

*May 09, 1999*

**PHONE BASED E-MAIL SERVER**  
*(Master's Project)*

*Lab: ECSL at Stony Brook*  
*Professor: Tzi-cker Chiueh*  
*M.S student: Adil Attari*

## **Abstract :**

Lately there has been many commercial products in the software market exploring the tremendous need of communication between people. While phone is one of the most classical way of live communication, e-mail has become a standard cross boundaries way of communicating. It is cheap and do not assume the other person is present to take your message instantly. A combination of those two powerful communication tools seems to get another even more promising tool for communication. Today in the industry we find products that exploit this idea, yet requiring an additional piece of hardware to seize the text and dialing in to a remote e-mail server. Our approach is to use regular phone handsets to dial to your home machine –supposedly up, and your voice modem monitoring incoming phone calls, will allow you to send an e-mail, and forward each other calls to an answering machine.

## **Introduction :**

Phone based e-mail is a very smart way of sending e-mail while you are away of your machine. Moreover, our way of approaching the problem bypass the need of purchasing an additional piece of hardware to type in your messages. A regular phone (connected to a carrier) will do. This convenience come with the price of having your machine (at home) up. As PCs are becoming common at our places for multipurpose use, this assumption doesn't seem too much demanding.

A typical scenario would be that you are away from your machine and want to send e-mail. Assuming that our Software is running on your machine, your voice modem will handle the incoming calls. Via a menu selection you will be prompted to either leave a message (usual phone calls), or to send an e-mail. The second option selected, you will be asked to enter a password, and than, voice directions will help you to know at each stage of the procedure of sending e-mail you are.

## **Platform :**

Our software is running on a LINUX platform, Red Hat 5.2. Since it is a UNIX-like O.S, it is characterized b the ease of communication with the serial port, that is, once opened, it returns a descriptor we will use to read from or write to data.

## **Hardware :**

We used a Simultaneous Voice and Data modem. The whole communication with the modem will be in the modem put in Voice mode. In order to use the playing/

recording voice files capabilities, we used a package that provide tools to drive voice modems[1].

### **Structure of the Program:**

Our program acts as automata (the Definite one). Depending on triggering actions (phone ring) or phone keys entered, it changes its state (Fig –1). The initial state would be when your daemon is running on your machine, having set your modem to its voice mode. A phone ring will make the transition from the initial state to the “menu select” state. In other words, the modem picks up the call and a greeting message along with a voice menu is played. People calling you may leave a message when selecting to do so. The second option of sending email will be protected by a simple personnel password you have already set up.

The program goes on, with switching from state to state depending on the characters you enter. There are two distinctions to make here:

One is that while hearing a selection, a simple key strike-representing the number of your selection will change the state of your program. While the second is, inside the composing procedure, three key strikes, with the last as ‘\*’ means that you Composed one character in your message/address; whereas striking ‘#’, means finishing Entering text and moving to the next step (either from composing message to composing email address or from composing email address to sending email).

Using the voice capabilities of the modem we are also able to play back each character entered. Therefore the user can always go one step back, and correct the last character entered.

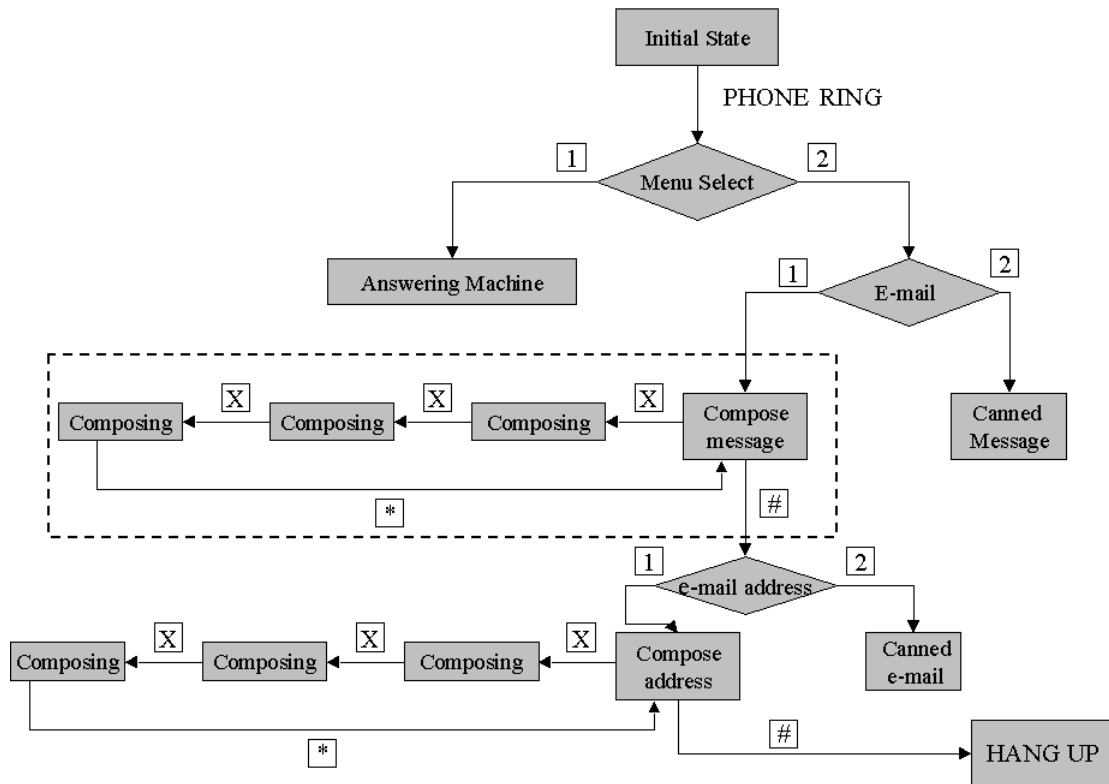


Figure-1.

*This figure represents the different states into which the program may be at a given time. State transition is operated when a phone's key is stroke, with the only exception at the initial state where transition is triggered by a phone ring.*

*The area in the dashed rectangle, represents the details of which could be just one State, the message composition state.*

## **Communication with the modem:**

Through the serial port, we send AT command [2]. That is the case for the initialization procedure as well as for setting the modem in the voice mode or enabling the modem to send or receive voice messages.

An SVD modem is characterized by its three modes:

Data mode, is the default one, where your modem is enabled to send and receive digital data. (PPP connection to your ISP for instance).

Fax mode, enable your modem to send and receive faxes.

Voice/Audio mode, the one with particular interest to our project, supports three submodes:

1. Online Voice Command mode :  
This mode allows sending AT commands to the modem without aborting the telephone line connection.
2. Voice Receive mode :  
This mode allows the recording of voice data, and is used when recording a message for example.
3. Voice Transmit mode :  
Transmitting voice data is possible via this mode.

Extensive discussion about those modes, AT commands specification can be found in the Rockwell Semiconductor Systems website.

As far as our software is concerned, the communication with the modem was boiled down to two main problems:

First is, how to write data to the modem and how to read data from it. As stated before, we just write string of chars to the serial port where the modem is connected, and read from it string of chars. However, as obvious as it seems to be, this operation needed some tedious manipulation of Reverse Engineering techniques. The problem is mainly of synchronization, the modem sends sometimes some “bizarre” characters due to an non empty buffer, or due to noise in the line. Our approach is to keep reading from the buffer and throw away all the chars we judge as garbage. This is made possible by the state structure of the program: we know in advance what kind of information (alphanum, \* or #) we are expecting to read from the buffer in each stage of the program.

The second is the switch to Voice mode. The switch in itself is not the problem, this is done in a straight forward manner by sending the AT#CLS=8 command to the modem. Rather the problem was of figuring out the right sampling to use, along with the baud rate . We also explored the virtue of such mode that allows the sending of special chars to the modem in the stream and can be interpreted as a modem command. For example, the occurrence of the sequence <DLE><ETX> in a voice stream is interpreted as flushing the modem buffers. This property helped a lot for regularizing the behavior of the modem : Therefore, while playing a voice file through the modem, and pending

to its end the sequence <DLE><ETX> we obtain a cleaning of its buffers at the same time.

**Alphabet :**

A major question would be, how can we code the Alphabet plus Numbers on such a key limited keyboard as the one of a phone’s handset? Since each phone key has the label of three Letters, we alleviate the ambiguity of selecting one letter from the three by entering its Sequential number in the key followed by the ‘\*’ key. For example on key #2, where the sequence ‘ABC’ appears, A has sequence number 1, B has 2 and C has 3. Therefore, to enter ‘C’, the user should strike the following keys: 2 + 3 + \*. At a first glance such a schema appears to be very hard and long; however, the intuition underneath it is quite simple and obvious. Experience has shown that your hands will learn it quicker than you might think. (we experience it with students from different departments : humanities, management; and they seem to pick it up pretty quickly).

However, two optimizations has been included to reduce both the time of typing in Characters and reducing the risk of repetitive errors:

The first is a shortcut mechanism; some frequently used group of characters, will be referred to by combos similar to those coding a letter (Fig-2). For example “edu” will be entered by hitting 3 + 8 and “com” by hitting 2 + 6 (plus the “\*” at the end).

Group of chars	Phone keys	Mnemonic
edu	3 + 8 + *	E + U
com	2 + 6 + *	C + O
org	6 + 7 + *	O + R

Figure –2

*The mnemonic corresponds to the phone key where the letter occurs.*

The second optimization is to have pre-written messages that we call “canned messages”. In fact at each stage where you are susceptible of composing a text, you are first asked whether you would prefer to send out a canned message. Typical canned messages would be: < I am running late to the meeting>, <I am on my way home >, <class is canceled today>, <call me up ASAP> . The same argument holds for an address book.

### **Sending E-mail :**

Hitting the “#” key after entering the e-mail address sends your message. A simple pipe is opened to the mail program in Linux, with the appropriate parameters: Address and subject on the line command plus message body.

### **Enhancements :**

Including a reading e-mail component to the software seems to be a good idea, and will provide a full e-mail capability server. A simple synthesizer will do or any other text-to-speech software. Such application is already in use, and is mostly included in cellular phones, while the idea of sending e-mail via phone without the help of any other device seems to be new.

### **Conclusion :**

SVD modems offer larger capabilities than regular modems. The additional, Voice/Audio mode opens the door widely for new applications in Computer Telephony.

The idea of our software seems to be simple, however a lot of people are enthusiastic about it, and find it extremely useful. Many applications of our software in the industry come to mind: ISPs could offer Phone e-mail accounts for people who do not have an e-mail account, or for people who want to be able to send e-mail even if they are away from their machines.

We have been contacted by a local phone company in New York City which is, And I quote its director, “hugely interested in our product”. The adaption of the product To the commercial environment would be the incoming step.