DATA TRANSMISSION OVER VIRTUAL TELEPHONE LINE

BY

UJJWAL VERMA-041037

HITESH VATSYAYAN-041045

KUMAR ROHIT-041046

MAY-2008

Submitted in partial fulfillment of the Degree of Bachelor of Technology

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY-WAKNAGHAT

CERTIFICATE

This is to certify that the work entitled, "Data Transmission over virtual telephone line" submitted Ujjwal Verma, Hitesh Vatsyayan and Kumar Rohit in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering in 2008 of Jaypee University of Information Technology has been carried out under our supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Prof. Sunil Bhooshan

Mr. Vinay Kumar

ACKNOWLEDGEMENT

We would like to express our sincere thanks and gratitude to **Prof. Sunil Bhooshan** and **Mr. Vinay Kumar,** Department of Electronics and Communication Engineering, under whose able guidance we managed to give our endeavor the desired shape, their help, stimulating suggestions and constant encouragement helped us at every phase of the project. Their enthusiasm and the view of producing quality substance has casted a deep and a long lasting impression on us.

We thank all our colleagues for their constant support and encouragement. Their unobtrusive support and suggestions bolstered our confidence as usual. Their inspiring words will be a guiding force in all our endeavors to attain greater heights.

Ujjwal Verma | Hitesh Vatsyayan | Kumar Rohit

TABLE OF CONTENT

Title Page....1

Certificate....2

Acknowledgement....3

List of Figures and Tables....6

List of Abbreviations....7

CHAPTER I

<u>Virtual Telephone Line</u>....9

- **<u>Introduction</u>**
- **<u>Circuit diagram</u>**
- **<u>Component list</u>**
- **<u>Functional description</u>**

<u>Testing and Fault Diagnosis</u>....16

- **<u>Functional test</u>**
- **<u>Diagnostic test</u>**

CHAPTER II

Wave File Format..21

File Structure

Wave File Header - RIFF Type Chunk

Wave File Chunks

Format Chunk - "fmt "

Data Chunk - "data"

Final Layout of the Wave Format

Codes for format conversion..33

Advantages..34

Operational limitations..35

Bibliography..36

LIST OF FIGURES AND TABLES

Figure 1:	Circuit diagram for line simulator
Figure 2:	Snapshots
Figure 3:	WAV file format
Figure 4:	Basic layout
Table 1:	Component list for line simulator
Table 2:	Frequencies and time delays
Table 3:	RIFF chunk format
Table 4:	Wav format chunk values
Table 5:	Common WAV compression codes
Table 6:	Data chunk format
Table 7:	Interlaced stereo wav samples

LIST OF ABBREVIATIONS

1. RIFF - Resource Interchange File Format
2. GPRS - General Packet Radio Service
3. LAN - Local Area Network
4. DPDT - Double Pole Double Throw
5. DIP - Dual Inline Package
6. ID - Information Data
7. PCM - Pulse Code Modulation
8. ADPCM - Adaptive Delta Pulse Code Modulation
9. GSM - Global System for Mobile Communication
10. MPEG - Moving Picture Experts Group

ABSTRACT

The project "Data transmission over Virtual Telephone Line" is a part of our B. Tech curriculum at Jaypee University of Information Technology, Solan. Several methods for exchange of files are available between two computers. These includes internet, GPRS, LAN, etc. We are going to suggest a new technique of data transfer which is more effective as compared to the present available techniques. In this method we convert the file to be transmitted to an audio file (*.wav). The file can be of any format. The converted audio file is transmitted over virtual telephone line. At the receiver end we receive the audio file and convert it back to the original file.

We have tried to break the complex process into parts so as to make our presentation simple and easy to understand. We hope this effort of ours will encourage people to find simple applications of the complex technology around us that affects our daily lives. We hope that this document will be an aid to the students who want to pursue further in this direction and achieve new heights in this field of technology.

Virtual Telephone Line

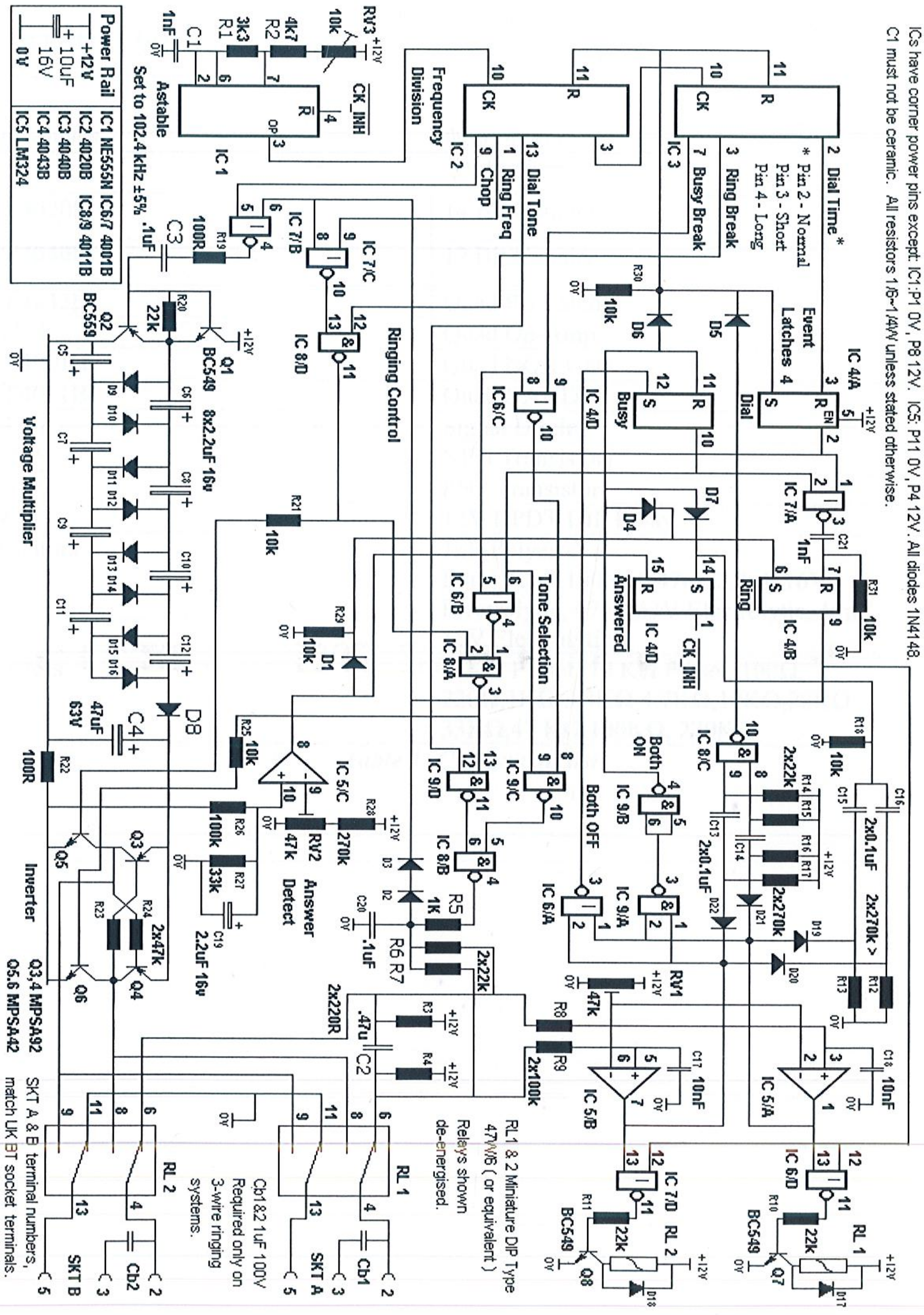
Introduction

Have you ever wanted to link two local telephone devices without using a telephone line?

The following text describes construction, testing and use of a virtual telephone line which can simulate a telephone call / connection between any two local telephone devices.

If your work requires you to demonstrate any form of telephony equipment, Virtual Telephone Line can easy and efficient. We don't require having any telephone line, just connecting your equipment to Virtual Telephone Line can make you do all the things that require telephone line.

Circuit Diagram

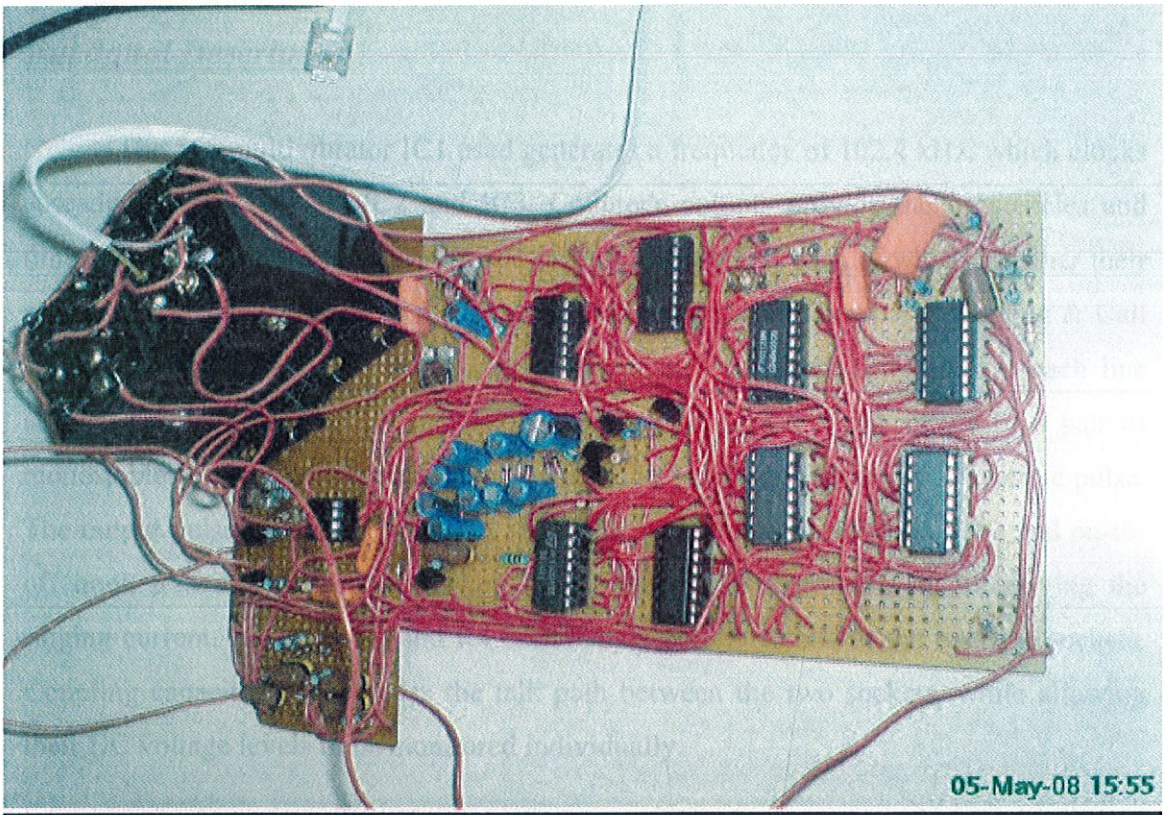


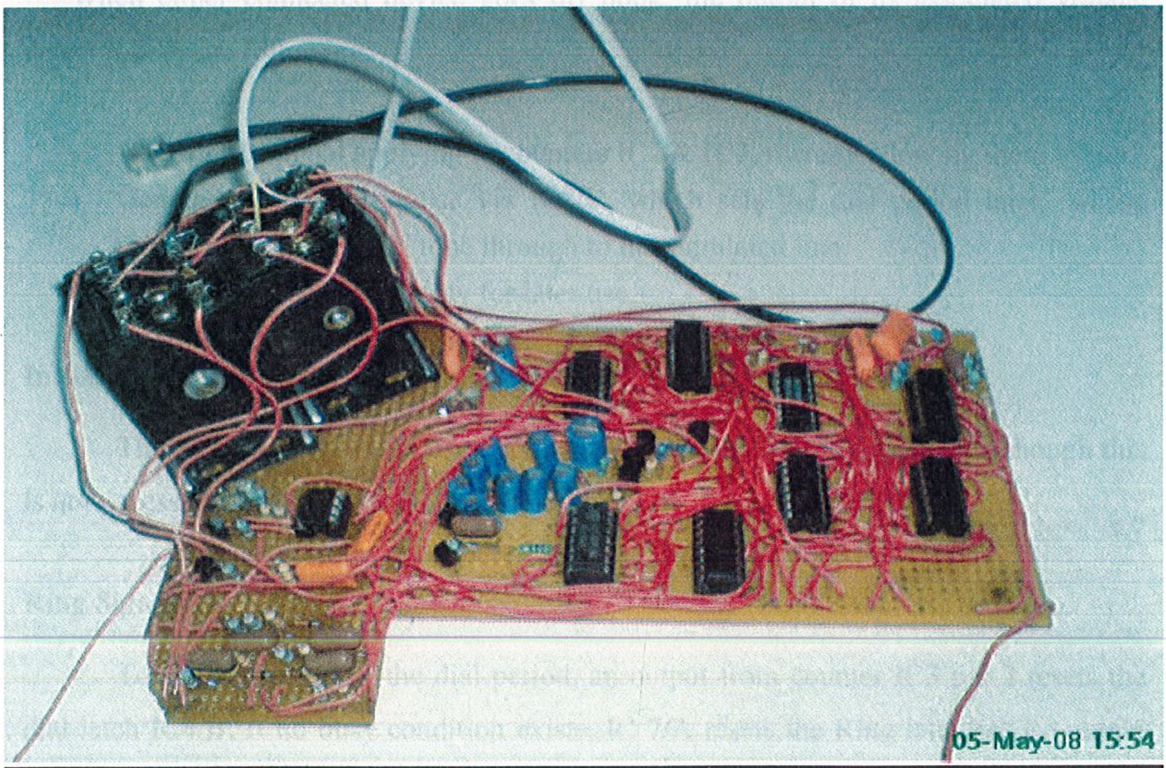
Components List

Components	Description
NE555N	Multivibrator
HCF4020B	14 Bit Counter
HCF4040B	12 Bit Counter
HCF4043B	Quad RS Latch
LM324	Quad Op-Amp
HCF4001B	Quad NOR Gate
HCF4011B	Quad NAND
1N4148	Signal Diode
BC 549	NPN Transistor
BC559	PNP Transistor
47W/6	12V DPDT DIP Relay
Capacitors	1nF(Polyester Film),10nF,100nF,0.47nF,2.2μF 16V Electrolytic, 47μF 63V Electrolytic, 10μF 16V Electrolytic
Resistors	47KΩ Preset, 10 KΩ Preset, 100Ω, 220Ω,1KΩ,3.3KΩ,4.7KΩ,10KΩ,22KΩ, 33KΩ,47 KΩ,100KΩ, 270KΩ

Table 1: Component list

Snapshot of Virtual Telephone Line





Fuctional Description

The 555 multivibrator IC1 used generates a frequency of 102.4 kHz, which clocks cascaded binary counters IC2 and IC3. Counter's outputs provide the frequencies and time periods required. IC4 is comprised of four R/S latches, each of which in either their set or reset states, store the four events: Dialing period, Line Busy, Call Ringing & Call Answered. Op-amps IC5A & IC5B each detect on / off hook conditions at each line socket. The outputs of IC5A and IC5B are each connected to the inputs of a pair of monostables, one of which is triggered by a positive pulse, the other by a negative pulse. The output pulses of the monostables signal the occurrence of off-to-on hook and on-to-off hook transitions at either socket. IC5C detects call answering by monitoring the ringing current. Resistors R3 and R4 each supply current to one of the two line sockets. Coupling capacitor C2 provides the talk path between the two sockets, while allowing their DC voltage levels to be monitored individually.

When either connected device goes off hook, the output of its associated voltage detector does the following:

- Clears reset signal applying to counters IC2 & IC3, thereby allowing them to run.
- Generates a positive pulse via IC8/C, which sets the dial period latch, whose output switches the dial tone through to the simulated line.
- Selects the appropriate relay for later use.

Initializing a Call

The modem software in use may be set to dial an arbitrary number, although this is not necessary. Only TONE dialing should be used.

Ring Signal

To signal the end of the dial period, an output from counter IC3 pin 2 resets the dial latch IC4/B. If no busy condition exists, IC 7/A resets the Ring latch whose output energizes the selected ringing relay and enables the Ring Frequency, Ring Break and

Voltage Multiplier signals. Ringing continues until either the call is answered by a device at the other socket, or the caller hangs up.

Answer Detection

If the call is answered, op-amp IC5C senses across R22 the increased ringing current, sending its output high. This output sets the Ring latch, disabling the ring signal generator and de-energizing the associated relay, thereby restoring the line connection. The Answered latch (IC4D) is reset, disabling all frequency generation for the duration of the call.

Busy Condition

Whenever both telephones are off-hook simultaneously, the busy latch is set, preventing any subsequent ringing until both telephones are hung up. In this state, when the dial period ends, the 400 Hz tone is allowed to persist, interrupted by the Busy-Break signal from IC3/7. This produces a busy tone recognizable by modems and other automatic dialing equipment.

When a call is ended by one of the telephones hanging up, a positive pulse is sent to IC4/D pin 14, setting the Answered latch, which re-enables the astable IC1. In this state, the busy latch remains set; therefore the other telephone receives a dial tone followed by a busy tone. This arrangement prevents a new call being initialized until both telephones are placed on-hook together.

Ringing Tone

A simulated ringing tone is provided to the caller, derived from the dial tone and pulsed at the ringing frequency. Diodes D2 and D3 attenuate the ringing tone relative to the levels of the dial and busy tones. This is done to prevent modems from interpreting the ringing tone as a 'disconnected' or 'unavailable' tone, which would lead to premature disconnection.

Ring Signal Generator.

The voltage required to simulate a ringing signal, is generated by a *Cockcroft-Walton* capacitor-based voltage multiplying circuit. The circuit is pulsed from the lsb output of the first binary counter, via an npn/pnp push pull buffer circuit. Driving the buffer from the counter's output, ensures a 1/1 mark-space ratio, essential for this type of multiplier. The ring signal is typically around 45 volts, sufficient to meet the 40 volt minimum for automatic answering equipment, without exceeding the 50 volt threshold considered safe for electronic projects.

Testing and Fault Diagnosis

Functional Test

The following is a description of normal operation.

With the unit powered, telephones were connected to both the sockets, which will be referred to as A and B.A dialing tone could be heard for approximately 5 seconds when A was picked up. At the end of the dialing period, the appropriate relay will operate and a ringing tone appeared through the receiver. B will be ringing in single rings, not paired. Hang up A. The relay drops out and ringing stops. Above steps were repeated, this time lifting B when it starts to ring. With both telephones off-hook, voice communication is possible between them with no tones or pulses being heard. Both telephones were restored and then A was picked up. Before the dial tone finishes, B was picked up. At the end of the dial period, the dial tone changes to a busy tone, audible in both telephones. B was put off then in A we heard a dial tone followed by a continuous busy tone. When B was picked up a dial tone appeared in both telephone followed by a busy tone. All the steps were repeated treating B as A and A as B. The unit now can be used with any telephone apparatus.

We required at least one and preferably two telephones. If we have only one telephone available, we would require in addition a telephone line plug with pins 2 and 5 shorted together. When inserted into either socket, this plug simulates a telephone that is off-hook. To simulate making a call, plug was inserted into either socket. To simulate the answering of a telephone, after the ringing signal has started, the plug was inserted into ringing socket.

Off-hook Detectors

Monitoring IC5 pin 7, RV1 was adjusted so that with the telephone on-hook (hung up) the voltage is +12V, and when the telephone is off-hook the voltage swings to 0V.

Same procedure (monitoring IC5 pin 1) was repeated with telephone connected to socket B.

Telephones were connected to both sockets. Pin 10 was monitored for a positive pulse whenever either telephone is taken off-hook.

With both telephones off-hook, pin 14 (IC4) should be low. A positive pulse appeared on pin 14 when each telephone is placed on-hook while the other is off-hook. With both telephones on-hook, pin 14 was at a high level, sourced from IC9 pin 14.

Fault diagnosis consists simply of following the signals back to their source at IC5 pins 1 and 7.

Call Answered Detector

Hang up connected telephones. Monitoring IC5 pin 8, RV2 was adjusted to a point where the voltage is low. Typically, pin 9 will be at around 0.4 Volts. This amplifier detects answering by monitoring for an increase in ringing current, therefore during functional testing, RV2 may require re-adjustment.

Astable

IC1 was oscillating by monitoring for a pulse train on pin 3. If a fault is holding pin 4 low, the wire to pin 4 was disconnected temporarily. With IC1 working, RV3 was adjusted until the frequency on pin 3 is 102.4 kHz +/- 5%. If using an oscilloscope, the periodic time should be 9.76µS. The table below describes the location specific settings:.

Table of Frequencies & Time Delays Available from Divider Outputs Using Standard Clock Frequency 102.4 KHz			
IC 2 pin 9	Freq Hz = 51200.00	Period (S) = 0.0000	ON time (S) = 0
IC 2 pin n/a	Freq Hz = 25600.00	Period (S) = 0.0000	ON time (S) = 0
IC 2 pin n/a	Freq Hz = 12800.00	Period (S) = 0.0000	ON time (S) = 0
IC 2 pin 7	Freq Hz = 6400.00	Period (S) = 0.0002	ON time (S) = 0.0001
IC 2 pin 5	Freq Hz = 3200.00	Period (S) = 0.0003	ON time (S) = 0.00015
IC 2 pin 4	Freq Hz = 1600.00	Period (S) = 0.0006	ON time (S) = 0.0003

IC 2 pin 6	Freq Hz = 800.00	Period (S) = 0.0013	ON time (S) = 0.00065
IC 2 pin 13	Freq Hz = 400.00	Period (S) = 0.0025	ON time (S) = 0.00125
IC 2 pin 12	Freq Hz = 200.00	Period (S) = 0.0050	ON time (S) = 0.0025
IC 2 pin 14	Freq Hz = 100.00	Period (S) = 0.0100	ON time (S) = 0.005
IC 2 pin 15	Freq Hz = 50.00	Period (S) = 0.0200	ON time (S) = 0.01
IC 2 pin 1	Freq Hz = 25.00	Period (S) = 0.0400	ON time (S) = 0.02
IC 2 pin 2	Freq Hz = 12.50	Period (S) = 0.0800	ON time (S) = 0.04
IC 2 pin 3	Freq Hz = 6.25	Period (S) = 0.1600	ON time (S) = 0.08
IC 3 pin 9	Freq Hz = 3.125	Period (S) = 0.32	ON time (S) = 0.16
IC 3 pin 7	Freq Hz = 1.563	Period (S) = 0.64	ON time (S) = 0.32
IC 3 pin 6	Freq Hz = 0.781	Period (S) = 1.28	ON time (S) = 0.64
IC 3 pin 5	Freq Hz = 0.391	Period (S) = 2.56	ON time (S) = 1.28
IC 3 pin 3	Freq Hz = 0.195	Period (S) = 5.12	ON time (S) = 2.56
IC 3 pin 2	Freq Hz = 0.098	Period (S) = 10.24	ON time (S) = 5.12
IC 3 pin 4	Freq Hz = 0.049	Period (S) = 20.48	ON time (S) = 10.24
IC 3 pin 13	Freq Hz = 0.024	Period (S) = 40.96	ON time (S) = 20.48
IC 3 pin 12	Freq Hz = 0.012	Period (S) = 81.92	ON time (S) = 40.96
IC 3 pin 14	Freq Hz = 0.006	Period (S) = 163.84	ON time (S) = 81.92
IC 3 pin 15	Freq Hz = 0.003	Period (S) = 327.68	ON time (S) = 163.84
IC 3 pin 1	Freq Hz = 0.002	Period (S) = 655.36	ON time (S) = 327.68

Table 2: Frequencies and Time Delays

Diagnostic Section

Off-Hook Detectors: - If the outputs of IC5 pins 7 & 1 do not respond when a telephone is lifted off-hook, check for a change in level on the corresponding input pins 5 & 3. If there is no change there either, follow back to pin 6 of the corresponding relay. Verify that none of the relay is energized at this point by checking for a low on the output of the driver gates. If the relays are OK, there must be a wiring fault associated with the line sockets. Ensure that pin 11 of both relays is connected to 0V.

Clock Frequency: - Normally the oscillator IC1 should be halted only by a low level on pin 4, due to the Answered latch (IC4/D) being reset. If there is no output from IC1 with pin 4 high, check the timing components: C1, R1, R2 & RV3. With both telephones either hung up or disconnected, IC4 pin 14 should be high from IC9 pin 4 (via D7). The output of the answer detector (IC5 pin 8) should be low. If it is high, adjust RV2 until it goes low.

Dial tone:- When either of the handset is lifted, regardless of the position of the other handset if connected, there should be a positive pulse reaching IC4 pin 4 from IC8 pin 10, causing IC4 pin 2 to go high. If no pulse occurs at IC4 pin 4, follow the signal back through IC8/C and the monostables that drive it. At this point IC4 pin 2 should go high. Pin 3 of IC4 should have been low throughout, only going high to signal the end of the dial period.
If IC4 pin 2 does go high, check the path of the dial tone frequency from IC2 pin 13, through the gating system to IC8 pin 4.

Busy Tone: - All the tones heard in the handset are derived from the dial tone signal from IC2 pin 13. In the busy state, IC4 pin 10 is high and IC4 pin 2 is low. This allows the Busy Break signal to pass through IC6/C and IC9/C to IC8/B where it breaks up the dial tone on pin 6.

Ring Voltage Generator: - Bring the unit to the point at which ringing should start and ensure that the appropriate relay operates. With no telephone connected at the called socket, check for 48-56 volts at the cathode of D8. When there is a telephone connected at the called socket, this voltage will drop by a few volts typically. If OK, go to the next section. IC7 pin 6 should be low from IC4 pin 9.
Check for a pulse train on IC7 pin 4 derived from IC2 pin 9 (50 kHz). If OK, check for a pulse on the emitters of Q1 or Q2. If OK, check for voltage multiplication along the four

stages of the multiplier. If multiplication is taking place but the output is lower than expected, it is probably being loaded by a fault associated with the inverter.

Ring Voltage Inverter: - During the ringing period, monitor for an intermittent 25 Hz pulse at IC8 pin 11 and on IC8 pin 3. IC7 pin 8 must be low. There should be a constant pulse on IC7 pin 9, interrupted by the ring break signal on IC8 pin 12.
In ringing mode, the intermittent a.c. ringing voltage should be measurable across the collectors of Q3 & Q4. This voltage will be present whenever the ring tone is heard in the connected handset.

Answer Detector: - Referring to IC5, during ringing, the voltage on pin 10 is around 0.2 volts. When a ringing telephone is answered, this voltage rises to the point at which pin 8 goes high, which sets the ring latch and resets the answered latch. If ringing pulse continues after answering, lower the voltage on pin 9 using RV 2. Conversely, if the ring signal cuts out immediately after it start, increase the voltage on pin 9.

Ring Tone: - The ringing tone is derived at IC9/D by interrupting the dial tone using the ringing frequency. To attenuate the ringing tone, its amplitude is clamped by d2 & d3. The reason for this is that if the amplitude of this tone is too high, the modem may interpret it as an attempt at connection from an answering modem, leading to premature disconnection and a 'NO CARRIER' message.

Wave File Format

Overview

The Wave file format is Windows' native file format for storing digital audio data. It has become one of the most widely supported digital audio file formats on the PC due to the popularity of Windows and the huge number of programs written for the platform. Almost every modern program that can open and/or save digital audio supports this file format, making it both extremely useful and a virtual requirement for software developers to understand. The following specification gives a detailed description of the structure and inner workings of this format.

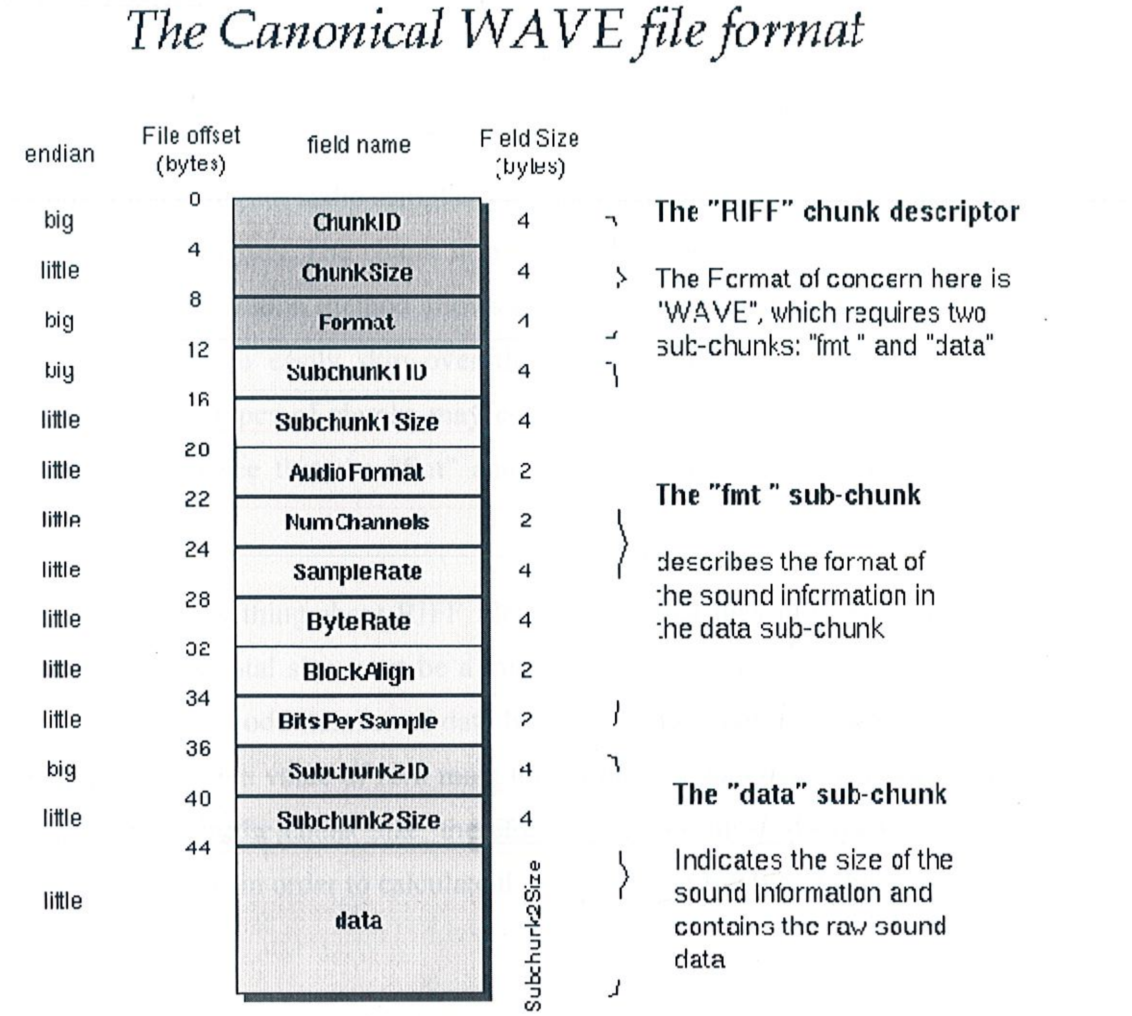


Figure 3: WAV file format

File Structure

Wave files use the standard RIFF structure which groups the files contents (sample format, digital audio samples, etc.) into separate chunks, each containing it's own header and data bytes. The chunk header specifies the type and size of the chunk data bytes. This organization method allows programs that do not use or recognize particular types of chunks to easily skip over them and continue processing following known chunks. Certain types of chunks may contain sub-chunks. For example, in the diagram below, you can see that the "fmt" and "data" chunks are actually sub-chunks of the "RIFF" chunk.

One tricky thing about RIFF file chunks is that they must be word aligned. This means that their total size must be a multiple of 2 bytes (ie. 2, 4, 6, 8, and so on). If a chunk contains an odd number of data bytes, causing it not to be word aligned, an extra padding byte with a value of zero must follow the last data byte. This extra padding byte is not counted in the chunk size, therefore a program must always word align a chunk headers size value in order to calculate the offset of the following chunk.

Wave File Header - RIFF Type Chunk

Wave file headers follow the standard RIFF file format structure. The first 8 bytes in the file is a standard RIFF chunk header which has a chunk ID of "RIFF" and a chunk size equal to the file size minus the 8 bytes used by the header. The first 4 data bytes in the "RIFF" chunk determine the type of resource found in the RIFF chunk. Wave files always use "WAVE". After the RIFF type comes all of the Wave file chunks that define the audio waveform.

Basic Wave File Layout

Chunk ID "RIFF"
Chunk Data Size

RIFF Type ID "WAVE"

Chunk ID "fmt "
Chunk Data Size

Sample Format Info

Chunk ID "data"
Chunk Data Size

Digital Audio Samples

Chunk Header

Chunk Data Bytes

Figure 4: Basic Layout

Wave File Chunks

There are quite a few types of chunks defined for Wave files. Many Wave files contain only two of them, specifically the Format Chunk and the Data Chunk. These are the two chunks needed to describe the format of the digital audio samples and the samples themselves. Although it is not required by the official Wave file specification, it is good practice to place the Format Chunk before the Data Chunk. Many programs expect the chunks to be stored in this order and it is more sensible when streaming digital audio from a slow, linear source such as the Internet. If the format were to come after the data, all of the data and then the format would have to be streamed before playback could start correctly.

All RIFF Chunks and therefore Wave Chunks are stored in the following format. Notice that even the above mentioned RIFF Type Chunk conforms to this format.

Offset	Size	Description
0x00	4	Chunk ID
0x04	4	Chunk Data Size
0x08	Chunk Data Bytes	

Table 3: RIFF chunk format

The rest of this document goes through the different types of Wave chunks, describing the format of their data bytes and what they mean. You can use the table of contents at the beginning of this document to help find the chunk type you are interested in.

Format Chunk - "fmt "

The format chunk contains information about how the waveform data is stored and should be played back including the type of compression used, number of channels, sample rate, bits per sample and other attributes.

Offset	Size	Description	Value
0x00	4	Chunk ID	"fmt " (0x666D7420)
0x04	4	Chunk Data Size	16 + extra format bytes
0x08	2	Compression code	1 - 65,535
0x0a	2	Number of channels	1 - 65,535
0x0c	4	Sample rate	1 - 0xFFFFFFFF
0x10	4	Average bytes per second	1 - 0xFFFFFFFF
0x14	2	Block align	1 - 65,535
0x16	2	Significant bits per sample	2 - 65,535
0x18	2	Extra format bytes	0 - 65,535
0x1a		Extra format bytes *	

Table 4: Wav format chunk values

, * read following text for details

Chunk ID and Data Size

The chunk ID is always "fmt " (0x666D7420) and the size is the size of the standard wave format data (16 bytes) plus the size of any extra format bytes needed for the specific Wave format, if it does not contain uncompressed PCM data. Note the chunk ID string ends with the space character (0x20).

Compression Code

The first word of format data specifies the type of compression used on the Wave data included in the Wave chunk found in this "RIFF" chunk. The following is a list of the common compression codes used today.

Code	Description
0 (0x0000)	Unknown
1 (0x0001)	PCM/uncompressed
2 (0x0002)	Microsoft ADPCM
6 (0x0006)	ITU G.711 a-law
7 (0x0007)	ITU G.711 Âµ-law
17 (0x0011)	IMA ADPCM
20 (0x0016)	ITU G.723 ADPCM (Yamaha)
49 (0x0031)	GSM 6.10
64 (0x0040)	ITU G.721 ADPCM
80 (0x0050)	MPEG
65,536 (0xFFFF)	Experimental

Table 5: common Wav compression codes

Number of Channels

The number of channels specifies how many separate audio signals that are encoded in the wave data chunk. A value of 1 means a mono signal, a value of 2 means a stereo signal, etc.

Sample Rate

The number of sample slices per second. This value is unaffected by the number of channels.

Average Bytes Per Second

This value indicates how many bytes of wave data must be streamed to a D/A converter per second in order to play the wave file. This information is useful when determining if data can be streamed from the source fast enough to keep up with playback. This value can be easily calculated with the formula:

```
AvgBytesPerSec = SampleRate * BlockAlign
```

Block Align

The number of bytes per sample slice. This value is not affected by the number of channels and can be calculated with the formula:

```
BlockAlign = SignificantBitsPerSample / 8 * NumChannels
```

Significant Bits Per Sample

This value specifies the number of bits used to define each sample. This value is usually 8, 16, 24 or 32. If the number of bits is not byte aligned (a multiple of 8) then the number of bytes used per sample is rounded up to the nearest byte size and the unused bytes are set to 0 and ignored.

Extra Format Bytes

This value specifies how many additional format bytes follow. It does not exist if the compression code is 0 (uncompressed PCM file) but may exist and have any value for other compression types depending on what compression information is need to decode the wave data. If this value is not word aligned (a multiple of 2), padding should be added to the end of this data to word align it, but the value should remain non-aligned.

Data Chunk - "data"

The Wave Data Chunk contains the digital audio sample data which can be decoded using the format and compression method specified in the Wave Format Chunk. If the Compression Code is 1 (uncompressed PCM), then the Wave Data contains raw sample values. This document explains how an uncompressed PCM data is stored, but will not get into the many supported compression formats.

Wave files usually contain only one data chunk, but they may contain more than one if they are contained within a Wave List Chunk ("wavl").

Offset	Length	Type	Description	Value
0x00	4	char[4]	chunk ID	"data" (0x64617461)
0x04	4	dword	chunk size	depends on sample length and compression
0x08				sample data

Table 6: Data chunk format

Multi-channel digital audio samples are stored as interlaced wave data which simply means that the audio samples of a multi-channel (such as stereo and surround) wave file are stored by cycling through the audio samples for each channel before advancing to the next sample time. This is done so that the audio files can be played or streamed before the entire file can be read. This is handy when playing a large file from disk (that may not completely fit into memory) or streaming a file over the Internet.

The values in the diagram below would be stored in a Wave file in the order they are listed in the Value column (top to bottom).

One point about sample data that may cause some confusion is that when samples are represented with 8-bits, they are specified as unsigned values. All other sample bit-sizes are specified as signed values. For example a 16-bit sample can range from -32,768 to +32,767 with a mid-point (silence) at 0.

Time	Channel	Value
0	1 (left)	0x0053
	2 (right)	0x0024
1	1 (left)	0x0057
	2 (right)	0x0029
2	1 (left)	0x0063
	2 (right)	0x003C

Table 7: interlaced stereo wav samples

As mentioned earlier, all RIFF chunks (including WAVE "data" chunks) must be word aligned. If the sample data uses an odd number of bytes, a padding byte with a value of zero must be placed at the end of the sample data. The "data" chunk header's size should not include this byte.

Final Layout of the Wave Format

Offset	Size	Name	Description

The canonical WAVE format starts with the RIFF header:

0	4	**ChunkID**	Contains the letters "RIFF" in ASCII form (0x52494646 big-endian form).
4	4	**ChunkSize**	36 + SubChunk2Size, or more precisely: 4 + (8 + SubChunk1Size) + (8 + SubChunk2Size) This is the size of the rest of the chunk following this number. This is the size of the Entire file in bytes minus 8 bytes for the two fields not included in this count: ChunkID and ChunkSize.
8	4	**Format**	Contains the letters "WAVE" (0x57415645 big-endian form).

The "WAVE" format consists of two subchunks: "fmt " and "data":
The "fmt " subchunk describes the sound data's format:

12	4	**Subchunk1ID**	Contains the letters "fmt " (0x666d7420 big-endian form).
16	4	**Subchunk1Size**	16 for PCM. This is the size of the rest of the Subchunk which follows this number.
20	2	**AudioFormat**	PCM = 1 (i.e. Linear quantization) Values other than 1 indicate some form of compression.

22	2	**NumChannels**	Mono = 1, Stereo = 2, etc.
24	4	**SampleRate**	8000, 44100, etc.
28	4	**ByteRate**	= SampleRate * NumChannels * BitsPerSample/8
32	2	**BlockAlign**	= NumChannels * BitsPerSample/8 The number of bytes for one sample including all channels. I wonder what happens when this number isn't an integer?
34	2	**BitsPerSample**	8 bits = 8, 16 bits = 16, etc.
	2	**ExtraParamSize**	if PCM, then doesn't exist
	X	**ExtraParams**	space for extra parameters

The "data" subchunk contains the size of the data and the actual sound:

36	4	**Subchunk2ID**	Contains the letters "data" (0x64617461 big-endian form).
40	4	**Subchunk2Size**	= NumSamples * NumChannels * BitsPerSample/8 This is the number of bytes in the data. You can also think of this as the size of the read of the subchunk following this number.
44	*	**Data**	The actual sound data.

Codes for Format conversion

As we have already discussed that we can convert the file of any given format into wav format and then we can transmit it using our method, this procedure secures the hidden data and provides the secured transmission over the unsecured channels.

The conversion of the format can be done using the programs given in the CD-ROM.

Advantages and uses

Fax Machine as Scanner and Printer.

Virtual Telephone Line helps your fax machine in functioning both as a monochrome full-page scanner and as a high-resolution printer.

To initialize a connection, just set the fax or PC to dial any arbitrary number (or no number).

Link two local computers to play modem games.

Many computer games allow gaming between two players on different PCs via modem or serial cable. If a serial connection is not possible, you can link two local computers via Virtual Telephone Line.

Secured transmission

Using the methods of conversion explained above we can achieve a secured transmission over an unsecured channel. In this process first we can convert the any required file to be transmitted into a .wav file and then transmit it using virtual telephone line and at the receiver end same file can be regenerated back.

Operational Limitations

1. This unit is not designed to simulate long distance transmission calls.
2. This unit is not intended for use as long distance intercom system.
3. When first connected to this system electronic phones may take a few seconds to start working.

BIBLIOGRAPHY

Vishwanathan T. *Telecommunication switching systems and networks,* Prentice Hall of India, 2005

Bai Y, *The Windows Serial Port Programming Handbook*, Auerbach Publications London.

Buchanan W.J, *The Handbook of Data communication and Network,* Kluwer Academic Publishers Boston.

Bosse J and Devefak F, *Signaling in Telecommunication Networks,* Wiley Interscience

Sedra A and Smith K, *Microelectronic Circuits*, Oxford University Press

www.netghost.narod.ru/gff/graphics/summary/micriff.htm

www.lightlink.com/tjweber/StripWav/WAVE.html

http://veryuseful.info/linesim/