UNIVERSITY OF HOUSTON

PROJECT THEMIS
Information Processing
Systems

ONR Contract N00014 68 A 0151

On The Use Of Very Low Cost Terminals

Albert Newhouse
Robert A. Sibley, Jr.

RE 3-69
March, 1969

25

On the Use of Very Low Cost Terminals

Albert Newhouse and Robert A. Sibley, Jr.
University of Houston
Houston, Texas, U.S.A.

Sponsored by

Information Systems Branch
Office of Naval Research
United States of America
Contract No. N00014-68-A-0151

Initially presented at the
International Conference on Remote Data Processing
Paris, March, 1969

With this note, we will attempt to briefly characterize the early results of an effort that seeks to maximize the number of places (cities, buildings, rooms, desks, ...) from which people can interact with computers in a meaningful way. Because of the extraordinary availability and low cost of the ordinary telephone, it was chosen as the cornerstone of the project. A set of applications were then identified. These applications include desk calculator services (scientific and business), information storage and retrieval (public assistance and private industry), and computer program development.

The studies have involved the design and implementation of a specific hardware/software system. Telephone signals are interfaced and multiplexed into an Interdata Model 4 computer in the same manner as teletype messages. Output is generated by a Cognitronics Speechmaker as a consequence of selection codes that are presented to a specially designed buffer. Requirements for large data collections and substantial computational services will be satisfied by a high speed connection to the University's central computing facility.

In turn, the touchtone telephone, other hardware, a desk calculator application, other applications and software will be discussed.

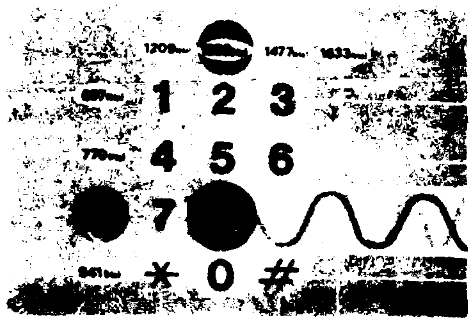
<u>The touchtone telephone</u>

The primary motivation for this project came from the realization that the touchtone telephone was fast becoming a common appliance in the United States. The ten-button version, first introduced in 1964, was quickly followed by a twelve-button version in 1968; and the flexibility added by the two extra buttons appears crucial in making really practical use of the device as a remote terminal. The device, under a variety of names, is produced by several manufacturers.

In most sections of Houston, Texas, the Bell System will install a line with one or more touchtone telephones for an initial charge of $5.00 and a monthly rental charge of $2.25 ($1.75 for a private residence). This charge includes maintenance and, of course, no supplies are required. By 1971, it is anticipated that conversion to the touchtone telephone will be complete in Houston; and by 1975, it will be possible to use this device throughout the United States for ordinary telephone service. This means that if we are able to develop techniques whereby the touchtone telephone can be used as a remote computer terminal in practical ways and for practical purposes, then it will be possible and even probable that computer services can be made available to every business and every household that can afford telephone service.

For data input from the telephone, our budget, if not the state of the art, causes us to ignore the possibility of voice communications, and instead, turn to the twelve buttons that appear on the face of the device. Ten of the buttons are the same ones that are used for switching purposes in the course of ordinary telephone service. The additional two currently have no meaning as far as the telephone company is concerned, and are apparently there to facilitate data communication. For practical purposes, the rate at which the buttons can be depressed is limited by the capabilities of the user rather than by the device; and for most every person, that rate is much faster than what would be achieved with a dial telephone. With such a button device, of course, every entry takes the same amount of time, regardless of which button is used (on a dial telephone, an "8" takes approximately three times as long as a "1"). For the automatic entry of sequences of characters, a wide variety of equipment is available at fairly reasonable prices. In general, equipment of this type transmits at a rate of 10 characters per second. Of particular interest at the moment, is a telephone that is capable of reading a card that has a capacity of fourteen characters. These cards are typically plastic, and have perforated holes that can be easily punched with an ordinary pencil. By punching a "stop" hole, one may provide for interruptions in the reading process, so that manual entries can be made at any point. Such a device costs about $3.50 a month more than the ordinary touchtone telephone.

For output, it is necessary to use the audio facility, and, fortunately, appropriate hardware for generating messages from modest vocabularies is both available and economical. The disadvantages of not being able to obtain hard copy are quite substantial for many applications, and indeed catastrophic for some. But the situation is not without advantages. Eyesight is not required, and no disturbance is created even under very crowded circumstances.

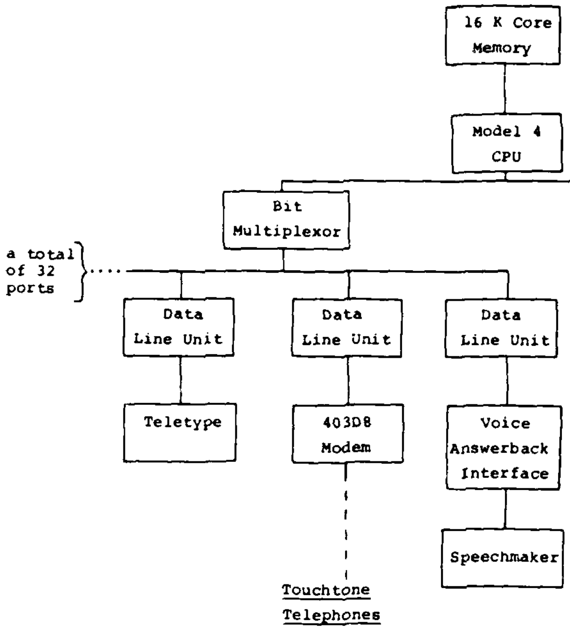


The twelve buttons are arranged in the form of a matrix that has four rows and three columns. A frequency, or tone, is associated with each row (697, 770, 852 and 941 Hz from top to bottom) and with each column (1209, 1336, 1477 and 1633 hz from left to right). The depression of a button then, results in the transmission of two tones, one corresponding to its row and one corresponding to its column. The extra column tone is obviously there in anticipation of a fourth column of buttons; but at this time, plans are not being discussed.

Reliability is certainly not a problem, for a transmission error rate of 1 in 1 x 10^5 is projected; and this is without any software error detection and correction.

For those areas where touchtone telephones are not yet available, at least one company (Metroprocessing Corporation of America) is marketing a portable device that, for all practical purposes, has the same characteristics as a touchtone telephone, and it may be easily attached to any dial telephone through the use of a coupler that slips onto the mouthpiece. One must, of course, continue to use the dial for establishing the connection. The "Spartan FT-1240" is 2 x 4 x 8 inches in size, weighs 2 1/2 pounds and uses one transitor radio battery for power. Prices appear to range from $145. to $250.

Other hardware

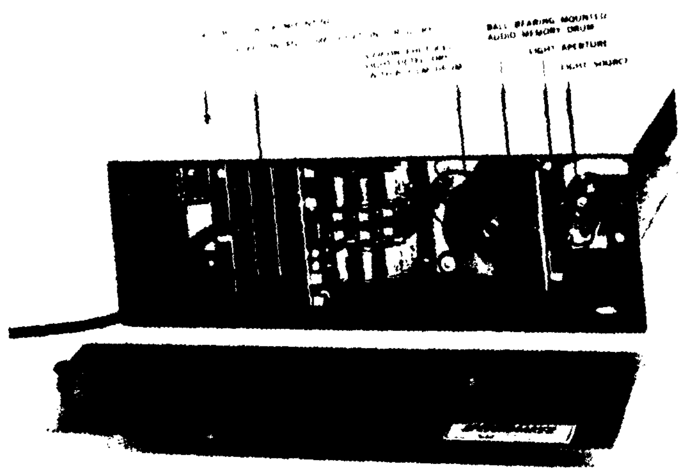


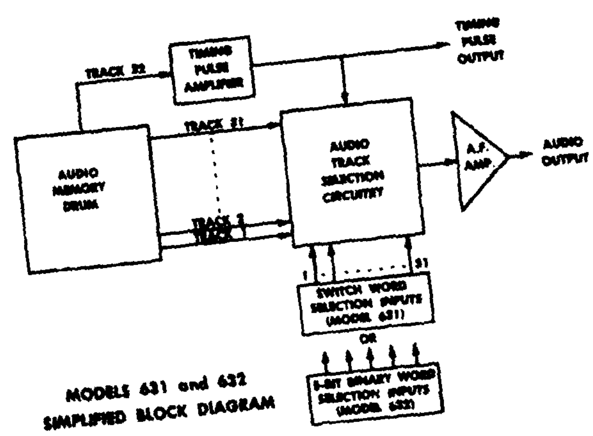
Let us now consider the hardware system that we have designed for use in conjunction with the touchtone telephone. A 403D8 modem is used as the interface between the telephone line and the computer system proper. In addition to the normal interfacing functions, this modem converts each pair of tones into a legitimate teletype character; and, in fact, allows the computer to treat the line in the very same manner that it would treat a teletype line.

The modem also provides for three answer back tones (1017, 1875 and 2025 Hz) that may be transmitted back to the telephones under the control of the computer.

The digital computer is an Interdata Model 4 with 16K bytes of 980 nanosecond (for 2 bytes or 16 bits) main memory. There is also 1,536 16-bit words of 400 nanosecond read-only memory that is used to implement the standard and optional instruction sets. The read-only memory is easily programmed without special equipment, and the manufacturer facilitates this kind of use by providing documentation, an assembler and a simulator for the microinstructions. While we are obtaining an extra read-only memory board and are studying the possibility of implementing special instructions, no specific advantage for this project has yet been identified. The standard instructions and data formats are similar to those of the System 360, and provide a convenient set of operand-register, register-to-register and register-to-indexed-core operations. A register-to-register add is executed in 3.2 microseconds, and a register-to-indexed-core add requires 6.0 microseconds.

To the multiplexor channel of the Model 4, we attach a controller and teletype for console operations, and a bit oriented multiplexor (BIM) that provides ports for 32 Data Line Units, each of which may in turn be connected to a slow speed device. Initially, just three of the ports on the BIM are utilized. One is used for a 403D8 Modem that handles a telephone line for touchtone input. A second one is used for the voice answer back device via its interface. And the third port is used for a local teletype that can simulate additional touchtone inputs and/or voice outputs.





31-Word Cognitronics Voice Response System

A Cognitronics "Speechmaker Model 632" is used in the initial configuration to generate audio responses. This device uses 31-word vocabularies that are stored on easily interchangeable photographic film drums. A binary decoding matrix is used to select one of the 31 words (or silence) on the basis of a 5-bit code. Reading is accomplished by passing a light beam through one of the 31 information tracks on the drum. The beam is moderated by the recorded audio signal and is picked up by silicon photo sensitive cells that are located inside of the rotating drum. The output of the photo cells is then amplified and passed on to a telephone line.

Once each 625 milliseconds, the Speechmaker generates a timing pulse. The system then has 25 milliseconds in which to present the code of the next word or silence. That address must then remain in place throughout the 600-millisecond playing period. At the end of the playing period, the next timing pulse is generated.

Recording sessions represent the primary expense involved in producing a vocabulary; and thus, an original vocabulary costs almost six times as much as a vocabulary that uses words that are already present in the manufacturer's files. So, to economize while experimenting with vocabulary, only words that have been previously recorded were chosen for the initial vocabulary. The words were recorded with a feminine voice and are centered within the 600-millisecond time period.

The ten digits: "ZERO," "ONE," "TWO," "THREE," "FOUR," "FIVE," "SIX," "SEVEN," "EIGHT" AND "NINE" are used in conjunction with "POINT and "MINUS" to generate numeric answers.

Examples: "ONE-TWO-FIVE"
"SEVEN-POINT-TWO-THREE"
"MINUS-SIX-ZERO-POINT-THREE"

The digits are also used to enumerate certain identities (i.e., "ITEM," "NUMBER," "LETTER," "ERROR," "LINE," "FILE" and "NAME"). In certain cases, this is as we would want it. In others, it is simply a substitute for a larger vocabulary.

Examples: "LINE-FIVE"
"LETTER-TWO-SIX"
"ERROR-FOUR-TWO"

"ANSWER," "IS," "READY," "NOW" (with terminal inflection), "WAIT," "GOOD-BYE" (with terminal inflection) and "REPEAT" are used as cues, while "OF" and "AND" are used as connectives.

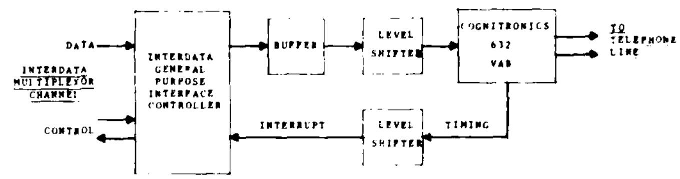
Examples: "ANSWER-IS-EIGHT-FOUR"
"READY"
"ITEM-TWO-OF-FILE-FOUR"
"TWO-THREE-AND-SEVEN-SIX"

Finally, "YES," "NO" and "NOT" round out the initial vocabulary.

Examples: "YES"
"NOT-FILE-SEVEN"

The interface between the Model 4 and the Speechmaker was designed for us by the Houston Research Institute and allows the system to treat the Speechmaker just as it would an ordinary teletype with the single exception of timing pulses which are connected to an interrupt line.

For the future, much larger vocabularies and multiplexing capability will be required.



INTERDATA COMPUTER COGNITRONICS 632VAB INTERFACE

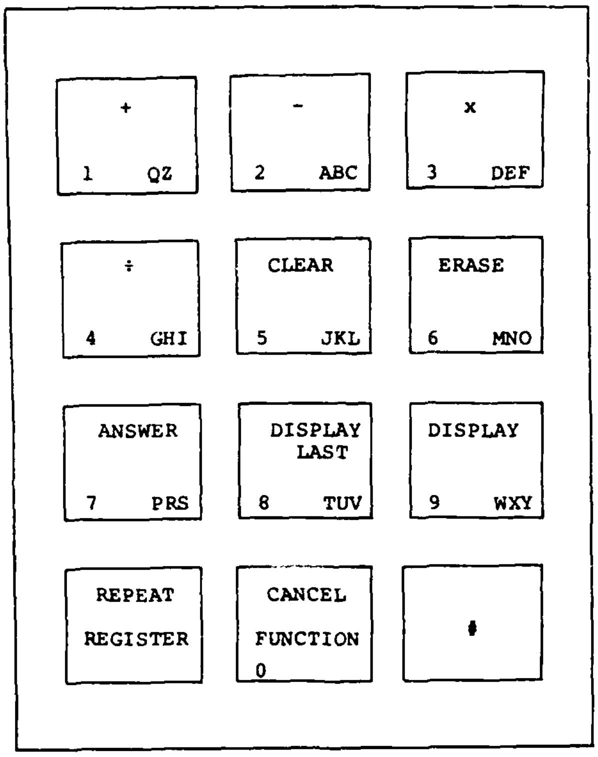
Furthermore, practical use of several of the suggested applications will require additional core memory and a high speed communications link with a large scale digital system. Local mass storage may also be required.

The total value of our initial hardware system is less than $40,000.

A desk calculator

As a first possibility for application, we have considered the use of the touchtone telephone as a desk calculator. The design calls for the standard arithmetic operators, a wide range of mathematical functions, control over the arithmetic processes, a variety of control operations and a comprehensive set of readout options.

In the following, the syntax and semantics of a desk calculator language will be described in some detail. While it may be of interest to some in its own right, the primary intent is to illustrate what can be done, language-wise, with the twelve buttons and a minimum number of depressions. For each unit of the language, a formal description of the syntax is followed by an informal description of the syntax and semantics. A few examples are also included. The figure below may be consulted for the position of each syntactical constant (button) on the keyboard.



A calculating sequence:

calc-seq: = $_1${operation} function EXIT #

A calculating sequence is defined to be a sequence of operations (at least one) terminated by consecutive depressions of the six buttons: function, E, X, I, T, #.

The processing of a calculating sequence is initiated by the entry of an appropriate "mode" sequence. When the end of the calculating sequence is recognized, control is returned to the operating system for a new mode indication or for "sign-off".

The operations:

```
operation: = add|subtract|multiply
             |divide|clear|erase
             |answer|display-last
             |display|repeat|cancel
             |function
```

There are eleven basic operations, each beginning with a distinct one of eleven buttons. A twelfth basic operation, "cancel", (same button as "function") is distinguishable from "function" since it is always followed immediately by a "#". All operations end with the twelfth button, #. Extension capability is obtained via function.

Addition, Subtraction, Multiplication and Division:

```
     add:= + {value #|# function}
subtract:= - {value #|# function}
multiply:= x {value #|# function}
  divide:= ÷ {value #|# function}
```

Examples:

```
+  2.57 #
÷  1.32146 #
x  # function SIN # 1.5 #
-  # function PI #
-  register 3 #
```

Any of the four arithmetic operators may be followed by a simple value (a constant or the contents of a register) or by a function value. In either case, the appropriate operation is performed with (1) the indicated operand, and (2) the contents of the "active register". The result of the operation is then placed into the "active register".

Clear:

clear: = clear [register] #

Examples: clear register 2 #
clear #

To perform a "clear" operation, one depresses the clear button followed by either the terminator (#) or a register indication. If a register is indicated, that register is cleared to zero and becomes the "active register". Also, all previous operators and their operands are erased from the "record". That is, the historical record of the operations most recently performed (capable of being played back) is reinitialized.

If no register is indicated, all ten registers are cleared and Register 1 becomes the active register by default. The record of operations performed is reinitialized as in the previous case.

Erase:

erase: = erase #

Example: erase #

This operation may be used to delete the last arithmetic operation (addition, subtraction, multiplication, division or arithmetic function). This means that the operation will have had no effect on any register, and is removed from the record.

Answer:

answer: = answer [register] #

Examples: answer #
answer 9 #

If a register is given, the value currently in that register is played. Otherwise, the contents of the active register is played.

Display-last

display-last: = display-last #

Example

display-last #

This operation results in a playback of the arithmetic operation most recently performed.

Display:

display: = display #

Example display #

In this case, the request is for a playback of the current record of arithmetic operations; that is, those that have been entered most recently. The maximum length of the record is dependent upon the buffer size selected.

Repeat

repeat: = repeat #

Example: repeat #

The last message played is repeated as a consequence of this entry.

Cancel

cancel: = cancel #

Playing of the current message is terminated.

Function

function: = function ${}_{1}^{8}${letter|digit} #

${}_{0}${value #}

Examples: function MAX # 2 # 8.75 # register 2 #
function X #
function MOD # register 3 # 4 #
function SKIP #

A function entry consists of a depression of function followed by a function name that may consist of up to eight letter or digit depressions. Function names must, of course, be unique. After the function name and its terminator, there may be a sequence of operands separated by terminators. As usual the operation ends with a terminator.

Functions may or may not have values. If a function does not have a value, it may still have an arithmetic effect on a register(s) (e.g., complement the active register), or it may simply initiate a control operation (e.g., function EXIT). When a function does have a value, it must be used as the operand of an arithmetic operator (e.g., + # function COS # register 2 #).

Five control functions, in addition to "EXIT", are included in the basic set. Each of these functions will be discussed in the paragraphs immediately following.

"ATROL" is a control function that is used to adjust the parameters that control arithmetic operations and the presentation of results. Its single argument is a string of characters that indicate (1) the number of significant digits to be carried, (2) the position of the decimal point, and (3) the roundoff convention. For instance, function ATROL # XXX.XR # calls for five significant digits with the decimal point located between the third and fourth digits from the left. The right most digit is to be rounded upward only if the digit that would have followed is five or greater. This particular setting is chosen by default if the user does not make a setting with ATROL. If the right most character is "U", results are always rounded upward; and if the last character is "D", results are always rounded downward (truncated).

"DEFINE" is a control function that may be used to preserve a sequence of operations for future use. Its single parameter is the name that is to be associated with the preserved sequence of operations. The sequence that is preserved begins with the first operation following the "DEFINE" and continues until a matching "END" function is encountered.

```
Example:      +     2.8      #
              function DEFINE # CUBE #
                        clear register 0 #
                            + register 2 #
                            x register 2 #
                            x register 2 #
                       answer #
              function END #
              function CUBE #
               + register 0 #
```

In this example, "CUBE" is defined as a sequence of operations that uses Register 0 to calculate the cube of the value in Register 2. It also plays the resulting value. The definition sequence, "DEFINE" through "END" does not result in an execution of the defined sequence, but simply records it for future use during the current calculating sequence. In the example, the two operations following the "END" do invoke the sequence just defined and utilize the result. Systems maintenance is required in order to make a sequence available after the current calculating sequence is concluded.

"LOOP" is a control function that may be used to call for the repeated execution of a sequence of operations. Such a sequence begins with the operation just following a LOOP function and continues until a matching END function is encountered. The looping is controlled on the basis of four operands that must be provided along with the LOOP function. The first operand indicates which register is to be used as the counter for the iteration process. The second operand gives the initial value for the counter; the third operand gives the increment to be added at the beginning of each successive iteration; and finally, the fourth operand gives the terminal value; that is, the value that is not to be exceeded.

Example:

```
clear #
function LOOP # register 8 # 1 # 3 # 9 #
          +  register 8 #
          answer  #
function END #
```

In the case of this example, the loop would be executed three times. On the first iteration, the counter register, Register 8, would have a value of 1; on the second iteration, a value of 4; and finally, a value of 7. A fourth execution of the loop would not occur since the next appropriate value, 10, is greater than the given terminal value, 9.

"TEST" results in a comparison of its two operands. If they are equal, all operations are skipped until a matching "END" is encountered.

Example:

```
function TEST # register 2 # 6 #
     +    6.0  #
function END #
```

Here, the "+" operation between the "TEST" and the "END" will be carried out only if Register 2 does not have a value of 6.

Value:

value: = number|register

Values are used as operands for arithmetic operations and functions.

Number:

$$\text{number:} = {}^{16}_{1}\{\text{digit}\}_i \; [.{}^{16-i}_{0}\{\text{digit}\}]$$

Examples: 1685324

231.7

0.6419

2.

A number may consist of up to sixteen digits with a decimal point between any two digits or after the rightmost digit.

Register:

register: = register digit

Examples: register 2

register 5

The calculator is designed with ten registers (1, 2, ..,9) that may be used for arithmetic purposes.

Registers are either active or inactive. When a register is active, its value is normally one of the operands involved in an arithmetic operation. Whether inactive or active, the value in a register may be played or used as an operand.

A Sample Calculator Sequence

clear #	All ten registers are cleared, and by default, Register 1 is selected
+ 2.5 #	2.5 is added into Register 1
- 1.2 #	1.2 is subtracted from Register 1
display #	"+2.5-1.2" is played
erase #	The last arithmetic request (-1.2) is erased, and thus Register 1 = 2.5
- 1.4 #	1.4 is subtracted from Register 1
display #	"+2.5-1.4" is played

display-last #	"-1.4" is played
answer #	"1.1" is played
repeat #	"1.1" is played again
clear register 3 #	Register 3 is cleared and selected
+ 2.0 #	2.0 is added to Register 3
x 3.8 #	Register 3 is multiplied 3.8
÷ 1.5 #	Register 3 is divided by 1.5
answer #	"5.06" is played
+ register 1 #	The contents of Register 1 (1.1) is added to Register 3
display #	"2.0 x 3.8 ÷ 1.5 + Register 1" is played
repeat #	"+2.0 x 3.8 ÷ 1.5 + Register 1" is played
answer #	"6.16" is played
answer register 3 #	"6.16" is played
answer register 1 #	"1.1" is played
repeat #	"1.1" is played
+ # function PI #	The value of the pseudo function PI(3.14...) is added to Register 3
function X # 3 #	The function X causes the active register to be multiplied by the negative of the operand
answer #	"-58.1" is played
function EXIT #	This function effects an exit from the calculator mode

Other applications

Principles and indeed, much of the design and implementation from the desk calculator application can be applied directly to wide variety of applications. Here, the beginnings of a list of such applications will be presented. The ordering of the list has no significance.

A wide variety of information storage and retrieval applications can be handled in an effective fashion by a touchtone telephone system. For instance, a classified advertising service can allow businesses and/or citizens to enter information about items that they wish to sell. On the other hand, inquiries can be made about the availability, specific characteristics and prices of items that are on the market. And, as an important byproduct of the entry/inquiry process, statistics can be gathered on the nature of consumer interests. An example of a special purpose system of this type is the real estate system that has been developed by the Realtron Corporation of Detroit, and is operational in several parts of the United States. Style, price range, number of bedrooms, location and up to fifteen other preference items are entered into the system via touchtone telephones. The Multiple Listing Service file is then searched, and code numbers for appropriate homes played back over the telephone. The code numbers are then used to locate the files that correspond to the selections. Information concerning current real estate prices is also gleaned from the system.

An entry/inquiry system can also be used as a sales aid in branch stores or, for that matter, in the customer's own office and/or home. Inquiries, can be made, orders placed, and confirmations obtained; all from most any location, and without the need for human attention at the central location.

Also, a very similar entry/inquiry system can be used for a variety of management information applications. One such system, developed by North American Aviation in cooperation with IBM and General Telephone, preceded touchtone telephones, and allowed engineers to dial a code number followed by the number of an engineering drawing. A file of some 75,000 drawings were searched, and the latest changes listed via audio response.

For the small business, a set of very valuable services can be provided at a modest cost. Inventory control, accounting,payroll, data analysis and projections, cost estimating and bidding and mailing list maintenance are all feasible.

A touchtone telephone system can also be a very valuable programming aid. One project, "Tele-CUPL" seeks to provide the touchtone user with a rather general purpose conversational programming system that is based upon the Cornell University Programming Language. And, indeed, a quite acceptable syntax has been developed in spite of the small number of buttons available. Nevertheless, it is doubtful that a great deal of programming will ever be done from such a terminal. But inquiries about the status of jobs, relatively minor updates to program and data files, as well as requests for executions, can all be handled in excellent fashion from most any location and at any time.

Finally, there are many special services that can be provided to various categories of users. Assistance in tax reporting, calculation of menu proportions and check account balancing, just to mention a few, can be added to those applications already mentioned for the home. For the advertising and public relations business, public opinion and preference surveys will be facilitated in significant measure. And stock quotations are already a reality, at least for a limited circle of users.

Software

With the very limited number of unique codes that are available for data input from a touchtone telephone, the development of practical syntax must be of serious concern; and this is particularly true if a large proportion of the input data is necessarily alphabetic and arbitrary. Fortunately, we have found that, for the kind of applications mentioned above, it is seldom necessary to allow arbitrary alphabetic data. For the most part, it can be avoided by giving careful consideration to syntactical patterns and coding choices. When it is necessary to allow arbitrary alphabetic information, one must simply resort to some sort of a multiple despression coding system. Our current choice involves the use of two depressions for each character, with the digit indicated by the second depression selecting from the characters associated with the first depression.

In designing the programming support for the hardware configuration and set of applications mentioned above, a very simple approach has been chosen. First of all, the nature and price of the hardware has led to the conclusion that a dedicated hardware-software system is both economical and desirable. Secondly, the nature of the applications has encouraged us to assume that all users can be assigned an equal priority; and that the segments of computer time and memory space required by any given user are extremely small, and approximately equal to those required by any other user. As a consequence of these conclusions and assumptions, input/output processing, interrupt handling, scheduling and core allocation are all greatly simplified. And user identification, accounting and operation selection are all accomplished by the most straight forward kind of a multi-way switch, interpretation process.

Conclusion

We find the prospect of providing significant data processing services to wide classes of users at very low prices, to be a very exciting idea and a very practical objective.

References

Conway, R. W., and Morgan, H. L. "Tele-CUPL: A Telephone Time-sharing System," Communications of the ACM, 10, 9 (Sept. 1967), pp. 538-542.

Davidson, L. A. "A Pushbutton Telephone for Alphanumeric Input," Datamation, 12, 4 (April 1966), pages 27-31.

"Engineering Dial Computer to Get Apollo/Saturn Changes," Datamation 12, 2 (Feb. 1966), p. 79.

"How Students Use IBM's Experimental Home Computing System," (Press Release), IBM Corporation Advanced Systems Development Division (April 13, 1966).

Marill, T., Edwards, D., and Freurzeiz, W. "Data Dial: Two-Way Communications with Computers from Ordinary Dial Telephones," Communications of the ACM, 6, 10 (Oct. 1963), pp. 622-624.

Martin, James Design of Real-Time Computer Systems, Prentice-Hall, 1967.

Meacham, L. A., Power, J. R., and West, F. "Tone Ringing and Pushbutton Calling," Bell System Technical Journal, No. 37 (1958), pp. 339-360.

"Telephone I/O," Datamation, 12, 7 (July 1966), p. 89.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) University of Houston	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP Unclassified
3. REPORT TITLE "On the Use Of Very Low Cost Terminals"	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Technical Report	
5. AUTHOR(S) (First name, middle initial, last name) Albert Newhouse Robert A. Sibley, Jr.	
6. REPORT DATE March, 1969	7a. TOTAL NO OF PAGES 23 — 7b. NO. OF REFS 8
8a. CONTRACT OR GRANT NO N00014-68-A-0151 b. PROJECT NO	9a. ORIGINATOR'S REPORT NUMBER(S) RE 3-69 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
10. DISTRIBUTION STATEMENT Reproduction in whole or in part is permitted for any purpose of the United States Government. This document has been approved for public release and sale; its distribution is unlimited	
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY ONR-Washington

13. ABSTRACT

This effort seeks to maximize the number of places (cities, buildings, rooms, desks,...) from which people can interact with digital computers in a meaningful way. Because of the extraordinary availability and low cost of the ordinary telephone, it was chosen as the cornerstone of the project. A set of possible applications were then identified. These applications include desk calculator services (scientific and business), information storage and retrieval (public assistance and private industry); and computer program development. For each category of applications, the identification of specific services is followed by a study of input/output language problems, economic considerations and human factor problems.

These studies have involved the design and implementation of a specific hardware/software system. Telephone signals are interfaced and multiplexed into an Interdata Model 14 computer in the same manner as teletype messages. Output is generated by a Cognitronics Speechmaker as a consequence of selection codes that are presented to a specially designed buffer. Requirements for large data collections and computational services are satisfied by a high speed connection to the University's central computing facility.

DD FORM 1 NOV 65 1473 (PAGE 1)
S/N 0101-807-6801

UNCLASSIFIED
Security Classification

UNCLASSIFIED
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Terminals Telephone Applications Audio						

DD FORM 1 NOV 65 1473 (BACK)
(PAGE 2)

UNCLASSIFIED
Security Classification