Design, Implementation and Testing of a Flexible, Intelligent Modem Architecture for Power Line Communications

by

Barry Buternowsky


A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

DEPARTMENT OF ELECTRICAL ENGINEERING

We accept this thesis as conforming to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

January 1992

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

(Signature)

Department of Electrical Engineering

The University of British Columbia
Vancouver, Canada

Date February 25, 1992
Abstract

Intrabuilding electric power distribution lines provide easy access and universal coverage for data communications. However, power line communication channels exhibit unpredictable levels of impedance, attenuation and noise. To enable effective communications requires appropriate modulation schemes and communication protocols.

This thesis describes the design, implementation and testing of a modular, flexible, intelligent modem architecture, based on binary phase-shift keying (BPSK). The modem operates at user selectable baud rates of 2.4, 4.8, 9.6 and 19.2 kbps. Most of the design is digital, to facilitate eventual implementation using VLSI. A microcontroller resident on the modem enables judicious allocation of data link layer functions between modem and host microcomputer. Data link functions performed by the microcontroller include byte and packet synchronization, and control of host-modem communications. The overall data link protocol design results in a flexible, user specified packet format and length, which accommodates to different data rates. The modem architecture is designed for use with various medium access protocols including CSMA, polling or token passing. Addition of forward error correction capability in the form of on-board hardware is also feasible.

Evaluation of the modem architecture was performed in a four-storey industrial building which contains good, fair and poor quality communication links. Results for same phase and cross phase transmissions were collected and analyzed for two different test environments: (1) short length links of approximately 30 meters within the same room location; and, (2) moderate to very long links (greater than 60 meters) among different floors. Results from tests (1) indicate that same phase transmission is in general advantageous over cross phase transmission. This advantage for BERs less than $10^{-3}$ for transmission at 19.2 kbps can be large as 27 dBmV. Evaluation of results from tests (2) indicate that power line channel communication characteristics are directly related to the channel’s length. The BER was found to be in a range from $10^{-1}$ to $10^{-3}$ for transmission at 19.2 kbps. Comparison of channel throughput for continuous one-way transmission without forward error correction indicates 2.4 kbps would be appropriate for long distance, 4.8 and 9.6 kbps for medium distances, and 19.2 kbps for relatively short distances.
# Table of Contents

Abstract ......................................................................................................................... ii

List of Figures ............................................................................................................... vi

List of Tables ............................................................................................................... ix

Acknowledgements ..................................................................................................... x

1 Introduction .............................................................................................................. 1
  1.1 MOTIVATION ........................................................................................................ 1
  1.2 BACKGROUND .................................................................................................... 1
  1.3 OVERALL MODEM DESIGN CONSIDERATIONS ............................................ 3
  1.4 OUTLINE OF THESIS ......................................................................................... 4

2 Power Line Channels ............................................................................................... 6
  2.1 ELECTRICAL POWER DISTRIBUTION ........................................................... 6
    2.1.1 Power Systems Topology .............................................................................. 6
    2.1.2 Building Wiring Plans ................................................................................. 6
  2.2 ATTENUATION ................................................................................................. 8
    2.2.1 Attenuation in Industrial Buildings .............................................................. 8
    2.2.2 Attenuation in Residential Buildings ........................................................... 11
  2.3 NOISE ............................................................................................................. 13
  2.4 SIGNAL FADING ............................................................................................. 16
  2.5 POWER LINE LOADING ................................................................................. 16

3 Hardware Architecture—Design and Implementation ........................................... 19
  3.1 OVERVIEW OF MODEM HARDWARE ARCHITECTURE ......................... 19
  3.2 TRANSMITTER AND RECEIVER SUPPORT COMPONENTS ................... 19
    3.2.1 Power Supply ............................................................................................. 19
    3.2.2 The Clock Generator ............................................................................... 20
    3.2.3 A Description of the Microcontroller Control Pins ................................. 20
    3.2.4 The Host/Modem Interface ...................................................................... 22
  3.3 MODEM TRANSMITTER HARDWARE ......................................................... 22
    3.3.1 BPSK Modulator ....................................................................................... 22
    3.3.2 The Transmit Low Pass Filter .................................................................. 24
    3.3.3 The Power Amplifier ............................................................................... 25
Table of Contents (Contd)

3.4 MODEM RECEIVER HARDWARE ............................................ 25
   3.4.1 Line Coupling Network ............................................. 25
   3.4.2 Receive Band Pass Filter ......................................... 26
   3.4.3 The Costas Loop ................................................... 26
   3.4.4 Bit Synchronization Circuit ...................................... 32
   3.4.5 Integrate-and-Dump Filter and Host Interface .............. 35

4 Firmware Design and Implementation ...................................... 36
   4.1 OVERVIEW OF MODEM FIRMWARE ORGANIZATION .......... 36
   4.2 MODEM INITIALIZATION MODULE .................................. 38
   4.3 MODEM TRANSMIT MODULE ........................................... 38
      4.3.1 Transmitted Packet Format: Preamble and Data .......... 39
      4.3.2 Host-Modem Communication .................................. 40
      4.3.3 XON/XOFF Host-Modem Flow Control ...................... 42
      4.3.4 Synchronous Handshaking for Asynchronous
            Communication ............................................... 43
   4.4 MODEM RECEIVE MODULE ............................................ 44
      4.4.1 Data Link Layer ............................................... 45
      4.4.2 Discussion of Data Link Layer Parameters .......... 47
      4.4.3 Carrier Detection .......................................... 50

5 Test Results in Different Environments .................................. 54
   5.1 DISCUSSION OF TEST PARAMETERS ................................. 54
   5.2 TEST RESULTS UNDER WHITE NOISE CONDITIONS ............ 57
   5.3 TEST ENVIRONMENT AND PROCEDURE ............................ 58
   5.4 SAME PHASE AND CROSS PHASE TESTS ON LAB SYSTEM ... 60
   5.5 TESTS ON GENERAL IN-BUILDING ELECTRICAL
      DISTRIBUTION SYSTEM ............................................ 68
      5.5.1 Bit Error Rate ............................................... 70
      5.5.2 Block Error Rate ............................................ 70
      5.5.3 Percent of Lost Packets ................................... 71
      5.5.4 Summary .................................................... 71
   5.6 TESTS FOR TRANSMISSIONS FROM LAB TO
      GENERAL SYSTEM .................................................. 72
      5.6.1 Bit Error Rate ............................................... 72
      5.6.2 Block Error Rate ............................................ 72
      5.6.3 Percent of Lost Packets ................................... 73
      5.6.4 Error-free Throughput .................................... 73
      5.6.5 Summary .................................................... 74
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 Conclusions</td>
<td>75</td>
</tr>
<tr>
<td>6.1 CONCLUDING REMARKS</td>
<td>75</td>
</tr>
<tr>
<td>6.2 SUGGESTIONS FOR FURTHER WORK</td>
<td>76</td>
</tr>
<tr>
<td>References</td>
<td>78</td>
</tr>
<tr>
<td>A Hardware Schematics</td>
<td>83</td>
</tr>
<tr>
<td>B Firmware Listing</td>
<td>94</td>
</tr>
</tbody>
</table>
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Residential electrical power distribution</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Commercial/Industrial three-phase electrical power distribution</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>Attenuation during the work day in an industrial building [1]</td>
<td>11</td>
</tr>
<tr>
<td>2.4</td>
<td>Attenuation in an industrial building at night [1]</td>
<td>11</td>
</tr>
<tr>
<td>2.5</td>
<td>Attenuation for residential in-building power line with resistive load [1]</td>
<td>12</td>
</tr>
<tr>
<td>2.6</td>
<td>Attenuation across phases in residential power lines [1]</td>
<td>13</td>
</tr>
<tr>
<td>2.7</td>
<td>Residential power line bandwidth [1]</td>
<td>14</td>
</tr>
<tr>
<td>2.8</td>
<td>Distribution network load profile during one day [50]</td>
<td>17</td>
</tr>
<tr>
<td>2.9</td>
<td>Distribution network load profile during one week [50]</td>
<td>18</td>
</tr>
<tr>
<td>3.1</td>
<td>Output of shift register 74HCT164 in modulator circuit</td>
<td>23</td>
</tr>
<tr>
<td>3.2</td>
<td>Synthesized sinusoidal carrier</td>
<td>24</td>
</tr>
<tr>
<td>3.3</td>
<td>Second order butterworth LPF</td>
<td>24</td>
</tr>
<tr>
<td>3.4</td>
<td>Second Order Butterworth Band Pass Filter</td>
<td>26</td>
</tr>
<tr>
<td>3.5</td>
<td>BPSK Costas loop demodulator</td>
<td>28</td>
</tr>
<tr>
<td>3.6</td>
<td>Digital Costas loop</td>
<td>28</td>
</tr>
<tr>
<td>3.7</td>
<td>Spectrum of square wave (c(t))</td>
<td>29</td>
</tr>
<tr>
<td>3.8</td>
<td>Block diagram of loop filter chip</td>
<td>30</td>
</tr>
<tr>
<td>3.9</td>
<td>Inserted and deleted pulses of the loop filter chip</td>
<td>30</td>
</tr>
<tr>
<td>3.10</td>
<td>BER vs loop filter bandwidth</td>
<td>31</td>
</tr>
<tr>
<td>3.11</td>
<td>Percent lost packets vs loop filter bandwidth</td>
<td>31</td>
</tr>
<tr>
<td>3.12</td>
<td>Bit synchronization circuit</td>
<td>33</td>
</tr>
<tr>
<td>3.13</td>
<td>Non-synchronized bit stream</td>
<td>33</td>
</tr>
<tr>
<td>3.14</td>
<td>Perfectly synchronized bit stream</td>
<td>34</td>
</tr>
<tr>
<td>3.15</td>
<td>Bit Sync waveforms for different up/down values</td>
<td>34</td>
</tr>
<tr>
<td>3.16</td>
<td>Noisy bit stream</td>
<td>34</td>
</tr>
<tr>
<td>3.17</td>
<td>Integrate and dump matched filter</td>
<td>35</td>
</tr>
<tr>
<td>4.1</td>
<td>Overall view of modem firmware</td>
<td>37</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.2</td>
<td>OSI reference model</td>
<td>37</td>
</tr>
<tr>
<td>4.3</td>
<td>Commands sent to the modem by the host during initialization</td>
<td>38</td>
</tr>
<tr>
<td>4.4</td>
<td>Transmit module data link layer functions</td>
<td>39</td>
</tr>
<tr>
<td>4.5</td>
<td>Transmit module physical layer functions</td>
<td>39</td>
</tr>
<tr>
<td>4.6</td>
<td>Transmit module block diagram</td>
<td>41</td>
</tr>
<tr>
<td>4.7</td>
<td>Transmitted packet format</td>
<td>41</td>
</tr>
<tr>
<td>4.8</td>
<td>Circular transmit buffer</td>
<td>42</td>
</tr>
<tr>
<td>4.9</td>
<td>Receive module physical layer functions</td>
<td>44</td>
</tr>
<tr>
<td>4.10</td>
<td>Receive module data link layer functions</td>
<td>45</td>
</tr>
<tr>
<td>4.11</td>
<td>Data link layer flow chart</td>
<td>46</td>
</tr>
<tr>
<td>4.12</td>
<td>Packet Sent to Host</td>
<td>47</td>
</tr>
<tr>
<td>4.13</td>
<td>Performance of data link layer</td>
<td>48</td>
</tr>
<tr>
<td>4.14</td>
<td>Correlation of SYNC word</td>
<td>49</td>
</tr>
<tr>
<td>4.15</td>
<td>High and low confidence bits</td>
<td>51</td>
</tr>
<tr>
<td>4.16</td>
<td>Time taken for new_confidence to fall below confidence_threshold</td>
<td>53</td>
</tr>
<tr>
<td>4.17</td>
<td>Bits to terminate lock state for cur_conf of 0, 1, &amp; 2</td>
<td>53</td>
</tr>
<tr>
<td>5.1</td>
<td>Packet format for Bit Error Rate Tester</td>
<td>54</td>
</tr>
<tr>
<td>5.2</td>
<td>Noisy and quiet periods on a channel</td>
<td>56</td>
</tr>
<tr>
<td>5.3</td>
<td>Modem performance in white noise: BER and BLKER</td>
<td>58</td>
</tr>
<tr>
<td>5.4</td>
<td>BLKER and percent lost packets vs BER for white noise</td>
<td>58</td>
</tr>
<tr>
<td>5.5</td>
<td>Three phases of a power line</td>
<td>59</td>
</tr>
<tr>
<td>5.6</td>
<td>BER for lab system</td>
<td>61</td>
</tr>
<tr>
<td>5.7</td>
<td>BLKER for lab system</td>
<td>62</td>
</tr>
<tr>
<td>5.8</td>
<td>BER vs BLKER for lab system</td>
<td>63</td>
</tr>
<tr>
<td>5.9</td>
<td>Percent lost packets for lab system</td>
<td>64</td>
</tr>
<tr>
<td>5.10</td>
<td>Variation in BER as plotter location is varied</td>
<td>65</td>
</tr>
<tr>
<td>5.11</td>
<td>Frequency spectrum of output of bandpass filter</td>
<td>66</td>
</tr>
<tr>
<td>5.12</td>
<td>Waveforms of output of demodulator and bandpass filter</td>
<td>67</td>
</tr>
<tr>
<td>5.13</td>
<td>BER and BLKER vs time for x to y channel at 63 dBmV</td>
<td>68</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Figure 5.14</td>
<td>Electrical engineering building</td>
<td>69</td>
</tr>
<tr>
<td>Figure 5.15</td>
<td>BER for 70 dBmV data signal on General System</td>
<td>70</td>
</tr>
<tr>
<td>Figure 5.16</td>
<td>BLKER for 70 dBmV data signal on General System</td>
<td>71</td>
</tr>
<tr>
<td>Figure 5.17</td>
<td>Percent lost packets for 70 dBmV data signal on general system</td>
<td>71</td>
</tr>
<tr>
<td>Figure 5.18</td>
<td>BER for 70 dBmV data signal on Lab to General System</td>
<td>72</td>
</tr>
<tr>
<td>Figure 5.19</td>
<td>BLKER for 70 dBmV data signal on Lab to General System</td>
<td>73</td>
</tr>
<tr>
<td>Figure 5.20</td>
<td>Percent lost packets for 70 dBmV data signal on lab to general system</td>
<td>73</td>
</tr>
<tr>
<td>Figure 5.21</td>
<td>Error-free throughput for 70 dBmV data signal on lab to general system</td>
<td>74</td>
</tr>
</tbody>
</table>
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Error free lengths for 38.4 kbps, industrial building</td>
<td>14</td>
</tr>
<tr>
<td>2.2</td>
<td>Error free lengths for 19.2 kbps, industrial building</td>
<td>14</td>
</tr>
<tr>
<td>2.3</td>
<td>Error free lengths for 4.8 and 19.2 kbps, industrial building</td>
<td>14</td>
</tr>
<tr>
<td>2.4</td>
<td>Summary of Tables 2.1-2.3 with single bit errors</td>
<td>15</td>
</tr>
<tr>
<td>3.1</td>
<td>Shift register waveforms: unweighted and weighted</td>
<td>23</td>
</tr>
<tr>
<td>4.1</td>
<td>Mean time between false preambles</td>
<td>50</td>
</tr>
<tr>
<td>4.2</td>
<td>new_confidence values for different current_confidence values</td>
<td>52</td>
</tr>
<tr>
<td>5.1</td>
<td>Sub-buses in General and Lab Systems</td>
<td>69</td>
</tr>
<tr>
<td>5.2</td>
<td>Phases of Power Lines at Test Locations</td>
<td>69</td>
</tr>
</tbody>
</table>
Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. R. W. Donaldson for his support and guidance during the course of this thesis.

A special thank-you is extended to my family for their endless supply of love and encouragement throughout this thesis. I would also like to thank Mr. William Cheung for making available the software tools that were used during this research.

Partial funding for this thesis was provided by the Natural Sciences and Engineering Council of Canada in the form of an academic scholarship for which I am grateful.
1 Introduction

1.1 MOTIVATION

The use of intrabuilding communications involving computers and other terminal equipment has increased rapidly over the past few years. The need for interoffice communications, local area networks, security control, and energy use management in buildings is creating a demand for faster, cheaper, and more reliable systems and services. Conventional wireline communications media including twisted pair and coaxial cable are suitable for carrying baseband and broadband communications. For wide area networks, media include fibre optic, microwave, and satellite links which support broadband communications [8].

The driving cost, in networking, after computer and other terminal equipment, is the installation of cables [8]. These installation costs can be substantial even for relatively simple networks. Although a network may have been functioning for an extended period of time, as it expands so do the expenses of installing new lines.

Power line communication takes advantage of existing copper lines within the building walls. A natural interface between power lines and the communications device exists in the form of a standard wall plug. There are no extra costs for expanding the network, save the cost of additional modem and terminal devices. In addition, the physical location of terminals can be changed at will, thus creating a network which is flexible in response to terminal locations and user needs.

Power lines are not designed for use as communication channels. They exhibit unpredictable and varying levels of impedance, noise, and attenuation. These impairments must be overcome to enable effective communications.

1.2 BACKGROUND

Electric power distribution lines have been used by the power industry for distribution automation including remote meter reading, load management, and other
applications [6]. Because of various degradative factors including large amounts of noise generated by electrical equipment, most of these systems have data rates below 100 bps.

Recent developments in local area networks (LANs) and independent microprocessor controlled equipment enhances the benefits of using power lines as a communication medium for local networking inside buildings. By communicating over power lines one has potentially access to a network which encompasses an entire building. With sufficient bandwidth, transmission of digital voice as well as data is feasible. Automated home control is one possibility. Previous home bus systems have been developed [37-43] which use dedicated twisted pair and coaxial cable for communicating at 9600 bps. The features provided by the home bus can be extended to industrial buildings which, because of their considerable size, are excellent candidates for power line communications.

Several commercial intrabuilding power line communication systems have been developed during the last few years. These include modems from Signetics, NONWIRE, BSR, ExpertNet, and National Semiconductor [6,12]. These products have data rates ranging from 120 bps to 1.2 kbps and operate at carrier frequencies from 30 kHz to 150 kHz. The modulation schemes include amplitude shift keying and noncoherent frequency shift keying. These schemes are simple to implement, and inexpensive to manufacture. However, their performance capabilities are severely limited. In addition to these commercial products, several research systems have been reported. A pseudonoise spread spectrum modem that operated at 60 bps used the AC crossings of the 60 Hz power signal for synchronization [4]. A spread spectrum minimum shift keyed modem with fixed maximum baud rate of 19.2 kbps was documented in [1].

Channel impairments severely limit the data transmission speed and accuracy achievable on power line networks. At low data rates, power line impulse noise is relatively small, energy per data bit is high, and reliable performance is achievable at relatively low transmitter power levels. As the transmitted bit period decreases (when transmission speed increases), the detrimental effects of impulse noise and fading increase. At moderate to high transmission speeds (9.6 kbps and higher), impulse noise becomes the dominant factor, on some links, in determining the performance of the communication channel.

Increasing interest in powerline communications as a cost effective alternative for LAN implementation motivates extensive testing of possible future LAN sites. This
testing will provide valuable information concerning the overall potential of the site, and will also provide important information which must be considered in the design of the network. This knowledge is important if future commercial developments are to be realized.

A power line LAN must be commercially competitive with other network media. Dedicated wireline or optical communication networks have a very high transmission capacity and reliability. It is therefore necessary that the cost of power line LAN facilities be kept as low as possible, while maintaining adequate reliability and throughput, and low message delay. Although several modems have been developed and tested, there remains an absence of a low-cost, reliable modem architecture and associated data link control protocol with adequate performance capabilities for medium-speed, in-building applications.

In this thesis, a unique microprocessor-based power line modem was developed and tested. This modem uses BPSK (binary phase shift keying) modulation which is superior to DPSK (differential PSK) by 1 dB in additive white Gaussian noise, and is superior to coherent OOK (On-Off shift keying) and coherent FSK (frequency shift keying) by 3 dB [10]. A data link layer protocol which resides on the modem performs byte and packet synchronization. Optimization of channel throughput is provided through variability of packet size and baud transmission rate. Baud rates from 2.4 kbps to 19.2 kbps are dynamically selected through software. An industrial building was selected as the test site. Performance results were recorded for short and long transmission distances and for good and for very poor quality channels.

1.3 OVERALL MODEM DESIGN CONSIDERATIONS

Figure A.1 in Appendix A is a block diagram of the modem design. Shown in this schematic are the signals passed to each modem component and the clear interdependence among the components. An important block is the control circuitry which contains a microcontroller. The microcontroller's program is stored in an external EPROM which allows the microcontroller to supervise all aspects of the modem functioning. In part, the modem's maximum baud rate is limited by instruction execution speed. The other overriding restriction in developing high speed modems is hardware complexity which if allowed to proceed unchecked would produce an impractically costly product.
The design philosophy of microcontroller-based modems is derived from several motivations. These include flexibility, modem intelligence, and chip count (or cost). A microcontroller-based modem's functionality often can be modified by simply rewriting the microcontroller's firmware. Functions can be added, operational problems resolved, and options expanded with minimal modem service time. As processors become smaller and faster, microcontroller-based systems become smarter. Tasks otherwise performed in host computers are implemented within the modem itself. Because the microcontroller replaces many elementary chips, the overall chip count and physical size are reduced. A smaller number of chips generally results in a less expensive and more reliable system.

Some subsystems of the modem developed in this thesis were tested earlier, in preliminary form at 9.6 kbps transmission rate [21]. These subsystems include some parts of the modulator, demodulator, and carrier detector. The modem described herein, goes well beyond earlier work, and is thoroughly tested on various links in a harsh operating environment. Several changes in design philosophy and implementation have enabled an increase in the transmission rate to 19.2 kbps. Appropriate data link control functions are also developed, judiciously partitioned for implementation on the modem or within the PC host.

The purpose of this thesis is to design, implement and evaluate a cost-effective power line modem which operates with a high degree of reliability under variable channel conditions, at data rates up to 19.2 kbps. Design simplicity is emphasized. Generally, overly complex schemes are more expensive and less reliable than simpler schemes. Where possible, techniques which lend themselves to all-digital VLSI implementation, or failing that, mixed digital/analog application specific integrated circuits — ASICs, are used. Two very important design criteria are high data rates and baud rate selectivity. The modem must have the ability to change baud rates as needed. Modem architecture should be modular and amenable to the future addition of on-board medium access protocols including CSMA, polling, or both. It is useful to accommodate user-defined packet format and length as dictated by application and channel quality. Testing of the modem on a wide variety of actual power line communication channels is essential.

1.4 OUTLINE OF THESIS

The remainder of this thesis comprises five additional chapters and two appendices. Chapter 2 contains a description of the power line communications channel. This description includes a summary of some previous work by others and provides the
necessary background for understanding power line channel impairments. Signal attenuation and noise characteristics for both industrial and residential buildings are analyzed. These are the two major channel impairments.

Chapter 3 describes the overall modem hardware design approach and architecture as well as hardware circuitry within the modem. This description includes analysis concerning choice of parameters for low pass and band pass filters, and the design of a digitally realized modulator and demodulator. Design and analysis of an all-digital symbol synchronization means is also included. Coupling circuitry to enable interfacing of the modem to the power line is described.

Chapter 4 describes the design and implementation of the data link layer functions stored in firmware and executed on the modem's microcontroller. The data link layer performs byte and packet synchronization. The firmware also includes a user interface which allows various parameters such as modem baud rate to be changed dynamically. The distribution of data link functions between the modem and the host is designed to enable maximum application flexibility.

Chapter 5 describes the results from numerous performance tests based on measurements within an industrial building. These tests determine operating parameters for the power line modem as packets are sent from a transmitting host (IBM-PC compatible 386) to a receiving host. Several different parameters are tabulated including bit error rate, block error rate, percent of lost packets, and throughput. The tests were performed over different times of the day, and different powerline communication channels, at various transmitter power levels.

Chapter 6 contains a summary of the thesis work and performance results, and provides suggestions for further research.

The appendices provide detailed modem circuitry and firmware listings.
2 Power Line Channels

In this chapter, details of the dominant properties of power lines — attenuation, noise levels, and fading — are described for industrial and residential sites. The information concerning these two factors is derived from various sources including [1] and [6]. In order to understand the characteristics and behavior of intrabuilding power line communication, it is necessary to first understand the way electrical power is distributed within a building.

2.1 ELECTRICAL POWER DISTRIBUTION

2.1.1 Power Systems Topology

Three phase electrical power is distributed from generating plants to consumers over large and complex networks. High voltage transmission lines transmit power from generating plants to electrical substations. The transmission lines vary in length from tens to hundreds of kilometers. Next, a distribution network delivers electrical power from substations to a number of distribution transformers [4]. The distribution network varies in size as well. It is small for villages and extensive for metropolitan areas. The secondary (load) side of a distribution transformer connects to the circuit panels in a building. A single distribution transformer may provide electrical power for many buildings in residential districts, or for single buildings in industrial sectors.

2.1.2 Building Wiring Plans

In residential housing or apartment units, the secondary side of the distribution transformer delivers split-single-phase power to circuit panels by two 120V (180° out of phase) 60 Hz lines and a neutral conductor as shown in Figure 2.1. The neutral conductor is normally connected to the grounded circuit panel. Electrical power is distributed throughout the building on general purpose branch circuits which usually consist of twisted pair copper wiring. One wire of the twisted pair is connected to a 120V line, and the other is connected to neutral. Small loads are interfaced to the branch circuits using standard wall plug-in receptacles.
Figure 2.1 Residential electrical power distribution
Electrical power is supplied to large appliances on special individual dedicated branches. Washing machines, refrigerators, dishwashers, and freezers all have their own 120V branch circuit. Appliances with large heating elements such as electric ranges, water heaters and clothes dryers usually require a 240V branch circuit. These appliances are connected to both 120V lines, and the neutral. The heating element within these appliances is connected directly across the two 120V lines while the motor (in the clothes dryer) is connected to a single 120V line and neutral.

Commercial and industrial buildings are typically supplied with three-phase electrical power as indicated in Figure 2.2. In very large buildings, each floor may be supplied by separate three-phase transformers. Standard branch circuits consisting of a 120V line and a neutral line supply small loads. Larger loads are supplied by circuits that deliver either single-phase or polyphase power.

2.2 ATTENUATION

Attenuation is the amount by which a signal decreases in amplitude during transmission. Many resistive loads are normally attached to power lines in buildings; as a result the amount of attenuation will depend on number, types and location of loads. It is expected that industrial buildings will have higher attenuation than residential ones, and as the next two sections show, this is in fact the case.

2.2.1 Attenuation in Industrial Buildings

There have been a number of measurements regarding the attenuation characteristics of power lines [1,5,6]. A brief synopsis of the measurements will be given here and only the major observations will be noted.

It is agreed that the power line transmission characteristics resemble those of a low pass filter. The cut off frequency is typically located somewhere between 70 kHz and 150 kHz. Where it lies at any particular time is dependent on the loading profile of the line.
Figure 2.2 Commercial/Industrial three-phase electrical power distribution
A set of transmission curves was obtained in [1] for the electrical engineering building at the University of British Columbia, which is classified as an industrial building. Several attenuation curves appear in Figures 2.3 and 2.4 where the symbol “L” means “local”, the symbol “R” means “remote”, the symbol “Ø” means phase, and the symbols “A,B,C” denote the three phases of a three phase power line. The measurements for Figure 2.3 were taken during working hours from 8:30-4:00, and the measurements for Figure 2.4 were taken after 6:00 PM. The term “local” refers to the power lines physically wrapped in the same cable, whereas “remote” refers to the power lines not connected to the “local” power cable directly.

For each test in Figures 2.3 and 2.4 the receiver was connected to “local” phase B while the transmitter was moved to “local” and “remote” phase locations as indicated in Figures 2.3 and 2.4. There are several interesting observations based on Figures 2.3 and 2.4: (1) There is 5-7 dB attenuation from 30 to 70 kHz on “local” phase B; (2) There is 15-30 dB attenuation on “local” phases A and C; (3) All of the remote phases produce similar attenuation curves except for an underlying attenuation of approximately 25 to 30 dB; (4) There are no strictly narrow band dropouts in any of the curves; and (5) There is no frequency above 30 kHz where all of the power lines exhibit minimal attenuation of 5-7 dB.

Overall, the graphs are consistent with expectations and can be segregated into three groups. The power line with the least attenuation is local phase B. Next in terms of attenuation are the local power lines, A and C. The highest attenuation levels are on remote A, B, and C lines.

An explanation provided in [1] for the equality of the attenuation for the three “remote” phases A, B, and C is that such equality is due to signal leak-throughs from electrical equipment connected to different phases, and from capacitive coupling on transformer windings. This statement is supported by [43] which states that the impedance of power lines is determined by two parameters, the loads connected to the network and the impedance of the distribution transformer.
2.2.2 Attenuation in Residential Buildings

A residential building is powered by a 110 V split single phase circuit. Because the two circuits are 180 degrees out of phase, one can be denoted as "phase 0," and the other as "phase 180." In the following three figures (Figures 2.5-2.7) the symbol "Ø 0" denotes a test condition where the receiver and the transmitter are both connected to the "phase 0" power line. Also, the symbol "Ø 180" denotes a test condition where the receiver is
connected to the "phase 0" power line, and the transmitter is connected to the "phase 180" power line.

Attenuation measurements for a residential building are shown in Figure 2.5. This figure shows that the cross-phase "Ø 180" test resulted in higher attenuation than the same-phase "Ø 0" test. Figure 2.5 also shows a same phase test with a resistive load attached to the power line. This test is denoted by the symbol "Ø 0 R Load." As expected, the attenuation is larger in the "Ø 0 R Load" test than in the "Ø 0" test. The change in attenuation is approximately 4 dB.

![Figure 2.5 Attenuation for residential in-building power line with resistive load [1]](image)

Results for cross-phase transmission in a residential building are shown in Figure 2.6. The symbol "Ø 180 R Path" denotes the cross-phase test with a resistive path connected between the "phase 0" and the "phase 180" power lines. As shown in Figure 2.6 there is a notable improvement of 3-5 dB in signal transmission capability when this resistive path is present.

Figure 2.7 shows attenuation results in a residential building for same-phase "Ø 0" and cross-phase "Ø 180" tests over an extended frequency range to 600 kHz. The roll off for the "Ø 0" test occurs at about 400 kHz and at about 200 kHz for the "Ø 180" test.
A notable difference between the industrial and residential attenuation graphs is that both the same-phase and cross-phase attenuations have larger bandwidths in a residence than in an industrial building. The residential power lines have less attenuation because the electrical loads draw less power than those in an industrial building.

2.3 NOISE

The results of a study of the error characteristics in an industrial setting are summarized in [5] for bit rates from 1.2 kbps to 38.4 kbps using a carrier of 115 kHz. Tables 2.1-2.3 contain statistics on error free lengths using a PSK modem [5].

To interpret Tables 2.1-2.3 consider Table 2.3. This table contains data for 4.8 and 1.2 kbps tests. Under the “1.2 kbps” heading of Table 2.3 are two tests that produced bit error rates of 9.6E-04 and 3.0E-03. The entries under the “9.6E-04” column yield the following conclusions: (1) — 25% of the error free lengths were of lengths at most 550 bits long; (2) — 25% of the error free lengths were of lengths 2000 bits and over; and (3) — 50% of the error free lengths were of lengths between 550 and 2000 bits.
2 Power Line Channels

Figure 2.7 Residential power line bandwidth [1]

Table 2.1 Error Free Lengths for 38.4 kbps, Industrial Building

<table>
<thead>
<tr>
<th>Percentage</th>
<th>38.4 Same Phase</th>
<th>38.4 Cross Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6.60E-04</td>
<td>1.30E-03</td>
</tr>
<tr>
<td>25%</td>
<td>1100</td>
<td>500</td>
</tr>
<tr>
<td>50%</td>
<td>1900</td>
<td>1200</td>
</tr>
<tr>
<td>75%</td>
<td>3000</td>
<td>2000</td>
</tr>
</tbody>
</table>

Table 2.2 Error free lengths for 19.2 kbps, industrial building

<table>
<thead>
<tr>
<th>Percentage</th>
<th>19.2 Same Phase</th>
<th>19.2 Cross Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4.9E-04</td>
<td>1.2E-03</td>
</tr>
<tr>
<td>25%</td>
<td>1000</td>
<td>600</td>
</tr>
<tr>
<td>50%</td>
<td>2000</td>
<td>1000</td>
</tr>
<tr>
<td>75%</td>
<td>3500</td>
<td>1700</td>
</tr>
</tbody>
</table>

Table 2.3 Error free lengths for 4.8 and 19.2 kbps, industrial building

<table>
<thead>
<tr>
<th>Percentage</th>
<th>4.8 kbps</th>
<th>1.2 kbps</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4.2E-04</td>
<td>1.4E-03</td>
</tr>
<tr>
<td>25%</td>
<td>600</td>
<td>370</td>
</tr>
<tr>
<td>50%</td>
<td>1300</td>
<td>600</td>
</tr>
<tr>
<td>75%</td>
<td>3100</td>
<td>1100</td>
</tr>
</tbody>
</table>

To further interpret this information, Table 2.4 was constructed. It contains all the information from Tables 2.1-2.3 but lists the data in order of bit error rate (BER). It also
includes a column for the percentage of single bit errors (i.e. of all the errors that occur: single, double, triple, etc., what percent are single). Note that a single bit error is one which is immediately preceded and followed by at least one correctly received bit.

<table>
<thead>
<tr>
<th>BER</th>
<th>Percentile</th>
<th>Percent of Single Bit Err</th>
<th>Test Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5E-04</td>
<td>2000</td>
<td>5000</td>
<td>98</td>
</tr>
<tr>
<td>4.0E-04</td>
<td>1200</td>
<td>2100</td>
<td>96</td>
</tr>
<tr>
<td>4.2E-04</td>
<td>600</td>
<td>1300</td>
<td>96</td>
</tr>
<tr>
<td>4.9E-04</td>
<td>1000</td>
<td>3100</td>
<td>99</td>
</tr>
<tr>
<td>6.6E-04</td>
<td>1100</td>
<td>3500</td>
<td>95</td>
</tr>
<tr>
<td>9.6E-04</td>
<td>550</td>
<td>2000</td>
<td>70</td>
</tr>
<tr>
<td>1.2E-03</td>
<td>600</td>
<td>1700</td>
<td>96</td>
</tr>
<tr>
<td>1.2E-03</td>
<td>650</td>
<td>1900</td>
<td>94</td>
</tr>
<tr>
<td>1.2E-03</td>
<td>900</td>
<td>2100</td>
<td>93</td>
</tr>
<tr>
<td>1.3E-03</td>
<td>500</td>
<td>2000</td>
<td>90</td>
</tr>
<tr>
<td>1.4E-03</td>
<td>370</td>
<td>1100</td>
<td>91</td>
</tr>
<tr>
<td>2.3E-03</td>
<td>300</td>
<td>900</td>
<td>94</td>
</tr>
<tr>
<td>2.8E-03</td>
<td>300</td>
<td>800</td>
<td>86</td>
</tr>
<tr>
<td>3.0E-03</td>
<td>220</td>
<td>610</td>
<td>84</td>
</tr>
<tr>
<td>3.0E-03</td>
<td>250</td>
<td>700</td>
<td>55</td>
</tr>
<tr>
<td>4.1E-03</td>
<td>230</td>
<td>500</td>
<td>83</td>
</tr>
<tr>
<td>4.8E-03</td>
<td>170</td>
<td>500</td>
<td>78</td>
</tr>
<tr>
<td>4.8E-03</td>
<td>170</td>
<td>500</td>
<td>76</td>
</tr>
<tr>
<td>5.2E-03</td>
<td>200</td>
<td>500</td>
<td>65</td>
</tr>
<tr>
<td>5.3E-03</td>
<td>120</td>
<td>400</td>
<td>70</td>
</tr>
</tbody>
</table>

Table 2.4 Summary of Tables 2.1-2.3 with single bit errors

From Table 2.4 it should be noted that the error free lengths for the 25, 50, and 75 percentiles decrease as the BER increases. This is as expected since the larger BER will cause more errors in the data. This larger number of errors implies that any two errors will be separated by a smaller number of bits.

The predominant type of error is the single bit error. In some cases single bit errors comprise 98 percent of all errors. In Table 2.4 it is interesting to compare entries 5 and 6. The former (38.4 kbps, BER of 6.6 \times 10^{-4}) has a single bit error percentage approaching 95% and the latter (1.2 kbps, BER of 9.6 \times 10^{-4}) has only 70% of single bit errors. This suggests that noise impulses must be of sufficient duration to corrupt the shorter 38.4 kbps bit but not the longer 1.2 kbps bit.
Many different types of appliances produce noise. Relatively low noise levels result from induction motors, florescent lighting, and other appliances at the frequency range up to 100 kHz when compared to the noise levels of light dimmers [6,43] and universal motors. Television receivers produce a significant amount of noise at harmonics of the 15734 Hz horizontal line rate [26].

A solid state light dimmer can be used with up to 100 watts of incandescent lighting to provide continuously variable lamp brightness. The dimmer is wired in series with the incandescent lights and controls lamp brightness by switching on and off rapidly through the use of triacs. Universal motors are found in vacuum cleaners, mixers, blenders, sewing machines, sanders, drills, and saws. Universal motors contain brushes, and their performance is similar to that of dc motors: when a load is placed on the motor, the speed decreases; and when the voltage to the motor is increased, the speed increases. The noise generated by motors with brushes has a random amplitude and frequency and can cause radio interference. The noise spectrum of the light dimmer is approximately 25 dB greater than the universal motor at 100 kHz [26,43].

It has been shown that forward error correction is an effective means to combat errors on power line communication channels [23].

2.4 SIGNAL FADING

At times, periodic 120 kHz signal fading is observed on power line channels [6]. Fading occurs on same-phase and cross-phase channels, with cross-phase channels occasionally producing severe fading. One cause of fading is rectifier circuits within power supplies. When a rectifier turns on, either once or twice during a 60 Hz cycle, it places a large capacitance directly across the power circuit. This causes the impedance seen by a power line coupled transmitter or receiver to change at a 60 or 120 Hz rate. Since the signal voltage is developed across this time varying impedance, the signal becomes amplitude modulated at a 60 or 120 Hz rate [43].

2.5 POWER LINE LOADING

The number of appliances connected to an electrical power distribution network constantly changes. Streetlights, air conditioning, electric public transport, and electric steel mills all contribute to the distribution network’s loading characteristics. Figure 2.8 shows the loading profile of a distribution network during a typical day [50]. This figure contains curves for industrial, commercial, domestic, and total system electric power.
loading. This figure shows periods of increased system power consumption at 12:00 pm, 3:00 pm, and 6:00 pm. Minimum power system power consumption occurs during the night from 2:00 am to 7:00 am.

![Graph showing load profile](image)

Figure 2.8 Distribution network load profile during one day [50]

Figure 2.8 indicates commercial power loading increases to its maximum at 10:00 am and remains constant until 5:00 pm when it decreases to its minimum. Typical commercial customers include retail businesses, financial institutions, and commercial office real-estate. Industrial customers tend to use the most power at 8:00 am. The industrial load also increases at 12:00 pm and 4:00 pm. Domestic users consume the largest overall amount of power. Figure 2.8 shows increases of load for domestic users at 8:00 am, 12:00 pm, and 6:00 pm which is when it attains its maximum.

A daily load profile [50] for Saturday through Friday for a distribution system is contained in Figure 2.9. This figure shows the hourly trends in Figure 2.8 occur consistently throughout the work-week. Power consumption is much less during weekend periods than during Monday to Friday.
The amount of noise and attenuation on a power line is generally related to the number and types of loads connected to the power line. An increase in loading will tend to increase the noise level and attenuation [43]. Although Figures 2.8 and 2.9 are loading profiles for entire distribution networks, the loading profiles of individual buildings can be expected to be similar in profile.
3 Hardware Architecture—Design and Implementation

3.1 OVERVIEW OF MODEM HARDWARE ARCHITECTURE

The microcontroller-based modem designed, implemented and tested as part of this thesis work is conveniently sectioned into two major parts: the receiver and the transmitter. Neither of these two parts can function alone. Each part needs the essential signals supplied by other common components. These support components are described in Section 3.2 and include the power supply, clock generator, control circuitry, and host interface. All schematics are found in Appendix A. The transmitter and receiver hardware design and implementation are described in Sections 3.3 and 3.4, respectively.

The modem was designed to facilitate digital implementation and eventual realization using VLSI technology. A modular architecture was developed, to enable selection or variation of important modem parameters, with minimal impact on most components and subsystems. Many operational changes are realizable by changing microcontroller software.

The following modem subsystems, described in detail below, are realized digitally: clock generator, microcontroller, host-modem interface, BPSK modulator (exclusive of the operational amplifier), Costas loop, bit synchronizer and integrate and dump detector. Analog subsystems include the line-coupling network, transmit power amplifier, transmit lowpass filter, receive bandpass filter and power supply. Some of these analog subsystems could be realized using ASIC technology.

3.2 TRANSMITTER AND RECEIVER SUPPORT COMPONENTS

3.2.1 Power Supply

The power supply shown in Figure A.9 uses a bridge rectifier to convert the power line AC 110 V to its positive and negative DC components. Three regulators convert these DC values to +5, +12, & -12 V which are appropriate modem power supply voltages.
3.2.2 The Clock Generator

The clock generator is shown in Figure A.4. Two signals are generated by the clock generator: \textit{Tx Data Clk} and \textit{Baud*24}. The \textit{Tx Data Clk} signal is used by the microcontroller to clock the transmit bit into the transmit latch of the modulator. The \textit{Baud*24} signal is used as a sampling signal by the integrate-and-dump filter circuitry (Figure 3.17). The clock generator generates the \textit{Baud*24} signal by dividing the 14.7456 MHz signal three times: \(14.7456 \text{ MHz} + 16 + 6 + (1, 2, 4, \text{ or } 8)\). The 74HCT164, 74HCT92, and 74HCT393 function as the three dividers.

3.2.3 A Description of the Microcontroller Control Pins

The purpose of this section is to provide an explanation of the control signals which interface to the microcontroller in the control unit. The control unit is shown in Figure A.2 and consists of three chips: an Intel 8-bit microcontroller 80C31BH-1; an 8-bit address latch 74HCT374; and a 128 kbit (16 kbyte) ultra-violet light erasable electrically programmable read only memory, 27128A EPROM. The microcontroller fetches instructions from the EPROM and executes them. The EPROM contains the microcontroller's firmware. The microcontroller is clocked by a 14.7456 MHz oscillator. Hence, the machine cycle of the microcontroller is \(14.7456 \text{ MHz} + 12 = 1.2288 \text{ MHz}\) because 12 clock cycles are required for the inner functioning of the microcontroller instruction cycle which involves fetch, decode and execute operations.

The microcontroller uses 18 pins to interface with the address latch and EPROM. These pins are:

- **Pins 39-32 (AD0-AD7):** contain the lower order byte of the address sent to the EPROM. These 8 lines also function as the microcontroller's data bus. Pins 39-32 are known as port 0 of the microcontroller.

- **Pins 21-28 (A8-A15):** contain the higher order byte of the address sent to the EPROM.

- **Pin 30 (ALE/\overline{P}):** contains the address line enable signal for the address latch. The address latch will output an address to the EPROM when this signal is high.

- **Pin 29 (PSEN):** contains the program store enable. The EPROM will output data onto the data bus when this signal is low.
The microcontroller uses 12 pins for controlling the modem hardware. These pins are:

- **Pins 2 & 8 (Baud0 & Baud1)**: control the baud rate. These two pins are connected to the multiplexor, 74HCT153, in the clock generator circuit (Figure A.4). Baud0 & Baud1 can select one of four baud rates: 19.2, 9.6, 4.8, or 2.4 kbps.

- **Pin 1 (Tx Bits)**: sends the current bit to the modulator.

- **Pin 4 (Rx/Tx*)**: controls the relay in the coupling network. If the signal is high/low, the transmitter is not connected/connected to the channel. In other words, if the Rx/Tx* signal is high/low, the modem is in receive/transmit mode.

- **Pin 7 (DCD)**: is set high when the microcontroller detects the BPSK data carrier. (Aside: In our modem, the carrier signal is 115.2 kHz.) The DCD signal is sent to the host via the host interface. Thus, the host may implement carrier detection in its communication protocols.

- **Pin 3 (Sync Detect)**: is set high whenever the microcontroller detects a synchronization sequence. The synchronization sequence is sent at the beginning of every packet as shown in Figure 4.7 in Section 4.3.1. The Sync Detect signal is used for debugging and performance testing purposes.

- **Pin 12 (Rx Bits)**: receives the demodulated signal from the bit synchronization circuitry.

- **Pin 14 (Baud*24)**: receives the baud × 24 signal from the clock generator circuitry. The Baud*24 signal is used to sample the Rx Bits signal as shown in Figure 3.17.

- **Pin 13 (Bit Sync)**: receives the bit synchronization signal from the bit synchronization circuitry. The Bit Sync signal is used to control the integrate and dump filter described in Section 3.4.4.

- **Pin 15 (Tx Data Clk or TxC (DTE))**: receives the transmit clock from the clock generator circuitry. This signal provides the bit timing for the modem transmitter.

- **Pin 5 (RTS)**: is reserved for Request To Send information and is currently unused.
Pin 6 (CTS): is used to send Clear to Send information to the host. The use of the CTS signal is described in Section 4.3.4.

The microcontroller uses 2 pins for communications with the host. These pins are:

- **Pin 11 (TxD):** is used for transmitting data (bytes) to the host in asynchronous format.
- **Pin 10 (RxD):** is used for receiving data (bytes) from the host in asynchronous format.

### 3.2.4 The Host/Modem Interface

The host and modem are interfaced to each other via a standard RS232 connection. The interface is depicted in Figure A.10. Two microelectronic chips are used to make the TTL-RS232 signal conversion; the 1488 chip converts TTL to RS232, and the 1489 chip converts RS232 to TTL. A DB25 connector is used for external access to the RS232 signals.

### 3.3 MODEM TRANSMITTER HARDWARE

The transmitter consists of several parts, including modulator, transmit lowpass filter, and power amplifier.

#### 3.3.1 BPSK Modulator

The modulator, shown in Figure A.4, gets its clocking signal *Tx Data Clk* from the clock generator, its data *Tx Bits* from the microcontroller, and the control signal *Rx/Tx* from the microcontroller. The modulator sends its output *Tx Signal* to the transmit low pass filter.

The 115.2 kHz BPSK carrier is generated as a staircase sine wave. The serial-in parallel-out shift register chip 74HCT164 receives a 1.8432 MHz clock signal and divides the signal by 16 to produce a set of 8 parallel waveforms. The waveforms are depicted in Figure 3.1. Each of these 115.2 kHz waveforms is passed through a resistor. As shown in Figure A.4 the outputs of the 8 resistors are summed. The output signal of the summing circuit is an approximation to a sine wave.
To select the values of the resistors to generate the sine wave carrier, consider Table 3.1. This table lists in its second column the sum of the unweighted 8 parallel waveforms denoted as $\text{sum}(a,b,...,h)=a[n]+b[n]+c[n]+d[n]+e[n]+f[n]+g[n]+h[n]$. We notice that $\text{sum}(a,b,...,h)$ increases for $n=0,...,8$ and decreases for $n=8,...,15$. The third column of Table 3.1 lists the values of a cosine signal: $-\cos(360n/16)+1$. This cosine signal has a behavior similar to $\text{sum}(a,b,...,h)$. The cosine signal increases for $n=0,...,8$ and decreases for $n=8,...,15$. If we assign a weight to each of the 8 parallel waveforms $a$-$h$, we can equate columns two and three of Table 3.1. The appropriate equation is shown below, where $A$-$H$ are the weights assigned to waveforms $a$-$h$ in Figure 3.1.


<table>
<thead>
<tr>
<th>$n$</th>
<th>$\text{sum}(a,b,...,h)$</th>
<th>$-\cos(360n/16)+1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>1,15</td>
<td>1</td>
<td>0.076</td>
</tr>
<tr>
<td>2,14</td>
<td>2</td>
<td>0.293</td>
</tr>
<tr>
<td>3,13</td>
<td>3</td>
<td>0.617</td>
</tr>
<tr>
<td>4,12</td>
<td>4</td>
<td>1.000</td>
</tr>
<tr>
<td>5,11</td>
<td>5</td>
<td>1.383</td>
</tr>
<tr>
<td>6,10</td>
<td>6</td>
<td>1.707</td>
</tr>
<tr>
<td>7,9</td>
<td>7</td>
<td>1.924</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>2.000</td>
</tr>
</tbody>
</table>

Table 3.1 Shift register waveforms: unweighted and weighted
Solving for the weights $A-H$ yields: $A=H=0.07612$, $B=G=0.216$, $C=F=0.3244$, and $D=E=0.3827$. The values for the resistors $R_a,...,R_g$ in Figure A.4 in terms of $R_h$ are as follows: $R_a=R_h$; $R_b=R_g=A/B=0.351R_h$; $R_c=R_f=A/C=0.234R_h$; and $R_d=R_e=A/D=0.199R_h$. The resistor $R_h$ is set to 100 kΩ. Figure 3.2 shows the synthesized sinusoidal waveform.

![Figure 3.2 Synthesized sinusoidal carrier](image)

### 3.3.2 The Transmit Low Pass Filter

Figure 3.3 shows an active second order Butterworth low pass filter which is also referred to as a Sallen-Key circuit [9]. The method used to calculate the values for the capacitors and resistors is described below.

![Figure 3.3 Second order Butterworth LPF](image)
For a second order filter $R_a$ and $R_b$ must satisfy the equation $Q = \frac{R_a}{2R_a R_b}$, for $Q=0.71$. A solution to this equation is $R_a = 8250 \, \Omega$ and $R_b = 4870 \, \Omega$. The capacitors $C_1$ and $C_2$ are set to 100 pf. The resistors $R_1$ and $R_2$ are frequency scaled to yield the desired cut-off frequency $f_o$ using the following relationships:

$$
R_{new} = k_m R_{old}
$$
$$
C_{old} = k_m k_f C_{new}
$$
$$
k_f = 2\pi f_o
$$

where $C_{old} = R_{old} = 1, f_o = 139 \, kHz, C_{new}=100pf$. Thus, $k_m = 11842$ and $R_1=R_2=R_{new} = 11.8 \, k\Omega$. The notation "old" in $C_{old}$ and $R_{old}$ denotes the values that the resistors and capacitors have before the above frequency scaling, and the notation "new" denotes the values of the components after frequency scaling. Figure A.7 shows the low pass filter with the above calculated values.

### 3.3.3 The Power Amplifier

The power amplifier circuit shown in Figure A.8 is taken from the application notes for the power amplifier LM384 [13]. The amplifier circuit has a variable resistor for changing the gain of the amplifier for performance tests described in Chapter 5. The amplified signal is sent from the amplifier first through the relay, then through the 1:1 transformer, and finally onto the power line. The relay is closed by the $Rx/Tx^*$ signal from the microcontroller. Because the relay must be activated by a high current, the $Rx/Tx^*$ signal is passed through the 7406 high-current driver chip shown in Figure A.8.

### 3.4 MODEM RECEIVER HARDWARE

The receiver consists of several sections which are described in the order which the signal flows through them from the power line channel. This order is the line coupling network, receive bandpass filter, Costas loop demodulator, bit synchronizer, and modem control unit.

#### 3.4.1 Line Coupling Network

The line coupling network is shown in Figure A.8. The received signal enters the modem though the AC plug. The signal passes through the 1:1 transformer to the receive band pass filter in Figure A.6.
3.4.2 Receive Band Pass Filter

The fourth order Butterworth filter [9] shown in Figure A.6 consists of two cascaded second order filters. A second order Butterworth filter is shown in Figure 3.4.

![Second order Butterworth band pass filter](image)

Figure 3.4 Second order Butterworth band pass filter

The two criteria used to select the resistor and capacitor values in the second order filter are bandwidth $BW=56 \text{ kHz}$ and centre frequency $f_0=115.2 \text{ kHz}$. The capacitors are conveniently fixed at 1 nf. Using the bandwidth and centre frequency criteria, the values $R_2=5.6 \text{ k} \Omega$ and $R_1=317 \Omega$ can be found from the following equations:

$$BW = \frac{1}{2\pi} \frac{2}{R_2 C} = \frac{1}{2\pi} \frac{2}{10^{-9}}$$

$$f_0 = \frac{1}{2\pi} \frac{1}{C \sqrt{R_1 R_2}} = \frac{1}{2\pi} \frac{1}{10^{-9} \sqrt{R_1 R_2}}$$

The resistors $R_a$ and $R_b$ are connected to the front of the second order BPF and implement a voltage divider such that $R_a//R_b = R_1$ or $R_1 = \frac{R_a R_b}{R_a + R_b}$. The choices $R_a = 2700 \ \Omega$ and $R_b = 360 \ \Omega$ are convenient and satisfy this equation.

3.4.3 The Costas Loop

The Costas loop shown in Figure 3.5 is a well-known method of retrieving a BPSK carrier and demodulating the signal simultaneously. Because there is no carrier component in the BPSK signal on which to lock, a conventional phase lock loop cannot be used. The frequency spectrum of the BPSK waveform is symmetric about its carrier frequency $f_c$ and this property is used to synchronize the locally generated carrier [10,25].

The BPSK carrier may also be recovered using a squaring loop [34]. With such a loop the received BPSK signal must be multiplied with the local carrier signal and filtered to obtain the baseband information signal.
The Costas loop has several variations. One variation involves the use of a hard limiter in the bandpass filter preceding the loop. This hard limiter replaces automatic gain control (AGC) circuits which are used to maintain a constant signal level into the receiver’s front end so that linearity of receiver operation is maintained [49]. Another variation involves the use of a hard limiter in the in-phase channel of the Costas loop [47]. This hard limiter allows the designer to replace the third multiplier (shown in Figure 3.5) with a chopper. The variation that was implemented in our modem uses hard limiters in both in-phase and quadrature-phase channels of the Costas loop. Before this variation is explained, the workings of the general Costas loop is described briefly.

As illustrated in Figure 3.5, quadrature components of the local carrier multiply the received signal which is then passed through a low pass filter to remove the frequency component at $2f_c$. The resulting quadrature signals are multiplied together to isolate the error signal $\phi_e - \sin(\phi_e)$. The voltage controlled oscillator (VCO) uses this error signal to adjust the local carrier phase.

An aberration of BPSK signalling is that there is a 180° phase ambiguity in the recovered carrier. Because of this uncertainty we cannot be sure whether we are receiving $m(t)$ or $-m(t)$ from the Costas loop. There are two ways of removing this ambiguity [10]. The first way is to send a known test sequence over the channel before the information packet so that a sense of the transmitted signal’s polarity can be determined. The second method is to use differential coding and decoding. The BER of differentially encoded BPSK is approximately 1dB worse than uncoded BPSK. Our modem uses a test sequence (or preamble) to determine data polarity and is discussed in Section 4.4.1.

The Costas loop demodulator can also be implemented digitally (Figure 3.6), with a digital loop filter chip replacing the LPF and VCO, and with choppers replacing the two multipliers labelled 1 and 2 in Figure 3.5. Multiplier 3 in Figure 3.5 is replaced with an XOR gate as illustrated in Figure 3.6. Figure 3.6 can be termed a digital Costas loop [11]. Figure A.5 contains the final implementation of the demodulator.
\[ s(t) = A_c m(t) \cos(w_c t) \]

\[ I_0 = A_0 \sin(w_c t + \phi_e) \]

\[ v_1(t) = \frac{1}{2} A_c A_o m(t) \cos(\phi_e t) \]

\[ v_2(t) = \frac{1}{2} A_c A_o m(t) \sin(\phi_e t) \]

**Figure 3.5 BPSK Costas loop demodulator**

**Figure 3.6 Digital Costas loop**

In Figure 3.6, we replaced multipliers 1 and 2 with two choppers. It will be shown next that for practical considerations this replacement is valid.

A chopper with an input \( s(t) \) that is triggered by a square wave \( c(t) \) produces an output which is identical to multiplying the signal \( s(t) \) by \( c(t) \) [10]. The square wave \( c(t) \) is periodic with period \( T_o \), and is represented as follows during the time interval \([ (k-1)T_o, kT_o ]\), where \( k \) is any integer:

\[ c(t) = \begin{cases} A, & (k-1)T_o < (k-\frac{1}{2})T_o \\ -A, & (k-\frac{1}{2})T_o < kT_o \end{cases} \]

Function \( c(t) \) has Fourier transform

\[ C(f) = \sum_{n=-\infty}^{\infty} C_n \delta(f-nf_o) \]
Hardware Architecture

where

\[ C_n = \begin{cases} \frac{-j A}{n \pi}, & n = \text{odd} \\ 0, & n \text{ otherwise} \end{cases} \]

The magnitude \(|C(f)| = \sum_{n=0}^{\infty} |C_n| \delta(f-nf_o)| is shown in Figure 3.7.

![Figure 3.7 Spectrum of square wave c(t)](image)

Because the received signal has a bandwidth less than \(f_o\), the spectrum \(W(f)\) of the output \(w(t)\) of the chopper is

\[ w(t) = c(t) * s(t) \Leftrightarrow W(f) = C(f) * S(f) = \sum_{n=-\infty}^{\infty} C_n \delta(f-nf_o) * S(f) \]

It follows that

\[ W(f) = C_1 S(f-f_o) + C_3 S(f-3f_o) + C_5 S(f-5f_o) + \ldots \]

The first term \(C_1 S(f-f_o)\) is exactly that which occurs when multiplying \(s(t)\) by the carrier signal of frequency \(f_o\) using an analog multiplier. The higher order components are removed by the LPF. Therefore, replacing multipliers 1 and 2 in Figure 3.5 with the two choppers is a valid replacement.

The loop filter chip 74HCT297 is the heart of the carrier tracking loop. The functionality of the chip is shown in Figure 3.8. The modulo counter (random walk filter [33]) is triggered by the clock signal and is controlled by the \(\text{up/down}\) signal. If the counter overflows, then the \(\text{ins}\) line is set. If the counter underflows, the \(\text{del}\) line is set. The insert/delete circuit is triggered by the clock signal. This circuit outputs pulses at a rate of \(\text{clock} + 2\). When the \(\text{ins}\) line goes high the insert/delete circuit inserts an extra pulse in
its output. When the del line goes high the insert/delete circuit deletes a pulse from its output. Figure 3.9 shows the clock/2 pulses signal with an inserted and a deleted pulse.

![Diagram of loop filter chip](image)

Figure 3.8 Block diagram of loop filter chip

![Inserted and deleted pulses of the loop filter chip](image)

Figure 3.9 Inserted and deleted pulses of the loop filter chip

The bandwidth of the loop filter chip is set by the four input control lines of the counter. These lines are able to set the modulus \(m\) of the counter to values between \(2^3\) and \(2^{17}\). Because the \(\text{clock}=14.7456\ \text{MHz}\), the maximum number of inserted and deleted pulses is

\[
\frac{14.7456 \times 10^6}{m} = \text{extra pulses/second.}
\]

Therefore the range of pulses generated is

\[
\frac{\text{clock}}{2} \pm \frac{\text{extra pulses/second}}{2} = \frac{14.7456 \times 10^6}{2} \pm \frac{14.7456 \times 10^6}{m}
\]

Since the output of the the loop filter chip is divided by 64 (Figure A.5), the local carrier square wave will have a frequency range of

\[
115.2 \cdot \left(1 \pm \frac{2}{m}\right) \text{kHz}
\]

Thus, the bandwidth can be chosen within the range of 1.75 Hz to 28.8 kHz. A small (or narrow) bandwidth will track the carrier very closely producing a small phase jitter, and performs well under low SNR conditions. Conversely, a large bandwidth will produce a local carrier with a large amount of phase jitter. There is one disadvantage to a
very narrow bandwidth; a carrier recovery loop with a very narrow bandwidth will take a long time to lock onto the signal. To choose the best bandwidth one must compromise between phase jitter and lock (or acquisition) time. The amount of phase jitter can be estimated by recording the modem's BER performance under noisy conditions. These BER results are shown in Figure 3.10. Similarly, the length of acquisition time can be estimated by tabulating the percentage of packets which are lost (see Section 5.1) as shown in Figure 3.11.

![Figure 3.10 BER vs loop filter bandwidth](image1)

![Figure 3.11 Percent lost packets vs loop filter bandwidth](image2)
Figure 3.10 shows that BER stays fairly constant at $8 \times 10^{-4}$ until the loop bandwidth is increased above 7200 Hz, at which point the BER increases rapidly with further bandwidth increase. Figure 3.11 shows that bandwidth values between 1800 and 7200 Hz produce minimal lost packet percentages. Based on these two results, a bandwidth of 1800 Hz was selected.

As shown in Figure 3.5, the in-phase arm of the IQ demodulator will produce the demodulated bit stream. This bit stream (Rx Bits signal) is used as an input to the bit synchronization circuit.

### 3.4.4 Bit Synchronization Circuit

It is assumed in addition to carrier phase and frequency knowledge that the receiver has accurate knowledge of when an incoming symbol started and when it is finished. This knowledge is required in order to know the proper symbol integration interval. Clearly if the receiver integrates over an interval of an inappropriate length, or over an interval that spans two symbols, the ability to make accurate symbol decisions will be degraded [25].

Timing information is usually derived from the data signal itself and based on some meaningful optimization criterion which determines the steady-state location of the timing instants. A distinction can be made between three different kinds of methods [27]. The first class of synchronizers is transition based. The threshold crossings of the received baseband signal are compared with the sampling phase. A correction of the sampling phase is initiated as a result of this comparison. The mean location of the crossings is estimated and the optimum sampling instant and maximum eye opening are assumed to be halfway between these crossings. These synchronizers are also called early-late or split-gate synchronizers [23].

The second class of synchronizers uses a signal derivative at the sampling instances. This derivative, or at least its sign, is usually correlated with the estimated data to produce the updating information required for the timing loop. These synchronizers include data-directed and decision feedback loops

A third class of symbol synchronizers uses nonlinear processing to generate and filter out a spectral line at the clock frequency. The nonlinearity is generally supplied by a square law device.
The early-late gate is one of the most popular symbol synchronizers because of its simplicity and is the synchronizer of choice here. The all-digital bit synchronization circuit appears in block diagram form in Figure 3.12 and is detailed in Figure A.3. Often a sequential filter (also known as a random walk filter) is inserted between the phase detector and counter [28,33] to narrow the synchronizer's bandwidth.

**Figure 3.12** Bit synchronization circuit

Figure 3.13 contains the waveforms of Figure 3.12 when the **Rx Bits** and **Bit Sync** signals are not synchronized. The vertical dashed lines denote the local bit boundaries set by the falling edges of the **Bit Sync** signal. The bit synchronization circuit synchronizes these bit boundaries with transitions in the **Rx Bits** signal. The left half of Figure 3.13 shows a received bit of the **Rx Bits** signal that is "early." The bit synchronization circuit will respond to the "early" bit by setting the **up** signal. When the **up** signal is high the counter outputs the **Bit Sync** signal at a slightly higher frequency. Thus, the **Bit Sync** signal will "move" to the left until the **Bit Sync**'s boundaries match the boundaries of the received bits in the **Rx Bits** signal. The right half of Figure 3.13 depicts the reverse situation whereby the received bit is "late."

**Figure 3.13** Non-synchronized bit stream

A perfectly synchronized **Bit Sync** signal is shown in Figure 3.14.
The bit synchronization circuit adjusts the Bit Sync signal by stretching or shrinking the high part of the Bit Sync signal. The Bit Sync signal can be one of the waveforms shown in Figure 3.15 (the numbers 11, 12, & 13 represent the number of cycles of the Baud*24 signal shown in Figure 3.12). The Bit Sync signal can be either 23, 24 or 25 cycles of the baud*24 signal which implies that the synchronizer's bandwidth is approximately 0.08 times the bit rate. Thus, the Bit Sync signal will be perfectly synchronized within 12 transitions of the Rx Bits signal.

Because the Bit Sync signal can be synchronized even under noisy conditions as shown in Figure 3.16, the recovered bit clock should not appreciably degrade the performance of the receiver.

The Rx Bits and the Bit Sync signal are processed by the receiver's final stage which is the integrate-and-dump filter.
3.4.5 Integrate-and-Dump Filter and Host Interface

The digital integrate-and-dump filter [10] is located inside the microcontroller and is shown in Figure 3.17.

![Integrate and dump matched filter](image)

Figure 3.17 Integrate and dump matched filter

This filter is all-digital and is implemented via hardware and firmware using the three input signals: \( Rx\, Bits, \) \( Baud*24, \) and \( Bit\, Sync. \) The first two signals are connected to Timer 0. (Two hardware timers are located in the microcontroller: Timer 0 and Timer 1.) Timer 0 is set to a mode for measuring the length of pulses [18]. This mode causes Timer 0 to be clocked by \( Baud*24 \) whenever the \( Rx\, Bits \) signal is high. Timer 0 is read and reset to zero by the firmware on the falling edge of the \( Bit\, Sync \) signal. The effect of this procedure is to sample the \( Rx\, Bits \) signal 24 times per bit interval. Section 4.4 describes the method used to determine whether the "received" value read from Timer 0 is a 1 or 0.

The microcontroller buffers up eight "received" bits and sends them to the host on the \( TxD \) line shown in Figure A.2. The bits are sent at a baud rate of 38.4 kbps using asynchronous serial communication.
4 Firmware Design and Implementation

4.1 OVERVIEW OF MODEM FIRMWARE ORGANIZATION

Use of a microcontroller provides flexibility needed to easily vary important modem parameters, without alterations to hardware. Such parameter changes are realizable using software resident in a host microcomputer interfaced to our modem.

An important design consideration is the allocation of data link layer functions between the host and microcontroller. Our division of tasks maximizes the flexibility of the packet structure, while keeping communication-related tasks performed by the host to a minimum. Packet flexibility is useful, to enable service of different communication applications.

The microcontroller uses random access memory (RAM) and read only memory (ROM). This RAM consists of 128 bytes located on the microcontroller. The microcontroller stores its program variables in this RAM. The ROM, located on the EPROM, consists of 16 kbytes. The microcontroller reads instructions from the EPROM and executes them. These instructions comprise the modem's firmware.

The firmware code is separated into the initialization module, the transmit module, and the receive module. The initialization module initializes the modem's parameters, the transmit module sends bits to the modulator, and the receive module receives bits from the demodulator. This chapter describes these three modules in detail. The term "mode" is used to describe the fact that the microcontroller is executing instructions found in a certain module. Figure 4.1 illustrates the three modes in the firmware.

The phrases "data is sent up to the host" and "data is sent down from the host" are used throughout this chapter. The phrases originate from the Open Systems Interconnection (OSI) reference model (see Figure 4.2) within which the modem occupies the lowest layer and the host occupies the higher layers [10]. When the host wants to transmit a packet it sends the packet "down" to the modem. The modem receives this packet and transmits it over the power line. Similarly, when the modem receives a packet from the power line, the modem sends the packet "up" to the host.
Channel communication between transmitter and receiver can be one of three types, asynchronous, synchronous, and intermittent synchronous [24]. An asynchronous bit pipe is one in which individual characters are framed (start, stop bits). This asynchronous technique is not useful for very high data rates. A synchronous bit pipe is one where bits within a character are sent at a fixed rate but successive characters can be separated by variable delays. Dummy bits are sent during idle time. This synchronous technique is useful when high data rates are not important. An intermittent bit pipe is one where the data link layer of the data terminal equipment (DTE) supplies bits synchronously to the modem when it has data to send, but supplies nothing when it has no data. Thus the modem must
distinguish between 0, 1, and idle. The capability to transmit nothing is very important for multi-access channels [24].

4.2 MODEM INITIALIZATION MODULE

The modem contains many parameters which may be initialized from the host computer. This initialization must be performed at the beginning of a communication session. To initialize the modem, the host sends the modem a data link escape character followed by an "I". These two characters are represented as "DLE I". When the modem receives "DLE I" it enters the initialization mode. In this mode commands shown in Figure 4.3 are sent between the modem and its host. The command "B", sent from the host, changes the baud rate at which bits are transmitted over the power line by the modem. The rates available are 19.2, 9.6, 4.8, and 2.4 kbps. The parameters accessed by the commands "C", "S", and "K", sent from the host, are explained in later sections of this chapter. The commands "?", "Q", "P", and "V" send information from the modem to the host. When the modem receives the "R" command from the host, the modem leaves the initialization mode and returns to the receive mode.

Figure 4.3 Commands sent to the modem by the host during initialization

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>print help information</td>
</tr>
<tr>
<td>C</td>
<td>set data carrier detection threshold</td>
</tr>
<tr>
<td>S</td>
<td>set synchronization word</td>
</tr>
<tr>
<td>L</td>
<td>set data link escape character</td>
</tr>
<tr>
<td>Q</td>
<td>print user selectable parameters</td>
</tr>
<tr>
<td>K</td>
<td>set the start of packet indicator</td>
</tr>
<tr>
<td>B</td>
<td>set the baud rate</td>
</tr>
<tr>
<td>P</td>
<td>print the baud rate</td>
</tr>
<tr>
<td>V</td>
<td>print the firmware version number</td>
</tr>
<tr>
<td>Z</td>
<td>reset the modem</td>
</tr>
<tr>
<td>R</td>
<td>leave the initialization mode and go to receive mode</td>
</tr>
</tbody>
</table>

4.3 MODEM TRANSMIT MODULE

Figure 4.1 showed that the modem makes a decision when it receives two characters from the host. If the two received characters are "DLE I", the modem switches to initialization mode. If the two received characters are not "DLE I" the modem switches to transmit mode.
The transmit module performs physical layer as well as data link layer functions. The physical layer functions are performed by manipulating hardware signals. The data link layer functions are performed entirely within the microcontroller. Figure 4.4 shows the data link layer functions which the microcontroller performs. Figure 4.5 illustrates the transmit module's physical layer functions. Because the data link functions are much more complex than the physical layer functions, discussion of the transmit module will concentrate on the data link layer.

![Figure 4.4 Transmit module data link layer functions](image)

![Figure 4.5 Transmit module physical layer functions](image)

A block diagram of the transmit module is shown in Figure 4.6. As seen in this figure the first function performed by the modem is to reset the Rx/Tx* pin to the low logic level. When the Rx/Tx* pin is low the modem's hardware transmit power amplifier is connected to the power line. The next function performed by the modem is to transmit a preamble data sequence over the power line. While the modem is transmitting the preamble the host sends data bytes down to the modem. These bytes are received by the modem and are stored in its transmit buffer. When the modem finishes transmitting the preamble, it transmits the data stored in its transmit buffer. As shown in Figure 4.4, the host continues to send down bytes to the modem while the modem is transmitting data from the transmit buffer.

### 4.3.1 Transmitted Packet Format: Preamble and Data

The preamble is modulated onto the powerline with the data following immediately thereafter. A byte is defined to be 8 bits long, and a word is defined to be 16 bits long. In the present version, V2.17, of the firmware, 9 bytes of preamble are sent. These bytes consist of three and one-half synchronization words (SYNC) and one start of packet indicator word (STRT) as shown in Figure 4.7. A word contains two bytes, a most significant (ms) and least significant (ls) byte. The "one-half SYNC word" is the least
significant (ls) byte of SYNC. Section 4.4.2 includes more discussion of preamble length and SYNC and STRT sequences. The data field of Figure 4.7 can be any number of bytes in length.

The modem does not implement higher data link operations such as CRC, addressing, flags, packet length indicators, and character (or bit) stuffing. These functions are implemented in the host. The data field in Figure 4.7 will contain the packet format implemented by the host. The modem is almost completely transparent to the host. The only restriction of the data field content (Figure 4.7) is the first two bytes. These two bytes are described in Section 4.2. Because of this high degree of transparency, the modem can transmit and receive any type of data including text and binary formats. Transparency is important because the higher the order of transparency the lower the amount of additional overhead on a packet due to character and bit stuffing.

4.3.2 Host-Modem Communication

From Figure 4.4 it was seen that the host sends data down to the modem while the modem transmits data over the power line. The modem buffers the data in its transmit buffer and will transmit this data at a later time. The modem is capable of transmitting data over the power line at rates of 2.4, 4.8, 9.6, and 19.2 kbps. The host sends data down to the modem at 38.4 kbps using an asynchronous serial communication format. This format consists of one start bit, eight data bits, and one stop bit. Taking this format overhead into consideration, the maximum throughput of the host-modem link is reduced to 30.27 kbps.

As was mentioned earlier, the modem contains only 128 bytes of RAM. Of this 128 bytes only 40 are used for the transmit buffer. This buffer is used in a circular fashion as shown in Figure 4.8. The “buffer occupied” portion of the buffer contains bytes that are yet to be transmitted over the power line. The arrow labelled “1” in Figure 4.8 indicates the direction that the “start of data” arrow moves as data is removed from the circular buffer. The arrow labelled “2” in Figure 4.8 shows the direction that the “end of data” arrow moves as data enters the buffer.
reset $Rx/Tx^*$ pin

transmit preamble
over power line and
buffer data received from
host in transmit buffer

remove byte from transmit buffer

set bit_count = 8

received byte from host?

buffer byte received from host in
transmit buffer

send bit to modulator on low edge
of $Tx Data Clk$

decrement bit_count

is bit_count = 0?

is transmit buffer empty?

*set $Rx/Tx^*$ pin

Figure 4.6 Transmit module block diagram

Is byte | word | word | word | word | 1 or more bytes
---|---|---|---|---|---
SYNC | SYNC | SYNC | SYNC | STRT | DATA

Figure 4.7 Transmitted packet format
When the circular transmit buffer becomes full, two options are possible. The first option is for the modem to stop placing data in the transmit buffer, and to ignore any data bytes sent down by the host. The second option is for the host to stop sending data bytes down to the modem. Of the two options, the second is preferred. To implement this second option the modem must inform the host whenever the transmit buffer becomes full. This capability called handshaking is discussed in the next section. A useful generalization of this capability results when the modem has complete control over the host-modem link. This generalization is discussed in Section 4.3.4.

4.3.3 XON/XOFF Host-Modem Flow Control

The XON/XOFF protocol is a common method of implementing host-modem handshaking. Two control bytes XON and XOFF are sent up to the host by the modem. XON means "transmission ON," and XOFF means "transmission OFF." When the modem’s transmit buffer becomes full, the modem sends XOFF up to the host. Similarly, when the modem’s transmit becomes empty, the modem sends XON up to the host.

This protocol has two important disadvantages. The first disadvantage is that the protocol malfunctions whenever the XON/XOFF bytes are not received by the host. The second is that the protocol is not able to alter the transfer rate of the host-modem link. The first disadvantage became evident during testing when it was noticed that the host personal computer did not always receive the XON/XOFF control bytes. A missing XON byte caused the modem’s transmit buffer to empty prematurely, and a missing XOFF byte caused the modem’s transmit buffer to overflow. The second XON/XOFF protocol disadvantage became evident when the modem was tested at 2400 bps. At 2400 bps the modem missed some of the data bytes sent down from the host. This problem occurs because the host sends data bytes down to the modem faster than the modem can buffer
them. The modem is capable of buffering at a rate of 2400 bytes per second. In Section 4.3.2, the host-modem link was calculated to be 30.27 kbps which is equal to 3784 bytes per second. Since the host sends data bytes down to the modem at 3784 bytes per second, and since the modem buffers these bytes at 2400 bytes per second, data bytes are lost and the packet is corrupted.

To overcome the XON/XOFF protocol impairments, a new handshaking protocol was developed. This new protocol is described in the next section.

4.3.4 Synchronous Handshaking for Asynchronous Communication

The two disadvantages of the XON/XOFF protocol discussed in Section 4.1.3 are resolved by using a hardwired handshaking protocol. The CTS line was chosen to implement this protocol.

In the host computer, there is an indicator bit called Delta Clear to Send (DCTS). This indicator bit is located in the Modem Status Register (MSR) of the host's serial port. The DCTS bit is set whenever the CTS line changes state. The bit is reset whenever the host computer reads the Modem Status Register.

There are two rules in the protocol to which the host and modem adhere. The modem toggles the CTS line whenever it is able to receive a data byte from the host. The host must only send a data byte down to the modem if its DCTS bit is set.

The hardwired handshaking protocol is able to avoid the two disadvantages of the XON/XOFF protocol. The modem's transmit buffer will never overflow because the host will only send data bytes whenever the CTS line is toggled. Whenever the modem's buffer becomes full, the modem stops toggling the CTS line. The second disadvantage of the XON/XOFF protocol is solved because the modem is able to control the rate at which data is sent down from the host. The modem toggles the CTS line at the same rate at which it is able to buffer data bytes sent down from the host.

There are two possible problems which could cause the hardwired protocol to fail. The first problem is a failure by the host of detecting one or more toggles. This is in fact not a problem for the following reason. Whenever the host detects a toggle, the host will send a byte of data down to the modem. If the host does not detect a toggle the host will not send down a data byte. The modem will not notice this lack of data because the modem's buffer already contains several bytes of data which must be transmitted. A
second potential problem is the detection by the host of a spurious toggle. This is a more serious problem because the host may send down a byte of data to the modem before the modem is able to accept it. Thus it is possible for a data byte to be lost on a spurious toggle. The chance of a spurious toggle occurring is very remote and will occur only if the modem/host hardware malfunctions. Neither of these two problems have been detected in testing.

The above approach to host-modem communication is advantageous over purely synchronous communication because the timing is not rigid. As long as the host sends data to the modem at a rate high enough to prevent the modem's transmit buffer from emptying, the protocol will work as required. Another advantage is the fact that the host-modem communication rate is much higher than the power line baud rate. For purely synchronous communication the host-modem communication rate is equal to the power line baud rate. This difference in host-modem communication rates would translate into time saved by the host. This time could be used by the host to perform other processing tasks.

4.4 MODEM RECEIVE MODULE

As shown in Figure 4.1, the receive module is the default module. This means that after the modem finishes transmitting or initializing, the modem returns to receive mode.

Like the transmit mode, the receive module performs physical and data link layer functions. Figure 4.9 shows the receive module's physical layer functions. Figure 4.10 illustrates the data link layer functions performed by the receive module.

![Diagram of receive module physical layer functions](image)

Figure 4.9 Receive module physical layer functions
The physical layer performs two functions shown in Figure 4.9. The integrate-and-dump matched filter was described from a hardware perspective in Section 3.3.4. In firmware terms, a counter acts as an integrator. The dumping action of the filter is realized by "reading" and subsequently zeroing it. The table look-up function implements thresholding required to translate dump values from the integrate and dump matched filter into "received" bits. The thresholding procedure is

\[
\text{bit} = \begin{cases} 
1, & \text{value} > 12 \\ 
0, & \text{value} \leq 12 
\end{cases}
\]

where value is in the range 0 to 25.

4.4.1 Data Link Layer

As shown in Figure 4.10, the data link layer receives a bit stream from the physical layer. The bit stream is analyzed in order that bytes and packets can be extracted and sent to the host. To perform these extractions the data link layer was designed as a real-time finite state automata containing four states. The four states are Sync_Search, Sync_Verify, Sync/Strt, and Lock shown in Figure 4.11. Item SR in Figure 4.11 is a 16-bit shift register that contains 16 bits from the received bit stream. The items SYNC and STRT are explained in Section 4.3.1.

The complexity of the automata is limited by two microcontroller restrictions. These include microcontroller instruction operand width and instruction execution speed. The microcontroller is only able to process operands which are a byte wide. Thus, any word-wide operations must be converted to byte-wide operations. The microcontroller has an instruction frequency of 1.2288 MHz. When the modem is operating at 19.2 kbps, the microcontroller can execute a maximum of 64 1-cycle instructions. Thus, the automata must perform its functions within 64 instruction cycles.
The first state of the automata in Figure 4.11 is Sync_Search. In this state the microcontroller continuously searches for the 16-bit synchronization word SYNC in the bit stream. When SYNC is detected with three or fewer bit errors, the microcontroller switches to state Sync_Verify. In this state the microcontroller buffers 16 bits in shift register SR and compares SR with SYNC. This comparison is to verify that the microcontroller has correctly synchronized itself to the synchronization pattern in the preamble. A single bit error is allowed in this verifying comparison. This state also determines whether the bits are inverted and sets an inversion flag to this effect. Bits may be inverted because of the 180° phase ambiguity discussed in Section 3.4.3. Upon successful SYNC verification, the microcontroller switches to Sync/Strt state. As shown in Figure 4.11 the microcontroller buffers 16 bits in SR. SR is compared against SYNC and STRT while allowing one bit error. If SR contains SYNC the microcontroller stays in Sync/Strt, but if SR contains STRT the microcontroller switches to the Lock state.

Upon entering the Lock state, the microcontroller has accomplished the first two functions of Figure 4.10: byte synchronization and preamble stripping. The Lock state performs the final function of packet delimiting. Figure 4.12 shows the received data and its delimiters all of which are sent up to the host. SOP and EOP are the Start of Packet and End of Packet delimiters, respectively, as detailed later in Section 4.4.2. The flow chart of Figure 4.11 does not show the delimiting action of the Lock state. The SOP delimiter is attached when the state switches from Sync/Strt to Lock, and the EOP delimiter is attached when the state switches from Lock to Sync_Search. Otherwise, the functioning of the
Lock state is as shown in Figure 4.11. The microcontroller buffers 8 bits in the the lower half of the shift register SR_LOW and sends SR_LOW to the host if the data carrier is detected.

<table>
<thead>
<tr>
<th>SOP</th>
<th>DATA</th>
<th>EOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>word</td>
<td>1 or more bytes</td>
<td>word</td>
</tr>
</tbody>
</table>

Figure 4.12 Packet sent to host

4.4.2 Discussion of Data Link Layer Parameters

The description of the receive module's data link layer in Section 4.2.1 contains many parameters that were fixed when the finite state automata was being designed. The parameters in the automata were selected such that the functions of the data link layer are as powerful as possible within the limits imposed in Section 4.2.1.

The length of the preamble was selected after consideration of several factors. The first consideration is packet overhead. To reduce packet overhead, it is necessary that the preamble contain as few bits as possible. The second consideration is reliable data link layer performance. The data link layer must be able to recognize a high percentage of incoming packets, facilitate byte and packet synchronization, and properly strip the preambles. Any faulty behavior of the data link layer will result in packet corruption and loss of channel throughput. A third consideration is host processing time. The host is busy performing a variety of tasks, of which only one is communication with its modem. The data link layer of the modem must monitor the bit stream coming from the physical layer and must pass only valid packets up to the host. Any random bits in the bit stream which have the appearance of valid data must be deleted. The host will then spend a minimum amount of time processing data arriving from its modem, and will be able to dedicate itself to other tasks.

The first consideration in the above paragraph identifies the need for a short preamble. The finite state automata discussed in Section 4.2.1 requires that the preamble contain a minimum of two SYNC words followed by a single STRT word. The data link layer was tested with preambles containing two SYNC words plus incremental numbers of additional bits from 0 to 48. These additional bits are segments of the SYNC words (ie. 4 additional bits means the least significant nibble of the SYNC word is transmitted). A direct relationship was found between the number of additional SYNC bits in the preamble and the probability that the automata is able to synchronize to the preamble as shown in
Figure 4.13. Since there is little improvement in the performance of the automata for preambles containing more than 24 additional bits, the preamble as shown in Figure 4.7 was chosen to contain three and a half SYNC words.

![Figure 4.13 Performance of data link layer](image)

Considerations two and three motivated the use of 16-bit SYNC and STRT identities. Sixteen-bit words are processed as two separate bytes by the microcontroller because of the byte-wide operand restriction discussed in Section 4.2.1. The SYNC word was chosen to be Ox3ca9, where Ox denotes hexadecimal, and the STRT word was chosen to be Ox3c56. The binary representation of SYNC is 0011110010101001 and the binary representation of STRT is 0011110001010110. The SYNC word contains patterns of alternating ones and zeroes which is desirable for demodulator and hardware synchronization circuits. The STRT word has the beneficial property that eight error bits are required to turn STRT into SYNC. The SYNC word was also chosen because of its desirable correlation feature. A good synchronization codeword is one that has the property that the absolute value of its “correlation sidelobes” is small. A correlation sidelobe is the value of the correlation of a codeword with a time-shifted version of itself [23,25]. This correlation is shown in Figure 4.14. The word Ox3ca9 was identified via an exhaustive computer search that selected 16-bit words based on correlation and number of 1-0 and 0-1 transitions.
Considerations two and three were used to decide on the number of bit errors allowed for a preamble word in each state of the automata. Sync_Search allows three bit errors, Sync_Verify allows 1 bit error, and Sync/Strt allows 1 bit error.

Two probabilities characterize the performance of a system using a synchronization word. These are the probability of a missed detection and the probability of false alarm [25]. Clearly, the system designer would wish both probabilities to be as small as possible. These are conflicting objectives. In order to decrease the probability of a miss, the system designer may allow less than perfect correlation of an incoming synchronization word. That is, a word may be accepted even if it contains a small number of errors. This, however, enlarges the number of symbol patterns that will be accepted and thereby increases the probability of a false alarm.

We can evaluate the data link layer performance using these considerations. To perform this evaluation, we determine the rate at which the automata falsely triggers on a randomly noisy bit stream. To find this rate we first calculate the total number $N_i$ of random bit error patterns of length 16 bits which can successfully pass through the individual states of the data link layer, as follows:

$$N_1 = 2 \left[ \binom{16}{3} + \binom{16}{2} + \binom{16}{1} + \binom{16}{0} \right]$$

$$= 1394 \text{ for Sync_Search State}$$

$$N_2 = 2 \left[ \binom{16}{1} + \binom{16}{0} \right]$$

$$= 34 \text{ for Sync_Verify State}$$
\[ N_3 = \left\lceil \frac{16}{1} + \binom{16}{0} \right\rceil \\
= 17 \text{ for Sync/Strt State} \]

The total number of patterns which will match the preamble is \((1394)(34)(17) = 8 \times 10^5\). The total number of patterns possible are \((2^{16})(2^{16})(2^{16}) = 2.8 \times 10^{14}\). Therefore, the probability of a random bit sequence matching the preamble is \(2.84 \times 10^{-9}\). Thus, on average the data link layer would detect a false preamble once in \(3.5 \times 10^8\) random bits. The mean time between false preambles for different baud rates is tabulated in Table 4.1. This table shows that a modem operating at 19.2 kbps will produce false preambles every 5 hours, on average. This mean time is large enough to fulfill the reliability requirements described as consideration two above.

<table>
<thead>
<tr>
<th>baud rate (kbps)</th>
<th>seconds</th>
<th>minutes</th>
<th>hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.2</td>
<td>18229.2</td>
<td>303.8</td>
<td>5.1</td>
</tr>
<tr>
<td>9.6</td>
<td>36458.3</td>
<td>607.6</td>
<td>10.1</td>
</tr>
<tr>
<td>4.8</td>
<td>72916.7</td>
<td>1215.3</td>
<td>20.3</td>
</tr>
<tr>
<td>2.4</td>
<td>145833.3</td>
<td>2430.6</td>
<td>40.5</td>
</tr>
</tbody>
</table>

Table 4.1 Mean time between false alarms

Figure 4.12 shows that two delimiters SOP and EOP are sent up to the host. The host uses the delimiters to identify packets within its own buffers. SOP and EOP were each chosen to be 16 bits long. Because a packet contains random data, it is better in terms of probability of spurious delimiter occurrence to use a 16 bit delimiter instead of an eight bit delimiter. The probability that 16 bits of random data is a delimiter is \(2^{-16} = 0.000015\) whereas the probability that 8 bits of random data is a delimiter is \(2^{-8} = 0.0039\). The SOP and EOP were arbitrarily chosen to be 0x3738 and 0xa3a4, respectively. Because SOP and EOP are generated by the modem and are not transmitted over the power line, the probability that a delimiter contains a bit error is very small. In fact, the probability of a delimiter bit error is equal to the probability of a bit error in RS232 communication.

4.4.3 Carrier Detection

As shown in Figure 4.11 the finite state automata switches from Lock state to Sync_Search state if a data carrier is not detected. Carrier detection is performed within the Lock state. This detection is based on a computed value called confidence. When confidence is high the modem is “confident” that a carrier signal is present on the power line. The DCD (Data Carrier Detect) pin is set high whenever confidence is high. The
DCD signal is connected to the host to enable the host to use protocols that require carrier detection.

Figure 4.9 shows that the physical layer produces a confidence value for every bit sent to the data link layer. A bit's confidence value is calculated from the dump values produced by the integrate and dump matched filter as shown in Figure 4.9. The relationship between confidence and dump values is confidence = |12-dump|. Figure 4.15 shows a bit which has a high confidence value and a bit which has a low confidence value. The confidence value of the high confidence bit in Figure 4.15 is 11 = |12-23|, and the confidence value of the low confidence bit is 1 = |12-13|.

![Figure 4.15 High and low confidence bits](image)

...
of 0 to 254. *old_confidence* was the calculated *new_confidence* for the previous bit. The value for *new_confidence* is calculated for every bit.

Previously it was said that the *DCD* signal is set whenever *confidence* is “high.” The term “high” is subjective. *confidence* is said to be “high enough” whenever *new_confidence* exceeds the quantity *confidence_threshold*. The value for *confidence_threshold* must be low enough that the Lock state will tolerate some bit errors, and high enough that the Lock state will terminate at the end of a received data packet.

To determine a value for *confidence_threshold* it is useful to calculate the values of *new_confidence* for different values of *current_confidence*. Table 4.2 tabulates these calculations. If we subjectively rate the *current_confidence* values as poor, marginal, good, and excellent as shown in Table 4.2, a selection for the value of *confidence_threshold* can be made. The value chosen for *confidence_threshold* is 192 which falls under the “good” category.

<table>
<thead>
<tr>
<th>category</th>
<th>current confidence</th>
<th>new_confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>poor</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>poor</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>poor</td>
<td>2</td>
<td>34</td>
</tr>
<tr>
<td>poor</td>
<td>3</td>
<td>56</td>
</tr>
<tr>
<td>marginal</td>
<td>4</td>
<td>78</td>
</tr>
<tr>
<td>marginal</td>
<td>5</td>
<td>100</td>
</tr>
<tr>
<td>marginal</td>
<td>6</td>
<td>122</td>
</tr>
<tr>
<td>marginal</td>
<td>7</td>
<td>144</td>
</tr>
<tr>
<td>good</td>
<td>8</td>
<td>166</td>
</tr>
<tr>
<td>good</td>
<td>9</td>
<td>188</td>
</tr>
<tr>
<td>good</td>
<td>10</td>
<td>210</td>
</tr>
<tr>
<td>excellent</td>
<td>11</td>
<td>232</td>
</tr>
<tr>
<td>excellent</td>
<td>12</td>
<td>254</td>
</tr>
</tbody>
</table>

Table 4.2 *new_confidence* values for different *current_confidence* values

As mentioned above, the Lock state is terminated whenever the *new_confidence* value falls below *confidence_threshold*. The time between the end of a packet and the instant that the Lock state is terminated is the *transition time*. Bits that are sent up to the host during the *transition time* are *transition bits*, illustrated in Figure 4.16.
The number of transition bits depends on the value of new_confidence at the end of the packet and on the values of current_confidence during the transition time. Figure 4.17 illustrates this dependence for current_confidence values of 0, 1, and 2. Figure 4.17 shows that the number of transition bits produced is at most eight. As mentioned in Section 4.4.1 the microcontroller sends data up to the host in bytes. Therefore the modem will send a maximum of 1 byte to the host during the transition time.
5 Test Results in Different Environments

The 19.2 kbps modem was used to perform many tests in various, actual operating environments. The purpose of the tests was to obtain various communication performance parameters, including bit error rate, block error rate, and percent of lost packets. These tests were carried out in the Electrical Engineering building at the University of British Columbia. This building is known to provide a harsh operating environment for power line communications.

Before the modem was tested on power line channels various statistics regarding its operation were collected for tests under white noise conditions. These statistics are described in Section 5.2 and verify the modem's correct operation.

5.1 DISCUSSION OF TEST PARAMETERS

To test modem performance, packets (blocks) are sent from the transmitting host to the receiving host. The receiving host analyzes received packets in real-time and generates results discussed later in this section. The software used to analyze the received packets is called A Bit Error Rate Tester for Power Line Modems [20]. The packets have a format shown in Figure 5.1.

![Packet format for Bit Error Rate Tester](image)

The packet in Figure 5.1 indicates that a packet has three parts, a header, a data field, and a trailer. The header and trailer identify the beginning and end of a packet, respectively. The header is terminated by a cyclic redundancy check (CRC) number. The receiving host calculates its own CRC number for the header. If the transmitted CRC and the locally calculated CRC are identical then the header contains no detected errors and the packet is forwarded for analysis. The analysis is performed on a test pattern within the packet's data field.
The test pattern in the data field consists of 125 characters: "AAAAAAAAAA BBCCDDEEFFGGGGHJJJJ-MM.MMM. A bit error occurs whenever a bit within the data field is in error. A block error occurs when one or more bit errors occur within the data field of the packet. Throughput is determined by the following formula.

\[
\text{throughput} = \text{bit rate} \cdot \left( \frac{\text{number of packets received}}{\text{number of packets transmitted}} \right) \cdot \left( \frac{\text{number of data bytes/packet}}{\text{total number of bytes/packet}} \right)
\]

The \textit{number of data bytes/packet} is the number of bytes within the data field which is 125. The \textit{total number of bytes/packet} includes all the bytes in Figure 5.1 as well as the preamble. This total comes to 145 bytes. The above formula for throughput is based on continuous single-direction transmission of packets and does not include idle time between packets. Idle time was placed between transmitted packets in order to allow time for the receiver's synchronization circuits to fall into a random state. This is done to simulate actual channel traffic. Also, the calculation for throughput does not take into consideration packets with errors. This can be done by multiplying throughput by \((1 - \text{BLKER})\).

Because of the functionality of the modem's data link layer described in Chapter 4, and the method used to obtain "operating" data, packets can be \textit{lost}. These lost packets are the reason \(\text{number of received packets} \leq \text{number of transmitted packets}\). Packets are lost in two ways. If there are many errors within a transmitted preamble (Figure 4.7), the modem's data link layer will not recognize the start of a packet. Although the rest of the packet may not contain any errors, the modem will not send any part of the packet up to the host. Thus, the modem itself loses a packet when it cannot recognize a packet's preamble. A second way in which a packet may be lost is when a packet is discarded by the receiving host. If the packet header contains an error (which is caught by the CRC) the packet is not analyzed. The Bit Error Rate Tester discards the packet and looks for the next packet. Packet discards occur because the Tester does not have any method for determining whether any part of the header is correct. Therefore in both cases, a "lost packet" is a packet whose beginning can not be identified. The parameter \textit{percent lost packets} is calculated as

\[
\text{percent lost packets} = \left(1 - \frac{\text{number of packets received}}{\text{number of packets transmitted}}\right) \cdot 100
\]

Normally, "raw" statistical value is obtained by analyzing \textit{all} data received from a modem. This can be achieved by using a device called a Data Error Analyzer. An
"operating" value is calculated by analyzing data after minimal processing of the raw data. This processing is necessary for generating statistics when using packetized data.

Errors tend to occur in bursts on power lines [19]. Figure 5.2 shows an illustration of noisy and quiet periods on a power line. The probability of a packet being lost is high if the preamble or header is transmitted during a noisy period. The nature of bursty channels is that a noisy period is followed by a fairly low noise or quiet period. Within a quiet period random errors occur. Depending on the relationship between the lengths of noisy and quiet periods, the data field of a packet which is not lost may fall more often in quiet periods or in noisy periods. If a quiet period is much larger than a noisy period, the data field will fall more often within a quiet period. Therefore, "operating" data is data which is obtained from packets whose preamble and header have a low probability of being in a noisy period. Statistics from tests that produce a small percentage of lost packets can be assumed to not suffer from this noisy-quiet period phenomena.

Figure 5.2 Noisy and quiet periods on a channel

BER is calculated for a single packet using the following formula.

\[
BER = \frac{\text{number of bits in error in data field}}{\text{total number of bits in data field}}
\]

The total number of bits in data field is equal to 1000. Because BER is a statistical number, its value is most accurate when it is based on a very large number of samples. This can either be achieved by transmitting one extremely long packet or many shorter packets. The solution is to send packets of moderate length (125 data bytes) and calculate the BER based on the total number of data bits for all the received packets.

The block error rate (BLKER) is calculated as shown in the following equation.

\[
BLKER = \frac{\text{number of received packets with error in the data field}}{\text{total number of packets received}}
\]

A block is considered to be in error if any of the bits in its data field is in error.
5.2 TEST RESULTS UNDER WHITE NOISE CONDITIONS

Figures 5.3 and 5.4 contain performance curves for the modem for additive white gaussian noise. Figure 5.3 (a) contains BER results for the modem under white noise conditions. In such a case, bit errors are independent under optimum reception. The method for determining $Eb/No$ is described in [1]. This figure shows that the modem is operating properly and the deviation from theory is approximately 1-2 dB. This deviation can be attributed to carrier and symbol recovery error [25]. The graph in Figure 5.4 (a) shows the relationship between BLKER and BER. The theoretical curve is calculated using the following equation [24].

$$BLKER = 1 - (1-BER)^l$$

where $l$ is the number of bits in a packet (here it is the number of bits in the data field) which is 1000 bits. The graph in Figure 5.4 (b) shows percent lost packets vs BER. The theoretical curve is calculated by determining the probability that the synchronization sequence will contain too many errors and thus will not be recognized by the finite state automata in the data link layer (Chapter 4). The probability of an errored preamble is found by the relationship $P(incorrect\ preamble) = 1 - P(correct\ preamble)$. $P(correct\ preamble)$ can be determined from the following relationship,

$$P(correct\ preamble) = P(correct\ SYNC\ 1) P(correct\ SYNC\ 2) P(correct\ STRT)$$

It was stated in Chapter 4 that SYNC 1 can be inverted and have at most 3 errors, SYNC 2 can be inverted and have at most 1 error, and STRT cannot be inverted and can have at most one error. Using these conditions the three probabilities can be calculated. As an example $P(correct\ SYNC\ 2)$ is given next where $p$ is the probability of a bit error.

$$P(correct\ SYNC\ 2) = \left[ \binom{16}{0}(1-p)^{16} + \binom{16}{1}(1-p)^{15}p + \binom{16}{15}(1-p)p^{15} + \binom{16}{16}p^{16} \right]$$

Figure 5.4 (b) also contains a “not synced” and a “not analyzed” curve. The not analyzed curve represents the packets not available for analysis by the software (ie. host level). The not synced curve represents the packets lost at the modem level. The difference between the not analyzed and not synced curve represent packets which were passed up to the host by the modem but were discarded by the host because of errors in the packet header detected by the CRC.
5 Test Results...

Figure 5.3 Modem performance in white noise: BER and BLKER

Figure 5.4 BLKER and percent lost packets vs BER for white noise

5.3 TEST ENVIRONMENT AND PROCEDURE

Within the Electrical Engineering building, electrical power is delivered through two systems. One system provides three phase power to all laboratories and machine shops. The other system provides three phase power to wall outlets and lighting fixtures
in hallways, classrooms, and offices. The first system is called a Lab System and the second system is called a General System (my terminology). Each of these systems has a 1600 amp bus in the Switch Room in the basement (first floor) from which power is distributed. The flow of electrical power consists of a path from the three phase 4160V/120V transformer to the General System bus, across a copper bus bar or “link”, and onto the Lab System bus [22]. Sometime in the future, this link may be replaced by a reactor. The reactor would limit the current flowing from the General System bus to the Lab System bus. This current limiting feature would protect the three phase transformer from high current flow whenever a short circuit fault appeared on the Lab System. The three phases of a power line are denoted as $\phi_x$, $\phi_y$, and $\phi_z$. The relationship between the three phases are shown in Figure 5.5.

![Figure 5.5 Three phases of a power line](image)

Three types of tests were performed on these systems. The first test is composed of same and cross phase tests within the Lab System. A second test is composed of same and cross phase tests within the General System. A third test is same and cross phase tests performed across the link between the Lab System and the General System. The transmitter RMS output voltage $V_T$ is used as a parameter in the tests. This parameter is measured at the output of the transmitter’s power amplifier and is tabulated in terms of dBmV. This decibel measure is defined as follows [10] where $V_T$ is in volts:

$$\text{dBmV} = 20 \log \left( \frac{V_T}{10^{-3}} \right)$$

The data signal that appears on the power line channel at the output of the modem is approximately 2 dBmV lower than the signal at the output of the power amplifier. This attenuation is caused by the transformer in the line-coupling network.

The length of each test in terms of bits and blocks was selected such that the number of bits in error exceeded 100 and the number of received packets was at least
1000. For tests that produced BER values better than $1 \times 10^{-5}$, the test was terminated after 10000 packets ($1 \times 10^7$ bits) were received.

5.4 SAME PHASE AND CROSS PHASE TESTS ON LAB SYSTEM

The location of the tests is a communications laboratory which contains approximately 20 computers (or workstations) and other test equipment. The transmitter and receiver are separated by approximately 100 feet of power line. Tests are performed on all three phases of the power line producing nine different channels. Three of the channels are same phase and six of the channels are cross phase. Figure 5.6 shows the BER results for the nine channels.

Figure 5.6 (a) shows that channel x to z has an approximate 22 dBmV advantage over x to x and that channel x to y has an approximate 20 dBmV advantage over x to x. Figure 5.6 (b) shows that channel y to z is 6.6 dBmV worse than y to y and channel y to x is 10 dBmV worse than y to y. Figure 5.6 (c) shows that z to x and z to y are 15 dBmV and 27 dBmV worse, respectively, than z to z.

Comparing same phase channels only, y to y performs best producing a BER of $1 \times 10^{-5}$ at 41 dBmV. For cross phase channels only, y to z performs the best with a BER of $2 \times 10^{-6}$ at 46 dBmV.

Figure 5.7 shows BLKER results for the nine channels. The BLKER results are consistent with the BER results in Figure 5.6. A more useful method of determining the characteristics of the nine channels is to look at the BER vs BLKER curves in Figure 5.8.

Figure 5.8 compares the BER vs BLKER curves of the nine channels against a curve for random (equiprobable) errors. The random curve was discussed in Section 5.2. Figure 5.8 (a) shows that the x to y channel has characteristics similar to a random noise channel. This figure also shows that channel x to z behaves like a random noise channel for BER<10^{-5}. Channel x to x does not behave like a random noise channel. Figure 5.8 (b) shows that channel y to x behaves like a random noise channel for BER<10^{-6}. Channel y to y behaves similar to a random noise channel. Channel y to z does not behave like a random noise channel but its shape suggests it will at low BER (<10^{-7}). Figure 5.8 (c) shows that channel z to y behaves like a random noise channel (with small deviation) for BER<10^{-5}. Channel z to x behaves like a random noise channel for BER<10^{-7}. Channel z to z does not behave like a random noise channel but the curve suggests it might at approximately BER<10^{-8}.
The results in Figures 5.6-5.8 clearly indicate that different communication links have large differences in BER at any given voltage level.
Figure 5.7 BLKER for lab system
Figure 5.8 BER vs BLKER for lab system

Figure 5.9 shows results for percent lost packets vs transmitter output voltage. These results are consistent with the results in Figure 5.6.
5 Test Results...

In addition to the above tests a Tektronix plotter was connected to the z to z channel, and the BER was measured and is shown in Figure 5.10. The plotter was connected at the receiver, midway between transmitter and receiver, and at the transmitter. It was stated previously that the distance between transmitter and receiver is approximately 100 feet (30 meters). This figure shows a degradation of approximately 20 dBmV for BER $< 10^{-3}$ in the normal channel (plotter not attached). Figure 5.11 (a) and 5.11 (b) show the frequency spectrum at the output of the receiver bandpass filter before
the plotter was connected and after it was connected, respectively. Figure 5.11 (b) shows the distortion of the received signal spectrum caused by the connection of the plotter. Figure 5.12 (a) and 5.12 (b) show two receiver time domain waveforms. The bottom trace is the output of the bandpass filter and the upper trace is the output of the demodulator. Figure 5.12 (b) shows severe fading in the bottom trace which results in a smaller (in amplitude) signal from the demodulator (upper trace). The plotter was chosen for this test because of its clear and drastic effect on the transmitted signal. This effect should not be construed as being indicative of the effects of all electrical equipment. The amount of noise, attenuation, and fading produced by electrical equipment varies over a wide range. The characteristics of the power line and the effects of electrical equipment on the power line are described in Chapter 2 and the references referred to therein.

![Figure 5.10 Variation in BER as plotter location is varied](image)

Figure 5.10 Variation in BER as plotter location is varied
Figure 5.11 Frequency spectrum of output of bandpass filter: (a) without, and (b) with plotter connected.
Figure 5.12 Waveforms of output of demodulator (upper trace) and bandpass filter (lower trace): (a) without, and (b) with plotter connected
Figure 5.13 shows the variation of BER and BLKER during a work day. Figure 5.13 (a) shows increases in bit errors at 12:00 pm and at 6:00 pm. Figure 5.13 (b) shows block errors approximately double at 6:00 pm. The results in Figure 5.13 are consistent with the trends suggested by Figure 2.6 in Chapter 2.

![Figure 5.13 BER and BLKER vs time for x to y channel at 63 dBmV](image)

5.5 TESTS ON GENERAL IN-BUILDING ELECTRICAL DISTRIBUTION SYSTEM

This test determines the modem's performance parameters across a building. A transmitting modem is connected to a wall outlet in Room 458, the communications laboratory. The receiving modem is moved to different locations throughout the building. The dimensions of the building are shown in Figure 5.14 [22]. The physical distance between two modems is not directly proportional to the length of the connecting power line wire. Because wiring is distributed from a circuit breaker panel on each floor, it is possible for wall outlets that are within a few feet of each other to be separated by 200 feet (60 meters) or more of wiring. Also, because the circuit breaker panel distributes single phase power from a three phase power source, wall outlets may be supplied from any one of three phases. The circuit breaker panel on each floor is connected directly to the General System bus (Section 5.3) via a sub-bus. Table 5.1 contains the names of all the General and Lab System sub-buses. This nomenclature follows from [22]. Because physically adjacent floors do not have the shortest powerline communication path,
transmission among adjacent floors will not necessarily be advantageous over non-adjacent floor transmission. For example, a signal transmitted from the fourth floor to the third floor would have the following circuitous route: the signal would travel to the fourth floor circuit panel (panel D); propagate down sub-bus D to the basement; travel across the General System bus to sub-bus C; propagate up sub-bus C to the third floor circuit panel (panel C); and finally, travel on the third floor to reach the receiver.

![Diagram of electrical engineering building]

Table 5.1 Sub-buses in General and Lab Systems

<table>
<thead>
<tr>
<th>Floor</th>
<th>General System</th>
<th>Lab System</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>2,3</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>4</td>
</tr>
</tbody>
</table>

The phases for the locations indicated in Figure 5.14 are listed in Table 5.2. It is seen in Table 5.2 that different locations are served by different power phases. Thus, tests encompass both same phase and cross phase conditions.

Table 5.2 Phases of power lines at test locations

<table>
<thead>
<tr>
<th>Location</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Room 113</td>
<td>Z</td>
</tr>
<tr>
<td>Room 214</td>
<td>Y</td>
</tr>
<tr>
<td>3rd Floor E. Wing Stairwell</td>
<td>X</td>
</tr>
<tr>
<td>4th Floor N. Wing Stairwell</td>
<td>X</td>
</tr>
<tr>
<td>Room 402</td>
<td>Y</td>
</tr>
<tr>
<td>Room 458</td>
<td>X</td>
</tr>
</tbody>
</table>
5.5.1 Bit Error Rate

Figure 5.15 shows BER results for a transmitted signal of 10.5 dBmV. Reception in rooms 113 and 214 produce very similar BER values. BER for 19.2 kbps is $3 \times 10^{-3}$ and BER for 9.6 kbps is $9 \times 10^{-4}$. Reception on the third floor test produces BER values slightly better than the lower two floors. The BERs for the third floor stairwell are $1 \times 10^{-3}$ for 19.2 kbps and $5 \times 10^{-5}$ for 9.6 kbps. BER for Room 458 which is the same room as the transmitter and on the same circuit breaker (from panel D) produces BERs for all baud rates less than $10^{-6}$. BERs for baud rates of 4.8 kbps and 2.4 kbps are less than $10^{-6}$ at all locations.

![Figure 5.15 BER for 70 dBmV data signal on General System](image)

5.5.2 Block Error Rate

Figure 5.16 shows BLKER results for a 70 dBmV transmitted signal. Almost 100% of the blocks for 19.2 kbps received in Rooms 113 and 214 contained errors. BLKER for 9.6 kbps is 32% for Room 113, 53% for Room 214, and below 5% for the third floor stairwell. BLKER is 0 for baud rates of 2.4 and 4.8 kbps at all locations tested. Figures 5.15 and 5.16 show that the third floor test location has lower error rates than both the first and second floor locations. An explanation for this observation is that the Third Floor Stairwell is connected to the same phase (phase X) as the transmitter. Rooms 113 and 214 are connected to phases Z and Y, respectively.
5.5.3 Percent of Lost Packets

Figure 5.17 shows the percent of lost packets for a 70 dBmV transmitted signal. The percent of lost packets for 2.4 kbps varied from 0 to 5%, and the percent of lost packets for 19.2 kbps varied from 0 to 42%. Figure 5.17 suggests that same phase and cross phase considerations are important.

5.5.4 Summary

The results show that baud rates of 2.4 and 4.8 kbps should be used for long distance communication within a building in the absence of forward error correction coding. The baud rate of 19.2 kbps is appropriate for short distance communication, and 9.6 kbps should be used for medium distance communication. The results also show that the baud rate of 2.4 kbps is the least affected of the four baud rates by cross phase
impairments. The results indicate the need for a modem with some form of baud rate flexibility.

5.6 TESTS FOR TRANSMISSIONS FROM LAB TO GENERAL SYSTEM

The transmitter is connected to Øy power line of the Lab System as described in Section 5.4. The receiver is connected to various locations in the General System as described in Section 5.5.

5.6.1 Bit Error Rate

Figure 5.18 shows BER results for a transmitted signal of 70 dBmV. Rooms 113 and 402 produced the best and worst BER values, respectively. The BER value for 2.4 kbps in Room 402 is significantly higher than values produced from other locations. It can be implied from this high BER at low data rates that the power line channel between Room 402 and Room 458 is extremely poor.

Figure 5.18 BER for 70 dBmV data signal on Lab to General System

5.6.2 Block Error Rate

Figure 5.19 shows BLKER results for a transmitted signal of 70 dBmV. The BLKER values for 9.6 and 19.2 kbps are very high, and exceed 80%. The lowest BLKER for 4.8 kbps is 20% in Room 113. All BLKER values for 2.4 kbps are below 1% except for Room 402 which produced a BLKER of 35%.
5 Test Results...

5.6.3 Percent of Lost Packets

Figure 5.20 shows the percent of lost packets for a 70 dBmV transmitted signal. At 19.2 kbps, packet loss ranges from 45% to 95%. At 2.4 kbps packet loss is below 20%. Figure 5.20 does not show any advantage in same phase transmission relative to cross phase transmission.

5.6.4 Error-free Throughput

Figure 5.21 shows the error-free throughput (as defined in Section 5.1) for a 70 dBmV transmitted signal. This figure shows that low baud rates produce higher throughput values than higher baud rates. Transmitting at 2.4 kbps produces the highest throughput at receiving locations in Room 113, Room 402, and the fourth floor stairwell. The baud rate of 4.8 kbps produces the highest throughput at receiving sites in Room 214 and the third floor stairwell.
5 Test Results...

Figure 5.21  Error-free throughput for 70 dBmV data signal on lab to general system

5.6.5 Summary

The results in this section clearly reflect the observations stated in Section 5.5 concerning the architecture of the test site's power distribution system. Figure 5.18 showed that the BER increased as the physical location of the receiver neared that of the transmitter. This trend in Figure 5.18 is explained by the fact that the "link" between the General and Lab Systems is located on the first floor. The transmitted signal originates on the fourth floor in Room 458, and propagates over the Lab System to the first floor where it crosses the link. The signal continues to propagate from the link throughout the General System towards the fourth floor. Therefore Room 113 is closest to the transmitter, and the Fourth Floor Stairwell is farthest. The Fourth Floor Stairwell although being the closest physically, is in fact the farthest through the power line channel, and its BER, BLKER, and percent of lost packets results are among the highest (and worst) in the building.
6 Conclusions

6.1 CONCLUDING REMARKS

This thesis has pursued the development of an intelligent modem architecture for use on interbuilding power lines. Modem functionality and cost have been primary design and evaluation considerations.

The basic task of any modem is to provide a means of communicating over actual, non-ideal channels. The transmitting part of a modem accepts information from its host, and modulates this information for transmission over a channel. The receiving part of a modem recovers transmitted information from a channel, and sends this information up to its host. This basic limited functionality is insufficient when the modem is used as a "transceiver" under actual network conditions. In an actual network a host is constantly utilizing the modem's abilities to send and receive data among a number of other host computers. All of the hosts are potentially busy with other tasks, and should spend minimal time in communicating with their modems. An intelligent modem which does much of the processing otherwise done by a host is advantageous over a non-intelligent modem.

The power line modem developed in this thesis is microcontroller-based. Several important revisions and innovations to the earlier preliminary design enabled operation at transmission rates up to 19.2 kbps. A data link layer is provided, to enable extensive processing by the modem itself. This processing includes byte and packet synchronization and recognition. Judicious division of data link layer functions between modem and host facilitate choice of packet length and format, and enable accommodation to a variety of applications.

Many test results obtained using an industrial building as a test site are documented in this thesis. The building contains two different systems of three phase power distribution. One system distributes electrical power to many laboratory and machine shop rooms. The other system distributes electrical power to lighting fixtures and wall outlets. The purpose of the tests was to determine the modem's performance under actual
6 Conclusions

The tests were used to obtain extensive data to estimate bit error rate, block error rate, percentage of lost packets, and throughput.

To facilitate comparisons an initial test was performed in an isolated location in the Lab System. The results indicated that performance on some channels differed from performance using the best channel by as much as 30 dBmV.

Another test was performed using different locations within the industrial building. The results show that nearness of the receiver to transmitter can be a dominating factor in the number of transmission errors. Baud rates of 9.6 kbps and 19.2 kbps resulted in relatively high BER performance for long transmission distances. These two baud rates produced bit error rates in the range of $10^{-3}$ and block error rates above 30%. On the other hand, baud rates of 2.4 kbps and 4.8 kbps produced bit error rates below $10^{-6}$ and block error rates below 1%.

The test results give an indication of the conditions under which each baud rate would be appropriate. Rates of 2.4 kbps and 4.8 kbps are relatively slow but provide low BER and BLKER values even under long transmission paths. Rates of 19.2 kbps and 9.6 kbps produce relatively high through rates and acceptable BER and BLKER on short transmission paths.

The test results also indicate that performance at 19.2 kbps is very sensitive to power line noise. The BLKER may change by as much as 30% over a two hour period. The lower baud rates, especially 2.4 kbps, are not as sensitive as 19.2 kbps to channel impairments, in our test environment.

6.2 SUGGESTIONS FOR FURTHER WORK

There are several potential enhancements to the 19.2 kbps modem. The first enhancement would double the bit rate of the modem to 38.4 kbps. This rate increase could be accomplished in one of two ways. The first approach is to replace the current microcontroller with one which operates at twice the speed. In order to operate at 38.4 kbps, the modem's clock circuitry would need adjustment. Using the current modulating carrier, such adjustment would result in three carrier periods per data bit. The second approach is to change the data carrier modulation from BPSK to QPSK. This alternative would maintain the symbol rate at 19200 symbols per second while doubling the bit rate to 38.4 kbps. In order to change the modulation scheme to QPSK major changes would be required to the modulator and demodulator circuitry. Also, the modem's firmware would
need slight alteration. Such enhancements would be appropriate in relatively noise-free environments, for those applications which require high bit rates.

Another enhancement which could be made to the modem is to replace the microcontroller with a digital signal processing (DSP) chip. With this DSP chip the received signal could be demodulated using DSP techniques, and the Costas loop would be eliminated. Familiarity with the required DSP chip would be needed in order to determine whether bit rates above 9.6 kbps can be obtained.

A third possible modem enhancement is to implement a forward error correction (FEC) algorithm. The addition of this function could greatly improve the modem's BER and BLKER parameters on noisy links [48].

A further enhancement would involve the incorporation of medium access control as a part of the modem circuitry. Such control is needed, to enable orderly access to a network, by otherwise unsynchronized access arrangements. A modified version of CSMA is an appropriate access control scheme [44]. Polling may be useful for some applications. An implementation which allows either CSMA or polling could be useful. Preliminary work indicates that the modem described in this thesis can be readily modified to accept medium access control circuitry on the modem circuit board.

A very large scale integration (VLSI) implementation of the modem would provide several advantages. The power consumption would be reduced substantially. The modem would be smaller and cheaper to produce. Digital components are more reliable than analog components and hence an all-digital BPSK modem would have this added reliability, relatively, over a digital-analog modem.

The power line modem could be implemented as an internal personal computer (PC) card. The modem would take its power from the PC backplane and packets could be sent to and from the modem by the host PC using direct memory access (DMA). This approach is very reliable but is somewhat complicated as it requires a thorough understanding of the techniques needed to design an integrated PC card. Connection to the power line channel would be accomplished via an external connector.
References


References


References


Appendix A

Hardware Schematics

The following schematics detail PSK modem hardware.

A.1 MODEM OVERVIEW
A.2 CONTROL UNIT
A.3 BIT SYNCHRONIZATION
A.4 MODULATOR
A.5 COSTAS LOOP DEMODULATOR
A.6 RECEIVE BAND PASS FILTER
A.7 TRANSMIT LOW PASS FILTER
A.8 LINE COUPLING, RELAY, AND POWER AMP
A.9 POWER SUPPLY
A.10 TERMINAL INTERFACE
2ND ORDER BUTTERWORTH LOWPASS FILTER

\[ f_0 = 132.6 \text{ KHz} \]

- \( R1 \): 12K
- \( R2 \): 12K
- \( C1 \): 100PF
- \( C2 \): 100PF
- \( U18 \)
- \( R3 \): 4.87K 1%
- \( R4 \): 8.25K 1%

SUPPLY DECOUPLING
NEAR OP AMP
Appendix B

Firmware Listing

Following is a listing of firmware source code in the modem microcontroller.
offs20.a51

This is the microcontroller program that controls the 9600 bps
FSK power line modem.

- August 16/89: added synchronous communication capability
- If P1.7 is unconnected (or high) the software uses an asynchronous
  communication protocol for interacting with the host. If it is low
  the software uses a synchronous communication protocol for
  interacting with the host.
- Oct 30/89: added new mechanism to initiate transmit mode. Now, receiving
  a character from the DTE automatically puts the modem into transmit
  mode. The modem also understands the following escape sequences:
    (esc, esc) — send the esc character to the modulator
    (esc, R) — terminate transmit mode, go to receive mode
    (esc, '\') — reset the modem
    (esc, "any") — send "any"
- Also, the action that is now taken on transmit buffer underrun is
  to go back to receive mode.
- September 21, 1990: 16 bit syncs and start of packet indicator (BDS)
- January 30, 1991: Removed the code which did the bit synchronisation.
- May 1, 1991: Timer 0 is used instead of the external counter.
- Firmware able to be used for rates up to 19.2 kbps
- May 15, 1991: Software selectable baud rates (15.2/9.6/4.8/2.4)

hardware addresses
    .equ PORT, 0

ascii characters
    .equ CR, h'0D'
    .equ LF, h'0A'
    .equ ESC, h'1B'
    .equ XON, h'11'
    .equ XOFF, h'13'
    .equ STX, h'40'
    .equ ETX, h'41'
    .equ STXHE, h'5D'
    .equ ETXHE, h'61'
    .equ CONF_MAX, h'CO'
    .equ H_EOF1, h'37'
    .equ H_EOF2, h'38'
    .equ H_EOF3, h'a3'
    .equ H_EOF4, h'a4'

8051 register assignments
    .opf Z_TIME, R1 ; next sampling time
    .opf SAMPLE, R2 ; samples
    .opf STATE, R4 ; state of receiver
    .opf COUNT, R5 ; bit counter
    .opf conf, R6 ; confidence in received bit
    .opf avg_conf, R7 ; running average of confidence
    .opf TX_BITS, P1.0 ; to modulator

DATA_MEMORY assignments
    .segment .data
    .org 0x20

FLAG:
    .rs 1 ; storage for flags
    .equ ESCAPE, (FLAG - 0x20) + 0
    .equ XOFF, (FLAG - 0x20) + 1

SR:
    .rs 2 ; input shift register (16 bits)
    .equ SRH, SR
    .equ SRL, SR + 1

CONF_THRESH:
    .rs 1 ; dcd threshold (byte)

INVERT:
    .rs 1 ; inversion byte

SYNC:
    .rs 2 ; sync word (16 bits)
    .equ SYNCH, SYNC
    .equ SYNF, SYNC + 1

ST_O_P:
    .rs 2 ; 16-bit start of packet indicator
    .equ STOP, ST_O_P
    .equ STOP_L, ST_O_P + 1

LINK_RSC:
    .rs 1 ; data link escape character

transmit related constants and data
    .equ BUF_SIZE, 40
    .equ XON_LMT, (BUF_SIZE * 1 / 4)
    .equ XOFF_LMT, (BUF_SIZE * 3 / 4)

band_rate:
    .rs 1
n_stop:
    .rs 1
n_sync:
    .rs 1
buffer:
    .rs BUF_SIZE

START:
    jmp INIT
    .org 0x40
INIT:
: set up stack pointer
: note: stack is only 32 bytes deep (0x60 to 0x7f) !

mov SP, 0x60

: reset the bits that control the modem hardware
mov A, 0x7f
mov P0, A
mov P1, A
mov P2, A
mov P3, A

: set up the 8051's internal counters
: T0: integrate and dump counter for rx bits
: T1: used as baud rate generator for serial port in async operation

mov T0L, $0
mov T0H, $0
setb TR0 ; turn on timer/counter 0
clr RI
setb IT1 ; bit sync in INT1
clr EA ; disable all interrupts
clr TI
mov TM1, #255 ; baud rate = 38.4 to host
setb TR1 ; turn on timer 1
setb CTS
clr CTS

: set up 8051's serial port

mov SCON, #b'00101000' ; mode 1, rx enabled
mov SM0, $00 ; set as rts
mov SM1, $00

: say hi to the dce

mov DPTR, @halt
acall puts
clr A
acall putch ; send a zero to terminate the hello
: string

mov LINK, ESC, $2C
mov COMP_THRES, $COMP_MAX ; 192
mov STWCLOW, $80
mov STWCHIGH, $80
mov STWBCON, $80
mov STWBCEN, $80
mov baud_rate, #192
acall ch_baud

: initialise by generating a sampling time 1 clock cycle
: beyond the present

init_rx:
setb IT1 ; bit sync in INT1
clr IE1 ; clear any previous edges
setb RELAY_CONTROL
setb DCD
setb CTS
mov INVERT, #0 ; set to no invert

acall delay ; wait for relay to settle
jmp ss ; ----- sync search -----
sync_search:
jnb RI, cont1
jmp transmit
cont1:
mov A, avg_conf ; get average confidence
mov DPTR, conf_mul_tab
movc A, @DPTR
add A, conf
mov avg_conf, A

clms A, COMP_THRES, nest1

nest1:
mov DCD, C

ss:

jnb IE1, ss ; wait for falling edge
mov A, TLO
mov TLO, $0 ; clear counter
clr IE1 ; clear edge indicator

mov DPTR, $bit_tab ; map sum into proper output
movc A, @DPTR
add A, conf ; get received bit in C, conf in A
mov conf, A ; store
rnc A

mov byte into shift register and count number of ones in sr XOR sync

clr SYNST
setb sync.detect line
rnc A
mov A, SRHIGH
rnc A
mov SRHIGH, A
xrl A, SRHIGH ; XOR with sync
mov byte into byte into shift register
rnc A
mov A, @DPTR
movc A, @DPTR ; count ones
rnc A
mov R0, A ; save in R0
mov A, SRLOW
rnc A
mov SRLOW, A
xrl A, SRLOW ; XOR with sync
mov A, @DPTR
movc A, @DPTR ; count ones
add A, R0 ; add ones from other half

: determine how close sr is toSYNC

mov DPTR, $sync ok
movc A, @DPTR
js sync_search ; to sync search if errors
jmp transmit ; ----- SYNC_VRFY -----

sync_vrfy:

jnb RI, cont2
jmp transmit
cont2:
mov A, avg_conf ; get average confidence
mov DPTR, conf_mul_tab
movc A, @DPTR
add A, conf
mov avg_conf, A

clms A, COMP_THRES, nest12

nest12:
mov DCD, C

ss:

jnb IE1, ss ; wait for falling edge
mov A, TLO
mov TLO, $0 ; clear counter
clr IE1 ; clear edge indicator
mov     SHUF, #R_ZDP2  ; send XOM to get host started
mov     avg_conf, #0xa0
1jmp    sync_arach    ; return to sync state

sl:     jnb    IEL, sl     ; wait for falling edge
mov     A, TL0        ; read length of pulse
mov     TL0, #0       ; clear counter
cir     IEL           ; clear edge indicator
mov     DPTR, #bit_tab ; map sum into proper output
mov     A, 8A + DPTA   ; get received bit in C, conf in A
mov     conf, A       ; store confidence
mov     A, SHIGH      ; move bit into shift register
rcc     A
mov     SHIGH, A     ; save the bytes into the shift reg
djns    COUNT, sync_lock ; do we have all eight bits yet?
mov     SLOW, A      ; yes, save in case next st is search
xch     A, R3        ; save byte; send prev. to host
mov     SHUF, A      ; another byte

:  ; This is the transmit portion of the modem control firmware.
:  ; Characters are received from the host over a 38.4K async line.
:  ; These characters are transmitted to the other modem(s) over the
:  ; power line at various rates (2.4K - 19.2K bps). The microcontroller
:  ; can buffer up to 9 characters. It uses the XOM/KOFF protocol to
:  ; communicate to the host when to send it characters. (An XOM is sent
:  ; to signify that its buffer is getting empty. An XOM is sent to
:  ; signify that its buffer is getting full. When the host stops
:  ; sending data the buffer empties and the controller switches to
:  ; receive mode.
:  ; using registers:  R0 - input pointer
:  ; R1 - output pointer
:  ; R2 - buffer counter
:  ; R3 - sync counter
:  ; R7 - bit counter

:  ; transmit:
:  ; mov     A, SHUF        ; get character
:  ; cir     R1             ; clear indicator
:  ; cpl     CTS
:  ; cjns    A, LINKEC, tx2  ; is it the escape character?
:  ; acall   getch          ; yes, get another character
:  ; cjns    A, $'N', tx2    ; is this one 'N'?
:  ; acall   cmdwait
:  ; ajmp    init_rx
:  ; cjns    A, $#N', tx1    ; is this one 'N'?
:  ; ajmp    init_rx
:  ; txl:    cjns    A, #0, tx2    ; yes, go to receive mode
:  ; ajmp    init
:  ; tx2:    mov     buffer, A  ; save first char in buffer
:  ; setb    DCD          ; turn off dcd
:  ; cir     RELAY_CONTROL ; set relay to tx position
:  ; mov     R0, #bufsize1 ; in_ptr
:  ; mov     R1, #bufsize2
:  ; mov     R2, #2        ; buf_count contains one char
:  ; cir     KOFF

:  ; receive:
:  ; mov     A, $XOM       ; send XOM to get host started
:  ; acall   send_preamble ; send out the preamble
:  ; loop:   djns   R2, tx3  ; any bytes in buffer?
:  ; ajmp    init_rx      ; no, go to receive mode
:  ; tx3:    cjns    R2, #XOM, tx4 ; is buffer getting empty?
:  ; jnb     XOFF, tx5      ; yes, don't send XOM if not XOFF
:  ; mov     SHUF, $XOM    ; XOFF, so send XOM
:  ; cir     KOFF
:  ; tx4:    mov     A, #R1    ; get next character
:  ; cir     KOFF
:  ; xch     A, R3        ; save byte; send prev. to host
:  ; mov     SHUF, A      ; another byte

:  ; send_preamble:
:  ; send three syncs and one at of pack
:  ; using registers:  R0 - input pointer
:  ; R1 - output pointer
:  ; R2 - buffer counter
:  ; R3 - sync counter
:  ; R7 - bit counter

:  ; anotherosc:  mov     R7, #16  ; bit counter for sending words
:  ; ldsync:      mov     SHIGH, #off
:  ; mov     SLOW, #0x00
:  ; a jmp    ploop
:  ; ploop:  ; mov     R0, #bufsize2
:  ; cir     R2
:  ; inc     R0  ; advance buffer pointer
:  ; tx:    mov     R2, #bufsize2
:  ; send an XOFF if appropriate
```plaintext
<table>
<thead>
<tr>
<th>Label</th>
<th>Instruction</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>setb</td>
<td>XE0FF</td>
<td>: note that XOFF was sent</td>
</tr>
<tr>
<td>tx3p:</td>
<td>jnb</td>
<td>: wait for clk high</td>
</tr>
<tr>
<td></td>
<td>jb XE0FF, tx10p</td>
<td></td>
</tr>
<tr>
<td>tx10p:</td>
<td>mov</td>
<td>: rotate shift register</td>
</tr>
<tr>
<td></td>
<td>A, SRHIGH</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mov</td>
<td>: output bit is in carry</td>
</tr>
<tr>
<td></td>
<td>A, SRLOW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mov</td>
<td>: output bit</td>
</tr>
<tr>
<td></td>
<td>A, SRLOW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mov</td>
<td>: update word counter</td>
</tr>
<tr>
<td></td>
<td>COUNT, anothorone</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ret</td>
<td></td>
</tr>
</tbody>
</table>

: ----------------- end of transmit code -----------------

: ********** Subroutines **********

: wait for commands:
: ESC to initialise the modem,
: C to set cdc threshold,
: D to set async word,
: L to set link escape character,
: Q to print out modem registers,
: X to set start of packet
: 'x'' is ignored

cmdwait: mov A, \CR  ; send out a <CR><LF>
  acall echo
  acall getch  ; wait for a character
  acall toupper ; convert to upper case
  js cmdwait   ; ignore 'x''

chkquest: cjne A, 8'c', chkq
  acall init
  js jmp cepass

chkq: cjne A, 8'c', chks
  acall puts
  acall getbyte
  mov COMF.THR, A
  jmp cmdwait

chks: cjne A, 8'c', chki
  acall getbyte
  mov DFTK, #prompt
  acall puts
  mov 80, 8SYNC
  acall getword
  jmp cmdwait

chki: cjne A, 8'c', chkk
  acall getbyte
  mov DFTK, #prompt
  acall puts
  acall getbyte
  mov LINK, ESC, A

chk: cjne A, 8'c', chkk
  acall printinfo
  jmp cmdwait

chk: cjne A, 8'c', chkk
  acall puts
  mov R0, 8SYNC_P
  acall getword
  jmp cmdwait

chk: cjne A, 8'c', chkk
  acall getch
  mov baud_rate, A
  acall ch_baud
  jmp cmdwait

chk: cjne A, 8'c', chk
  acall getch
  mov baud_rate, A
  acall puts
  mov A, baud_rate
  acall putc
  jmp cmdwait

chk: cjne A, 8'c', chks
  acall DFTK, #version
  acall puts
  jmp cmdwait

chk: cjne A, 8'c', cmd_ret
  jmp EXIT

ch_baud: mov A, baud_rate
  cjne A, 8'c', chk96
  clr BAUD0
  clr BAUD1
  jmp baud_re

ch96: cjne A, 8'c', chk96
  setb BAUD0
  clr BAUD1
  jmp baud_re

ch8: cjne A, 8'c', chk8
  clr BAUD0
  setb BAUD1
  jmp baud_re

ch24: cjne A, 8'c', chd24
  setb BAUD0
  clear BAUD1
  setb baud_re

chd24: clr BAUD2
  clr BAUD2
  baud_re: ret

: Print the zero terminated string pointed to by DFTK to the DTH.

: ----------------- puts -----------------------------

puts: clr A
  mov A, 8x + DFT
  inc DFT
  ; get next character
  inc DFT
  ; advance pointer
  js ps2
  ; jump out if end-of-string
  acall echo
  ; send out characters
```
gd1:      cjne A, $'A', +13 ; try for a letter
         jc getdigit ; ch is < 'A'
         cjne A, $'F'+1, +13
         jnc getdigit ; ch is > 'F'
         call echo ; display
         add A, $'A'+10 ; subtract offset
         cli C ; return with C clear
         ret

; --------------- getbyte -----------------
; Reads a 2 digit hex number from the serial port. Returns the value in
; A. C will be set if CR was pressed.
;                          getbyte
getbyte:     mov B, #0 ; assume 0
         call getdigit ; try to get a digit
         jc gbl ; jump out if CR pressed
         mov B, A ; store in B
         call getdigit ; try to get a digit
         jc gbl ; jump out if CR pressed
         xch A, B ; combine two bytes
         swap A
         orl A, B
         cli C ; return with C clear
         ret
         mov A, B ; retrieve partial result
         ret ; return (C set)

; --------------- getword -----------------
; Reads a 4 digit hex number from the serial port. Stores the value at
; the location pointed to by R0. Note that the MSbyte is stored at the
; low address.
;                          getword
getword:    cli A ; clear out the value
         mov @R0, A
         inc R0
         mov @R0, A
         mov B, #4 ; initialize digit counter
         call getdigit ; read a digit
         jc gwl ; get out if CR was pressed
         xch A, @R0
         swap A
         xch A, @R0
         dec R0
         xch A, @R0
         swap A
         xch A, @R0
         xchd A, @R0
         inc R0
         djns R0
         gwl ; go do the next digit
         ret

; --------------- delay ------------------
; Delays execution by ~512 clock cycles. Destroys R0.

; --------------- bitcount ----------------
; Returns the number one bits in A. Destroys DF, R0.
bitcount:   mov DPTR, #01bits ; point at table
         movc A, @A+DPTR ; get value from table
         ret ; done

; ********* CONSTANT STRINGS *********************

helpinfo:   "\r\t19200 BPS Intra-Building Power Line Modem. Command Summary"
         "\r\tC - Enter CD threshold"
         "\r\tS - Enter baud rate"
         "\r\tR - Print user settable parameters"
         "\r\tP - Print help info"
         "\r\tT - Initialize"
         "\r\tESC X - User mode"
         "\r\t"

; ********** LOOKUP TABLES *********************
oonebits:   0, 1, 1, 2 ; (0, 1, 2, 3)
         1, 2, 2, 3 ; (4, 5, 6, 7)
         1, 2, 2, 3 ; (8, 9, 10, 11)
         2, 3, 3, 4 ; (12, 13, 14, 15)
         1, 2, 2, 3 ; (16, 17, 18, 19)
         2, 3, 3, 4 ; (20, 21, 22, 23)
         1, 2, 2, 3 ; (24, 25, 26, 27)
         3, 4, 4, 5 ; (28, 29, 30, 31)
         1, 2, 2, 3 ; (32, 33, 34, 35)
         2, 3, 3, 4 ; (36, 37, 38, 39)
         2, 3, 3, 4 ; (40, 41, 42, 43)
         3, 4, 4, 5 ; (44, 45, 46, 47)
         2, 3, 3, 4 ; (48, 49, 50, 51)
         3, 4, 4, 5 ; (52, 53, 54, 55)
This is a table of the values to be output to port 1 according to the number of bits in the early/late values. Bit 4 gives the decoded bit value and bit 5-6 give the confidence value. Bit 4 is set to one so that pi.6 can be used as an input port.

Note that we go up to 26 because the bit sync can move to 25 samples.

<table>
<thead>
<tr>
<th>bit_tab:</th>
<th>12 &lt;= 1</th>
<th>0</th>
<th>number of ones = 0</th>
<th>10 &lt;= 1</th>
<th>1</th>
<th>number of ones = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 &lt;= 1</td>
<td>0</td>
<td>number of ones = 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 &lt;= 1</td>
<td>0</td>
<td>number of ones = 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 &lt;= 1</td>
<td>0</td>
<td>number of ones = 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 &lt;= 1</td>
<td>0</td>
<td>number of ones = 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 &lt;= 1</td>
<td>0</td>
<td>number of ones = 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 &lt;= 1</td>
<td>0</td>
<td>number of ones = 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 &lt;= 1</td>
<td>0</td>
<td>number of ones = 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 &lt;= 1</td>
<td>0</td>
<td>number of ones = 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 &lt;= 1</td>
<td>0</td>
<td>number of ones = 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 &lt;= 1</td>
<td>1</td>
<td>number of ones = 11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 &lt;= 1</td>
<td>2</td>
<td>number of ones = 12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 &lt;= 1</td>
<td>3</td>
<td>number of ones = 13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 &lt;= 1</td>
<td>4</td>
<td>number of ones = 14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 &lt;= 1</td>
<td>5</td>
<td>number of ones = 15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 &lt;= 1</td>
<td>6</td>
<td>number of ones = 16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 &lt;= 1</td>
<td>7</td>
<td>number of ones = 17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 &lt;= 1</td>
<td>8</td>
<td>number of ones = 18</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 &lt;= 1</td>
<td>9</td>
<td>number of ones = 19</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 &lt;= 1</td>
<td>10</td>
<td>number of ones = 20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 &lt;= 1</td>
<td>11</td>
<td>number of ones = 21</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 &lt;= 1</td>
<td>12</td>
<td>number of ones = 22</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 &lt;= 1</td>
<td>13</td>
<td>number of ones = 23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 &lt;= 1</td>
<td>14</td>
<td>number of ones = 24</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 &lt;= 1</td>
<td>15</td>
<td>number of ones = 25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 &lt;= 1</td>
<td>16</td>
<td>number of ones = 26</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This table gives tab[1] = round(21.0/22.0*1) where 1 is unsigned.