Multi - Utility Telephone System

PROJECT WORK

Done by

S. ARUN KUMAR

A. S. SENTHIL KUMAR

M. VINOD KUMAR

Under the guidance of

MR. G.C.THIYAGARAJAN, M.E.

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE & ENGINEERING
Of Bharathiar University, Coimbatore.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641 006.

MARCH 2002

ACKNOWLEDGEMENT

We express our deep sense of gratitude to ***Dr.K.K.Padmanabhan Ph.D., Principal, Kumaraguru College of Technology, Coimbatore***, for providing us permission to carry out this project work.

With profound sense of gratitude and regards, we acknowledge with great pleasure the guidance and support extended by ***Dr.S.Thangasamy Ph.D., Head of the Department of Computer Science and Engineering*** ,for his valuable and continuous guidance, suggestions, constructive criticisms and persistent encouragement.

We extend our heartfelt thanks to our Assistant Professor ***Mrs.S.Devaki M.S.,*** for providing us with valuable suggestions and help whenever required to make our project successful.

We express our deep sense of respectful gratitude to our guide ***Mr.G.C.Thiyagarajan M.E., Lecturer, Department of Electronics and Communication Engineering*** for his valuable guidance, keen suggestions innovative ideas, inspiration, discussions, helpful criticisms and kind encouragement in all the phases of this project work. It has been a great pleasure to work under his guidance.

We like to express our special thanks to all staff members and Lab technicians in the Department of Computer Science and Engineering and in the Department of Electronics and Communication Engineering who helped us for the successful completion of the project. Finally we express our deep sense of gratitude to our parents, friends and all other who had been directly or indirectly involved with this project, for their invaluable help and consideration towards us.

SYNOPSIS

Multi-Utility Telephone System is a telephony application project that enhance easy use of telephone system through the computer. The objective of the project is to make use of his telephone system when the user is out of station. It is the MUTS software that makes possible this goal.

MUTS software package consists of two major parts. One is the MUTS home software that is installed in the home computer which is connected along the telephone line through the modem. Another is the MUTS remote software that is installed in any remote computer connected along a telephone line through modem which makes the owner possible to communicate to home MUTS software.

Any calls coming to home when the user is out of station was answered by the Home MUTS software. It stores the called person's phone number, date and time at which the call is made. When the owner contacts the home MUTS software through the remote MUTS software, the home MUTS software supplies with all the information that is stored in the Home Computer. Another utility provided is the home automation. The owner can issue commands through the remote MUTS software to the home MUTS software that instructs the Home computer to control external appliances that are connected along the parallel port.

Thus, the Home MUTS software, the Remote MUTS software and the hardware appliance control circuit connected along the parallel port for automation makes possible to reach our project target.

CONTENTS

1. INTRODUCTION

Communication has become a Herculean task nowadays. The technologies available today shrink the world and make it still smaller. Although the technologies improve day to day, the usage of the resources available is limited to certain constraints and also the trends in technologies requires more of investigation and investment that the previous ones.

MUTS- Multi Utility Telephone System makes use of the available telephone line for its communication. Although people use telephone line only for voice communication, we have tried to have a better usage of the available resources there by achieve certain features which may be difficult to perform in a single system.

Communication has become a part of our daily life and makes man hours shorter and sweeter. It has achieved a lot within a few decades and has reached greater heights. Our project also tries to contribute its part to the existing communication world by using the available resources and make better use of it.

1.1 EXISTING SYSTEM

The existing caller identification system have certain functionalities which are limited to its usability. The existing system is capable of trapping caller information and stores them in the device along the caller ID machine. The information stored in the existing system can be viewed only by the user only when he returns home. The present telephone utility system can be used along with a separate answering machine wherein the voice of the calling person can be stored after an intimation from the answering machine kit. The voice of the caller is stored and then it can be played only when the user comes home and operates the system. It is not possible to connect to the kit from the remote and hear the voices or messages of the called person.

Thus, though the present systems are capable of clipping caller identification number, it is possible to view or go through the information only when the user goes to the current system and travel through the system manual. It is not possible to connect it from the remote and access it .Also certain software's available are not compatible with the ports available and those systems cannot be reconfigured in future if there is any need to change the settings and configuration of the system . In order to resolve all the above conflicts and disadvantages we planned to come out with an enhanced, yet user friendly product which would make the user more comfortable and feel easier to use the software thereby to go along with the telephone communication line.

1.2 PROPOSED SYSTEM AND ITS ADVANTAGES:

MUTS focuses entirely new idea of making use of the Telephone at home when its user is out of station. The utility provided are like never before. The concept of controlling home appliances from remote system and viewing information of missed calls makes the proposed system an outstanding. The important thing is all these utilities are performed by making a simple phone call.

MUTS is totally different from the products existing in the market . The system is built with certain features which make more user-interactive and also provides high end processing for the users.

MUTS uses a software at the home computer which is made to run wherever the user or the administrator of the system is out of home. The product is made to auto execute whenever the system is under operation thereby he software is capable of monitoring the communication path and wait for any requests or any calls from the remote. It continuously checks for any requests from the communication port and is will be ready to answer any requests from the remote user.

The modules of our proposed system includes

1. Caller information Storage
2. Retrieval of stored information from the remote.
3. Automation of home appliances and control from the remote system.

The Caller identification part of the software is quiet tedious where the system has to monitor the communication path continuously and store the information of the Caller with the Callers numbers, date and time at which the caller has communicated with the home system. The caller information is stored in a log file and is capable of retrieving it from the remote system after undergoing authentication and verification of the user. Users other than the administrator cannot get into the system and access the files. Retrieval of information from the home system to the remote system is made possible in our project where no other system have this feature inbuilt. The existing system neither use telnet or ftp to retrieve the data- just the existing Telephone line. Here the system will be capable of retrieving or transferring the information to the remote system through the already established communication path after authentication.

The Automation of Appliances at home through the computer system is totally a new feature incorporated in our project. The home system is connected with our control kit wherein the appliances are connected according to the specifications. In the present system , the appliances may be connected to either a separate control kit or through the available telephone. Here, the system will be automated to control the appliances at the remote thereby reducing the problem of using unnecessary , yet costlier control kits. Our system makes use of the available parallel port in the computer and can control at the minimum of eight appliances when directly connected. We can go for more appliances to be controlled by using simple decoders. MUTS will be available with a simple electronic circuit which performs the operation of switching between the computer and the home appliances.

Some of the supplementary utilities makes user capable of talking with the caller when the user is at home either by using the parallel connection of the telephone set or by using a microphone and a speaker.

2. SYSTEM REQUIREMENTS

2.1 PRODUCT DEFINITION:

Multi utility telephone system is a telephone based communication software which is capable of clipping caller identification number when the user gets any call from the remote. The software is also capable of storing the date and time at which the call is obtained, along with the caller ID. MUTS have three main units namely,

1. The home unit stores the caller ID number and information for the calls that are unanswered by the user
2. The Remote unit which the user makes use of it from the remote to retrieve the information from the home unit.
3. Home Automation system where the electrical appliances in the house can be controlled from the remote system.

The unit at the either end has a dialer, which is a simulation of the real telephone. The user can make use of the dialer software and call any person through the telephone communication line. These are a supplementary utility provided.

The home appliance control device, attached along the home computer system makes use of the parallel port and the user at the remote side sends signals which acts as a switch from the remote and the user could control the appliances effectively in the way he wishes.

The user at the remote system also can send data or message to the home system thereby making the others in the house to know the whereabouts of the user.

2.2 PROJECT PLAN:

2.2.1. Software Feasibility:

The MUTS software is highly feasible due to the following consideration.

1. Required Resources - Telephone lines available, can be connected to the computers at both ends along with the modem.
2. Project members capable of solving the modules.
3. Study materials and other help tools available.
4. The software can be built by using the existing software technology.

2.2.2. Resources :

1. Human Resources

   Efficient and hardworking persons three in number are capable of performing project modules.

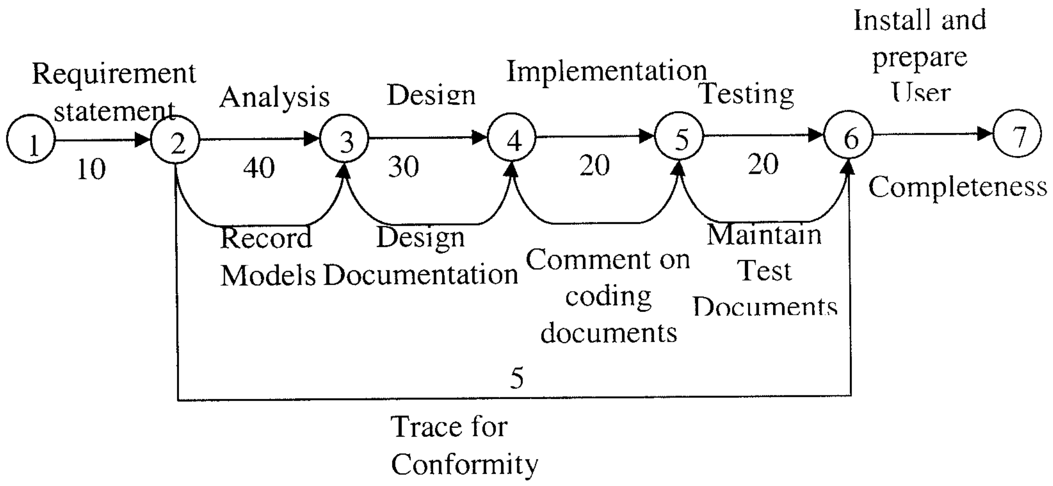
2. Technical Resources

   Two telephone lines dedicated for communication between the systems.

   Two modems attached along the system which acts as an interface between the telephone line and the computer system.

2.2.3. Project Scheduling:

The following project network is constructed by PERT technique(Program Evaluation and Review Technique).

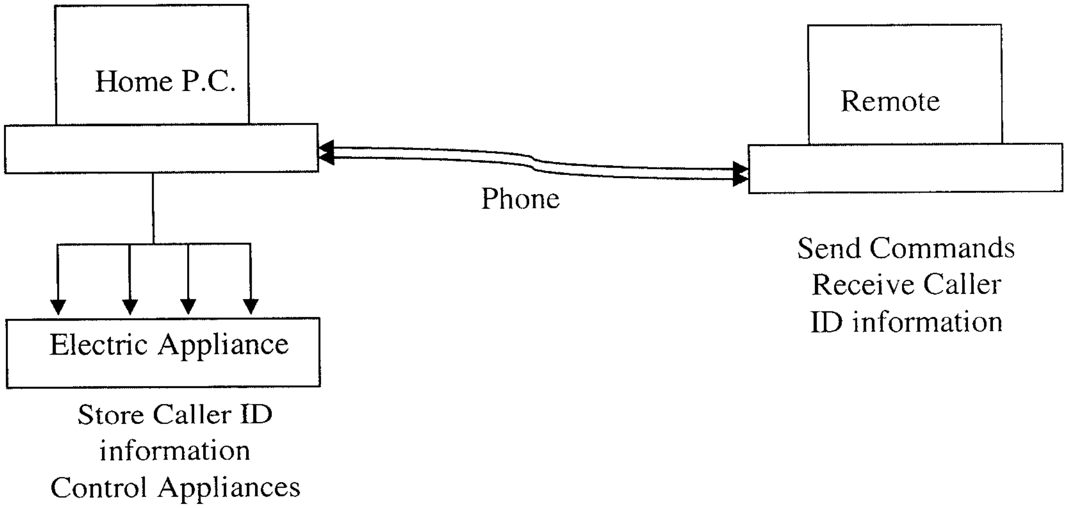


3. SOFTWARE REQUIREMENT SPECIFICATION

3.1 Requirement Definition :

1. Software that can store caller ID , date and time of call , when a call is made to home , and when owner is out of home.
2. Suitable software that enables owner or software user to read the stored information.
3. Automation Utility, that enables the user to switch ON or OFF the external electric appliances at home.

3.2 System Model :



3.3 System Evolution :

1. Both the Home p.c. and remote p.c. are connected with a phone line through a modem for communication.
2. Both the system are capable of sending and receiving data.
3. The Home p.c. is always powered ON (only the processor and not the power consuming monitor)

3.4 Functional Requirements:

1. Home p.c. connected to the phone line should be capable of answering all incoming calls.
2. Authenticate the callers. If the particular call was not from the owner or the user, the Home p.c. should record the callers phone number, date and time at which call is made, in the system.
3. For the owners call from the remote software, authentication is done and the call let in.
4. Remote computer issues commands for automation at home.
5. Home computer sends remote p.c. with the information required.

3.5 Non-Functional Requirements :

1. Authenticate every call.
2. Respond to the commands if it is owners call.

3. Cut the call after storing caller information when it not the owner or the user.
4. The Home p.c. software should reset automatically when a call is answered.
5. A suitable uninterrupted power supply(UPS) to be connected to system to provide power backup.

3.6 Hardware Requirements:

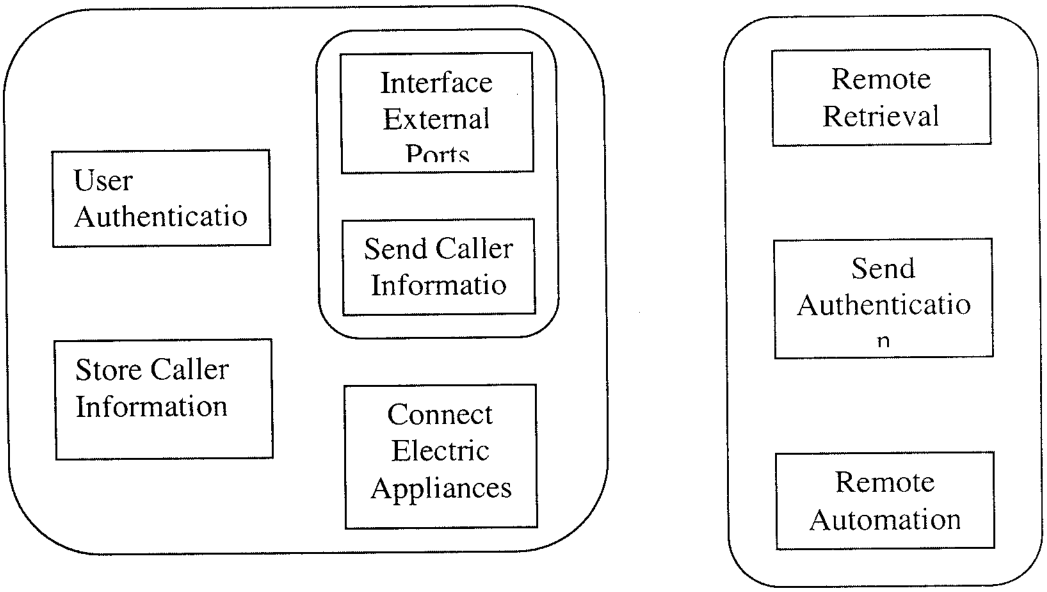
1. Circuit capable of switching external devices.
2. This circuit does not cause any side affect to the existing power line.

4. DESIGN DOCUMENTATION:

4.1 View Point Analysis:

From the requirements, we infer the following

.



There can be two view points that can be reorganized.

1. Home side view point

User Authentication

Store Caller Information

Send Caller Information

Interface with External Electric Appliances

2. Remote Side Viewpoint

Send Authentication information

Remote Retrieval

Remote Automation

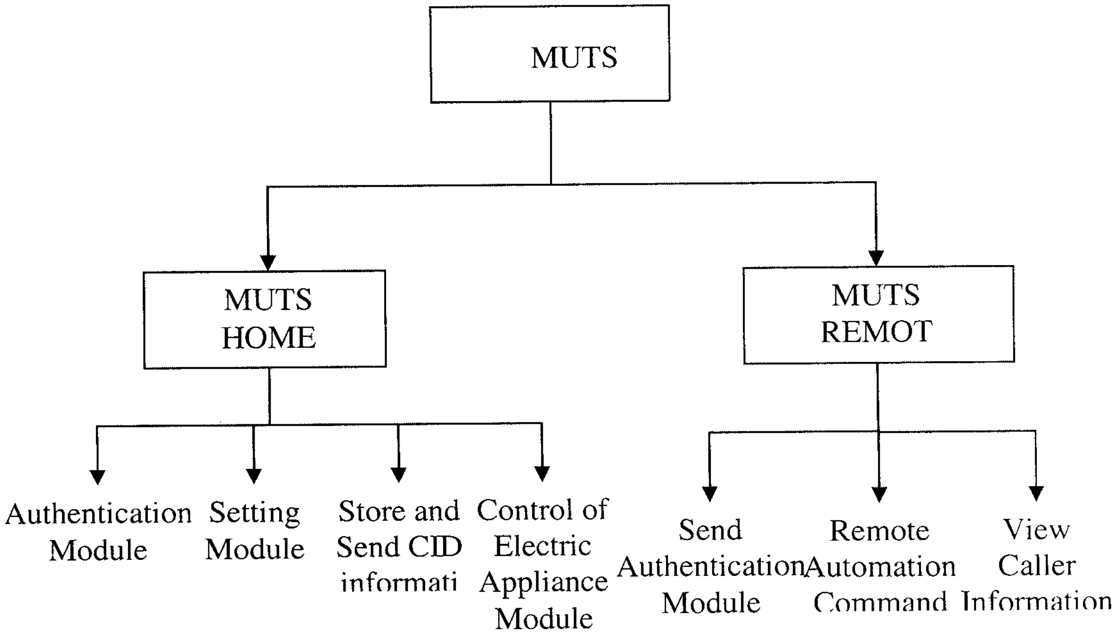
4.2 Modularisation :

The MUTS is divided into two major sub division .They are

1. MUTS Home
2. MUTS Remote

The MUTS Home part of our project consists of Authentication Module for validation of user, Setting Module which initializes the hardware settings, Store and send Caller information Module and finally the control of Electric Appliances module.

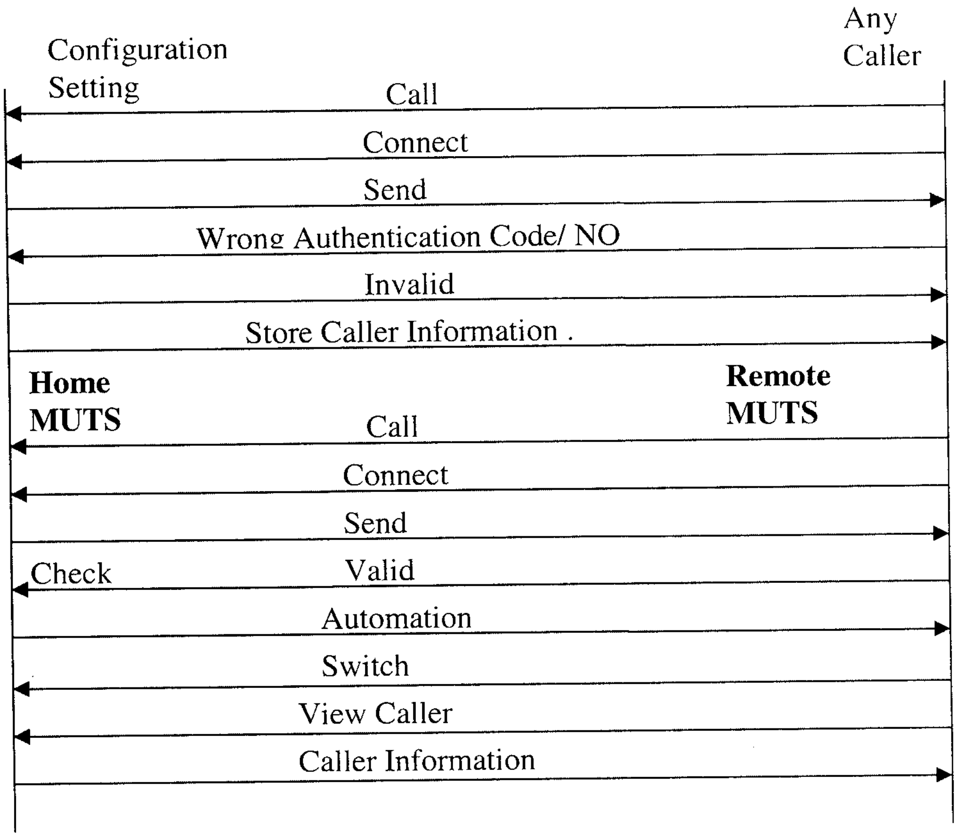
The MUTS Remote part of our project has the Send Authentication Module which requests the user to send authentication code, Remote Automation Commanding module and finally View Caller information module


MUTS
MUTS HOME
MUTS REMOT
Authentication Module
Setting Module
Store and Send CID informati
Control of Electric Appliance Module
Send Authentication Module
Remote Automation Command
View Caller Information

4.3 Dynamic Modeling:

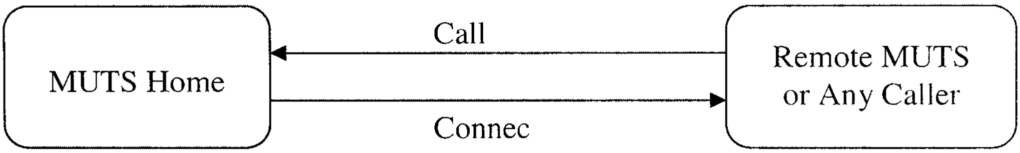
4.3.1 Event trace:

Home MUTS

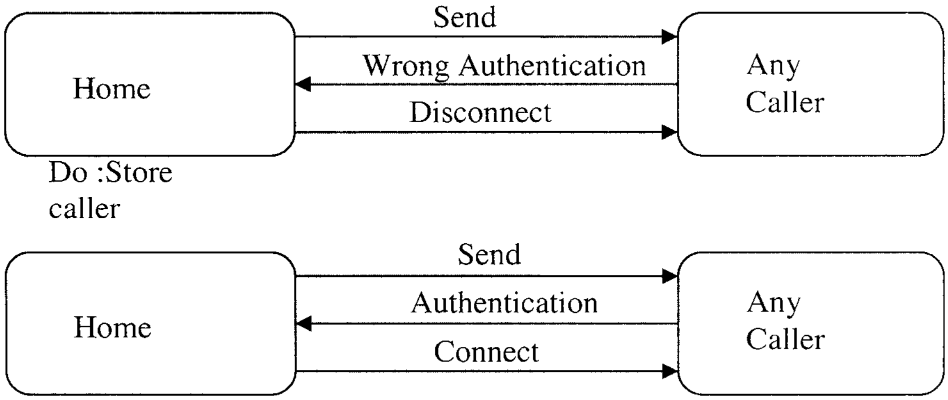


4.3.2 Dynamic Model :

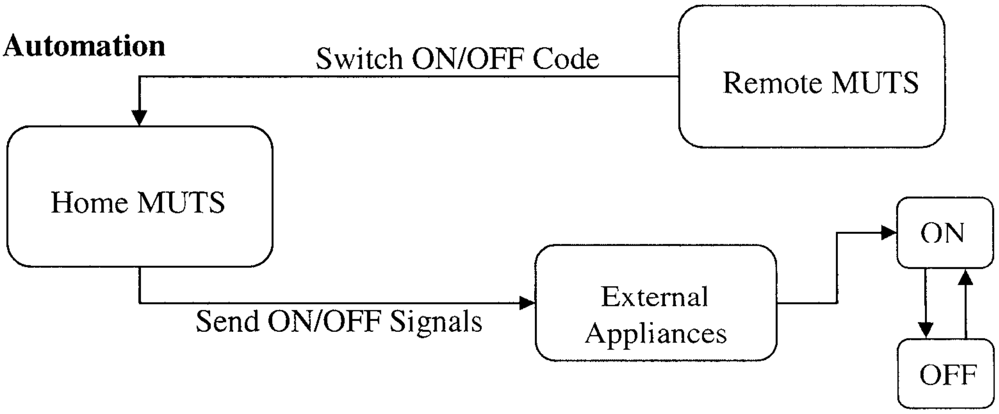
Call Home



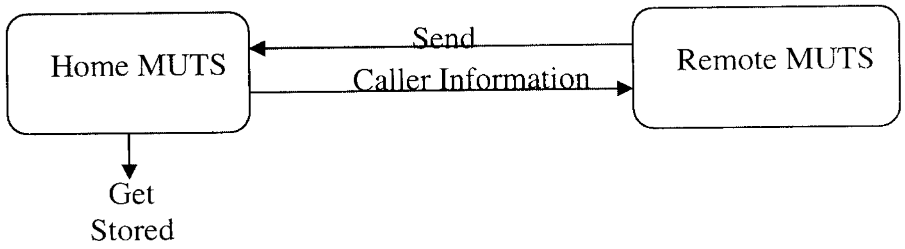
Authenticate



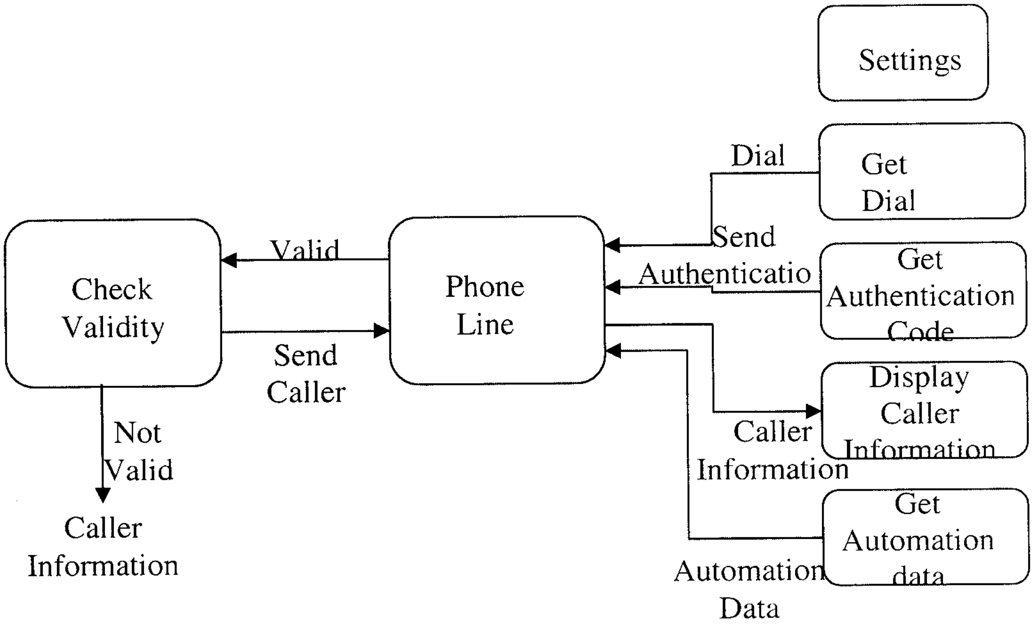
Automation



View Caller



4.4 Functional Model :



4.5 Data Dictionary:

MUTS Home : Software entity, responsible for all the processing at home side.

MUTS Remote : Software entity, responsible for all the operation at remote end.

Dial Number : Number that is dialed.

Connect : Code from phone line after a connection is established.

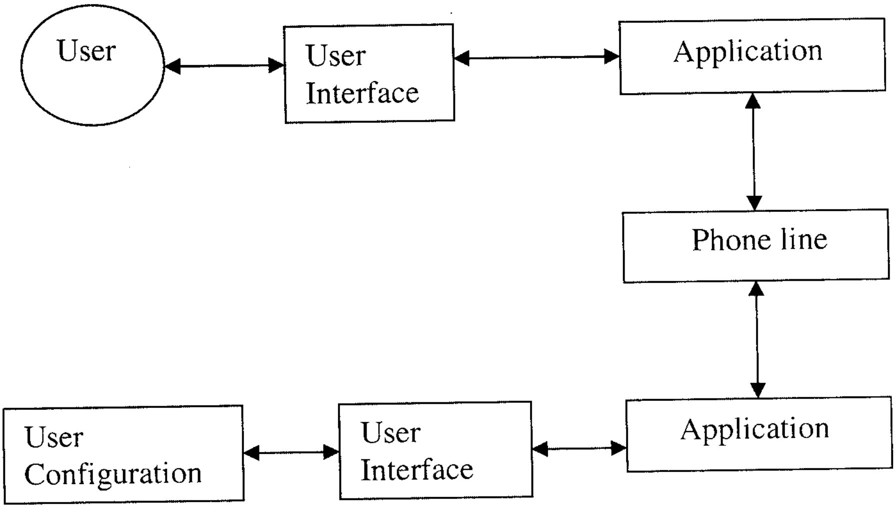
Caller Information : Refers the details of caller other than owner. It includes

1. Phone Number of Caller
2. Date of Call
3. Time of Call

Authentication Code: Code that is sent to the Home MUTS from Remote MUTS for recognizing the owner. Returns Valid (or) Invalid from Home MUTS.

Automation Data : The data passed from Remote MUTS for switching home appliances. Built by Remote MUTS from user interface.

4.6 User Interface Prototyping :



5. IMPLEMENTATION

5.1 Software used

Microsoft Visual Basic 6.0

Operating system: Microsoft Windows 2000

5.2 Hardware used

Computer system – 2 in number with the configuration

Pentium II

64 MB RAM

Functional parallel port in one system

Modems

HSB 56 Internal Modem

Motorola 56k External modem

Telephone line with two terminal connections

For the Interface circuit

Resistors

Capacitors

Diode

Transistor

Transformer

Relays

5.3 Implementation Issues

From the design document, it was worked out to use Visual Basic 6.0 as programming language for developing MUTS. The reasons for selecting software for devising MUTS are as follows

1. Visual Basic provides the component for communicating with the Modem and hence with the Telephone line
2. Visual basic easy and good User Friendly GUI which enables any user to interact with the MUTS software
3. Visual Basic is easy to implement the function-oriented design that is designed.
4. Windows is the most preferred and widely used Operating system in India.
5. Windows supports all the features the Visual basic provides.
6. C is used to interact with the parallel port. This is because C is the closest language to the Hardware components.

The Implementation issues that need serious consideration are as follows:

5.4 Modem overview

Modem is an acronym for Modulator Demodulator. A modem is a device that converts data from digital computer signals to analog signals that can be sent over a phone line. This is called modulation. The analog signals are then converted back into digital data by the receiving modem. This is called demodulation.

A modem is fed digital information, in the form of ones and zeros, from the CPU. The modem then analyzes this information and converts it to analog signals that can be sent over a phone line. Another modem then receives these signals, converts them back into digital data, and sends the data to the receiving CPU The data format used is

Data length	:8
Parity	:None
Stop Bit	:1

The modem used for MUTS are HSB 56 Internal Modem and Motorola 56 K external Modem. Many modems are used to check MUTS is interoperable.

5.5 AT Command set:

Long ago, Hayes set a standard for modem commands with its Smartmodem 300. Most modem manufacturers adopted this command set in order to call themselves "Hayes compatible." The command set used by the Smartmodem 300, as well as most modems today (with a few advanced commands), is known as the AT command set. AT stands for attention, and precedes all (with the exception of A/) commands directed to the modem. For example, when dialing, it is necessary for either the software or the user to issue an ATDT or ATDP command followed by the number and enter. AT tells the modem that it is about to receive a command. DT tells it to dial by tone, while DP tells it to pulse dial. Finally, the modem dials the number given to it after the command. Different modems do have slightly different command sets, but generally most modems follow the standard set by Hayes.

Most of the modem supports the standard and extended Hayes AT command set. The AT prefix (also known as the Attention Code), signals the modem that one or more commands are to follow. These commands are industry standard language used to communicate with the modem. Your modem is always either in the command mode, or the on-line mode. The modem starts up in command mode when it is first switched on. Commands are only accepted by the modem when it is in command mode. Commands input when the modem is on-line, are treated as data, not as commands

Some of the AT commands used are:

Command	Option	Description
A	(none)	Answer incoming call
A/	(none)	Repeat Last Command. Re-issues the previous command to the modem.(Do not press Return; the command executes as soon as the / is pressed.)
D	(none)	Dial a Number. Instructs the modem to dial the telephone number that you enter
E		Echo Asynchronous (Keyboard) Input to Terminal. Determines whether the characters you type at the keyboard are displayed (echoed) to the terminal-emulation window (if it is active) or to the communications application
	E0	Disable
	E1	Enable
H		Hook
	H0	Go on Hook (disconnect from the telephone line; hang up)
	H1	

		Go off Hook (connect to the telephone line)
L		Speaker Volume. This parameter determines the volume, for call-progress monitor only, of sounds such as dialing, ringing, busy, negotiation.
	L0, L1	Low
	L2	Medium
	L3	High
M		Speaker Control
	M0	Off
	M1	On During Training Only
	M2	Always On
	M3	Off during dialing; on during call progress; off during data transfer
O		Return to On-Line Mode. This parameter determines whether the modem initiates a retrain after changing from escape mode to data mode, or after a semi-colon in dial strings.
P		Pulse Dial
T		Tone Dial. This command instructs the modem to use DTMF tone dialing
Z		Reset Modem Parameters to Default Configuration
+ VCID or #CID		Caller ID enable

5.6 Microsoft Communication Control:

The MSComm control provides serial communications for the application by allowing the transmission and reception of data through a serial port. This Communications control allows adding both simple serial port communication functionality to the application and advanced functionality to create a full-featured, event-driven communications tool. The Communications control provides an interface to a standard set of communications commands. It allows establishing a connection to a serial port, connect to another communication device (a modem, for instance), issue commands, exchange data, and monitor and respond to various events and errors that may be encountered during a serial connection. The following are the list of properties that MSComm control have.

Properties	**Description**
CommPort	Sets and returns the communications port number.
Settings	Sets and returns the baud rate, parity, data bits, and stop bits as a string.
PortOpen	Sets and returns the state of a communications port. Also opens and closes a port.
Input	Returns and removes characters from the receive buffer.
Output	Writes a string of characters to the transmit buffer
CommID	Port number in which the Modem is connected

InBufferCount	Returns the number of characters waiting in the receive buffer
InBufferSize	Sets and returns the size of the receive buffer in bytes
InputLen	Sets and returns the number of characters the **Input** property reads from the receive buffer.
OutBufferCount	Returns the number of characters waiting in the transmit buffer. You can also use it to clear the transmit buffer
OutBufferSize	Sets and returns the size, in bytes, of the transmit buffer
PortOpen	Sets and returns the state of the communications port (open or closed). Not available at design time.
RThreshold	Sets and returns the number of characters to receive before the **MSComm** control sets the **CommEvent** property to **comEvReceive** and generates the **OnComm** event. If value is 0 disables receiving.
Settings	Sets and returns the baud rate, parity, data bit, and stop bit parameters
SThreshold	Sets and returns the minimum number of characters allowable in the transmit buffer before the **MSComm** control sets the **CommEvent** property to **comEvSend** and generates the **OnComm** event. The value if 0 disables sending.

The MSComm control is used in MUTS to communicate with the modem and hence the phone line.

5.7 DAO Data Control:

This control is used in MUTS to store the caller information. The caller's phone number, Date of call and time of call are stored. The connectivity is to a text file as it is easy to manage and occupies less memory.

5.8 Interface to External electrical appliances

The external appliances are connected to the MUTS Home computer through the Parallel port. The short description of the parallel port is given below.

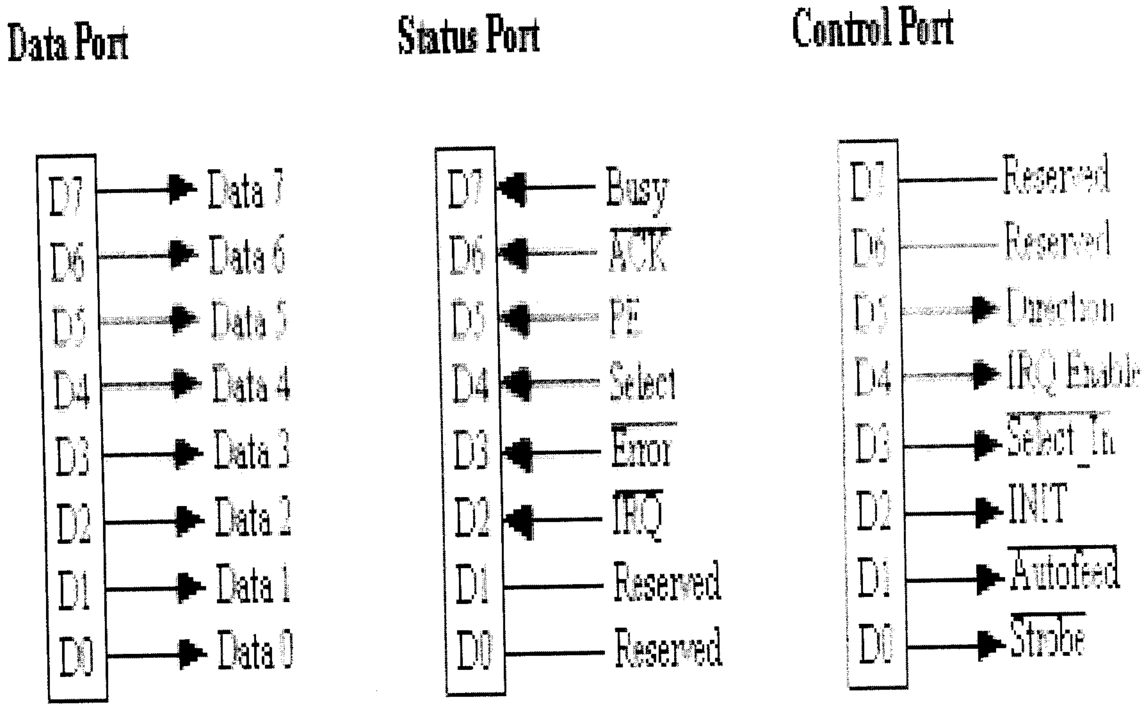
Parallel port

Each printer port consists of three port addresses; data, status and control port. These addresses are in sequential order. That is, if the data port is at address 0x0378, the corresponding status port is at 0x0379 and the control port is at 0x037a.

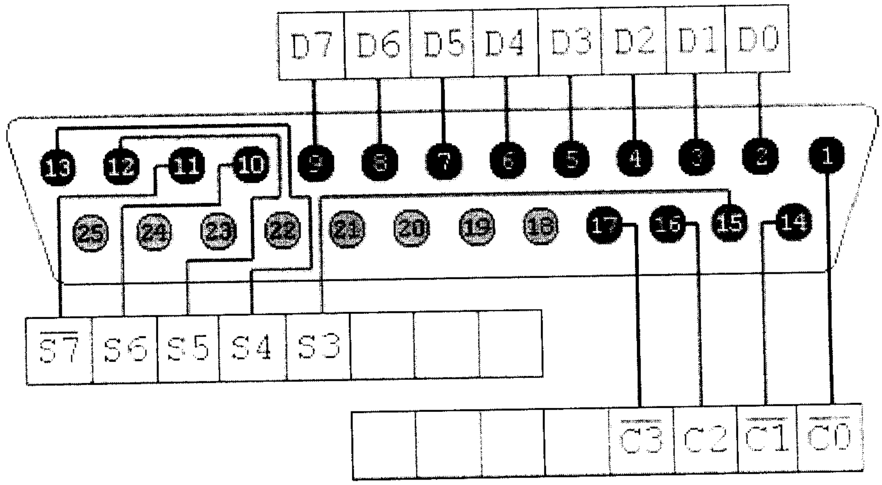
Each printer port consists of three port addresses; data, status and control port. These addresses are in sequential order. That is, if the data port is at address 0x0378, the corresponding status port is at 0x0379 and the control port is at 0x037a.

In MUTS programming LPT 1 is used with the corresponding address.

LPT 1		LPT 2		LPT 3	
408	409	40A	40B	40C	40D
Low byte	High byte	Low byte	High byte	Low byte	High byte

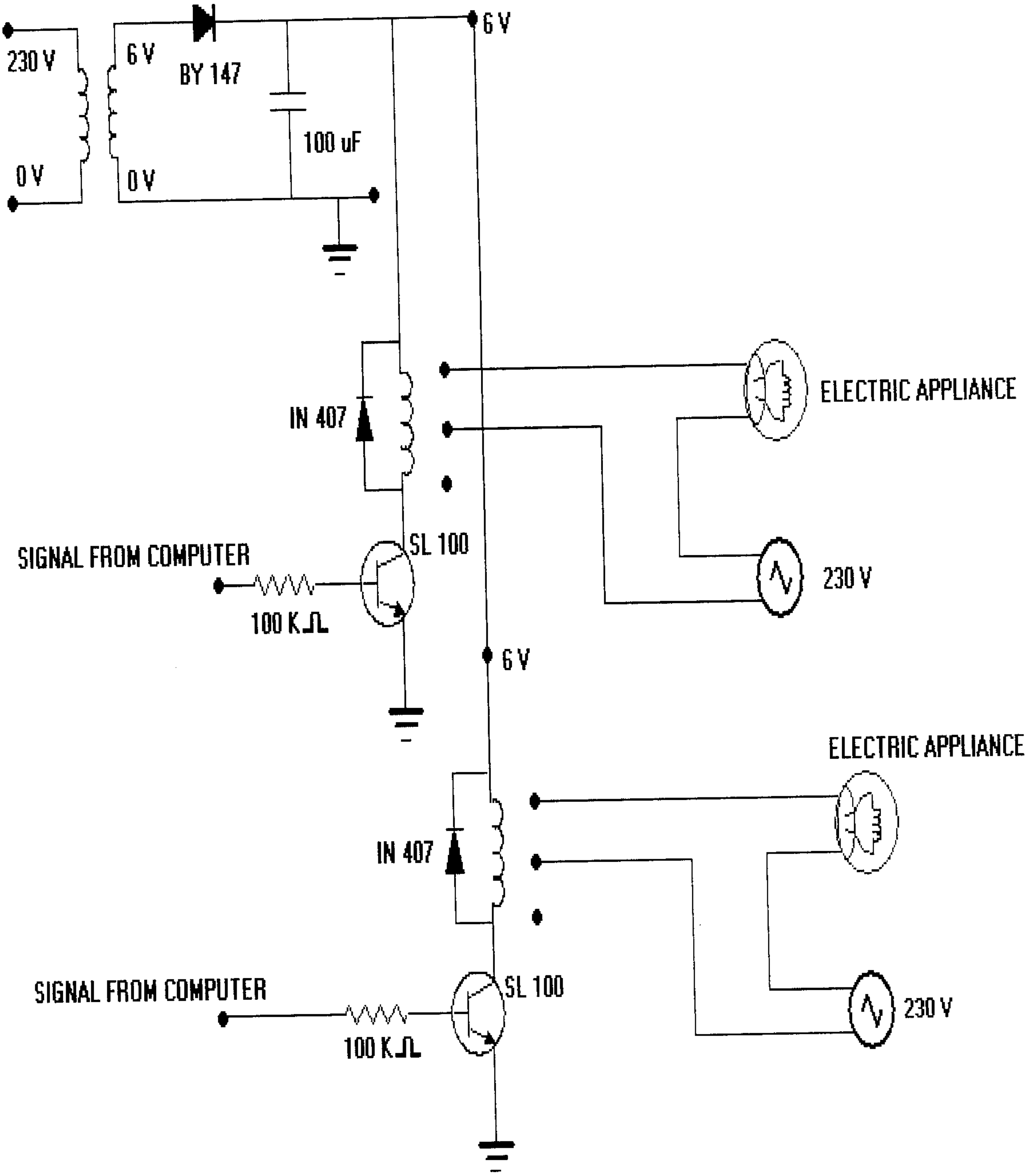


The pin diagram for the printer port showing data port, status port and control port.



The programming for sending data to the parallel port is done in C and is called from Visual basic.

Interfacing Circuit



The signal from the parallel port of the computer is sent to the interface circuit that is shown above. Value 1 is passed through the signal line to ON the appliance and value 0 to OFF the corresponding appliance. The transistor SL100 acts a switching device and whenever value 1 is passed, the transistor switches to ON and whenever the value passed is 0, the transistor switches to OFF state. When the transistor switches to ON state, the relay is switched and connects the 230 V power supply line with the appliance and makes it work. If value 0 is passed, the relay disconnects the circuit and brings the appliance to OFF state.

The transformer is used to supply a constant 6V D.C. to the relay for its operation. A 230-6V transformer is used which step downs the 230V to 6V for the working of the relay in the circuit. A BY147 half wave rectifier is used to convert the obtained signal to D.C. voltage. A 100uF capacitor acts as a filter which rectifies the presence of unnecessary A.C. voltage if any. Then the 6V D.C. signal is sent to the relay for its operation.

The Electric appliance is connected with a 230V power supply at one end and the other end with the relay switch. Switching can be performed according to the signal sent from the computer.

6. PRODUCT TESTING

6.1 UNIT TESTING:

Test 1: Communication between MUTS Home & MUTS Remote Software

Procedure: The test is performed to check whether MSComm control is able to communicate in correct sequence. Sample values are passed from one system to another.

Expected results: The message sent from one system should be ale to reach the other.

Results obtained: The MUTS home and MUTS Remote system could receive and send the messages in sequence and proved correct

Conclusion: The communication module between MUTS home and MUTS remote are correct.

Test 2: Hardware circuit

Procedure: The test is performed on the hardware circuit for its correct functionality. Trigger input are given as a simulation of the parallel port. The hardware circuit should be capable of switching the external appliances on and off. Two appliances such as electrical bulb are connected.

Expected results: when the first pin input is triggered the first light should be switched on or off accordingly.

Results obtained: In accordance to the input trigger the electrical bulb is switching on or off.

Conclusion: The hardware circuit responds to the trigger correctly.

Test 3: Answering calls other than owner

Procedure: A call is to be made through any phone. The MUTS Home should be capable of identifying that the call is other than the owner and should store the caller information and disconnect the call

Expected results: The call should be disconnected after storing the caller information.

Results obtained: MUTS Home identifies the unauthorized user and store the information and disconnect the call abruptly.

Conclusion: MUTS Home works properly for unauthorized users

Test 4: Answering calls to owner

Procedure: A call is made from MUTS Remote and it should acknowledges valid

Expected results: The call should be the identified valid.

Results obtained: The call is identified by reading the authorization code and acknowledged by giving the other options for home automation & caller info.

Conclusion: Home MUTS works properly in answering calls

6.3 INTEGRATION TESTING:

Test on: Overall integration of Home MUTS and Remote MUTS

Procedure: Both the MUTS Home and MUTS Remote are loaded in two systems connected to the phone lines. The software is checked for correctness of all the modules.

Expected results: The software should work properly with the correctness without any module conflict.

Results obtained: The software is able to communicate and do the required utilities namely caller info and home automation.

Conclusion: The software works correctly without any module conflict and is perfect.

6.4 ACCEPTANCE TESTING:

The acceptance test is done as a demonstration by our MUTS development team to the Project coordinator and guide. The software is accepted conforming to all the requirements specified in the Software Requirement Specification.

7. FUTURE ENHANCEMENTS

MUTS, although have essential features incorporated within it, the project still have certain featured to be added to make it more usable. Some of the future enhancements to be made are:

1. On Line Discussion forum
2. Receive and transfer voice messages.

The first feature planned to enhance was making our system for online discussion. Here the system will be connected to multi users and can be made to have a discussion forum between peoples.

The next enhancement would be receiving voice message from the caller, store it. Whenever the user calls the home system for any messages, the system must be capable of playing the voice to the remote thereby notifying the caller's identification. The system should also read out the caller ID information to the user at the remote.

The Home MUTS software programming can be embedded in chip as Assembly Language Program and the same functionality can be achieved without the need of having any computer at the home side.

8. CONCLUSION:

The MUTS is completed with full satisfaction and in scheduled period. Having all the functionality to support a phone user from tracing his phone calls and automating his home from a remote place, MUTS leaves his user carefree of house and the phone. With a simple call made to his house the user of MUTS is able to Control his electrical appliances and view the missed call's complete information. MUTS when given future enhancements will emerge as unbeatable telephone communication software with unmatchable usage.

Reference:

Peter H. Anderson, Department of Electrical Engineering, Morgan State University. Paper titled . *Use of a PC Printer Port for Control and Data Acquisition* Src:http://et.nmsu.edu

Dan Petit, *Real Visual Basic*, Addison-Wesley, 2000.

James F. Peters and Witold Pedrycz, 2000, *Software Engineering-An Engineering approach*, John Wiley & Sons, Inc.

N.N.Biswas, *Basic Telephony*, 1986.2e

Steven Holzner,*THE BLACK BOOK Visual Basic 6*, Dreamtech press, 2000.

Thiagarajan Viswanathan, *Telecommunication Switching Systems and Networks*, Prentice Hall of India. 1998

Microsoft Developer Network January 2000 edition

www.electronics.com

www.modems.com

www.modems.org

www.motorola.com

SOURCE CODE

Home MUTS source code

1.Main form

' This module helps the user to move to different modules

```
Private Sub MDIForm_Load()
Toolbar1.Buttons(1).Enabled = False
End Sub

Private Sub MDIForm_Terminate()
End
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1:
 process.Show
Case 2:
 process.Show
 process.Visible = False
 dialer.Show
Case 3:
 setting.Show
Case 6:
   End
End Select
End Sub
```

2. SETTINGS FORM:

' This module gets the hardware details about modem and communication line

```
Public port As Integer
Public mode, spk, ccid As String

Private Sub c1_Click(Index As Integer)
port = 1
End Sub

Private Sub C2_Click(Index As Integer)
port = 2
End Sub

Private Sub C3_Click(Index As Integer)
port = 3
End Sub
```

```
Private Sub C4_Click(Index As Integer)
port = 4
End Sub

Private Sub C5_Click(Index As Integer)
port = 5
End Sub

Private Sub C6_Click(Index As Integer)
port = 6
End Sub

Private Sub C7_Click(Index As Integer)
port = 7
End Sub

Private Sub C8_Click(Index As Integer)
port = 8
End Sub

Private Sub Check1_Click()
Option3.Enabled = True
Option4.Enabled = True
End Sub

Private Sub cid_Click()
ccid = "at#cid"
End Sub

Private Sub Command1_Click()
setting.Visible = False
MDIForm1.Toolbar1.Buttons(1).Enabled = True
MDIForm1.Toolbar1.Buttons(2).Enabled = True

End Sub

Private Sub Option1_Click()

End Sub

Private Sub off_Click()
spk = "atm0"
End Sub

Private Sub on_Click(Index As Integer)
spk = "atm2"
End Sub
```

```
Private Sub pulse_Click(Index As Integer)
mode = "p"

End Sub

Private Sub tone_Click()
mode = "t"
End Sub

Private Sub vcid_Click()
ccid = "at+vcid"
End Sub
```

3. DIALER :

```
' This module helps the user to call remote phone numbers

Dim dnumber As String, temp As String, inpstr As String
Dim eventdone As String
Dim b As Integer
Dim X As Long
Dim valuser As Boolean
Dim tosend As Integer
Dim firstrec As Boolean
Dim longstr As String
Private Sub back_Click()
If (Len(temp) = 0) Then
 MsgBox ("No Number to go back")
Else
temp = Mid(temp, 1, Len(temp) - 1)
numberBox.Text = temp
End If
End Sub
Private Sub Command1_Click(Index As Integer)
temp = numberBox.Text
dnumber = temp + Trim(Str(Index))
numberBox.Text = dnumber
temp = dnumber
End Sub
Private Sub Command4_Click()
temp = numberBox.Text
dnumber = temp + Trim("*")
numberBox.Text = dnumber
temp = dnumber
End Sub
Private Sub Command5_Click()
temp = numberBox.Text
dnumber = temp + Trim("#")
numberBox.Text = dnumber
```

```
temp = dnumber
End Sub

Private Sub dialBut_Click()
temp = numberBox.Text
dnumber = "atd" + setting.mode + numberBox.Text
process.phone.Output = dnumber + vbCr
End Sub
Private Sub Form_Load()
Timer1.Enabled = False
headTopic.Caption = "MULTI UTILITY TELEPHONE SYSTEM"
process.phone.Output = "at" + vbCr
process.phone.Output = spk + vbCr
process.phone.Output = ccid + vbCr

End Sub
Private Sub Form_Unload(Cancel As Integer)
Unload Me
End Sub
Private Sub hangUp_Click()
process.phone.Output = "ath0" + vbCr
End Sub

Private Sub redial_Click()
numberBox.Text = temp
process.phone.Output = "atdp" + temp + vbCr
End Sub
Private Sub reset_Click()
numberBox.Text = ""
dnumber = ""
process.phone.Output = "atz0" + vbCr
End Sub
```

4.PROCESSING:

```
'This module looks for any communication with the home system

Dim dnumber As String, temp As String, inpstr As String
Dim eventdone As String
Dim b As Integer
Dim X As Long
Dim valuser As Boolean
Dim tosend As Integer
Dim firstrec As Boolean
Dim longstr As String
Private Sub Form_Load()
Timer1.Enabled = False
On Error GoTo errlbl
```

```
phone.CommPort = setting.port
phone.PortOpen = True
phone.Output = "at" + vbCr
phone.Output = spk + vbCr
phone.Output = ccid + vbCr
valuser = False
firstrec = True
tosend = 1
GoTo end_fn
errlbl:
 MsgBox Err.Description + "  Check Settings"
 setting.Visible = True
Exit Sub
end_fn:
End Sub

Private Sub phone_OnComm()
If phone.CommEvent Then
   inpstr = phone.Input
End If
If inpstr <> "" Then
Call check
End If
End Sub
Private Sub check()
 Dim ln As Integer
 If InStrRev(inpstr, "RING") Then
   'Timer1.Enabled = True
   phone.Output = "ATA" + vbCr
   eventdone = "connected"

 End If
 If eventdone = "wait_auth" Then
   a = Trim(inpstr)
   If a = Trim(Str(setting.password.Text)) + vbCr Then
     phone.Output = "valid" + vbCr
     eventdone = "valid"
     valuser = True
   End If
 End If

 If InStrRev(inpstr, "CONNECT") <> 0 Then
   phone.Output = "send" + vbCr
   eventdone = "wait_auth"

 End If

 If InStrRev(inpstr, "NO CARRIER") <> 0 Then
 Randomize
 X = (Rnd) * 1000000

 Data1.Recordset.MoveLast

 Data1.Recordset.AddNew
 Data1.Recordset.Fields(0) = Trim(Str(X))
 Data1.Recordset.Update

 Data1.Recordset.AddNew
 Data1.Recordset.Fields(0) = Format(Now, "MM-DD-YY")
 Data1.Recordset.Update

 Data1.Recordset.AddNew
 Data1.Recordset.Fields(0) = Format(Now, "hh:mm:ss")
 Data1.Recordset.Update
End If

If InStrRev(inpstr, "CTRL") Then
  eventdone = "control"
  GoTo end_fn
End If

If InStrRev(inpstr, "q") Then
     lite = Split(inpstr, "q")
     ln = Val(lite(1))

  Select Case ln
  Case 0
    a = Shell("c:\muts\c 0", vbMaximizedFocus)
  Case 1
    a = Shell("c:\muts\c 1", vbMaximizedFocus)
  End Select
End If

 If InStrRev(inpstr, "ECTRL") Then
   eventdone = "excontrol"
 End If

 If InStrRev(inpstr, "LOG") Then
   eventdone = "log"
   Data1.Recordset.MoveFirst
   Do
      longstr = longstr + Trim(Data1.Recordset.Fields(0)) + "#"
      Data1.Recordset.MoveNext
      i = 10
      While i > 1
        i = i - 1
    Wend

    longstr = longstr + Trim(Data1.Recordset.Fields(0)) + "#"
    Data1.Recordset.MoveNext
    i = 10
    While i > 1
      i = i - 1
    Wend

    longstr = longstr + Trim(Data1.Recordset.Fields(0)) + "#"
     Data1.Recordset.MoveNext
     i = 10
     While i > 1
       i = i - 1
     Wend

  Loop While (Not Data1.Recordset.EOF)
  longstr = longstr + "*"
  phone.Output = longstr + vbCr
        eventdone = "endlog"
      GoTo end_fn
       End If

end_fn:
End Sub

Private Sub Timer1_Timer()
If Not valuser Then

    Timer1.Enabled = False
    Randomize
    X = (Rnd) * 1000000

    Data1.Recordset.MoveLast
    Data1.Recordset.AddNew
    Data1.Recordset.Fields(0) = Trim(Str(X))
    Data1.Recordset.Update
    Data1.Recordset.AddNew
    Data1.Recordset.Fields(0) = Format(Now, "MM-DD-YY")
    Data1.Recordset.Update
    Data1.Recordset.AddNew
    Data1.Recordset.Fields(0) = Format(Now, "hh:mm:ss")

    Data1.Recordset.Update
    phone.Output = "ath0" + vbCr
  End If
  End Sub
```

Remote MUTS source code:

1.Main form

```
' This module helps the user to move to different modules

Public flgfirst As Boolean

Private Sub MDIForm_Load()
flgfirst = True
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)
Select Case Button.Index
 Case 1
   dialer.Show
 Case 2
   SETTING.Show
 Case 3
   automate.Show
 Case 4
   callerid.Show
 Case 6
   End
End Select
End Sub
```

2. SETTINGS FORM:

```
' This module gets the hardware details about modem and communication line

Public portnum  As Integer
Public mode, spk, cid As String
Private Sub Check1_Click()
Option9.Enabled = True
Option10.Enabled = True
End Sub

Private Sub Command1_Click()
Me.Visible = False
MDIForm1.Toolbar1.Buttons(1).Enabled = True
End Sub

Private Sub Option1_Click()
portnum = 1
End Sub

Private Sub Option10_Click()
cid = "at#cid"
```

```
End Sub

Private Sub Option11_Click()
spk = "atm2"
End Sub

Private Sub Option12_Click()
spk = "atm0"
End Sub

Private Sub Option2_Click()
portnum = 2
End Sub

Private Sub Option3_Click()
portnum = 3
End Sub
Private Sub Option4_Click()
portnum = 4
End Sub
Private Sub Option5_Click()
portnum = 5
End Sub
Private Sub Option6_Click()
portnum = 6
End Sub
Private Sub Option7_Click()
portnum = 7
End Sub
Private Sub Option8_Click()
portnum = 8
End Sub

Private Sub Option9_Click()
cid = "at+vcid"
End Sub

Private Sub pulse_Click()
mode = "p"
End Sub

Private Sub tone_Click()
mode = "t"
End Sub
End If
GoTo endfn
lbl:
MsgBox (Err.Description + vbCr + "Check the Settings")
MDIForm1.Toolbar1.Buttons(1).Enabled = False
Exit Sub
endfn:
End Sub

Private Sub hash_Click()
temp = numberBox.Text
numberBox = temp + "#"
End Sub
Private Sub numberBox_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
  dial.Value = True
End If
End Sub

Private Sub phone_OnComm()
   inpstr = phone.Input
   Text1.Text = Text1.Text + Trim(inpstr)
     Call check
End Sub

Private Sub redial_Click()
numberBox.Text = temp
phone.Output = "atd" + pt + numberBox + vbCr
ProgressBar1.Value = 0
Timer1.Enabled = True
End Sub

Private Sub star_Click()
temp = numberBox.Text
numberBox = temp + "*"
End Sub

Private Sub Timer1_Timer()
ProgressBar1.Value = ProgressBar1.Value + 10
If ProgressBar1.Value >= 100 Then
Timer1.Enabled = False
End If
End Sub
Private Sub check()

If InStrRev(inpstr, "send") Then
   Login.Show
End If
```

```
If eventdone = "log" Then
 Text2.Text = Text2.Text + inpstr
End If

If eventdone = "sent" Then
   If InStrRev(Text1.Text, "valid") Then
      MsgBox "Valid Password"
      MDIForm1.Toolbar1.Buttons(3).Enabled = True
      MDIForm1.Toolbar1.Buttons(4).Enabled = True
      eventdone = "valid"
      Login.Visible = False
   Else
      MsgBox "Invalid Password"
      eventdone = "invalid"
   End If
End If
i = i + 1
If i = 5 Then i = 1
End Sub
```

4. LOGIN

```
   'Gets the Authentication code from the user

Option Explicit
Public LoginSucceeded As Boolean
Private Sub auto_Click()
automate.Show
End Sub

Private Sub cmdCancel_Click()
   'set the global var to false
   'to denote a failed login
   LoginSucceeded = False
   Me.Hide
End Sub

Private Sub cmdOK_Click()
   Timer1.Enabled = True
   dialer.phone.Output = txtPassword.Text + vbCr
   Login.Visible = False
   dialer.phone.Output = "" + vbCr
End Sub

Private Sub Form_Load()
   dialer.Visible = False
End Sub
```

```
Private Sub Timer1_Timer()
Timer1.Enabled = False
  dialer.eventdone = "sent"
   dialer.phone.Output = "" + vbCr
End Sub
```

5. CALLER ID:

```
   ' This  module  retrieves the caller ID number and time of call

Dim i, x As Integer
Dim info
Dim temp As String
Dim infoind As Integer

Private Sub Command1_Click()
temp = dialer.Text2.Text
   info = Split(temp, "#")
   While (infoind < UBound(info) + 1)
      Select Case x
      Case 1:
         List1.AddItem (info(infoind))
         x = 2
      Case 2:
         List2.AddItem (info(infoind))
         x = 3
      Case 3:
         List3.AddItem (info(infoind))
         x = 1
      End Select
      infoind = infoind + 1
    Wend
    eventdone = "endlog"
 End Sub
 Private Sub Form_Load()
 dialer.phone.Output = "LOG" + vbCr
 dialer.eventdone = "log"
 x = 1
 End Sub
```

6.Automation:

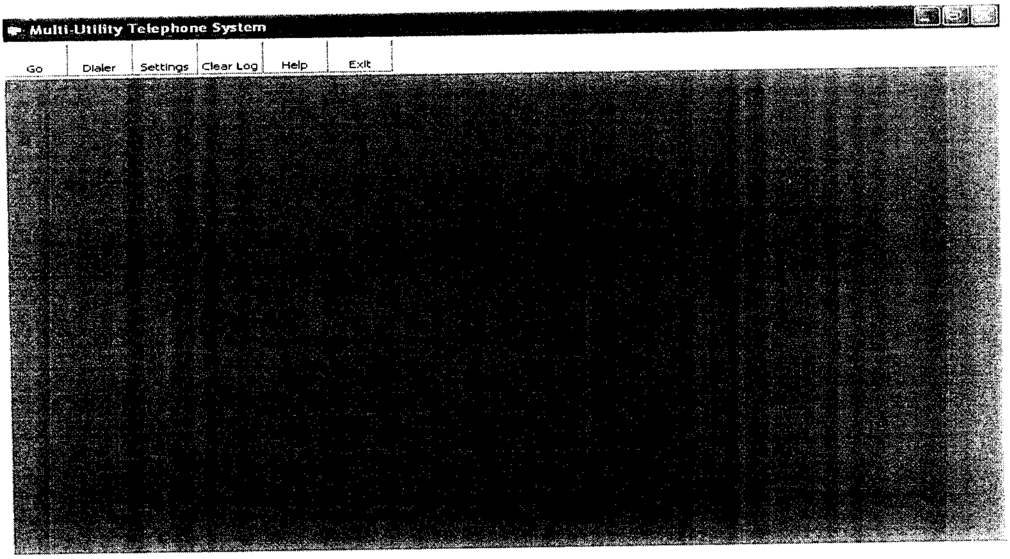
```
'This module performs the control of Electric Appliances connected with the
Home 'system

Dim light(8) As Integer
Dim x As Integer

Private Sub C1_Click()
```

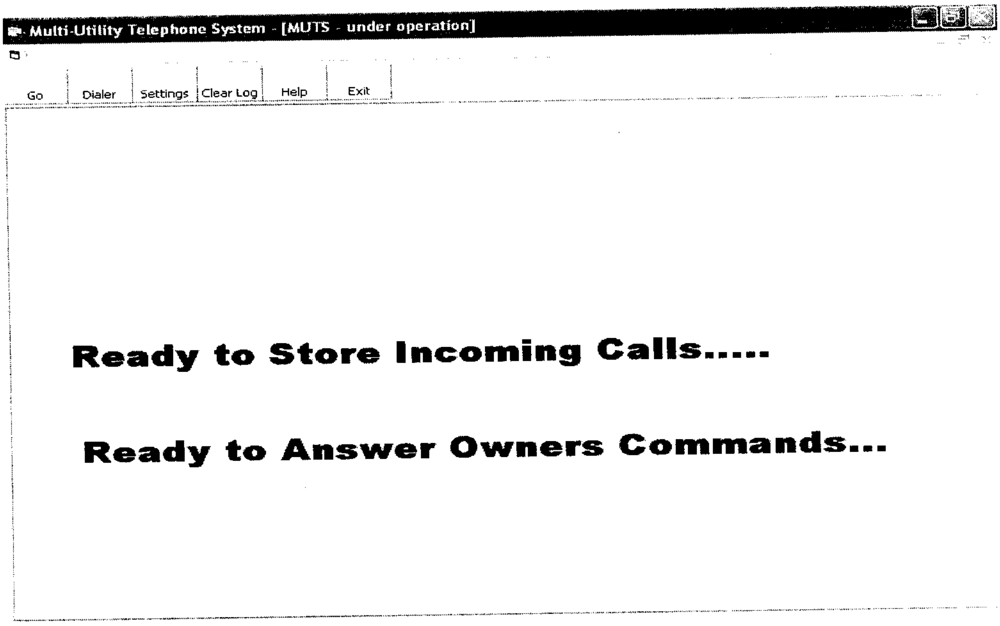
```
If Option1.Value = True Then
   light(0) = 1
Else
   light(0) = 0
End If
End Sub

Private Sub C2_Click()
If Option3.Value = True Then
   light(1) = 1
Else
   light(1) = 0
End Sub

Private Sub C3_Click()
If Option5.Value = True Then
   light(2) = 1
Else
   light(2) = 0
End If
End Sub

Private Sub C4_Click()
If Option7.Value = True Then
   light(2) = 1
Else
   light(2) = 0
End If
End Sub

Private Sub C5_Click()
If Option9.Value = True Then
   light(2) = 1
Else
   light(2) = 0
End If
End Sub

Private Sub C6_Click()
If Option11.Value = True Then
   light(2) = 1
Else
   light(2) = 0
End If

End Sub

Private Sub C7_Click()
```

```
If Option13.Value = True Then
   light(2) = 1
Else
   light(2) = 0
End If
End Sub

Private Sub C8_Click()
If Option15.Value = True Then
   light(2) = 1
Else
   light(2) = 0
End If

End Sub

Private Sub Command1_Click()

Unload Me

End Sub

Private Sub Command2_Click()
While (i <= 7)
   If light(i) = 1 Then
      x = x + (2 ^ i)
   End If
   i = i + 1
Wend
MsgBox x
dialer.phone.Output = "q" + Str(x) + vbCr
x = 0
End Sub

Private Sub Form_Load()
dialer.eventdone = "control"
End Sub

Private Sub Form_Unload(Cancel As Integer)
dialer.phone.Output = "CTRL" + vbCr
dialer.eventdone = "endcontrol"
End Sub
```

USER INTERFACE

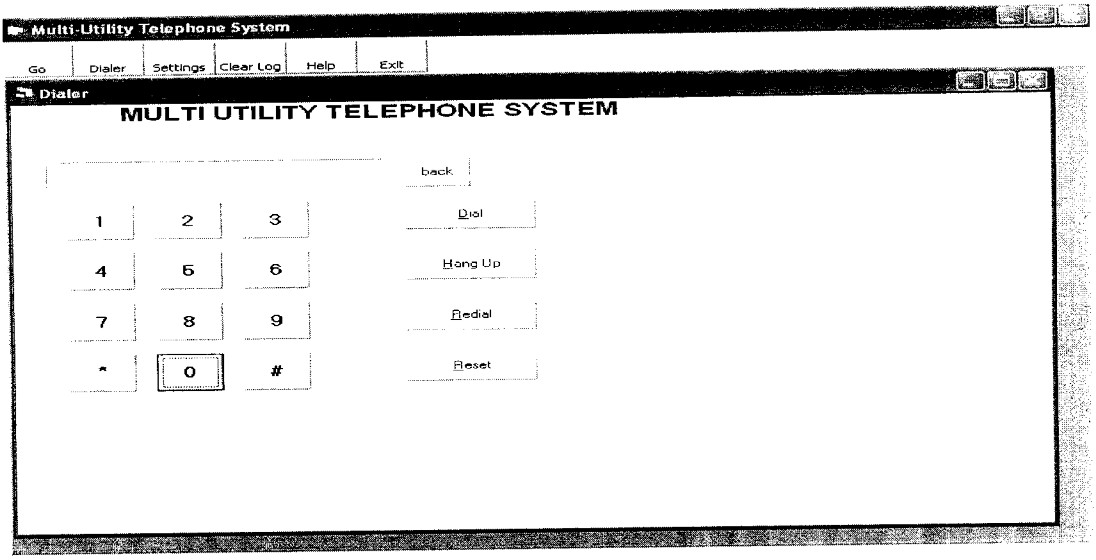
HOME MUTS INTERFACE :

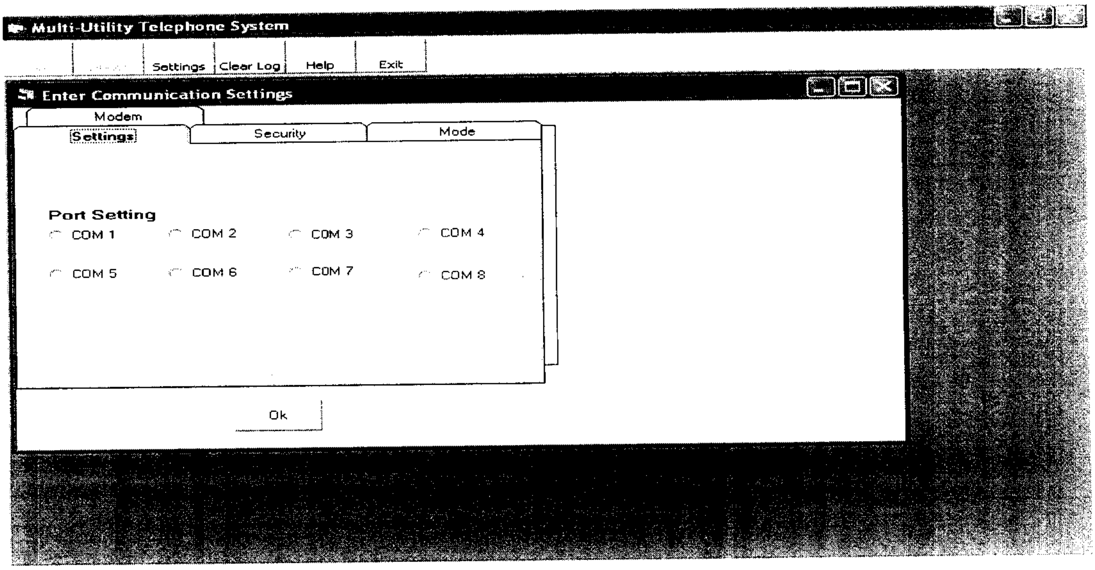




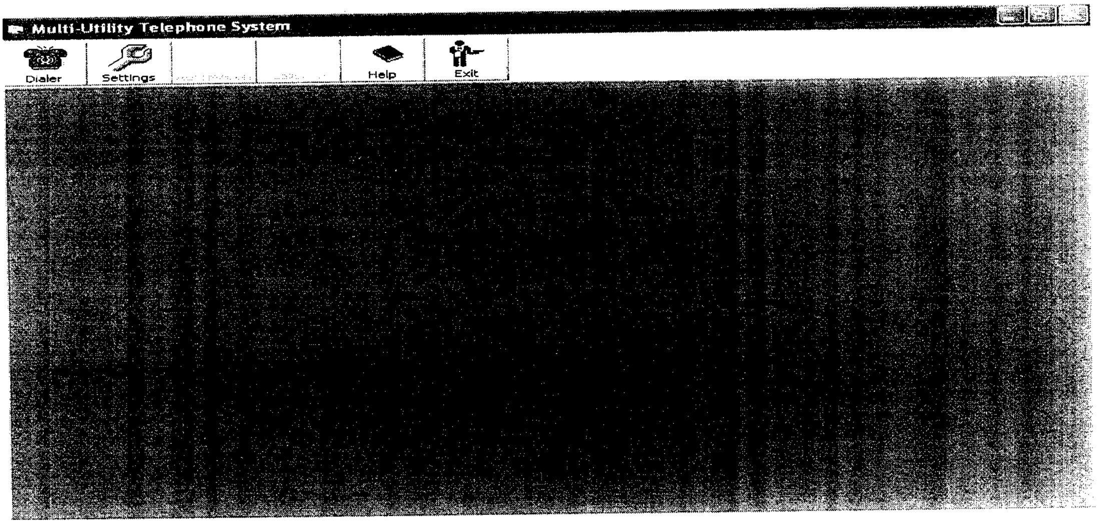
Ready to Store Incoming Calls.....

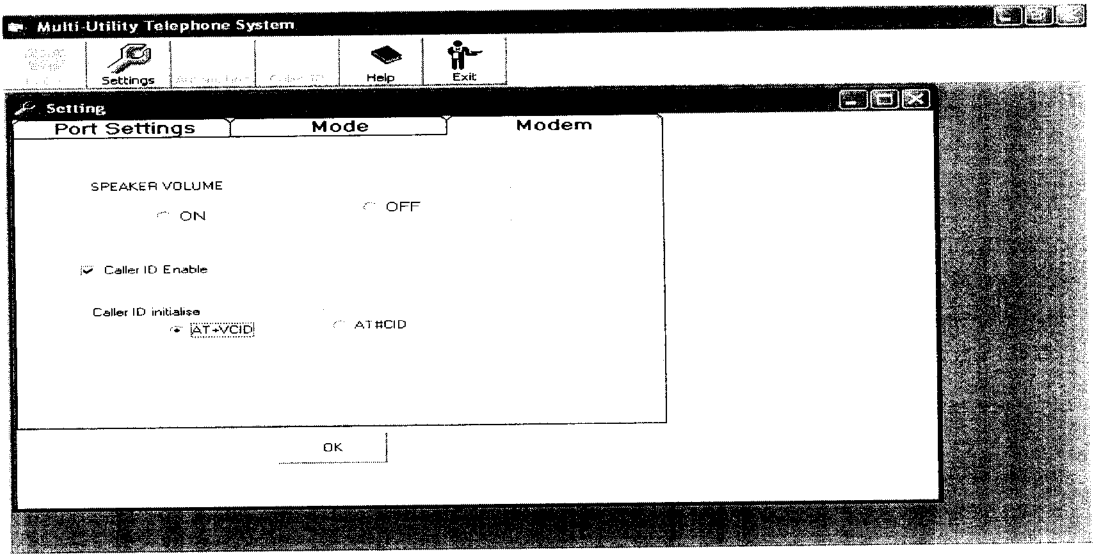
Ready to Answer Owners Commands...


Multi-Utility Telephone System
Go
Dialer
Settings
Clear Log
Help
Exit
Dialer
MULTI UTILITY TELEPHONE SYSTEM
back
1
2
3
4
5
6
7
8
9
*
0
#
Dial
Hang Up
Redial
Reset


Multi-Utility Telephone System
Settings
Clear Log
Help
Exit
Enter Communication Settings
Modem
Settings
Security
Mode
Port Setting
○ COM 1
○ COM 2
○ COM 3
○ COM 4
○ COM 5
○ COM 6
○ COM 7
○ COM 8
Ok

REMOTE MUTS USER INTERFACE





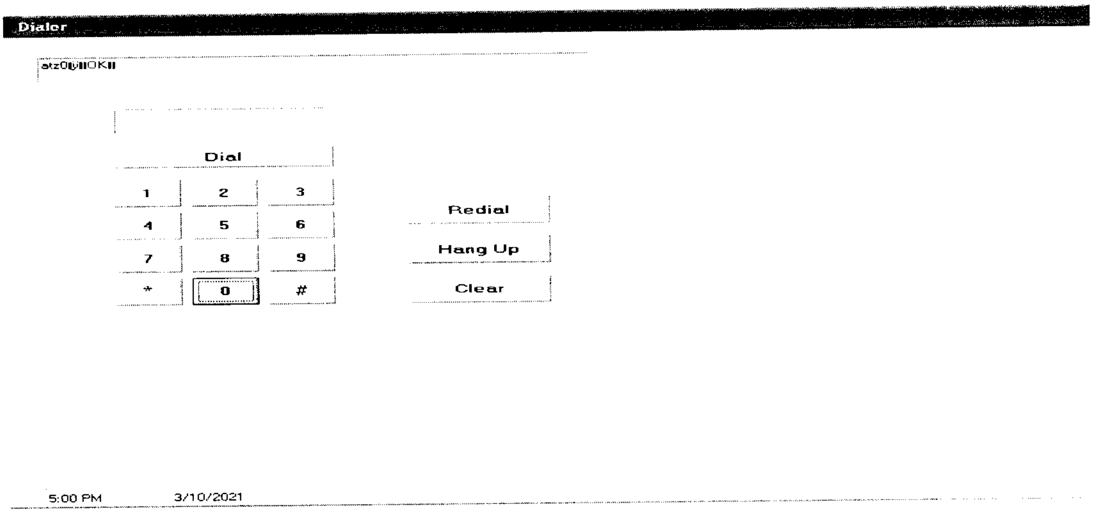
SPEAKER VOLUME

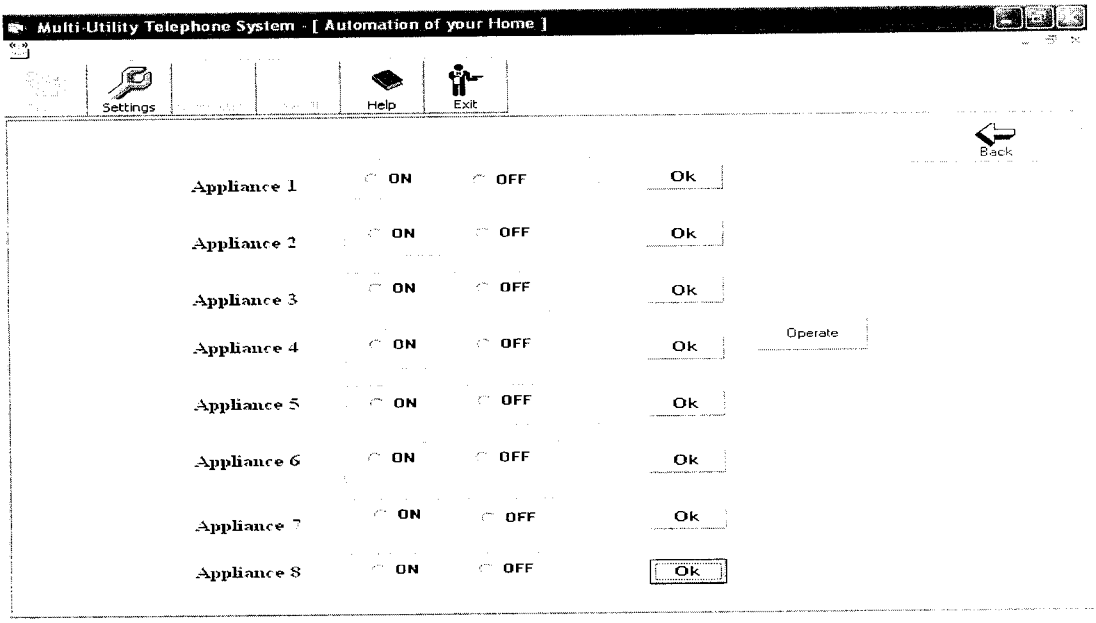
○ ON ○ OFF

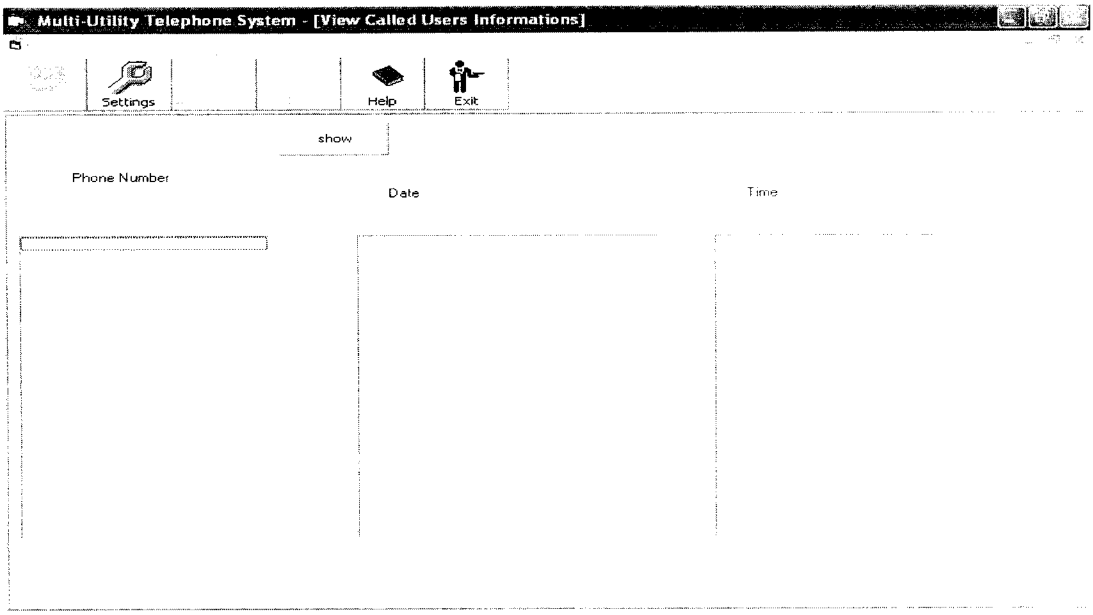
☑ Caller ID Enable

Caller ID initialise

● AT+VCID ○ AT#CID


Dialer
atz0▯▯▯OK▯▯
Dial
1
2
3
4
5
6
7
8
9
*
0
#
Redial
Hang Up
Clear
5:00 PM
3/10/2021


Multi-Utility Telephone System - [Automation of your Home]
Settings
Help
Exit
Back
Appliance 1 ○ ON ○ OFF Ok
Appliance 2 ○ ON ○ OFF Ok
Appliance 3 ○ ON ○ OFF Ok
Appliance 4 ○ ON ○ OFF Ok
Appliance 5 ○ ON ○ OFF Ok
Appliance 6 ○ ON ○ OFF Ok
Appliance 7 ○ ON ○ OFF Ok
Appliance 8 ○ ON ○ OFF Ok
Operate



Multi-Utility Telephone System - [View Called Users Informations]