Timed Power Line Data Communication

A Thesis Submitted to the College of
Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in the Department of Electrical Engineering
University of Saskatchewan
Saskatoon, Saskatchewan, Canada

By

Kevin Wade Ackerman

PERMISSION TO USE

In presenting this thesis, in partial fulfillment of the requirements for a degree of Master of Science from the University of Saskatchewan, the researcher agrees that the Libraries of this University may make it freely available for inspection. The researcher further agrees that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor(s) who supervised this thesis work. In their absence, this permission can also be given by the Head of the Department, or the Dean of the College of Graduated Studies and Research at the University of Saskatchewan. Any copying, publication, or use of this thesis or parts thereof for financial gain without the researcher's written permission is strictly prohibited by law. Due recognition shall be given to the researcher and to the University of Saskatchewan in any scholarly use which may be made of any material in this thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or in part should be addressed to:

Head of the Department of Electrical Engineering
57 Campus Drive
University of Saskatchewan
Saskatoon, Saskatchewan, Canada
S7N 5A9

University of Saskatchewan

Timed Power Line Data Communication

Candidate: Kevin Ackerman

Supervisor: David E. Dodds

Supervisor: Carl McCrosky

M.Sc. Thesis Submitted to the

College of Graduate Studies and Research

ABSTRACT

With the ever increasing demand for data communication methods, power line communication has become an interesting alternative method for data communication. Power line communication falls into two categories: one for data transmission between sites in the power grid and the other for home or office networking. When considering home or office networking, existing methods are either too slow for tasks other than simple automation, or are very fast with a higher cost than necessary for the desired function. The objective in this work is to develop a lower cost communication system with an intermediate data transmission rate.

At first glance, power line communication looks like a good option because of the availability of power outlets in every room of a building. However, the power conductors were installed solely for the purpose of distributing 60 Hz mains power and, for data signals, they exhibit very high attenuation, variable impedance and there is radio frequency shielding. Furthermore, many of the 60 Hz loads produce radio frequency interference that impedes data communication. Previous research has shown that much of the noise is *time synchronous* with the 60 Hz mains frequency and that the majority of data errors occur during these periods of high noise.

This work develops a power line communication protocol that coordinates transmissions and uses only the *predictable* times of lower noise. Using a central control strategy, the power line 60 Hz mains signal is divided into 16 timeslots and each timeslot is monitored for errors. The central controller periodically polls all stations to learn which timeslots have low noise and it then controls all transmissions to make the best use of these good timeslots. The periodic polling allows the system to adapt to changes in electrical loading and noise. This control strategy has been achieved with modest complexity and laboratory measurements have shown throughput approaching 70% of the modem bit rate.

ACKNOWLEDGEMENTS

I wish to thank and acknowledge my two supervisors, Prof. David Dodds and Dr. Carl McCrosky. Without their expertise, guidance, and advice, the goals of this thesis would have been much more difficult to obtain. Their efforts in providing me with constant feedback and encouragement are greatly appreciated. These are truly brilliant men who I respect and admire.

I must also express my thanks to the University of Saskatchewan for the excellent faculty and staff there. The institution has provided me with outstanding education that will aid me throughout my career. I would also like to thank the management, staff and other students at TRLabs. In particular, Garth Wells who helped greatly with the design and fabrication of hardware and Jack Hanson who provided guidance for me at TRLabs and while visiting a conference. I would also like to show appreciation for the students at TRLabs that provided me with a pleasant working environment and often provided me with help, in particular Bernie Boos and Bernardo Celaya who were always great to talk to in a time when I needed some technical help or wanted to talk about things other than work.

I have been very fortunate to receive financial assistance from the University of Saskatchewan, TRLabs, and the National Science and Engineering Research Council of Canada (NSERC). This financial assistance has made life a lot easier through my years of schooling.

Last and certainly not least, I thank my family and friends. My mom Elaine, brother Brent and his fiancé Glenna, sister Shelly and her husband Larry and their three kids Jayna, Dylan and Kaylin. All of you have always expressed great interest in my work even though I know sometimes it is quite hard to understand! However, my greatest supporter has been my amazing fiancé Kelly. Thank you for taking care of me when I'm sometimes too busy to! Without your love and support this task would have been much more difficult. Everything in life makes a lot more sense since we've been together. I love you.

DEDICATION

This thesis is dedicated to my late father Ernie Ackerman. Since dad's passing in early 2001, life hasn't been and never will be the same. Dad, I know that somewhere, somehow you are looking down on me saying… "That's my boy".

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ac	Alternating Current
ARQ	Automatic Repeat Request
ACK	Acknowledgement
BER	Bit Error Rate
BPF	Bandpass Filter
BW	Bandwidth
CDMA	Code Division Multiple Access
CRC	Cyclic Redundancy Check
dB	Decibels
FDMA	Frequency Division Multiple Access
FSK	Frequency Shift Keying
Hz	Hertz
kb/s	Kilo-bits per second
kHz	Kilo-hertz (kilo = 10^3)
LSB	Least significant bit
MAC	Media Access Control
Mbps	Mega-bits per second
MHz	Mega-hertz (kilo = 10^6)
ms	Millisecond (milli = 10^{-3})
MSB	Most significant bit
PLC	Power Line Communication
QAM	Quadrature amplitude modulation
RTS/CTS	Request to Send/Clear to Send
SNR	Signal to Noise Ratio
TDMA	Time Division Multiple Access
UART	Universal Asynchronous Receiver Transmitter

1. Introduction

1.1 Motivation For This Work

In the past twenty years, data networks have gone from being an experimental technology to becoming a key tool for business and entertainment used by companies and homes worldwide. Companies use networks to transfer data files and share applications between computers, as well as to share access to network devices such as printers and workstations. This demand by companies is large, but the demand for data communication in the home is also becoming significant. Home users, who often have more than one computer, are looking to data communication networks to share information between computers. They are also looking to networks for the 'automation' of their home – including applications such as security systems, network gaming, and controlling heating, air conditioning and other household appliances.

The design of a network should consider several factors, of which the two most important are predicted network traffic and installation cost. The nature of traffic generated by applications such as email, streaming audio or video, file transfer, control systems, application or resource sharing, etc. predicts the type of service needed. The various types of traffic can have different throughput, data integrity, latency, and other requirements. A simple control system network that performs functions such as turning lights on and off, opening and closing the garage door, and controlling the air conditioner does not require high speeds. A high speed network would be much better utilized by a multiple computer network where there is a large amount of file and application sharing or video.

The cost factor refers to the installation cost of a network. High speed networks often require more expensive equipment than low speed networks, so for low speed networks it

is not economically smart to install high speed equipment. Installation cost is also affected by the actual setup of the network. Wireless equipment is becoming popular because it is simple to set up and provides high speed and high mobility (computers can access the network as long as they are within a certain distance of the access point). However, the wireless equipment may be too costly for low to medium speed applications. Another solution is to use dedicated network cabling but this is also a high cost solution because retrofitting a home with the required cabling becomes a time consuming and expensive job. Also, once network cabling is installed in a home or office, it does not lend itself easily to reconfiguration – resulting in down time when location of network entities changes.

What is missing is a medium speed technology that is low cost and allows for easy and ubiquitous network access. This thesis addresses a possible solution to the problem of mobility, ease of installation, and cost of networks by using the in-building power distribution system.

1.2 System Proposal

Using the in-building power distribution system as a communications medium can be referred to as a "power line" network, and the advantage is that the wiring for the network is already in place and thus there is no requirement for dedicated network cable. Power conductors have excellent coverage because every room in a building has power and thus the network is easy to access. If equipment has to be re-configured or moved, it is as simple as moving the equipment and plugging it in to the power line.

Another advantage is that any device that requires connection to the power line also has access to a network. The heating ventilation and air conditioning (HVAC), refrigerator, microwave oven, and various other appliances are devices which, with installed logic, could be controlled and communicate through the power line to make a 'smart' home.

It would seem then that using the power line as a communications medium would render all other network strategies obsolete. If the medium was favorable for communication, this would perhaps be the case. However, it is difficult to communicate data effectively because the medium was not designed for data transmission. Attenuation, variable impedance, and noise are three factors which make this a harsh medium, making it difficult to achieve optimum signal transfer, low distortion, and high signal-to-noise ratios (SNR).

Several technologies have been explored to combat the harsh channel characteristics. Low speed technology that is readily available only supports simple home automation. Most recently, high frequency modulation combined with sophisticated multiplexing and error control coding techniques have been explored. This technology provides high speed, however, untwisted power wires radiate the signal and this causes interference with radio communications. What is missing is a 'medium speed' technology that can support applications such as control of devices, network gaming, low resolution image sequences from cameras, security applications and several other applications. The only current power line communication method that can support these applications is the high speed method, which provides much higher speeds than needed and carries a cost that is higher than necessary.

This thesis explores a method of providing medium speed communication while minimizing complexity and cost by using "timed transmission". Prior work has been done in this area by Jack Hanson at the University of Saskatchewan and at Telecommunications Research Laboratories [1]. While previous methods of power line communication use continuous time transmission, this proposed method is similar to time division multiple access (TDMA) in that users are given specific time periods (timeslots) in which to transmit and receive while other timeslots are not used. This idea came from Hanson's work which observed that interference on the power line is synchronous and stationary in position with respect to the 60 Hz mains signal [1]. Noise appears only for short intervals, causing a high number of bit errors for a short period of time. These noisy intervals should be avoided. By detecting when noisy timeslots occur and avoiding

them, bit error ratios can be dramatically reduced. Also, during periods of low noise, the transmission bit rate can be increased to further improve the throughput of the system.

1.3 Objectives Of The Thesis

This thesis has three main objectives. The first objective is to review previous studies on the power line as a transmission medium and previous and current methods of communicating data on the power line. Included in this information will be the results of Hanson's research that indicates power line noise is synchronous and fixed in position with respect to the zero crossings of the 60 Hz mains signal. The second objective is to take advantage of this information by designing a protocol. First, general protocol requirements will be studied along with several existing communication protocols. This study will be used in order to aid in the design of a new protocol for timed power line communication. The third objective is to implement and test the performance of the protocol. Two half-duplex modems will be built and data communication performance will be tested in various noise environments to observe the improvements obtained by using the new protocol. Message transmission success and throughput will be the main focus of the performance study.

2. Introduction to Power Line Communication

Power line communication, also known as PLC, uses existing power distribution wires to communicate data. This, however, is not a new idea. In 1838 the first remote electricity supply metering appeared and in 1897 the first patent on power line signaling was issued in the United Kingdom [2]. In the 1920's two patents were issued to the American Telephone and Telegraph Company in the field of "Carrier Transmission over Power Circuits". One would think that the long-ago conceived idea of power line communications would be well developed by now. However, this is not the case because the power line is not well suited for data communication.

There are two main applications for power line communication - one for broadband Internet access to the home and the other for home and office networking. This work focuses on using power lines for home and office networking.

Home and office networks typically use Ethernet or wireless devices. Ethernet provides high speed networking, but requires dedicated category 5 (CAT5) cabling which would need to be installed in the home. Wireless devices are now becoming more popular and work quite well, but provide speeds that are excessive for simple applications. Performance of wireless networks is also affected by line of sight obstructions such as walls. One major attraction of power line communication is the high availability of power outlets. The average North American home has an average of three power outlets per room resulting in choice of location and mobility – “as long as there is a power socket, there is a connection to the network”. The high node availability is why this technology has tremendous market potential.

Power line communication technology has been slow to evolve because the lines were designed solely for the purpose of 60 Hz main power distribution. Unfortunately, power lines are a rather hostile medium for data transmission. The medium has varying impedance, considerable noise, and high attenuation all which can change as different types of devices are connected to the electrical supply.

2.1 Communications Background

In order to better understand PLC, the following section provides an overview of a general communications system. This will include a discussion of the elements of a communications system, the methods for transmitting data, and performance measures.

2.1.1 Communications System Model [3]

Figure 2.1 shows a simplified model of a digital communications system. The overall objective of a communications system is to communicate information (the transmission of digital information this thesis considers) from a source to a destination over some channel.

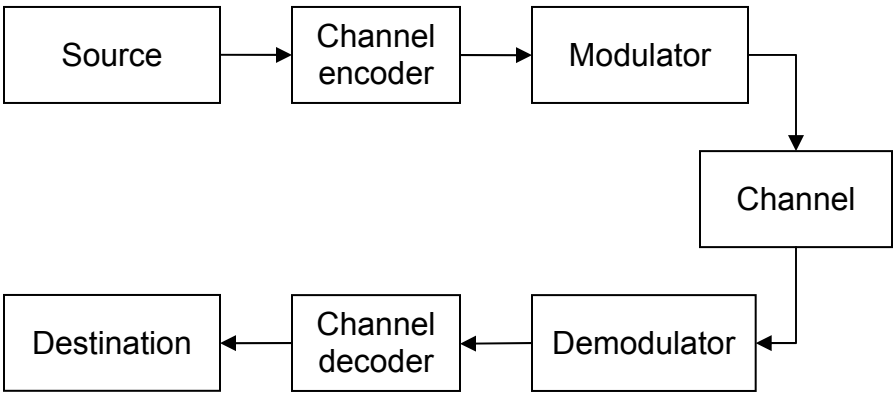


Figure 2.1 Communication system model.

Source and Destination: The source can be any digital source of information. If the source is analog such as speech, then an analog to digital converter must precede the transmitter. At the receiving end, the decoded information is delivered to the destination.

The source may also compress redundant data, which minimizes the number of bits transmitted over the channel, but can also create a loss of source information. The data is unpacked at the destination to either an exact replica of the source information (lossless data compression) or a distorted version (lossy data compression).

Channel Encoder and Channel Decoder: Channel coding reduces the bit error probability by adding redundancy (extra check bits) to the bit sequence. The check bits are computed over a k-symbol input sequence to create an n-symbol output code sequence. This determines the *code rate* Rc where $Rc = k/n$ and $Rc \leq 1$. This is the ratio of the number of actual data bits to the total number of bits transmitted. The channel decoder uses the extra bits to detect and possibly correct errors which occurred during transmission. The number of extra bits added depends on how much error detection and correction is needed. Channel coding (also known as error control coding) is a heavily studied area. It is used to improve performance over noisy channels (such as the power line). Two major classes of codes exist: *block codes* and *convolutional codes*. Block codes are implemented by combinational logic circuits. Reed-Solomon (RS) codes are a popular block code. Convolutional codes (also known as *tree codes* or *trellis codes*) are implemented by sequential logic circuits [4].

Channel Modulator and Channel Demodulator: The purpose of the modulator is to take the encoded data and produce an analog signal suitable to propagate over the channel. The data is converted from a stream of bits into an analog signal. An M-ary modulator takes a block of Y binary digits from the channel encoder to select and transmit one of M analog waveforms at its disposal where $M = 2^Y$ and $Y \geq 1$. At the receiver, the demodulator tries to detect which waveform was transmitted, and convert the analog information back to the sequence of bits. Modulation is typically performed by varying the amplitude, the phase, or the frequency of a high-frequency *carrier* signal. For example, if the input signal of the modulator is used to vary the amplitude of the carrier signal, the modulation is called Amplitude Shift Keying (ASK). There are several other modulation techniques including FSK (Frequency Shift Keying), PSK (Phase Shift Keying) and QAM (Quadrature Amplitude Modulation).

Channel: The channel can be any physical transmission medium including coaxial cable, twisted pair, optical fibre, air, water, or for this work - the power line. It is important to know the characteristics of the channel, such as the attenuation and noise level because these parameters directly affect the performance of the communication system.

2.1.2 Description of Important Performance Parameters

Symbol Rate - This is the transmission rate or number of symbols per second from the modulator. If the signal duration is T seconds then the symbol rate is 1/T symbols per second. The symbol rate is also known as the baud rate [4].

Bits per second (bps) – Also known as bit rate, bits per second is directly related to the symbol rate. If each symbol represents Y bits and the symbol rate is 1/T baud, then the bit rate is Y*(1/T) bps. On high quality channels it is easier to send more bits with one symbol, resulting in higher bps [4].

Bit Error Probability (P_b) – P_b is the probability that a bit is incorrectly received at the destination. This is an important performance measure for any digital communication system that is affected by noise and the disturbances in the channel.

Bandwidth (BW) – The range of frequencies used by the communication system. For a specific communication method, the bandwidth needed is proportional to the symbol rate. Bandwidth is a limited resource and is often constrained to a certain small range.

Bandwidth Efficiency – This is the ratio between the bit rate and the bandwidth of a communication system (bps/BW). Today a telephone system can achieve a bit rate of 56.6 kbps using a bandwidth of 4 kHz, so the bandwidth efficiency is 56.6/4 = 14.15 bps/Hz.

Noise – This is an unwanted signal on the channel that interferes with the desired signal. Noise on the power line is a sum of many different disturbances originating from devices such as television receivers, computers, and vacuum cleaners. The amount of noise can drastically affect the quality of communication.

Attenuation – When the signal is propagating from the transmitter to the receiver the signal gets attenuated (loses power). If the attenuation is high, the received signal power can become low and might not be detected. Attenuation is shown to be high on a power line, and this puts a restriction on the distance from the transmitter to the receiver.

Signal to Noise Ratio (SNR) – This is the ratio of received power to noise power. A higher SNR makes for easier communication because noise has a smaller effect on the signal. SNR is also affected by attenuation, which reduces signal power and thus SNR. SNR can be increased by using filters to reduce noise outside of the bandwidth occupied by the signal.

Diversity – Used to reduce the error probability of harsh channels. Examples of diversity are *time diversity* and *frequency diversity.* In time diversity the same information is transmitted at different time instants with the idea that if the channel is bad at some time instance it might not be at another. Frequency diversity transmits the same information in different frequency bands. It can be compared to having two antennas transmitting at different frequencies; if one of them fails the other might work. Several variations of time and frequency diversity exist. This thesis explores a form of time diversity, although not exactly as described above.

2.2 Prior Art and Modern Methods

2.2.1 X10

The X10 specification was designed for low-bandwidth signaling over power lines within the home. The product was developed by a company in Scotland - Pico Electronics - with the first shipped product to market in 1978. The patent on the standard has since

expired and prices have fallen sharply. X10 applications include controlling lights and thermostats as well as devices like the stereo amplifier, garage door opener, television receiver and more [5].

The X10 system is simple and easy to use. It transmits over the electrical wiring using on off keying (OOK). More precisely, it uses 120 kHz signal bursts, each one millisecond long. These signal bursts are synchronized to the zero crossings (both positive and negative) of the ac power line signal. The specification allows for a signal burst to be within a maximum of 200 µs of the zero crossing point. Each bit transmitted occupies two zero crossings; a binary 1 is represented by a burst followed by a no-burst, while a binary 0 is a no-burst followed by a burst. Also, each one millisecond burst is equally transmitted 3 times to coincide with the zero crossing point of all three phases in a three phase distribution system. Figure 2.2 shows the timing relationship of these bursts relative to the zero crossing.

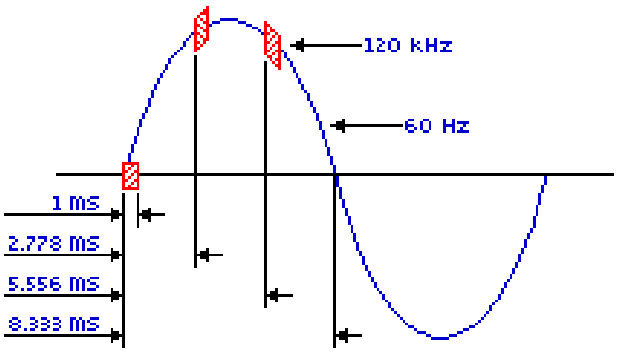


Figure 2.2 X10 timing on 60 Hz waveform [5].

An X10 packet, shown in Figure 2.3, encompasses eleven cycles of the power line. It begins with a start of packet identifier consisting of the sequence 'burst, burst, burst, no-burst', which occupies the first two cycles (four zero crossings). The next four cycles represent the House Code, and the last five cycles represent a number code (1 through 16) or a Function Code (On, Off, etc.) This complete block is always transmitted twice, with 3 power line cycles between each group of 2 codes. Hence the total number of power line cycles required to complete a transmission is 2*11 + 3 = 25 power line cycles [6].

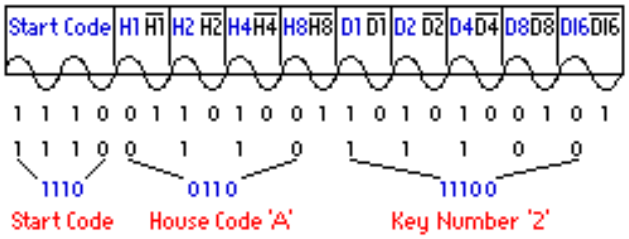


Figure 2.3 X10 packet format [5].

With this simplicity also comes a low bit rate. 25 power line cycles are required to transmit a frame - which consists of 11 bits. The resulting bit rate is then 60*(11/25) = 26.4 bits/second. This bit rate is only useful for trivial applications, and is much too slow for transmission of audio, video, network gaming traffic, and other higher bandwidth network traffic. However, X10 is inexpensive and easy to use; hence it is a popular choice for basic home automation.

2.2.2 Lonworks

Lonworks is a network protocol created by Echelon Corporation [7] and is intended to support communication between control devices or nodes. Each node in the network – a switch, sensor, motor, motion detector, etc. - performs a simple task. The overall network performs a complex control application such as automating a building [5].

Early standards for the Lonworks protocol used spread spectrum modulation. Spread spectrum communication techniques can be used to improve performance in the presence of tonal noise (noise that is present at specific frequencies only). Spread spectrum improves performance by using a wider bandwidth for communication than what is required. The amount of improvement depends on the available bandwidth, or in other words, the degree of spreading. It was first used with a bandwidth of 100 kHz - 400 kHz, but this band was found to be too narrow to provide acceptable performance given the type of noise present on the power line. In addition, European regulations prohibit power line signaling above 150 kHz due to potential interference with low frequency licensed radio services [8].

Instead, Echelon's latest product, the PLT-22 transceiver, operates using a novel Dual Carrier Frequency mode along with Digital Signal Processing (DSP). The purpose of the DSP is to provide adaptive carrier and data correlation, impulse noise cancellation, tone rejection, and low overhead error correction. The PLT-22 communicates using BPSK with frequency ranges 125 kHz - 140 kHz (primary) and 110 kHz - 125 kHz (secondary). The primary frequency range is used unless impairments prevent successful communication in this range. When this occurs, the PLT-22 automatically switches to the secondary frequency range. The PLT-22 communicates at a raw bit rate of 5 kbps. This is much faster than X10, and hence more useful for more complex control of electrical devices [9].

2.2.3 CEBus

In 1984, the Electronic Industries Alliance (EIA) Consumer Electronics Group began an effort whose goal was the formulation of a standard for a communication network for consumer products in the home. The standard came to be called the Consumer Electronic Bus (CEBus) [10]. The suite of specifications includes communication on many different types of medium including power line, twisted pair cable, coaxial cable, infrared, radio frequency, and fibre optic. The suite of specifications was labeled EIA-600. The full specification was released in 1992.

This work is concerned with the physical layer coding employed by CEBus. CEBus uses non return to zero (NRZ), pulse width encoding. There are four symbols: ‘1’, ‘0’, EOF, EOP. These symbols are encoded using chirp spread spectrum in the bandwidth 100 kHz to 400 kHz. In spread spectrum, the carrier signal frequency is swept over a range of frequencies. CEBus employs a sequence of up and down frequency sweeps of the carrier that in total for one symbol occupies a period of 100 μs. This symbol interval is the shortest symbol time “1”, or unit symbol time. A 0.1 % margin of error is also defined (100 ns for 100 μs). Also, the time to transmit binary “1” is a unit symbol time (100 μs), while to transmit a binary 0, two unit symbol times are used (200 μs). For random binary

data, the average symbol time is then 150 μs, for a bit rate of 7.5 kbps. A unit symbol time is shown in Figure 2.4.

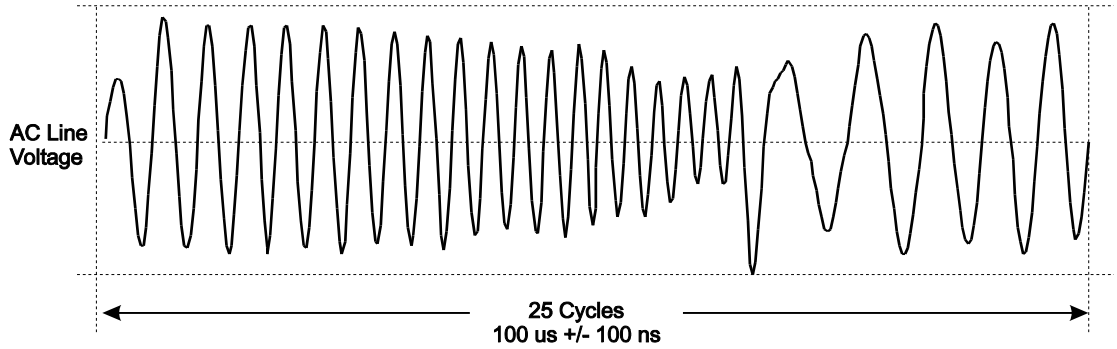


Figure 2.4 CEBus spread spectrum chirp [10].

One other interesting note in the CEBus standard is coupling between power phases within the home. There are two 60 Hz phases, L1 and L2, in a home that are 180 degrees apart in phase. Home 120V electrical devices - appliances, lights, motors, etc. - normally connect to either L1 or L2. Only 240V devices that connect to L1 and L2 simultaneously provide a signal path between these two branches other than the minimal coupling provided by the distribution transformer. Therefore, a CEBus 120V device on L1 may not communicate with a CEBus 120V device on L2 due to inadequate signal coupling between L1 and L2. To help solve this problem, the CEBus standard says that a signal coupler should be placed between L1 and L2 when needed to improve signal propagation within the power line network [10].

2.2.4 HomePlug

HomePlug is a non-profit consortium founded in March 2000 by thirteen leading IT companies who have a mutual interest in high-speed networking technologies over power lines. Its membership of now more than 80 companies includes companies specializing in semi-conductor manufacturing, hardware/software supply, and service. The goal of the consortium is to create an open specification for high speed power line networking technology and to promote new products to accelerate its adoption. In June 2001, the HomePlug v1.0 specifications were published.

The HomePlug specification is the most complex of all power line technologies. To achieve higher bit rates, higher frequencies and bandwidth must be used than that for X10, Lonworks and CEBus as discussed above (whose frequencies are less than 500 kHz). HomePlug communicates using Orthogonal Frequency Division Multiplexing (OFDM) in the 4.49 to 20.7 MHz frequency band. This method of multiplexing divides up the available bandwidth into sub-bands. These sub-bands are mathematically orthogonal, meaning that for the specific symbol rate they are placed at specific intervals in the frequency domain that minimizes interference between them. In the bandwidth 0 - 25 MHz, there are 128 evenly spaced sub-carriers of which HomePlug uses 84, from the band 4.49 to 20.7 MHz (carriers 23-106 inclusive) [11].

Before forming a symbol to be transmitted, data bits are processed using several error control coding schemes. Data bits are modulated onto the sub-carriers using differential quadrature phase shift keying (DQPSK), or differential binary phase shift keying (DBPSK). The Inverse Fast Fourier Transform (IFFT) is used at the transmitter to create individual channel waveforms. The whole process is reversed at the receiver [12].

In addition to this, HomePlug adapts to channel conditions. Special frames are sent and analyzed by receivers to determine which of the 84 sub-carriers are available for communication. Tone Maps (TM) are then created and used by sender-receiver pairs to adapt to varying channel conditions. Only good sub-carriers are used for communication. Also, the modulation scheme can be changed (DBPSK or DQPSK), and the error-control coding can be modified. Altogether, 139 distinct physical data rates are available from 1Mbps to 14.1 Mbps [11].

Several manufacturers have demonstrated HomePlug technology and it looks promising. Field tests with HomePlug V1.0 devices in 500 homes show that 80% of outlet pairs were able to communicate with each other at about 5 Mbps or higher, and 98% could support data rates greater than 1 Mbps [13]. The HomePlug alliance has announced plans for the development of next generation specifications. Named HomePlug AV, the new specification will be designed to support distribution of data and multimedia-streaming

entertainment including High Definition television (HDTV) and data rates of 100 Mbps throughout the home [11].

The ability to adapt is the real strength of HomePlug. Obviously if the power line channel becomes harsh for communication, data rates will be slow, but reliability will be maintained. Note that HomePlug employs a method of frequency diversity. It also uses complex error control coding and modulation techniques that are good for reliability, but are computationally demanding, power consuming and expensive. HomePlug provides high enough data rates for medium speed, but its complexity and cost is more than necessary.

2.2.5 The Need for a Medium Speed Technology

In the coming years, people are likely to use PLC to network anything that is electrically powered such as heating, ventilation, and air conditioning (HVAC). Different devices and applications will have different throughput requirements ranging from 10 bps to 100 Mbps or more. Simple control of devices (turning lights on and off, controlling thermostat, etc.) is achievable with low bit rates (< 5 kbps). High quality video and high-speed computer networking is at the opposite end of the spectrum requiring speeds up to 100 Mbps.

From the different technologies given above one can see that there exists low cost, reliable systems (X-10, Lonworks, and CEBus) and that high speed systems exist using the HomePlug protocol. These systems work well for what they were designed, but have several drawbacks. There really isn't a device designed for "medium speed" (100 kbps to 1 Mbps). Therefore this thesis focuses on a low complexity, low cost medium speed technology for power line communication using "timed transmission".

2.3 The Channel

The characteristics of the channel must first be explored in order to design a timed transmission protocol. Varying impedance, considerable noise, and high attenuation are

the main issues. Channel characteristics depend on the location of the transmitter and receiver in the specific power line infrastructure and are both time and frequency dependent. Hence, the channel can be described as random time varying with a frequency-dependent signal-to-noise ratio (SNR) over the communication bandwidth [14].

2.3.1 Attenuation

Figure 2.5 shows attenuation measurements taken on a single power phase in the Engineering building at the University of Saskatchewan by Hanson [1]. Measurements were taken at outlets separated by 4 meters (13 feet) and 23 meters (75 feet).

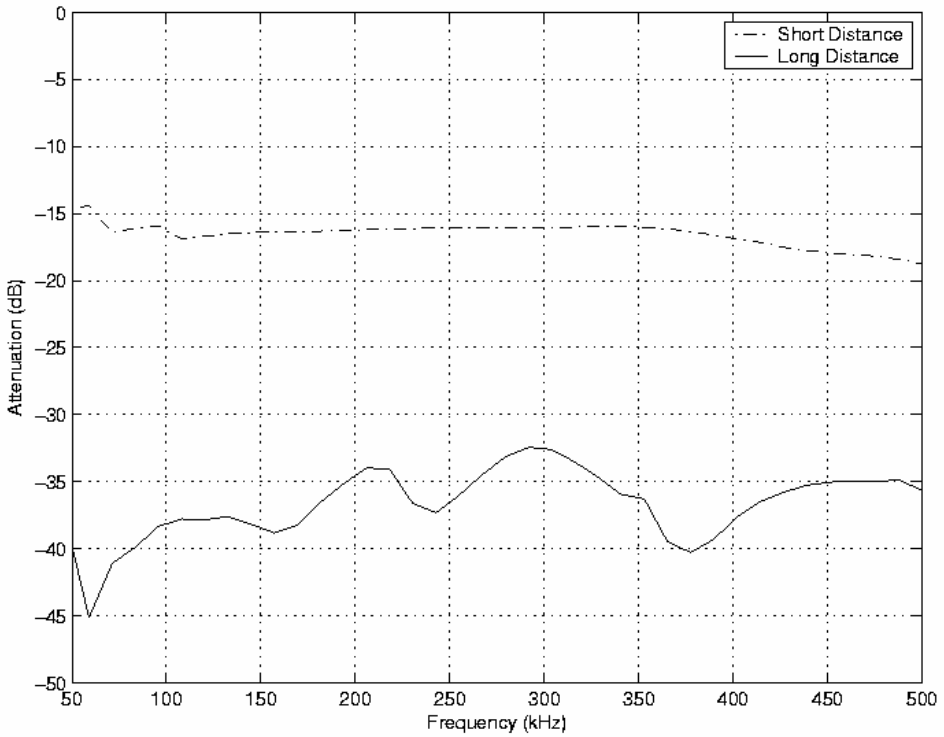


Figure 2.5 Attenuation vs. Frequency [1].

The short distance measurement, performed on hallway outlets free of electrical loads, shows a relatively flat attenuation between 15 and 20 dB. The long distance measurement, also free of electrical loads, shows attenuation between 33 and 43 dB with some variation with frequency.

Also, branches (stubs) on power cables can create nulls in the transmission characteristic. However, assuming stub lengths are less than 100 meters, nulls in the transmission characteristic which occur at a stub length of λ/4 will not occur at frequencies less than 1 MHz. (100 meter stub creates a null at 1 MHz, and shorter stubs create nulls at > 1MHz).

2.3.2 Impedance

Power line impedance is important because a transmitter must match this impedance over the desired frequency range to avoid frequency dependent distortion of a broadband signal. Nicholson and Malack [14] measured line impedance in the frequency range 20 kHz to 30 MHz at 36 different commercial locations in the United States. They found that the characteristic impedance increased with frequency. The impedance, averaged over all sites, was approximately 1 Ω at 20 kHz increasing to 100 Ω at 30 MHz. Similar results were obtained in European countries and in Japan. Nicholson and Malack explained that any variation from site to site was attributed to variations in load connected to the line. Figure 2.6 is extracted from their work to show the frequency range 50 to 500 kHz.

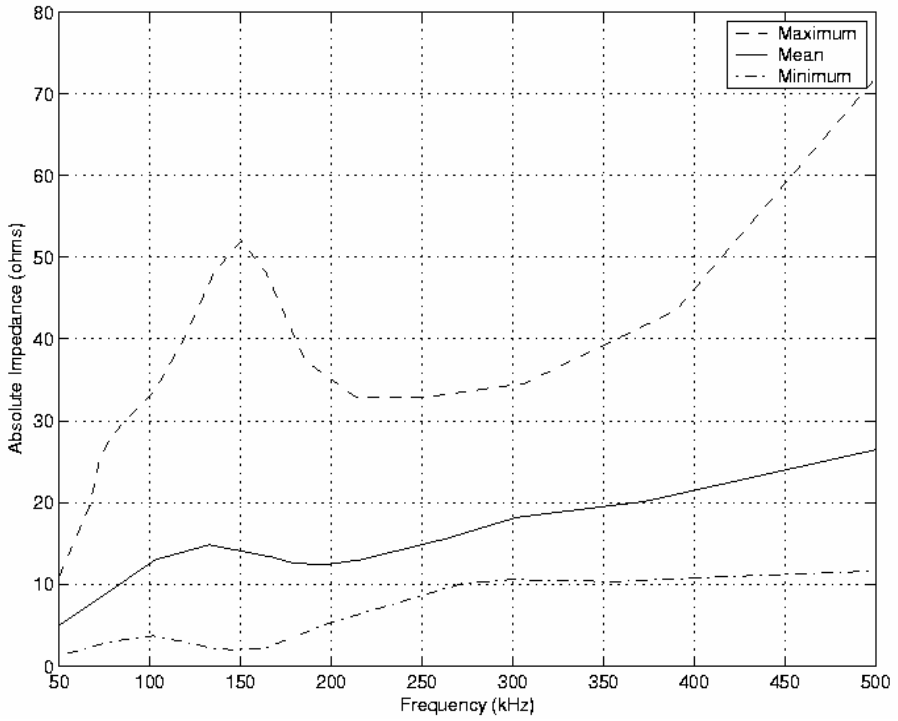


Figure 2.6 Impedance vs. Frequency [14].

2.4 Time Characteristics of Noise

Prior research by Hanson has also shown that the majority of noise on the power line is impulsive and synchronous in time with the power signal (60 Hz in North America). In other words, the noise occurs at regular intervals that are fixed in time with respect to zero crossings of the 60 Hz mains signal. This means that at these specific times the probability of bit error is much higher than that at other times during one cycle of the power line signal. All prior communication methods available for study have ignored this fact and have been designed to get around the noise problem by analyzing the frequency characteristics of the power line noise rather than the time characteristics. Given that the power line noise is synchronous with the 60 Hz mains frequency, a method can be devised to avoid these times of high noise and high bit error probability and communicate during low noise times. This will dramatically reduce bit error probability, and increase throughput.

2.4.1 Noise due to Office Load Devices

To investigate the communications capability and the time characteristic of noise on the power line conductors, Hanson performed measurements at several outlets within two commercial buildings. The first was the Engineering building at the University of Saskatchewan campus and the second was at Telecommunication Research Laboratories (TRLabs) which is located in an office building in a research park. The representative trace shown in Figure 2.7 was recorded in an office at TRLabs.

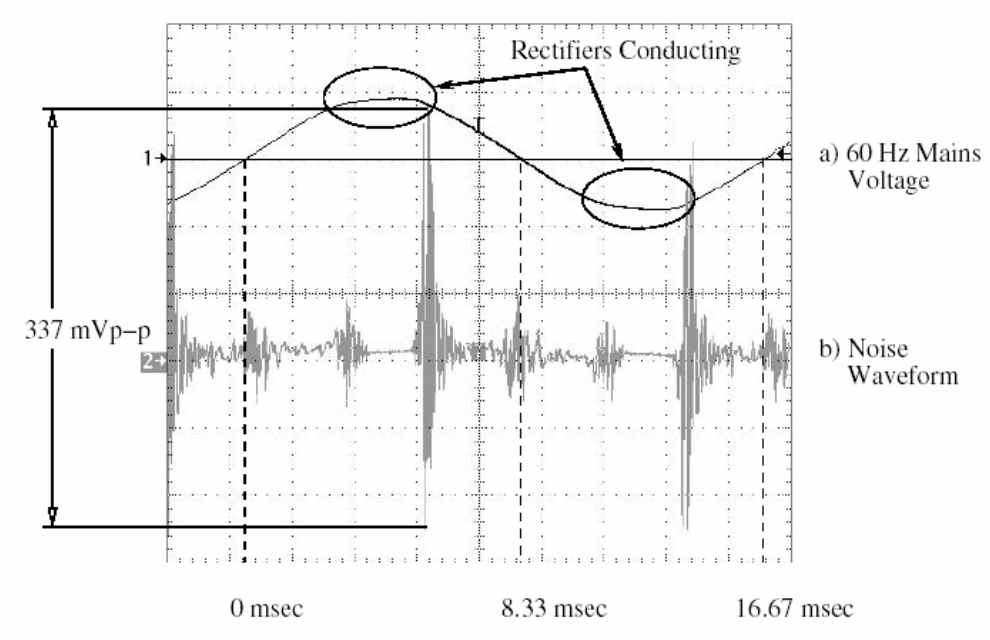


Figure 2.7 Measured noise in an office environment [1].

During each 60 Hz cycle, there were six noise bursts: one at each zero crossing caused by fluorescent lamps extinguishing and reigniting, a pair caused by power supply rectifiers when they begin to conduct and another pair caused by power supply rectifiers when they stop conducting.

Hanson also developed a simple laboratory data transmission system to evaluate transmission errors caused by noise on the power conductors. The data source was a 32767-bit maximal length sequence at the rate of 50 kb/s constructed with a shift register with taps that are exclusive-ORed together to create the sequence. To simplify clock recovery at the receiver, the source data was coded into a bi-phase level (Manchester I) data stream by exclusive-ORing the source data with a clock that is at twice its data rate. This coded data stream was used to frequency shift key (FSK) a 200 kHz carrier. The modulator circuit was developed around an Exar XR-2206 integrated circuit and the mark and space frequencies were set at 150 kHz and 250 kHz respectively.

In the receiver, errors were detected using a self-synchronized descrambler. This is a shift register that shifts in the received data stream and has taps at specific locations in the

register. An exclusive-OR combination of these taps indicates if there is any bit in error in the register. This is possible by constructing a shift register with taps like that at the source. Because a single bit error cycles through the shift register, it will go through the location of each register tap and hence the error indicator will indicate multiple bit errors (in Hanson's design there was three) for a single bit error. A Tektronix 544A digital storage oscilloscope was used to display the 60 Hz signal, the noise signal, and the detected error signal. By setting the oscilloscope to average over 10-minute intervals, the trace voltage graphically illustrated the average error rate. Traces were transferred to a PC at 10 minute intervals and averaging was done over a 9-hour period. Figure 2.8 shows the resulting traces.

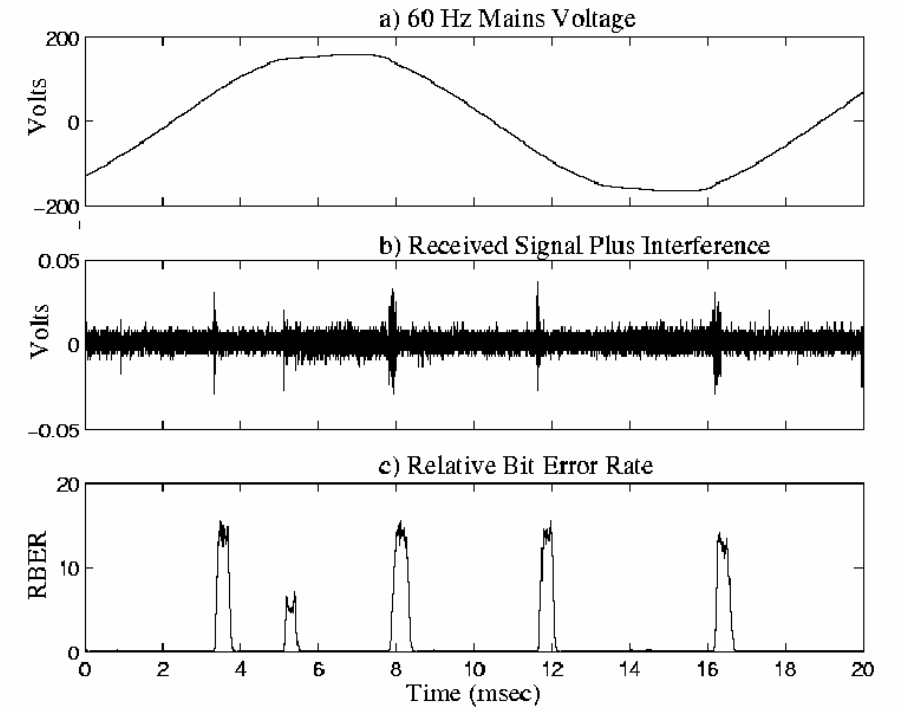


Figure 2.8 Bit error test in an office environment [1].

The traces in Figure 2.8 were obtained in a typical office environment with each office having one or more personal computers and two computer work areas with several workstations present in both. The simple error detector clearly shows the strong correlation between the stationary interference and the errors, and that certain portions of

the cycle should be avoided during transmission. In this case, approximately 10% of the 60 Hz mains cycle is unusable due to noise [1].

2.4.2 Noise in the Home Environment

The home environment frequently includes lamp dimmers and this presents an additional noise source. These devices switch off at the zero current point in the 60 Hz mains cycle and switch on again after a delay that is increased to make the lamp more dim. The switch-on results in a large and rapid current transition that introduces large high frequency noise resulting in data errors at the switch-on time. Figure 2.9 shows how the control setting for lamp dimming affects the position of the shaded error bursts within the 60 Hz mains cycle. This test was conducted using the same transmitter/receiver pair as in the office test. Bit errors caused by the lamp dimmer are shaded.

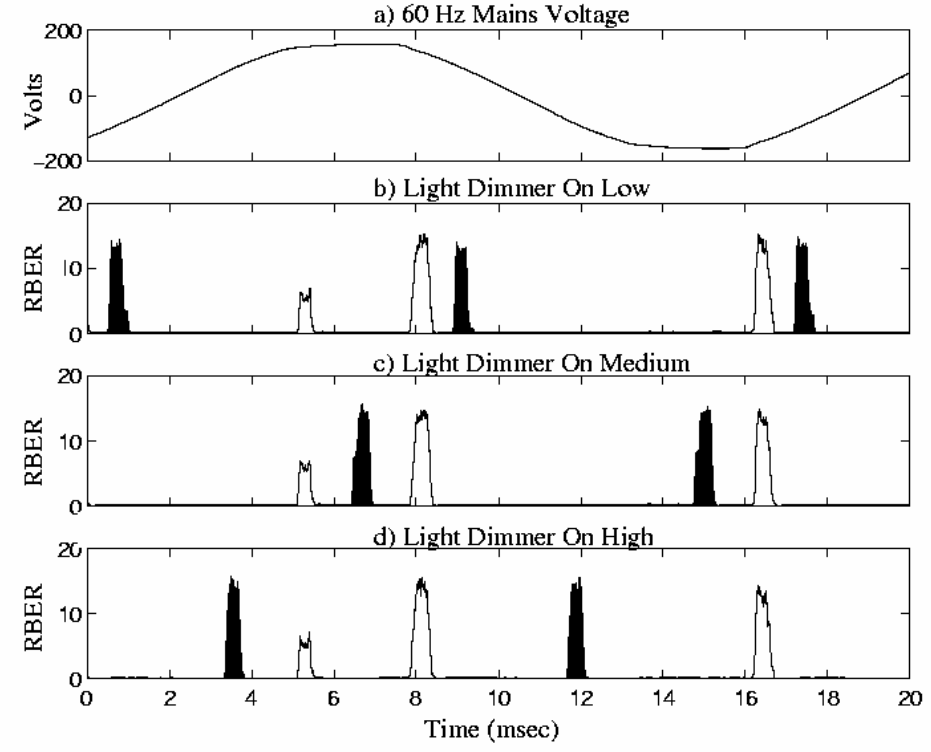


Figure 2.9 Bit error test in a home environment with a lamp dimmer [1].

Again, observe two key points: noise times and hence bit error times are synchronous with the power line cycle, and the error times can change depending on the type of load (in this case, changing the brightness of the lamp).

2.4.3 Noise Frequency Spectrum

The frequency spectrum of the noise is also important. If noise is outside of the frequency range that is being used for communications, it can be filtered and its effect can be minimized or eliminated. Figure 2.10 shows an oscilloscope trace of the frequency spectrum of noise present on a typical office power line. This trace was obtained at the offices of TRLabs. The horizontal scale is 50 kHz/division and the vertical scale is 20 dB/division. The horizontal axis displays frequencies from 0 to 500 kHz. The conclusion that can be drawn from this trace is that the frequency spectrum of the noise is approximately white – meaning that it is over all frequencies, and cannot simply be filtered to eliminate its effect on communications.

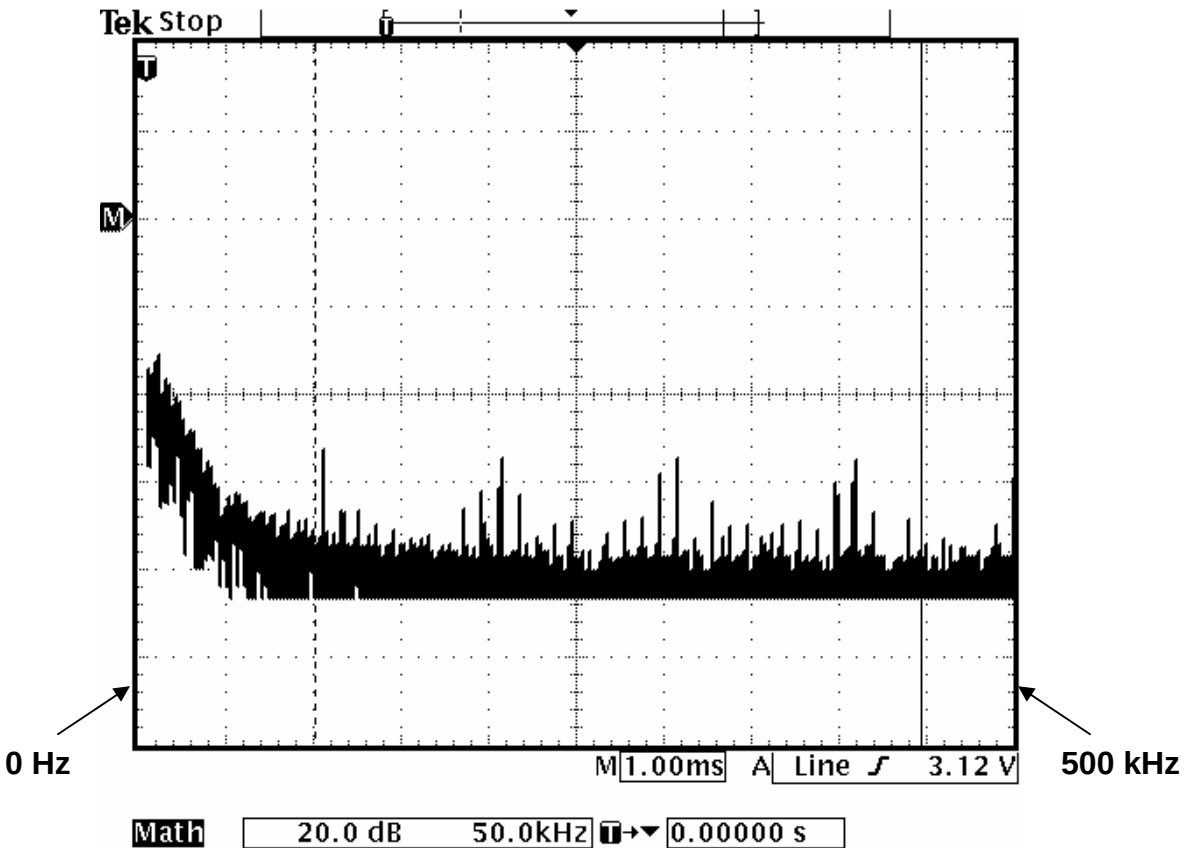


Figure 2.10 Noise frequency spectrum

3. Design of a Protocol to Seek and Adapt to Power Line Noise

3.1 The Need for a Protocol

Hanson's research [1] has shown that noise on the power line is impulsive and synchronous in time with the power signal (60 Hz in North America). The noise occurs at regular intervals that are fixed in time with respect to zero crossings of the 60 Hz mains signal and at these times there is higher probability of data errors. To take advantage of this information, a communications protocol must be developed.

A communications protocol, in a general sense, is a set of underlying rules that prescribe the nature and form of interactions between a network's interconnected devices [15]. Many protocols exist, each with specific conditions in mind. By drawing on the design of other protocols and incorporating new information from Hanson's research, a new protocol can be developed that is suitable for data communication on the power line.

3.2 Choices in Designing a Protocol

There are several variables to consider in the design of a communications protocol. These include the type of medium that is targeted, system architecture, and flow and error control techniques. The following is a review of the choices that will be considered in the design of this protocol.

3.2.1 Distributed or Centralized Architecture

The first consideration is whether to design a distributed or centralized architecture. The difference lies in how control of the system is distributed.

A centralized architecture has some type of controller that implements the control logic for all the I/O points connected to it. This type of system can consist of sensors and actuators that are controlled by some central controller [7]. This can be viewed as a Master/Slave setup where there is one Master and several Slaves. The Master is in charge of coordinating who can talk and when.

A distributed system allows a number of *intelligent* devices to communicate directly with each other. No intervening supervisory controller is required to poll devices for information and then retransmit that information to other devices. No supervisory device is charged with responsibility for system-wide control algorithms. This means that every device is capable of publishing information directly to other devices on the network [7].

3.2.2 A Multiaccess Medium

In the case of the power line medium, nodes are not joined by one-to-one communication links. Rather, the medium is one in which the received signal at one node depends on the transmitted signal at one or more other nodes. Typically, such a received signal is the sum of the attenuated transmitted signals from a set of other nodes, corrupted by distortion, delay, and noise. Such a medium is called a multiaccess medium. A *medium access control* (MAC) method is needed to manage communication on a multi-access medium. The purpose of the MAC is to allocate the multiaccess medium (the power line) among the various nodes [16].

There are two general access methods, multiple access and random access. Applications requiring continuous transmission (e.g., voice and video) are generally allocated dedicated channels for the duration of communication. Sharing bandwidth through dedicated channel allocation is called multiple access. Bandwidth sharing for users with bursty transmissions (like data) generally use some form of random channel allocation that does not guarantee channel access. Bandwidth sharing using random channel allocation is called random access [17].

3.2.3 Channel Sharing

Methods of channel sharing include frequency-division (FDMA), time-division (TDMA), code-division (CDMA), and combinations of these methods. In FDMA, the total system bandwidth is divided into orthogonal channels that are nonoverlapping in frequency and are allocated to the different users. In TDMA, time is divided into nonoverlapping timeslots that are allocated to the different users. In CDMA, time and bandwidth are used simultaneously by different users, modulated by orthogonal or semi-orthogonal spreading codes. Using the orthogonal spreading codes the receiver can separate out the signal of interest from the other CDMA users with no residual interference between users [17].

3.2.4 Messages and Packets

In an application requiring communication between two or more users, a message is one unit of communication from one user to another. This message is represented as a string of bits. In interactive communication, user 1 might send a message to user 2, user 2 might reply with a message to 1, who responds with another message, and so on.

Even though a message is a unit of communication, it may not be feasible to send an entire message at once. The message can actually be broken down into shorter bit strings called packets. Packets can then be assembled at the receiver to re-form the message. Packets contain control overhead which is necessary to facilitate communication, ensuring reliable communication, controlling congestion, achieving synchronization, and so on [16]. The format of packets is important in order to perform this function.

The size of packets is also an important consideration. If packets are small, then reliability will be high because a small packet has a lower probability of error. However, the actual data throughput using small packets will be lower because each packet requires overhead (thus the data to overhead ratio is low). By using larger packets the data to overhead ratio is higher, but reliability may suffer because larger packets have a higher probability of error. The optimum size of packet depends on the error rate of the channel, and the desired reliability and data throughput. This concept is shown in Figure 3.1.

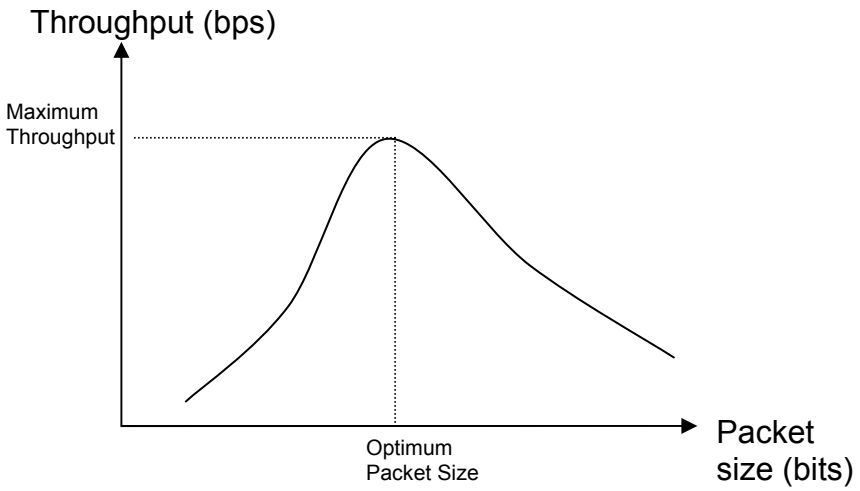


Figure 3.1 Data Throughput vs. Packet Size.

3.2.5 Efficient Use of the Channel

As with any communication method, efficiency is important. Efficiency in this sense refers to fair and quick access to the channel's available communication capability. The basic considerations are as follows:

- When there is only one user of the channel, that user should be able to use the entire communication capability of the channel.
- When there is more than one user, each user should have equal access to the channel.
- There should be low delay to gain channel access.
- Collisions among communicating devices should be minimized.

If the number of users is dynamically changing, the protocol should have the ability to detect when nodes need to send data and be able to quickly reassign access to the channel. There is also the issue of priority – whether a certain user is more important or has a more important message to send. Nodes with higher priority should have greater access privilege to the channel [18].

3.2.6 Error Detection

Electromagnetic waves traveling over a transmission medium may encounter noise, causing errors. It is important to note that errors occur not only because of channel interference, but also because of collisions (more than one node trying to send at a time). Therefore there must be some method that quickly and properly detects bit errors.

Single-bit errors are the most common type of data error and are the easiest to detect and correct. However, multiple-bit errors or burst errors are possible too. For example, the byte 11000001 may change to 10000000. Burst errors occur when noise interferes with the transmission for a longer period of time causing a change in several consecutive bits.

A variety of methods are available for detecting transmission errors. Extra bits add a form of redundancy, and must be appended to a packet to detect any errors. The simplest form of error detection would be to send a copy of each bit, and if the receiver sees that there is a disagreement in the first and second bit, it knows there is an error. This is inefficient, because it requires that the entire data stream is sent twice. It is also ineffective in the presence of error bursts.

Practical methods require less overhead and the probability of detecting errors is much higher, as well some methods are capable of correcting errors. The simplest method is parity coding. Packets are organized into blocks of bits, and a parity bit is attached based on the number of 1's in the data block. Even parity means that there is an even number of 1's in the encoded data, including the parity bit. Similarly, odd parity means there are an odd number of 1's in the encoded data, including the parity bit. This scheme detects all single bit errors and burst errors that affect an odd number of bits, but does not provide error correction capability [18] [15].

An extension of parity is an arithmetic checksum. In this method, the sender divides the sending data unit into equal segments of *n* bits. Then ones-complement arithmetic is used to add the segments together to get the result in n-bit form. In ones-complement arithmetic, **the final carry is added to the binary sum**. For example, the sum of the 4-

bit words 1000 + 1101 = 0101 + 1(final carry) = 0110. This sum is complemented and appended to the data as the checksum field. At the receiver, the received data (including the checksum field) is divided into segments of *n* bits and when added together should give a result of all 1s. If not, a data error has occurred. This effectively computes parity vertically, where we have *n* columns and some number of rows. This detects all 1 bit errors, all burst errors up to length *n*, and some larger burst errors. This is shown in Figure 3.2.

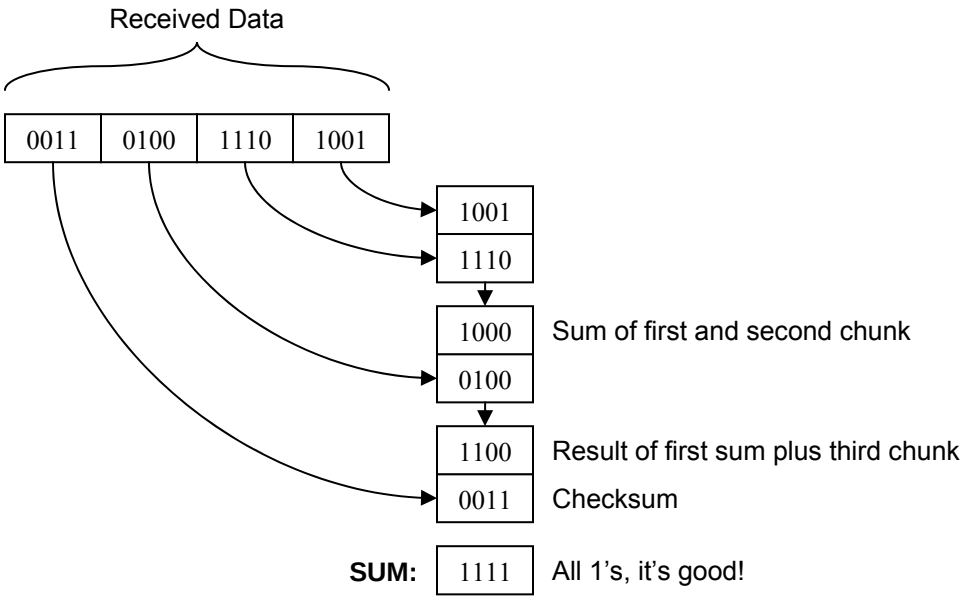


Figure 3.2 An arithmetic checksum.

The idea of parity check can also be extended to include parity check codes. The idea is to start with a bit string and to generate parity checks on various subsets of the bits. The transformation from the string of data bits to the string of data bits and parity checks is called a parity check code or *linear code*. The word *code* refers to the transformation itself, and we refer to an encoded bit string (data plus parity checks) as a *code word* [15]. Several variations of linear codes exist, one of the most popular being *Hamming codes,* which are capable of detecting multiple bit errors and correcting single bit errors.

Another method is to use a Cyclic Redundancy Check (CRC). This is an efficient method of generating codewords at the transmitter and checking at the receiver. These are most often used for error detection today. Bit strings are treated as polynomials, and nodes perform algebraic manipulation of these polynomials using shift registers and flip-flops. A modulo-2 division is implemented by means of a generator polynomial G(x) of degree n over the message polynomial M(x) representing the message. The n-bit check sequence or the checksum is the remainder of the modulo-2 division of M(x) by G(x). This checksum is appended to and sent with the message M(x). The receiver performs the same operation on the received data by dividing M(x) (including the CRC) by the same G(x) and the result of the division will give a remainder of 0. If the result is not 0, transmitting errors occurred.

When a station finds errors, it must perform some form of error control, which is discussed next.

3.2.7 Flow and Error Control

Flow and error control procedures establish the rules for exchanging data between devices over a communication link. They ensure that all the related packets arrive at their destination accurately and in order [19]. Physical limitations of the stations must be taken into consideration. Buffer space, processing capability, and transmission line errors are some limitations that must be considered in obtaining proper flow and error control.

A common error control approach is to have the receiver send a message to the transmitter indicating that an error has occurred in the preceding transmission, or that it received the preceding transmission correctly. The message effectively tells the sender whether or not it needs to resend any packets. Thus this type of error control is usually called automatic repeat request (ARQ) [15].

Until the transmitter has received an acknowledgement (ACK) that a packet was received successfully, it must store that packet in its memory for retransmission if necessary.

There are two major strategies for storing packets until they have been acknowledged. These are the *stop-and-wait* and *sliding window* protocols.

The stop-and-wait protocol is used when receiver buffer space is limited, and the sender may not continue to transmit until an ACK is received. The sender sends one packet and then waits for an acknowledgement from the receiver before resuming transmission. This method is simple, but has shortcomings. If the transmitter sends a packet that gets lost on its way to the receiver, no ACK will come back. To solve this, the transmitter must have a 'time-out' period after which time it will re-send the pending data packet. Another problem arises if the receiver sends an ACK and it gets lost on its way back to the transmitter. The transmitter will time-out and resend the data packet resulting in duplicate copies of the packet at the receiver. However, since there is only one packet pending at a time, a duplicate can be detected easily by using a sequence bit that alternates between 0 and 1. If two consecutive packets arrive at the receiver with the same sequence bit, acknowledgement loss or damage must have occurred and the receiver simply discards one packet. The same concept of the stop-and-wait protocol for single packets applies when considering ACKs for entire messages [15].

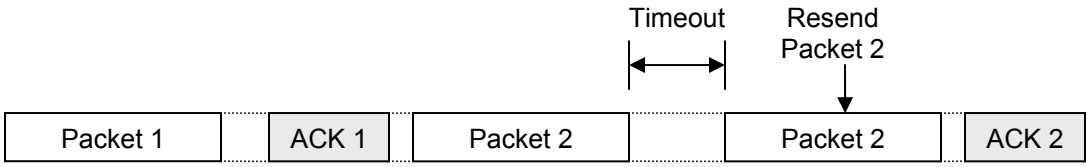


Figure 3.3 Stop-and-wait protocol.

The alternative to the stop-and-wait protocol is the sliding window protocol. In this protocol, the sender keeps track of sent and correctly received packets by using two counters. Figure 3.4 shows two arms (as in the hands of a clock) that represent the sent packet counter and the correctly received packet counter. In the illustration, the sent packet counter indicates that packets up to and including number 6 have been sent. The correctly received counter is advanced when the receiver sends back an ACK and in the illustration, the counter indicates that packets up to and including number 4 have been

correctly received. The difference between the sent packet count and the correctly received packet count indicates the number of packets that have been sent but not acknowledged. Modulo 8 counters have been shown in the illustration and these counters overflow or "wrap around" when the sender continuously sends a long sequence of packets. In the protocol, the number of states in the counter is known as the window size and in this example the window size is 8.

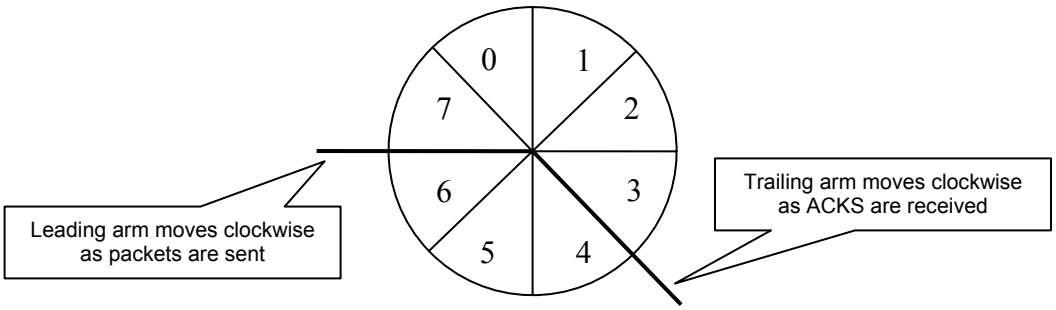


Figure 3.4 Sliding window protocol, sender's "window" [15].

The receiver's window is designed in a similar fashion. The two sliding arms indicate the receive packet counter (leading arm) and the acknowledged packet counter (trailing arm). As a packet is acknowledged, the two counters advance to indicate the range of packets that the receiver is expecting. If a packet is received outside of this range, it is taken as a lost ACK and the packet is discarded [15].

This is an extension of the stop-and-wait protocol that allows a sequence of packets to be processed simultaneously, resulting in more efficient use of the channel. This requires that packets have *sequence numbers*. This creates a limiting factor, because the number of pending packets between sender and receiver may not exceed the limit allowed by the sequence number. That is, if the sequence number consists of 3 bits, then packets 0-7 may be generated. The same sequence number must be used for the next group of packets.

Notice that when using the sliding window that the packets are not always numbered 0-7, but rather a continuous increment. Also, to allow error-free operation of the protocol, the maximum number of pending packets may not exceed half of the window size (ie. 4 for a window size of 8). To prove this, consider a transmission scenario allowing 5 packet transmissions. Assuming the start sequence number from 0, the transmitter may send packets 0-4 and will record these in the sending window. The receiver also receives the packets and adjusts the window accordingly to expect packets 5,6,7,0 and 1 (5 more new packets). Now suppose that the acknowledgement for packet 0 is lost. The sender will send this packet again. The receiver, expecting packets 5,6,7,0 and 1, will accept packet 0 as part of the new set of expected packets. Since packet 0 was from the old set of packets, the protocol has failed.

There are also 2 ARQ mechanisms that can be used with the sliding window protocol. These are *go-back-n* and *selective repeat*. In go-back-n, when a transmission error occurs, the sender retransmits all the packets in the current window (including ones that were received correctly). In selective-repeat ARQ, the transmitter performs retransmission of only the packets that were damaged.

3.3 Review of Existing Multiple Access Protocols

Systems in which multiple users share a common channel in a way that can lead to conflicts are widely known as contention systems [20]. There are many different contention systems, and in order to fully understand and design an appropriate protocol for timed power line communication we will now explore some contention protocols. The underlying principle of all the contention (random access) protocols is quite similar.

3.3.1 ALOHA

For transmitting data, the simplest random-access technique using Automatic Repeat Request (ARQ) is the ALOHA contention protocol. ALOHA is the building block for many protocols. Whenever demand arises, each station transmits a short data packet on a common channel shared with other users. Some packets will experience collision with

others, and thus be corrupted or garbled. Collisions are detected, and the packet is retransmitted, perhaps after a random delay, with the process being repeated until successful. The destination detects errors by means of built-in error-detection coding. The destination then arranges to send acknowledgement packets (ACKS) back to the sender for each correctly received packet; where an ACK is not received after a suitable delay, the sender retransmits the relevant packet. This requires that the sender store each packet it sends until it receives and ACK for that packet. This is known as an ARQ (automatic repeat request) scheme [21].

Applicable concepts of ALOHA

Pure ALOHA is a good starting point for our protocol. Using built-in error detection and ACKs to ensure successful communication are useful concepts.

3.3.2 Slotted ALOHA

Pure ALOHA is somewhat inefficient, with a maximum channel utilization of 18.4 % [20]. An improvement to pure ALOHA was published in 1972 by Roberts that doubles the capacity of an ALOHA system to 36.8 % [20]. His proposal was to divide time up into discrete intervals, each interval corresponding to the length of one packet. This method became known as *slotted ALOHA* because it uses timeslots. In contrast to pure ALOHA, a terminal is not permitted to send whenever it wants. Instead it is required to wait for the beginning of the next timeslot [20].

Applicable concepts of Slotted ALOHA

Slotted ALOHA is even more applicable to our system because of the idea of using timeslots to avoid noise. One of the difficulties with implementing a slotted ALOHA system is the need for synchronization between stations. In the case of the power line network, it is easy to achieve this by synchronizing to and creating timeslots within the 60 Hz power line cycle since every node in the network is connected to the power line.

3.3.3 CSMA/CD and CSMA/CA

To further improve performance, many multi-access systems take advantage of the fact that a node can hear whether other nodes are transmitting. Instead of data being sent

whenever it is ready, a transmitter must first sense the transmissions of other stations. This is known as *carrier sensing* and incorporating this into a protocol is known as *Carrier Sense Multiple Access*. Multiple access refers to the fact that more than one node can obtain access to the link.

This *listen before transmit* function can be used to implement both persistent and nonpersistent CSMA protocols. In *persistent CSMA* if the channel is busy, the station waits until it becomes idle. When the station detects an idle channel, it transmits a packet. If a collision occurs, the station waits a random amount of time and starts all over again. In *nonpersistent CSMA* a conscious attempt is made to be less greedy. Before sending, a station senses the channel. If no one else is sending, the station begins doing so. However, if the channel is already in use, the station does not continually sense it for the purpose of seizing it immediately upon detecting the end of the previous transmission. Instead, it waits a random period of time and then repeats the algorithm. Depending on the amount of channel activity, this can lead to better channel utilization, but longer delays than persistent CSMA [20] [19].

Ethernet systems are based on CSMA/CD, Carrier Sense Multiple Access with *Collision Detect*. Collision Detect describes the ability of the station to detect that another station is transmitting at the same time and ceases transmitting immediately. It is nonpersistent – a node waits a random time after another transmission has completed to try resending its data [19].

Localtalk uses CSMA/CA, Carrier Sense Multiple Access with *Collision Avoidance*. Collision avoidance means that the protocol attempts to minimize the occurrence of collisions on the link. However, it does not sense collisions. Rather it uses Request to Send/Clear to Send (RTS/CTS) handshaking. When a station has data to send, it sends a request (RTS) and waits for a clear (CTS) and then begins sending. If a RTS is sent and no CTS comes back, this infers that a collision occurred. The idea is that the RTS and CTS are short and failed transmission due to a collision is quickly detected. Also, it

employs non-persistent CSMA by waiting a random amount of time before trying again [22].

Applicable concepts of CSMA

CSMA is a great concept, but perhaps too complex – it requires hardware for both carrier sensing and collision detecting, and there is the possibility that the carrier sensing can be fooled on a noisy medium. However, RTS/CTS handshaking is a good idea and a variation of this will be implemented in the protocol.

3.3.4 Reservation Slotted ALOHA

The idea of reservation slotted ALOHA is to have the protocol behave like normal ALOHA under low channel loads and move gradually over to some kind of time division multiplexing (TDM) as the channel load grows. There are several variants of this, one which was proposed by Crowther et al. in 1973 [20]. His idea is simple – whenever a transmission is successful, the station making the successful transmission is entitled to that slot in the next group as well. Thus as long as a station has data to send, it can continue doing so indefinitely. In essence it allows a dynamic mix of ALOHA and TDM, with the number of slots devoted to each node varying with demand. When a user is finished with a slot, it lies idle for one group after which time it can be picked up by another user. In the example below user A is using the second slot, but after 2 groups, no longer needs it. User D senses it is idle in the next group (group 3) and picks it up in group 4. Shaded timeslots indicate collisions where more than one station tries to use an idle timeslot [20].

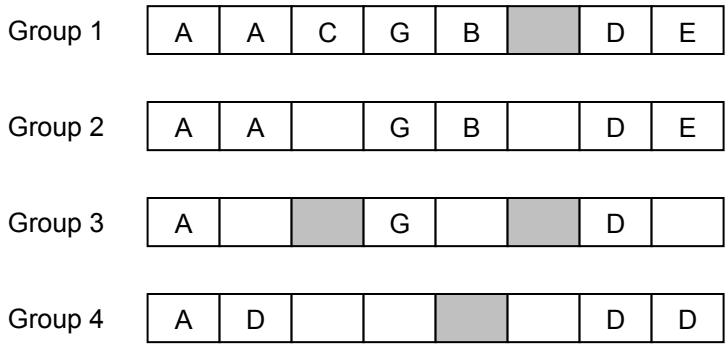


Figure 3.5 Reservation slotted ALOHA [20].

Another interesting addition to Reservation ALOHA proposed by Roberts [20] requires stations to make advance requests before transmitting. Each group of timeslots contains, say, one special slot which is divided into a number of smaller subslots used to make reservations. When a station wants to send data, it broadcasts a short request packet during one of the reservation subslots. If the reservation is successful (i.e., no collision), then the next regular slot (or slots) is reserved. Of course, all nodes must keep track of the number of slots that are reserved so to not interfere with others.

Applicable concepts of Reservation Slotted ALOHA

The idea of reserving slots is useful to the timed power line protocol. However, using a special reservation slot may be too difficult considering the power line architecture and the fact that any single slot's transmission characteristics can change drastically as the electrical loads change. However, by using RTS/CTS as we will soon discuss, the protocol effectively reserves slots for nodes.

3.4 Details of the Power Line Communication Protocol

The important variables discussed earlier are applicable to almost all protocol designs. However, because of the special medium and the synchronous nature of noise on the power line [1], a protocol must be designed that is able to detect when noise is occurring. Once it detects when noise occurs within the 60 Hz cycle, noise in future cycles can be

avoided because of its synchronous nature. It was also shown that noise times can change depending on the type of load connected to the line. Hence, once communication is established in times of low noise, the protocol must continuously or periodically test the line in order to detect when noise times change and modify communication times as necessary.

Another goal is to have the protocol as simple as possible. This power line protocol will most likely be implemented as an add-on part to various home appliances, HVAC, and other devices that require reliable, medium speed communication. Simplicity will result in lower cost devices and hence a more attractive solution for power line communication applications.

The review of existing protocols extracts several ideas that can be applied to the timed power line communication protocol. However, there are still many things that need to be defined.

3.4.1 Designing a Centralized or Distributed System

In deciding what type of system to design, several factors are considered. Designing a centralized system has certain advantages. One master has the power to designate which slave is allowed to talk and when, and it allows slave nodes to be simpler. Having a master will help take advantage of the fact that noise is synchronous to the power line cycle because it can detect this and designate communication times simply and efficiently. Slaves can make requests to the central controller to ask for channel access, and can only send data when the master sees fit and designates the channel accordingly.

However, there is one serious consideration when using a centralized architecture: not every node will have equal channel quality when communicating with the master. In fact, throughout a system of nodes connected to the power line, different pairs of nodes may have dramatically different communication capability than other pairs of nodes. This may be because of spatial location and the type of loads that are close to each node. This is shown in Figure 3.6. Node 2 is able to hear both nodes 1 and nodes 3. However,

node 1 and node 3 have a much longer distance between them, and may not be able to hear each other as well. Also, the lamp connected to the line may affect node 1 but have little effect on nodes 2 or 3. Similarly, nodes 2 and 3 may be affected by the computer power supply which does not affect node 1.

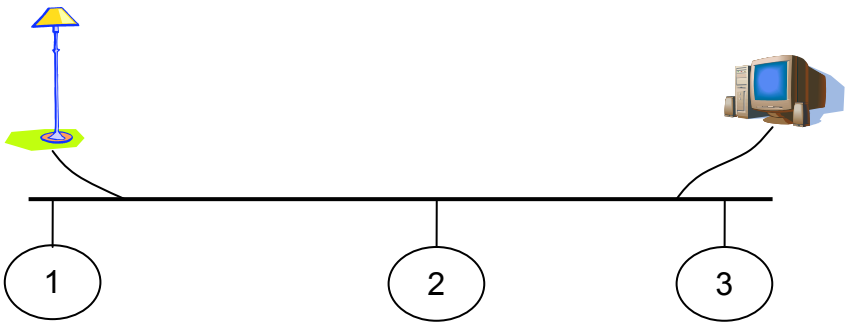


Figure 3.6 Nodes at different points on the line have different channel characteristics.

If we use a distributed architecture, each node will have equal communication logic. This has the advantage that the system will be easy to add nodes to and when a node is inactive it can simply "be quiet" or easily disconnected from the system without affecting other nodes. The downside to this type of system is the increase in complexity. Channel allocation becomes a much more difficult problem with no central controller to allocate it. Carrier sensing and collision detection/avoidance is necessary. Also every single node will have to keep a record for every other node in the system that keeps track of good timeslots for that node. This is because every pair of nodes can have much different channel characteristics than every other pair of nodes as shown above in Figure 3.6.

For this work, we have chosen a centralized architecture because it reduces complexity and the problem of deciding which timeslots to use becomes a much easier problem. Also channel allocation is simpler because there is no need for carrier sensing and collision detection. Therefore a typical network layout is illustrated in Figure 3.7.

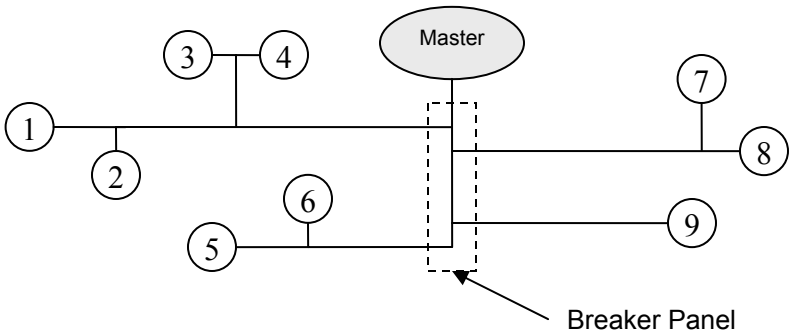


Figure 3.7 Typical layout of home or office network.

The best location for the master is at the breaker panel as this position gives the master the best location for being able to talk to and listen to all slaves.

The above figure brings up another important issue. The breaker panel presents a large attenuation between lines. For example, slaves 1-4 (on the same line) may have reliable communication, but none of them may be able to talk to slaves 5-9. This is because the signal power coming from any sender gets divided among all other lines connected to the breaker panel and also reflected back to the sender because of impedance mismatch. This problem can be solved by using an electrical repeater between lines or having the master relay all incoming messages out onto the other lines. However, the focus of this thesis is the design of a protocol, and the breaker panel problem will be left as future work.

3.4.2 Synchronizing to Zero Crossings and Using Timeslots

In the chosen method, the 60 Hz power line cycle is divided into timeslots. The power line cycle is approximately 16.67 ms long. This conveniently divides into 16 timeslots that are just over one millisecond long (1.04 ms to be exact). The remainder of this thesis will refer to timeslots as one millisecond. Each node must be synchronized to the 60 Hz zero crossings and keep track of timeslots itself. Each node contains timing information and knows that every one millisecond after the start of a power line cycle means the beginning of a new timeslot. This results in timeslots numbered 0 through 15 in one

power line cycle. Each power line cycle will bring on a new group of timeslots as shown in Figure 3.8.

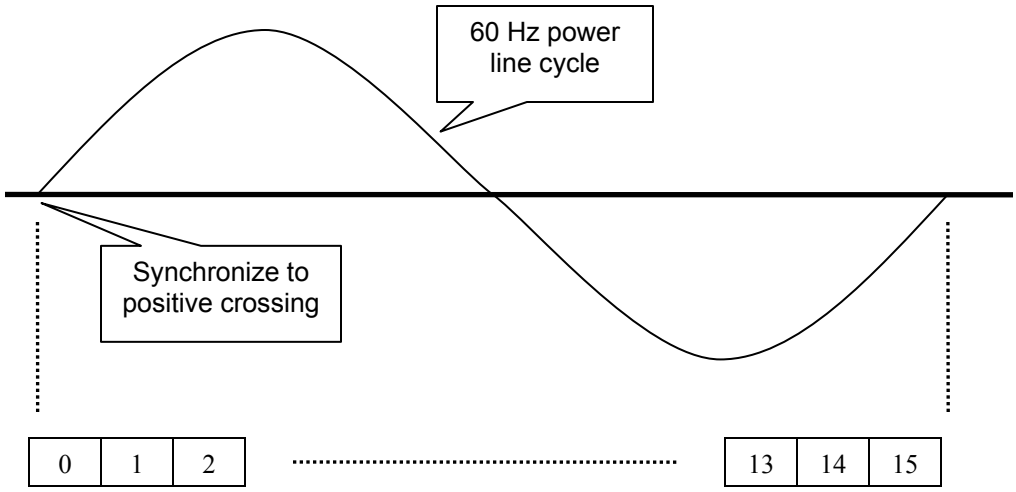


Figure 3.8 60 Hz power line cycle is divided into timeslots 0 through 15.

3.4.3 Timeslot Tables

Timeslot tables are another key to the protocol. Because the power line interference is synchronous, it will only cause certain timeslots to be unusable for data communication. Each node must keep track of this by updating an internal table. The table will contain a field for each timeslot that will indicate for each timeslot:

- GOOD (suitable for communication) - DEFAULT
- BAD (unsuitable for communication)

The timeslot tables will be updated based on the success/failure of received data and attempted transmissions over a period of time. When the system starts up, there is no knowledge of the channel, so the quality of timeslots must be checked. The master will broadcast special 'check' packets and each slave will check the quality of received packets over a period of time. If a node receives many errors in a particular timeslot it will flag this timeslot BAD. Next the master polls each slave, asking which timeslots it thinks are good and bad. The master then compares the tables of all the slave nodes and puts together a master table. This table only has entries GOOD and BAD for each

timeslot. GOOD timeslots in the master table are the *union* of GOOD timeslots between all slaves, otherwise timeslots are labeled BAD. This is shown in Figure 3.9 (with only 4 timeslots for simplicity).

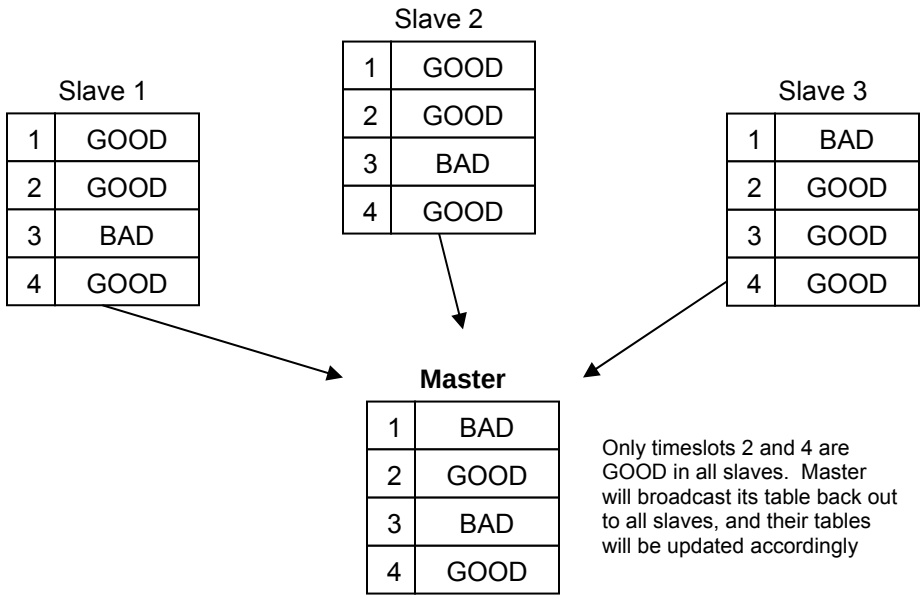


Figure 3.9 Master and slave timeslot tables.

Once the master has determined the master table, it will broadcast this to every slave. Since the timeslot table is extremely important information, it will broadcast it for one power line cycle in every timeslot. Slaves will check to make sure they get 2 of the same copies before they commit the master timeslot table to memory. The slaves then all have the same timeslot table, and will only attempt communication in those timeslots.

The Master does one more thing that is essential to timeslot coordination, and that is continuous checking and polling. In timeslots labeled BAD, the master will continue to broadcast check packets that will be checked for quality by each slave. The master will periodically poll each slave to see how well it is receiving the check packets, and if every slave responds that a BAD timeslot is now GOOD, then that timeslot will be updated to GOOD in the master table and rebroadcast to everyone. It is important to note that while one slave is talking to another in GOOD timeslots, they may determine that a GOOD

timeslot is actually BAD based on the number of errors. They will give this information to the master when they are polled again for their timeslot table.

The period of polling for timeslot tables and updating slaves with the master table is yet another variable in this protocol. Loads on the power line can and do change. Consider a computer power supply – once a computer is turned on, it usually stays on for quite a while (minutes to hours). Also, a home light dimmer position may change position causing a new noise spike but once it is in a position, it usually stays there for a lengthy period. For the purposes of this thesis, the period of polling for timeslot tables is chosen to be 10 seconds. This means that if a timeslot changes from GOOD to BAD or vice versa it may not be recognized for 10 seconds, but this is reasonable considering that it will probably not change again for a considerable period of time. However, it is *extremely important* to note that 10 seconds is an estimation by the author, and the optimum period between polling may in fact be much different.

3.4.4 Initiating Transmission

The master node controls all data transmission on the line. When a slave has data to send to the master or any other slave, it must first wait for the master to poll it. The master does this every 10 seconds as described above. Each slave's response will include information on timeslot quality (SQ) and a request to send (RTS) that indicates the number of 60 Hz cycles it would like channel access for. The master will then decide on how to allocate the channel to each of the requesting slaves. The master's decisions on channel allocation will then depend on the number of slaves requesting channel access and the amount of data each slave has to send. If there were too many cycles requested by the slaves, the master will not be able to satisfy each slave's request, and will scale back each slave's channel access equally.

Once it has decided on the division of channel access, it will send a Clear to Send (CTS) to a specific slave indicating to it that it has access to the channel for a certain number of 60 Hz cycles. The designated slave will send its data in GOOD timeslots within those

cycles after which time it will quit transmitting. The master will then send a CTS to another slave approving channel access for a certain number of cycles.

This process is a variation of Reservation Slotted ALOHA combined with Request to Send/Clear to Send (RTS/CTS) discussed earlier. It is actually more of a reservation protocol, because the slaves are effectively making channel reservation requests to the master.

Also note by making the master designate all channel access, this eliminates the need for difficult channel access methods like carrier sensing and collision detection. RTS/CTS is employed, but a slave is only allowed to make a RTS when the master asks it to.

3.4.5 Flow and Error Control

Once a slave is clear to transmit its data to another node in the system, it begins sending packets in the GOOD timeslots that it has been designated. Messages will be broken down into packets that are 1 timeslot long. Each message will have a sequence bit, which is a bit in the header that is the same for every packet in the message. The first data packet it sends will be the Data Beginning (DB) packet (this is the beginning of the message). In every GOOD timeslot after that, it will send Data Middle (DM) packets until the number of GOOD timeslots passed is equal to the message length and will finish the data message with the Data End (DE) packet. In the very next GOOD timeslot the receiver will send an ACK packet which acknowledges the entire message (packets from DB to DE). If an ACK was received, then the sequence bit is toggled and the next message is sent (if any) beginning again with DB.

If there were any errors in the transmission of the data packets, a NACK will be sent back to the transmitter. The transmitter will then send the entire message over again with the same sequence bit. The same action is also followed by the transmitter if no ACK or NACK is received in the very next GOOD timeslot. If the receiver actually sent an ACK and it was lost, it will receive a duplicate copy, which it will recognize because of the sequence bit, and simply discard one of the copies.

Notice that the ACK strategy is the *stop-and-wait* protocol which we discussed earlier. In this case the timeout period for receiving ACKS is simple – just one timeslot. The sender knows what timeslot to expect an ACK in and if one isn't received, then it times out. Using stop-and-wait helps achieve our goal of simplicity and has sufficient performance for our protocol.

3.4.6 Packet Format

A packet will be of a rather simple format, loosely based on an Ethernet packet [23]. Packets are a fixed length of one millisecond, as shown in Figure 3.10.

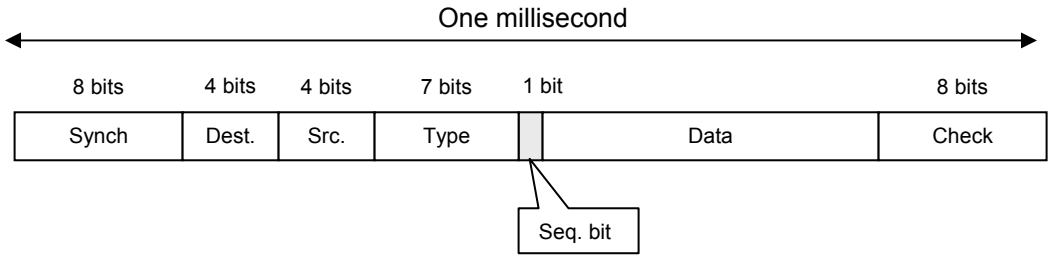


Figure 3.10 Packet format.

The packet fields are as follows:

- Synch. Byte (8 bits) – this is a simple pattern (01111110) that indicates the start of a packet
- Destination (4 bits) – the intended receiver identification number
- Source (4 bits) – the sender identification number
- Type (7 bits) – the packet type can consist of the following types:
  - Data packets (sent in one timeslot)
    - Data Beginning – first data packet.
    - Data Middle.
    - Data End – last data packet.
    - ACK – Acknowledgement packet. Sent from receiver to sender and acknowledges a data message (a group of packets with the same sequence number). Sent after a Data End packet is received
    - Check – packet is used strictly to check the quality of a timeslot.
  - Control Packets (NOTE: Each of these types must be sent multiple times within one power line cycle). The data field varies depending on the type of control packet.

        - PING – Master sends this to ask a slave if it has any updates. Slave has 1 whole power line cycle to respond, within which it may send requests for channel access and/or timeslot table updates. The data field is all zeros.
        - SQ – Slot Quality (slave to master). Slave responds after PING indicating to master which timeslots are GOOD/BAD. The data field contains which timeslots were GOOD/BAD. A bit is binary '1' if the timeslot is GOOD and '0' if a timeslot is BAD. The first bit is timeslot 0, and the last bit is timeslot 15.
        - RTS – Request to send data (slave to master). Slave responds after PING packet. The data field in this packet must contain the number of power line cycles that the slave wants channel access (access to all GOOD timeslots for a number of power line cycles)
        - CTS – Clear to send data (master to slave). The data field indicates to slave for how many timeslot groups (60 Hz) cycles it can have full channel access
- Sequence bit (1 bit) – used for determining retransmissions and duplicate packets at the receiver
- Data – length dictated by modulation technique and symbol rate. For example, at 100 kbps 100 bits can be fit into one timeslot. 32 bits are necessary for overhead, which means the data field is 68 bits.
- Error checking (8 bits) – 8-bit CRC used for detecting errors

Note that the total number of bits in a packet will be dictated by the bit rate selected for transmission. The number of header and error checking bits remains the same (32 bits or 4 bytes), and the number of data bits varies. For example, 56 kb/s would allow for a total of 56 bits in a packet, hence 24 bits or 3 bytes of data. 112 kb/s would allow for 80 bits or 10 bytes of data.

3.4.7 Master and Slave State Diagrams

The master and slave will always have to be in a defined state in order for this protocol to function properly. The following state diagrams provide order and clarity to many of the concepts discussed previously.

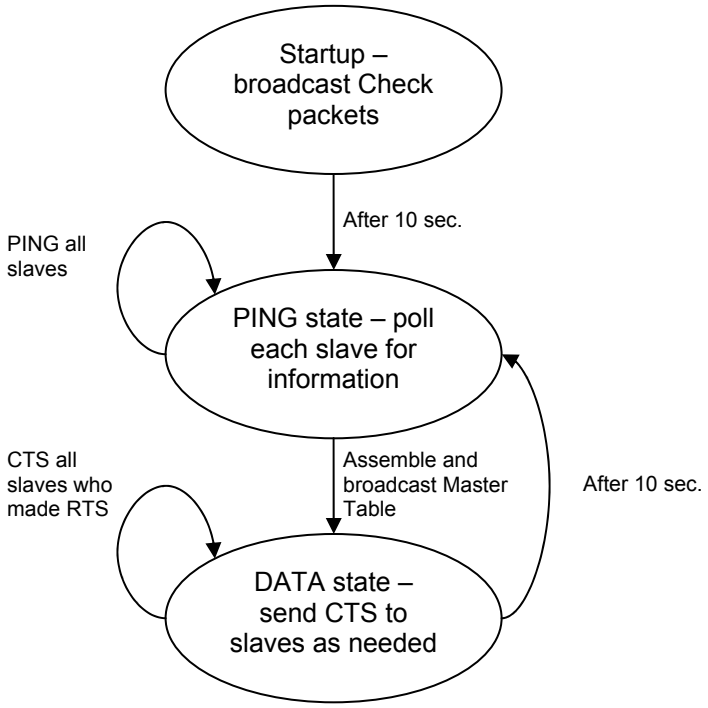


Figure 3.11 Master state diagram.

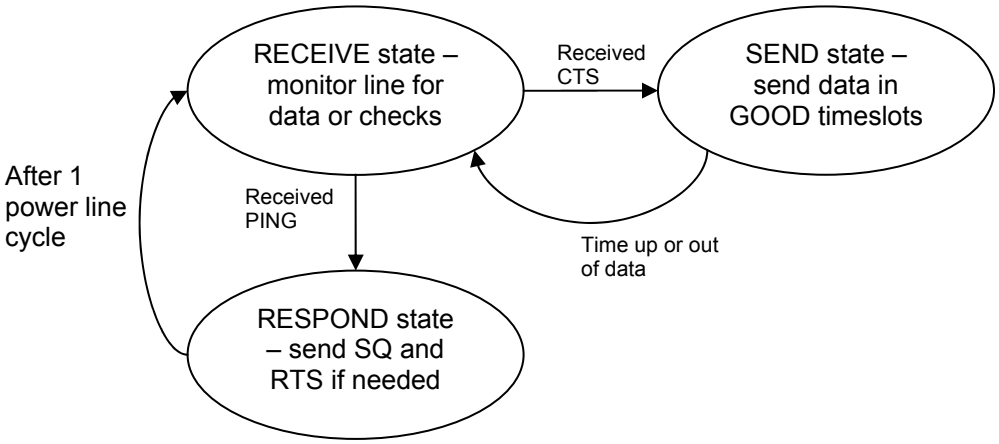


Figure 3.12 Slave state diagram.

3.4.8 Protocol Operation Example

Features of the protocol have been discussed up to this point, but it may be a bit overwhelming to put all of these facts together to determine how the protocol works. For that reason, an example follows, step by step, of a startup procedure followed by a data transmission. For this example, the master is just starting up, and no nodes in the system have any knowledge of the channel.

1) The master broadcasts check packets to all slaves in every timeslot for 600 60 Hz cycles. The number of power line cycles should be sufficient to get an accurate representation of GOOD/BAD timeslots (this will be a 10 second test).
2) The master will then send out a PING to the first slave (starting with slave number 0), in every timeslot for one power line cycle and wait for a response from the slave for the next entire power line cycle.
3) If the slave exists, it will respond throughout the next cycle with SQ (slot quality) packets and RTS (request to send) packets. SQ packets will be sent in the first half of the GOOD timeslots, and RTS in the second half. If it has no data to send, the data field in the RTS packet will be 0.
4) The master will try all 16 destination numbers, and if there is no node for a sequence number there will simply be no response. The master will also remember which nodes are present in the system based on whether there was a response or not. If there was no response from a slave when it was actually present but could not be heard, this will be taken as a slave that does not exist.
5) The master will then create its master table indicating all common GOOD and BAD timeslots for the system and broadcast it in every timeslot for one power line cycle.
6) The master then sends a CTS in every timeslot for one power line cycle to the first slave indicating in the data field how many power line cycles it is allowed to have access to the channel for (depending on the number of cycles the slave requested).
    a. If a slave's RTS was not heard by the master, it will not be included in the channel allocation.
    b. If a slave does not hear a CTS, it will miss its opportunity to transmit and the channel will be left idle for this period. Also, if by error a slave hears a valid CTS for two slaves (including itself) it will *not consider the CTS's valid*. This will avoid collisions in the case that an error causes a valid CTS to two different slaves. The slave will then have to request channel access again when it is polled.
    c. If the master received too many RTS's, that is, it cannot satisfy the channel demand from everyone, it will only grant each slave what it is able to. It will reduce the number of power line cycles each slave has access to, with no preference given to any. Slaves will realize this and have to make another RTS in the next cycle. If this continues, the

slaves will have to reduce the amount of time they request access to the channel.

7) The slave will use all of the GOOD timeslots for the number of power line cycles it has been allowed to send its data.
    a. The slave's first timeslot transmission will be the Data Start packet, and the destination node will receive this and watch for Data Middle packets, and finally the Data End packet
    b. After the receiver verifies the Data End packet, it will send an ACK packet to indicate if it received the entire message, or a NACK if there was an error
    c. If the message was not received correctly (a NACK came back, or no ACK was received), the transmitter will send it again with the same sequence bit. If the ACK was actually sent by the receiver and it was lost, it will recognize the duplicate packet because of the sequence bit
    d. When an ACK is received, the sender will begin with another Data Start packet, and so on until it has no more data to send or it is out of its allotted time for channel access
    e. When a sender is finished transmitting all its data or it runs out of the power line cycles allocated to it, it simply quits transmitting
8) Notice that senders are only sending in GOOD timeslots. The master, however, broadcasts 'check' packets in BAD timeslots that the sender is skipping so that the quality of timeslots can be monitored in case of any change.
9) Every 10 seconds, the master will repeat its PING cycle asking for updated timeslot tables and data requests from each slave, assembling the master table, and rebroadcasting it to all slaves, and finally sending CTS to slaves who requested channel access.

The entire process is illustrated in Figure 3.13.

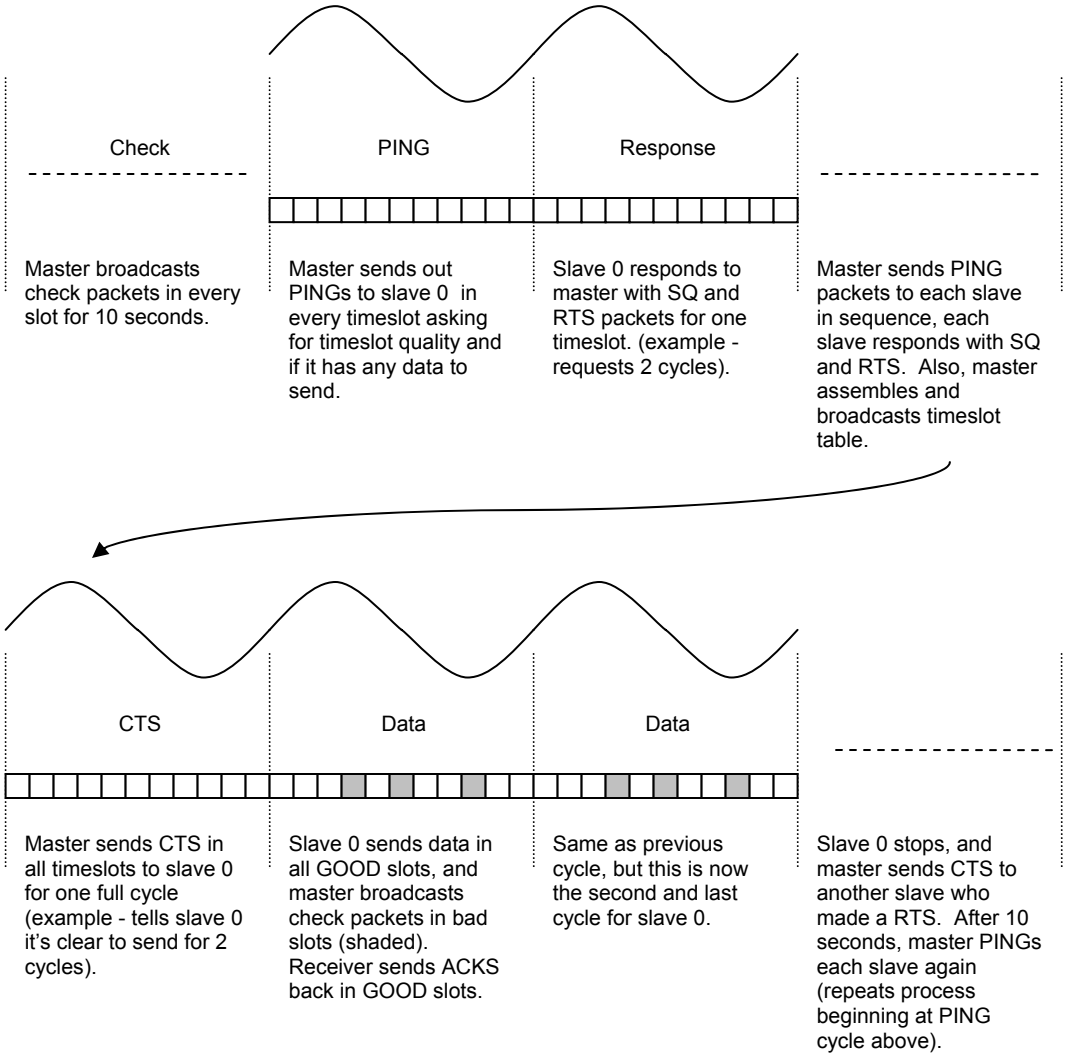


Figure 3.13 Example of protocol operation.

4. Half Duplex Modem Design

4.1 Bandwidth Considerations

Bandwidth for an ideal transmission system is defined as the range of frequencies passed by the system in an undistorted manner. Similarly, the bandwidth of a signal is defined as the range of frequencies contained in the signal. A proper match between signal and transmission system results if the passband of the system extends over the complete range of frequencies occupied by the signal [19]. In the case of power line transmission, there are governing bodies that place restrictions on the signal bandwidth. In the United States this governing body is the Federal Communications Commission (FCC) and in Canada there is Industry Canada.

The FCC states that frequencies 9 - 490 kHz are allowed for signaling on the power line [24]. It is also stated that transmitting on the power line creates an unintentional radiator and that this must not cause harmful interference. The system must also be able to accept other interference that may be caused by other radiators such as an authorized radio station, by industrial, scientific and medical equipment, or by another intentional or unintentional radiators [24].

Industry Canada provides guidelines for transmission on the power line in the bandwidth 0 to 535 kHz including the carrier voltage limits for installation in residential and office buildings as shown below [25].

Below 9 kHz:	No limits
9 - 95 kHz:	15.0 volts pk-pk
95 - 105 kHz:	Restricted – Loran C time signal frequency
105 - 185 kHz:	15.0 volts pk-pk
185 - 535 kHz:	$.45\ (B/D)^{1/2}$ volts pk-pk or 15.0 volts pk-pk, whichever is the lesser voltage. B = 6 dB bandwidth in kHz. D = duty cycle (D = 1.0 for continuous transmission)

This work will adhere to the bandwidth of 105 kHz to 490 kHz which will satisfy restrictions placed by both the FCC and Industry Canada.

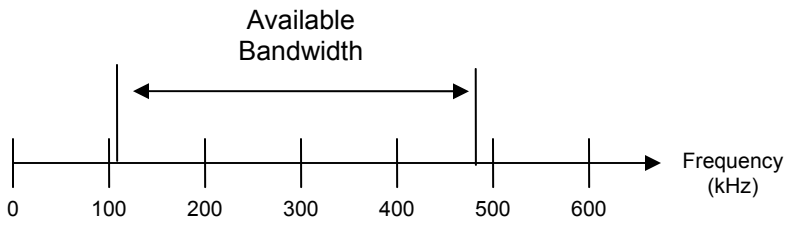


Figure 4.1 Available spectrum for power line communication.

4.2 Modulation

Basic digital signals have a low-pass characteristic in that the frequency content of the signals extend from 0 Hz to some maximum value. These are referred to as baseband signals [19].

Transmission systems have some available bandwidth, and this seldom includes the lower frequency range of the baseband signal. Hence transmission systems must use some form of modulation to shift the frequency content of a signal from one region of the spectrum to another. The simplest form, amplitude modulation, involves multiplying a signal by a sinusoidal carrier signal. This moves the baseband signal to a frequency range centered about the frequency of the carrier and the occupied bandwidth is doubled. Demodulation is performed at the receiver for shifting the received modulated signal frequencies back to baseband [19].

4.2.1 Choice of Modulation Scheme

There are several forms of modulation available. Some of the better known ones are amplitude shift keying (ASK), frequency shift keying (FSK), phase shift keying (PSK), and spread spectrum as discussed earlier in this thesis. The passband modulation scheme chosen for this thesis is frequency shift keying (FSK).

Frequency shift keying has two different frequencies, one for mark (binary 1) and one for space (binary 0). This is the same thing as modulating an FM carrier with a binary digital signal. Frequency shift keying is a popular choice for modulating data signals on the power line because it is easy to generate and demodulate. However, it is somewhat spectrally inefficient as compared to other methods so this limits data rates. Since the focus of this thesis is the communication protocol, optimizing data rate is not considered, and frequency shift keying (FSK) was selected because of its ease of implementation.

The spectrum of an FSK signal is shown below. Its bandwidth is determined by two factors: the mark and space frequencies chosen and the symbol rate [19].

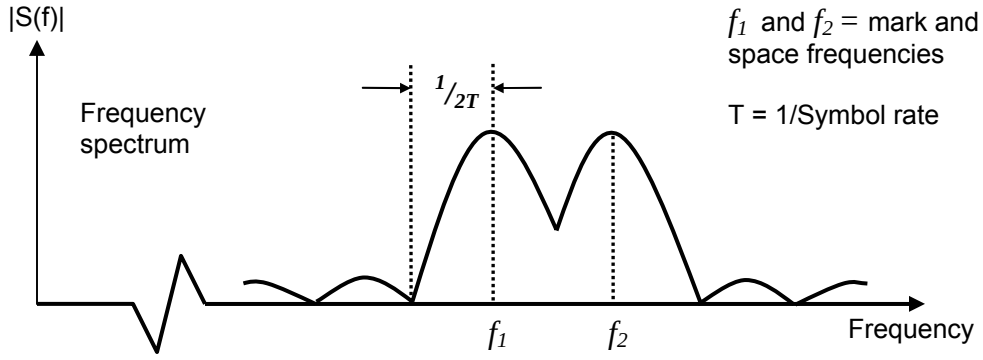


Figure 4.2 FSK Spectrum.

This work will use mark and space frequencies of 150 kHz and 250 kHz respectively, and a symbol rate of 60 ksymbols/sec. From Figure 4.2, it can be concluded that the

bandwidth occupied by this signal is from 120 kHz to 280 kHz, for a total bandwidth of 160 kHz.

4.3 Universal Asynchronous Receive/Transmit (UART)

Universal Asynchronous Receive Transmit (UART) is a device that uses start-stop asynchronous format. This format is typically character based with a start bit, an 8-bit character, and one or 2 stop bits for a total of 10 or 11 bits. It is often used for applications that need a simple method of communicating low-speed data between equipment. It is a form of serial communication, meaning that one bit is transmitted at a time. The term asynchronous is used because it is not necessary to send clocking information with the data that is sent. Typical applications of asynchronous links are connections between terminals and computer equipment [26].

The protocol is character oriented, and transmission is carried out without continuous character synchronization between the transmit and receive devices. In this mode each character or "chunk" of data has to be identified separately, and hence the beginning and the end of each character must be marked. These are the 'start' and 'stop' bits respectively.

Instead of having transmit and receive clocks connected, the receiver synchronizes itself based on the start bit. The receiver oversamples the incoming data stream, usually by a factor of 16, (16x but can also be 8x or 32x) and uses some of these samples to determine the bit value. The oversampling clock is synchronized by the leading edge of the start bit. If 16x oversampling is used, every 16 samples from the oversampling clock will represent a data bit. Traditionally the middle three samples of the 16 samples are used. Two UARTs can communicate like this if parameters such as bit rate and character length are the same for both transmitter and receiver.

When a complete byte is received, a flag is set to indicate that a data byte is ready to be read. This can cause an interrupt if the device is programmed to do so. Similarly, when a complete byte is transmitted this causes an event which can cause an interrupt indicating

that another byte can be transmitted. When data bits are not transmitted in the UART protocol, a continuous stream of ones is transmitted, called the idle condition. Since the start bit is always zero, the receiver can detect when data is once again present on the line.

UART characters were chosen for testing the power line protocol because of ease of implementation. Communication can occur over a single wire, and clocking is taken care of by the transmit and receive oversampling clocks so no clock transmission or clock recovery needs to be implemented. If the maximum character size and one start and one stop bit is used, the overhead involved with using UART characters is 2/10 (or 20%), but several characters can still be fit into one timeslot and actual operation of the power line protocol can be studied which is the focus of this thesis. The UART transmission format is shown in Figure 4.3.

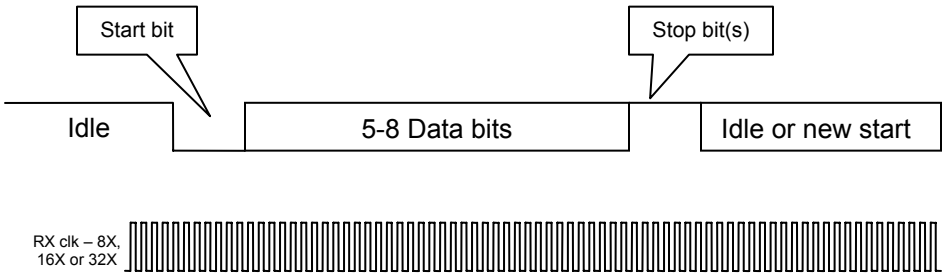


Figure 4.3 UART character format.

4.4 Hardware Implementation

Figure 4.4 shows the half-duplex modem block diagrams, and in the following subsections are descriptions of each system block. Each modem is half-duplex because it has the ability to both talk and receive, but not at the same time (which would make them full-duplex).

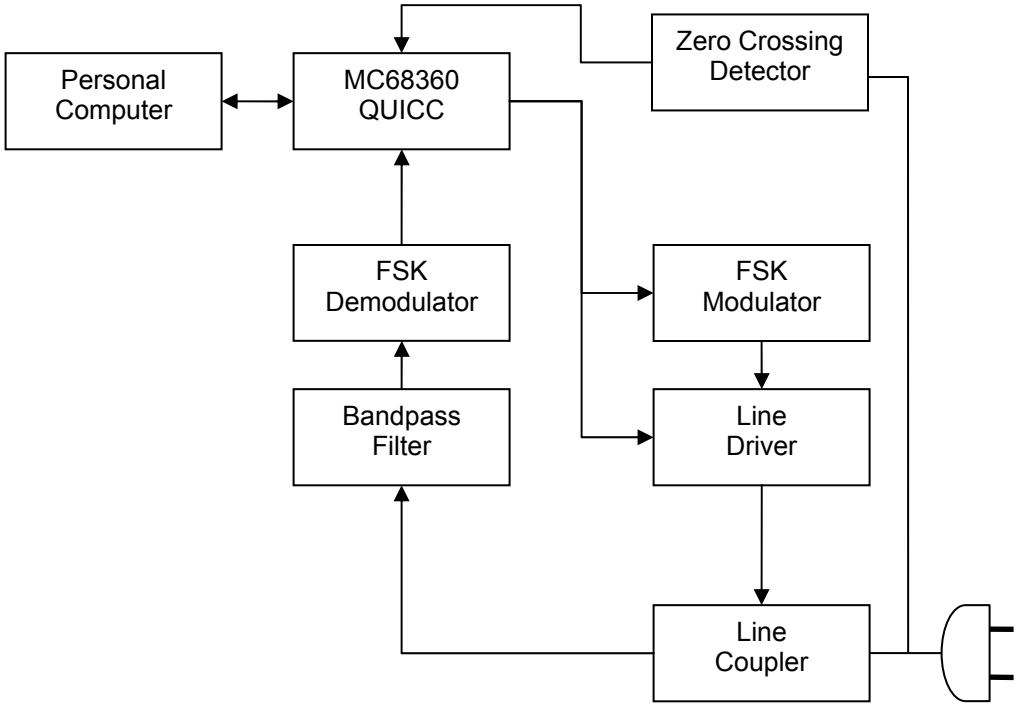


Figure 4.4 System block diagram for both master and slave.

4.4.1 Personal Computer

The PC is needed for loading, starting and stopping execution of programs on the MC68360 microprocessor. This computer runs Slackware Linux [27] and is loaded with libraries specific to the MC68360. Program loading is done through the computer's parallel port.

4.4.2 Zero Crossing Detector

The zero crossing trigger is simply a power line "clipper". It uses an LM311 voltage comparator to monitor its input (from the power line) and if the voltage is positive, it outputs +5 V. If the input voltage is negative, the zero crossing trigger outputs 0 V. Obviously when considering the 60 Hz power line cycle as an input, the zero crossing trigger will output a 5 Vo-p 60 Hz square wave. This trigger as designed may cause 'bounce' at the zero crossings (due to noise). To avoid this, the microcontroller was programmed to ignore multiple edges within one millisecond of the initial edge.

4.4.3 MC68360 QUICC

The MC68360 is a Motorola microprocessor designed for communications applications. QUICC stands for QUad Integrated Communications Controller because there are four serial communication controllers on the device [26]. For this application however, only one is needed. The controller has an operating system on it called RTEMS (Real Time Executive for Multiprocessor Systems) [28]. RTEMS provides a means to send events between processes and helps processes and events occur within a specific time interval (hence the name real time).

The QUICC is the control center for the system. It contains the program implementation of the protocol designed in Chapter 3, and once the program is loaded into the QUICC by the personal computer, it begins execution of the program immediately. The QUICC uses the 60 Hz, 5 Vo-p square wave input from the zero crossing trigger. An edge-triggered interrupt is used to synchronize transmission and reception to the zero crossings of the power line cycle. Once it receives the zero crossing interrupt, it uses an internal timer that interrupts every one ms. This timer interrupt is used to count out timeslots 0 through 15.

The transmit and receive lines are connected to the first of the four QUICC's serial communication controllers (SCCs). There are several modes that must be set for the for SCC to operate properly and of particular importance are the transmit/receive data format and baud rate. Several communication protocols are available on the chip and, as described earlier, the chosen method was to use the Universal Asynchronous Receive/Transmit (UART). The UART was programmed to transmit and receive data one byte at a time, and receive interrupts were programmed to indicate whenever a byte of data is received. After a byte is received and the interrupt is triggered, a routine executes that adds the byte to the received data and handles it depending on its type.

4.4.4 FSK Modulator

The FSK modulator is an Agilent 33120A 15 MHz Function/Arbitrary Waveform Generator. Using the FSK mode on the 33120A is convenient because it allows fully

programmable mark and space frequencies and signal amplitude [29]. The 33120A has a 50 Ω output impedance and does not have the current output capable of driving the line, which is why a line driver is needed.

4.4.5 Line Driver

The line driver is more complicated than one might first think. Instead of just having a buffer that drives any signal on its input onto the power line, there has to be some method of disconnecting the buffer from the line when there is no data being sent. This is because UART characters are transmitted onto the line, and when there is no data being sent a simple buffer would constantly drive the '1' frequency onto the line (idle state). If some other device is trying to drive UART characters onto the line, this device will not be able to hear them because its own line driver is driving the idle state. The most obvious solution would be to used a tri-state buffer, but no part could be found at reasonable cost that had the output current necessary to drive a signal onto the power line.

This problem is solved by using a resistor between the buffer output and the line and a switch which only connects the buffer to the FSK source when there is data to be sent and connects it to ground otherwise as shown in Figure 4.5. An input from the MC68360 tells the switch when data is to being sent or not (and hence whether to connect the buffer input to the FSK source or ground). When the buffer input is ground, it drives its output to 0 V and the resistor acts as a termination for any signal that is coming in on the line. When the buffer input is connected to the FSK source, the buffer's output must drive the FSK signal onto the line through the resistor.

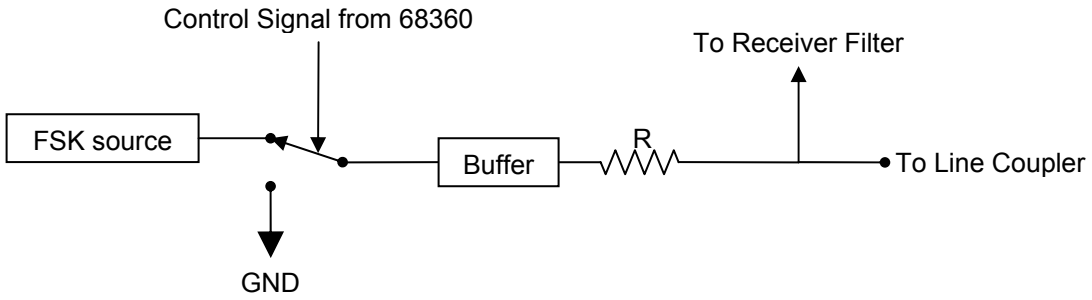


Figure 4.5 Line driver.

The choice of resistor is important in order to maximize power transfer onto the line and maintain the highest possible amplitude when terminating incoming signals. A 10 ohm resistor was chosen for this implementation because it is a reasonable approximation to the line impedance for the frequency range that we are interested in (105 – 490 kHz).

4.4.6 Bandpass Filter

A bandpass filter is needed because of the broad spectrum and high amplitude of noise on the line. The filter passes the frequency range of communication (determined by FSK mark and space frequencies and the data rate) and rejects all other frequencies. This filter was implemented in the same way as in Hanson's research [1].

4.4.7 Demodulator

Once the line signal is filtered and amplified, it is demodulated using an FSK demodulator chip (XR2211). It has a programmable center frequency f_o and if it detects a signal above or below this frequency it outputs +5 V or 0 V respectively. The demodulator implementation is described in the XR2211 data sheet (in the appendix), and can be seen in Hanson's research [1].

4.4.8 Line Coupler

The line coupler is necessary to interface the communication electronics to the power line and also to isolate them from the high voltage levels found on the power line. The line coupler is a simple device consisting of a resistor, capacitor and transformer. Proper selection of the resistor and capacitor ensures that the 60 Hz mains signal is greatly attenuated so it doesn't affect the communication electronics, yet higher frequencies pass through with little attenuation.

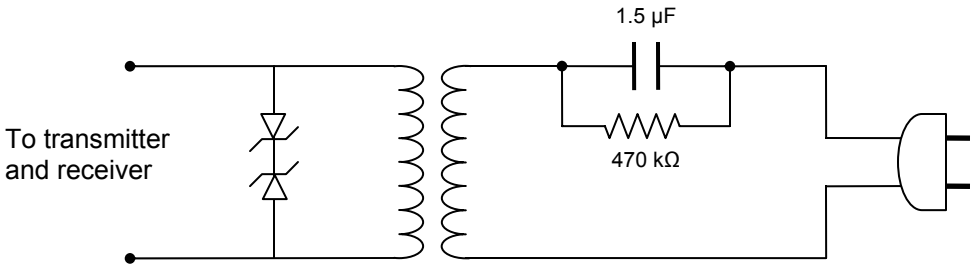


Figure 4.6 Line coupler.

4.5 Software Implementation

For the purposes of testing in this thesis, only the master and one slave were programmed. The 68360 microprocessors were programmed using the UART, a timer, and an external edge-triggered interrupt to implement the function of the protocol. Beyond programming these device-specific features, a model of the protocol for both the master and slave was programmed.

Once the 68360 is loaded, it begins execution immediately. Upon receiving an edge-triggered interrupt from the zero crossing detector, it resets a timeslot number index to 0 and also resets an internal timer. The timer counts to one millisecond and then causes an interrupt that is used to increment the timeslot number index. The timer does not stop upon interrupting however; it resets immediately and counts to one millisecond again, repeating its interrupt. Once the timeslot number index reaches 15, it stops the timer and waits for the next edge trigger from the zero crossing trigger. This process repeats as long as the zero crossing trigger continues.

After the master and slave have established a common timeslot table (as described in Chapter 3), the master enters a ‘data’ mode where it sends variable length data messages to the slave, only in GOOD timeslots. In BAD timeslots, the master sends ‘check’ packets so quality can be continuously monitored (these are interleaved with slave transmissions). The range of message lengths that it sends to the slave is programmable

(must be written into the actual programs, then compiled and loaded into the microcontrollers), by specifying *message_length_range_beginning* and *message_length_range_end*. The length of the first message will be *message_length_range_beginning*, and the next message sent will be *message_length_range_beginning* + 1, and so on. After each message is sent, the master expects an ACK in the next timeslot after the end of the message as described in the protocol section in Chapter 3. Once the message length reaches *message_length_range_end*, one 'message length sweep' has been completed and the master starts over sending a message of length *message_length_range_begin*. The master sweeps through the range of message lengths for a number of times that is specified by the programmable constant *num_sweeps.* Once it has completed the message length sweep *num_sweeps* number of times, it only sends 'check' packets in every timeslot. After a total of 10 seconds since the beginning of the cycle, the master sends another 'ping' to the slave and the slave responds with the quality of timeslots. The master then repeats this entire cycle.

The slave checks messages and stores the number of each message length it received without error. The slave also keeps a count of the number of CRC errors in each timeslot so that it can continuously identify GOOD and BAD timeslots. This is possible because even though a message may span multiple timeslots, each timeslot has only one packet (which contains a CRC). The master also counts and stores the number of ACKS it received for each message length in the message length range. The process including startup is shown in Figure 4.7.

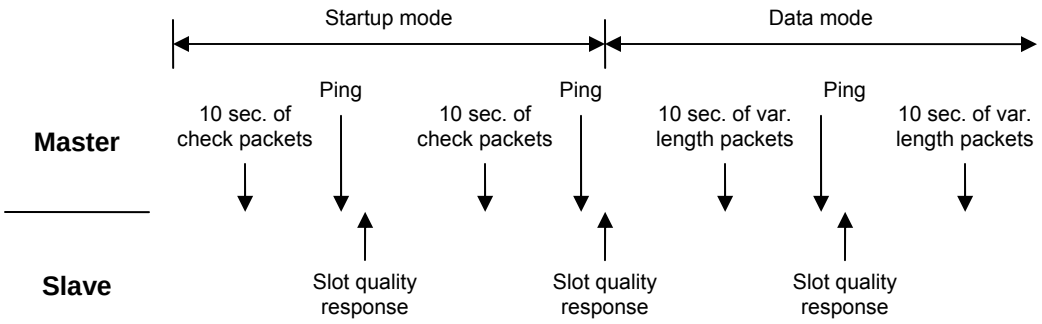


Figure 4.7 Master/Slave Communication Process.

In order to monitor the quality of reception by the slave on an oscilloscope, at the beginning of each timeslot the slave sets an output pin high if it correctly received a packet in that timeslot in the last power line cycle. Also, both the master and slave have output pins that are set high at the beginning of a timeslot that is GOOD in the timeslot table and set low if the timeslot is BAD in the timeslot table. This pin's output should be exactly the same on the master and slave when viewed on an oscilloscope once the master and slave timeslot tables have been coordinated.

Also for reporting purposes, the KERMIT protocol [30] was used. This is a program that facilitates communication between computers using the serial port. In this case, it was used to connect the 68360 to a PC. Using this interface, every time that a 10 second period is completed, both the master and slave send text reports which are printed on the screens of their respective PCs. The slave reports the number of power line cycles that data was received in (which is 600 for 10 seconds), the number of errors it received in each timeslot, and the number of times it received each message length without error. The master reports the number of power line cycles it sent data in (600), and the number of ACKS it received for each message length.

For the CRC-8 checkword generation and verification, the program used a shift register implementation which involves shifting and exclusive-OR operations. A generator polynomial is necessary, which is 0x07 for CRC-8. The generator polynomial is common to both the transmitter and receiver and is necessary for CRC generation and

decoding. A 2-input exclusive-OR gate is placed in between bit positions in the shift register that correspond to binary '1' in the generator polynomial. The inputs to this exclusive-OR gate are 1) the previous bit and 2) the output of the shift register (which is fed back). The output of the exclusive-OR is fed into the next bit position. In CRC generation, an 8-bit shift register is cleared and the entire bit string is shifted in. Eight zeros follow the data bits and the resulting content of the shift register is the desired checksum. This is appended to and transmitted with the data.

When receiving, a similar process is carried out. Each byte received is shifted into an 8-bit register and the output of the shift register is exclusive-ORed with the bits at the positions in the shift register corresponding to the bit positions that are 1 in the generator polynomial (same as above). After the last received byte (the checksum) is shifted into the register, the contents will be all 0 if there were no transmission errors. If any bits in the shift register are not 0, a transmission error has occurred. Figure 4.8 illustrates the CRC shift register implementation.

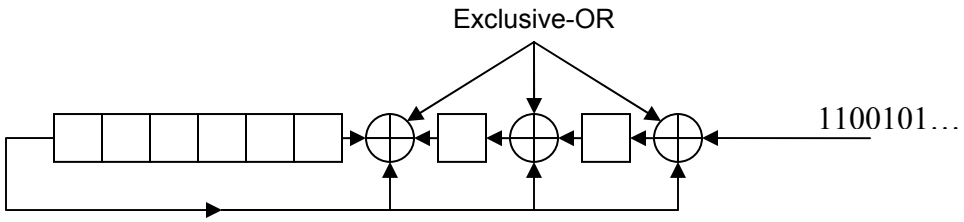


Figure 4.8 CRC shift register implementation (transmitter and receiver).

5. Protocol Performance

This chapter presents and discusses the performance of the power line protocol described in Chapter 3 using the implementation described in Chapter 4. All tests were performed at the offices of Telecommunications Research Laboratories (TRLabs) in Saskatoon, Saskatchewan. Two desks were used, referred to as desk 1 and desk 2 in each test; each desk connected to the power line and had multiple bench outlets. The distance from the bench outlets to the wall socket (which each desk plugged into) was approximately 3 meters. In addition, extension cords were often used to increase distance between the master and slave. Each test shown in this chapter was conducted over a period varying from 5 to 20 minutes and with master and slave in different locations, depending on the type of test. These short term tests were considered sufficient to provide meaningful performance measures for the timed power line communication protocol.

For all tests, the same mark and space frequencies were used (150 and 250 kHz respectively) with a symbol rate of 60 ksymbols/sec. This signal occupies a bandwidth of 180 kHz. The allowed transmitting amplitude (from Chapter 4.1) is then .45(B/D) = $.45(180/1.0)^{1/2}$ = 6.04 Vp-p. For all of the tests in this chapter, a transmitting amplitude of 6.0 Vp-p was used *unless otherwise noted*.

5.1 Data Transmission Performance

Figure 5.1 shows the 60 Hz mains voltage and the received noise signal (after the line coupler). One can notice the 60 Hz mains voltage is a sinusoid, but has peaks that are somewhat flattened. This is because of power supplies loading the 60 Hz mains voltage at its peaks. Once the 60 Hz mains voltage begins to decrease, the power supply diodes stop conducting, and the flattening effect disappears until the next peak. When looking at

the received noise trace, it is obvious that the highest level of noise is caused by the switch-off of the current to the power supply.

Also, notice the information available on the oscilloscope display. Each trace is numbered on the left hand side, and in this case there is trace 1 and 4. The scale is shown in the lower left corner. Channel 1 has a scale of 100 V/division and channel 4 is using 500 mV/division (these are vertical divisions). The time per division (horizontal divisions) is also seen in the lower middle, and is 2.00 ms/division. Other information available on the oscilloscope is triggering information (how the display is synchronizing to the signals). In the lower middle and right of the display, the oscilloscope is showing that its trigger source is 'Line' at 1.10 V, and the display is shifted to 4.18 ms *after* this trigger point (indicated by the arrow pointing to the right just before 4.18 ms).

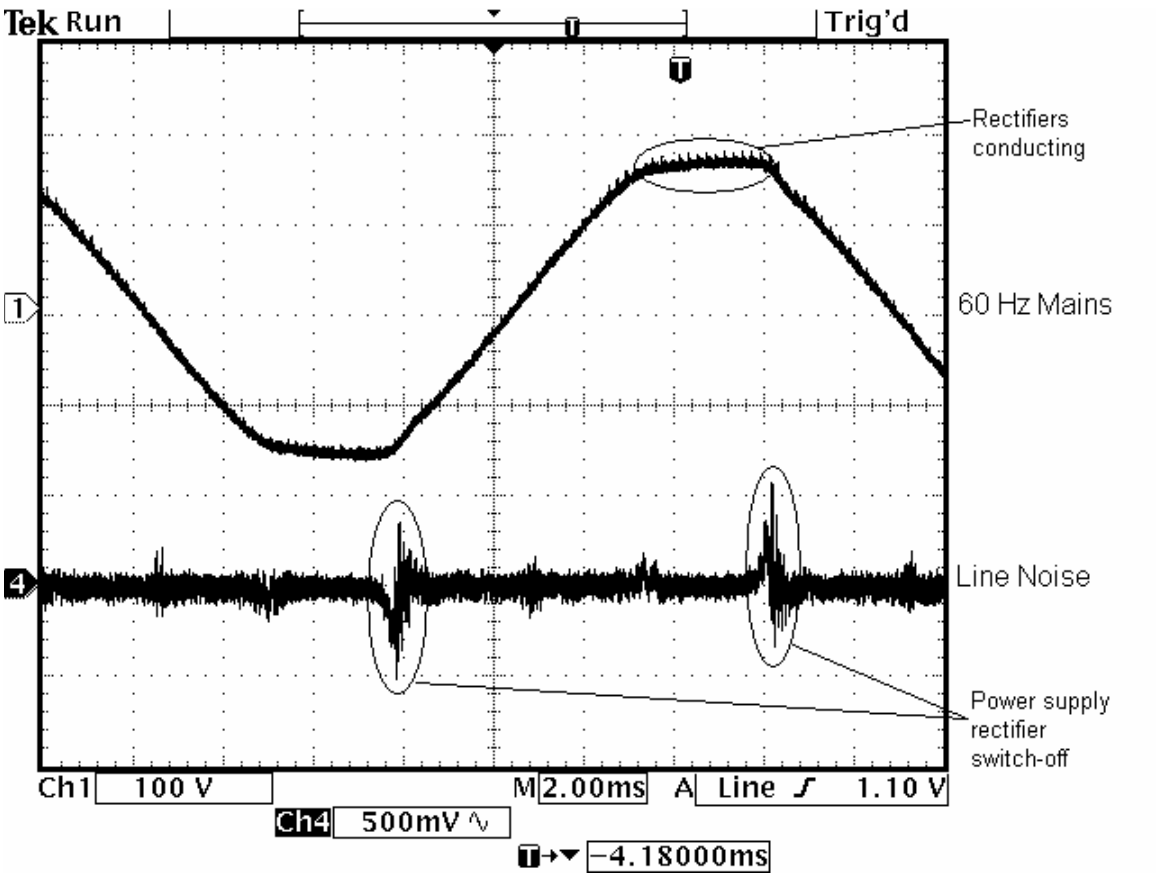


Figure 5.1 AC mains voltage and power line noise.

For the first data transmission test, check characters were transmitted continuously and checked at the receiver using the microprocessor setup described in Chapter 4. Recall that this system was designed using UART characters for transmission, so this is the smallest data entity that can be checked for errors. If a character was received without error, a logic '0' was produced on an output pin of the microprocessor. If the character had any errors, a logic '1' was produced on the same output pin. Also, the 60 Hz mains voltage was divided into one millisecond timeslots and checked for errors. Similarly when checking timeslot errors, if any byte within a one millisecond timeslot was in error, a logic '1' was produced on an output pin of the microprocessor for the entire timeslot in the *next* power line cycle, and if no errors occurred a logic '0' was produced.

Figure 5.2 shows the results of this test, performed with the transmitter on desk 1 and the receiver on desk 2 (separation of 3 meters) with a 3 meter extension which provided a total separation of approximately 6 meters. The test was conducted during business hours on a weekday. A Tektronix oscilloscope was set up in average mode in order to observe the byte and timeslot error characteristic. The average mode on the oscilloscope takes the running average of 4, 8, 16, 32, 64, 128, 256 or 512 consecutive traces and displays this average. For this test, an average of 256 was used. The traces are (in order from top to bottom): 60 Hz mains voltage, received signal and noise, byte errors, and timeslot errors. As expected, the byte and timeslot errors occur at the same time position as the high level of noise produced by the power supplies.

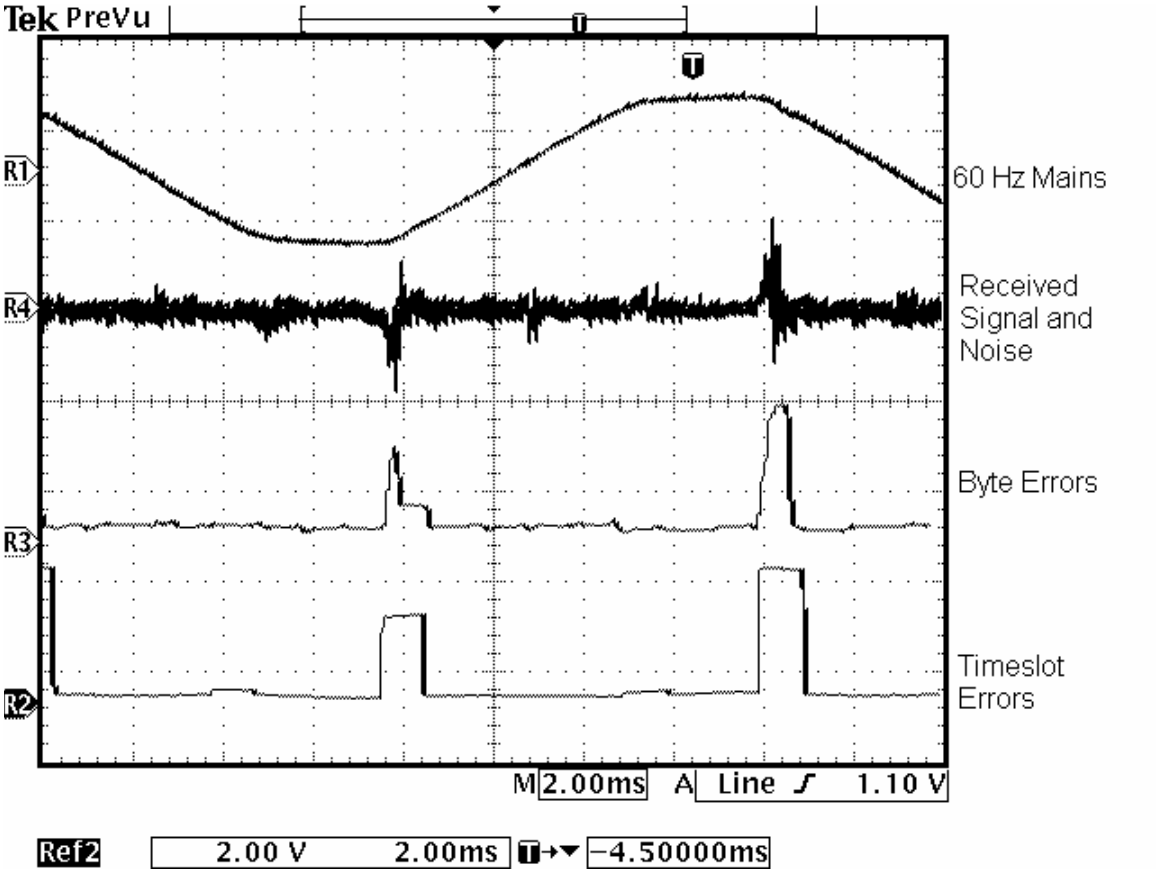


Figure 5.2 Byte and timeslot errors.

For the same test setup, the number of errors in each timeslot and timeslot error percentage for each timeslot was recorded and displayed by the microprocessor. Timeslot error percentage was used here rather than bit error percentage because the program was set up to count timeslot errors, not bit errors. A single timeslot error means that at least one bit in the timeslot was incorrect. Results are shown in Table 5.1. Figure 5.3 is a graphical representation of this data. Note the *relative* error ratios – timeslots 5 and 13 have many times the errors of all other timeslots.

Recall that an 8-bit CRC was used to detect errors in the packets. Even though CRC has excellent error detecting capability, there are still some errors that the CRC will not detect. The error detecting ability of the CRC provides detection of all burst errors of 8 bits or less, and all error patterns of an odd number of bits [18]. The error patterns that are *not* detected are multiples of the generator polynomial – in this case 100000111. In

fact, an error burst of greater than 8 bits is detected with probability $1-0.5^8 = 0.996$ [18]. This is a small number of errors, and not necessary to detect because the only concern of the protocol is detecting an error ratio and comparing it to a threshold. The small number of errors missed will have a negligible effect on whether the error rate reaches the threshold or not.

Table 5.1 Microprocessor report of timeslot errors.

Total number of cycles 60712		
Timeslot	# Errors	Error %
0	260	0.43%
1	257	0.42%
2	231	0.38%
3	482	0.79%
4	486	0.80%
5	13383	22.04%
6	130	0.21%
7	149	0.25%
8	188	0.31%
9	525	0.86%
10	423	0.70%
11	345	0.57%
12	126	0.21%
13	14156	23.32%
14	217	0.36%
15	147	0.24%

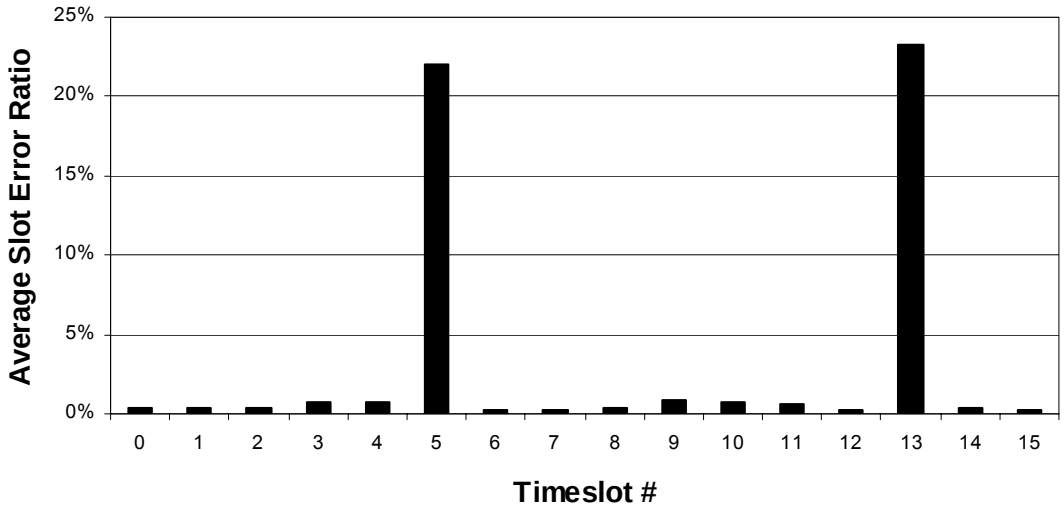


Figure 5.3 Graphical representation of timeslot errors.

Probability of transmission error vs. the number of timeslots used (out of 16) was tested for two fixed message length transmissions (length 5 and 15 timeslots). The packets were transmitted continuously so all timeslots were used equally. The duration of this test was approximately 5 minutes. A successful transmission included an ACK, meaning the time to complete a message transaction was 6 and 16 timeslots respectively. The number of timeslots used was varied from 1 to 16, using the lowest error ratio timeslots first and the highest error ratio timeslots last. The first 14 timeslots had approximately the same error ratio (< 1%) while the last two had error ratio > 20%. Figure 5.4 shows the results of this test and as expected, using the high error ratio timeslots results in a much higher probability of error. Also note that the longer message (length 15) is affected more by the high error ratio timeslots. If a longer packet is used, there is higher probability that it will use a high error ratio timeslot and thus higher average probability of error.

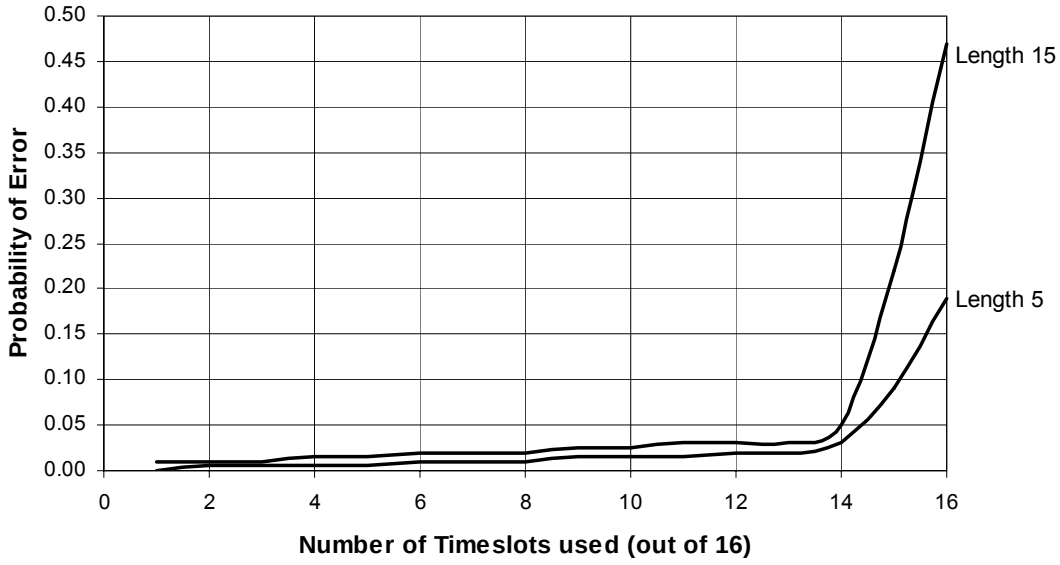


Figure 5.4 Probability of Error vs. Number of Timeslots used (out of 16).

A similar test was conducted to test efficiency vs. number of timeslots used. Again, a 5 minute test was conducted. Ideal efficiency varies linearly with the number of timeslots used (ranging from 1 to 16). If the full capacity of the channel is used (all 16 timeslots)

and there are no data errors, then efficiency is 1.0. If half of the channel is used (8 timeslots) and there are no data errors, then efficiency is 0.5. In the practical case however, data errors cause efficiency to deviate from the ideal case. Again, two fixed length messages were used (length 5 and 15, with message transaction time 6 and 16 timeslots respectively) and the number of timeslots used varied using low error ratio timeslots first and high error ratio timeslots last. As can be seen in Figure 5.5, as the number of low error ratio timeslots used increases there is little deviation from the ideal efficiency curve. However, as the last two timeslots are included for data transmission, not only does the curve fall below ideal efficiency, the efficiency actually decreases which is an obvious indication that these timeslots should not be used for data transmission. Also note that using the high error ratio timeslots has less effect on efficiency of the shorter transmission (length 5) than the longer transmission (length 15). One conclusion that can be drawn from this is that if efficiency does not decrease by adding a higher error ratio timeslot, then it can be used for transmission. The threshold error ratio that causes efficiency to decrease varies with message length. Details of this are studied later in the chapter.

Figure 5.5 Efficiency vs. Number of Timeslots used (out of 16).

5.2 Adapting to BAD Timeslots

Adapting to bad timeslots is the key to the timed power line transmission protocol. For this test, the transmission environment was exactly the same as in the previous section. The master was placed on desk 1 and the slave on desk 2 with a 3 meter extension cord for a total distance of approximately 10 meters. The timeslot error ratio was the same as displayed in Table 5.1.

It is important to choose an appropriate timeslot error threshold for qualifying a timeslot as good or bad. In this case, it is obvious which timeslots are bad as timeslots 5 and 13 have a much higher error ratio than all other timeslots as illustrated in Figure 5.3. Therefore in this case the threshold error ratio for labeling a timeslot BAD was programmed to be 10%, as this easily labels the timeslots.

Figure 5.6 shows an oscilloscope capture of the protocol in action. The traces are (in order from top to bottom): 60 Hz mains voltage, GOOD/BAD timeslot indicator, and received signal and noise at the slave. Notice that the amplitude of the noise (measured in the bandwidth < 500 kHz) exceeds the signal amplitude by several multiples at certain times. The GOOD/BAD timeslot indicator simply produces a high output when the current timeslot is GOOD, and a low output when the current timeslot is BAD. At the time of capturing this image, the master and slave had been communicating for at least one PING cycle (the necessary time to communicate timeslot table information). As expected, the protocol has labeled two timeslots BAD and they correspond in time to the position of the switch-off point of the power-supply rectifiers.

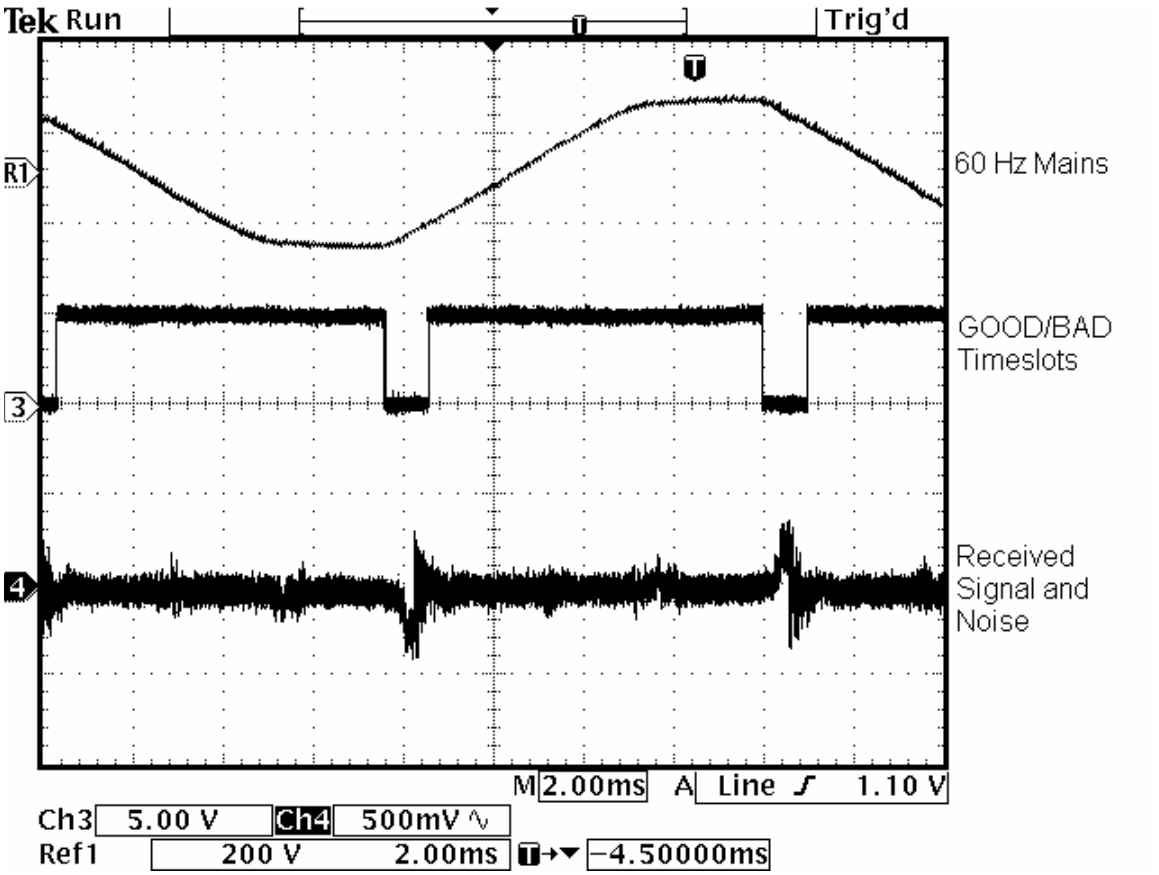


Figure 5.6 AC mains voltage, GOOD/BAD timeslots and received signal.

5.3 Protocol Performance Tests

The above test results agree with the results of Hanson's work [1] in that data errors are synchronous with the 60 Hz mains voltage. Chapter 3 described the protocol to adapt to the timeslots with high error rates, and the purpose of this section is to study the performance of the protocol. In particular, the performance improvement in adapting to the synchronous power line noise is studied.

Performance can be measured by several factors. For this work, probability of transmission success, packet throughput, and transmission efficiency were used as performance measures. There are several variables that must be kept constant between tests such as distance between master and slave, transmit amplitude, time of day, and loads connected to the line. For all of the performance tests in this section, the same test

setup was used as in Section 5.2 with master on desk 1 and slave on desk 2 and a 3 meter extension cord for a total transmission length of approximately 10 meters. The transmit amplitude was kept constant, and all tests were conducted in early afternoon (1-3 p.m.) on a weekday. To verify that the test comparisons were fair, timeslot error ratio was measured before a performance test was conducted to ensure that it was approximately the same as in Table 5.1

The first performance measure is probability of transmission success. For this test, the master and slave go through a check cycle to determine their timeslot tables before sending test data. The test data consisted of variable length messages varying from 1 timeslot to 32 timeslots (two full power line cycles) in length. The test was conducted for a period of five minutes. The success probability was then calculated as the number of successful transmissions divided by the number of attempts. Recall that a successful message transaction consists of the message plus the corresponding ACK. This means that a message of length 1 timeslot will actually require 2 timeslots to send.

Figure 5.7 shows the results of the success probability vs. message length test. As expected, shorter packets have a higher probability of success. The important thing to note from this figure is how quickly probability of success decreases with message length when the master and slave do not adapt to power line noise. This is because as the messages become longer, there is a greater chance that they will transmit through at least one of the high noise times caused by the power-supply rectifiers. However, when the master and slave adapt to power line noise, the rectifier noise is avoided and the probability of success vs. message length remains high.

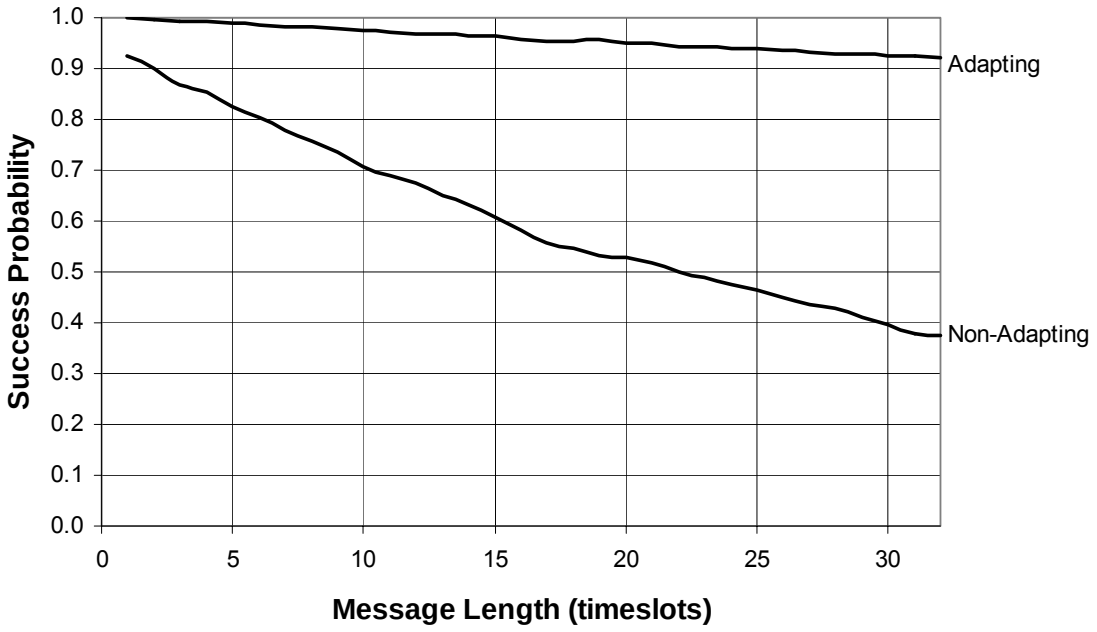


Figure 5.7 Success Probability vs. Message Length.

The next performance measure is timeslot throughput vs. message length. In this test, the master and slave performed their initialization procedure to establish their timeslot tables, and fixed length data messages were transmitted. The number of successful transmissions for each message length was counted, and the throughput was calculated using the amount of time that each message length was sent (range 1 to 32).

Throughput in this sense refers to the capacity of the channel being used. It is expressed on a scale of 0 to 1. It takes into account ACK packets, but does not include overhead within packets (it is not the actual data throughput). For example, a message of length 1 actually requires 2 timeslots (1 for data and 1 for ACK) the peak throughput for message length 1 is 0.5. Hence as messages become longer, throughput has the potential to become higher, but transmission errors can cause it to become lower.

Figure 5.8 shows the throughput vs. message length for message lengths 1 through 32. Curves for non-adapting, adapting, ideal adapting, and an ideal channel are shown. In this particular case, adapting to the noise means avoiding 2 timeslots meaning there is

only 14/16 or 87.5% of the channel available. This actually caused throughput to be slightly lower for adapting than for not adapting for short message lengths (<3). However, once message length increases, adapting to the power line noise shows a considerable performance increase over not adapting. The gap between the ideal channel curve and the ideal adapting curve is due to not using all timeslots. The gap between the ideal adapting curve and the actual adapting curve is due to the errors caused by the good timeslots. Again, as mentioned earlier, if signal amplitude increased the good timeslots showed almost no errors resulting in the actual adapting curve nearly matching the ideal adapting curve.

It is also interesting to note the peak throughputs on the adapting and non-adapting curves. While adapting, the peak throughput is approximately 0.8, while for non-adapting it is just under 0.7. While adapting, the peak throughput is obtained and maintains relatively constant over a significant range of message lengths. When no adapting is performed, the peak throughput is reached at a message length of approximately 5 but falls steadily with increasing message length. Therefore adapting creates a distinct advantage in that not only does it provide higher throughput, but also allows a variety of message lengths to receive high throughput rates.

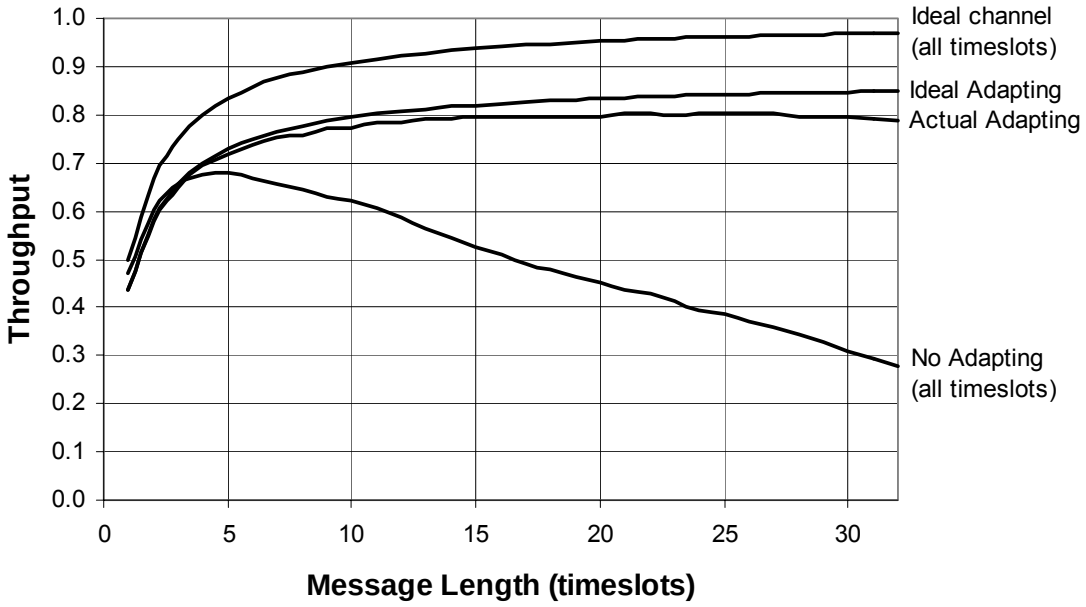


Figure 5.8 Throughput vs. Message Length.

5.4 Testing in Various Environments

The testing in the previous section provided results in an office power line transmission environment with two bad timeslots caused by power supply rectifiers. Further testing was done to study other variables such as distance between nodes, transmitter amplitude, threshold for determining a timeslot bad, and performance with more than two bad timeslots.

5.4.1 Creating different environments

Three new scenarios were created in order to more fully test the performance of the timed power line protocol. This section describes each scenario, demonstrates the adapting ability of the protocol, and shows performance results in each scenario.

Scenario 1: A Power Macintosh connected as a noise producing load

The first new scenario had both the master and slave connected to the power line on desk 1 along with a Power Macintosh 8500/120 computer as a noise producing load on the mains supply. Figure 5.9 shows an oscilloscope trace with the computer connected to the line. The traces are: 60 Hz mains signal, data errors, GOOD/BAD timeslot indicator, and the received signal and noise. Table 5.2 shows the timeslot error percentage. Here the effects of the switch-off of the power supply rectifiers is obvious, but also note the effects of the switch-on point on the positive half cycle of the mains voltage. The reason that only the switch-on point in the positive half cycle was noticed is uncertain, but possibly because rectifier positive half cycle created a larger noise spike or caused noise at a frequency that interfered with the modulated signal. The protocol detects these three bad timeslots, which is seen in the GOOD/BAD trace.

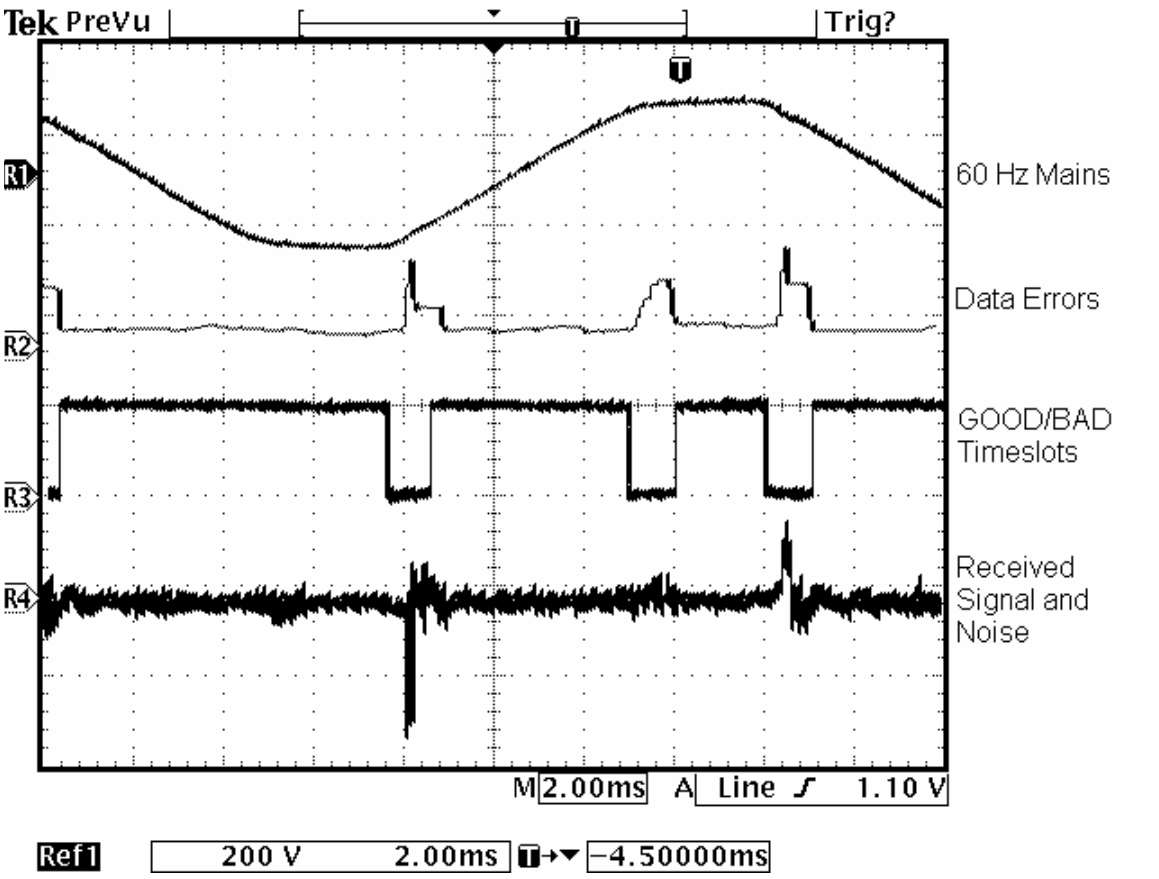


Figure 5.9 Oscilloscope trace with Macintosh computer as a noise producing load.

Table 5.2 Timeslot errors with Macintosh computer as a noise producing load.

Total number of cycles 36901		
Timeslot	# Errors	Error %
0	200	0.54%
1	108	0.29%
2	211	0.57%
3	245	0.66%
4	66	0.18%
5	3497	9.48%
6	93	0.25%
7	116	0.31%
8	229	0.62%
9	139	0.38%
10	3326	9.01%
11	350	0.95%
12	56	0.15%
13	4946	13.40%
14	112	0.30%
15	166	0.45%

Throughput vs. message length was also tested in this environment. The timeslot error percentage with the rectifiers causing error rates around 10 % for the bad timeslots is lower than that seen in earlier tests, so the threshold for determining a timeslot bad was adjusted to 5 %. Peak throughput is actually higher when not adapting than when adapting, because now only 13 out of the 16 timeslots are used and the error rates of the bad timeslots are not as high as in earlier tests (~ 20 % tested earlier). Rather, peak throughput is about the same for each case, but adapting to noise has a throughput advantage for message lengths longer than 5 or 6. Adapting presents the advantage once again of a steady peak throughput with increasing message length, while non-adapting throughput decreases dramatically with longer message lengths. The results of this test are shown in Figure 5.10.

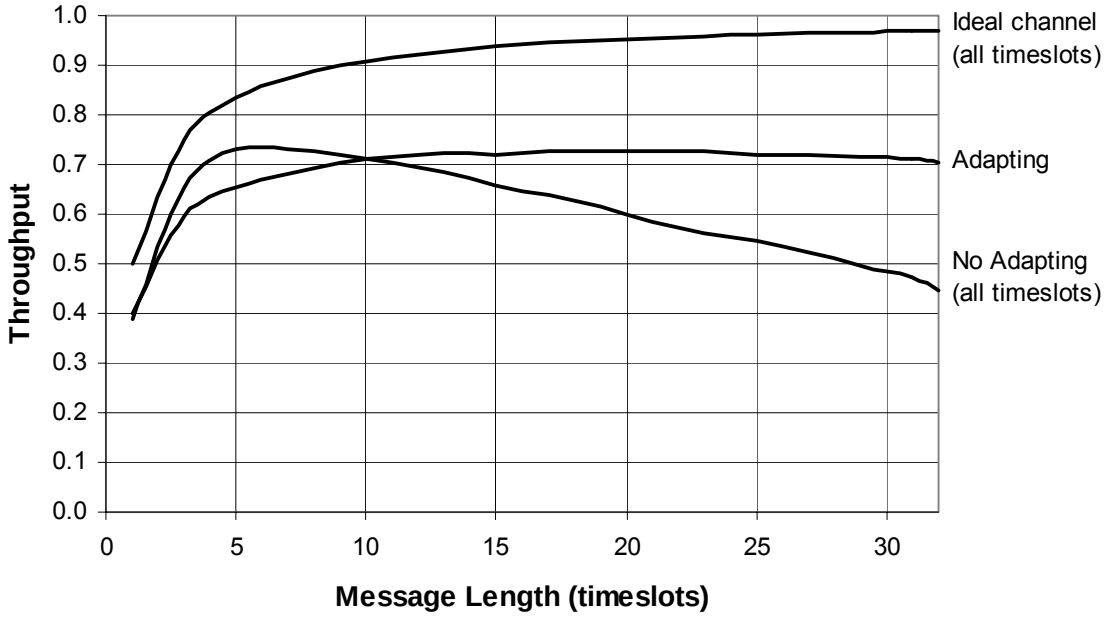


Figure 5.10 Throughput vs. Message Length with Macintosh computer as a noise producing load.

Scenario 2: Lamp dimmer connected as a noise producing load

In this test the master was on desk 1 and slave on desk 2 and a new scenario was created connecting a lamp dimmer. The distance between the Master and Slave was approximately 10 meters, which included a 3 meter extension cord used to connect the slave on desk 2. The lamp dimmer was of the type typically found in homes. With no load connected to the lamp dimmer, it creates no switching noise, so a desk lamp with a 60 W bulb was connected as a load. Figure 5.11 shows the results of connecting the lamp dimmer to the line. In this figure, the lamp dimmer was set on high. The noise spikes caused by the lamp dimmer are noticeable, as seen in the lower trace. The middle trace shows the protocol adapting to the noise spikes, labeling the noise spike time as a BAD timeslot. To demonstrate the continuous adapting ability of the protocol, the lamp dimmer was adjusted from the high (bright) position shown in Figure 5.11 to the low

(dim) position shown in Figure 5.12. Adapting to the noise spikes and labeling timeslots BAD took 10 seconds (the period of time between PINGs from master to slave).

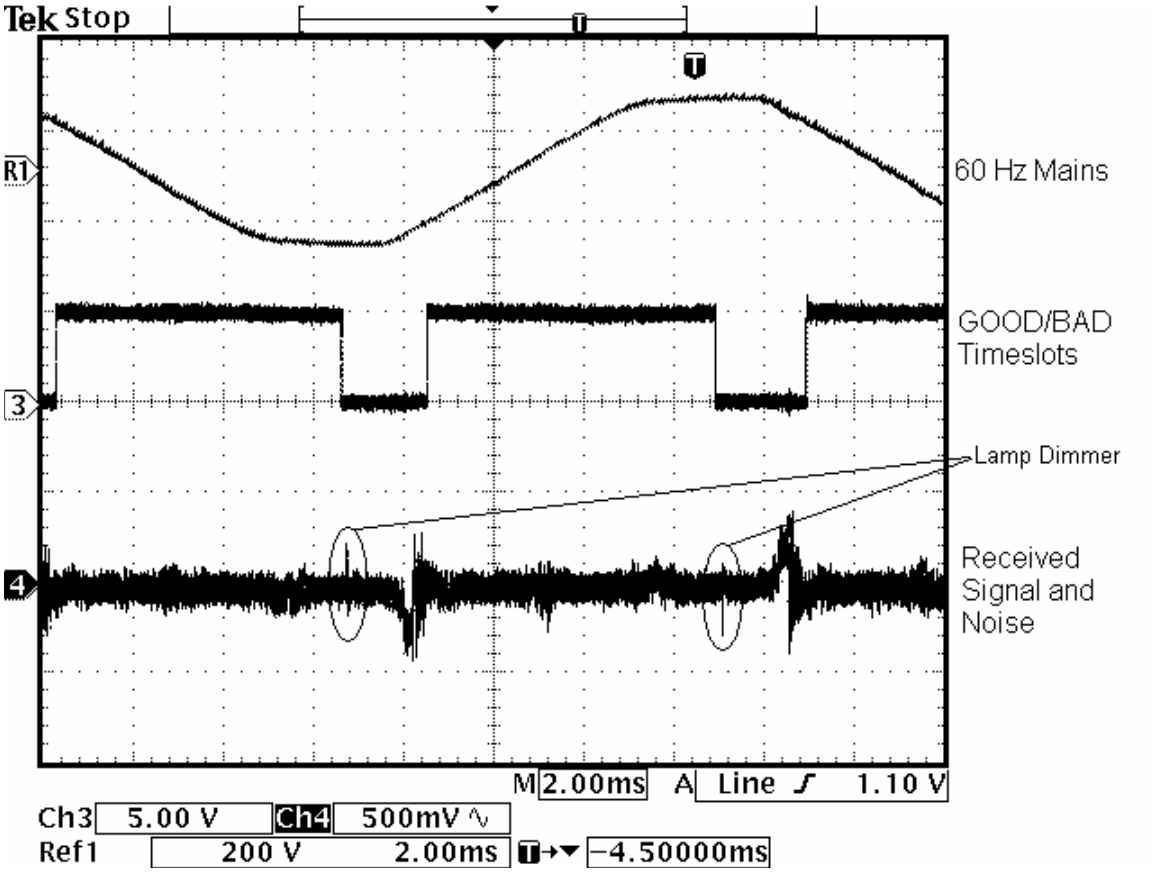


Figure 5.11 Oscilloscope display with lamp dimmer in high position.

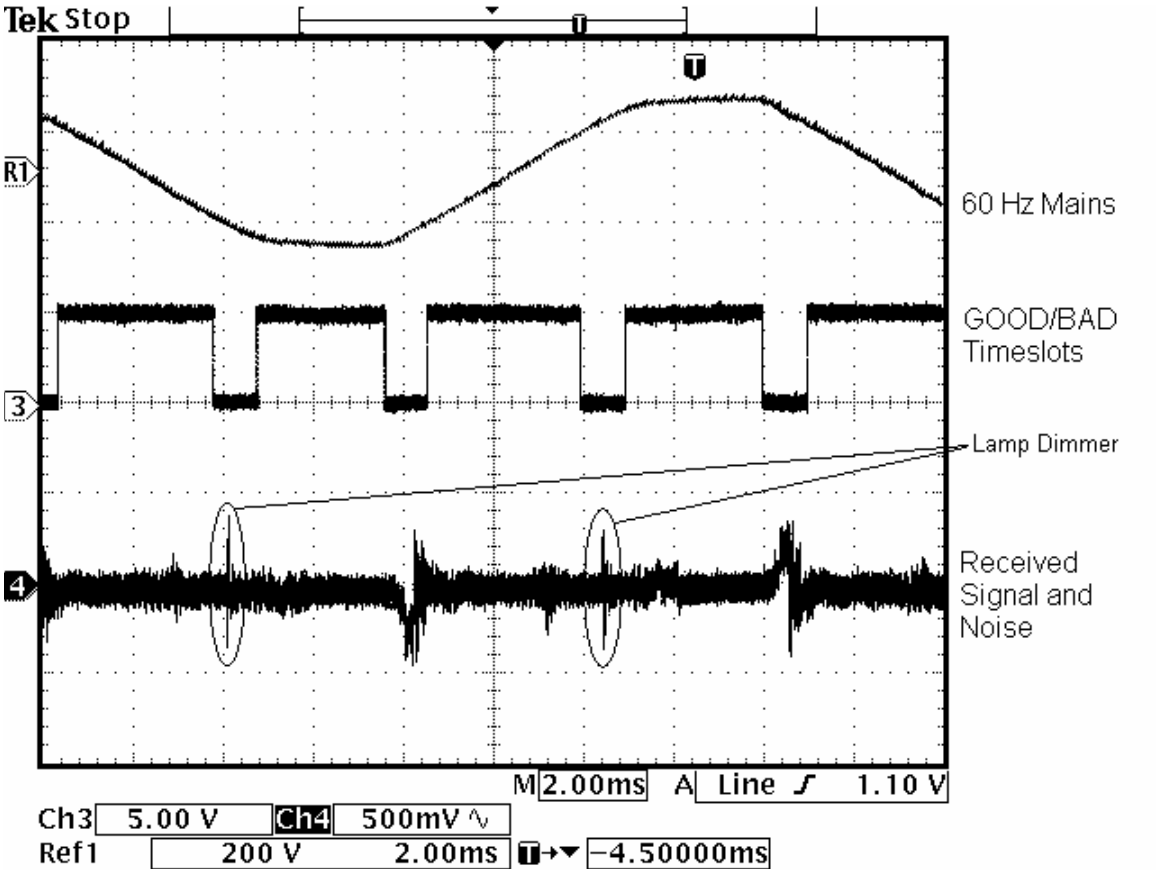


Figure 5.12 Oscilloscope display with lamp dimmer in low position.

With the lamp dimmer set on the low setting, timeslot error percentage is shown in Table 5.3. Here there are four BAD timeslots: the two power-supply rectifier switch-off points and two lamp dimmer switch-in times. In this case, the lamp dimmer spikes cause a large number of timeslot errors (almost 60 %), larger than the timeslot errors caused by the power-supply rectifiers (approx. 20 %).

Figure 5.13 shows the throughput vs. message length for the timeslot error percentage scenario in Table 5.3. Here the advantage of adapting is obvious. The gap between the adapting curve and non-adapting curve is also large because there are four timeslots with high error rates, resulting in a high probability of error in the non-adapting case. Since only 75 % (12/16) of the channel capacity is used in the adapting case, there is a large reduction when compared to the ideal case. When not adapting, peak throughput is 0.5

and is only attainable with a message length of 4. Even though only 12 of 16 timeslots are used when adapting, throughput is much higher (almost 0.7) and is steady over a wide range of message lengths.

Table 5.3 Timeslot error percentage with lamp dimmer in low position.

Total number of cycles 29891		
Timeslot	# Errors	Error %
0	17868	59.78%
1	85	0.28%
2	66	0.22%
3	173	0.58%
4	173	0.58%
5	6172	20.65%
6	46	0.15%
7	69	0.23%
8	17466	58.43%
9	84	0.28%
10	117	0.39%
11	107	0.36%
12	41	0.14%
13	7358	24.62%
14	50	0.17%
15	101	0.34%

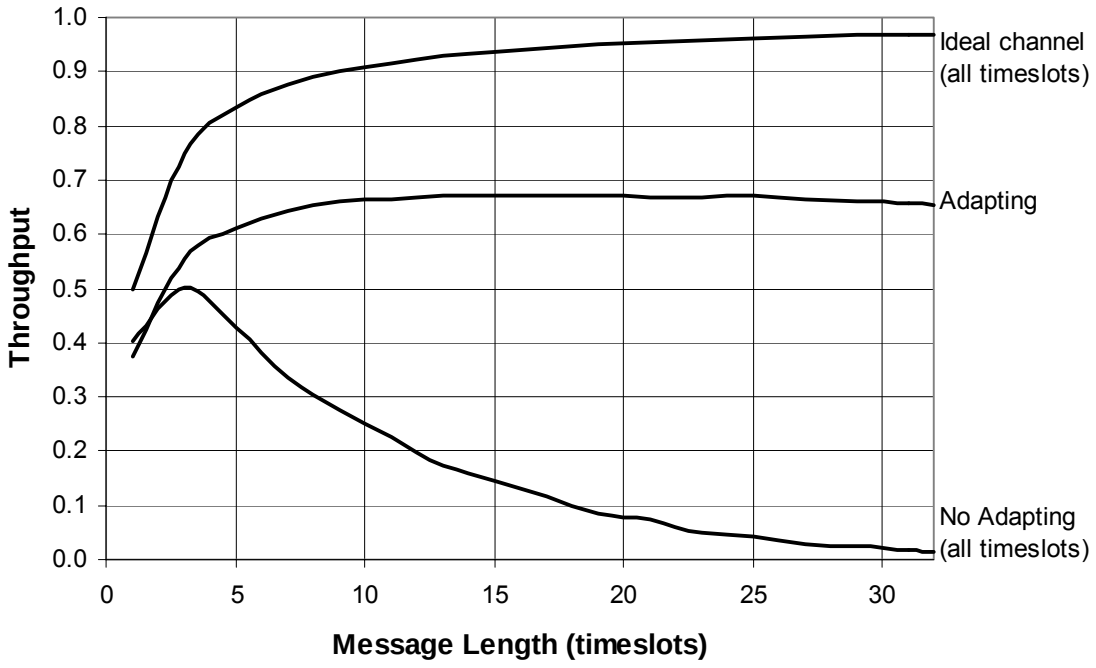


Figure 5.13 Throughput vs. Message Length with lamp dimmer in low position.

Scenario 3: Generally noisy channel

This scenario has master and slave both connected to the power line on desk 1 with no other loads. However, transmitting amplitude was set low (100 mV) so that there was more noise relative to the signal throughout the 60 Hz mains cycle. Table 5.4 shows the timeslot error % in this scenario. As before, the power-supply rectifiers caused the most errors but all other timeslots have a significant number of errors (1-3%), although not enough to be considered bad (5% threshold chosen). The results of the throughput test are shown in Figure 5.14. As before, adapting provided a more consistent throughput with varying message length, although it dropped off more than it did in other cases because of the errors caused by the used timeslots.

Table 5.4 Generally noisy channel timeslot error percentage.

Total number of cycles 52421		
Timeslot	# Errors	Error %
0	709	1.4%
1	636	1.2%
2	606	1.2%
3	1054	2.0%
4	1664	3.2%
5	11824	22.6%
6	523	1.0%
7	663	1.3%
8	819	1.6%
9	1548	3.0%
10	1790	3.4%
11	1478	2.8%
12	1038	2.0%
13	12560	24.0%
14	635	1.2%
15	683	1.3%

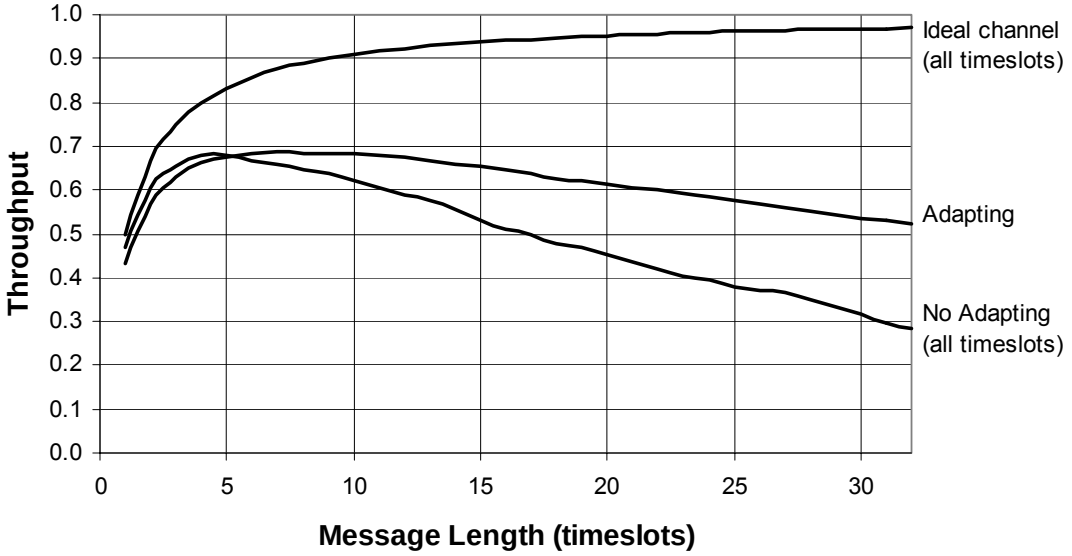


Figure 5.14 Throughput vs. Message Length for generally noisy channel.

5.5 Optimizing the GOOD/BAD Error Threshold

The cases presented above show the need to study an optimum threshold percentage for determining when a timeslot should not be used for transmission. To study this, master and slave were both placed on desk 1, and amplitude was adjusted to obtain specific timeslot error percentages. Results were then gathered for the throughput vs. message length and efficiency vs. number of timeslots used programs. A theoretical analysis follows.

In this setup, there were two bad timeslots, numbers 5 and 13. Throughput vs. message length was tested for three different timeslot error percentages for these two timeslots; each curve is shown in Figure 5.15. While adapting, the curve stayed the same for each case because timeslots 5 and 13 were always avoided and all other timeslots had essentially no errors. In case 1 (error percentages 22-23%) adapting was obviously beneficial as throughput was higher than that for not adapting throughout the message length range. In case 2 (error percentages 8-10%), adapting is still advantageous because

it provides a higher and more consistent throughput over the range of message lengths. In case 3 (error percentages 3-4%) however, throughput is higher throughout the message length range when not adapting, hence adapting provides no advantage. From these results it can be concluded that adapting is advantageous with two timeslots with error percentages > 10%, but also depends on the message length used.

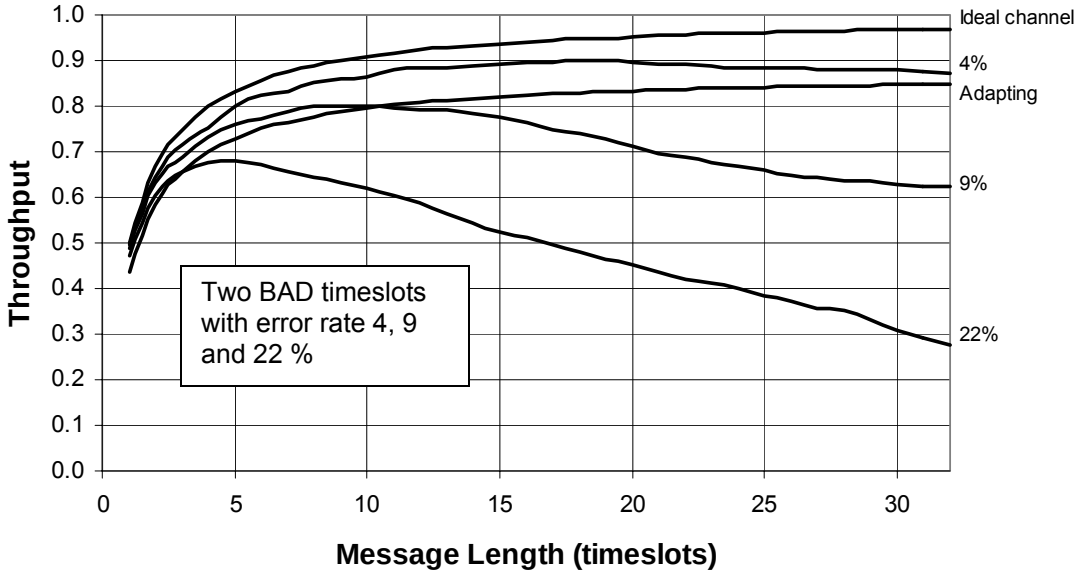


Figure 5.15 Throughput vs. Message Length to test adapting effectiveness.

The error percentages for the above tests were obtained by adjusting the amplitude of the transmitter, which was difficult to do accurately. For this reason, a theoretical analysis was conducted, which gives much more accurate results. The following theoretical analysis chooses timeslot error threshold for optimizing throughput.

In the simplest theoretical case, consider a single bad timeslot and no errors in any other timeslots. If the protocol avoids the timeslot, the throughput will be 15/16 or 93.75% of the peak throughput. Hence when not adapting, the threshold for determining a timeslot bad will occur when the throughput is 93.75% of the ideal throughput. The length of the packet has a direct effect as well, because shorter packets have less probability of hitting bad timeslots. For a message of length 15 (occupies 16 timeslots because of the ACK) every timeslot will be used once in each message transmission. Assuming all timeslots

have no errors except the bad timeslot, the bad timeslot must have a 93.75% success rate (or a 6.25% error rate) to give the same throughput as when avoiding the timeslot. Above 6.25% error rate, the timeslot can be labeled BAD and below 6.25% the timeslot can be labeled GOOD. Figure 5.16 shows the threshold percentage vs. message length when considering a single timeslot. Notice for short messages (< 5 timeslots in length) error threshold becomes quite high, reaching 50 % for a message of length 1.

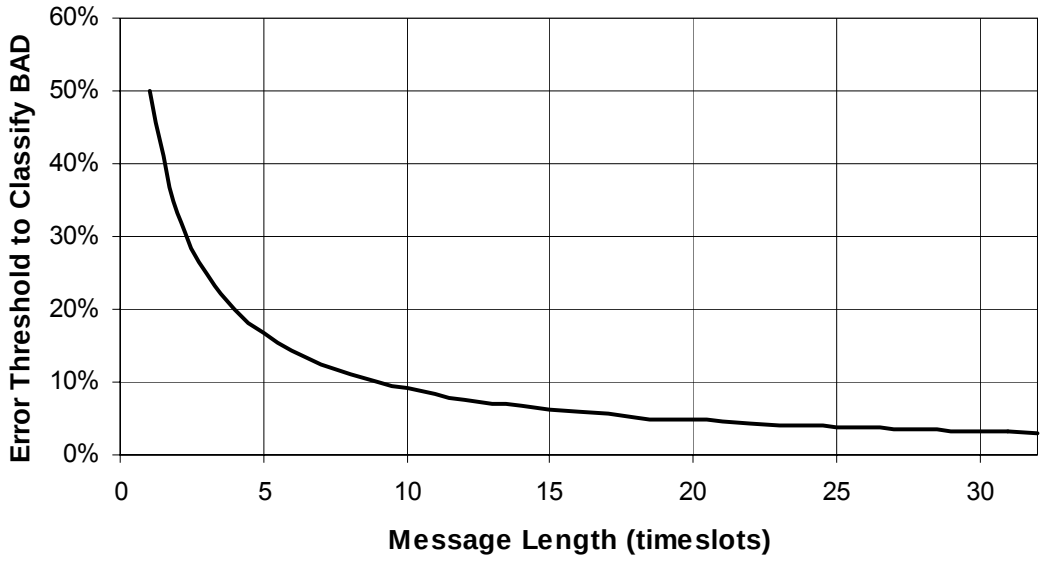


Figure 5.16 Error Threshold to Classify BAD vs. Message Length for a single timeslot.

A similar argument can be followed when considering one timeslot already bad, and adding a second bad timeslot. If one timeslot is already bad, adapting throughput is 15/16 of the ideal channel throughput. When considering adapting to the second BAD timeslot, the throughput would be 14/16 or 87.5% of the peak. This means that the threshold for considering the second timeslot bad is (14/16)/(15/16) or 14/15 = 93.3%. For a message length 14 that occupies all used timeslots (15 timeslots in this case) the timeslot under consideration must have at least a 93.3% success rate or less than 6.7% error rate to be considered good.

As stated earlier the message length affects the optimum threshold error rate. The above analysis assumes that each timeslot is used once in each message transmission. For a message length 31 (occupies 32 timeslots, or 2 full power line cycles) every timeslot will be used on average twice in every message transmission. That means that the success rate for each message must still be 93.75 %, but the required success rate for the timeslot is *higher*. If the success rate of the timeslot is SR, and for a message length of 31 the timeslot is used on average twice, then SR*SR = .9375, or SR = .968 or 96.8 %.

Equation 5.1 is a general equation for the threshold for considering a timeslot bad. If a timeslot has error rate above this, it can be labeled BAD and with an error rate below this it can be labeled GOOD. Note that this formula assumes that a single timeslot is causing all of the errors, that is, all other timeslots being used have no errors.

$$optimum\_errorthreshold = 1 - \left(\frac{numcurrentlygood - 1}{numcurrentlygood} \right)^{\left(\frac{numcurrentlygood}{messagelength+1} \right)} \qquad (5.1)$$

For example, for a message length of 10 timeslots and all timeslots currently GOOD, the optimum threshold percentage is 1-(15/16)^(16/11) = .091 or 9.1 %. For a longer message, say 20 timeslots, the optimum threshold percentage is 1-(15/16)^(16/21) = .048 or 4.8 %. A length of 20 timeslots and only 10 currently GOOD results in 1-(9/10)^(10/21) = .467. This shows that the optimum error threshold is affected much more by the message length than by the number of timeslots currently good. In essence, for longer message lengths, the optimum error threshold is smaller.

When considering two timeslots at once with different error rates, the analysis is similar. For a desired success probability however, this will be the probability that *both* timeslots are good (assuming that they are independent events). Hence when considering the first two BAD timeslots, a 14/16 or 87.5% success rate or less than 12.5% failure rate must be obtained for the combination of the timeslots to consider them both GOOD, hence P(Both GOOD) > .875 is the target, where P(Both GOOD) is the probability that both timeslots are GOOD. P(Both GOOD) > 0.875, or P(first GOOD)*P(second GOOD) > 0.875. For

two timeslots with the same error rate, and using each timeslot once in every message transmission, the success threshold for each is 93.5% (error threshold 6.5%) which is close to the threshold when considering a single timeslot (6.25%).

If a pair of timeslots gives the desired success threshold of 93.5% but one timeslot has higher error rate than the other (e.g. 7% and 5%) then it is better to label the first timeslot (7%) BAD and the second timeslot (5%) GOOD because the second timeslot is below the threshold for considering a second timeslot bad. Therefore formula 5.1 given above is considered a reasonable error threshold approximation to consider a timeslot BAD for a given message length, one timeslot at a time from highest error rate to lowest.

5.6 Other Factors that Affect the Channel

The above analysis shows that the noise characteristics affect throughput and efficiency and how well the timed power line protocol performs. These characteristics vary due to the type, location and number of loads connected to the line as shown by the Macintosh computer and lamp dimmer tests conducted above. Other factors which can affect performance include time of day, distance between nodes and the transmitting amplitude.

5.6.1 Time of Day

Change in channel characteristics due to time of day is caused by loads connected to the line. During the day there are more lights turned on, more computers turned on, and generally much more load on the power line than at night. This creates a noisier channel with higher attenuation than at night.

To test the effect of time of day, the master was connected to the power line at desk 1 and the slave at desk 2 with a 3 meter extension cord. One way transmission (from master to slave) was carried out with the master simply broadcasting check packets to the slave, and the slave reporting the running average timeslot error percentage every 10 seconds. The nighttime test was conducted first. The amplitude of the master's transmitter was set purposely to cause a small number of errors in all timeslots except where the power-

supply rectifiers conduct (where there were a large number of errors). 10 five-minute tests were conducted at different times in the evening ranging from 7 p.m. to 2 a.m. For the daytime test, the master's transmitter was set to the same amplitude as was used for the nighttime test. This was essential in measuring the daytime performance relative to the nighttime performance. Again, 10 five-minute tests were conducted at different times of day ranging from 10 a.m. to 4 p.m. Table 5.5 shows the timeslot error percentage average for night and day testing. In this case, the timeslot errors at the rectifier switch-off points remained about the same. This may be because power supplies connected to this line are not turned off at night. Another explanation is that the maximum number of errors that can be caused in those timeslots already occurs at night because the zener diodes in the wall coupler are clipping the noise spikes, meaning that a further increase in amplitude of the noise spike has little or no effect. All other timeslots, however, behaved as expected and the error percentage was significantly higher during the day than at night.

Table 5.5 Average timeslot error percentages for night and day tests.

Timeslot	Day Error %	Night Error %
0	1.4%	0.4%
1	1.2%	0.4%
2	1.2%	0.4%
3	2.0%	0.8%
4	3.2%	0.8%
5	22.6%	22.0%
6	1.0%	0.2%
7	1.3%	0.2%
8	1.6%	0.3%
9	3.0%	0.9%
10	3.4%	0.7%
11	2.8%	0.6%
12	2.0%	0.2%
13	24.0%	23.3%
14	1.2%	0.4%
15	1.3%	0.2%

5.6.2 Distance between nodes

This test was conducted with both the master and slave on desk 1. Since the variable under study was the distance between nodes, the transmitting amplitude remained constant. Distance between the nodes was varied by using extension cords. Three extension cords of length 3 meters were used, along with two cords of length 30 meters. The first measurement was conducted with the master and slave connected to the same bank of outlets on desk 1. The actual distance through the power line was about 30 centimeters. Again the master sent check packets in each timeslot, and the slave checked the quality of the received data and reported it on the computer screen every 10 seconds. Timeslot error percent for this distance was averaged over a period of approximately five minutes, then a length of extension cable was added and the test repeated. It should be noted that extension cable is not the same cable as power conductors installed in homes and offices, but is a reasonable approximation for this test. Also, this was the only way to know the actual wiring distance between nodes because power conductor length in walls is not predictable without opening the wall. The test setup is shown in Figure 5.17.

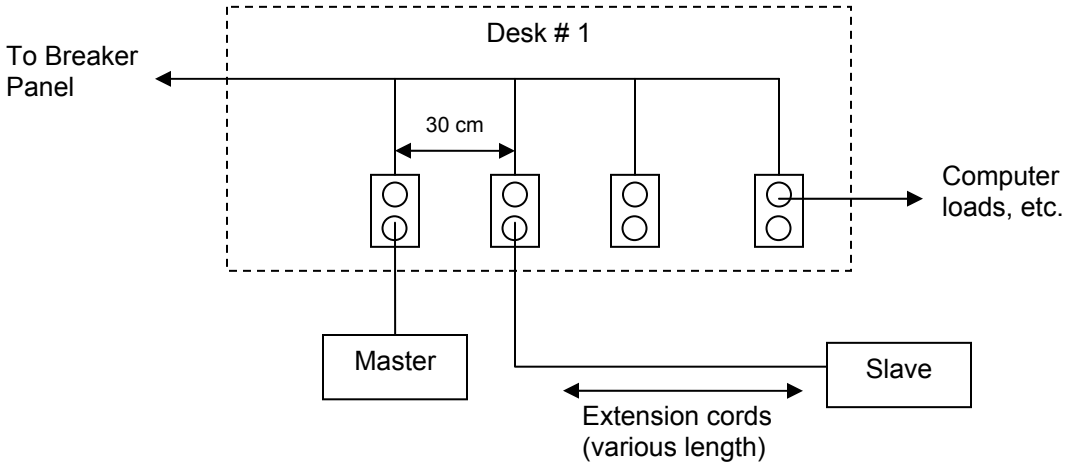


Figure 5.17 Station separation test setup

Table 5.6 shows the result of the distance test for each timeslot. As expected, increasing distance between nodes caused more errors because of attenuation in the cables. Note that timeslots 5 and 13 showed approximately a threefold increase in errors when

distance was changed from 3 to 6 meters. It is assumed that, at this point, the noise from the power supply switch-off points began to dominate over the data signal level. It is important to note that the highest error timeslots remained the highest for each length of cable, supporting the argument that these timeslots have a higher error ratio *relative* to the other timeslots.

Figure 5.18 shows the *average* timeslot error rate (averaged over all timeslots) vs. length. Note the sharp rise at short distance is due to timeslots 5 and 13 discussed above. At long distances, an exponential increase in errors occurs because loss over the length of the extension cord causes the signal to begin disappearing. At the longest length (69), the error rate isquite high in all timeslots. This is due not only to the length between the nodes, but the small signal amplitude used in order to study length effects. To see if communication was even possible at 69 meters, the signal amplitude was increased and the *timeslot error percent became similar to that when the nodes are close together with small signal amplitude*.

Table 5.6 Timeslot Error Ratio vs. Station Separation.

		Station Separation (meters)							
		On Desk	3	6	9	30	39	60	69
Timeslot #	0	0.0%	0.0%	0.1%	0.2%	1.8%	5.3%	18.8%	32.7%
	1	0.0%	0.0%	0.0%	0.3%	1.4%	3.9%	15.7%	25.7%
	2	0.0%	0.0%	0.0%	0.3%	2.1%	5.4%	18.9%	29.7%
	3	0.0%	0.0%	0.0%	0.3%	2.4%	6.2%	21.7%	37.1%
	4	0.0%	0.0%	0.0%	0.5%	2.3%	7.0%	23.3%	33.6%
	5	15.8%	16.2%	48.6%	53.4%	60.1%	66.3%	79.7%	85.8%
	6	0.0%	0.0%	0.1%	0.2%	2.6%	4.7%	18.5%	27.4%
	7	0.0%	0.0%	0.0%	0.2%	1.4%	3.5%	13.7%	24.0%
	8	0.0%	0.0%	0.0%	0.6%	4.9%	10.7%	24.3%	35.5%
	9	0.0%	0.0%	0.0%	0.3%	2.7%	5.3%	17.8%	33.5%
	10	0.0%	0.0%	0.0%	0.2%	1.1%	2.8%	10.8%	19.1%
	11	0.0%	0.0%	0.0%	0.2%	0.4%	1.4%	8.2%	20.7%
	12	0.0%	0.0%	0.0%	0.1%	0.2%	0.7%	4.9%	13.6%
	13	19.2%	20.0%	61.7%	62.7%	68.9%	73.4%	81.8%	86.9%
	14	0.0%	0.0%	0.0%	0.2%	1.3%	3.6%	14.5%	24.2%
	15	0.0%	0.0%	0.0%	0.2%	1.1%	3.5%	14.8%	24.5%

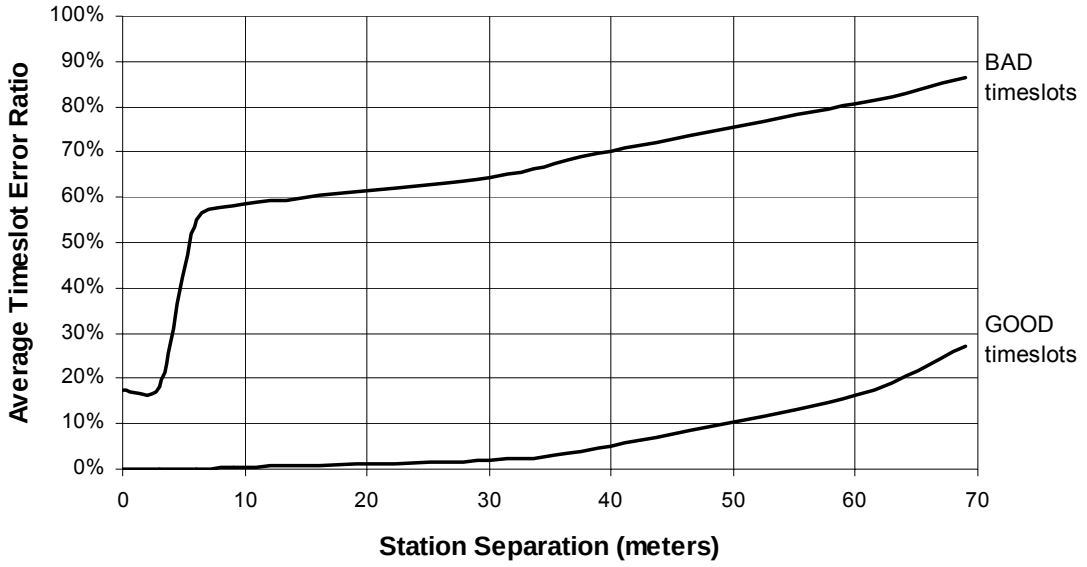


Figure 5.18 Average Slot Error Ratio vs. Station Separation.

5.6.3 Effects of Signal Amplitude

The results of the station separation test showed the importance of transmitting signal amplitude. For the signal amplitude test, the master was placed on desk 1 and slave on desk 2 (separation approx. 3 meters) with a 3 meter extension cord. To cause a high number of errors, the lamp dimmer was also used and adjusted so that its noise spike occurred at the same time as the rectifier switch-off point. Signal amplitude was set low to start (100 mV), and then increased while monitoring timeslot error percentage. Again, the master sent check packets to the slave which monitored timeslot quality and reported it every 10 seconds. Timeslot error percentage was averaged over a period of approximately 5 minutes for each signal amplitude.

Table 5.7 and Figure 5.19 show the results of the amplitude test. Timeslot error percentage is quite high in two timeslots at low amplitude and, as expected, it decreases as amplitude increases. The signal amplitude as shown on the graph is relative to the starting amplitude (the lowest amplitude measurement). Of particular interest was to see if the amplitude could become large enough to overcome the errors. In this particular setup, it could not; the amplitude increased to a point where it began to overwhelm the

receiver (amplitude exceeded the allowed range by the demodulator), and increasing the amplitude further caused more errors.

Table 5.7 Timeslot Error Ratio vs. Amplitude.

		Signal Amplitude Relative to Starting Amplitude of 100 mV							
		1	1.5	2	2.5	3	3.5	4	4.5
Timeslot	0	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%
	1	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.8%
	2	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	1.5%
	3	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.4%	9.5%
	4	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	2.8%
	5	54.4%	19.9%	12.2%	7.4%	5.2%	4.7%	5.3%	27.5%
	6	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.7%	12.3%
	7	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.7%
	8	0.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.3%
	9	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.5%
	10	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	1.2%	18.8%
	11	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	1.2%
	12	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	13	61.0%	34.2%	24.3%	14.4%	8.9%	7.5%	7.8%	40.8%
	14	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%	5.5%
	15	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

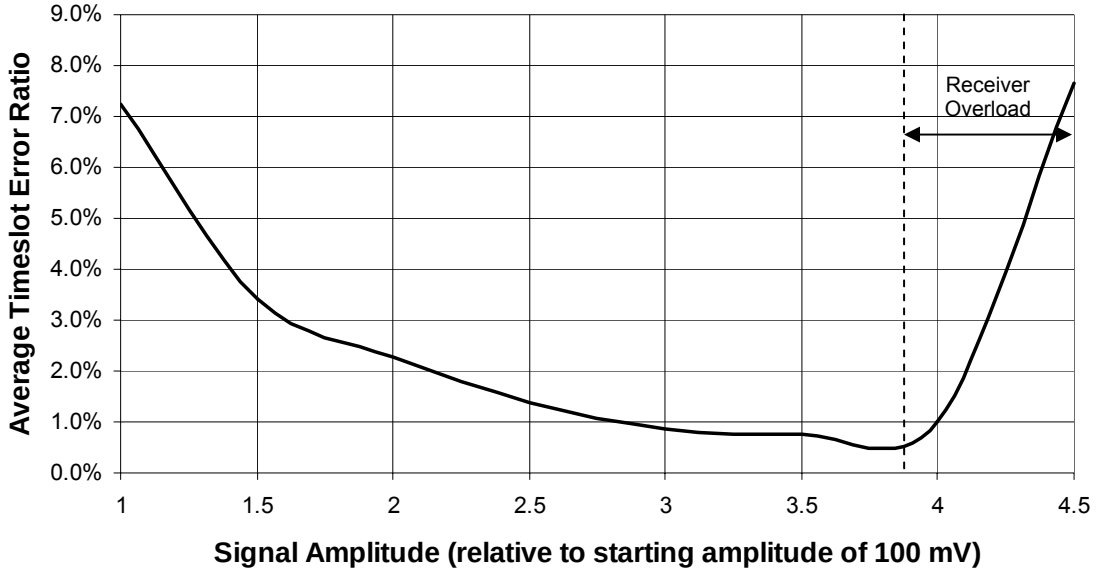


Figure 5.19 Average Timeslot Error Ratio vs. Signal Amplitude.

5.7 Protocol Drawbacks

5.7.1 Multi-phase Network

One drawback to the designed protocol is that the design has considered transmission on only a single phase. In an office environment with three phase supply, this means that approximately two thirds of power outlets would be unavailable for transmission. Also, since not all phases are present in one room, access to the network would be limited. A solution to this problem would be to form a separate network for each phase and connect them through a network bridge. A network bridge bridges data across two or more networks (in this case the three phases). A bridge does not forward all data that it receives; rather it has built in logic that makes forwarding decisions based on the destination of the data. In Figure 5.20 if computer A wants to talk to computer C, and the network bridge recognizes that they are on different networks, it forwards the data on computer C's network. If computer B wants to talk to computer D, the bridge does no forwarding of data because B and D are on the same phase or network. This approach reduces the amount of data flooding the network.

The multi-phase issue may also cause problems for the master/slave design chosen. With a single master coordinating all three slaves with a single timeslot table, there would be a large portion of the power line cycle labeled BAD because the power-supply rectifier switch-off points on one phase occur at different points relative to the other phases. If, for example, each phase has two bad timeslots due to power-supply rectifiers then the three-phase network will have six bad timeslots. One possible solution to this is to have the master and the network bridge work together to know when to forward packets. A master table is still constructed for the entire network (all three phases) and forwarded on to all slaves, but it takes into account the phase shift of each phase. For example, if timeslots 5 and 13 are bad in each phase due to power-supply rectifiers, then the master constructs a timeslot table with 5 and 13 labeled BAD even though timeslots 5 and 13 occur at different times for each phase relative to the other two phases. If data needs

forwarded from phase 1 to phase 2 and is received by the bridge in timeslot 4, then it will forward the data to phase 2 in its timeslot 4 (which occurs at a different time than phase 1's timeslot 4).

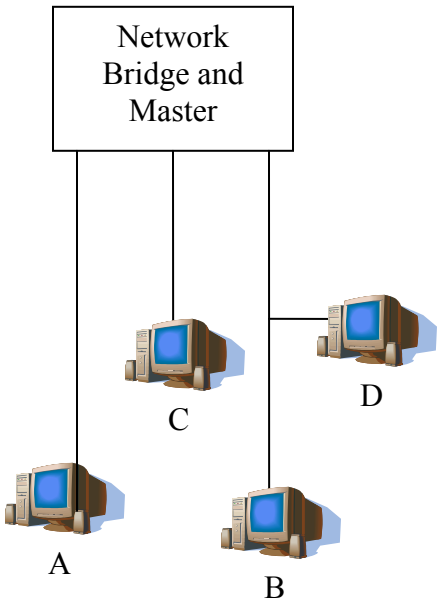


Figure 5.20 Three phase power line network.

5.7.2 Centralized vs. Distributed Architecture

This design has used a centralized architecture (master/slave setup). This has provided certain advantages such as timeslot table coordination, coordination of slave transmission (eliminates the need for carrier sensing and collision detection), and a general sense of organization. However using this centralized architecture has certain drawbacks.

The main drawback is the master timeslot table. Each timeslot in the master table is the union of GOOD timeslots of all slaves. If at least one slave labeled a timeslot BAD, then it is labeled BAD in the master table and broadcast to all slaves. If two slaves who are spatially close to each other have good transmission success in all timeslots, they are still limited to those timeslots labeled GOOD and BAD in the master table. One solution to this is to have each node keep track of a timeslot table for every other node. When one

node wants to talk to another, it negotiates a timeslot table at that time and carries out communication. This approach has the advantage that two nodes can communicate with each other with a timeslot table that is optimized between the two of them, and not affected by other nodes' timeslot tables. This is a distributed architecture now, where all nodes have equal access to the channel. Disadvantages to *this* approach are each node must contain enough memory to store a timeslot table for every other node in the system, and now methods such as carrier sensing and collision detection/avoidance must be employed to avoid collisions and ensure equal channel access because now every node makes its own decisions on when to transmit.

A further improvement is to have a blend of the two architectures where there exists a master but its only purpose is to coordinate who is allowed to talk and when. Timeslot tables are left up to each slave to negotiate with other slaves and there is no master table. Now there would be no need for methods such as carrier sensing and collision detection/avoidance, and each node pair can negotiate independent timeslot tables which optimizes communication.

5.7.3 Excess Overhead

One major problem with this protocol is the amount of overhead required for successful detection of BAD timeslots, communicating this information, and allocating channel access. There are two basic categories of overhead in this design, packet overhead and control overhead.

Packet overhead is the overhead included in each packet. As shown in Chapter 3, each timeslot contains a complete packet including header and checksum. This is necessary in order to be able to successfully detect where errors are occurring and to flag BAD timeslots. At low bit rates this becomes more of an issue because the number of overhead bits remains the same but the number of data bits must decrease in order for the packet to fit in the one millisecond timeslot. For the purposes of this work, 60 kb/s was used, and less than half of the bits sent were actual data. If higher bit rates are used, the overhead to data ratio is much lower.

Control overhead refers to the time dedicated to communicating timeslot table information and controlling channel access. Every 10 seconds, the master performs a polling cycle, in which each slave is polled for information for one power line cycle and each responds for one cycle. If all 16 slaves are present in the system, this overhead amounts to 32 cycles. One power line cycle is 16.67 ms so this causes a pause of approximately 0.5 seconds. If each slave requests channel access in their response (all slaves have data to send, this is the worst case) then the master needs one power line cycle for each slave requesting channel access. This is a total of another 16 power line cycles over the 10 seconds. In 10 seconds there are 600 power line cycles. The total overhead in the worst case is 48 power line cycles, resulting in 48/600 or 8 % overhead for timeslot table coordination and controlling channel access.

6. Conclusion

6.1 Summary

This thesis has reviewed published work on power line electrical characteristics and has proposed a method of avoiding synchronous noise present on the power line.

Hanson's research [1] showed that synchronous noise on the power line causes a high number of bit errors. This result was confirmed in this thesis with the half-duplex modems and programs that tested byte and timeslot errors. As expected, a high error ratio was discovered at the same location of the synchronous noise. In the tests conducted, the noise spikes from power supply rectifier switch-off caused the most errors. In other environments, however, it may be found that other noise spikes can cause additional errors, such as those caused by fluorescent lights (near the zero crossings of the 60 Hz mains signal).

A protocol was designed by combining new ideas with ideas from previously designed protocols. The protocol was designed as a centralized architecture with a single master and up to 16 slaves. The protocol divides the 60 Hz mains signal into 16 timeslots (approx. one millisecond each). Each node in the system monitors received data for errors in each timeslot and labels timeslots with too many errors as BAD. The master controls traffic on the power line by polling slaves to see if they have data to send or have any information on BAD timeslots. By doing this, the master makes a union of all GOOD timeslots in the system (assembles a *timeslot table)* and broadcasts this information to all slaves. Once all of the slaves have the timeslot table, the master designates the channel to a particular slave that needs to send data for a number of 60 Hz cycles fitting the amount of data that slave has to send. The slave will only communicate data in GOOD timeslots, and the master will send a check pattern in BAD timeslots so

their quality can be monitored. Once this number of cycles has expired, the master will designate the channel to another slave and so on. Every 10 seconds, the master polls each slave again to see if they have data to send or have any new timeslot table information and then the process repeats. This process of polling and continuous checking of timeslots allows the protocol to adapt to power line noise, detecting when new noise sources appear or disappear.

6.2 Conclusions

This method of adapting to synchronous noise worked well. Performance was tested with and without adapting to noise. More specifically, packet throughput and probability of transmission success were tested. These two factors were measured for various message lengths (where a message length refers to the number of timeslots occupied by a message) and for several environments.

Based on the research performed in this thesis, it is found to be advantageous to avoid synchronous noise on the power line. This results in data transmission at times of low probability of error and increased performance. It was concluded that throughput and probability of success could be significantly improved by adapting to power line noise. The error threshold for determining that a timeslot is BAD was also investigated. It was found that the best threshold error ratio for determining a timeslot BAD was approximately 6 % (a bad CRC 6 % of the time). This threshold, however, varies depending on the number of timeslots that are already labeled BAD, and on the message length under consideration. If there are already a few timeslots labeled BAD, the optimal threshold error ratio is slightly higher. Also, for shorter message lengths, the optimal threshold error ratio is higher. These two variables were then used in determining an equation for calculating the error threshold for determining a timeslot BAD.

Improvements in throughput and reliability were also discovered. Throughput varies depending on the number of BAD timeslots, but for a typical case of two BAD timeslots the throughput obtained when adapting to noise was over 80 % of the peak throughput as

compared to approximately 60 % of the peak throughput when not adapting to noise. Also, when not adapting to noise, maximum throughput peaks at low message length and drops off rapidly for higher message lengths. When adapting however, throughput is much more constant with varying message length. By choosing appropriate modulation, it is thought that throughputs of 100 to 500 kb/s may be possible.

The major contribution of this work is the design of a protocol to take advantage of the fact that noise is synchronous with the 60 Hz power line cycle. Tests have shown that the protocol was able to adjust the timing of data transmission to avoid the synchronous noise. Tests also showed that the protocol is robust in that it adjusts to changes in the noise characteristics. The protocol, along with relatively simple hardware allows for a low cost solution to medium speed power line networking.

6.3 Future Work

The results of this work have shown that data communication performance can be increased by avoiding synchronous noise. Of significant importance is the error ratio in times of high noise *relative* to that in times of low noise. Future work suggestions are provided below to further take advantage of this fact. In some of these future work suggestions, it may not be necessary to construct hardware and test on the power line where simulations may suffice. Other tests require hardware and actual power line testing.

6.3.1 Modulation

The half-duplex modems designed for this research used FSK modulation. FSK, although simple to implement, is spectrally inefficient. In the same bandwidth targeted in this research (105 kHz – 490 kHz) a much more spectrally efficient modulation scheme should be explored. If synchronous noise is avoided, some M-ary form of modulation such as QAM that has multiple bits per symbol seems possible in this transmission environment. A transmit/receive pair should be designed that can select between different levels of QAM, such as 4-QAM (QPSK) which has 2 bits per symbol to 16-QAM which has 4 bits per symbol. For initial tests, simplex (one-way) communication

can be performed to test error rates with different modulation schemes. Data transmission performance should be measured in typical environments within an office building and within a home. Once a particular modulation scheme is considered appropriate, half-duplex or even full-duplex modems should be designed to test protocol performance and throughput rates under the new modulation scheme. This will increase the number of bits per timeslot and reduce the overhead-to-data ratio. For example, with FSK a symbol rate of 50 kb/s means that 50 bits can be fit into a one millisecond timeslot, however 32 bits are used for overhead (header and error checking). Hence 32/50 or over 64 % of the bits are overhead (not considering UART start/stop bits), and 18/50 or 36 % of the bits are data. If QPSK is used (4 bits per symbol) at 50 ksymbols/sec, 200 bits could be fit into a one millisecond timeslot for a rate of 200 kb/s. Now only 32/200 bits or 16 % of the bits are overhead, and 168 bits of data are possible – a significant improvement. With two BAD timeslots, the effective bit rate would be 200 kb/s * (14/16) = 175 kb/s. Note that this is the bit rate and not throughput (which would consider overhead).

6.3.2 Selectable Bits Per Symbol

Discrete multi-tone systems (DMT) [31] are currently being developed for power line communications. DMT is similar to OFDM that is used in the HomePlug specification, but instead of rejecting bad frequency bands, it still uses them and applies a lower number of bits per symbol to them. High SNR frequency bands are assigned a high number of bits per symbol. The HomePlug specification is exploring this method and claims that rates exceeding 100 Mbits/sec can be attained.

The idea of DMT is interesting to this work because what DMT is doing in the frequency domain could also be done in the time domain. Instead of selecting a number of bits per symbol for a frequency channel, bits per symbol can be selected based on the SNR (or error ratio) in each timeslot. Low noise timeslots could be assigned a high number of bits per symbol while high noise timeslots such as the power-supply rectifier switch-off points may be assigned a small number of bits per symbol or completely avoided.

6.3.3 Length of Timeslots

The length of timeslots may be another area for improvement in this protocol. The FSK demodulator contains a phase locked loop (PLL) which loses lock when a noise spike occurs hence causing bit errors for some time after the spike because of the time the PLL takes to regain lock. If this factor is eliminated, one may find that noise spikes cause errors for much shorter times than measured in this work.

Timeslots of one millisecond were chosen for this research because they are long enough to carry sufficient data for adapting and control, yet short enough to provide some definition in selecting bad timeslots. If it is found that the time of interfering noise is much shorter than one millisecond, the period of time that needs to be avoided is also shorter. By avoiding this shorter time, the amount of time the channel can be used is larger. However, as timeslots become shorter, the fraction of the timeslot used for overhead becomes larger. This needs to be explored in order to obtain a balance.

6.3.4 Amplitude Adapting

As shown earlier in this chapter, transmitting amplitude greatly affects the timeslot error rate. However, turning the amplitude too high resulted in the transmit signal overwhelming the receiver and timeslot error rate began to increase. Some method of determining an appropriate transmit amplitude must be explored. One possibility is to have each slave start up with some default transmit amplitude and have the master listen to transmissions from each slave. If the master finds that it cannot hear a particular slave very well or if other slaves have problems hearing that slave, the master may send a message to that slave indicating that it should increase its transmitting amplitude. However, if a node is far away from some other nodes in the system and it has to transmit at high amplitude to be heard properly, those nodes that are close to it may be overwhelmed by this transmit signal. To solve this problem, automatic gain control (AGC) could be used which changes the receiver filter gain depending on the amplitude of the received signal (for high amplitude, filter gain decreases).

The idea of amplitude adapting may not be feasible or more difficult than its worth for the gain in performance it may provide, but is provided here as another possible improvement to the timed power line communication protocol.

6.3.5 Breaker Panel Connection

In Chapter 3, the problem of transmitting through a breaker panel was mentioned. The problem is that the breaker panel represents high attenuation and slaves connected on different legs that branch off the breaker panel will not be able to transmit to each other. One solution is to have a 'multi-legged' master that runs a distinct copy of the protocol on each leg where there are slaves. The slaves on each leg communicate with slaves on other legs by bridging through the master, while the slaves on one leg communicate with each other without bridging through the master. The master must have a table that keeps track of which slaves are on which leg. The timeslot tables are independent between legs of the system, and to communicate between legs, the master will 're-route' a packet in one timeslot to another timeslot on the other leg if necessary (if a timeslot is GOOD on one leg but BAD on another). The master will need to have some number of buffers to perform this bridging operation. If the master's buffers are filling up, it will stop sending CTS's on the leg that is filling up buffers. In essence, the master takes over responsibility for getting data from a slave on one leg to a slave on another leg.

6.3.6 Using Consecutive Timeslots

If a sender finds that it is using consecutive timeslots, the ability to make longer packets that stretch over those timeslots should be explored. For example, if a sender is using timeslots 5 and 6 and making complete packets for each of these timeslots, it could make a single packet that is 2 timeslots long that begins in timeslot 5 and ends in timeslot 6. There will be only one header and one checksum for the packet resulting in less overhead per packet hence increasing the amount of data that can be sent. This concept is shown in Figure 6.1. To do this, the packet must have some indication that it is part of a longer packet, perhaps by a length field or a beginning, middle, and end of packet indication.

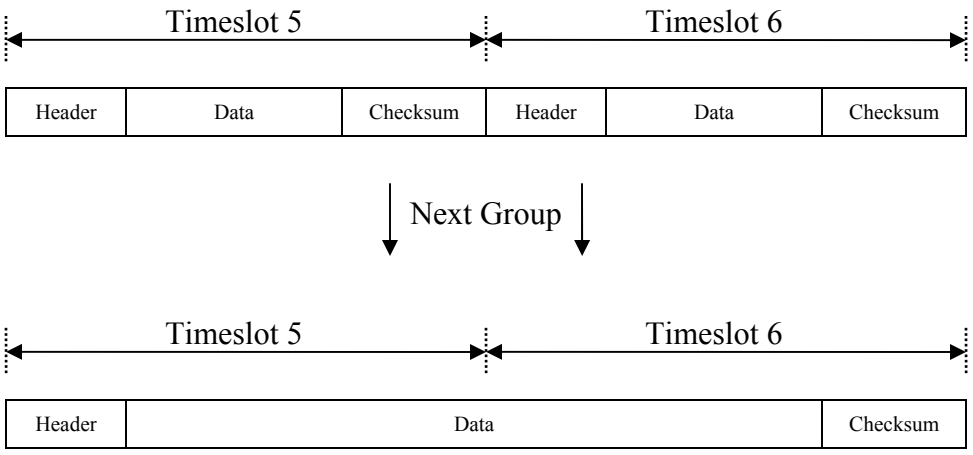


Figure 6.1 Concatenating timeslots.

One problem with this improvement is that if there is an error detected for the concatenated packet, it must be thrown away without knowledge of which timeslot the error occurred in. If there are a lot of errors occurring with concatenated packets, they sender may decide to reduce the packet size, and concatenate fewer timeslots. Only when there are no concatenated timeslots (each packet is only 1 timeslot long) can a node determine if there is one timeslot that is becoming bad. This process of breaking down a longer, concatenated packet into smaller packets and eventually zooming in on a single timeslot is the only way that a node can determine if a timeslot has become bad or not. Once it has narrowed down the timeslot that is bad and begins avoiding it, it may again begin concatenating timeslots to make longer, more efficient packets.

6.3.7 Other Protocol Considerations

Several variables in the protocol may be explored such as the optimum message length, the period of time between PINGs (master to slave request for information), and error correction.

This thesis did not select a specific message length; rather it explored the protocol performance over a range of message lengths. The message length chosen for data

communication may depend on the application. If only a small amount of data needs to be sent and it needs to be sent with low latency (highest probability of success) then small messages may be chosen. If high throughput is a goal then longer messages may be chosen. In this thesis, the longest message tested was 32 timeslots (two power line cycles). The length of messages should be up to the sending node, and can change depending on the type of data sent. Recall that the message length also has an effect on the threshold for considering a timeslot bad.

The period of time between PINGs was selected to be 10 seconds for this work, but this is another variable that was selected that may be optimized. 10 seconds was chosen because it is short enough to effectively communicate changes in the power line noise characteristic and answer demand from slaves to send data, yet it is long enough for nodes to check enough data to construct a correct timeslot table. If there are many active nodes in the system and their demand for the channel changes quickly, then a shorter PING period may be more suitable. However, if there are few active nodes sending little data then a longer PING period may be more suitable. Perhaps the PING period could be a variable that is tuned by the master depending on the channel activity. This is a small detail compared to other issues in the power line protocol, but is provided as thought for further improvement.

References

[1] Jack Hanson, M.Sc. thesis, University of Saskatchewan "Timed Power Line Data Transmission", December 1997.

[2] Niovi Pavlidou, A. J. Han Vinck, Javad Yazdani and Bahram Honary, "Power Line Communications: State of the Art and Future Trends", IEEE Communications Magazine, Vol.41 No. 4 pp. 34-39, April 2003.

[3] Lars Selander, "Power-Line Communications: Channel Properties and Communication Strategies", *M. Sc.* , Lund University, 1999.

[4] L.H. Charles Lee, *Error-Control Block Codes for Communications Engineers*. Norwood, MA: Arctech House, 2000.

[5] Glen Carty, *Broadband Networking*. Berkeley, CA: Osborne/McGraw-Hill, 2002.

[6] Transmission Theory for X10 Technology, http://www.x10.com/technology1.htm

[7] Introduction to the Londworks System, Version 1.0. Echelon Corporation, 1999.

[8] Echelon Corporation, *A Power Line Communication Tutorial – Challenges and Technologies.*

[9] Echelon Corporation, *PLT-22 Power Line Transceiver Data Sheet, 2001-2002.*

[10] Electronic Industies Alliance, *EIA-600.31 – Physical Layer & Medium Specification,* February 1995.

[11] M. K. Lee, R. E. Newman, H. A. Latchman, S. Katar and L. Yonge, "HomePlug 1.0 Powerline Communication LANs – Protocol Description and Performance Results version 5.4", *International Journal of Communication Systems, 2000; 00:1–6, [Version: 2002/09/18 v1.01],* John Wiley & Sons, Ltd. 2002.

[12] John Turnbull and Simon Garret, *Broadband Applications and the Digital Home*. London, UK: The Institution of Electrical Engineers, 2003

[13] M. K. Lee, R. E. Newman, H. A. Latchman, S. Katar and L. Yonge, "Field Performance Comparison of IEEE 802.11b and HomePlug 1.0", *Proceedings of the 27th Annual IEEE Conference on Local Computer Networks.* IEEE, 2002.

[14] J.R. Nicholson and J.A. Malack, "RF Impedance of Power Lines and Line Impedance Stabilization Networks in Conducted Interferences Measurement",

IEEE Transactions On Electromagnetic Compatibility, Vol. EMC 15, No 2 pp. 84-86, May. 1973.

[15] Youlu Zheng, Shakil Akhtar, *Networks for Computer Scientists and Engineers.* New York, NY: Oxford University Press, 2002.

[16] Dimitri Bertsekas, Robert Gallager, *Data Networks, Second Edition.* Englewood Cliffs, New Jersey: Prentice-Hall, 1992.

[17] Edgar H. Callaway Jr., *Wireless Sensor Networks – Architectures and Protocols.* Boca Raton, Florida: Auerbach Publications, 2004.

[18] Jim Kurose, Keith Ross, *Computer Networking: A Top-Down Approach Featuring the Internet.* Addison-Wesley Longman, 1999.

[19] Joseph L. Hammond, Peter J.P. O'Reilly, *Performance Analysis of Local Computer Networks.* Addison-Wesley Publishing Company, 1986.

[20] Andrew S. Tanenbaum, *Computer Networks, Second Edition.* Englewood Cliffs, New Jersey: Prentice-Hall, 1989.

[21] B. G. Evans, *Satellite Communication Systems, 3rd Edition.* London, UK: The Institution of Electrical Engineers, 1999.

[22] Gursharan S. Sidhu, Richard F. Andrews, Alan B. Oppenheimer, *Inside AppleTalk, Second Edition.* Addison-Wesley Publishing Company, 1990.

[23] Mark Norris, *Gigabit Ethernet Technology and Applications.* Norwood MA: Artech House, Inc., 2003.

[24] Federal Communications Commission, *Part 15 – Radio Frequency Devices, Sec. 15.113 – Power Line Carrier Systems.* 47CFR15.113, Title 47 Volume 1, October 2001.

[25] Industry Canada, *Low Power Licence-Exempt Radiocommunication Devices (All Frequency Bands).* RSS-210, Issue 5, November 2001.

[26] Motorola, *MC68360 Quad Integrated Communications Controller User's Manual.*

[27] The Slackware Linux Project, http://www.slackware.com

[28] On-Line Applications Research, RTEMS user's manual, http://www.rtems.com/onlinedocs/freesupport/manuals

[29] Hewlett Packard, *HP 33120A Function Generator/Arbitrary Waveform Generator User's Guide*. August 1997.

[30] The Kermit Project, University of Columbia, http://www.columbia.com/kermit/

[31] Francisco Javier Canete, Jose Antonio Cortes, Luis Diez, and Jose Tomas Entrambasaguas, "Modeling and Evaluation of the Indoor Power Line Transmission Medium", IEEE Communications Magazine, Vol. 41 No. 4 pp. 41-47, April 2003.

A. Hardware Pictures

Below are some pictures taken at the TRLabs technology forum. The forum was held in Saskatoon, SK on October 19 & 20, 2004.

Figure A.1 Complete Hardware Setup

In Figure A.1, the master is on the right, and slave on the left. They communicate through the power line by connecting to an extension cord that goes behind the table and connects to the power line. The oscilloscope in the middle displays the signal and noise on the line. In the background is a poster describing the setup and theory.

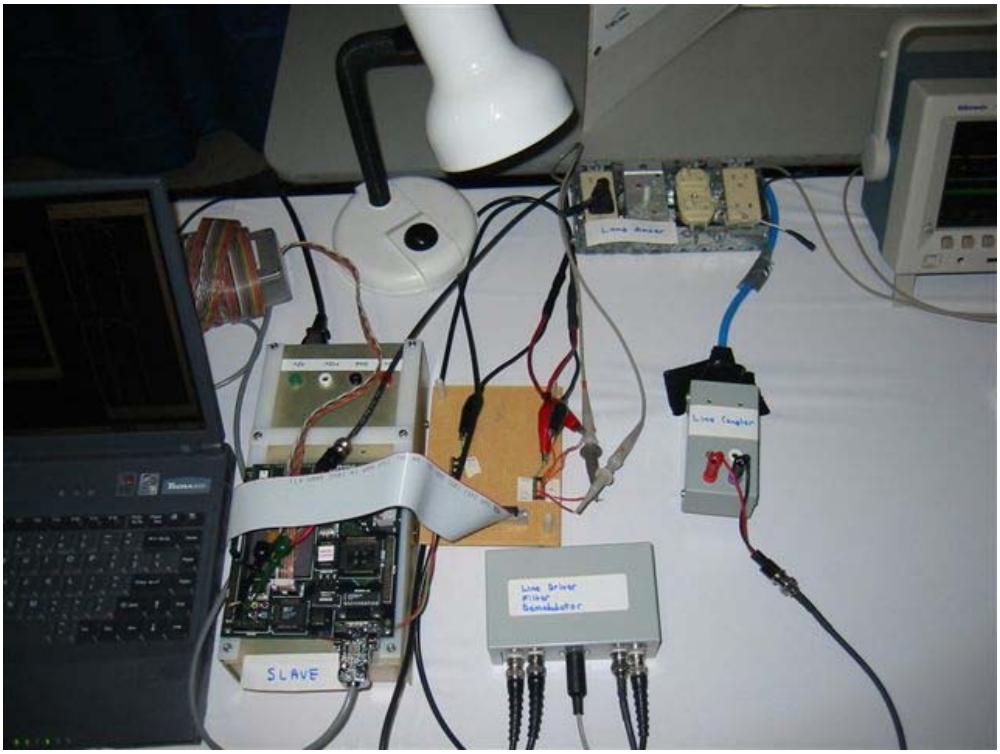


Figure A.2 Close-up of hardware setup

Figure A.2 is a close-up picture of the slave hardware. As described earlier in this chapter it consists of a PC (the laptop), MC68360 (the box immediately to the right of the laptop), a line coupler (small box furthest to the right) and all other electronics are inside the box closest to the bottom with five plugs connected to it.

Figure A.3 Close-up view of project boxes

Figure A.3 shows the line coupler boxes (bottom), the line driver/filter/demodulator boxes (top right) and a 60 Hz trigger box (left).

Figure A.4 Myself (left) and my supervisor, Prof. David Dodds (right).

B. Federal Communications Commission Part 15

The Federal Communications Commission part 15 is instrumental in determining appropriate power line communication transmission bandwidth and is included here as reference.

[Code of Federal Regulations]
[Title 47, Volume 1]
[Revised as of October 1, 2001]
From the U.S. Government Printing Office via GPO Access
[**CITE**: **47CFR15.113**]

[Page 691-692]

TITLE 47--TELECOMMUNICATION

CHAPTER I--FEDERAL COMMUNICATIONS COMMISSION

PART 15--RADIO FREQUENCY DEVICES--Table of Contents

Subpart B--Unintentional Radiators

Sec. 15.**113** Power line carrier systems.

Power line carrier systems, as defined in Sec. 15.3(t), are subject only to the following requirements:

(a) A power utility operating a power line carrier system shall submit the details of all existing systems plus any proposed new systems
or changes to existing systems to an industry-operated entity as set forth in Sec. 90.63(g) of this chapter. No notification to the FCC is required.

(b) The operating parameters of a power line carrier system (particularly the frequency) shall be selected to achieve the highest practical degree of compatibility with authorized or licensed users of the radio spectrum. The signals from this operation shall be contained within the frequency band 9 kHz to 490 kHz. A power line carrier system shall operate on an unprotected, non-interference basis in accordance with Sec. 15.5 of this part. If harmful interference occurs, the electric power utility shall discontinue use or adjust its power line carrier operation, as required, to remedy the interference. Particular attention should be paid to the possibility of interference to Loran C operations at 100 kHz.

(c) Power line carrier system apparatus shall be operated with the minimum power possible to accomplish the desired purpose. No equipment authorization is required.

(d) The best engineering principles shall be used in the generation of radio frequency currents by power line carrier systems to guard against harmful interference to authorized radio users, particularly on the fundamental and harmonic frequencies.

(e) Power line carrier system apparatus shall conform to such engineering standards as may be promulgated by the Commission. In addition, such systems should adhere to industry approved standards designed to enhance the use of power line carrier systems.

(f) The provisions of this section apply only to systems operated by a power utility for general supervision of

[[Page 692]]

the power system and do not permit operation on electric lines which connect the distribution substation to the customer or house wiring. Such operation can be conducted under the other provisions of this part.

[54 FR 17714, Apr. 25, 1989; 54 FR 32339, Aug. 7, 1989]

C. MC68360 Code

Code listings for the protocol implementation on the Motorola MC68360 are on the CD-ROM included with this thesis. All coding was done in the 'c' programming language. There are two sections of code: one for the master and one for the slave. The basic structure of each is the same:

1) makefile – file to make the compile and build of the code much easier.
2) uart.h – header file contains important constants.
3) init.c – contains initialization code. Initializes such things as the UART, any necessary interrupts, and timers.
4) transmit.c – contains code for the transmit operation. Decides what data to send and when, and sends bytes to the UART.
5) handlers.c – contains all the interrupt handlers in the system. These include external interrupt (60 Hz trigger), the timer interrupt (for the one millisecond timeslots), receive data interrupt (whenever a byte of data is received in the UART). Each handler routine contains code appropriate for handling the interrupt.

The major differences in the code between the master and slave are in the transmit.c and handlers.c programs. The transmit.c program in the master contains much more logic for sending appropriate data and control packets, while in the slave it contains logic only for sending ACK packets. The handlers.c program, on the other hand, is much more complex in the slave than it is in the master. In addition to the timer and external interrupts found in both the master and slave, the slave must check the integrity of every packet, what type it is, and keep track of timeslot errors. Other small differences exist in the master and slave programs uart.h and init.c but are mainly initializing different variables.

D. XR2211 Data Sheet

Following is the XR2211 data sheet. This chip was instrumental in the design as it provided the FSK demodulation function.

XR-2211

FSK Demodulator/
Tone Decoder

...the analog plus company™

June 1997-3

FEATURES

- Wide Frequency Range, 0.01Hz to 300kHz
- Wide Supply Voltage Range, 4.5V to 20V
- HCMOS/TTL/Logic Compatibility
- FSK Demodulation, with Carrier Detection
- Wide Dynamic Range, 10mV to 3V rms
- Adjustable Tracking Range, ±1% to 80%
- Excellent Temp. Stability, ±50ppm/°C, max.

APPLICATIONS

- Caller Identification Delivery
- FSK Demodulation
- Data Synchronization
- Tone Decoding
- FM Detection
- Carrier Detection

GENERAL DESCRIPTION

The XR-2211 is a monolithic phase-locked loop (PLL) system especially designed for data communications applications. It is particularly suited for FSK modem applications. It operates over a wide supply voltage range of 4.5 to 20V and a wide frequency range of 0.01Hz to 300kHz. It can accommodate analog signals between 10mV and 3V, and can interface with conventional DTL, TTL, and ECL logic families. The circuit consists of a basic PLL for tracking an input signal within the pass band, a quadrature phase detector which provides carrier detection, and an FSK voltage comparator which provides FSK demodulation. External components are used to independently set center frequency, bandwidth, and output delay. An internal voltage reference proportional to the power supply is provided at an output pin.

The XR-2211 is available in 14 pin packages specified for military and industrial temperature ranges.

ORDERING INFORMATION

Part No.	Package	Operating Temperature Range
XR-2211M	14 Pin CDIP (0.300”)	-55°C to +125°C
XR-2211N	14 Pin CDIP (0.300”)	-40°C to +85°C
XR-2211P	14 Pin PDIP (0.300”)	-40°C to +85°C
XR-2211ID	14 Lead SOIC (Jedec, 0.150”)	-40°C to +85°C

EXAR Corporation, 48720 Kato Road, Fremont, CA 94538 ◆ (510) 668-7000 ◆ FAX (510) 668-7017

BLOCK DIAGRAM

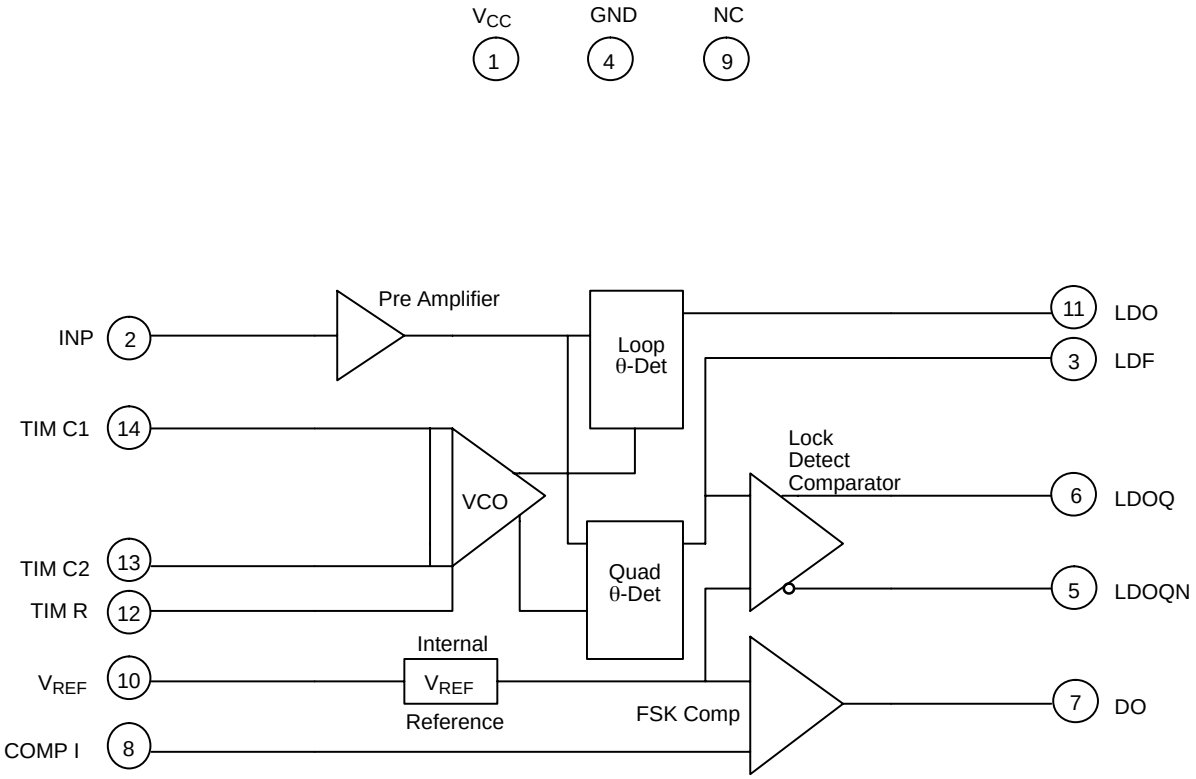
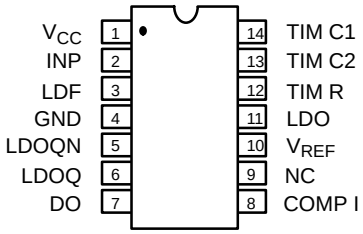
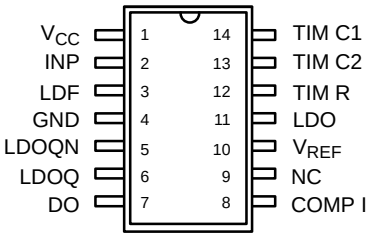


Figure 1. XR-2211 Block Diagram

PIN CONFIGURATION



14 Lead CDIP, PDIP (0.300")



14 Lead SOIC (Jedec, 0.150")

PIN DESCRIPTION

Pin #	Symbol	Type	Description
1	V_{CC}		**Positive Power Supply.**
2	INP	I	**Receive Analog Input.**
3	LDF	O	**Lock Detect Filter.**
4	GND		**Ground Pin.**
5	LDOQN	O	**Lock Detect Output Not.** This output will be low if the VCO is in the capture range.
6	LDOQ	O	**Lock Detect Output.** This output will be high if the VCO is in the capture range.
7	DO	O	**Data Output.** Decoded FSK output.
8	COMP I	I	**FSK Comparator Input.**
9	NC		**Not Connected.**
10	V_{REF}	O	**Internal Voltage Reference.** The value of V_{REF} is V_{CC}/2 - 650mV.
11	LDO	O	**Loop Detect Output.** This output provides the result of the quadrature phase detection.
12	TIM R	I	**Timing Resistor Input.** This pin connects to the timing resistor of the VCO.
13	TIM C2	I	**Timing Capacitor Input.** The timing capacitor connects between this pin and pin 14.
14	TIM C1	I	**Timing Capacitor Input.** The timing capacitor connects between this pin and pin 13.

ELECTRICAL CHARACTERISTICS

Test Conditions: V_{CC} = 12V, T_A = +25°C, R_O = 30KΩ, C_O = 0.033μF, unless otherwise specified.

Parameter	Min.	Typ.	Max.	Unit	Conditions
General					
Supply Voltage	**4.5**		**20**	V	
Supply Current		4	**7**	mA	$R_0 \geq$ 10KΩ. See *Figure 4.*
Oscillator Section					
Frequency Accuracy		±1	**±3**	%	Deviation from $f_O = 1/R_0 C_0$
Frequency Stability					
Temperature		±20	±50	ppm/°C	See *Figure 8.*
Power Supply		0.05	**0.5**	%/V	V_{CC} = 12 ±1V. See *Figure 7.*
		0.2		%/V	V_{CC} = ± 5V. See *Figure 7.*
Upper Frequency Limit	**100**	300		kHz	R_0 = 8.2KΩ, C_0 = 400pF
Lowest Practical Operating Frequency			0.01	Hz	R_0 = 2MΩ, C_0 = 50μF
Timing Resistor, R_0 - See *Figure 5*					
Operating Range	**5**		2000	KΩ	
Recommended Range	5			KΩ	See *Figure 7* and *Figure 8.*
Loop Phase Dectector Section					
Peak Output Current	**±150**	±200	**±300**	μA	Measured at Pin 11
Output Offset Current		1		μA	
Output Impedance		1		MΩ	
Maximum Swing	**±4**	± 5		V	Referenced to Pin 10
Quadrature Phase Detector					Measured at Pin 3
Peak Output Current	**100**	300		μA	
Output Impedance		1		MΩ	
Maximum Swing		11		V_{PP}	
Input Preampt Section					Measured at Pin 2
Input Impedance		20		KΩ	
Input Signal					
Voltage Required to Cause Limiting		2	10	mV rms	

Notes

Parameters are guaranteed over the recommended operating conditions, but are not 100% tested in production.

***Bold face parameters** are covered by production test and guaranteed over operating temperature range.*

DC ELECTRICAL CHARACTERISTICS (CONT'D)

Test Conditions: V_{CC} = 12V, T_A = +25°C, R_O = 30KΩ, C_O = 0.033μF, unless otherwise specified.

Parameter	Min.	Typ.	Max.	Unit	Conditions
Voltage Comparator Section					
Input Impedance		2		MΩ	Measured at Pins 3 and 8
Input Bias Current		100		nA	
Voltage Gain	55	70		dB	R_L = 5.1KΩ
Output Voltage Low		300	**500**	mV	I_C = 3mA
Output Leakage Current		0.01	**10**	μA	V_O = 20V
Internal Reference					
Voltage Level	**4.9**	5.3	**5.7**	V	Measured at Pin 10
Output Impedance		100		Ω	AC Small Signal
Maximum Source Current		80		μA	

Notes
Parameters are guaranteed over the recommended operating conditions, but are not 100% tested in production.
***Bold face parameters** are covered by production test and guaranteed over operating temperature range.*

Specifications are subject to change without notice

ABSOLUTE MAXIMUM RATINGS

Power Supply 20V
Input Signal Level ..... 3V rms
Power Dissipation 900mW

Package Power Dissipation Ratings
CDIP 750mW
  Derate Above T_A = 25°C ..... 8mW/°C
PDIP 800mW
  Derate Above T_A = 25°C ..... 60mW/°C
SOIC 390mW
  Derate Above T_A = 25°C ..... 5mW/°C

SYSTEM DESCRIPTION

The main PLL within the XR-2211 is constructed from an input preamplifier, analog multiplier used as a phase detector and a precision voltage controlled oscillator (VCO). The preamplifier is used as a limiter such that input signals above typically 10mV rms are amplified to a constant high level signal. The multiplying-type phase detector acts as a digital exclusive or gate. Its output (unfiltered) produces sum and difference frequencies of the input and the VCO output. The VCO is actually a current controlled oscillator with its normal input current (f_O) set by a resistor (R_0) to ground and its driving current with a resistor (R_1) from the phase detector.

The output of the phase detector produces sum and difference of the input and the VCO frequencies (internally connected). When in lock, these frequencies are f_{IN}+ f_{VCO} (2 times f_{IN} when in lock) and f_{IN} - f_{VCO} (0Hz when lock). By adding a capacitor to the phase detector output, the 2 times f_{IN} component is reduced, leaving a DC voltage that represents the phase difference between the two frequencies. This closes the loop and allows the VCO to track the input frequency.

The FSK comparator is used to determine if the VCO is driven above or below the center frequency (FSK comparator). This will produce both active high and active low outputs to indicate when the main PLL is in lock (quadrature phase detector and lock detector comparator).

PRINCIPLES OF OPERATION

Signal Input (Pin 2): Signal is AC coupled to this terminal. The internal impedance at pin 2 is 20KΩ. Recommended input signal level is in the range of 10mV rms to 3V rms.

Quadrature Phase Detector Output (Pin 3): This is the high impedance output of quadrature phase detector and is internally connected to the input of lock detect voltage comparator. In tone detection applications, pin 3 is connected to ground through a parallel combination of R_D and C_D (see *Figure 3*) to eliminate the chatter at lock detect outputs. If the tone detect section is not used, pin 3 can be left open.

Lock Detect Output, Q (Pin 6): The output at pin 6 is at "low" state when the PLL is out of lock and goes to "high" state when the PLL is locked. It is an open collector type output and requires a pull-up resistor, R_L, to V_{CC} for proper operation. At "low" state, it can sink up to 5mA of load current.

Lock Detect Complement, (Pin 5): The output at pin 5 is the logic complement of the lock detect output at pin 6. This output is also an open collector type stage which can sink 5mA of load current at low or "on" state.

FSK Data Output (Pin 7): This output is an open collector logic stage which requires a pull-up resistor, R_L, to V_{CC} for proper operation. It can sink 5mA of load current. When decoding FSK signals, FSK data output is at "high" or "off" state for low input frequency, and at "low" or "on" state for high input frequency. If no input signal is present, the logic state at pin 7 is indeterminate.

FSK Comparator Input (Pin 8): This is the high impedance input to the FSK voltage comparator. Normally, an FSK post-detection or data filter is connected between this terminal and the PLL phase detector output (pin 11). This data filter is formed by R_F and C_F (see *Figure 3.*) The threshold voltage of the comparator is set by the internal reference voltage, V_{REF}, available at pin 10.

Reference Voltage, V_{REF} (Pin 10): This pin is internally biased at the reference voltage level, V_{REF}: $V_{REF} = V_{CC}/2 - 650mV$. The DC voltage level at this pin forms an internal reference for the voltage levels at pins 5, 8, 11 and 12. Pin 10 must be bypassed to ground with a 0.1μF capacitor for proper operation of the circuit.

Loop Phase Detector Output (Pin 11): This terminal provides a high impedance output for the loop phase detector. The PLL loop filter is formed by R_1 and C_1 connected to pin 11 (see *Figure 3.*) With no input signal, or with no phase error within the PLL, the DC level at pin 11 is very nearly equal to V_{REF}. The peak to peak voltage swing available at the phase detector output is equal to $2 \times V_{REF}$.

VCO Control Input (Pin 12): VCO free-running frequency is determined by external timing resistor, R_0, connected from this terminal to ground. The VCO free-running frequency, f_O, is:

$$f_O = \frac{1}{R_0 \cdot C_0} \; Hz$$

where C_0 is the timing capacitor across pins 13 and 14. For optimum temperature stability, R_0 must be in the range of 10KΩ to 100KΩ (see *Figure 9.*)

This terminal is a low impedance point, and is internally biased at a DC level equal to V_{REF}. The maximum timing current drawn from pin 12 must be limited to ≤ 3mA for proper operation of the circuit.

VCO Timing Capacitor (Pins 13 and 14): VCO frequency is inversely proportional to the external timing capacitor, C_0, connected across these terminals (see *Figure 6.*) C_0 must be non-polar, and in the range of 200pF to 10μF.

VCO Frequency Adjustment: VCO can be fine-tuned by connecting a potentiometer, R_X, in series with R_0 at pin 12 (see *Figure 10.*)

VCO Free-Running Frequency, f_O: XR-2211 does not have a separate VCO output terminal. Instead, the VCO outputs are internally connected to the phase detector sections of the circuit. For set-up or adjustment purposes, the VCO free-running frequency can be tuned by using the generalized circuit in *Figure 3*, and applying an alternating bit pattern of O's and 1's at the known mark and space frequencies. By adjusting R_0, the VCO can then be tuned to obtain a 50% duty cycle on the FSK output (pin 7). This will ensure that the VCO f_O value is accurately referenced to the mark and space frequencies.

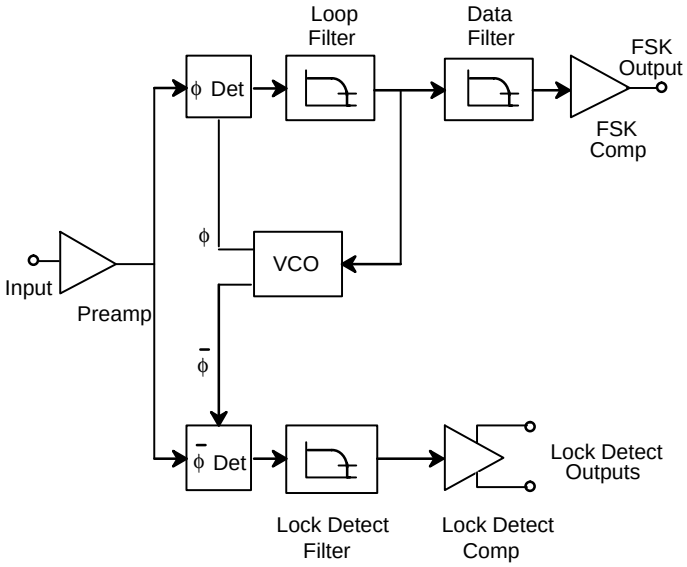


Figure 2. Functional Block Diagram of a Tone and FSK Decoding System Using XR-2211

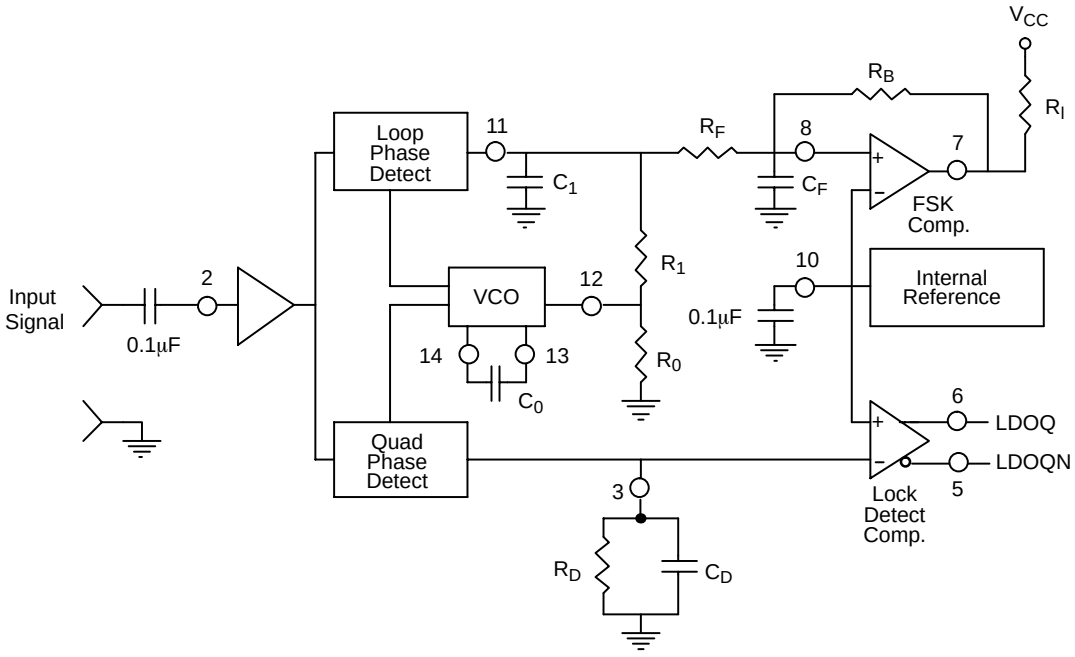


Figure 3. Generalized Circuit Connection for FSK and Tone Detection

DESIGN EQUATIONS

(All resistance in Ω, all frequency in Hz and all capacitance in farads, unless otherwise specified)

(See *Figure 3* for definition of components)

1. VCO Center Frequency, f_O:

$$f_O = \frac{1}{R_0 \cdot C_0}$$

2. Internal Reference Voltage, V_{REF} (measured at pin 10):

$$V_{REF} = \left(\frac{V_{CC}}{2}\right) - 650mV \text{ in volts}$$

3. Loop Low-Pass Filter Time Constant, τ:

$$\tau = C_1 \cdot R_{PP} \text{ (seconds)}$$

where:

$$R_{PP} = \left(\frac{R_1 \cdot R_F}{R_1 + R_F}\right)$$

if R_F is ∞ or C_F reactance is ∞, then R_{PP} = R1

4. Loop Damping, ς:

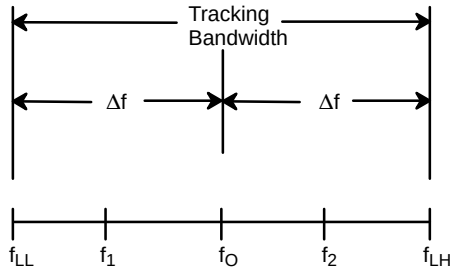
$$\zeta = \sqrt{\left(\frac{1250 \cdot C_0}{R_1 \cdot C_1}\right)}$$

Note: *For derivation/explanation of this equation, please see TAN-011.*

5. Loop-tracking

bandwidth, $\pm = \frac{\Delta f}{f_0}$

$$\frac{\Delta f}{f_0} = \frac{R_0}{R_1}$$



6. FSK Data filter time constant, tF:

$$\tau_F = \frac{R_B \cdot R_F}{(R_B + R_F)} \cdot C_F \ (seconds)$$

7. Loop phase detector conversion gain, Kd: (Kd is the differential DC voltage across pin 10 and pin11, per unit of phase error at phase detector input):

$$K_d = \frac{V_{REF} \cdot R_1}{10,000 \cdot \pi} \left[\frac{volt}{radian}\right]$$

Note: *For derivation/explanation of this equation, please see TAN-011.*

8. VCO conversion gain, Ko: (Ko is the amount of change in VCO frequency, per unit of DC voltage change at pin 11):

$$K_0 = \frac{-2\pi}{V_{REF} \cdot C_0 \cdot R_1} = \left(\frac{radian/second}{volt}\right)$$

9. The filter transfer function:

$$F(s) = \frac{1}{1 + SR_1 \cdot C_1} \ at\ 0\ Hz. \qquad S = J\omega \ and \ \omega = 0$$

10. Total loop gain. K_T:

$$K_T = K_O \cdot K_d \cdot F(s) = \left(\frac{R_F}{5,000 \cdot C_0 \cdot (R_1 + R_F)}\right)\left[\frac{1}{seconds}\right]$$

11. Peak detector current I_A:

$$I_A = \frac{V_{REF}}{20,000} \ (V_{REF} \ in \ volts \ and \ I_A \ in \ amps)$$

Note: *For derivation/explanation of this equation, please see TAN-011.*

APPLICATIONS INFORMATION

FSK Decoding

Figure 10 shows the basic circuit connection for FSK decoding. With reference to *Figure 3* and *Figure 10*, the functions of external components are defined as follows: R_0 and C_0 set the PLL center frequency, R_1 sets the system bandwidth, and C_1 sets the loop filter time constant and the loop damping factor. C_F and R_F form a one-pole post-detection filter for the FSK data output. The resistor R_B from pin 7 to pin 8 introduces positive feedback across the FSK comparator to facilitate rapid transition between output logic states.

Design Instructions:

The circuit of *Figure 10* can be tailored for any FSK decoding application by the choice of five key circuit components: R_0, R_1, C_0, C_1 and C_F. For a given set of FSK mark and space frequencies, f_O and f_1, these parameters can be calculated as follows:

(All resistance in Ω's, all frequency in Hz and all capacitance in farads, unless otherwise specified)

a) Calculate PLL center frequency, f_O:

$$f_O = \sqrt{F_1 \cdot F_2}$$

b) Choose value of timing resistor R_0, to be in the range of 10KΩ to 100KΩ. This choice is arbitrary. The recommended value is R_0 = 20KΩ. The final value of R_0 is normally fine-tuned with the series potentiometer, R_X.

$$R_O = R_O + \frac{R_X}{2}$$

c) Calculate value of C0 from design equation (1) or from *Figure 7*:

$$C_O = \frac{1}{R_0 \cdot f_0}$$

d) Calculate R1 to give the desired tracking bandwidth (See design equation 5).

$$R_1 = \frac{R_0 \cdot f_0}{(f_1 - f_2)} \cdot 2$$

e) Calculate C1 to set loop damping. (See design equation 4):

Normally, ς = 0.5 is recommended.

$$C_1 = \frac{1250 \cdot C_0}{R_1 \cdot \varsigma^2}$$

f) The input to the XR-2211 may sometimes be too sensitive to noise conditions on the input line. *Figure 4* illustrates a method of de-sensitizing the XR-2211 from such noisy line conditions by the use of a resistor, Rx, connected from pin 2 to ground. The value of Rx is chosen by the equation and the desired minimum signal threshold level.

$$V_{IN}\ minimum\ (peak) = V_a - V_b = \Delta V \pm 2.8mV\ offset = V_{REF}\frac{20,000}{(20,000 + R_X)}\ or\ R_X = 20,000\left(\frac{V_{REF}}{\Delta V} - 1\right)$$

V_{IN} minimum (peak) input voltage must exceed this value to be detected (equivalent to adjusting V threshold)

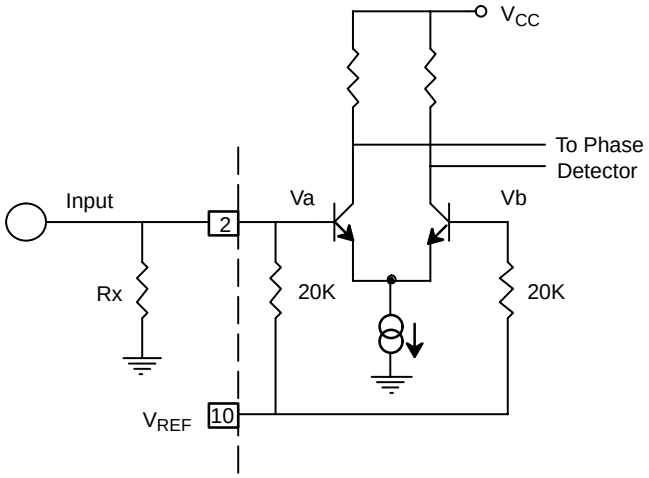


Figure 4. Desensitizing Input Stage

g) Calculate Data Filter Capacitance, C_F:

$$R_{sum} = \frac{(R_F + R_1)\cdot R_B}{(R_1 + R_F + R_B)}$$

$$C_F = \frac{0.25}{(R_{sum}\cdot Baud\ Rate)} \qquad Baud\ rate\ in\ \frac{1}{seconds}$$

Note: *All values except R_0 can be rounded to nearest standard value.*

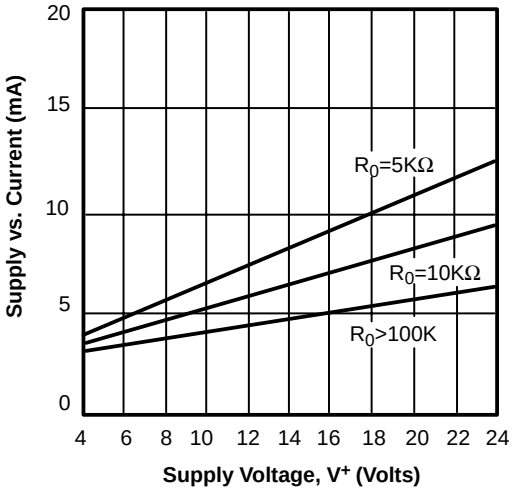


Figure 5. Typical Supply Current vs. V+ (Logic Outputs Open Circuited)

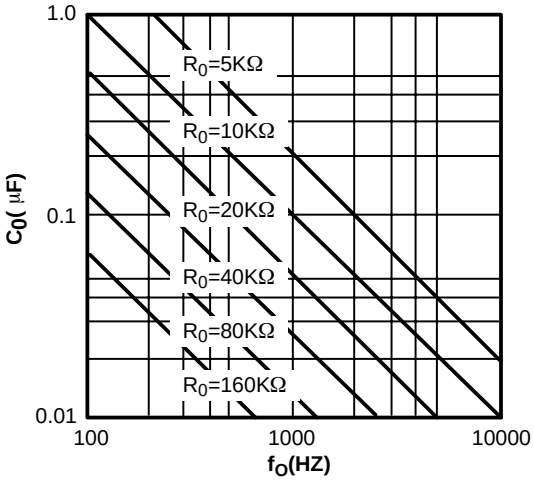


Figure 6. VCO Frequency vs. Timing Resistor

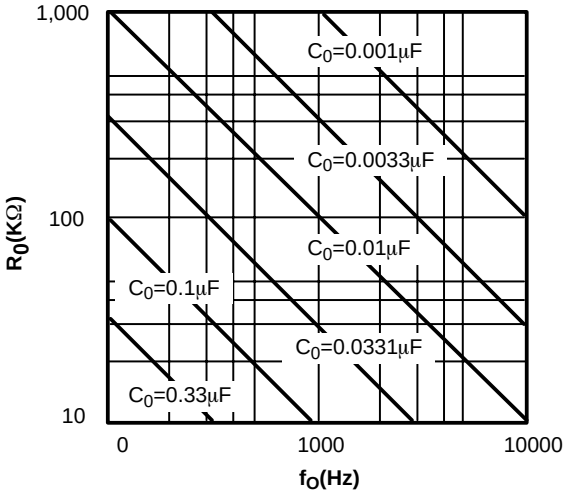


Figure 7. VCO Frequency vs. Timing Capacitor

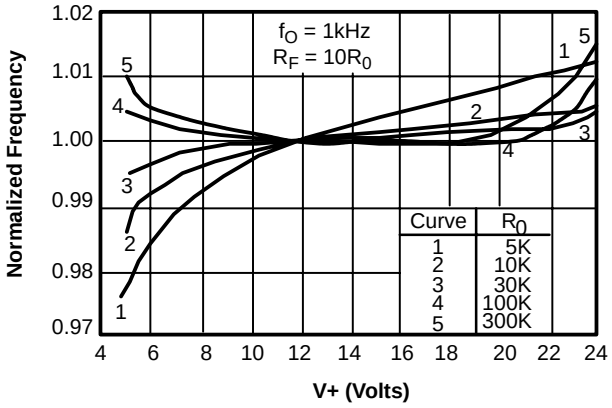


Figure 8. Typical f_O vs. Power Supply Characteristics

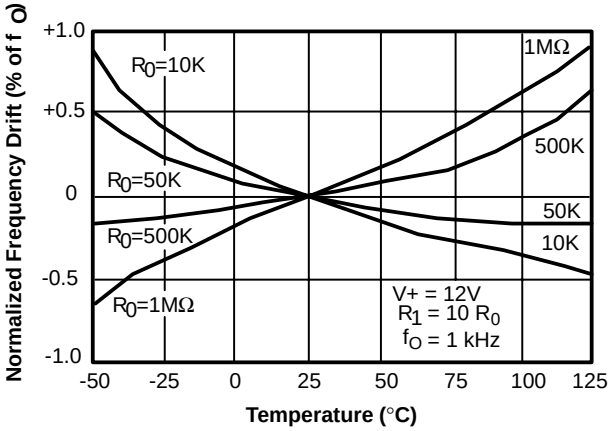


Figure 9. Typical Center Frequency Drift vs. Temperature

Design Example:

1200 Baud FSK demodulator with mark and space frequencies of 1200/2200.

Step 1: Calculate f_O: from design instructions

(*a*) $f_O = \sqrt{1200 \cdot 2200} = 1624$

Step 2: Calculate R_0 : R_0 =10K with a potentiometer of 10K. (See design instructions (b))

(*b*) $R_T = 10 + \left(\frac{10}{2}\right) = 15K$

Step 3: Calculate C_0 from design instructions

(*c*) $C_O = \frac{1}{15000 \cdot 1624} = 39nF$

Step 4: Calculate R_1 : from design instructions

(*d*) $R_1 = \frac{20000 \cdot 1624 \cdot 2}{(2200 - 1200)} = 51,000$

Step 5: Calculate C_1 : from design instructions

(*e*) $C_1 = \frac{1250 \cdot 39nF}{51000 \cdot 0.5^2} = 3.9nF$

Step 6: Calculate R_F : R_F should be at least five times R_1, R_F = 51,000·5 = 255 KΩ

Step 7: Calculate R_B : R_B should be at least five times R_F, R_B = 255,000·5 = 1.2 MΩ

Step 8: Calculate R_{SUM} :

$$R_{SUM} = \frac{(R_F + R_1) \cdot R_B}{(R_F + R_1 + R_B)} = 240K\Omega$$

Step 9: Calculate C_F :

$$C_F = \frac{0.25}{(R_{SUM} \cdot Baud\ Rate)} = 1nF$$

Note: *All values except R_0 can be rounded to nearest standard value.*

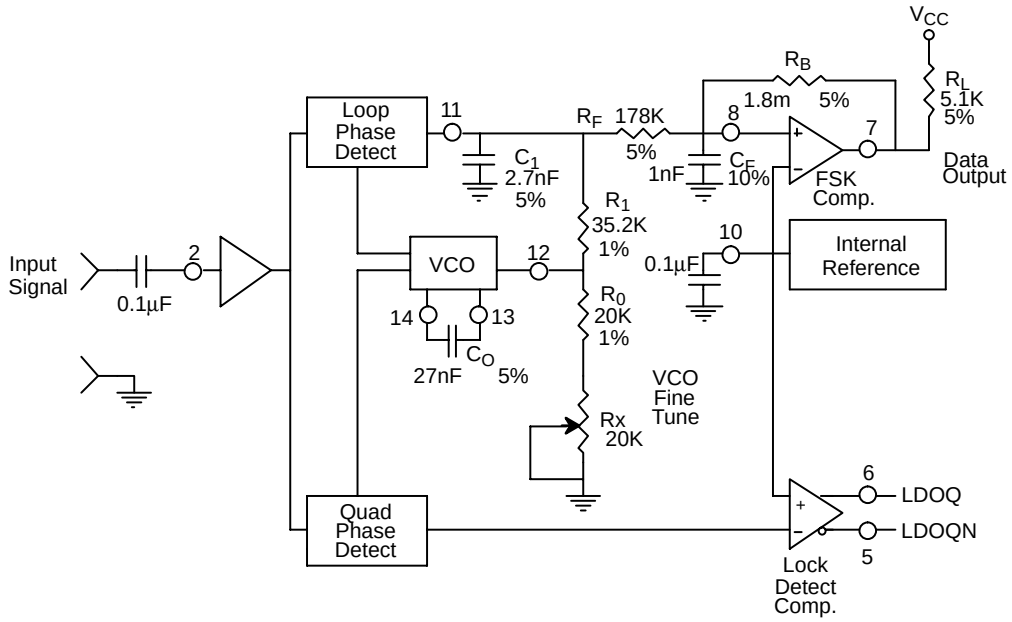


Figure 10. Circuit Connection for FSK Decoding of Caller Identification Signals (Bell 202 Format)

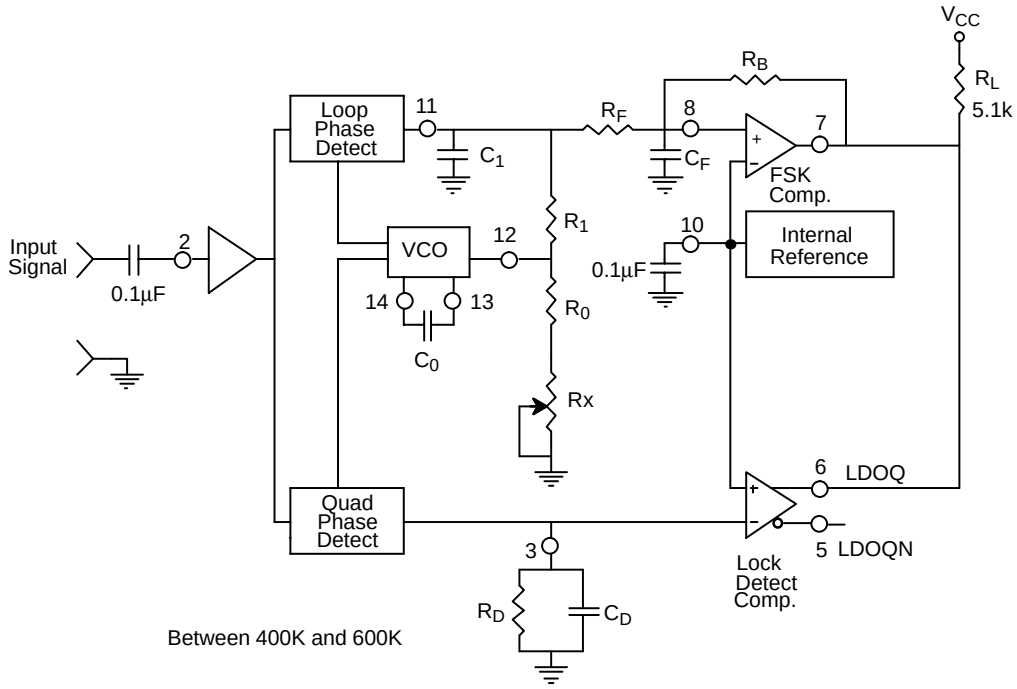


Figure 11. External Connectors for FSK Demodulation with Carrier Detect Capability

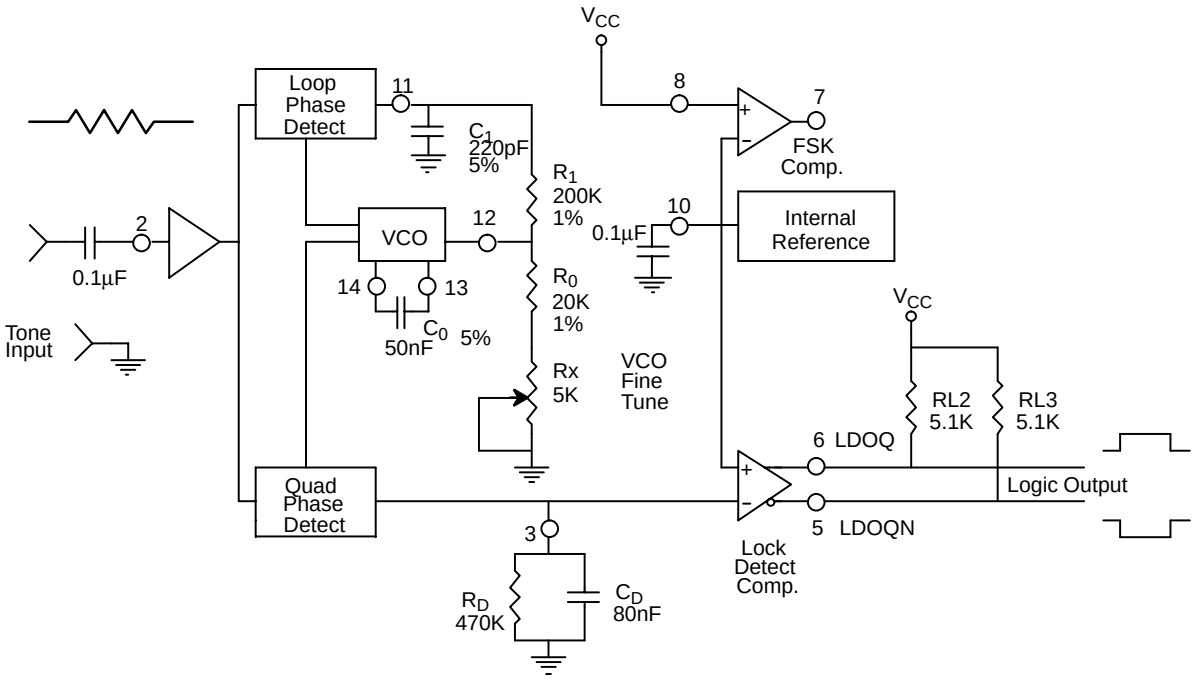


Figure 12. Circuit Connection for Tone Detection

FSK Decoding with Carrier Detect

The lock detect section of XR-2211 can be used as a carrier detect option for FSK decoding. The recommended circuit connection for this application is shown in *Figure 11.* The open collector lock detect output, pin 6, is shorted to data output (pin 7). Thus, data output will be disabled at "low" state, until there is a carrier within the detection band of the PLL and the pin 6 output goes "high" to enable the data output.

Note: *Data Output is "Low" When No Carrier is Present.*

The minimum value of the lock detect filter capacitance C_D is inversely proportional to the capture range, $\pm\Delta fc$. This is the range of incoming frequencies over which the loop can acquire lock and is always less than the tracking range. It is further limited by C_1. For most applications, $\Delta fc > \Delta f/2$. For $R_D = 470K\Omega$, the approximate minimum value of C_D can be determined by:

$$C_D > \frac{16}{\Delta f} \quad C \text{ in } \mu F \text{ and } f \text{ in Hz.}$$

C in µF and f in Hz.

With values of C_D that are too small, chatter can be observed on the lock detect output as an incoming signal frequency approaches the capture bandwidth. Excessively large values of C_D will slow the response time of the lock detect output. For Caller I.D. applications choose $C_D = 0.1\mu F$.

Tone Detection

Figure 12 shows the generalized circuit connection for tone detection. The logic outputs, LDOQN and LDOQ at pins 5 and 6 are normally at "high" and "low" logic states, respectively. When a tone is present within the detection band of the PLL, the logic state at these outputs become reversed for the duration of the input tone. Each logic output can sink 5mA of load current.

Both outputs at pins 5 and 6 are open collector type stages, and require external pull-up resistors R_{L2} and R_{L3}, as shown in *Figure 12.*

With reference to *Figure 3* and *Figure 12*, the functions of the external circuit components can be explained as follows: R_0 and C_0 set VCO center frequency; R_1 sets the detection bandwidth; C_1 sets the low pass-loop filter time constant and the loop damping factor.

Design Instructions:

The circuit of *Figure 12* can be optimized for any tone detection application by the choice of the 5 key circuit components: R_0, R_1, C_0, C_1 and C_D. For a given input, the tone frequency, f_S, these parameters are calculated as follows:

(All resistance in Ω's, all frequency in Hz and all capacitance in farads, unless otherwise specified)

a) Choose value of timing resistor R_0 to be in the range of 10KΩ to 50KΩ. This choice is dictated by the max./min. current that the internal voltage reference can deliver. The recommended value is R_0 = 20KΩ. The final value of R_0 is normally fine-tuned with the series potentiometer, R_X.

b) Calculate value of C_0 from design equation (1) or from *Figure 7* $f_S = f_O$:

$$C_O = \frac{1}{R_0 \cdot fs}$$

c) Calculate R_1 to set the bandwidth $\pm\Delta f$ (See design equation 5):

$$R_1 = \frac{R_0 \cdot f_0 \cdot 2}{\Delta f}$$

Note: *The total detection bandwidth covers the frequency range of* $f_O \pm \Delta f$

d) Calculate value of C_1 for a given loop damping factor:

Normally, $\varsigma = 0.5$ is recommended.

$$C_1 = \frac{1250 \cdot C_0}{R_1 \cdot \varsigma^2}$$

Increasing C_1 improves the out-of-band signal rejection, but increases the PLL capture time.

e) Calculate value of the filter capacitor C_D . To avoid chatter at the logic output, with R_D = 470KΩ, C_D must be:

$$C_D > \frac{16}{\Delta f} \quad C \text{ in } \mu F$$

Increasing C_D slows down the logic output response time.

Design Examples:

Tone detector with a detection band of ± 100Hz:

a) Choose value of timing resistor R_0 to be in the range of 10KΩ to 50KΩ. This choice is dictated by the max./min. current that the internal voltage reference can deliver. The recommended value is R_0 = 20 KΩ. The final value of R_0 is normally fine-tuned with the series potentiometer, R_X.

b) Calculate value of C_0 from design equation (1) or from *Figure 6* $f_S = f_O$:

$$C_0 = \frac{1}{R_0 \cdot f_S} = \frac{1}{20,000 \cdot 1,000} = 50nF$$

c) Calculate R_1 to set the bandwidth $\pm\Delta f$ (See design equation 5):

$$R_1 = \frac{R_0 \cdot f_O \cdot 2}{\Delta f} = \frac{20,000 \cdot 1,000 \cdot 2}{100} = 400K$$

Note: *The total detection bandwidth covers the frequency range of $f_O \pm \Delta f$*

d) Calculate value of C_0 for a given loop damping factor:

Normally, $\varsigma = 0.5$ is recommended.

$$C_1 = \frac{1250 \cdot C_0}{R_1 \cdot \varsigma^2} = \frac{1250 \cdot 50 \cdot 10^{-9}}{400,000 \cdot 0.5^2} = 6.25pF$$

Increasing C_1 improves the out-of-band signal rejection, but increases the PLL capture time.

e) Calculate value of the filter capacitor C_D . To avoid chatter at the logic output, with R_D = 470KΩ, C_D must be:

$$C_D = \frac{16}{\Delta f} \geq \frac{16}{200} \geq 80nF$$

Increasing C_D slows down the logic output response time.

f) Fine tune center frequency with 5KΩ potentiometer, R_X.

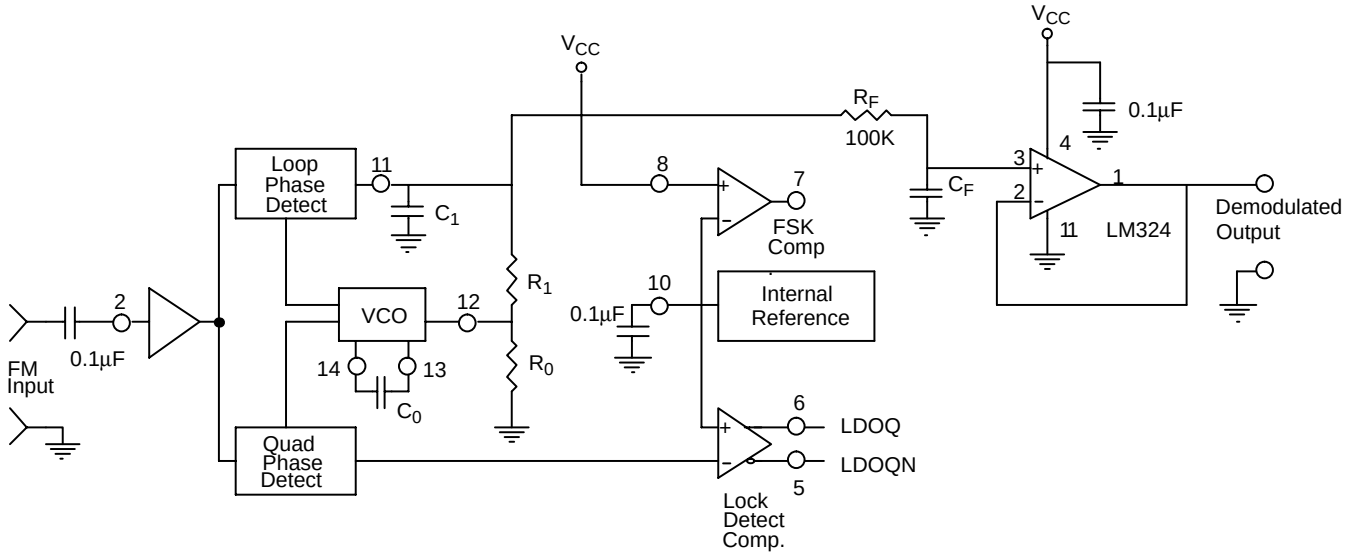


Figure 13. Linear FM Detector Using XR-2211 and an External Op Amp.
(See Section on Design Equation for Component Values.)

Linear FM Detection

XR-2211 can be used as a linear FM detector for a wide range of analog communications and telemetry applications. The recommended circuit connection for this application is shown in *Figure 13.* The demodulated output is taken from the loop phase detector output (pin 11), through a post-detection filter made up of R_F and C_F, and an external buffer amplifier. This buffer amplifier is necessary because of the high impedance output at pin 11. Normally, a non-inverting unity gain op amp can be used as a buffer amplifier, as shown in *Figure 13.*

The FM detector gain, i.e., the output voltage change per unit of FM deviation can be given as:

$$V_{OUT} = \frac{R_1 \cdot V_{REF}}{100 \cdot R_0}$$

where VR is the internal reference voltage ($V_{REF} = V_{CC}/2$ - 650mV). For the choice of external components R_1, R_0, C_D, C_1 and C_F, see the section on design equations.

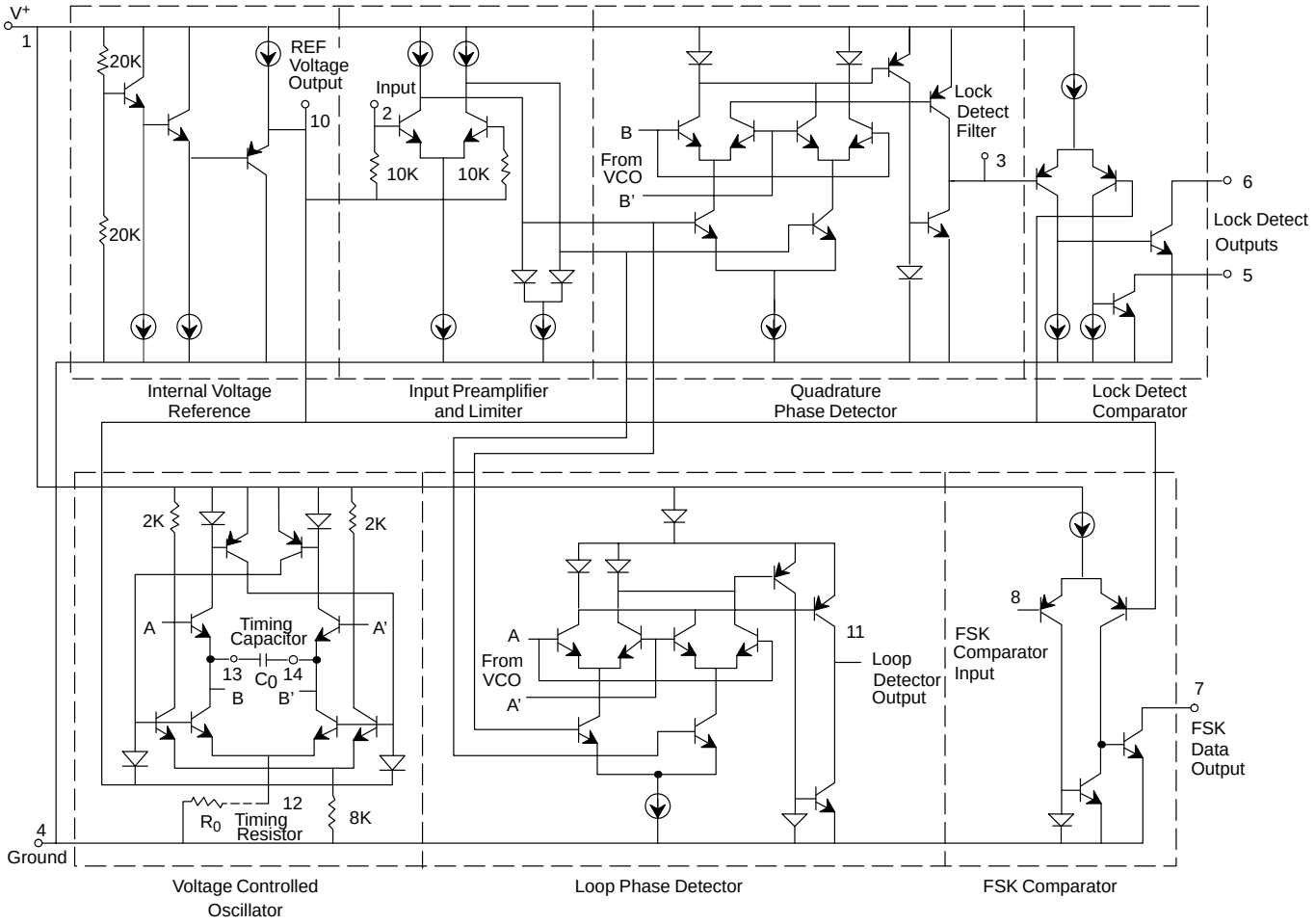
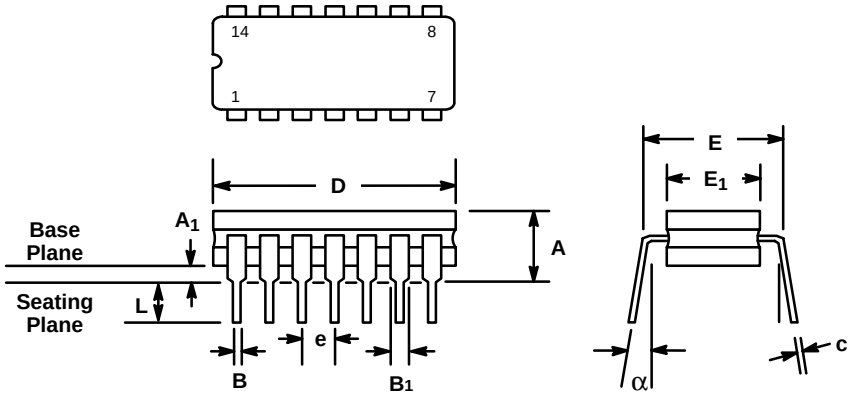


Figure 14. Equivalent Schematic Diagram

14 LEAD CERAMIC DUAL-IN-LINE (300 MIL CDIP)

Rev. 1.00

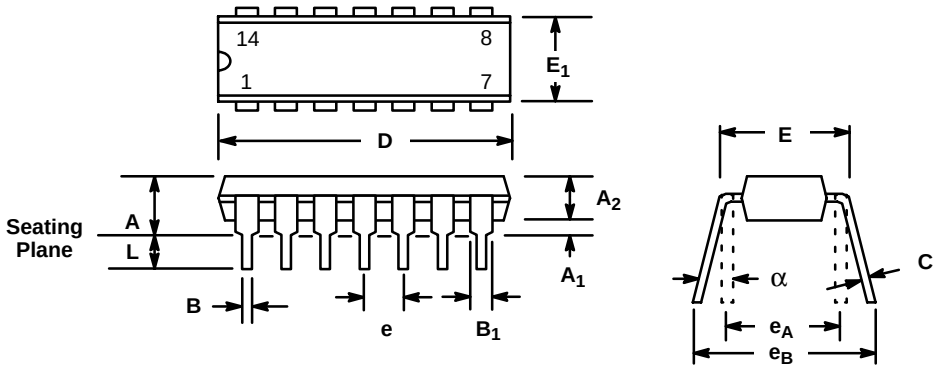


SYMBOL	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.100	0.200	2.54	5.08
A_1	0.015	0.060	0.38	1.52
B	0.014	0.026	0.36	0.66
B_1	0.045	0.065	1.14	1.65
c	0.008	0.018	0.20	0.46
D	0.685	0.785	17.40	19.94
E_1	0.250	0.310	6.35	7.87
E	0.300 BSC		7.62 BSC	
e	0.100 BSC		2.54 BSC	
L	0.125	0.200	3.18	5.08
α	0°	15°	0°	15°

Note: The control dimension is the inch column

14 LEAD PLASTIC DUAL-IN-LINE (300 MIL PDIP)

Rev. 1.00

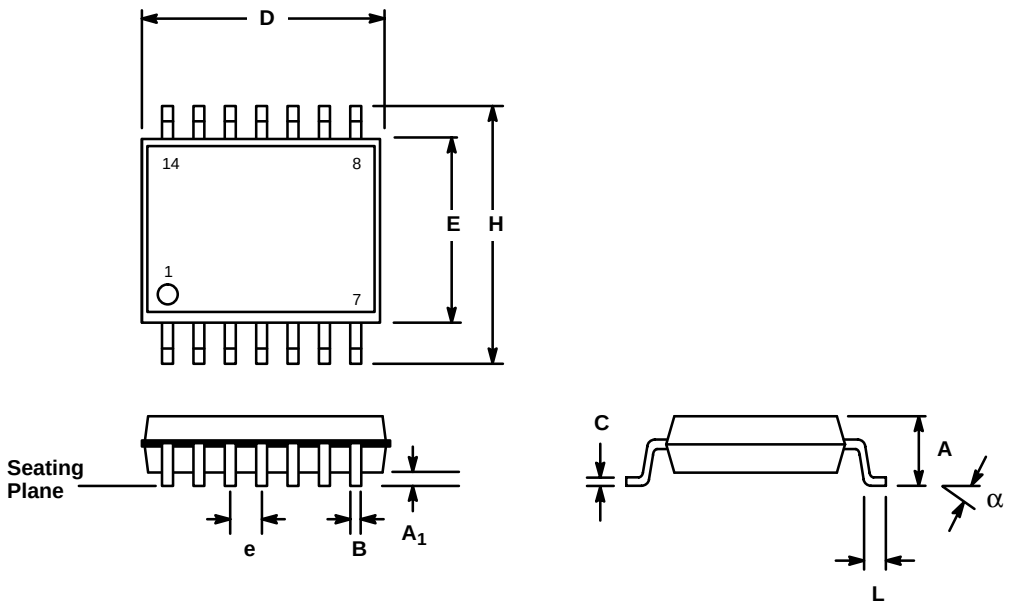


SYMBOL	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.145	0.210	3.68	5.33
A_1	0.015	0.070	0.38	1.78
A_2	0.115	0.195	2.92	4.95
B	0.014	0.024	0.36	0.56
B_1	0.030	0.070	0.76	1.78
C	0.008	0.014	0.20	0.38
D	0.725	0.795	18.42	20.19
E	0.300	0.325	7.62	8.26
E_1	0.240	0.280	6.10	7.11
e	0.100 BSC		2.54 BSC	
e_A	0.300 BSC		7.62 BSC	
e_B	0.310	0.430	7.87	10.92
L	0.115	0.160	2.92	4.06
α	0°	15°	0°	15°

Note: The control dimension is the inch column

14 LEAD SMALL OUTLINE
(150 MIL JEDEC SOIC)

Rev. 1.00



SYMBOL	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.053	0.069	1.35	1.75
A_1	0.004	0.010	0.10	0.25
B	0.013	0.020	0.33	0.51
C	0.007	0.010	0.19	0.25
D	0.337	0.344	8.55	8.75
E	0.150	0.157	3.80	4.00
e	0.050 BSC		1.27 BSC	
H	0.228	0.244	5.80	6.20
L	0.016	0.050	0.40	1.27
α	0°	8°	0°	8°

Note: The control dimension is the millimeter column

Notes

Notes

NOTICE

EXAR Corporation reserves the right to make changes to the products contained in this publication in order to improve design, performance or reliability. EXAR Corporation assumes no responsibility for the use of any circuits described herein, conveys no license under any patent or other right, and makes no representation that the circuits are free of patent infringement. Charts and schedules contained here in are only for illustration purposes and may vary depending upon a user's specific application. While the information in this publication has been carefully checked; no responsibility, however, is assumed for inaccuracies.

EXAR Corporation does not recommend the use of any of its products in life support applications where the failure or malfunction of the product can reasonably be expected to cause failure of the life support system or to significantly affect its safety or effectiveness. Products are not authorized for use in such applications unless EXAR Corporation receives, in writing, assurances to its satisfaction that: (a) the risk of injury or damage has been minimized; (b) the user assumes all such risks; (c) potential liability of EXAR Corporation is adequately protected under the circumstances.

Datasheet June 1997