AMBIENT INTELLIGENCE WITHIN AN ENVIRONMENT

PROJECT REPORT

MARCH-2003

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING

Submitted by

Bhargavy Gananathan

Nithya.V

Palliyil Veena Ravindranath

Under the guidance of

Mr.K.R.Baskaran, B.E.,M.S.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641006

KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore-641006

CERTIFICATE

This is to certify that project report entitled 'AMBIENT INTELLIGENCE WITHIN AN ENVIRONMENT' is a bonafide record of work done by:

Bhargavy Gananathan	- 9927S0048
Nithya.V	- 9927S0062
Palliyil Veena Ravindranath	- 9927S0063

in partial fulfillment of the requirements for the award of Bachelor of Engineering.

Head of the Department

Staff in charge

Submitted for the University Examination held on 20-3-2K3

Internal Examiner

External Examiner

PLACE: COIMBATORE
DATE: 20-3-2K3

ACKNOWLEDGEMENT

The success accomplished in this project would have not been possible without the timely help and guidance rendered by many people to whom we feel obliged and grateful. We wish to express our sincere and heartfelt gratitude to these personalities.

Dr. K.K. Padmanabhan B.Sc., (Engg) M.Tech., Ph.D., Principal, Kumaraguru College of Technology, Coimbatore for permitting us to undertake the project.

We also thank Dr. S.Thangasamy, Ph.D., Head of Department of computer Science and Engineering, Kumaraguru College of Technology and Mrs. S.Devaki, M.S., Assistant Professor for providing us guidance and making this project a successful one.

We wish to express our sincere gratitude to our project guide Mr.K.R.Baskaran,B.E.,M.S., for the interest and the support he gave us towards the completion of this project successfully.

Our deepest gratitude goes to Mrs.N.Chitra Devi M.E., Lecturer and all other staff of our department for their unrestrained attention at all stages to make this project a successful one. The motivation provided by all the staff remained a milestone for the completion of this project.

We express our sincere gratitude to Mr.Jaykumar, Mr.Balaji and other members of Cyber Soft India, Coimbatore, for their valuable support and guidance throughout the course of this project.

Our parents and all our friends deserve particular thanks for helping us out in various ways. Without their valuable assistance, this project would have fallen short of schedule and requirements.

There are so many people, so many minds all of whom have extended their hands to us in some way or the other. Neither words can express nor pages can fill our sincere gratitude.

CONTENTS

1. INTRODUCTION

Current status of the problem taken up

Today's system is one in which man must adapt to the appliances. Hence, work becomes tedious, time consuming and of course, less interactive.

Tomorrow's system, that is, the system that we have developed is one, which adapts to man's needs.

In the existing system, remote controls, smart cards etc are used. The 'Blue tooth' technology has been adapted. But, all these features have their own disadvantages. Taking the case of remote controls, it can be operated at a limited distance only. But, we have aimed at enabling the possibility of controlling the appliances from anywhere in this world. If we consider, the smart cards, the cost factor is a major drawback. Moreover, it has to be treasured safely.

Coming to the device controller, the present systems use the X-10 signal receptor which pose a major set back in the economy factor for most of our people. We have designed a circuit, which can be used as an alternative with a nominal rate.

Our system serves as a platform for remote control of various appliances by connecting arbitrary appliances through various arbitrary connections. Hope! It would act as a user- friendly system.

Relevance and importance

After getting the results of initial study, the investigation is expanded to more detailed feasibility study. The objectives at this stage are to provide solutions to the stated problems usually in the form of specifications to meet the user requirements and make recommendations for a new computer based system.

The system has been evaluated from a technical point of view first and if technically feasible, the system is tested for social and economic compatibility.

Technical Feasibility

The system is developed using Visual Basic 6.0, Web tools: HTML and ASP

Social Feasibility

The assessment of social feasibility is done along technical feasibility. Our project is developed in such a way that a user can use it very flexibly.

Economic Feasibility

Our system is justified by cost and benefit criteria to ensure that efforts is concentrated on projects, which will give test return at the earliest opportunity.

Interactive System

Perfect analysis is made by finding out what happens in the existing system, deciding what changes and new features are to be added and then defining exactly what the new system must do. The system analysis is largely concerned with determining, developing and agreeing to user's requirements and selecting those to be satisfied by an interactive solution. Prime opportunity to communicate with the users and conceive a join understanding of what the system should be doing, together with a view of the relative importance of the system features has been provided.

Advantages

The system is user friendly with improved efficiency. The advantages are listed below:

- Adaptability

  The system adapts to the user requirements.

- Mobility

  The system can be used in any environment.

- Modularity

  Our system is developed using a hierarchical approach by dividing into smaller modules.

2. LITERATURE SURVEY

A similar project was carried out using Perl and CGI programming. Though, it had it's own advantages, there are so many nifty shortcuts built into the Perl that we generally can accomplish a lot more in a lot less code in Visual Basic.

Perl's string-manipulation functions are complex that it's almost ridiculous to try. It is not an exaggeration to say that 90% of CGI programs involve heavy string manipulation.

On the contrary Visual Basic which is familiar to just about anyone who has ever done programming. It is easy to learn easy to use, it is fast and is popular. Above all, in a totally Microsoft environment, it can work quickly and efficiently.

Setting aside the software part, if we look into the hardware. I.e., the device controller, it was developed using x-10 signal receptor. This is costly equipment. Now, we have devised this controller keeping the cost factor in mind.

We obtained a good experience during the course of the project. It has strengthened our team spirit and has provided us with practical knowledge of how a hardware system is designed. In short it was a novel experience for us all.

3. SOFTWARE REQUIREMENTS

We have mainly used Visual Basic language in the Windows 98 operating system. The web tools used are HTML and ASP.

Introduction to Visual Basic

Visual Basic (6.0) is the newest addition to the family of Visual Basic products. Microsoft Visual Basic is the quickest and easiest way to create powerful application for Microsoft windows operating system. The VB programming system allows creating robust and useful applications that fully make use of the graphical User Interface (GUI). VB provides graphical environment in which we can visually design the forms and controls that become the buildings block of the application. VB supports many useful tools that will help us to be more productive.

Version 6.0 of VB is specifically designed to utilize the Internet. It comes with several controls and allows creating web-based applications called Active X executables.

New to VB 6.0 are the ISAPI application and dynamic HTML project templates. These templates provide a framework to develop server-side components as well as "smart web pages and applications".

Visual Basic continues to support the explorer style development environment modeled after windows explorer. This makes it easy for a computer user to jump right into creating applications with VB. Using VB powerful full-featured applications can be created.

Data access features allows to create database and front end applications for most popular database formats.

Windows 98

Windows 98 offers many new and exciting features in addition to any features from earlier versions of windows. Most important feature is it's flat 32 bit addressing. Windows 98 is a pre-emptive, multitasking to perform several actions at the same time. The advantages are powerful, mobile computing features using the Microsoft Network, a recycle bin for storing deleted files, and the inbox, which can be used for communication like fax, e-mail.

The other features are:

New improved interface

Windows 98 features the task button and task bar. Start button can be used to quickly open programs, find documents and use system tools. Task bar is used to switch between programs.

Windows Explorer

Windows explorer is a powerful way to browse through and manage our files, drives, and windows explorer is a powerful through and manage our files, drives and networks.

Improved multimedia support

We can enjoy the faster void capability for games, enhanced support for MS-DOS based games and improved performance for playing video based sound files.

Microsoft Exchange

We can use Microsoft exchange to view and work with all types of electronic communication including e-mail and faxes.

Microsoft Network

We can use this new, affordable, and easy to use online service to communicate with the people worldwide, using e-mail, bulletin boards, and the Internet.

Scan disk

The scan disk is used to check the hard disk for errors and fix the problems that are around.

Drive space

This is to compress the disk.

ASP

Active Server Pages (ASP) technology is actually a very simple idea. ASP let us to use the power of a web server to process user requests and provide dynamic, individualized content based or logic, file and database data, and also process the users individualized data. In other words, ASP lets multiple users simultaneously run a program on your web server.

ASP treats users as unique entities, even though all of them are running the same program on the same machine – our web server. Obviously, running a complex program for multiple users on one computer takes a lot of resources. Fortunately, Microsoft makes sure those resources are at our disposal .ASP provides:

- A way to save individualized data for each user
- Access to the file system
- Access to databases
- A means to launch and control any component object model (COM) component.

In short, ASP allows you to build multi user applications, which means we can leverage ASP to provide application scalability. If we plan our ASP applications carefully, they can scale from a single-sever application that can handle 30-100 simultaneous users and a few thousand hits per day to multi-server (web form) applications handling thousands of simultaneous users and millions of requests per day.

ASP isn't a single technology. Using ASP effectively requires that we learn a collection of tools, languages, techniques, and technologies.

Benefits of ASP

- ASP is language independent.
- ASP is for non-programmers.
- ASP code resides in text files.
- ASP code times out.
- ASP code is server-safe.
- ASP code doesn't require registration.
- ASP applications are usually small.
- We can upgrade ASP applications without stopping IIS.

HTML

In keeping with the original and ongoing theme of the Internet – openness and portability; HTML was developed. We can create, edit, or view the HTML code for a web page in any text-editor on any computer platform, such as windows notepad.

Although creating simple web pages in a text editor in easy, it can quickly turn into a grueling and mind-numbing task. That's why there are web-authoring tools such as Microsoft Front Page express and Macromedia Dream weaver, to create web pages in the same way as we create documents in word processor.

When we open a web page in our browser, we don't see the HTML code, instead, the browser interprets the HTML code and displays the page appropriately on the screen. If we are creating a web page in a text editor and want to view the file, then we have to save it and open in our browser. Then, we can continue to edit, save ones work, and view the results, switching back and forth between the text editor and the browser to see the effects.

The original intent of the HTML specification was to allow web authors to describe the structure of a page without spending too much time worrying about the look of the page-that part of the job was left to the browsers. Traditionally, each browser had it's own way of interpreting the look of the page, and web authors had to live with the fact that pages they created might appears somewhat differently in different browsers.

4. PROPOSED APPROACH TO THE PRODUCT

Software Configuration

LANGUAGE	:	VISUAL BASIC 6.0
OPERATING SYSTEM	:	WINDOWS 98
WEB TOOLS	:	HTML, ASP

Hardware Configuration (server)

- RD DISK: CPU:PIII 500 MHZ
- HA41 GB SCSI
- MAIN MEMORY : 256 MB
- DISPLAY : 15"
- KEYBOARD : 104 KEYS
- MOUSE : PS/2 SCROLL MOUSE
- DISKDRIVE: 1.44 MB
- CD DRIVE: 1.44 MB
- CD DRIVE : 52X
- MODEM : ISDN
- PRINTER

Hardware Configuration (client)

- CPU: PIII 500 MHZ
- HARD DISK : 41 GB SCSI
- MAIN MEMORY 256 MB
- DISPLAY : 15"
- KEY BOARD : 104 KEYS
- MOUSE : PS/2 SCROLL MOUSE
- DISKDRIVE: 1.44 MB
- CD DRIVE:52 X
- MODEM: ISDN

5. DETAILS OF THE DESIGN

The whole project is basically divided into three modules and finally integrated to bring about a full-fledged interactive system. The three modules are

- Remote Desktop Capturing
- Device Control and Programming
- Interaction over the internet

Remote Desktop Capturing

This is done within a LAN environment while one system acts as a client and the other acts as a server.

Client Side

A socket is first created and bound to the server port. After a smooth connection, user name and password is sent as data to the remote server. After successful login, sending a request data captures the remote desktop. The request is got after which the mouse ports and keyboard is connected. Then synchronization operations are done properly.

Server Side

A socket is created for any port and is made to listen during program initialization. The ports are watched continuously for any incoming data. The username and password are verified and an acknowledgement is sent, otherwise it is rejected. Then the result is got, processed and the results are returned. New ports are opened for the keyboard and mouse. Lastly, the win32 API functions are called to perform the operations.

Device control and Programming

A simple hardware interface circuit and it's driver software is developed to control the appliance with the help of relays. For checking the current status interactively, a software stimulator of the device controller is produced.

The hardware consists of ten easily available discrete ICs wired up for operating up to seven electrical devices. The software comprises mainly the graphics portion for a GUI (Graphics User Interface) based system. Control software has been written for one device only at present and can be expanded for control of more devices in the like manner. The circuit uses printer port 378 (hex) as output and 379h as input port. Thus, the device control and programming is done.

Interaction over the Internet

This module deals with how a user can get access to the server through the Internet and thus control the appliances connected to the device controller.

For this, an interactive web page is designed for the user, which is accessed by him after proper validity tests are performed.

Thus the user can view the status of each device as well as change the status depending on the requirements.

IMPLEMENTATION

DETAILS

6. IMPLEMENTATION DETAILS

Remote Desktop Capturing

This module deals with accessing the device controller within a LAN Network.

LAN Network is where all the computers located at a particular distance from one another communicates with each other. The idea proposed here is Remote Desktop Capture where by we capture the current desktop of the system to which the device controller is connected.

For the remote desktop capture two software namely the client side program & the server side program are written. The server program must always be in the running mode at the server. Any client on the network can access this server by means of obtaining the IP address of the server. Once the IP address of the server is known, we then connect to the remote server by means of a password and username. The password and username at the client side must be same as the one used by the user to login at the server.

All the ports at the server side are in the 'listen' mode. Once authentication is verified, then we can enter the remote server through these ports that are listening. The client program will then establish a connection with the server through these ports and hence transmission and reception of data begins.

We have implemented this module using the front-end tool Visual Basic 6.0. However, we have used a number of VC++ functions by importing the libraries to Visual Basic. The system libraries of VC++ make the Remote Desktop Capturing less complex.

Once the connection between the client & the server has been established, we proceed to the next step, where the mouse and the keyboard movements are also to be detected. Establishing connection through ports also does this. If the client gives the option to capture Remote Desktop, then the actual operations begin. The server sends the current size of it's desktop to the client. The client stores this size and creates a Bitmap for the new desktop that it has to redraw. Once the bitmap is populated with all the pixel details, the desktop along with the icons are slowly redrawn at the client. The next step is then to keep sending data to and from the client to redraw only that part of the desktop where changes have occurred. (E.g.: when we push 'start' button, popup menu is seen). Then all simple file operations like reading a file and writing into a file is done. An exact synchronization is to be established between the operations going on at the client side and the server side.

Once the user is done with all the operations, he can exit the software and get back the original desktop. The server will disconnect itself from the client and then continue it's 'listening' state so that it may be accessed again by some other user later. Thus the Remote Desktop Capture will help us to control and monitor the remote server, thus simplifying the operation of the device controller within a LAN.

Device control and Programming

Hardware Description

The printer port of the PC terminates into 25 in 'D' Connector. It consists of eight data output (pin Nos.2 through 9 for data bits D0 through D7 respectively) pertaining to port with address 378(hex). In case this port with base address 378h(referred to as LPT1 port) is not available on your PC, you may use alternate port with base address 278h or 3BCh after checking the availability program in DOS or any other appropriate program. Port 379h (or in lieu port 279h or 3 BD, in case you are using base address 278h or 3BCh is an input port. Only one bit i.e., D7(pin 11, used for busy signal during printer operation has been used for reading the status.

PC inverts this bit before reading it. Further, all pins of this input port are logic 1 when left unused. Thus, if pin 11 is made high, the port will read 127 (decimal) and when pin 11 is made low, the port will read 255(decimal). Pin Nos., 18 through 25 are ground pins. The usage of various pins/data-bits of output and input port in this project is summarized in Table 1.

The complete circuit diagram is shown in Fig.1. 1-of-8 decoder IC1 (74LS138) decodes the address of the selected device out via bits D0-D2 of port 378h. Only the output corresponding to selected device goes active low. Output O0 from IC1 is not used deliberately. The address of selected device is thus numerically equal to the weightage of the binary address bits itself.

For switching on/off of the addressed device, data bit D5 is always made logic 1. This causes mono flip-flop IC3 (74LS121) to produce a pulse of approximately 5 ms duration with timing component values as shown in Fig.1. The low-to-high going pulse from pin 6(Q0) of IC3, after passing through NOR buffer N5, is logically ORed with the decoded active-low outputs of decoder IC1 in seven NOR gates (N1 through N4 and N6 through N8) of IC9 and IC10 (74LS02).

The output of only one (out of seven) NOR gates, corresponding to the addressed device, outputs a low-to-high going pulse having the same pulse width as that of the mono flip-flop. This acts as a clock for corresponding D-latch formed out of four 74LS74 (dual D edge – triggered flip-flop) ICs (5 through 8). The 'D' inputs of all the latches, which are shorted together, are connected to D4 bit (representing On/off State of the addressed/selected device). I.e., pin 6 of 25 pins 'D' connector.

Since the clock is available to only one (addressed/selected) flip-flop the switch state (on/off) is reflected the output of only the concerned device flip-flop and this is used for switching on/off the relay assigned to the specific device via Darlington driver forming part of IC4 (ULN2003). Logic low at the output of ULN2003 indicates that the relay pertaining to selected device is on, and vice verse. The state (on/off) of the unselected device remains unaffected (i.e., latched to the previous state).

The outputs from ULN32003 (IC4) are also used for sampling the state of the devices by connecting them to the seven inputs (11 through 17) of 8-input multiplexer 74LS151(IC2). Input I0 is not used, as done earlier in case of O0 output of 74LS138(IC1). The negated output of IC2 (at pin 6) is connected to busy pin 11(bit D7 of port 379h) while address of the device whose status is to be checked is input to IC2 in parallel with IC1.

For expanding the system to switch on/off up to say, 15 devices, one may use additional address line (D3) after replacing IC1 with 75154 and IC2 with 741150 and adding additional latches and NOR gates. The high current/voltage Darlington transistor arrays (ULN2003) are capable of passing up to 500 mA of collector current. These can be driven with ITL logic inputs and the supply voltage can go up to 30V (max). As each output is provided with freewheeling diode internally, external diodes across relays are not required.

The circuit requires 5V and 12V supplies externally for operation of the interface circuit and the relays respectively. The supply may be tapped from a spare socket of SMPS of the PC itself.

Software Description

The software consists of graphics commands and is menu driver. The main interface commands used are:

- VbOut (post address data)
- VbInp (port address)

The VbOut command outputs from printer port and the VbInp command is used to get binary data byte from the specified Input port. Since pin11 is used for checking the status, the value of the status byte returned will be either decimal 127 or 255.

After checking the status value, the appropriate messages are displayed to the user. The status can also be changed from 'ON' to 'OFF' and vice versa, as the user requires.

The selection of the device is also done through the menus, after which the user decides for him, if he just wants to view the status or change the status by selecting the item from the drop-drown menu. After changing the status for a particular device, the confirmation of the changed status is also displayed to the user, interactively through the message boxes. Also, the current status will be stored in a file for future clarifications that can be accessed by the user in a LAN network or through the Internet.

After performing the desired operations, the program gets closed by choosing the close option from the menu. Thus the device programming is done for the Device Controller.

Interaction over the Internet

This module deals with getting user inputs, processing the inputs and displaying the output in client understandable format. For this purpose HTML is used.

ASP is used for service side processing.

User can gain access to the server, by providing a valid user name and password. The username and password verification is cone by a call to the appropriate ASP at the server.

Once the username and password is verified, the user can choose the room and the particular devices he wishes to control.

The user can view the present status of the device and also modify the status of the device.

Every time the user changes the status of the devices the change is written into a text file present at the server.

At any time the text file contains the present status of the device. So when a user requests to view the status of the device, the status is read from the file and displayed to the user.

Other options available to the user are

(i) Option to view the settings

(ii) Option to view the log file

Once the user logs off, the session is abandoned and any attempt to view the page by any other third person is in vain.

TABLE I

Port	Pin Nos	Data Bits	Usage
378h	2	D0-D2	For outputting the address of device
378h	6	D4	For outputting on/off (I/O) state
378h	6	D5	For turning monostable on/off(I/O)
379h	11	D7	For inputing status of addressed device
-	18-25	Gnd	To be merged with interface ground

PARTS LIST

Semiconductors	
IC1	-74LS138, 1-of-8 decoder/demultiplexer
IC2	-74LS151, 8-input multiplexer
IC3	-74LS121 monostable flip-flop
IC4	-ULN2003 high current/voltage Darlington transistor arrays
IC5-IC8	-74LS74 dual D-type flip-flop
IC9-IC10	-74LS02 quad 2-input NOR gate
Resistor(1/4 watt, ± 5% carbon)	
R1	-2.2Kilo-Ohm
Capacitor	
C1	-4.7 µF, 25V electrolyte
Miscellaneous	
RL1-RL7	-12V, 200-ohm relay, 25 pin D-type male and female connector 25 core -10 pin male female SIP connectors

7. TESTING

The philosophy behind testing is to find errors. The common view of testing is that it is performed to prove that there are no errors in a program. However this is virtually impossible since no program will be free and clear of errors. Therefore the useful and practical approach for tracing the error should be employed. Testing is the process of executing the program with the explicit intention of finding errors that make the program fail. Hence a successful test is one that finds an error.

During testing, the program to be tested is executed with a set of test cases and the output of the program is evaluated to determine if the program is performing the way as it is expected to be. Actually testing is the state of implementation, which is aimed at ensuring that the system works accurately and efficiently before liver operation commences.

Software testing is a crucial element of software quality assurance and represents the ultimate review of specification, design and coding. A series of testing are performed for the proposed system before the system is ready for acceptance.

Unit testing

Unit testing or model testing focuses on verification effort on smallest unit of software design module. Using the detailed description as guide, important control parts are tested to uncover errors within the boundary of the module.

We have done unit testing after completion of each and every module. This module is found to be working satisfactorily as regards to the expected output from the module.

Remote Desktop Capturing

The IP address of the remote system was given and then a connection was established. The desktop of the remote system was then successfully captured.

Device control and Programming

The status of the device was stored and updated in a file .The device was controlled accordingly.

Interaction over the Internet

The current status of the device was displayed to the client and the appropriate changes hence made are reflected.

Integration testing

Integration testing is a systematic testing technique for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface.

In this testing all the modules were integrated, compared and tested as whole sets of errors that are encountered are corrected.

The client was thus able to access the device from a remote system.

User acceptance testing

The system under consideration is tested for user acceptance by constantly keeping in touch with prospected system users at the time of developing and making changes wherever necessary. This is done in regard to the following points.

- ❖ Input and output screen design
- ❖ Online message to guide user
- ❖ Menu driven system
- ❖ Design of test data

Initially artificial data was be keyed in and after this testing is over, live data was keyed in from the source documents. Some modifications suggested by the user are incorporated and finally the system is proved to be working properly.

8. CONCLUSION

Computers and interfaces are playing an increasingly important role in every sector. Now, there is therefore a need for human technology to take advantage of new developments in informatics. Hence our project "Ambient Intelligence within an environment" regards to use the computer for giving its customers an executive and interactive service, which is fading in the present day systems.

The necessary peripheral devices are connected to the computers through a hardware interface circuit and its driver software with the help of relays. The set up facilitates the application for the execution of complex management functions of the devices by selected application software.

Thus, our project proposal draws to the major consideration that needs to be taken into account in interacting the user with the system and devices. This is followed by the consideration of system design and development, choice of hardware and software, implementation, running and evaluating the system.

We have developed this project keeping in mind that the future has to adopt new revolutions in computer technologies. It was designed with an intension to satisfy user requirements at higher end.

FUTURE

OUTLOOK

9. FUTURE OUTLOOK

The system is a flexible system and various changes can be brought about depending on the requirement.

Security features included are very less in our system. Except for the validation of the user name and password, nothing more is provided. Hence, more security features can be added.

The hardware, that is, the device controller can be used for operating up to seven electrical devices. We have at present done it for one device only.

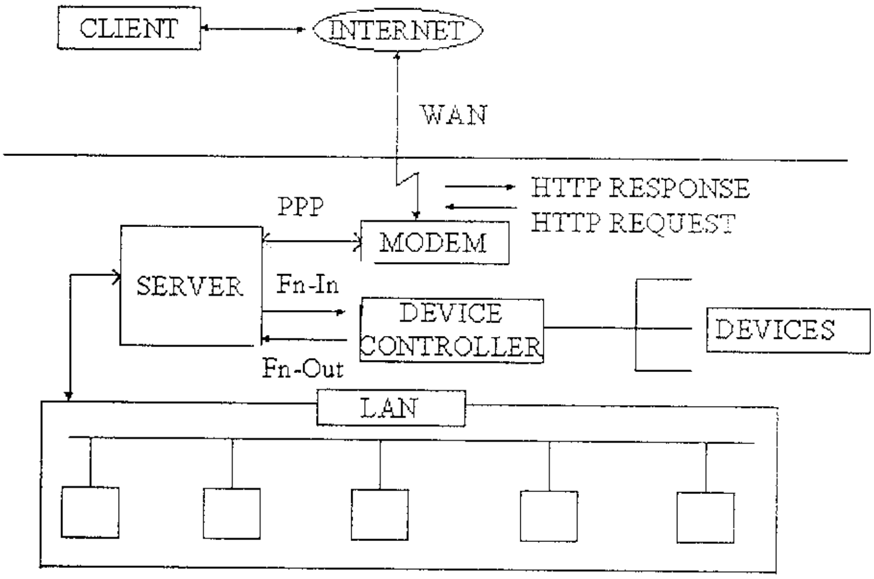
For expanding the system to switch on/off up to say, fifteen devices, one may use additional address line after replacing the ICs and by providing additional latches and NOR gates.

Also, we have one limitation, that is, the server should be operating for the twenty-four hours of the day so that the user can control the appliances. In future, this can be checked out with a better alternative.

Otherwise, we have tried our level best in developing a better system by satisfying all the objectives. We hop that his project has gone far ahead of the existing system features.

APPENDIX

BLOCK DIAGRAM



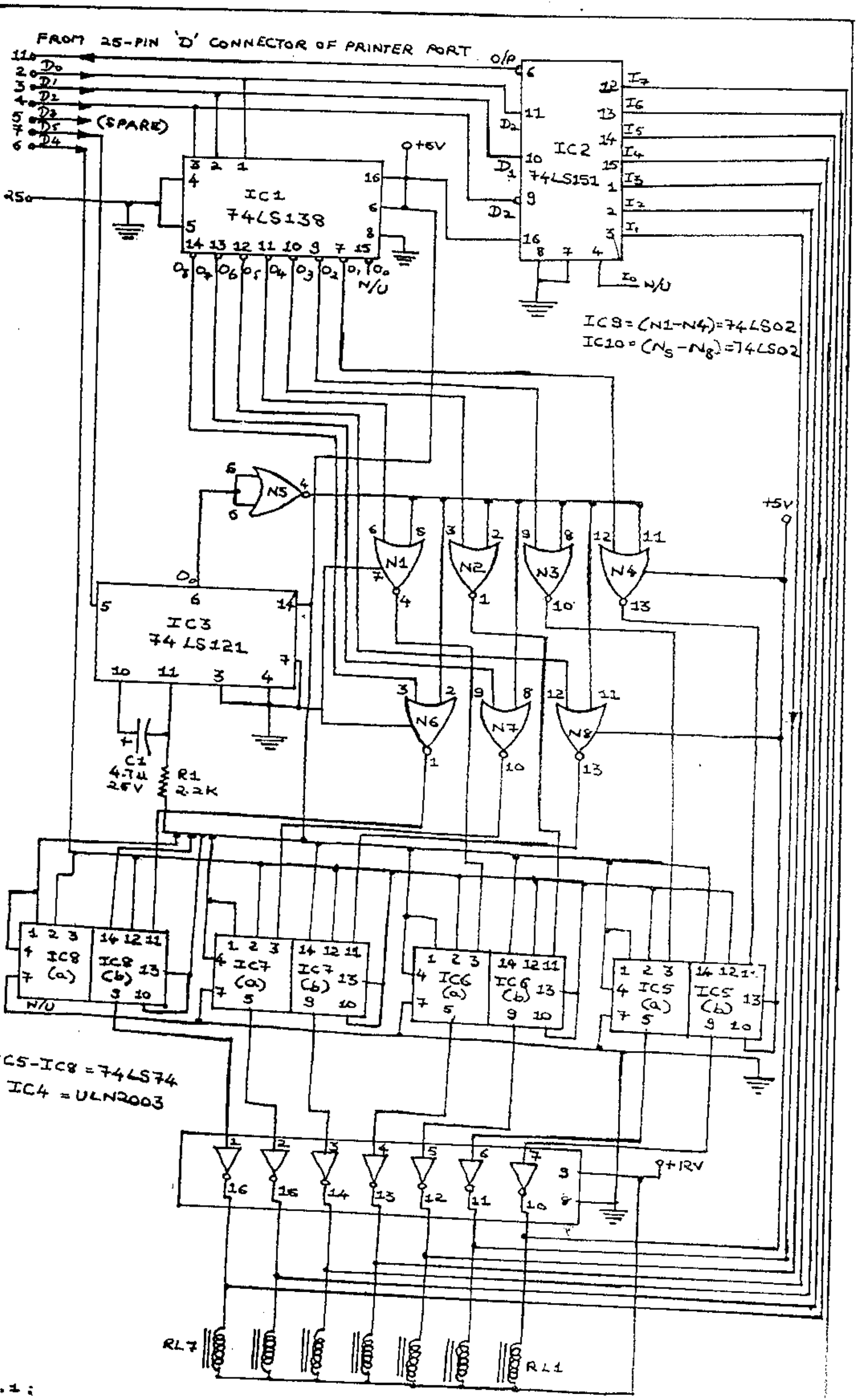
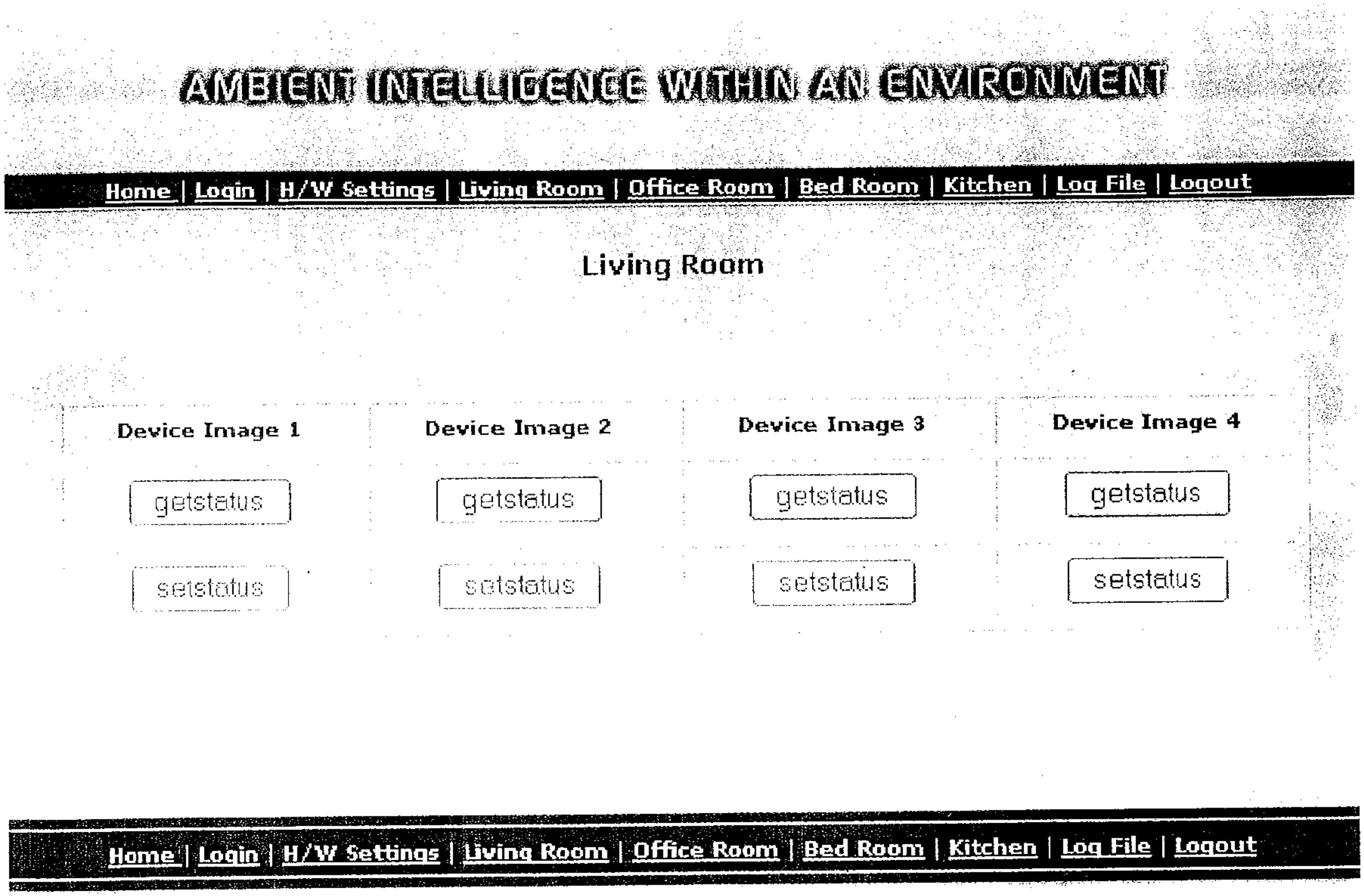
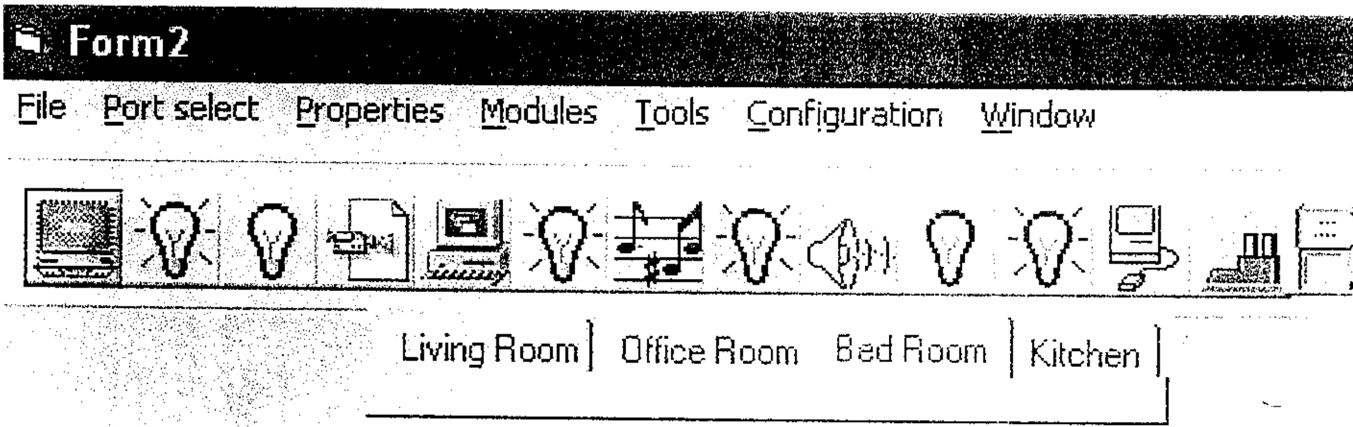


FIG. 1 :

SCHEMATIC DIAGRAM OF HARDWARE INTERFACE FOR CONTROLLING APPLIANCES.



AMBIENT INTELLIGENCE WITHIN AN ENVIRONMENT
Home | Login | H/W Settings | Living Room | Office Room | Bed Room | Kitchen | Log File | Logout
Living Room
Device Image 1
Device Image 2
Device Image 3
Device Image 4
getstatus
getstatus
getstatus
getstatus
setstatus
setstatus
setstatus
setstatus
Home | Login | H/W Settings | Living Room | Office Room | Bed Room | Kitchen | Log File | Logout



Form2
File
Port select
Properties
Modules
Tools
Configuration
Window
Living Room
Office Room
Bed Room
Kitchen

```
MODCLIENT:
Option Explicit

Public Username As String
Public Password As String

Public LocalPort As Double
Public LocalIP

Public RemoteIP As String
Public RemotePort As Double

Public TimeOut As Boolean

Public CaptureEnabled As Boolean
Public FilesystemEnabled As Boolean
Public TasksEnabled As Boolean
Public SysinfoEnabled As Boolean
Public RegistryEnabled As Boolean
'### changes to true if the server responds

Public Reply As Boolean
'### has the user logged in?
Public LogInState As Boolean
'### width and height of ther severs desktop
Public dx As Long
Public dy As Long
'### same as Reply - for mouse and keyboard-socket
Public MouseReply As Boolean
Public KbReply
'### i tested the transferrate....
'### debug ###
Public KBS As Double

'### constans which are sent to the server and back
'### this are the commands which the server interprets
Private Const Login_Request As String = "/LR"
Private Const Login_OK As String = "/LO"
Private Const Login_ERROR As String = "/LE"
Public Const GetRect As String = "/GR"


Public Function ConnectToServer() As Boolean

  With frmMain
    frmLogin.cmdConnect.Enabled = False
    frmLogin.cmdCancel.Enabled = False
    frmLogin.txtHost.Enabled = False
    frmLogin.txtPass.Enabled = False
    frmLogin.txtUser.Enabled = False
    .SB.SimpleText = "connecting..."
    .tmrTimeOut.Enabled = True
    TimeOut = False
    .wskMain.Close
    .wskMain.RemoteHost = RemoteIP
    .wskMain.RemotePort = RemotePort
    .wskMain.Connect
    Do Until .wskMain.State = sckConnected Or TimeOut = True
      DoEvents
    Loop
    .tmrTimeOut.Enabled = False

    If .wskMain.State = sckConnected Then
      ConnectToServer = True
      .SB.SimpleText = "connected"
      frmMain.mnuCapture.Enabled = True
    Else
      ConnectToServer = False
      .wskMain.Close
      frmLogin.cmdConnect.Enabled = True
      frmLogin.cmdCancel.Enabled = True
      frmLogin.txtHost.Enabled = True
      frmLogin.txtPass.Enabled = True
      frmLogin.txtUser.Enabled = True
      .SB.SimpleText = "Timeout"
    End If
  End With
End Function

Public Function Login() As Boolean
'### login
  With frmMain
    .SB.SimpleText = "logging in...please wait"
    .wskMain.SendData Login_Request & modClient.Username & modClient.Password
    Do Until modClient.Reply
      DoEvents
    Loop
    Login = modClient.LogInState
  End With
End Function

Public Function ReconnectToServer() As Boolean
```

```
End Function

Public Function Disconnect() As Boolean
   With frmMain
      .wskMain.Close
   End With
End Function

Public Sub MainDataArrival(ByVal bytesTotal As Long)
'### interpret the commands from the server
Dim Data As String
Dim Cmd As String * 3
Dim Val As String
   frmMain.wskMain.GetData Data
   Cmd = Left(Data, 3)
   Val = Mid(Data, 4)
      Select Case Cmd
         Case Login_OK: Call LogInOK        '### login is ok
         Case Login_ERROR: Call LogInErr    '### login is NOT ok
         Case GetRect: Call SetRect(Val)    '### response for the SetRect-command
      End Select
End Sub

Private Sub LogInOK()
   frmMain.SB.SimpleText = "logged on"
   modClient.LogInState = True
   modClient.Reply = True
   Unload frmLogin

   frmMain.wskFS.Close
   frmMain.wskFS.Connect modClient.RemoteIP, modClient.RemotePort + 1
   Do Until frmMain.wskFS.State = sckConnected
      DoEvents
   Loop
   frmMain.wskTasks.Close
   frmMain.wskTasks.Connect modClient.RemoteIP, modClient.RemotePort + 2
   Do Until frmMain.wskTasks.State = sckConnected
      DoEvents
   Loop
   frmMain.wskReg.Close
   frmMain.wskReg.Connect modClient.RemoteIP, modClient.RemotePort + 3
   Do Until frmMain.wskReg.State = sckConnected
      DoEvents
   Loop
   frmMain.wskConsole.Close
```

```
  frmMain.wskConsole.Connect modClient.RemoteIP, modClient.RemotePort + 4
  Do Until frmMain.wskConsole.State = sckConnected
    DoEvents
  Loop
End Sub

Private Sub LogInErr()
'### wrong login - close connections
  frmMain.SB.SimpleText = "login denied"
  frmMain.wskMain.Close
  modClient.LogInState = False
  modClient.Reply = True
  frmLogin.cmdConnect.Enabled = True
  frmLogin.cmdCancel.Enabled = True
  frmLogin.txtHost.Enabled = True
  frmLogin.txtPass.Enabled = True
  frmLogin.txtUser.Enabled = True
End Sub

Private Sub SetRect(strRect As String)
'### we get the width and height of the servers desktop
Dim aRect() As String
  aRect = Split(strRect, ";")
  dx = CLng(aRect(0))
  dy = CLng(aRect(1))
  modClient.Reply = True
End Sub
```

```
FRMLOGIN:
Option Explicit

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdConnect_Click()
'### set host and port
    modClient.RemoteIP = txtHost.Text
    modClient.RemotePort = 2208
    modClient.Username = txtUser.Text
    modClient.Password = txtPass.Text
    '### connect
    If modClient.ConnectToServer = True Then
    '### if success then log in
        Call modClient.Login
    Else


    End If

End Sub
```

```
FRMMAIN:
Option Explicit


Private Sub MDIForm_Load()
Dim Ret As Boolean
  '### extract needed temp-libraries

  On Error Resume Next
  Ret = ExtractLibs(104, "c:\zlib.dll")
End Sub

Private Sub mnuCapture_Click()
'### start capturing
  Load frmCapture
  frmCapture.Show
End Sub

Private Sub mnuConnect_Click()
'### connect
  Load frmLogin
  frmLogin.Show
End Sub

Private Sub mnuExit_Click()

  End
End Sub

Private Sub mnuVote_Click()
'

End Sub

Private Sub tmrTimeOut_Timer()
'### server did not respond within 5 seconds
'### this is a timeout
  modClient.TimeOut = True
End Sub

Private Sub wskMain_Close()
'### main connection is closed
'### so close all others
  wskMain.Close
  Unload frmCapture
  SB.SimpleText = "disconnected"
  mnuCapture.Enabled = True
```

```
End Sub

Private Sub wskMain_DataArrival(ByVal bytesTotal As Long)
'### look at modClient.bas
   modClient.MainDataArrival (bytesTotal)
End Sub
```

FRMCAPTURE:

```
Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal x As Long, ByVal y
As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc
As Long, ByVal ySrc As Long, ByVal dwRop As Long) As Long
Private Const SRCCOPY = &HCC0020          ' (DWORD) dest = source

Dim RecDib As New cDIBSection
Dim ZLib As New clsZLib
Dim i As Long
Dim tmpPos() As String
Dim Pos(1) As Long
Const CRate = 5

Private Sub Command1_Click()
'### exit capturing
   Unload frmCapture
   frmMain.Visible = True
End Sub

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
'### send the keycode of the pressed key
'### and wait for response
Dim bKeys As Byte
   bKeys = CByte(KeyCode)
   TCP_KB.SendData bKeys
   Do Until KbReply
      DoEvents
   Loop
   KbReply = False
End Sub

Private Sub Form_Load()
'### get the rect of the servers desktop
   modClient.Reply = False
   frmMain.wskMain.SendData modClient.GetRect
'### wait for reply
   Do Until modClient.Reply
      DoEvents
   Loop
   modClient.Reply = False
'### place the form in the upper left corner
   Me.Top = 0
   Me.Left = 0
'### always on top
   SetWindowPos Me.hwnd, HWND_TOPMOST, 0, 0, 0, 0, SWP_NOMOVE + SWP_NOSIZE
'### mainwindow invisible
```

```
   frmMain.Visible = False
'### same sizes as servers desktop
   Me.Width = Me.ScaleX(dx, vbPixels, vbTwips)
   Me.Height = Me.ScaleY(dy, vbPixels, vbTwips)
'### picturebox is a littlebit smaller :-)
   pVideo.Width = Me.Width - 100
   pVideo.Height = Me.Height - 200
   Call CapConnect
End Sub

Private Sub pVideo_KeyDown(KeyCode As Integer, Shift As Integer)
'### send the keycode of the pressed key
'### and wait for response
Dim bKeys As Byte
   bKeys = CByte(KeyCode)
   TCP_KB.SendData bKeys
   Do Until KbReply
      DoEvents
   Loop
   KbReply = False
End Sub

Private Sub pVideo_MouseUp(Button As Integer, Shift As Integer, x As Single, y As Single)
'### send the position and the button for a mouseclick
'### you can add functionality to say tell the server if the shift-key
'### is pressed - i have no time... *and am drunken* :-))
   Select Case Button
      Case 1
         TCP_Mouse.SendData x & ";" & y & ";" & 1
      Case 2
         TCP_Mouse.SendData x & ";" & y & ";" & 3
   End Select
   Do Until MouseReply
      DoEvents
   Loop
End Sub

Private Sub TCP_DataArrival(ByVal bytesTotal As Long)

Dim Ret As Long
Dim ByteArray() As Byte
'### okay, get the compressed bytearray
   TCP.GetData ByteArray, vbByte
'### decompress it
   Call ZLib.DecompressByte(ByteArray)
   DoEvents
```

```
'### make a dib from the bytearray
   Call RecDib.ParseByte(ByteArray)
'### and finally blit it to the actual position
   Ret = BitBlt(pVideo.hdc, Pos(0), Pos(1), dx / CRate, dy / CRate, RecDib.hdc, 0, 0, SRCCOPY)
'### send some reply - it doesn't matter what you send

   TCP.SendData "a"
End Sub

Private Sub TCP_KB_DataArrival(ByVal bytesTotal As Long)
'### server sent a reply for the keyboard - command
   KbReply = True
End Sub


Private Sub TCP_Mouse_DataArrival(ByVal bytesTotal As Long)
'### server sent a reply for the mouse - command
   MouseReply = True
End Sub

Private Sub TCP_Set_DataArrival(ByVal bytesTotal As Long)
'### server tells us the actual position for blitting
Dim tmp As String
   TCP_Set.GetData tmp, vbString
   tmpPos = Split(tmp, ";")
   Pos(0) = CLng(tmpPos(0))
   Pos(1) = CLng(tmpPos(1))
   TCP_Set.SendData "a"
End Sub

Public Sub CapConnect()
'### connect all data - sockets
Dim strHost As String
   strHost = modClient.RemoteIP
   TCP.Connect strHost, 2213
   TCP_Set.Connect strHost, 2214
   TCP_Mouse.Connect strHost, 2215
   TCP_KB.Connect strHost, 2216
   '### create a deviceindependent bitmap with 16 colors
   '### which is a fifth(crate) of ther servers desktop

   RecDib.Colors = 16
   Call RecDib.Create(dx / CRate, dy / CRate)
   i = 0
End Sub
```

```
MODMAIN:
Sub Main()

Dim Ret As Boolean
Dim strZlibPath As String
   strZlibPath = GetSysDir() & "\zlib.dll"
   '### only extract when file doesn't exist
   If Dir(strZlibPath) <> "zlib.dll" Then
      Ret = ExtractLibs(101, GetSysDir() & "\zlib.dll")
      If Ret = False Then
         '### if extraction fails, then exit program
         MsgBox "Could not extract needed libraries!", vbCritical, "Critical error"
         End
      End If
   End If
   '### load main form
   Load frmMain
   frmMain.Show
End Sub
```

MODTOOLS:

```
Public Declare Function SetWindowPos Lib "user32" (ByVal hwnd As Long, ByVal
hWndInsertAfter As Long, ByVal x As Long, ByVal y As Long, ByVal cx As Long,
ByVal cy As Long, ByVal wFlags As Long) As Long
Public Const SWP_NOMOVE = &H2
Public Const SWP_NOSIZE = &H1
Public Const HWND_TOPMOST = -1
Public Const HWND_NOTOPMOST = -2
Public Declare Function ShellExecute Lib "shell32.dll" _
    Alias "ShellExecuteA" (ByVal hwnd As Long, ByVal _
    lpOperation As String, ByVal lpFile As String, ByVal _
    lpParameters As String, ByVal lpDirectory As String, _
    ByVal nShowCmd As Long) As Long

Public Function Browser(Adresse As String, FRMhWnd As Long)

 Dim Ret As Long
  Ret = ShellExecute(FRMhWnd, "Open", Adresse, "", App.Path, 1)
End Function
```

MODDECLARES:

```
Option Explicit

Public Log As New clsLog
```

```
MODEXTRACTLIBS:
Option Explicit
Private Declare Function LoadLibrary Lib "kernel32" Alias "LoadLibraryA" (ByVal
lpLibFileName As String) As Long
Private Declare Function GetProcAddress Lib "kernel32" (ByVal hModule As Long, ByVal
lpProcName As String) As Long
Private Declare Function WaitForSingleObject Lib "kernel32" (ByVal hHandle As Long, ByVal
dwMilliseconds As Long) As Long
Private Declare Function CloseHandle Lib "kernel32" (ByVal hObject As Long) As Long
Private Declare Function CreateThread Lib "kernel32" (lpThreadAttributes As Any, ByVal
dwStackSize As Long, ByVal lpStartAddress As Long, ByVal lParameter As Long, ByVal
dwCreationFlags As Long, lpThreadID As Long) As Long
Private Declare Function GetExitCodeThread Lib "kernel32" (ByVal hThread As Long,
lpExitCode As Long) As Long
Private Declare Sub ExitThread Lib "kernel32" (ByVal dwExitCode As Long)
Private Declare Function FreeLibrary Lib "kernel32" (ByVal hLibModule As Long) As Long
Const STATUS_WAIT_0 = &H0
Public Declare Function GetInputState Lib "user32" () As Long
Declare Function GetSystemDirectory Lib "kernel32" Alias "GetSystemDirectoryA" (ByVal
lpBuffer As String, ByVal nSize As Long) As Long

Public Sub DoEventsEx()

   If GetInputState() Then DoEvents
End Sub


Public Function GetSysDir() As String
Dim strSyspath As String
   strSyspath = String(145, Chr(0))
   strSyspath = Left(strSyspath, GetSystemDirectory(strSyspath, 145))
   GetSysDir = strSyspath
End Function

Public Function ExtractLibs(intResNr As Integer, strPath As String) As Boolean

Dim intFileNumber As Integer
Dim bLibBuffer() As Byte


   On Error GoTo Errhandler
   bLibBuffer = LoadResData(intResNr, "CUSTOM")
   intFileNumber = FreeFile
   Open strPath For Binary Access Write As #intFileNumber
      Put #intFileNumber, , bLibBuffer
   Close #intFileNumber
```

```
  On Error GoTo 0
  ExtractLibs = True
  Exit Function
Errhandler:
  ExtractLibs = False
End Function

Public Function RegServe(ByVal Path$, Mode As Boolean) As Boolean

Dim insthLib&, lpLibAdr&, hThd&, lpExCode&
Dim procName$, Result&, okFlag As Boolean


  insthLib = LoadLibrary(Path)

  If insthLib Then
    If Mode Then
      procName = "DllRegisterServer"
    Else
      procName = "DllUnregisterServer"
    End If

    lpLibAdr = GetProcAddress(insthLib, procName)
    If lpLibAdr <> 0 Then

      hThd = CreateThread(ByVal 0, 0, ByVal lpLibAdr, ByVal 0&, 0&, 0&)
      If hThd Then

        Result = WaitForSingleObject(hThd, 5000)
        If Result = STATUS_WAIT_0 Then

        Call CloseHandle(hThd)
        okFlag = True
      Else

        Call GetExitCodeThread(hThd, lpExCode)
        Call ExitThread(lpExCode)
        Call CloseHandle(hThd)
      End If
    End If
  End If

  Call FreeLibrary(insthLib)
  End If
  If Not okFlag Then
    RegServe = False
  Else
    RegServe = True
  End If
End Function
```

```
MODSERVER:
Option Explicit

Public Username As String
Public Password As String

Public ClientIP As String
Public ClientPort As Double
Public LocalIP As String
Public LocalPort As Long

Public State As String



Public CaptureEnabled As Boolean
Public FilesystemEnabled As Boolean
Public TasksEnabled As Boolean
Public SysinfoEnabled As Boolean
Public RegistryEnabled As Boolean

Public Reply As Boolean



Private Const Login_Request As String = "/LR"
Private Const Login_OK As String = "/LO"
Private Const Login_ERROR As String = "/LE"
Private Const GetRect As String = "/GR"

Public Function Init() As Boolean

On Error GoTo InitError
  Log.Add "starting service...", frmMain.wskMain
  With frmMain.wskMain

     .Close

     .LocalPort = LocalPort

     .Listen
     frmCapture.TCP.Close
     frmCapture.TCP.Close
     frmCapture.TCP_Set.Close
     frmCapture.TCP_Set.Close
```

```
        frmCapture.TCP_Mouse.Close
        frmCapture.TCP_Mouse.Close
        frmCapture.TCP_KB.Close
        frmCapture.TCP_KB.Close
        frmCapture.TCP.LocalPort = 2213
        frmCapture.TCP_Set.LocalPort = 2214
        frmCapture.TCP_Mouse.LocalPort = 2215
        frmCapture.TCP_KB.LocalPort = 2216
        frmCapture.TCP.Listen
        frmCapture.TCP_Set.Listen
        frmCapture.TCP_Mouse.Listen
        frmCapture.TCP_KB.Listen
    End With
    '### TRUE zurückgeben, da kein Fehler aufgetreten ist
    Init = True
    frmMain.cmdStart.Caption = "Stop service"
    Log.Add "service started", frmMain.wskMain
On Error GoTo 0
    Exit Function
InitError:
    '### bei Fehler Port schließen und FALSE zurückgeben
    Log.Add "failed to start service - closing...", frmMain.wskMain
    With frmMain.wskMain
        .Close
        .Close
    End With
    Log.Add "all closed", frmMain.wskMain
    Init = False
End Function

Public Function Terminate()

    Log.Add "terminating session...", frmMain.wskMain
    With frmMain
        .wskMain.Close
        .wskMain.Close
        .cmdStart.Caption = "Start service"
    End With
    Log.Add "all closed", frmMain.wskMain
End Function

Public Function ReInit() As Boolean

    Log.Add "restarting service...", frmMain.wskMain
End Function
```

```
Public Function GetState() As String
   If modServer.State = "" Then

      With frmMain.wskMain
         Select Case .State
            Case sckClosed: GetState = "closed"
            Case sckClosing: GetState = "closing"
            Case sckConnected: GetState = "connected"
            Case sckConnecting: GetState = "connecting"
            Case sckListening: GetState = "listening"
         End Select
      End With
   Else

   End If
End Function

Public Sub MainConnectionRequest(requestID As Long)

   With frmMain.wskMain
      .Close
      .Accept requestID
   End With
   Log.Add "Request accepted", frmMain.wskMain
End Sub

Public Sub MainDataArrival(ByVal bytesTotal As Long)

Dim Data As String
Dim Cmd As String * 3
Dim Val As String
   frmMain.wskMain.GetData Data
   Cmd = Left(Data, 3)
   Val = Mid(Data, 4)
      Select Case Cmd
         Case Login_Request: Call CheckLogin(Val)
         Case GetRect: Call SendRect
      End Select
End Sub

Public Sub MainCloedByClient()

   Log.Add "connection closed from client", frmMain.wskMain
   modServer.Terminate
   modServer.Init
End Sub
```

```
Private Sub CheckLogin(strLogin As String)
  If strLogin = modServer.Username & modServer.Password Then
    frmMain.wskMain.SendData Login_OK
    Log.Add modServer.Username & " logged on", frmMain.wskMain
  Else
    frmMain.wskMain.SendData Login_ERROR
    Log.Add "Login denied", frmMain.wskMain
  End If
End Sub

Private Sub SendRect()
Dim Ret As Long
Dim DeskHwnd As Long
Dim DeskHdc As Long
Dim DeskRect As RECT


  DeskHwnd = GetDesktopWindow()

  DeskHdc = GetDC(DeskHwnd)

  Ret = GetWindowRect(DeskHwnd, DeskRect)
  frmMain.wskMain.SendData GetRect & CStr(DeskRect.Right) & ";" &
CStr(DeskRect.Bottom)
End Sub
```

```
SERVERFRMMAIN:
Option Explicit

Private Sub cmdStart_Click()
   cmdStart.Enabled = False
   '### wait for connections
   If cmdStart.Caption = "Start service" Then
      modServer.LocalPort = "2208"
      modServer.LocalIP = wskMain.LocalIP
      modServer.Username = txtUsername.Text
      modServer.Password = txtPassword.Text
      If modServer.Init Then
      Else
         '### an error occured
         MsgBox "Initialisation failed", vbCritical
      End If
   Else
      '### close sockets
      modServer.Terminate
   End If
   cmdStart.Enabled = True
End Sub

Private Sub Form_Load()
Dim Ret As Boolean
   '### extract needed temp-libraries
   On Error Resume Next
   Ret = ExtractLibs(104, "c:\zlib.dll")
   On Error GoTo 0
   '### load Capture Toolwindow and set it to invisible
   Load frmCapture
   frmCapture.Visible = False
End Sub

Private Sub Form_Unload(Cancel As Integer)

   Unload frmCapture
End Sub



Private Sub tmrState_Timer()

   SB.Panels(2).Text = modServer.GetState
End Sub
```

```
Private Sub wskMain_Close()

   modServer.MainCloedByClient
End Sub

Private Sub wskMain_ConnectionRequest(ByVal requestID As Long)

   modServer.MainConnectionRequest requestID
End Sub

Private Sub wskMain_DataArrival(ByVal bytesTotal As Long)

   modServer.MainDataArrival bytesTotal
End Sub
```

```
SERVERFRMCAPTURE:
Option Explicit
Private Sub Form_Load()
  Me.Left = 0
  Me.Top = 0
  Me.Caption = "Information"
  SetWindowPos Me.hwnd, HWND_TOPMOST, 0, 0, 0, 0, SWP_NOMOVE +
SWP_NOSIZE
End Sub

Private Sub TCP_Close()
  modCapture.C_Response = True
  modCapture.C_Set_Response = True
End Sub

Private Sub TCP_ConnectionRequest(ByVal requestID As Long)
  TCP.Close
  TCP.Accept requestID
  Me.Visible = True
End Sub

Private Sub TCP_DataArrival(ByVal bytesTotal As Long)
  modCapture.C_Response = True
End Sub

Private Sub TCP_KB_ConnectionRequest(ByVal requestID As Long)
  frmCapture.TCP_KB.Close
  frmCapture.TCP_KB.Accept requestID
End Sub

Private Sub TCP_KB_DataArrival(ByVal bytesTotal As Long)
Dim bKeys As Byte
  frmCapture.TCP_KB.GetData bKeys
  Call KB_KeyPress(bKeys)
  TCP_KB.SendData "a"
End Sub

Private Sub TCP_Mouse_ConnectionRequest(ByVal requestID As Long)
  frmCapture.TCP_Mouse.Close
  frmCapture.TCP_Mouse.Accept requestID
End Sub

Private Sub TCP_Mouse_DataArrival(ByVal bytesTotal As Long)
Dim strData As String
Dim aData() As String
  frmCapture.TCP_Mouse.GetData strData, vbString
  aData = Split(strData, ";")
  Select Case aData(2)
    Case 1: Call MCLeft(CLng(aData(0)), CLng(aData(1)))
    Case 3: Call MCRight(CLng(aData(0)), CLng(aData(1)))
  End Select
End Sub

Private Sub TCP_Set_ConnectionRequest(ByVal requestID As Long)
  TCP_Set.Close
  TCP_Set.Accept requestID
  Call modCapture.DoCapture
End Sub

Private Sub TCP_Set_DataArrival(ByVal bytesTotal As Long)
  modCapture.C_Set_Response = True
End Sub

Private Sub Timer1_Timer()

  frmMain.SB.Panels(3).Text = Q & " fps"
  Q = 0
End Sub
```

```
Const CRATE = 5
Dim CS(CRATE * CRATE * CRATE) As Long



Dim CS_Tmp As Long
Dim K As Long



   DeskHwnd = GetDesktopWindow()

   '### get devicecontext
   DeskHdc = GetDC(DeskHwnd)

   Ret = GetWindowRect(DeskHwnd, DeskRect)

   DIB.Colors = 16
   RecDib.Colors = 16
   Call DIB.Create(DeskRect.Right / CRATE, DeskRect.Bottom / CRATE)
   Call RecDib.Create(DeskRect.Right / CRATE, DeskRect.Bottom / CRATE)

   ENDE = False

   '### begin with first part of the desktop
   K = 0
   '### set reponse to FALSE
   modCapture.C_Response = False
   modCapture.C_Set_Response = False
   Do Until ENDE
      For yPos = 0 To DeskRect.Bottom Step (DeskRect.Bottom / CRATE)
         For xPos = 0 To DeskRect.Right Step (DeskRect.Right / CRATE)

               Ret = BitBlt(DIB.hdc, 0, 0, DeskRect.Right / CRATE, DeskRect.Bottom /
CRATE, DeskHdc, xPos, yPos, SRCCOPY)

               Call DIB.ToByte(ByteArray)

               Call ZLib.CompressByte(ByteArray)

               CS_Tmp = UBound(ByteArray)

               If CS_Tmp <> CS(K) Then
                  CS(K) = CS_Tmp
```

SERVERMODCAPTURE:

```
Option Explicit

Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal x As
Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal
hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As Long, ByVal dwRop As Long)
As Long
Private Const SRCCOPY = &HCC0020          ' (DWORD) dest = source
Private Const WHITENESS = &HFF0062        ' (DWORD) dest = WHITE

Private Declare Function GetPixel Lib "gdi32" (ByVal hdc As Long, ByVal x As Long,
ByVal y As Long) As Long
'### Desktop
Public Declare Function GetDesktopWindow Lib "user32" () As Long
Public Declare Function GetDC Lib "user32" (ByVal hwnd As Long) As Long

Public Declare Function GetWindowRect Lib "user32" (ByVal hwnd As Long, lpRect As
RECT) As Long
Public Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

Private Declare Function GetObjectAPI Lib "gdi32" Alias "GetObjectA" (ByVal hObject
As Long, ByVal nCount As Long, lpObject As Any) As Long
Private DIB As New cDIBSection
Private RecDib As New cDIBSection
Private ZLib As New clsZLib

Public C_Response As Boolean
Public C_Set_Response As Boolean
Dim ENDE As Boolean
Public Q As Double

Public Sub DoCapture()
Dim xPos As Long
Dim yPos As Long
Dim Ret As Long
Dim DeskHwnd As Long
Dim DeskHdc As Long
Dim DeskRect As RECT
Dim dibHdc As Long
Dim ByteArray() As Byte
Dim sValue As String
```

```
            On Error GoTo NoConn

            frmCapture.TCP_Set.SendData CStr(xPos) & ";" & CStr(yPos)

            Do Until C_Set_Response
              DoEvents
            Loop
            C_Set_Response = False

            frmCapture.TCP.SendData ByteArray
            '### auf Antwort warten
            '### wait for response
            Do Until C_Response
              DoEvents
            Loop
            C_Response = False
            On Error GoTo 0
          End If

          K = K + 1
          DoEvents
      Next xPos
    Next yPos

    xPos = 0
    yPos = 0

    K = 0

    Q = Q + 1
  Loop
  Exit Sub
 NoConn:
  '### an error occured and because of that: close all and go online again
  frmCapture.TCP.Close
  frmCapture.TCP_Set.Close
  frmCapture.TCP.LocalPort = modServer.LocalPort + 5
  frmCapture.TCP_Set.LocalPort = modServer.LocalPort + 6
  modServer.Terminate
  frmCapture.Visible = False
  modServer.Init
End Sub
```

SERVERMODMAIN:

```
Sub Main()

Dim Ret As Boolean
Dim strZlibPath As String
  strZlibPath = GetSysDir() & "\zlib.dll"

  If Dir(strZlibPath) <> "zlib.dll" Then
    Ret = ExtractLibs(101, GetSysDir() & "\zlib.dll")
    If Ret = False Then
      '### if extraction fails, then exit program
      MsgBox "Could not extract needed libraries!", vbCritical, "Critical error"
      End
    End If
  End If
  '### load main form
  Load frmMain
  frmMain.Show
End Sub
```

SERVERMODTOOLS:

```
Public Declare Function SetWindowPos Lib "user32" (ByVal hwnd As Long, ByVal
hWndInsertAfter As Long, ByVal x As Long, ByVal y As Long, ByVal cx As Long,
ByVal cy As Long, ByVal wFlags As Long) As Long
Public Const SWP_NOMOVE = &H2
Public Const SWP_NOSIZE = &H1
Public Const HWND_TOPMOST = -1
Public Const HWND_NOTOPMOST = -2


Public Declare Function ShellExecute Lib "shell32.dll" _
      Alias "ShellExecuteA" (ByVal hwnd As Long, ByVal _
      lpOperation As String, ByVal lpFile As String, ByVal _
      lpParameters As String, ByVal lpDirectory As String, _
      ByVal nShowCmd As Long) As Long

Public Function Browser(Adresse As String, FRMhWnd As Long)
 Dim Ret As Long
  Ret = ShellExecute(FRMhWnd, "Open", Adresse, "", App.Path, 1)
End Function
```

SERVERMODEXTRACTLIBS:

```
Option Explicit
Public Declare Function GetInputState Lib "user32" () As Long
Declare Function GetSystemDirectory Lib "kernel32" Alias "GetSystemDirectoryA"
(ByVal lpBuffer As String, ByVal nSize As Long) As Long
Private Declare Function LoadLibrary Lib "kernel32" Alias "LoadLibraryA" (ByVal
lpLibFileName As String) As Long
Private Declare Function GetProcAddress Lib "kernel32" (ByVal hModule As Long,
ByVal lpProcName As String) As Long
Private Declare Function WaitForSingleObject Lib "kernel32" (ByVal hHandle As Long,
ByVal dwMilliseconds As Long) As Long
Private Declare Function CloseHandle Lib "kernel32" (ByVal hObject As Long) As
Long
Private Declare Function CreateThread Lib "kernel32" (lpThreadAttributes As Any,
ByVal dwStackSize As Long, ByVal lpStartAddress As Long, ByVal lParameter As
Long, ByVal dwCreationFlags As Long, lpThreadID As Long) As Long
Private Declare Function GetExitCodeThread Lib "kernel32" (ByVal hThread As Long,
lpExitCode As Long) As Long
Private Declare Sub ExitThread Lib "kernel32" (ByVal dwExitCode As Long)
Private Declare Function FreeLibrary Lib "kernel32" (ByVal hLibModule As Long) As
Long
Const STATUS_WAIT_0 = &H0

Public Function ExtractLibs(intResNr As Integer, strPath As String) As Boolean
Dim intFileNumber As Integer
Dim bLibBuffer() As Byte

  '### 101 = zlib.dll("Libraries.RES")
  On Error GoTo Errhandler
  bLibBuffer = LoadResData(intResNr, "CUSTOM")
  intFileNumber = FreeFile
  Open strPath For Binary Access Write As #intFileNumber
    Put #intFileNumber, , bLibBuffer
  Close #intFileNumber
  On Error GoTo 0
  ExtractLibs = True
  Exit Function
Errhandler:
  ExtractLibs = False
End Function

Public Function RegServe(ByVal Path$, Mode As Boolean) As Boolean

Dim insthLib&, lpLibAdr&, hThd&, lpExCode&
Dim procName$, Result&, okFlag As Boolean
```

```
  insthLib = LoadLibrary(Path)

  If insthLib Then
    If Mode Then
      procName = "DllRegisterServer"
    Else
      procName = "DllUnregisterServer"
    End If

    lpLibAdr = GetProcAddress(insthLib, procName)
    If lpLibAdr <> 0 Then

      hThd = CreateThread(ByVal 0, 0, ByVal lpLibAdr, ByVal 0&, 0&, 0&)
      If hThd Then

        Result = WaitForSingleObject(hThd, 5000)
        If Result = STATUS_WAIT_0 Then

        Call CloseHandle(hThd)
        okFlag = True
      Else

        Call GetExitCodeThread(hThd, lpExCode)
        Call ExitThread(lpExCode)
        Call CloseHandle(hThd)
      End If
    End If
  End If

  Call FreeLibrary(insthLib)
  End If
  If Not okFlag Then
    RegServe = False
  Else
    RegServe = True
  End If
End Function


Public Sub DoEventsEx()
  If GetInputState() Then DoEvents
End Sub

Public Function GetSysDir() As String
'
```

```
Dim strSyspath As String
   strSyspath = String(145, Chr(0))
   strSyspath = Left(strSyspath, GetSystemDirectory(strSyspath, 145))
   GetSysDir = strSyspath
End Function
```

SERVERMODKEYBOARD:

```
Private Declare Sub keybd_event Lib "user32.dll" (ByVal bVk As Byte, ByVal bScan As
Byte, ByVal dwFlags As Long, ByVal dwExtraInfo As Long)
Const KEYEVENTF_EXTENDEDKEY = &H1
Const KEYEVENTF_KEYUP = &H2

Public Sub KB_KeyPress(KeyCode As Byte)
   '### key down
   keybd_event KeyCode, 0, 0, 0
   '### key up
   keybd_event KeyCode, 0, KEYEVENTF_KEYUP, 0
   DoEvents
End Sub
```

SERVERMODMOUSE:

```
Private Declare Function SetCursorPos Lib "user32" (ByVal x As Long, ByVal y As
Long) As Long
Private Declare Function GetCursorPos Lib "user32" (lpPoint As POINTAPI) As Long
Private Declare Sub mouse_event Lib "user32" (ByVal dwFlags As Long, ByVal dx As
Long, ByVal dy As Long, ByVal cButtons As Long, ByVal dwExtraInfo As Long)

Private Const MOUSEEVENTF_MOVE = &H1
Private Const MOUSEEVENTF_LEFTDOWN = &H2
Private Const MOUSEEVENTF_LEFTUP = &H4
Private Const MOUSEEVENTF_RIGHTDOWN = &H8
Private Const MOUSEEVENTF_RIGHTUP = &H10

Private Type POINTAPI
 x As Long
 y As Long
End Type

'### here, mousecommand are handled

Public Sub MCLeft(x As Long, y As Long)
'1
   SetCursorPos x, y
   mouse_event MOUSEEVENTF_LEFTDOWN, x, y, 0, 0
   mouse_event MOUSEEVENTF_LEFTUP, x, y, 0, 0
   DoEvents
   frmCapture.TCP_Mouse.SendData 1
End Sub

Public Sub MCRight(x As Long, y As Long)
'3
   SetCursorPos x, y
   mouse_event MOUSEEVENTF_RIGHTDOWN, x, y, 0, 0
   mouse_event MOUSEEVENTF_RIGHTUP, x, y, 0, 0
   DoEvents
   frmCapture.TCP_Mouse.SendData 3
End Sub
```

SELPORT:

```
Dim fso As FileSystemObject
Private Sub Combo1_Click()
If Combo1.SelText <> "Select" Then
Command1.Enabled = True
End If
End Sub

Private Sub Command1_Click()
  Set fso = New FileSystemObject
  Set comfile = fso.CreateTextFile(App.Path & "\log.txt")
With MSComm1
          .CommPort = Combo1.ListIndex + 1
          .Handshaking = 2 - comRTS
          .RThreshold = 2
          .RTSEnable = True
          .Settings = "9600,n,8,1"
          .SThreshold = 2
          On Error Resume Next
          .PortOpen = True
          If Err Then
           MsgBox "Port is already in use.", vbCritical
           Exit Sub
           End If

End With
comfile.Write (Combo1.ListIndex + 1)
MSComm1.PortOpen = False
Command1.Enabled = False
Combo1.Text = "Select"
Combo1.Enabled = False
Unload Me
Form5.Show

End Sub



Private Sub Form_Load()

Combo1.AddItem "comm1"
Combo1.AddItem "comm2"
Combo1.AddItem "comm3"
Combo1.AddItem "comm4"
```

```
End Sub



FRONT:
Private Sub Command26_Click()

End Sub

Private Sub MEXIT_Click()
End
End Sub

Private Sub mopen_Click()
Load Form1
Form1.Show
End Sub

Private Sub Mown_Click()
Load Form9
Form9.Show
End Sub

Private Sub MSETTING_Click()
Load Form9
Form9.Show
End Sub

Private Sub SSTab1_DblClick()
Load Form9
Form9.Show
End Sub

DEVSENDS:
Private Sub Command1_Click()

Unload Me
Form8.Show

End Sub

Private Sub Command2_Click()

End Sub
```

```
Private Sub Form_Load()
MsgBox "choose menu to open the comm port"
End Sub

Private Sub mopen_Click()
 On Error Resume Next
   Dim OpenFlag

   MSComm1.PortOpen = Not MSComm1.PortOpen
   If Err Then MsgBox Error$, 48

   OpenFlag = MSComm1.PortOpen

   mnuOpen.Checked = OpenFlag
   mnuSendText.Enabled = OpenFlag
   tbrToolBar.Buttons("TransmitTextFile").Enabled = OpenFlag

   If MSComm1.PortOpen Then
      ' Enable dial button and menu item
      mnuDial.Enabled = True
      tbrToolBar.Buttons("DialPhoneNumber").Enabled = True

      ' Enable hang up button and menu item
      mnuHangUp.Enabled = True
      tbrToolBar.Buttons("HangUpPhone").Enabled = True

      imgConnected.ZOrder
      sbrStatus.Panels("Settings").Text = "Settings: " & MSComm1.Settings
     ' StartTiming
   Else
      ' Enable dial button and menu item
      mnuDial.Enabled = True
      tbrToolBar.Buttons("DialPhoneNumber").Enabled = True

      ' Disable hang up button and menu item
      mnuHangUp.Enabled = False
      tbrToolBar.Buttons("HangUpPhone").Enabled = False

      imgNotConnected.ZOrder
      sbrStatus.Panels("Settings").Text = "Settings: "
     ' StopTiming
   End If


 End Sub
```

```
Private Sub mprop_Click()
Form6.Show vbModal

End Sub

Private Sub MSComm1_OnComm()

   Dim EVMsg$
   Dim ERMsg$

   ' Branch according to the CommEvent property.
   Select Case MSComm1.CommEvent
      ' Event messages.
      Case comEvReceive
         Dim Buffer As Variant
         Buffer = MSComm1.Input
         Debug.Print "Receive - " & StrConv(Buffer, vbUnicode)
        'ShowData Text1, (StrConv(Buffer, vbUnicode))
      Case comEvSend
      Case comEvCTS
         EVMsg$ = "Change in CTS Detected"
      Case comEvDSR
         EVMsg$ = "Change in DSR Detected"
      Case comEvCD
         EVMsg$ = "Change in CD Detected"
      Case comEvRing
         EVMsg$ = "The Phone is Ringing"
      Case comEvEOF
         EVMsg$ = "End of File Detected"

      ' Error messages.
      Case comBreak
         ERMsg$ = "Break Received"
      Case comCDTO
         ERMsg$ = "Carrier Detect Timeout"
      Case comCTSTO
         ERMsg$ = "CTS Timeout"
      Case comDCB
         ERMsg$ = "Error retrieving DCB"
      Case comDSRTO
         ERMsg$ = "DSR Timeout"
      Case comFrame
         ERMsg$ = "Framing Error"
      Case comOverrun
```

```
            ERMsg$ = "Overrun Error"
         Case comRxOver
            ERMsg$ = "Receive Buffer Overflow"
         Case comRxParity
            ERMsg$ = "Parity Error"
         Case comTxFull
            ERMsg$ = "Transmit Buffer Full"
         Case Else
            ERMsg$ = "Unknown error or event"
      End Select

      If Len(EVMsg$) Then
         ' Display event messages in the status bar.
         ' sbrStatus.Panels("Status").Text = "Status: " & EVMsg$

         ' Enable timer so that the message in the status bar
         ' is cleared after 2 seconds
'        Timer2.Enabled = True

      ElseIf Len(ERMsg$) Then
         ' Display event messages in the status bar.
         sbrStatus.Panels("Status").Text = "Status: " & ERMsg$

         ' Display error messages in an alert message box.
         Beep
         Ret = MsgBox(ERMsg$, 1, "Click Cancel to quit, OK to ignore.")

         ' If the user clicks Cancel (2)...
         If Ret = 2 Then
            MSComm1.PortOpen = False    ' Close the port and quit.
         End If

         ' Enable timer so that the message in the status bar
         ' is cleared after 2 seconds
         Timer2.Enabled = True
   End If
End Sub


Private Sub Text1_KeyPress(KeyAscii As Integer)

   ' If the port is opened...
   If MSComm1.PortOpen Then
      ' Send the keystroke to the port.
      MSComm1.Output = Chr$(KeyAscii)
      ' Unless Echo is on, there is no need to
      ' let the text control display the key.
      ' A modem usually echos back a character
'     If Not Echo Then
'         Place position at end of terminal
'        Text1.SelStart = Len(Text1)
'        KeyAscii = 0
'     End If
    End If

End Sub


DEVCTRL:
Declare Sub vbOut Lib "WIN95IO.DLL" (ByVal nPort As Integer, ByVal nData As Integer)
Declare Sub vbOutw Lib "WIN95IO.DLL" (ByVal nPort As Integer, ByVal nData As Integer)
Declare Function vbInp Lib "WIN95IO.DLL" (ByVal nPort As Integer) As Integer
Declare Function vbInpw Lib "WIN95IO.DLL" (ByVal nPort As Integer) As Integer

PDEVICE:
Public fl  As FileSystemObject
Public tfile As TextStream
Public sfile As TextStream
Public i As Integer
Public tt As String
Public status As String
Private Sub close_Click()
Unload Me
Form2.Show
'tfile.close
End Sub

Private Sub cs_Click()
i = vbInp(889)
If i = 127 Then
MsgBox "The device is On"
status = "ON"
End If
If i = 255 Then
MsgBox "The device is Off"
status = "OFF"
End If
'tfile.Write status
End Sub
```

```
Private Sub Form_Load()
'Set fl = CreateObject("scripting.FileSystemObject")
'Set tfile = fl.OpenTextFile("\\server\user\kavy\sam.txt", ForWriting)

'Set tfile = fl.CreateTextFile("\\server\user\kavy\sam.txt", True)
 End Sub

Private Sub Off_Click()
Command3.Visible = False
Command2.Visible = True
'Set tfile = fl.OpenTextFile("\\server\user\kavy\sam.txt", ForWriting)
i = MsgBox("Are you sure you want to turn off the device", vbYesNo, "Attention")
If i = vbYes Then
MsgBox "bb"
Command3.Visible = True
Command2.Visible = False
vbOut 888, 0
status = "OFF"
End If
'Dim t
't = Now()
' tfile.Write ("Off")
 'tfile.Write (t)
'tfile.WriteBlankLines (1)
Open App.Path & "\sam.txt" For Output As #1
Print #1, "OFF"
Close #1
End Sub

Private Sub On_Click()
Command2.Visible = False
Command3.Visible = True
'Set tfile = fl.OpenTextFile("\\server\user\kavy\sam.txt", ForWriting)
i = MsgBox("Are you sure you want to turn the device on", vbYesNo, "Attention")
If i = vbYes Then
Command2.Visible = True
Command3.Visible = False
MsgBox "hello"
vbOut 888, 255
status = "ON"
End If
'Dim t
't = Now()
' tfile.Write ("On")
' tfile.Write (t)
'tfile.WriteBlankLines (1)
```

```
Open App.Path & "\sam.txt" For Output As #1
Print #1, "ON"
Close #1

 End Sub

Private Sub Timer1_Timer()
'open for reading
Open App.Path & "\sam.txt" For Input As #1
Do Until EOF(1)
   Line Input #1, temp
Loop
Close #1
   If UCase(temp) <> status Then
   status = UCase(temp)
   If status = "ON" Then
   vbOut 888, 255
        ' MsgBox "CIRCUIT ON"
   ElseIf status = "OFF" Then
     'MsgBox "CIRCUIT OFF"
     vbOut 888, 0
   End If

        End If

        '   If status = "ON" Then
'  vbOut 888, 255
'       MsgBox "CIRCUIT ON"
'  ElseIf status = "OFF" Then
'    MsgBox "CIRCUIT OFF"
'    vbOut 888, 0
'  End If




'Dim ee As String
'

'Set sfile = f1.OpenTextFile("\\server\user\kavy\sam.txt", ForReading)
'If sfile.AtEndOfStream = False Then
```

```
'ee = sfile.ReadAll
'End If
'tr = Right(ee, 3)
"MsgBox tr
'If LTrim(tr) <> LTrim(status) Then
'If tr = "Off" Then
"vbout(888,255)
"   MsgBox "send data to circuit to turn device off"
'Else
"  MsgBox "send data to circuit to send the device on"
" vbout(888,127)
'End If
'End If
'sfile.close
'
'Sleep (500)
  End Sub
'
PROP:
Private Sub Form_Load()


Text1.Enabled = False
Text2.Enabled = False
Text3.Enabled = False
Text4.Enabled = False
Text5.Enabled = False
Text6.Enabled = False
Text7.Enabled = False
Text8.Enabled = False



End Sub

PROPERTIES:
Private iFlow As Integer, iTempEcho As Boolean


Sub LoadPropertySettings()
Dim i As Integer, Settings As String, Offset As Integer

' Load Port Settings
For i = 1 To 16
   cboPort.AddItem "Com" & Trim$(Str$(i))
Next i
```

```
' Load Speed Settings
cboSpeed.AddItem "110"
cboSpeed.AddItem "300"
cboSpeed.AddItem "600"
cboSpeed.AddItem "1200"
cboSpeed.AddItem "2400"
cboSpeed.AddItem "4800"
cboSpeed.AddItem "9600"
cboSpeed.AddItem "14400"
cboSpeed.AddItem "19200"
cboSpeed.AddItem "28800"
cboSpeed.AddItem "38400"
cboSpeed.AddItem "56000"
cboSpeed.AddItem "57600"
cboSpeed.AddItem "115200"
cboSpeed.AddItem "128000"
cboSpeed.AddItem "256000"

' Load Data Bit Settings
cboDataBits.AddItem "4"
cboDataBits.AddItem "5"
cboDataBits.AddItem "6"
cboDataBits.AddItem "7"
cboDataBits.AddItem "8"

' Load Parity Settings
cboParity.AddItem "Even"
cboParity.AddItem "Odd"
cboParity.AddItem "None"
cboParity.AddItem "Mark"
cboParity.AddItem "Space"

' Load Stop Bit Settings
cboStopBits.AddItem "1"
cboStopBits.AddItem "1.5"
cboStopBits.AddItem "2"

' Set Default Settings

Settings = Form5.MSComm1.Settings

' In all cases the right most part of Settings will be 1 character
' except when there are 1.5 stop bits.
If InStr(Settings, ".") > 0 Then
   Offset = 2
```

```
Else
   Offset = 0
End If

cboSpeed.Text = Left$(Settings, Len(Settings) - 6 - Offset)
Select Case Mid$(Settings, Len(Settings) - 4 - Offset, 1)
Case "e"
   cboParity.ListIndex = 0
Case "m"
   cboParity.ListIndex = 1
Case "n"
   cboParity.ListIndex = 2
Case "o"
   cboParity.ListIndex = 3
Case "s"
   cboParity.ListIndex = 4
End Select

cboDataBits.Text = Mid$(Settings, Len(Settings) - 2 - Offset, 1)
cboStopBits.Text = Right$(Settings, 1 + Offset)

cboPort.ListIndex = Form5.MSComm1.CommPort - 1

optFlow(Form5.MSComm1.Handshaking).Value = True
If Echo Then
   optEcho(1).Value = True
Else
   optEcho(0).Value = True
End If

End Sub


Private Sub cmdCancel_Click()
Unload Me
End Sub

Private Sub cmdOK_Click()
Dim OldPort As Integer, ReOpen As Boolean

On Error Resume Next

Echo = iTempEcho
OldPort = Form5.MSComm1.CommPort
NewPort = cboPort.ListIndex + 1
```

```
If NewPort <> OldPort Then                ' If the port number changes, close the old port.
  If Form5.MSComm1.PortOpen Then
      Form5.MSComm1.PortOpen = False
      ReOpen = True
  End If

  Form5.MSComm1.CommPort = NewPort        ' Set the new port number.

  If Err = 0 Then
    If ReOpen Then
      Form5.MSComm1.PortOpen = True
      Form5.mopen.Checked = Form5.MSComm1.PortOpen
      'Form5.mnuSendText.Enabled = Form5.MSComm1.PortOpen
      'Form5.tbrToolBar.Buttons("TransmitTextFile").Enabled =
frmTerminal.MSComm1.PortOpen
    End If
  End If

  If Err Then
    MsgBox Error$, 48
    Form5.MSComm1.CommPort = OldPort
    Exit Sub
  End If
End If


Form5.MSComm1.Settings = Trim$(cboSpeed.Text) & "," & Left$(cboParity.Text, 1) _
  & "," & Trim$(cboDataBits.Text) & "," & Trim$(cboStopBits.Text)

If Err Then
  MsgBox Error$, 48
  Exit Sub
End If

Form5.MSComm1.Handshaking = iFlow
If Err Then
  MsgBox Error$, 48
  Exit Sub
End If

SaveSetting App.Title, "Properties", "Settings", Form5.MSComm1.Settings
SaveSetting App.Title, "Properties", "CommPort", Form5.MSComm1.CommPort
SaveSetting App.Title, "Properties", "Handshaking", Form5.MSComm1.Handshaking
SaveSetting App.Title, "Properties", "Echo", Echo

Unload Me
```

```
End Sub

Private Sub Form_Load()

' Set the form's size
Me.Left = (Screen.Width - Me.Width) / 2
Me.Top = (Screen.Height - Me.Height) / 2

' Size the frame to fit in the tabstrip control
fraSettings.Move tabSettings.ClientLeft, tabSettings.ClientTop

' Make sure the frame is the top most control
fraSettings.ZOrder

' Load current property settings
LoadPropertySettings

End Sub


Private Sub optEcho_Click(Index As Integer)
If Index = 1 Then
   iTempEcho = True
Else
   iTempEcho = False
End If
End Sub

Private Sub optFlow_Click(Index As Integer)
iFlow = Index
End Sub
```

CANCEL:

```
DefInt A-Z
Option Explicit
Dim cancelsend
Const SWP_NOMOVE = &H2
Const SWP_NOSIZE = &H1

Private Sub Command1_Click()
  cancelsend = True
End Sub

Private Sub Form_Activate()
  ' Make this form a floating window that is always on top.
  SetWindowPos hWnd, -1, 0, 0, 0, 0, SWP_NOMOVE Or SWP_NOSIZE
End Sub

Private Sub Form_Deactivate()
  If cancelsend = False Then
    Form7.Show
  End If
End Sub
```

LOGIN

```
<html>
<head>
<title>login form</title>
</head>
<body bgcolor=RGB(213,213,255) text="white" topmargin=0>
<BR><center>
<img src=title.jpg></center><BR>

<table height=20 width=100% bgcolor=navy cellspacing=0 cellpadding=0 border=1>
<Tr><Td><center>
<font face=verdana color=white size=1><b> Home |
Login |
Settings |
Living Room |
Office Room |
Bed Room |
Kitchen |
Log File |
Logout
</td></tr></table>

<form method="post" action="http:\\localhost\nithya\login.asp">
<br><br>
<table border=1 align=right>
<tr>    <td><font face=verdana color=000000 size=1><b>Login Name:</b></td>
        <td><input name="username" type=text ></td>
</tr><tr>
        <td><font face=verdana color=000000 size=1><b>Enter Password:</b></td>
        <td><input name="passwrd" type=password passwordchar="&" ></td>
</tr><tr colspan=2>
      <TD>
        <td><input type="submit" value="login  "><input type="reset"></td>
</tr></table>

<BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR>

<table height=20 width=100% bgcolor=navy cellspacing=0 cellpadding=0 border=1>
<Tr><Td><center>
<font face=verdana color=white size=1><b> Home |
Login |
Settings |
Living Room |
Office Room |
Bed Room |
Kitchen |
Log File |
Logout
</td></tr></table>
```

LIVING ROOM

```
<html>
 <head>
      <title>login form</title>
      <script language="vbscript">

      sub status
        dim fso
        dim f1,a
        dim form1
        Set form1=document.form1
        Set fso=CreateObject("Scripting.FileSystemObject")
        set f1=fso.opentextfile("c:\inetpub\wwwroot\nithya\sam.txt",1)
          a=f1.read(2)
          if a="ON" then
           msgbox "light is On"
          else
           msgbox "light is Off"
          end if
      end sub


       sub change
          dim fso
          dim f1,a
          dim form1
          Set form1=document.form1
          Set fso=CreateObject("Scripting.FileSystemObject")
          set f1=fso.opentextfile("c:\inetpub\wwwroot\nithya\sam.txt",1)
             a=f1.read(3)
msgbox a
                    f1.cbse
         set f1=fso.opentextfile("c:\inetpub\wwwroot\nithya\sam.txt",2)
msgbox a
                      if a="ON" then
            f1.writeline "OFF"
            msgbox "u switched the light off"
           elseif a="OFF" then
            f1.write "ON"
            msgbox "u switched on the light"
          end if
          f1.close
          end sub

    </script>
       </head>
           <body bgcolor=RGB(213,213,255) text="white" topmargin=0>
           <BR><center>
           <img src=title.jpg></center><BR>
<table height=20 width=100% bgcolor=navy cellspacing=0 cellpadding=0 border=1>
```

```
          <Tr><Td><center>
 <a href="login.html"><font face=verdana color=white size=1><b> Home </a>|
 <a href="login.html"><font face=verdana color=white size=1><b> Login</a> |
 <a href="settings.html"><font face=verdana color=white size=1><b> Settings</a> |
 <a href="livingroom.html"><font face=verdana color=white size=1><b> Living
 Room</a> |
 <a href="officeroom.html"><font face=verdana color=white size=1><b> Office
 Room</a> |
 <a href="bedroom.html"><font face=verdana color=white size=1><b> Bed
 Room</a> |
 <a href="kitchen.html"><font face=verdana color=white size=1><b> Kitchen</a> |
 <a href="logview.html"><font face=verdana color=white size=1><b> Log File</a> |
 <a href="logout.html"><font face=verdana color=white size=1><b> Logout</a>
         </td></tr></table><BR>

 <center><font face=verdana size=2 color=navy><B>Living Room</b>
 <form method="post" name="form1"  >
 <br><br>

 <table border=1 align=center width=80% cellspacing=5 cellpadding=5>
   <tr>
          <td><font face=verdana color=000000
 size=1><center><b>Television</b></center></td>
          <td><font face=verdana color=000000
 size=1><center><b>Light</b></center></td>
          <td><font face=verdana color=000000
 size=1><center><b>Bulb</b></center></td>
          <td><font face=verdana color=000000
 size=1><center><b>VCR</b></ce nter></td>
  </tr>
  <tr>
         <td><center><input type=submit value="getstatus" OnClick="status"
 ></center></td>
         <td><center><input type=submit value="getstatus" ></center></td>
    <td><center><input type=submit value="getstatus" ></center></td>
    <td><center><input type=submit value="getstatus" ></center></td>
 </tr>
 <tr>
    <td><center><input type=submit value="setstatus"
 OnClick="change"></center></td>
         <td><center><input type=submit value="setstatus" ></center></td>
    <td><center><input type=submit value="setstatus" ></center></td>
    <td><center><input type=submit value="setstatus" ></center></td>
 </tr>
 </table>
  </form>

<BR><BR><BR><BR>
```

```
<table height=20 width=100% bgcolor=navy cellspacing=5 cellpadding=5 border=1>

<Tr><Td><center>
<a href="login.html"><font face=verdana color=white size=1><b> Home </a>|
<a href="login.html"><font face=verdana color=white size=1><b> Login</a> |
<a href="settings.html"><font face=verdana color=white size=1><b>  Settings</a> |
<a href="livingroom.html"><font face=verdana color=white size=1><b> Living
Room</a> |
<a href="officeroom.html"><font face=verdana color=white size=1><b> Office
Room</a> |
<a href="bedroom.html"><font face=verdana color=white size=1><b> Bed
Room</a> |
<a href="kitchen.html"><font face=verdana color=white size=1><b> Kitchen</a> |
<a href="logview.html"><font face=verdana color=white size=1><b> Log File</a> |
<a href="logout.html"><font face=verdana color=white size=1><b> Logout</a>
</td></tr></table><BR>

</body>
</html>
```

OFFICE ROOM

```
<html>
<head>
<title>login form</title>
<script language="vbscript">
sub status
dim fso
dim f1,a
dim form1
Set form1=document.form1
Set fso=CreateObject("Scripting.FileSystemObject")
set f1=fso.opentextfile("c:\inetpub\wwwroot\nithya\sam.txt",1)
a=f1.read(3)
if a="on" then
msgbox "light is on"
else
 msgbox "light is off"
end if
end sub
sub change

dim fso
dim f1,a
dim form1
Set form1=document.form1
Set fso=CreateObject("Scripting.FileSystemObject")
set f1=fso.opentextfile("c:\inetpub\wwwroot\nithya\sam.txt",1)
a=f1.read(3)

f1.close
if a="on" then
 set f1=fso.opentextfile("c:\inetpub\wwwroot\nithya\sam.txt",2)
  f1.writeline "off"
 msgbox "u switched the light off"
elseif a="off" then
 set f1=fso.opentextfile("c:\inetpub\wwwroot\nithya\sam.txt",2)

  f1.write "on"
 msgbox "u switched on the light"
end if
end sub

</script>
</head>
<body bgcolor=RGB(213,213,255) text="white" topmargin=0>
<BR><center>
<img src=title.jpg></center><BR>
<table height=20 width=100% bgcolor=navy cellspacing=0 cellpadding=0 border=1>
<Tr><Td><center>
<a href="login.html"><font face=verdana color=white size=1><b> Home </a>|
<a href="livingroom.html"><font face=verdana color=white size=1><b> Living
Room</a> |
<a href="officeroom.html"><font face=verdana color=white size=1><b> Office
Room</a> |
<a href="bedroom.html"><font face=verdana color=white size=1><b> Bed
Room</a> |
<a href="kitchen.html"><font face=verdana color=white size=1><b> Kitchen</a> |
<a href="logview.html"><font face=verdana color=white size=1><b> Log File</a> |
<a href="logout.html"><font face=verdana color=white size=1><b> Logout</a>
</td></tr></table><BR>
<center><font face=verdana size=2 color=navy><B>Office Room</b>
<form method="post" name="form1"   >
<br><br>

<table border=1 align=center width=80% cellspacing=5 cellpadding=5>
<tr>
        <td><font face=verdana color=000000
size=1><center><b>Computer</b></center></td>
        <td><font face=verdana color=000000
size=1><center><b>Light</b></center></td>
        <td><font face=verdana color=000000 size=1><center><b>Music
system</b></center></td>
        <td><font face=verdana color=000000
size=1><center><b>Bulb</b></center></td>
</tr>
<tr>
        <td><center><input type=submit value="getstatus" OnClick="status"
></center></td>
        <td><center><input type=submit value="getstatus" ></center></td>
   <td><center><input type=submit value="getstatus" ></center></td>
   <td><center><input type=submit value="getstatus" ></center></td>
</tr>
<tr>
   <td><center><input type=submit value="setstatus"
OnClick="change"></center></td>
        <td><center><input type=submit value="setstatus" ></center></td>
   <td><center><input type=submit value="setstatus" ></center></td>
   <td><center><input type=submit value="setstatus" ></center></td>
</tr>
</table>
</form>
<BR><BR><BR><BR>

<table height=20 width=100% bgcolor=navy cellspacing=5 cellpadding=5 border=1>
<Tr><Td><center>
<a href="login.html"><font face=verdana color=white size=1><b> Home </a>|
<a href="login.html"><font face=verdana color=white size=1><b> Login</a> |
<a href="settings.html"><font face=verdana color=white size=1><b>  Settings</a> |
<a href="livingroom.html"><font face=verdana color=white size=1><b> Living
Room</a> |
<a href="bedroom.html"><font face=verdana color=white size=1><b> Bed
Room</a> |
<a href="kitchen.html"><font face=verdana color=white size=1><b> Kitchen</a> |
<a href="logview.html"><font face=verdana color=white size=1><b> Log File</a> |
<a href="logout.html"><font face=verdana color=white size=1><b> Logout</a>
</td></tr></table><BR>



</body>
</html>
```

BED ROOM

```
<html>
<head>
<title>login form</title>
<script language="vbscript">
sub status
dim fso
dim f1,a
dim form1
Set form1=document.form1
Set fso=CreateObject("Scripting.FileSystemObject")
set f1=fso.opentextfile("c:\inetpub\wwwroot\nithya\sam.txt",1)
a=f1.read(3)
if a="on" then
msgbox "light is on"
else
 msgbox "light is off"
end if
end sub
sub change

dim fso
dim f1,a
dim form1
Set form1=document.form1
Set fso=CreateObject("Scripting.FileSystemObject")
set f1=fso.opentextfile("c:\inetpub\wwwroot\nithya\sam.txt",1)
a=f1.read(3)

f1.close
if a="on" then
 set f1=fso.opentextfile("c:\inetpub\wwwroot\nithya\sam.txt",2)
  f1.writeline "off"
 msgbox "u switched the light off"
elseif a="off" then
 set f1=fso.opentextfile("c:\inetpub\wwwroot\nithya\sam.txt",2)

  f1.write "on"
 msgbox "u switched on the light"
end if
end sub

</script>
</head>
<body bgcolor=RGB(213,213,255) text="white" topmargin=0>
<BR><center>
<img src=title.jpg></center><BR>
<table height=20 width=100% bgcolor=navy cellspacing=0 cellpadding=0 border=1>
<Tr><Td><center>
```

```
<a href="login.html"><font face=verdana color=white size=1><b> Home </a>|
<a href="login.html"><font face=verdana color=white size=1><b> Login</a> |
<a href="settings.html"><font face=verdana color=white size=1><b> Settings</a> |
<a href="livingroom.html"><font face=verdana color=white size=1><b> Living
Room</a> |
<a href="officeroom.html"><font face=verdana color=white size=1><b> Office
Room</a> |
<a href="bedroom.html"><font face=verdana color=white size=1><b> Bed
Room</a> |
<a href="kitchen.html"><font face=verdana color=white size=1><b> Kitchen</a> |
<a href="logview.html"><font face=verdana color=white size=1><b> Log File</a> |
<a href="logout.html"><font face=verdana color=white size=1><b> Logout</a>
</td></tr></table><BR>
<center><font face=verdana size=2 color=navy><B>Bedroom</b>
<form method="post" name="form1"  >
<br><br>

<table border=1 align=center width=80% cellspacing=5 cellpadding=5>
<tr>
        <td><font face=verdana color=000000 size=1><center><b>Audio
system</b></center></td>
        <td><font face=verdana color=000000
size=1><center><b>Light</b></center></td>
        <td><font face=verdana color=000000
size=1><center><b>Bulb</b></center></td>
        <td><font face=verdana color=000000
size=1><center><b>A/C</b></center></td>
</tr>
<tr>
        <td><center><input type=submit value="getstatus" OnClick="status"
></center></td>
        <td><center><input type=submit value="getstatus" ></center></td>
   <td><center><input type=submit value="getstatus" ></center></td>
   <td><center><input type=submit value="getstatus" ></center></td>
</tr>
<tr>
   <td><center><input type=submit value="setstatus"
OnClick="change"></center></td>
        <td><center><input type=submit value="setstatus" ></center></td>
   <td><center><input type=submit value="setstatus" ></center></td>
   <td><center><input type=submit value="setstatus" ></center></td>
</tr>
</table>
</form>
<BR><BR><BR><BR>

<table height=20 width=100% bgcolor=navy cellspacing=5 cellpadding=5 border=1>
<Tr><Td><center>
<a href="login.html"><font face=verdana color=white size=1><b> Home </a>|
<a href="login.html"><font face=verdana color=white size=1><b> Login</a> |
<a href="settings.html"><font face=verdana color=white size=1><b>  Settings</a> |
<a href="livingroom.html"><font face=verdana color=white size=1><b> Living
Room</a> |
<a href="officeroom.html"><font face=verdana color=white size=1><b> Office
Room</a> |
<a href="bedroom.html"><font face=verdana color=white size=1><b> Bed
Room</a> |
<a href="kitchen.html"><font face=verdana color=white size=1><b> Kitchen</a> |
<a href="logview.html"><font face=verdana color=white size=1><b> Log File</a> |
<a href="logout.html"><font face=verdana color=white size=1><b> Logout</a>
</td></tr></table><BR>


 </body>
 </html>
```

KITCHEN

```
<html>
<head>
<title>login form</title>
<script language="vbscript">
sub status
dim fso
dim fl,a
dim form1
Set form1=document.form1
Set fso=CreateObject("Scripting.FileSystemObject")
set fl=fso.opentextfile("c:\inetpub\wwwroot\nithya\sam.txt",1)
a=fl.read(3)
if a="on" then
msgbox "light is on"
else
 msgbox "light is off"
end if
end sub
sub change

dim fso
dim fl,a
dim form1
Set form1=document.form1
Set fso=CreateObject("Scripting.FileSystemObject")
set fl=fso.opentextfile("c:\inetpub\wwwroot\nithya\sam.txt",1)
a=fl.read(3)

fl.close
if a="on" then
 set fl=fso.opentextfile("c:\inetpub\wwwroot\nithya\sam.txt",2)
  fl.writeline "off"
 msgbox "u switched the light off"
elseif a="off" then
 set fl=fso.opentextfile("c:\inetpub\wwwroot\nithya\sam.txt",2)

  fl.write "on"
 msgbox "u switched on the light"
end if
end sub

</script>
</head>
<body bgcolor=RGB(213,213,255) text="white" topmargin=0>
<BR><center>
<img src=title.jpg></center><BR>
<table height=20 width=100% bgcolor=navy cellspacing=0 cellpadding=0 border=1>
<Tr><Td><center>
```

```
<a href="login.html"><font face=verdana color=white size=1><b> Home </a>|
<a href="login.html"><font face=verdana color=white size=1><b> Login</a> |
<a href="settings.html"><font face=verdana color=white size=1><b> Settings</a> |
<a href="livingroom.html"><font face=verdana color=white size=1><b> Living
Room</a> |
<a href="officeroom.html"><font face=verdana color=white size=1><b> Office
Room</a> |
<a href="bedroom.html"><font face=verdana color=white size=1><b> Bed
Room</a> |
<a href="kitchen.html"><font face=verdana color=white size=1><b> Kitchen</a> |
<a href="logview.html"><font face=verdana color=white size=1><b> Log File</a> |
<a href="logout.html"><font face=verdana color=white size=1><b> Logout</a>
</td></tr></table><BR>
<center><font face=verdana size=2 color=navy><B>Kitchen</b>
<form method="post" name="form1"   >
<br><br>

<table border=1 align=center width=80% cellspacing=5 cellpadding=5>
<tr>
        <td><font face=verdana color=000000
size=1><center><b>Chimney</b></center></td>
        <td><font face=verdana color=000000
size=1><center><b>Stove</b></center></td>
        <td><font face=verdana color=000000
size=1><center><b>Bulb</b></center></td>
        <td><font face=verdana color=000000
size=1><center><b>Fridge</b></center></td>
</tr>
<tr>
        <td><center><input type=submit value="getstatus" OnClick="status"
></center></td>
        <td><center><input type=submit value="getstatus" ></center></td>
   <td><center><input type=submit value="getstatus" ></center></td>
   <td><center><input type=submit value="getstatus" ></center></td>
</tr>
<tr>
   <td><center><input type=submit value="setstatus"
OnClick="change"></center></td>
        <td><center><input type=submit value="setstatus" ></center></td>
   <td><center><input type=submit value="setstatus" ></center></td>
   <td><center><input type=submit value="setstatus" ></center></td>
</tr>
</table>
</form>
<BR><BR><BR><BR>

<table height=20 width=100% bgcolor=navy cellspacing=5 cellpadding=5 border=1>
<Tr><Td><center>
<a href="login.html"><font face=verdana color=white size=1><b> Home </a>|
<a href="login.html"><font face=verdana color=white size=1><b> Login</a> |
<a href="settings.html"><font face=verdana color=white size=1><b>  Settings</a> |
<a href="livingroom.html"><font face=verdana color=white size=1><b> Living
Room</a> |
<a href="officeroom.html"><font face=verdana color=white size=1><b> Office
Room</a> |
<a href="bedroom.html"><font face=verdana color=white size=1><b> Bed
Room</a> |
<a href="kitchen.html"><font face=verdana color=white size=1><b> Kitchen</a> |
<a href="logview.html"><font face=verdana color=white size=1><b> Log File</a> |
<a href="logout.html"><font face=verdana color=white size=1><b> Logout</a>
</td></tr></table><BR>



</body>
</html>
```

SETTINGS

```
<html>
<head>
<title>login form</title>
</head>
<body bgcolor=RGB(213,213,255) text="white" topmargin=0>
<BR><center>
<img src=title.jpg></center><BR>

<table height=20 width=100% bgcolor=navy cellspacing=0 cellpadding=0 border=1>
<Tr><Td><center>
<a href="login.html"><font face=verdana color=white size=1><b> Home </a>|
<a href="login.html"><font face=verdana color=white size=1><b> Login</a> |
<a href="settings.html"><font face=verdana color=white size=1><b>  Settings</a> |
<a href="C:\Inetpub\wwwroot\Nithya\livingroom.html "><font face=verdana
color=white size=1><b> Living Room</a> |
<a href="officeroom.html"><font face=verdana color=white size=1><b> Office
Room</a> |
<a href="bedroom.html"><font face=verdana color=white size=1><b> Bed
Room</a> |
<a href="kitchen.html"><font face=verdana color=white size=1><b> Kitchen</a> |
<a href="logview.html"><font face=verdana color=white size=1><b> Log File</a> |
<a href="logout.html"><font face=verdana color=white size=1><b> Logout</a>
</td></tr></table>
<BR><BR><BR><BR><BR><BR>

<table height=20 width=100% bgcolor=navy cellspacing=0 cellpadding=0 border=1>
<tr><th>
 <font face=verdana color=white size=2><b><u>OFFICEROOM
 <th> <font face=verdana color=white size=2><b><u>BEDROOM
  <th><font face=verdana color=white size=2><b><u>LIVING ROOM
  <th><font face=verdana color=white size=2><b><u>KITCHEN
</tr>
<tr><td>
 <font face=verdana color=white size=1><b>COMPUTER
 <td> <font face=verdana color=white size=1><b>AUDIO SYSTEM
  <td><font face=verdana color=white size=1><b>TELEVISION
  <td><font face=verdana color=white size=1><b>CHIMNEY
</tr>
<TR>
<td>
 <font face=verdana color=white size=1><b>LIGHT
 <td> <font face=verdana color=white size=1><b>LIGHT
  <td><font face=verdana color=white size=1><b>LIGHT
  <td><font face=verdana color=white size=1><b>STOVE
</tr>
<TR>
<td>
 <font face=verdana color=white size=1><b>MUSIC SYSTEM
 <td> <font face=verdana color=white size=1><b>BULB
```

```
</tr>
<TR>
<td>
 <font face=verdana color=white size=1><b>BULB
 <td> <font face=verdana color=white size=1><b>A/C
  <td><font face=verdana color=white size=1><b>VCR
  <td><font face=verdana color=white size=1><b>FRIDGE
</tr></TABLE>
<BR><BR><BR><BR><BR><BR>
<table height=20 width=100% bgcolor=navy cellspacing=0 cellpadding=0 border=1>
<Tr><Td><center>
<a href="login.html"><font face=verdana color=white size=1><b> Home </a>|
<a href="login.html"><font face=verdana color=white size=1><b> Login</a> |
<a href="settings.html"><font face=verdana color=white size=1><b>  Settings</a> |
<a href="livingroom.html"><font face=verdana color=white size=1><b> Living
Room</a> |
<a href="officeroom.html"><font face=verdana color=white size=1><b> Office
Room</a> |
<a href="bedroom.html"><font face=verdana color=white size=1><b> Bed
Room</a> |
<a href="kitchen.html"><font face=verdana color=white size=1><b> Kitchen</a> |
<a href="logview.html"><font face=verdana color=white size=1><b> Log File</a> |
<a href="logout.html"><font face=verdana color=white size=1><b> Logout</a>
</td></tr></table>
</body>
</html>
```

SETTINGS

```
<html>
<head>
<title>login form</title>
</head>
<body bgcolor=RGB(213,213,255) text="white" topmargin=0>
<BR><center>
<img src=title.jpg></center><BR>

<table height=20 width=100% bgcolor=navy cellspacing=0 cellpadding=0 border=1>
<Tr><Td><center>
<a href="login.html"><font face=verdana color=white size=1><b> Home </a>|
<a href="login.html"><font face=verdana color=white size=1><b> Login</a> |
<a href="settings.html"><font face=verdana color=white size=1><b>  Settings</a> |
<a href="C:\Inetpub\wwwroot\Nithya\livingroom.html "><font face=verdana
color=white size=1><b> Living Room</a> |
<a href="officeroom.html"><font face=verdana color=white size=1><b> Office
Room</a> |
<a href="bedroom.html"><font face=verdana color=white size=1><b> Bed
Room</a> |
<a href="kitchen.html"><font face=verdana color=white size=1><b> Kitchen</a> |
<a href="logview.html"><font face=verdana color=white size=1><b> Log File</a> |
<a href="logout.html"><font face=verdana color=white size=1><b> Logout</a>
</td></tr></table>
<BR><BR><BR><BR><BR><BR>

<table height=20 width=100% bgcolor=navy cellspacing=0 cellpadding=0 border=1>
<tr><th>
 <font face=verdana color=white size=2><b><u>OFFICEROOM
 <th> <font face=verdana color=white size=2><b><u>BEDROOM
  <th><font face=verdana color=white size=2><b><u>LIVING ROOM
  <th><font face=verdana color=white size=2><b><u>KITCHEN
</tr>
<tr><td>
 <font face=verdana color=white size=1><b>COMPUTER
 <td> <font face=verdana color=white size=1><b>AUDIO SYSTEM
  <td><font face=verdana color=white size=1><b>TELEVISION
  <td><font face=verdana color=white size=1><b>CHIMNEY
</tr>
<TR>
<td>
 <font face=verdana color=white size=1><b>LIGHT
 <td> <font face=verdana color=white size=1><b>LIGHT
  <td><font face=verdana color=white size=1><b>LIGHT
  <td><font face=verdana color=white size=1><b>STOVE
</tr>
<TR>
<td>
 <font face=verdana color=white size=1><b>MUSIC SYSTEM
 <td> <font face=verdana color=white size=1><b>BULB
```

LOGOUT

```
<html>
<head>
<title>login form</title>
</head>
<body bgcolor=RGB(213,213,255) text="white" topmargin=0>
<BR><center>
<img src=title.jpg></center><BR>

<table height=20 width=100% bgcolor=navy cellspacing=0 cellpadding=0 border=1>
<Tr><Td><center>
<a href="login.html"><font face=verdana color=white size=1><b> Home </a>|
<a href="login.html"><font face=verdana color=white size=1><b> Login</a> |
<a href="settings.html"><font face=verdana color=white size=1><b>  Settings</a> |
<a href="livingroom.asp"><font face=verdana color=white size=1><b> Living
Room</a> |
<a href="officeroom.html"><font face=verdana color=white size=1><b> Office
Room</a> |
<a href="bedroom.html"><font face=verdana color=white size=1><b> Bed
Room</a> |
<a href="kitchen.html"><font face=verdana color=white size=1><b> Kitchen</a> |
<a href="logview.html"><font face=verdana color=white size=1><b> Log File</a> |
<a href="logout.html"><font face=verdana color=white size=1><b> Logout</a>
</td></tr></table>
<BR><BR><BR><BR><BR><BR>
<TABLE align=center border=2  cellSpacing=0 width="20%">
<form method="post" action="http:\\localhost\nithya\logout.asp">
<TR bgcolor=navy>
<TD><center><input type="submit" value="Logout
completely"></center></TD></form>
</TR></table>
<BR><BR><BR><BR><BR><BR><BR><BR>
<table height=20 width=100% bgcolor=navy cellspacing=0 cellpadding=0 border=1>
<Tr><Td><center>
<a href="login.html"><font face=verdana color=white size=1><b> Home </a>|
<a href="login.html"><font face=verdana color=white size=1><b> Login</a> |
<a href="settings.html"><font face=verdana color=white size=1><b> Settings</a> |
<a href="livingroom.asp"><font face=verdana color=white size=1><b> Living
Room</a> |
<a href="officeroom.html"><font face=verdana color=white size=1><b> Office
Room</a> |
<a href="bedroom.html"><font face=verdana color=white size=1><b> Bed
Room</a> |
<a href="kitchen.html"><font face=verdana color=white size=1><b> Kitchen</a> |
<a href="logview.html"><font face=verdana color=white size=1><b> Log File</a> |
<a href="logout.html"><font face=verdana color=white size=1><b> Logout</a>
</td></tr></table>
</body>
</html>
```

GLOSSARY OF TERMS

API	-	Application Programming Interface
ASP	-	Active Server Page
CGI	-	Common Gateway Interface
COM	-	Component Object Model
GUI	-	Graphical User Interface
HTML	-	Hyper Text Markup Language
HTTP	-	Hyper Text Transfer Protocol
IC	-	Integrated Chip
LAN	-	Local Area Network
PPP	-	Point to Point Protocol
VB	-	Visual Basic
WAN	-	Wide Area Network

REFERENCES

- Stephen and Steven Banick," Active Server pages 2.0 unleashed",Macmillan Computer publishing, 1999.

- Greg Perry," Teach Yourself Visual Basic 6", G.C. Jain for Techmedia,1999