833$</td>
<td>0.167$</td>
</tr>
<tr>
<td>Capacitor 10 uF</td>
<td>4</td>
<td>0.0833$</td>
<td>0.33$</td>
</tr>
<tr>
<td>Capacitor 100nF</td>
<td>2</td>
<td>0.0833$</td>
<td>0.167$</td>
</tr>
<tr>
<td>Capacitor 4 nF</td>
<td>1</td>
<td>0.0833$</td>
<td>0.0833$</td>
</tr>
<tr>
<td>Switch SW-PB</td>
<td>1</td>
<td>0.167$</td>
<td>0.167$</td>
</tr>
<tr>
<td>Crystal oscillator 4Mhz</td>
<td>1</td>
<td>0.167$</td>
<td>0.167$</td>
</tr>
<tr>
<td>Max 232</td>
<td>1</td>
<td>1$</td>
<td>1$</td>
</tr>
<tr>
<td>Resistance 16 KOhms</td>
<td>1</td>
<td>0.0833$</td>
<td>0.0833$</td>
</tr>
<tr>
<td>Resistance 100 Ohms</td>
<td>4</td>
<td>0.0833$</td>
<td>0.33$</td>
</tr>
<tr>
<td>Diodes 1N4148</td>
<td>4</td>
<td>0.0833$</td>
<td>0.33$</td>
</tr>
<tr>
<td>Transistor 2N222A</td>
<td>4</td>
<td>0.0833$</td>
<td>0.33$</td>
</tr>
<tr>
<td>Various RS-232 cables</td>
<td>2</td>
<td>15$</td>
<td>30$</td>
</tr>
<tr>
<td>Shipping using Fedex</td>
<td>N/A</td>
<td>N/A</td>
<td>158$</td>
</tr>
<tr>
<td>Tax</td>
<td>N/A</td>
<td>N/A</td>
<td>15$</td>
</tr>
</tbody>
</table>

**TOTAL** 621.1$
4 Implementation of the design

4.1 DTOPL Management Software

4.1.1 Overview

In order to be able to achieve our goals and to manage and control our project, we wrote a custom software using Delphi. This software has several functionalities:

- Handling Serial I/O communication
- Implementing our custom protocol
- Implementing Home Automation
- Implementing Security
- Providing connectivity with the web interface

4.1.2 Implementation of the software objectives

4.1.2.1 Handling Serial I/O communication

The first and most basic operation of our software is to be able to perform I/O operations over the serial port. This was achieved by using “ComPort Library ver. 3.0” for Delphi written by Dejan Crnila and maintained by Lars B. Dybdahl which is an open-source library. Using this library, we set up the interface to the serial port choosing the parameters that we need and writing and reading to it.
4.1.2.2 DTOPL custom protocol

Next, we had to make sure that the protocol is implemented by the software. In order to do this, there are 3 important functions and 1 important event:

- The first function is the custom made encapsulate function. It takes as input the different fields of a packet and outputs the packet as a string according to the packet type so it can be sent using the Comport Library’s function WriteStr.

- The second function is Comport Library’s WriteStr. It is responsible of handling all the hassle of the transmission of a string over the serial line.

- The third function is the custom made decapsulate function that that takes a string from the OnPacket event, converts it to a packet according to the packet type that was received and takes the right steps (e.g. when a “Advertise Address” packet is received, the Address table is updated).

- The fourth function which is actually Comport Library’s OnPacket event is used to indicate that a packet has been received and to call the decapsulate function so that the packet is decapsulated according to its type.
4.1.2.3 DTOPL Home Automation

The first objective is home automation. Thus, the software should allow the control of the microprocessor/relay devices. To do so, the software sends a Microprocessor/relay packet. Since we have 4 relays, the commands are in the 4 LSB (Least significant bits). So, a byte’s 4 MSB (Most significant bits are irrelevant and we fixed them to 0011. As for the 4 LSB, the first controls the first relay and so on. If the command is 0, the relay is turned OFF and if it is 1 the relay is turned ON. (e.g. 00110101 would turn the second and fourth relays ON and the first and third relays OFF)
4.1.2.4 DTOPL Security

The second objective is securing the data transfer. Thus, we implemented the RSA encryption as explained previously in this report by writing our own RSA Library using FGInt (A library written Walied Othman to handle large numbers mathematical computations) and implemented Encrypt, Decrypt and other intermediate functions such as modular exponentiation and primality tests.

4.1.2.5 DTOPL the web interface

The third objective of our project is to implement web interface to control the relays. It was written in JSP (Java Servlet Pages), hosted on an Apache Tomcat web server and connected to the software. It’s made of a simple form that shows four checkboxes (one for each relay). If a checkbox is checked the corresponding relay will be turned ON and if it’s unchecked the relay will be turned OFF.
4.1.3 Detailed description of the communication protocol

4.1.3.1 Overview

In order to have reliable communication, be able to easily implement our project and design our software, we had to design a custom communication protocol to regulate the way the information is sent and received. To do so, we took some ideas from the famous TCP/IP protocol and adapted them to our purposes adding and removing several functionalities.
4.1.3.2 Packet Format

Our protocol is based on the transmission of packets since it is one of the best and most reliable ways. We designed several types of packets however; all are based on one general format. The number between parentheses indicates the size in bytes of each field.

<table>
<thead>
<tr>
<th>Start (2)</th>
<th>Src (1)</th>
<th>Dest (1)</th>
<th>Type (1)</th>
<th>Seq (1)</th>
<th>Data (24)</th>
<th>Stop (2)</th>
</tr>
</thead>
</table>

Figure 21: General Packet Format

As we can see from the figure, our general purpose packet is made of 32 bytes and 7 different fields:

- Start: Indicates the start of a packet and is made of 2 bytes that are the ASCII characters STX (Start of text) and SOH (Start of heading)
- Src: Indicates the source device of a packet and is made of 1 byte = 8 bits so there can be a maximum of $2^8 - 3 = 253$ devices (The – 3 accounts for the fact that addresses 0, 1 and 255 are reserved for the undefined, server and broadcast addresses respectively)
- Dest: Indicates the destination device of a packet and is made of 1 byte = 8 bits so there can be a maximum of $2^8 - 3 = 253$ devices (The – 3 accounts for the fact that addresses 0, 1 and 255 are reserved for the undefined, server and broadcast addresses respectively)
- Type: Indicates the type of a packet and is made of 1 byte so there can be a maximum of 256 types.
- Seq: Indicates the sequence number of a packet and is made of 1 byte so the longest possible sequence is 256 packets long.
Data: Stores the payload (data) being transmitted in the packet and is made of 24 bytes so the can be a maximum of 24 bytes transmitted in one single packet. In the event that more than 24 bytes need to be transmitted, the data is broken down into parts of 24 bytes.

Stop: Start: Indicates the end of a packet and is made of 2 bytes that are the ASCII character EOT (End of Transmission) and ETX (End of text)

Note: This configuration will vary for the different packet types becoming smaller and made up of fewer fields.

4.1.3.3 Packet Types

- Single Data Packet (Type 1): This packet type is used for transmitting data of 24 bytes or less from Src to Dest.

<table>
<thead>
<tr>
<th>Start</th>
<th>Src</th>
<th>Dest</th>
<th>Type=1</th>
<th>Seq</th>
<th>Data</th>
<th>Stop</th>
</tr>
</thead>
</table>

Figure 22: Single Data Packet

- Multi-Packet Data Start (Type 2): This packet type is used to indicate the first packet in the sequence of a multi-packet transmission (i.e. when transmitting packets larger than 24 bytes from Src to Dest)

<table>
<thead>
<tr>
<th>Start</th>
<th>Src</th>
<th>Dest</th>
<th>Type=2</th>
<th>Seq=1</th>
<th>Data</th>
<th>Stop</th>
</tr>
</thead>
</table>

Figure 23: Multi-Packet Data Start

- Multi-Packet Data Continuation (Type 3): This packet type is used to indicate the packets in the sequence of a multi-packet transmission (i.e. when transmitting
packets larger than 24 bytes from Src to Dest) and that are between the multi-packet start and end

<table>
<thead>
<tr>
<th>Start</th>
<th>Src</th>
<th>Dest</th>
<th>Type=3</th>
<th>Seq</th>
<th>Data</th>
<th>Stop</th>
</tr>
</thead>
</table>

**Figure 24: Multi-Packet Data Continuation**

- Multi-Packet Data End (Type 4): This packet type is used to indicate the last packet in the sequence of a multi-packet transmission (i.e. when transmitting packets larger than 24 bytes from Src to Dest)

<table>
<thead>
<tr>
<th>Start</th>
<th>Src</th>
<th>Dest</th>
<th>Type=4</th>
<th>Seq</th>
<th>Data</th>
<th>Stop</th>
</tr>
</thead>
</table>

**Figure 25: Multi-Packet Data End**

- Ping (Type 5): This packet type is used by the Src to know if the Dest device is connected on the power line.

<table>
<thead>
<tr>
<th>Start</th>
<th>Src</th>
<th>Dest</th>
<th>Type=5</th>
<th>Stop</th>
</tr>
</thead>
</table>

**Figure 26: Ping**

- Ping Reply (Type 6): This packet type is sent in response to a ping packet. Once it is received by the device that initiated the ping, the pinging device will know that the Dest device is connected.

<table>
<thead>
<tr>
<th>Start</th>
<th>Src</th>
<th>Dest</th>
<th>Type=6</th>
<th>Stop</th>
</tr>
</thead>
</table>

**Figure 27: Ping Reply**
- Command ON (Type 7): This packet type is used to send an ON command from Src to Dest.

<table>
<thead>
<tr>
<th>Start</th>
<th>Src</th>
<th>Dest</th>
<th>Type=7</th>
<th>Data=1</th>
<th>Stop</th>
</tr>
</thead>
</table>

Figure 28: Command ON

- Command OFF (Type 8): This packet type is used to send an OFF command from Src to Dest.

<table>
<thead>
<tr>
<th>Start</th>
<th>Src</th>
<th>Dest</th>
<th>Type=8</th>
<th>Data=0</th>
<th>Stop</th>
</tr>
</thead>
</table>

Figure 29: Command OFF

- Advertise ON power (Type 9): This packet type is sent whenever a device connected to the power line is turned on so that the server knows it is on and gives it an address. Its Src is 0 since by this time the device has not yet acquired an address, its Dest is 1 (The server device) and its data is the device type (1 for server, 2 for PC and 3 for others) and a unique code.

<table>
<thead>
<tr>
<th>Start</th>
<th>Src=0</th>
<th>Dest=1</th>
<th>Type=9</th>
<th>Data=DevType, Unique Code</th>
<th>Stop</th>
</tr>
</thead>
</table>

Figure 30: Advertise ON Power

- Advertise OFF power (Type 10): This packet type is sent whenever a device connected to the power line is turned off. Its Src is the device’s address and its Dest is 255 (Broadcast) so that all the devices know it’s off and update their address table accordingly.

<table>
<thead>
<tr>
<th>Start</th>
<th>Src</th>
<th>Dest=255</th>
<th>Type=10</th>
<th>Stop</th>
</tr>
</thead>
</table>

Figure 31: Advertise OFF Power
- Assign Address (Type 11): This packet type is sent to assign an address to a device that has requested an address through the advertise ON power. Its Src is 1 the server’s address (since it is responsible of addressing) and its Dest is 0 since the device that requested an address does not have one yet. The data that it carries contains the data sent in the Advertise ON power packet (i.e. device type and unique code) plus the assigned address.

<table>
<thead>
<tr>
<th>Start</th>
<th>Src=1</th>
<th>Dest=0</th>
<th>Type=11</th>
<th>Data=DevType, Unique Code, Address</th>
<th>Stop</th>
</tr>
</thead>
</table>

**Figure 32: Assign Address**

- Advertise Address (Type 12): This packet type is sent whenever a device is assigned an address. Its Src is the newly assigned address and it Dest is the address 255 (Broadcast) so that all the devices know the new address and update their address table accordingly.

<table>
<thead>
<tr>
<th>Start</th>
<th>Src</th>
<th>Dest=255</th>
<th>Type=12</th>
<th>Data=DevType</th>
<th>Stop</th>
</tr>
</thead>
</table>

**Figure 33: Advertise Address**

- Assign Keys (Type 13): This packet type is sent to assign a key to a device. Its Src is 1 the server’s address (since it is responsible of assigning keys) and its Dest should be a device of type PC or other if it can handle encryption. The data that it carries are the keys.

<table>
<thead>
<tr>
<th>Start</th>
<th>Src=1</th>
<th>Dest</th>
<th>Type=13</th>
<th>Data=Private Key, Public Key, Modulus</th>
<th>Stop</th>
</tr>
</thead>
</table>

**Figure 34: Assign Keys**
- Advertise Keys (Type 14): This packet type is sent whenever a device was assigned keys. Its Src is the device’s address and it Dest is the address 255 (Broadcast) so that all the devices know the new keys and update their keys table accordingly.

| Start | Src | Dest=255 | Type=14 | Data=Private Key, Public Key, Modulus | Stop |

Figure 35: Advertise Keys

- Microcontroller/Relays (Type 15): This packet type is sent whenever a device wants to control the 4 relays. Its Src is the device’s address and it Dest is the address of the microcontroller. Its Data is made of 0,0,1,1,x0,x1,x2,x3 where x0, x1, x2 and x3 determine the states (1 ON/ 0 OFF) of relays 1, 2, 3 and 4.

| Start | Src | Dest | Type=14 | Data=0,0,1,1,x0,x1,x2,x3,x4 | Stop |

Figure 36: Microcontroller/Relays
4.1.4 Snapshots from the software interface

Figure 37: Graphical User Interface
Figure 38: Example of a “single data” packet being transmitted first encrypted (Hello) and then in plain text (Goodbye)
Figure 39: Example of a “multi-packet data” being transmitted (abcdefghijklmnopqrstuvwxyz)
Figure 40: Example of home automation opening relays 2, 3 and 4.
Figure 41: Example of ping/reply showing that a device is connected.
4.2 Home automation on PIC

One of the applications of DTOPL is home automation. A main advantage is the non-necessity to run new wires through the entire house or building in order to control the existing electrical appliances.

In our project, we chose to show the control of four 12V fans through a microcontroller (PIC16F84). We emphasized on how to deal and analyze the data coming from the power line communication protocol in order to turn on and turn off the respective AC system.

To put in a nutshell, the circuit we realized takes its instructions from the power line data, and chooses which of the four fans to turn on, and this at each different data packet received.

4.2.1 The circuit

![Picture of the controlling circuit]

Figure 42: Picture of the controlling circuit
In the above picture, we see that the microcontroller (PIC16F84) is the center part of the circuit. The packet containing the instructions comes from the serial port RS232 and is directed to the MAX232 level shifter. This level shifter is necessary before any microcontroller. The output of the MAX232 is directed towards the microcontroller, which contains the program that chooses which of the relays to turn on.

### 4.2.2 The program

The program we implemented on the microcontroller (PIC16F84A) had achieved the following objectives and results:

- Advertise on power that the relays are connected by sending a message “PIC 16F84 connected” to our main program.
- Read the data coming from the software that we developed.
- Put the received data in a register (JCJ).
- Send the data from the (JCJ) register to the output ports 4, 5, 6 and 7 in order to be able to give commands to the 4 relays.
- Synchronize between the data sent from the power lines and the data received using the rs096 and the wait routines.
- Shift banks between bank A and bank B using Bank routine.
- Reset using the SW-PB switch.
Following is a sample from the assembly code showing the sending of the data to different relays:

```
LOOP  btfs   RSflag    ; check RS232 data reception flag
  call  RSservice  ; if set, call RS232 echo
  bsf      STATUS,RP0
  bcf     TRISB,7
  bcf     TRISB,6
  bcf     TRISB,5
  bcf     TRISB,4
  bcf     STATUS,RP0
  btfsc   JCJ,0
  bsf      PORTB,4 // Sending to port 4
  btfs    JCJ,0
  bcf      PORTB,4
  btfsc   JCJ,1
  bsf      PORTB,5 // Sending to port 5
  btfs    JCJ,1
  bcf      PORTB,5
  btfsc   JCJ,2
  bsf      PORTB,6 // Sending to port 6
  btfs    JCJ,2
  bcf      PORTB,6
  btfsc   JCJ,3
  bsf      PORTB,7 // Sending to port 7
  btfs    JCJ,3
  bcf      PORTB,7
  goto LOOP
END
```
### 4.2.3 Cases

<table>
<thead>
<tr>
<th>Relay opened</th>
<th>Binary Stream (4 LSB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R4 R3 R2 R1</td>
</tr>
<tr>
<td>none</td>
<td>0000</td>
</tr>
<tr>
<td>R1</td>
<td>0001</td>
</tr>
<tr>
<td>R2</td>
<td>0010</td>
</tr>
<tr>
<td>R1 R2</td>
<td>0011</td>
</tr>
<tr>
<td>R3</td>
<td>0100</td>
</tr>
<tr>
<td>R1 R3</td>
<td>0101</td>
</tr>
<tr>
<td>R2 R3</td>
<td>0110</td>
</tr>
<tr>
<td>R1 R2 R3</td>
<td>0111</td>
</tr>
<tr>
<td>R4</td>
<td>1000</td>
</tr>
<tr>
<td>R1 R4</td>
<td>1001</td>
</tr>
<tr>
<td>R2 R4</td>
<td>1010</td>
</tr>
<tr>
<td>R1 R2 R4</td>
<td>1011</td>
</tr>
<tr>
<td>R3 R4</td>
<td>1100</td>
</tr>
<tr>
<td>R1 R3 R4</td>
<td>1101</td>
</tr>
<tr>
<td>R2 R3 R4</td>
<td>1110</td>
</tr>
<tr>
<td>R1 R2 R3 R4</td>
<td>1111</td>
</tr>
</tbody>
</table>

Figure 43: Table of possible cases for the controlling circuit

According to the ASCII table we have many combinations that end with these 4 LSB’s. Therefore in our program the format sent will hold the 4 MSB’s as don’t care and the 4 LSB’s according to the ASCII table. Hence, in order for R1 to be opened, an “A” or “a” could be sent.
5 Evaluation of the design

5.1 Results of Ariane modules performance after testing phase

5.1.1 Objectives

We conducted our experiment in order to test the reliability, efficiency and performance of the Ariane modules. We also had to check if the claims of Ariane regarding the speed rate of data transmission are true and if the numbers provided in the datasheet of the modules match the numbers we get in our experiments, using Ariane’s program.

5.1.2 Procedure

In carrying out our experiments, we used the Serial Diagnosis program developed by Ariane. We then performed the following steps:

![Diagram of the experiments procedure](image_url)

Figure 44: Diagram of the experiments procedure
1. Link PC1 to Ariane’s module1
2. Link PC2 to Ariane’s module2
3. Run Serial Diagnosis program PC1 and PC2
4. Choose which PC to transmit (in this case PC1)
5. Set the appropriate parameters in the program of PC1 (transmitter)
   a. Cycle: represents the time between two consecutive transmissions
   b. NB Bytes: represents the number of bytes transmitted in each packet
   c. Transmission remaining: represents the number of packets to transmit when the continuous transmission flag is set
6. Set the Continuous Transmission flag: this will start transmission from PC1 to PC2 according to the above diagram
7. Retrieve the number of packets received from PC2 program

Following are snapshots of the program:

![Serial Diagnosis](image)

**Figure 45: Transmission phase from PC1**
5.1.3 Graphical results

We conducted two types of experiments. The first is by setting the number of packets sent to 25 and keeping it constant, while changing the number of bytes per packet. We tried four different values of NB Bytes, 15, 30, 45 and 60 while varying the cycle length for each. We refined the cycle length when we were approaching the threshold point in order to get accurate measures of the bit rate (speed) of the modules and get at which rate the entire transmitted data is received.

The second type of experiments we conducted is to increase the load around the modules in order to increase the noise in the power line using different appliances. We then checked the number of packets received and compared it with the previous results in order to verify the claimed robustness of Ariane modules over noise.

The number of packets transmitted is constant through the whole experiment to 25 packets.
5.1.3.1 Type 1 experiment

15 bytes per packet:

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Packets received</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>7</td>
</tr>
<tr>
<td>200</td>
<td>19</td>
</tr>
<tr>
<td>201</td>
<td>19</td>
</tr>
<tr>
<td>202</td>
<td>20</td>
</tr>
<tr>
<td>203</td>
<td>20</td>
</tr>
<tr>
<td>204</td>
<td>25</td>
</tr>
<tr>
<td>205</td>
<td>25</td>
</tr>
<tr>
<td>210</td>
<td>25</td>
</tr>
<tr>
<td>225</td>
<td>25</td>
</tr>
<tr>
<td>250</td>
<td>25</td>
</tr>
<tr>
<td>500</td>
<td>25</td>
</tr>
<tr>
<td>1000</td>
<td>25</td>
</tr>
</tbody>
</table>

30 bytes per packet:

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Packets received</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>5</td>
</tr>
<tr>
<td>125</td>
<td>5</td>
</tr>
<tr>
<td>250</td>
<td>9</td>
</tr>
<tr>
<td>300</td>
<td>15</td>
</tr>
<tr>
<td>325</td>
<td>19</td>
</tr>
<tr>
<td>328</td>
<td>19</td>
</tr>
<tr>
<td>329</td>
<td>23</td>
</tr>
<tr>
<td>330</td>
<td>24</td>
</tr>
<tr>
<td>335</td>
<td>24</td>
</tr>
<tr>
<td>340</td>
<td>24</td>
</tr>
<tr>
<td>343</td>
<td>24</td>
</tr>
<tr>
<td>344</td>
<td>25</td>
</tr>
<tr>
<td>345</td>
<td>25</td>
</tr>
<tr>
<td>350</td>
<td>25</td>
</tr>
<tr>
<td>360</td>
<td>25</td>
</tr>
<tr>
<td>400</td>
<td>25</td>
</tr>
<tr>
<td>500</td>
<td>25</td>
</tr>
<tr>
<td>1000</td>
<td>25</td>
</tr>
</tbody>
</table>
45 bytes per packet:

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Packets received</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>5</td>
</tr>
<tr>
<td>200</td>
<td>8</td>
</tr>
<tr>
<td>300</td>
<td>13</td>
</tr>
<tr>
<td>400</td>
<td>13</td>
</tr>
<tr>
<td>450</td>
<td>17</td>
</tr>
<tr>
<td>465</td>
<td>22</td>
</tr>
<tr>
<td>468</td>
<td>22</td>
</tr>
<tr>
<td>469</td>
<td>24</td>
</tr>
<tr>
<td>470</td>
<td>25</td>
</tr>
<tr>
<td>475</td>
<td>25</td>
</tr>
<tr>
<td>500</td>
<td>25</td>
</tr>
<tr>
<td>1000</td>
<td>25</td>
</tr>
</tbody>
</table>

![Graph for 45 bytes per packet]

60 bytes per packet:

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Packets received</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>5</td>
</tr>
<tr>
<td>200</td>
<td>8</td>
</tr>
<tr>
<td>300</td>
<td>13</td>
</tr>
<tr>
<td>400</td>
<td>13</td>
</tr>
<tr>
<td>500</td>
<td>13</td>
</tr>
<tr>
<td>525</td>
<td>22</td>
</tr>
<tr>
<td>526</td>
<td>24</td>
</tr>
<tr>
<td>527</td>
<td>25</td>
</tr>
<tr>
<td>528</td>
<td>25</td>
</tr>
<tr>
<td>530</td>
<td>25</td>
</tr>
<tr>
<td>540</td>
<td>25</td>
</tr>
<tr>
<td>550</td>
<td>25</td>
</tr>
<tr>
<td>600</td>
<td>25</td>
</tr>
<tr>
<td>750</td>
<td>25</td>
</tr>
<tr>
<td>1000</td>
<td>25</td>
</tr>
</tbody>
</table>

![Graph for 60 bytes per packet]
Overall comparison:

Result analysis:

From the above measurements taken, we can say that as we increase the number of bytes per packet, the cycle increases which sounds logical. However we notice that the Ariane has certain limitations and start to do errors at high speed (less cycle) by receiving less packets than the sent packets. Therefore we were able to draw the following conclusions and thresholds for the different packet size:

- For a 15 byte packet: we can send a maximum of 73.5 bytes per sec with no packet loss.
- For a 30 byte packet: we can send a maximum of 87.2 bytes per sec with no packet loss.
- For a 45 byte packet: we can send a maximum of 95.7 bytes per sec with no packet loss.
- For a 60 byte packet: we can send a maximum of 113.8 bytes per sec with no packet loss.
5.1.3.2 Type 2 experiment

The experience that we came up with in order to test the reliability of the modules is to increase gradually the load on the power lines and check respectively the bit rate and the error bit rate. The following table contains the noise sources and their descriptions:

<table>
<thead>
<tr>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>6 incandescent lamps of 40W each, total of 240W</td>
</tr>
<tr>
<td>B</td>
<td>500W halogen lamp without magnetic transformer.</td>
</tr>
<tr>
<td>C</td>
<td>3 halogen lamps of 50W each, total of 150W</td>
</tr>
<tr>
<td>D</td>
<td>3 personal computers connected in parallel with their screens and UPS system with a total of 350 W</td>
</tr>
<tr>
<td>E</td>
<td>Hairdryer of 1800W output</td>
</tr>
</tbody>
</table>

Figure 47: Different load noise sources
The following table shows the results we came up with after conducting type 2 experiments for each load on the power line phase:

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Noise sources</th>
<th>Bit Rate Threshold</th>
<th>Packets Received</th>
<th>Packets transmitted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>D = 350W</td>
<td>15Bytes/204ms</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30Bytes/344ms</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45Bytes/470ms</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>A+D = 590W</td>
<td>15Bytes/204ms</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30Bytes/344ms</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45Bytes/470ms</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>A+B+D = 1090W</td>
<td>15Bytes/204ms</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30Bytes/344ms</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45Bytes/470ms</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>A+B+C+D = 1240W</td>
<td>15Bytes/204ms</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30Bytes/344ms</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45Bytes/470ms</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>A+B+C+D+E = 3040W</td>
<td>15Bytes/204ms</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30Bytes/344ms</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45Bytes/470ms</td>
<td>25</td>
<td>25</td>
</tr>
</tbody>
</table>

Figure 48: Type 2 experiment results
**Result analysis:**

We can conclude that Ariane is highly reliable since by increasing the load, only one packet has been lost in 2 of the 15 cases, and only in the 45 bytes per packet transmission. Therefore, based on our experiments, Ariane modules are 96% efficient.

---

### 5.1.4 Conclusion

We conclude from both types of experiments that the best packet size to be transmitted via the Ariane modules is a 32-byte packet as can be seen from our software.

---

### 5.2 Problems encountered

#### 5.2.1 Software

During the design of our protocol, we encountered 3 major problems:

- **5.2.1.1 Size of Packet**

  *Problem*

  The first problem was choosing an appropriate size for the packet. In order to achieve that, there were 2 considerations.

  The first is that transmitting larger packets would reduce overhead (i.e. if larger packets are allowed we would need to transmit a fewer number of packets thus fewer start / src / dest / type / stop fields).

  The second is that the larger the packet, the larger the line utilization and the larger the probability of losing packets (i.e. when transmitting a large packet, the line is kept busy...
by the transmitting side and since our system is half-duplex it cannot transmit and receive in the same time so if it’s transmitting for a long time and trying to receive in the same time, the receive buffer might be full and thus packets will be lost)

Solution

In order to solve this problem, we decided to use a variable packet size since in several cases we only need some fields and transmitting unneeded fields would incur larger overhead.

As for choosing the largest packet size (i.e. 32 bytes) it was done through the analysis of the performance test results made on the Ariane modules.

5.2.1.2 Unique Addressing

Problem

The second problem appeared when we were implementing our addressing scheme. In fact, in order for a device to get an address it sends an “Advertise ON Power” packet assuming its address is 0 to the server. When the server decides on an address it sends an “Assign Address” back to the address 0. But, the problem arises if several devices are turned on simultaneously. In this case, the same address might be assigned to different devices or an address might be assigned to the wrong address.

Solution

In order to solve this problem, we decided to transmit a unique random code within the “Advertise ON Power” packet which is sent back with the assigned address in the “Assign Address” packet. In this way, the device checks the unique code with its own code. If they’re the same then the address that was received is destined to it.
5.2.1.3 Connectivity and compatibility between several languages, protocols and standards

Problem

Since we are using different languages and different conventions, connectivity between them is an issue.

Solution

In order to solve this, we decided to use something that is language/protocol/standard neutral. The easiest and most reliable thing that we found to satisfy this requirement is a text file. Thus, we designed our web interface to write the form being submitted as a text file (i.e. Once the submit button is pressed, a file is written and it contains a string R0R1R2R3 where RX is 0 when the checkbox is unchecked and 1 when it is checked).

Figure 49: Example of a form submission where R0R1R2R3=0101
And, we deigned our software to monitor the file being written to whereby any modification to the file would trigger an OnChange event from the “DirectoryWatch v1.4” component written by “Angus Johnson” that we used for this purpose. Once this event is triggered the software sends the appropriate command to the relays.

5.2.2 Hardware

We encountered a few problems in the program of the microcontroller. We couldn't see any output at the pins of the PIC, even though a HIGH voltage (5.079V) could be detected at the other pins. After many reviews of the program, we found out that the whole PORTB was cleared after each of the iterations of the main loop, so the pins were displaying an output but it was too fast for us to detect. The solution was to define the outputs of PORTB outside the loop, after copying the information coming in another register than the main one.

A problem occurred in the connection of the serial port to the MAX232, where the connection was done in a mirror manner and one of the pins was soldered incorrectly. We connected pins 1, 3 and 4 instead of pins 2, 3 and 5, the pins actually needed.

However the problem was easily solved by soldering it in the reverse symmetrical manner after reading the pin numbers on a different female serial port terminal.
5.3 Design constraints

The main constraint in our design is an economical one. We could have created a much more complicated program and corresponding interface, and we could have hosted a website to test our system effectively and make it available online, but the main constraint is the budget we had for the project. The modules we bought from Ariane Controls that permits us to send data packets through power lines cost us more than the project budget alone, which restrained us from buying any additional expensive components for our system. This economical constraint prevented us from buying additional modules, as specified in the original design, and we had to change the initial circuit. We decided to build a single circuit that outputs control instructions to multiple devices instead of having a control circuit for each device alone. This is more practical for demonstration purposes and less expensive, but for an actual case, in a house, this will require running new wires (except for close electrical appliances), which collides with our main target.

Apart from the Ariane modules, we could find our system's components in the market easily, which implies that for a mass production of the controlling circuit, it wouldn't cost much and it would be practical to manufacture and assemble. The tests we conducted on Ariane's modules demonstrated the robustness of our system and hence its sustainability.

Our system is "environmentally" clean, there's no hazard except locally, only if the module or one of the relays burns out. Moreover, the project we designed costs relatively low in comparison with other similar home automation designs. From this we can say that our system is available to a wider range of customers in the market than pure home automation systems where one's need to run new wires and install many electronic devices.
6 Rooms for Design Development: AMR

6.1 Automatic Meter Reading overview

Now that the communication protocol has been established thanks to our program, and the Transmitters and Receivers modems of Ariane have been bought we think that one interesting application that can be done based on our FYP is the Automatic Meter Reading.

By this idea we mean that the electrical substation (supplier), via the power line network, will be able to send and receive data related to the status of the electrical meters installed at home such as the KWh rates, prepaid amounts without the need of an employee from EDL to go and check the status of your meter.

6.2 AMR Diagram description

![Diagram showing the AMR system](image)

Figure 50: Diagram showing the AMR system
Description of the AMR block Diagram and Information Flow:

The IU (Interface Unit) is an intelligent device, which can collect, process, and record power consumption data from the electric meter. It can be formed by the use of microcontrollers and has the following tasks to achieve:

- Picks up the pulse output of the meter and converts the measurement of the meter into a digital format suitable for data processing. Thus it will be possible to monitor the electrical load in real time.
- Saves the data collected in non-volatile memory protecting it hence against power failure. Therefore it will automatically resume normal operation when power returns after a power failure.
- Data stored in the IU are transmitted to the Data Concentrator Unit (DCU) software installed in the Host Computer via the Ariane RS-232 Power Line Modem. Communication is initiated by the DCU software, which polls the IU by calling its address (this feature is already developed in our program). Data received from the IU is stored in the software in order to be analyzed.

The Host Computer (HC) is the control center of the system, where all the functions of the system are controlled and monitored. The HC via its DCU software passes instructions and information requests to the IU by calling its address. The address codes are stored in the HC. In case of failures in self-diagnostics or any abnormal behavior of the IU, the DCU can also make requests to report. The DCU will convert the data received into a text file compatible with the corporation's existing Meter Reading Management System, and store it in the Hard Disk Drive and therefore manipulation of the data such as scheduling of the load, tariffs, power consumption can all be calculated.
Therefore the main tasks that should be achieved will be:

- Choose an appropriate electric meter.
- Design the Interface Unit
- Build-up on our Delphi program to write the DCU software; which is able to analyze data given from the electric meter and to study the load consumption on one phase.
7 Conclusion

During our work on this project, we studied several protocols and standards already on the market and compared them taking into considerations each one’s advantages and disadvantages in order to be able to choose the one that best suit our purposes. Once that our research and design were completed in the fall semester, we have built our system during this semester from the software to the hardware interfaces based on the experiences that we came up with in the performance analysis.

This project has deepened our technical knowledge in each aspect of the electrical engineering program from power systems analysis to communication protocols and programming languages such as Delphi and Assembly. Moreover all the objectives that we have set in the fall were achieved: the communication protocol software, the internet site, the home automation and the security encryption. In addition to that we note that our project can act as a base form for applying automatic meter reading on a one phase electric meter (a detailed design was presented in section 6). Finally we would like to thank our advisor Dr. Karaki for his support during both semesters and we will be glad to share any information with any prospective group that will base his FYP on our results.
8 References


[6] EN 50 065-1, “Signalling on low voltage electrical installations in the frequency range 3 KHz to 148.5 KHz; Part 1: General requirements, frequency bands and electromagnetic disturbances”, CENELEC, Brussels, 1991


[16] Bhaskar Bora, Implementation of RSA Algorithm, Bangalore, India, March 2003