WIRELESS INTRANET

PROJECT REPORT

P-1158

Submitted by

S. Elango
0027S0709

A. Tony Francis Praveen
0027S0117

Veera Sundari M.
0027S0118

Under the guidance of

Mrs. J. Cynthia M. E.
Lecturer

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering
In
Information Technology
Of
Bharathiar University, Coimbatore.

DEPARTMENT OF INFORMATION TECHNOLOGY

KUMARAGURU COLLEGE OF TECHNOLOGY,
COIMBATORE – 641006.

KUMARAGURU COLLEGE OF TECHNOLOGY
(Affiliated to Bharathiar University, Coimbatore)

CERTIFICATE

This is to certify that the project entitled

WIRELESS INTRANET

Submitted by

S. Elango
0027S0709

A. Tony Francis Praveen
0027S0117

Veera sundari .M
0027S0118

And submitted in partial fulfillment of the
Requirement for the award of the degree of the

Bachelor of Information Technology of
Bharathiar University, Coimbatore.

Professor & Head Of the Department
(Dr.S.THANGASAMY)

Project Guide
(Mrs. J. Cynthia M.E.,)

Certified that the candidates were examined by us in the project work
Viva voce examination held on 26/3/04.

Internal Examiner

External Examiner

Vital Info System
Computers Sales, Service & Networking
Head Office : 63, Padmavathipuram • Gandhi Nagar Post • TIRUPUR - 641 603.
Branch Office : Johnson Bhavanam • 30, KRG Nagar Main Street,
• Ganapathy • COIMBATORE - 641 006. Mobile : **98422 67838** Phone : **0421 - 475218**

March 15,2004

TO WHOMSOEVER IT MAY CONCERN

This is to certify that S. Elango, A. Tony Francis Praveen and Veera sundari.M , students of Kumaraguru College Of Technology, Coimbatore have completed their project on "Wireless Intranet", from December 2003 - April 2004. Their performance is very good and satisfactory.

We wish them all success to their future endeavours.

For Vital Info Systems,

R. Vijaya Kumar

R. Vijaya Kumar

Senior Executive - Technical

Declaration

We,

S. Elango	0027s0709
M. Veerasundari	0027s0118
A. Tony Francis Praveen	0027s0117

Declare that the project entitled "**Wireless Intranet**", is done by us and to the best of our knowledge, a similar work has not been submitted earlier to the Bharathiar University or any other institution, for fulfillment of the requirement of the course study.

This project report is submitted on the partial fulfillment of the requirement for all awards of the degree of Bachelor of Engineering in Information Technology Bharathiar University.

Place: Coimbatore.

Date : 26/3/04.

S.Elango

S.Elango

M.Veerasundari

A.Tony Francis Praveen

25/03/04

Countersigned by Staff- in- charge,

Mrs. J. Cynthia M.E.,
Lecturer,
Department of Information technology,
Kumaraguru College of Technology.

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

We express our sincere thanks to our chairman **Arutchelvar Dr. N. Mahalingam, B.Sc., F .I .E.** and correspondent **Prof. K. Arumugam, B.E., M.s., M.I.E.,** for all their support and ray of strengthening hope extended.

We are immensely grateful to our principal **Dr. K. K. Padmanabhan, B.Sc., M.Tech., Ph.D.,** for his invaluable support to come outs of this project.

We are extremely thankful to **Dr. S. Thangasamy, Ph.D.,** Head of the Department, Department of Computer Science and engineering for his valuable advice and suggestions throughout this project.

We are indent to express our heartiest thanks to **Mrs. J. Cynthia, M.E.,** project guide who rendered her valuable guidance and support to perform our project work extremely well.

We are indent to express our heartiest thanks to **Mrs. S. Devaki M.E.,** project coordinator who has helped us to perform the project work extremely well.

We are extremely thankful to **Ms. P. Sudha, B.E.,** Class Advisor, for providing us a great support throughout this project.

We are also thankful to all the faculty members of the department of computer science and engineering, Kumaraguru College of Technology, CBE for their valuable guidance, support and encouragement during the course of our project work.

We express our humble gratitude and thanks to our parents and family members who have supported and helped us to complete the project and our friends, for lending us valuable tips, support and co-operation throughout our project work.

ABSTRACT

The project "Wireless Intranet" is a project done on the proposal of the company "Vital Info Systems". The current wireless networks are very expensive and so it is not being used widely. This project is a network-based project aimed at reducing the cost of wireless networks.

The proposed project deals with various modules such as,

1. Hardware module.
2. Network application module.
3. User interface module.

The hardware module involves designing a low cost transceiver circuit which is capable of transmitting and receiving data. This circuit consists of a modulator/demodulator circuit which is used for modulation and demodulation of the signals. The tuner circuit is tuned to the desired frequency.

The network module involves transmitting the data as packets by adding both source and destination address. It also involves the process of uncovering the packet and checking whether the packet is destined to that system or not, if so, it is accepted and processed else discarded.

The user interface module involves designing an application which is very user interactive. It allows the user to browse the files in the network, share the files, and transmit them to the particular system in the network. The information about the systems in the network can also be viewed.

CONTENTS

INTRODUCTION

1. INTRODUCTION

Making the wireless data transmission technology available to the common man was our aim when we started the project. Recent trends in technology have brought the price/performance ratio to a better level and our project also contributes to make it still better.

We have achieved the above stated by developing GUI interface and a wireless card that will be attached to the serial port of all the computers in a wireless network. This wireless card is responsible for transmitting and receiving the data using electromagnetic wave spectrum. Our card is capable of co-existing with current wired networks.

Graphical User Interface was designed by us on Linux platform using Glade interface. This application provides the user with facilities like sending and receiving file to the specified user. It also provides file sharing among all users in this wireless network. A unique Serial Network Address (SNA) is provided to all the systems in the network through this application. Only the super user has the rights to access the SNA. It has the provision to view the system information of all the systems in the wireless network.

Wireless card has onboard transmitter and receiver circuit built into it. The flow of data through the wireless card is controlled by the PIC microcontroller. Electromagnetic spectrum mainly consists of analog waveforms. To make use of this spectrum, the digital signals from the system are converted into analog signals and then transmitted. In receiving mode the incoming analog signals are converted into machine understandable digital signals.

2.1. EXISTING SYSTEM

Wireless communication is still in its infancy and out of common man's reach. Wireless communication is likewise becoming the standard in the business and domestic world. Remote wireless Internet connection and wireless computer networks are appearing on the scene and will dramatically impact the way business does business in the near future. Wireless revolution is drastically changing the way the world communicates.

This revolution as branched into mobile phones, wireless Ethernets and satellites. The mobile phones and satellites have experienced a mammoth growth, but it was not reflected in wireless Ethernets. Existing wireless Ethernets are being used in high profile corporate.

Currently available wireless Ethernets have a head blowing price which limits the type of users to corporate managers and rich people. Also wireless Ethernets are currently available only in developed countries. It is not the fault of the developers but the technology they use is very costly. Hence the need for a simpler technology that costs less and hence will be accessible to all users. If this simpler technology can be conceived then it is just a matter of time before the world floats in wireless technology.

2.2. PROPOSED SYSTEM

Given the revolutionary strides that wireless technology has made in the recent past, we decided to build a system, which will establish wireless communication in a network. Wireless connectivity will allow the ability to move equipment throughout an office, and will allow collaborative communication between individuals, their appliances, and their environment. A wireless workforce is one characteristic of the business world of today. With a wireless network users can access information from anywhere within the network.

The system that we have designed implements a technology which charges less than the existing one. This provides network capability to all users without ruining the wall finishes with bulky wires. It costs same as that of the wired networks. Installation and maintenance of this system is so easy that anyone can handle it. We have provided the option of tuning the frequency at which the data is transmitted so that user can choose any frequency based on his network requirements, such an option is not provided in any of the existing systems.

The Graphical User Interface application is user interactive and has been designed on Linux platform, which is still unexplored and its full potential has not yet been utilized. Linux still remains as a programmer's operating system. The purpose of this application is to send and receive files to specific user or all users present in the network. The transfer baud rate can also be controlled from this application. User can choose between COM1 and COM2 ports based on his operational requirements.

SYSTEM ANALYSIS

3. FEASIBILITY ANALYSIS

This project is feasible for the given limited resources and bounded time. In this project, the feasibility is analyzed and the system is checked for its workability and effective use of resources.

During the feasibility analysis of the project, the following phases are considered very carefully.

- Technical feasibility
- Economic feasibility
- Behavioral feasibility

3.1. Technical Feasibility

The existing Ethernet network cards are fixed to the PCI(Peripheral Component Interface) slot whereas our system will be connected to serial port making it easy to plug and play.

3.2. Economic Feasibility

Economic feasibility deals with the analysis of the cost against benefits (i.e.) whether the benefits to be enjoyed due to the new system are worthy when compared for the costs to be spent on the system.

Additional costs incurred in the system construction, maintenance to work on solution is not a difficult task as this project has no overhead and is practically very feasible. The cost of the system proposed is not high and is well within the bounds.

3.3. Behavioral Feasibility

Behavioral Feasibility talks about how the system is going to behave when deployed in a large network. There is no need to know the operational details of the hardware because all controlling operations are done through the application.

HARDWARE & SOFTWARE SPECIFICATION

4.1. HARDWARE SPECIFICATION

Motherboard	:	**Intel 386.**
Cache	:	**32MB L1 Cache.**
RAM	:	**32MB.**
Hard Disk	:	**10GB.**
Processor	:	**80486DX**
Video System	:	**VGA or SVGA.**
Keyboard	:	**101 Keys enhanced.**

4.2. SOFTWARE SPECIFICATION

Operating System	:	**Linux**
Software used	:	**GTK 2.0, C in Linux.**
Designer	:	**Glade Interface Designer.**

REQUIREMENT SPECIFICATION

5.1. INTRODUCTION

5.1.1. Purpose

The main purpose of the project titled "Wireless Intranet" was to reduce the cost incurred in existing wireless networks and to easily convert the existing wired networks into wireless networks.

5.1.2. Scope

The scope of our project is vast. When proper frequency for transmitting over wide range is obtained and with few changes in our software product, it can be extended to communicate between networks.

5.1.3. Definitions, Acronyms and Abbreviations

WSDCP - Wireless Serial Device Control Protocol

SNA – Serial Network Address

LAN – Local Area Network

RS232C – Serial port interface standard

FSK – Frequency Shift Keying

FM – Frequency Modulation

PIC – Peripheral Interface Controller

Transceiver circuit – A device capable of transmitting and receiving data within the network.

5.1.4. Overview

Our project involves a transceiver circuit capable of transmitting and receiving data within the network. The coding helps in accessing the files destined for that system and then extracting the data from the packet. The transceiver circuit acts at a particular frequency and accepts signals at that frequency.

5.2. GENERAL DESCRIPTION

5.2.1. Product Perspective

Existing wireless Ethernet cards are costly (comes around $1300) whereas our product costs around a very small amount of about Rs.550.

5.2.2. Product Functions

Software

Transmission module

This module splits the data into packets, which is then transmitted to the receiver. The packet format is designed in such a way that it contains the source IP, destination IP and the data.

Receiver module

This module was designed to accept data in the form of packets & combining it to form a file. The system checks for its packet by checking its destination IP field of the packet.

GTK module

This module is used to design the user interface. It is used to retrieve inputs, port, baud rate, file to be transferred & destination hostname. It also displays the details of the file transferred.

Hardware module

Microcontroller unit

It acts as a buffer and receives data from the transmitter and sends to the system likewise it receives data from the system and delivers it to the transmitter.

FSK

Frequency shift keying is used to convert digital signal into analog signal, thereby performing modulation operation at the transmitter end. Similarly, the received analog signal is converted into digital signal in a demodulation operation at the receiver end.

FM Transmitter

It gets the input from the FSK and tunes it to the desired frequency necessary for transmission.

FM Receiver

It gets the input from the wireless medium and gives it to FSK for demodulation.

5.2.3. User Characteristics

User is required to have a very basic knowledge about computers, he/she is only required to operate in a GUI environment.

5.2.4. General Constraints

- To transmit data as packets we need to use a temporary file.
- Transmitter and Receiver have to be tuned to the correct frequency.

5.2.5. Hardware Constraints

Interference in frequencies is a major constraint for the hardware.

5.3. SPECIFIC REQUIREMENTS

5.3.1. Functional requirements

To perform our desired function certain inputs and processing capabilities are required as stated below.

5.3.1.2. List of Inputs

- Destination system- Name of the remote system to which file is to be sent.
- Port- Defines either COM1 or COM2 serial ports.
- Baud rate- Number of bits to be transmitted per second.
- Filename- Name of the file to be transmitted.

5.3.1.3. Information processing requirements

The destination system name given as input is checked with the look up table and its corresponding destination address is obtained through a process called NTAM (Name To Address Mapping) , then the packet is transferred to that host. The baud rate input accepted determines the speed of transmission. The ports COM1 and COM2 are also selected based on the input. The file that is to be transmitted is searched and is split into packets and then transmitted.

5.3.2. Performance Requirements

5.3.2.1. Availability

Components required for performance improvements like oscillator, PIC microcontroller, regulator, trimmer, capacitor, resistor, diode, inductor etc., are available in the market easily.

5.3.2.2. Capacity

Our project is capable of transmitting a file of any size.

5.3.2.3. Response time

Response time depends upon the hardware configuration of the system, file size and the selected frequency.

5.3.3. Design constraints

5.3.3.1. Standards compliance

There is no existing standards for our product and so we have developed our Own standard i.e., WSDCP (Wireless Serial Device Control Protocol).

5.3.3.2. Hardware Limitations

The transceiver circuit that has been designed by us is limited to a Transmission capacity of about 9Kbps.

5.3.3.3. External interface requirements

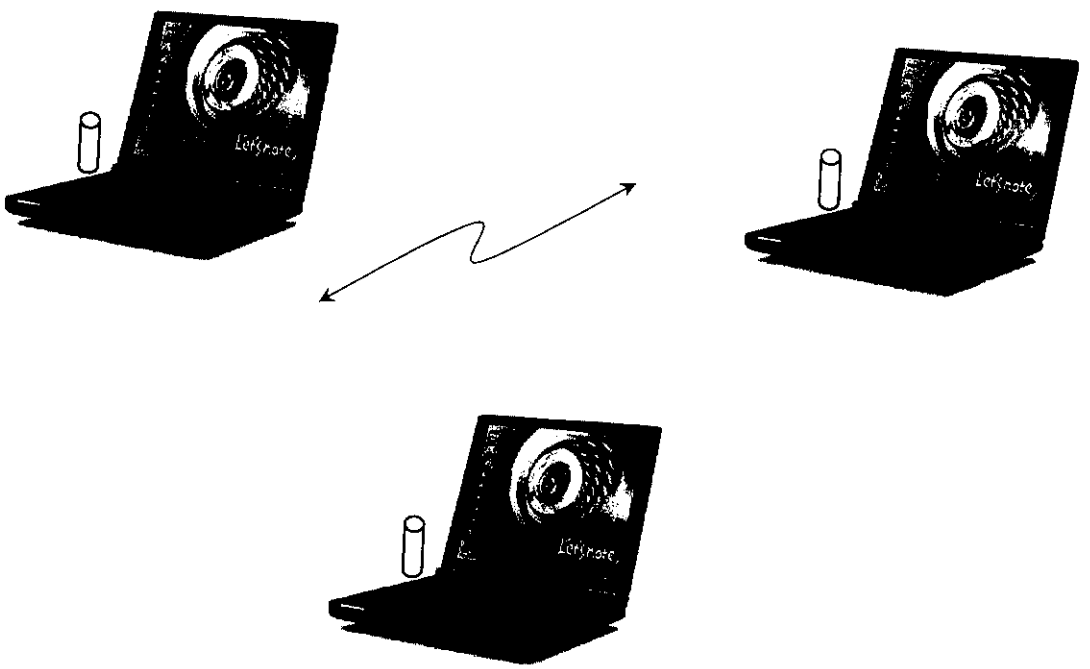
Transceiver circuit is the external interface required to transmit data between systems in the network. The transceiver circuit is attached to all the systems through the serial port.

5.3.3.4. User interfaces

Dialog boxes to

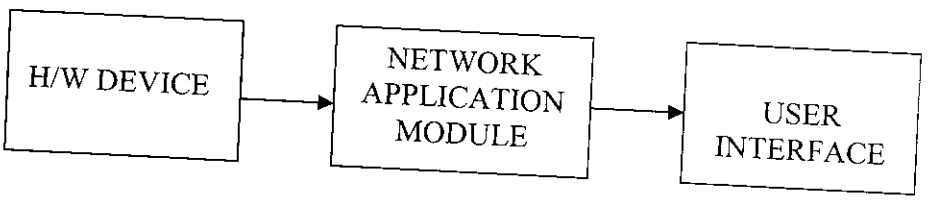
- Select the file.
- Select the operation.

5.4. OPERATIONAL DIAGRAM



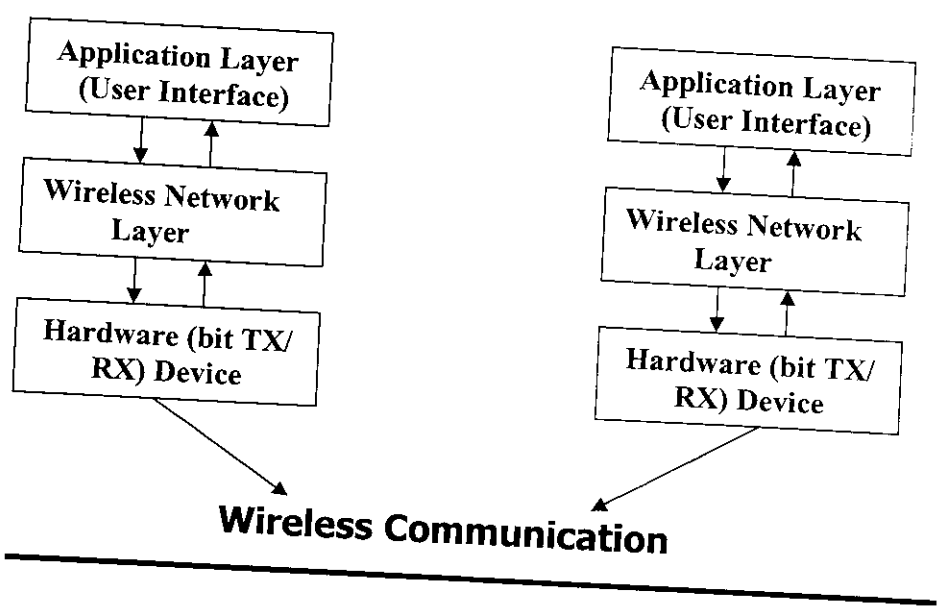
The above diagram describes the general operation of the project "Wireless LAN". The wireless card is connected to the serial port of each system in the network and the transceiver antennae are projected to send and receive the signals. The user interface application is installed in each system and configured as per requirements. Communication is done using electromagnetic waveguides.

5.5. SYSTEM FLOW DIAGRAM



The electromagnetic signal is received by the hardware device according to the tuned frequency. It is then verified whether it belongs to that particular system by checking the destination address part of the packet by the network application module. Also the addresses are truncated and original data stream is given to the user interface. User interface builds the original file.

Wireless Layered Architecture (WLA):



5.6. SYSTEM DESIGN

The design phase has been split into two modules namely

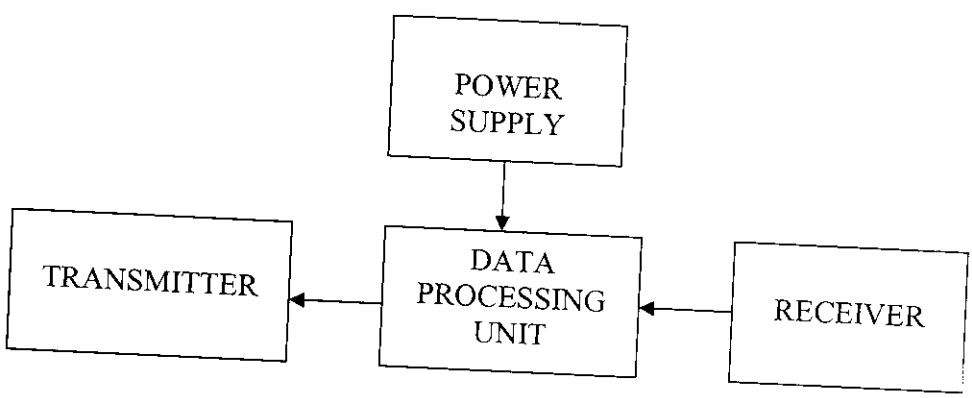
5.6.1 Hardware Design

5.6.2 Software Design

5.6.1 Hardware Design

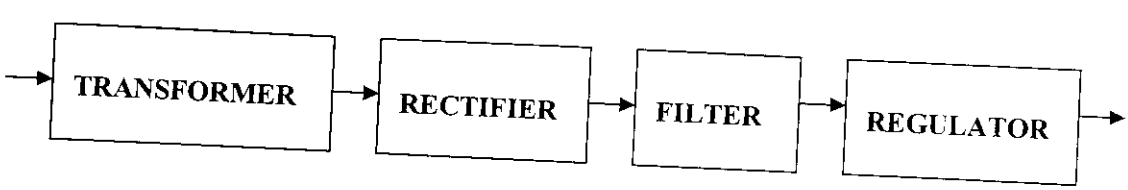
Hardware design deals with the circuitry of Power supply, Transmitter, Transceiver (Data Processing Unit) and Receiver.

General Hardware Block diagram:



5.6.1.1 Power Supply

Block Diagram



All electronic circuits work only with low D.C. voltage. Therefore we need a power supply unit to provide the appropriate voltage supply. This unit consists of

transformer, rectifier, filter and regulator. A.C. voltage typically 230V rms is connected to a transformer which steps that AC voltage down to the level to the desired AC voltage. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a DC voltage. This resulting DC voltage usually has some ripple or AC voltage variations. A regulator circuit can use this DC input to provide DC voltage that not only has much less ripple voltage but also remains the same DC value even if the DC voltage varies or the load connected to the output DC voltages changes.

Transformer

A transformer is a static (or stationary) piece with which electric power in one circuit is transformed into electric power of the same frequency in another circuit. It can raise or lower the voltage in a circuit but with a corresponding decrease or increase in current. It works with the principle of mutual induction. In our project we are using step down transformer for providing a necessary supply for the electronic circuits. We are using a 12-0-12 center tapped transformer.

Rectifier

The DC level obtained from a sinusoidal input can be improved 100% using a process called full-wave rectification. It uses 4 diodes in a bridge configuration. From the basic bridge configuration we see that two diodes (say D2 & D3) are conducting while the other two diodes (say D2 & D3) are conducting while the other two diodes (D1 & D4) are in "off" state during the period t=0 to T/2. Accordingly for the negative of the input the conducting diodes are D1 & D4. Thus the polarity across the load is the same.

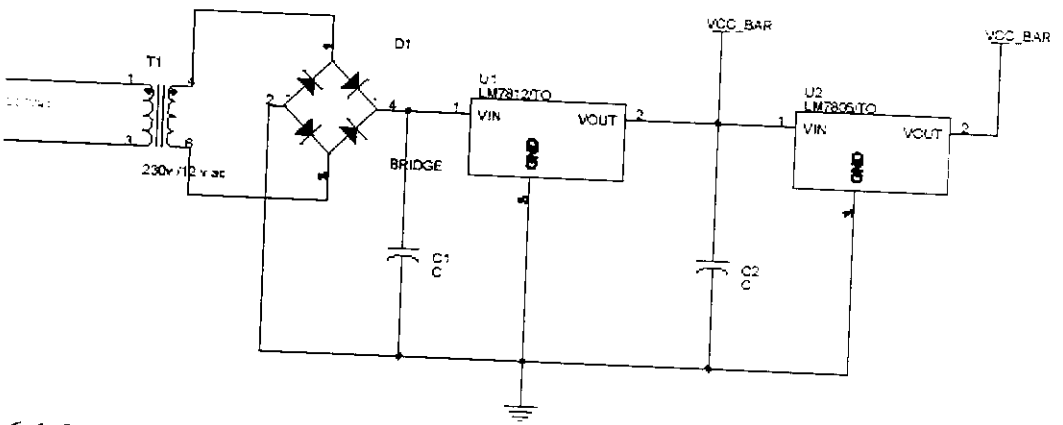
Filter

The filter circuit used here is the capacitor filter circuit where a capacitor is connected at the rectifier output, and a DC is obtained across it. The filtered waveform is essentially a DC voltage with negligible ripples, which is ultimately fed to the load.

Regulator

The output voltage from the capacitor is further filtered and finally regulated. Here we use two fixed voltage regulators namely LM7812, LM7805. The IC7812 is a +12V regulator and IC7805 is +5V regulator.

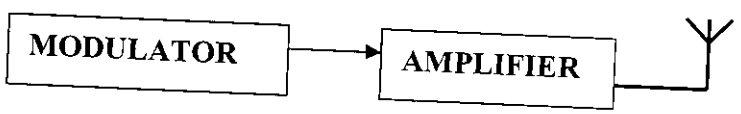
Circuit Diagram



5.6.1.2 Transmitter

*FOR VITALINFOSYSTEMS

Block Diagram



The transmitter circuit mainly contains two parts namely Modulator and Amplifier.

Modulator

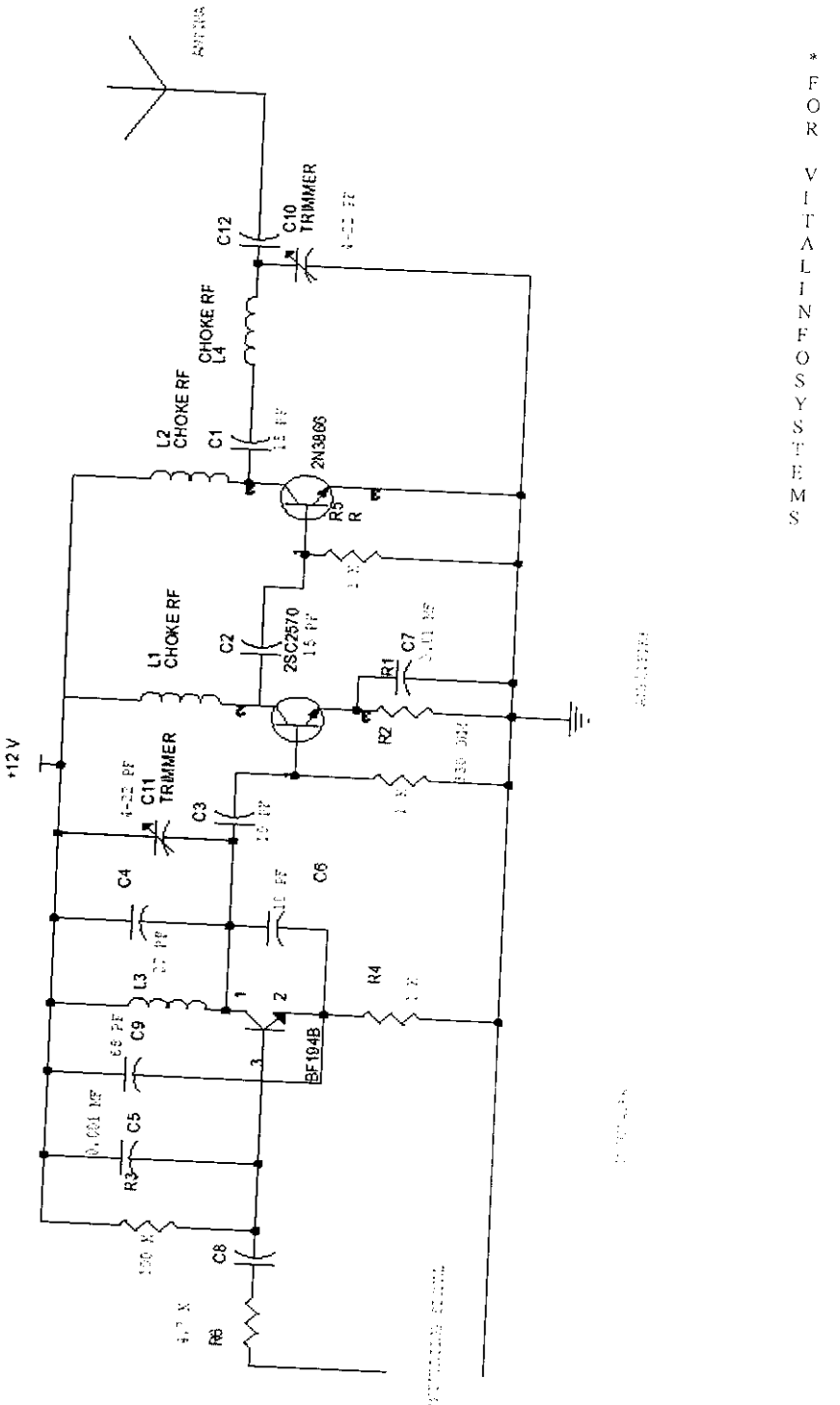
The signal from the data processing unit is in analog form, this is a wave of lower frequency hence it cannot travel a long distance. To rectify this problem we go for modulation which is nothing but adding a higher frequency carrier wave.

Amplifier

The signal from modulation block is a high frequency wave but still it might not reach the destination without loss, hence we have to increase the wave strength.

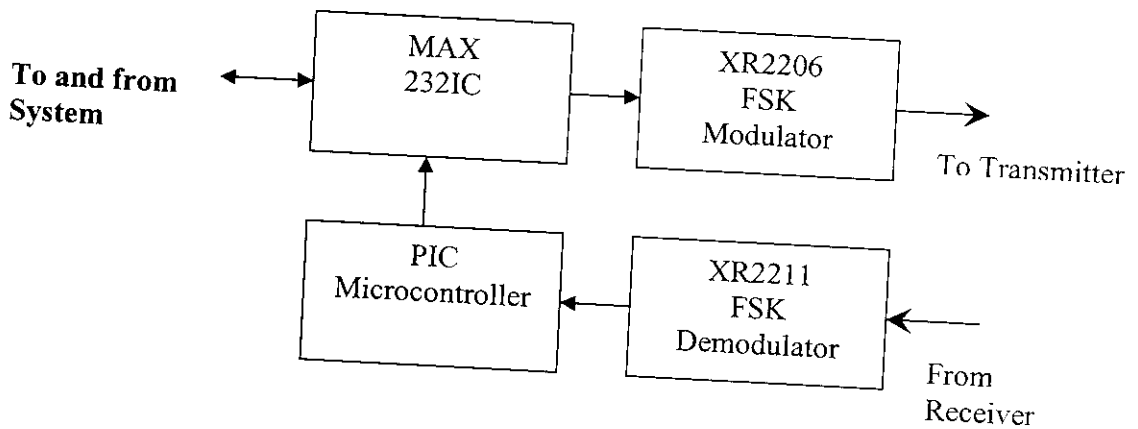
A class C amplifier is used for this purpose. By adjusting a trimmer we can adjust amplification value and transmit in the required frequency.

Circuit Diagram



5.6.1.3 Transceiver (Data Processing Unit)

Block diagram

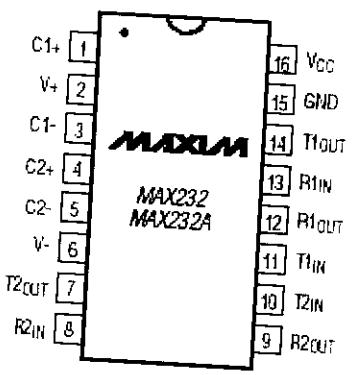


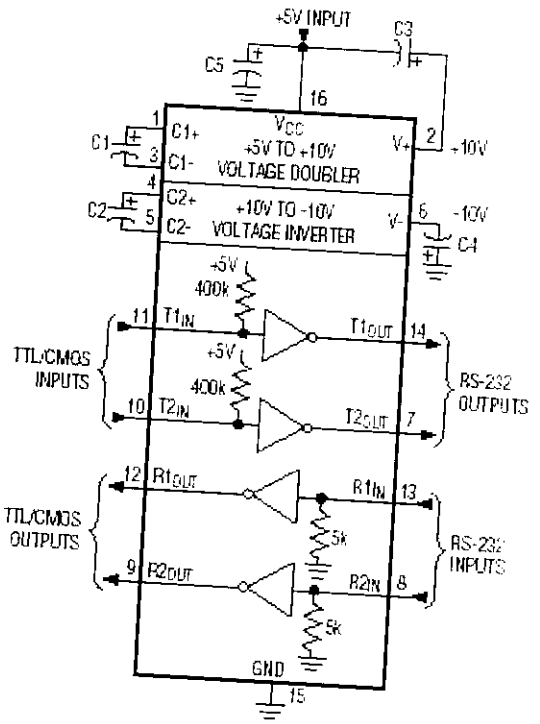
Data stream from the serial port is taken out using a 9 pin DIN connector. The MAX 232IC is used to convert data from the system which is in RS232 format this must be converted into TTL format. XR 2206 IC is used for FSK modulation purpose before the signal can be transmitted. XR2211IC demodulates the received signal and gives to a PIC microcontroller. The microcontroller gives the data which is in TTL format is given to MAX232IC to be converted into serial port RS232 format.

MAX232IC

This is a 16 pin IC. The serial port sends out data in RS232 format this format cannot be manipulated in an electronic circuit, it must be converted into TTL format. MAX232IC does just this. It also changes the voltage level to suit the circuit operations.

PIN Diagram and Internal Operation





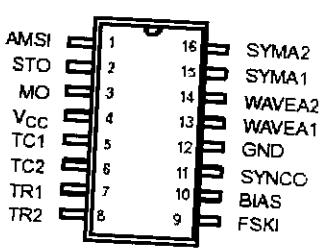
MAX232IC is a 16pin IC. 1.0 (micro) F capacitor is used in C1 (pin 1&3) for voltage doubling purpose (5V to 10V). C2 is connected to pin 4&5 used for voltage inverting purpose. Pin 2&6 gives the doubled voltage i.e., +10 V& -10 V. C3 & C5 capacitors are used for removing distortions in input and output voltages. Pins 10,11 are TTL inputs while pins 9,12 TTL outputs. Pins 7,14 are used as RS232 outputs and pins 13,8 are used for RS232 inputs. Pins 16,15 are used as 5V inputs.

XR2206 FSK Modulator

The XR-2206 is a monolithic function generator integrated circuit capable of producing high quality sine, square, triangle, ramp, and pulse waveforms of high-stability and accuracy. The output waveforms can be both amplitude and frequency modulated by an external voltage. Frequency of operation can be selected externally over a range of 0.01Hz to more than 1MHz.

The circuit is ideally suited for communications, instrumentation, and function generator applications requiring sinusoidal tone, FM, or FSK generation. It has a typical drift specification of 20ppm/°C. The oscillator frequency can be linearly swept over a 2000:1 frequency range with an external control voltage, while maintaining low distortion.

Pin Diagram



PIN DESCRIPTION

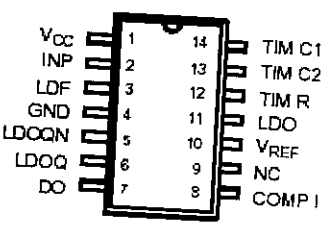
Pin #	Symbol	Type	Description
1	AMSI	I	Amplitude Modulating Signal Input.
2	STO	O	Sine or Triangle Wave Output.
3	MO	O	Multiplier Output.
4	V_{CC}		Positive Power Supply.
5	TC1	I	Timing Capacitor Input.
6	TC2	I	Timing Capacitor Input.
7	TR1	O	Timing Resistor 1 Output.
8	TR2	O	Timing Resistor 2 Output.
9	FSKI	I	Frequency Shift Keying Input.
10	BIAS	O	Internal Voltage Reference.
11	SYNCO	O	Sync Output. This output is a open collector and needs a pull up resistor to V_{CC}.
12	GND		Ground pin.
13	WAVEA1	I	Wave Form Adjust Input 1.
14	WAVEA2	I	Wave Form Adjust Input 2.
15	SYMA1	I	Wave Symetry Adjust 1.
16	SYMA2	I	Wave Symetry Adjust 2.

XR2211 FSK Demodulator

The XR-2211 is a monolithic system especially designed for data communications applications. It is particularly suited for FSK applications. It operates over a wide supply voltage range of 4.5 to 20V and a wide frequency range of 0.01Hz to 300 kHz. It can accommodate analog signals between 10mV and 3V, and can interface with conventional DTL, TTL, and ECL logic families.

The circuit consists of a basic PLL for tracking an input signal within the pass band, a quadrate phase detector which provides carrier detection, and an FSK voltage comparator which provides FSK demodulation. External components are used to independently set center frequency, bandwidth, and output delay. An internal voltage reference proportional to the power supply is provided at an output pin. The XR-2211 is available in 14 pin packages specified for military and industrial temperature ranges.

Pin Diagram



PIN DESCRIPTION

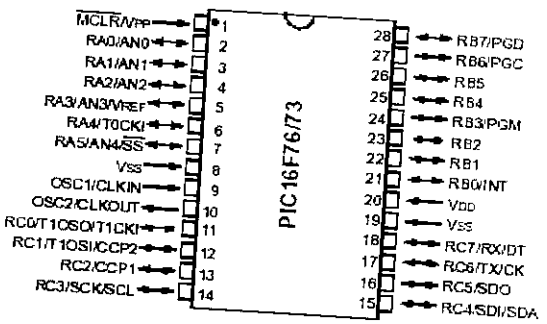
Pin #	Symbol	Type	Description
1	V_{CC}		**Positive Power Supply.**
2	INP	I	**Receive Analog Input.**
3	LDF	O	**Lock Detect Filter.**
4	GND		**Ground Pin.**
5	LDOQN	O	**Lock Detect Output Not.** This output will be low if the VCO is in the capture range.
6	LDOQ	O	**Lock Detect Output.** This output will be high if the VCO is in the capture range.
7	DO	O	**Data Output.** Decoded FSK output.
8	COMP I	I	**FSK Comparator Input.**
9	NC		**Not Connected.**
10	V_{REF}	O	**Internal Voltage Reference.** The value of V_{REF} is $V_{CC}/2$ - 650mV.
11	LDO	O	**Loop Detect Output.** This output provides the result of the quadrature phase detection.
12	TIM R	I	**Timing Resistor Input.** This pin connects to the timing resistor of the VCO.
13	TIM C2	I	**Timing Capacitor Input.** The timing capacitor connects between this pin and pin 14.
14	TIM C1	I	**Timing Capacitor Input.** The timing capacitor connects between this pin and pin 13.

PIC microcontroller (PIC16F73)

IC16F73 is a 28 pin 8 bit microcontroller. It has 3 I/O ports namely PORT A, PORT B, PORT C we have used PORT C which has default transmitter and receiver pins. Following features are present in this controller.

Power-on reset, Power-up timer and oscillator startup timer, Programmable code protection, Power saving sleep mode.

Pin diagram



Power supply is given through pins 8, 20. Pins 9, 10 are used for giving clock pulse input. First pin is used as a Master clear. We have used PORT C for our operations especially pins 17 & 18.

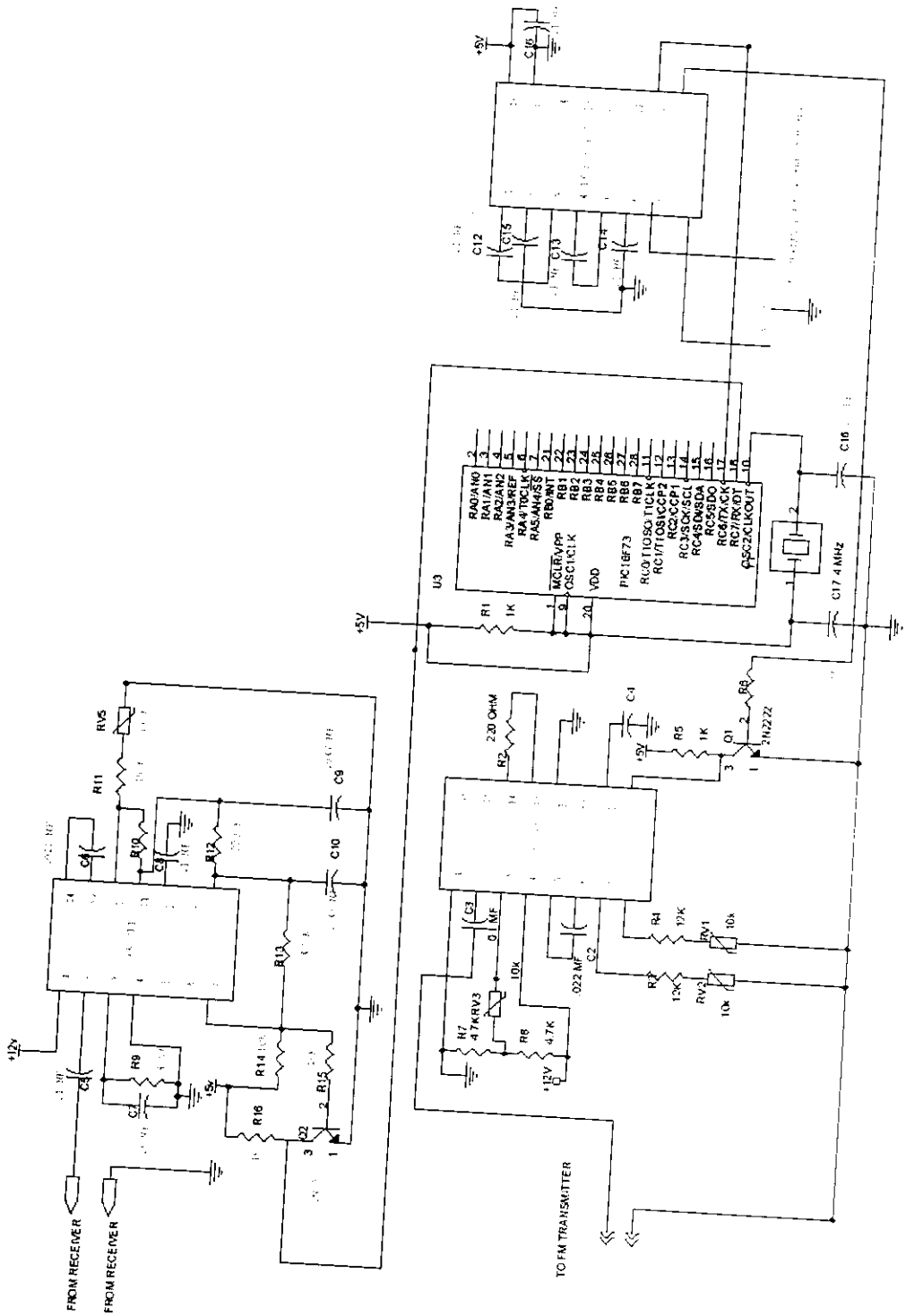
PORTC is an 8 bit wide, bi-directional port. The corresponding data direction register is TRISC. Setting a TRISC bit (='1') will make the corresponding PORTC pin an input. Clearing a TRISC bit (=0) will make the corresponding PORTC pin an output. If bit 6 of the TRISC register is set then it acts as an I/O port pin or USART Asynchronous Transmit or synchronous clock. If bit 7 is set then it acts as I/O port pin or USART Asynchronous Receive or synchronous data.

USART

-Universal Synchronous Asynchronous Receiver Transmitter

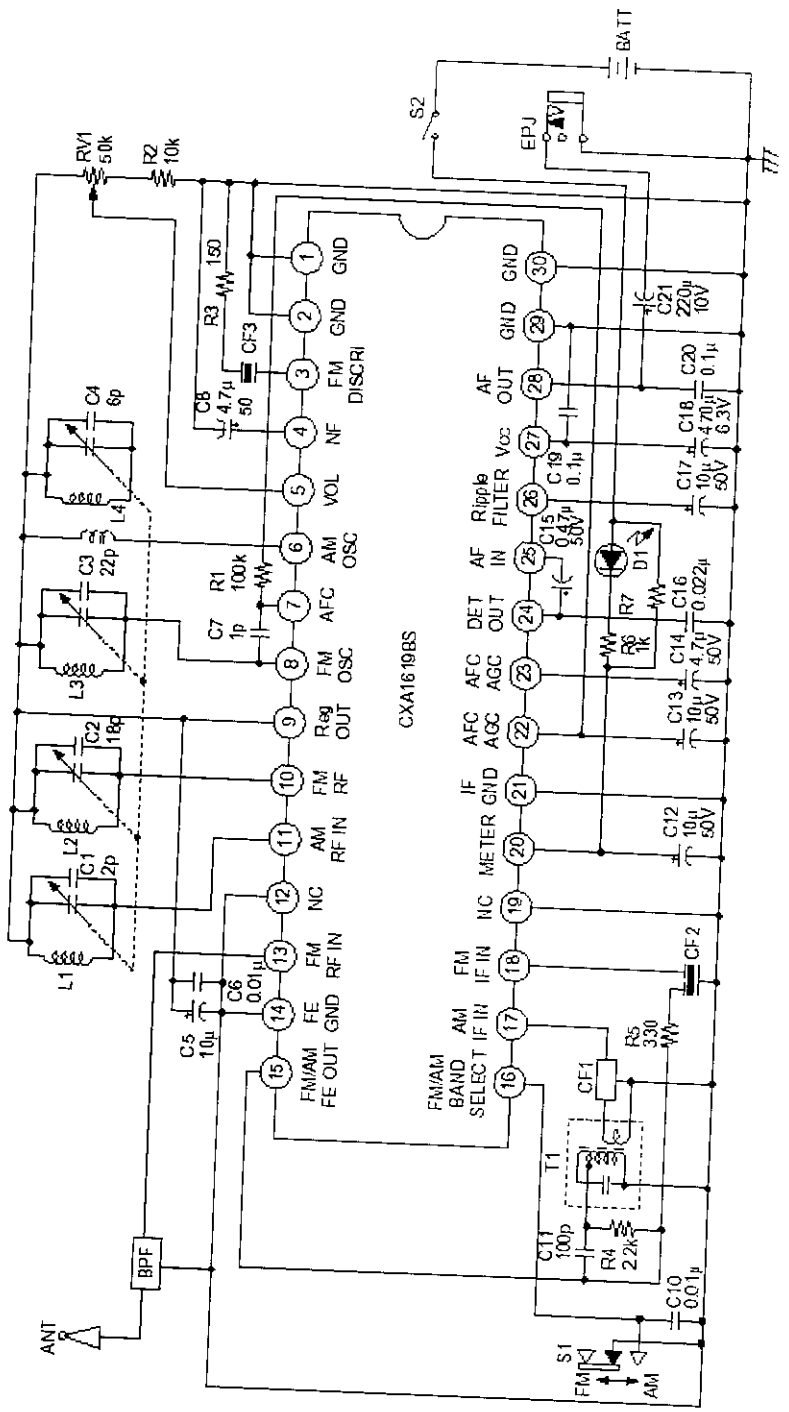
This section is one of the two serial I/O sections. The USART can be configured as a full duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers, or it can be configured as a half duplex synchronous system that can communicate with peripheral devices such as A/D or D/A .

Data Processing Unit circuit diagram



* FOR VITAL INFOSYSTEMS

5.6.1.3 Receiver



We have used a standard receiver which comes inbuilt with the CXA1619BS chip of SONY makers, it eliminates the carrier signal from the input wave. When this receiver is tuned to transmitter frequency data is received.

5.6.2 Software Design

The core coding of our project was developed in Bash shell environment on the Red Hat Linux 9.0 platform. The core consists of basic serial port transmission and reception code. Further splitting of data stream into packets, adding of source address and destination address to this packet, verification of this address capabilities were added to this core. This code was tested for files of all size and every time the original file was successfully transmitted and received on both sides. The core was basically designed for multipoint transmission.

For designing the Graphical User Interface of the core we used a programming language that was formerly not used frequently i.e., GTK (GIMP Tool Kit). This language allows the user to create a user friendly and powerful interfaces. The library file required for GTK are available in the Internet as free downloads.

The designing of the GUI was done using Glade Interface Designer, a package that comes inbuilt with Red Hat Linux. Glade allows the user to design two types of applications namely GNOME and GTK. For our project we have used GTK as the base. The library of Glade contains loads of controls like buttons, check buttons, list box, combo box, image box ... etc.

In GTK these controls are called Widgets. Each widget must be placed in its container widget or super widget. For instance, the window is the super widget inside that vertical and horizontal boxes are placed, these are sub widgets of the window. Inside each box one control can be placed, such an option has been given because resizing of the window is made easier without any separate code for it.

5.7. MODULE DESCRIPTION

The project "Wireless Intranet" consists of three modules. The three modules are:

- Hardware module
- Network application module
- User interface module

5.7.1. Hardware module

Data from the system is transferred using 9 pin female DIN connector. Data received from the system is in RS232C format. This is converted into TTL format using MAX232 IC. The data from MAX232 is sent through the microcontroller to IC2206 for FSK modulation. The analog output from FSK is sent to the transmitter unit wherein the frequency at which the data has to be transmitted is tuned. Receiver is tuned to the transmitter frequency and data is received. Received analog signal is given to IC2211 for FSK demodulation. Then this digital signal is sent to the MAX232 IC through microcontroller. From MAX232 IC data is sent to the system through 9pin connector.

5.7.2. Network Application module

This module is concerned with the transmission and reception of data on a software basis. Firstly, the name of the file to be transmitted is stored in a temporary file. The file to be transmitted is split into smaller packets of size 1K then an address header consisting of source address and destination address is added to this packet. These packets are stored in the same temporary file. This file

is sent to the serial port for transmission. The source address and destination address are obtained from previously defined tables.

When data is received through the serial port the address header is extracted and destination address is verified with this system's address. If they match, the packet is accepted else the packet is discarded. Once the packet is accepted the header is truncated and a temporary file is built with the data stream from the packet. This temporary file will contain the name of the original file. Using this information the original file is constructed and placed in the root directory.

When files shared in particular host are to be viewed a control packet along with the source and destination address is broadcasted. The specified host on receiving this control packet transmits the list of files that are shared in this system to the requesting host. This is called as file list packet. The requesting host then displays these file names to the user. The user can choose the file he wants to receive.

5.7.3. User Interface module

In this module we have designed an application which is very user interactive. This application allows the user to share files in the wireless network. Lookup table information and system information can be viewed by the user. If needed, the user can change the address of the system according to his needs. He can also select the port and transfer baud rate for the data to be transmitted. This application not only displays the files shared in other systems but also allows the user to request the file content. Provisions have been made so that the user can send a particular file to any one user of his choice who is present in this network.

SYSTEM TESTING

6. SYSTEM TESTING

Testing is an activity to verify that a correct system is being built and is performed with the intent of finding faults in the system. The system should be tested experimentally with test data so as to ensure that the system works according to the required specification.

6.1. LEVELS OF TESTING

The details of the software functionality tests are given below. The testing procedure that has been used is as follows

- Unit testing
- Integration testing
- Validation testing
- Output testing

6.1.1. Unit Testing

In this testing step, each module was found to be working satisfactory as per the expected output of the module. In the package development, each module was tested separately after it had been completed and checked with valid data. Unit testing exercises specific paths in the modules control structure to ensure complete coverage and maximum error detection.

Wireless card consists of four sections namely power unit, data processing unit, transmitter and receiver. The power unit gets an input of 230V and gives two constant outputs 12V and 5V, this output was checked using standard digital

voltmeter. In the data processing unit we check whether appropriate analog signals are generated based on digital input from the system. The analog signal was checked using CRO.

The transmitter board must transmit signals at a frequency designated by user. We check this using an analog audio signal. Similarly we tune the receiver to this frequency and checked whether the transmitted signal was received without loss.

6.1.2. Integration Testing

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the system has been integrated, a set of high-order tests were conducted. The main objective in this testing process was to take unit-tested modules and build a program structure that has been dictated by design. Here we have followed Top down integration method.

6.1.2. a. Software Integration

This method was an incremental approach to the construction of program structure. Modules were integrated by moving downward through the control hierarchy, beginning with the main programming module. The module subordinates to the main program module were incorporated to the main control structure.

6.1.2 b. Hardware Integration

All four units (power unit, data processing unit, transmitter and receiver) were integrated on to a single board. After power unit was connected and hot tracing was done to verify the flow of supply through the circuit. The analog output of data processing unit was given to the transmitter and the output of the transmitter was checked whether the same input analog signal was retrieved. The receiver circuit was tuned to the transmitter frequency and was tested whether it receives the transmitted signal correctly.

6.1.3. Validation Testing

After the hardware and software have been integrated and tested separately the whole system was assembled and interfacing errors have been uncovered and correction tests were made.

6.1.4. Output Testing

Output testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. The input system, output documents were checked and required modifications have been made to suit the specifications. Then using the test data prepared, the whole system was tested and found to be a successful one.

SYSTEM IMPLEMENTATION

7.1. IMPLEMENTATION

Implementation is the process of converting a new or revised system design into an operational one. The first task was implementation planning, wherein the methods were decided and the time scale was adopted. The implementation involves installing the software, connecting the hardware to the system and configuring the system in the wireless network.

Some of the aspects of the system implementation are

7.1.1 Installation

7.1.2 Configuration

7.1.3 Device Tuning

7.1.1. Installation

This was done by installing our software WSDCP in all the systems in the network. The hardware device was also attached to all the systems in the network through the 9 pin connector to the serial port. Antennas are projected to make the system ready for sending and receiving signals.

7.1.2. Configuration

Configuration involves registering the assigned system name and SNA (Serial Network Address) in each system in the wireless network.

7.1.3 Device Tuning

All transmitters and receivers have to be tuned to the same frequency.

CONCLUSION

8.1. CONCLUSION

The completed design and development of the system of "Wireless Intranet" is presented in this dissertation. The system has user friendly features. It is possible for any user to use this system.

The system has been tested by connecting with many systems and provides satisfactory performance. The hardware has been designed to transmit data without loss and the software provides a user interactive environment supporting the hardware.

The system is developed with the specifications given by the company. Since the requirements of any organization and their standards are changing day to day, the system has been designed in such a way that its scope and boundaries can be expanded in future with little modifications.

Instead of being restricted by wires, new wireless technology as freed users to remain connected to the network. We hope that the project serves the purpose of simplicity and reliability.

We have achieved our vision,

"Wireless at Wired Cost".

8.2. FUTURE ENHANCEMENTS

Currently our project has built a base in unchartered territory. With the technology that will be available in the future our project will successfully replace wired networks. The programming techniques used in the design of the system provides scope for further enhancements and implementation of any change, which may occur in the future.

Further enhancements that can be provided are

- Our device will be connected to the Ethernet card.
- Frequencies up to 5GHz can be used to establish a network spanning over one kilometer.
- Security can be enforced by cryptographic techniques.

In near future our project will be running on every computer in this entire world.

APPENDIX

9.1 CODING

```
#ifdef HAVE_CONFIG_H
#include <config.h>
#endif

# include "totalhead.h"
# include "widhead.h"
# include "globals.h"
# include "funchead.h"


struct termios oldio,newio;

struct tem{
        int sip,dip;
        char data[PACKSIZ];
        }pack;


gboolean on_button1_button_press_event      (GtkWidget      *widget,
                        GdkEventButton  *event,
                        gpointer        user_data)
{

 //this is the function handler for the buttton system1

 i=1;j=0;
 system1=create_system1();
 gtk_widget_show(system1);
 bflag=1;                       //button number flag
 lpoin=fopen("/root/filelist","r");
 fseek(lpoin,6L,SEEK_SET);
 while((ch=fgetc(lpoin))!=EOF)
 {
   if(ch!='\n')
   {
    flist[i][j]=ch;
    j++;
   }
   else
   {
    i++;j=0;
```

```
    }
   }
   k=0;cnt=0;
   for(j=1;j<=i;j++)
   {
    ret=rindex(flist[j],'/');
    butn[cnt] = gtk_button_new_with_label(ret);
    gtk_widget_show (butn[cnt]);

 gtk_fixed_put((GTK_FIXED(lookup_widget(GTK_WIDGET(system1),"fixed2"))),
 butn[cnt], 108, 56+k);

 gtk_signal_connect(GTK_OBJECT(butn[cnt]),"clicked",GTK_SIGNAL_FUNC(bu
 tton_event),NULL);
    cnt++;k+=40;
  }
  return FALSE;
}

gboolean on_button4_button_press_event       (GtkWidget      *widget,
                        GdkEventButton  *event,
                        gpointer        user_data)
{

  //this is the function handler for the buttton add

  filesel=create_fileselection1();
  gtk_widget_show(filesel);
  return FALSE;
}

void on_ok_button1_clicked (GtkButton *button, gpointer user_data)
{

 //this is the function handler for the buttton ok of file selection

 FILE *flst;
 buffer=gtk_file_selection_get_filename(filesel);
 flst=fopen("/root/filelist","r+");
 g_print(buffer);
 ret=rindex(buffer,'/');
 ret=ret+1;
 butn[cnt] = gtk_button_new_with_label(ret);
 gtk_widget_show (butn[cnt]);
```

```
    gtk_fixed_put((GTK_FIXED(lookup_widget(GTK_WIDGET(system1),"fixed2"))),
    butn[cnt], 108, 56+k);

    gtk_signal_connect(GTK_OBJECT(butn[cnt]),"clicked",GTK_SIGNAL_FUNC(bu
    tton_event),NULL);
      fseek(flst,-1L,SEEK_END);
      fprintf(flst,"%s",buffer);
      fclose(flst);
      flst=fopen("filelist","a+");
      fseek(flst,0L,SEEK_END);
      fprintf(flst,"\n#");
      fclose(flst);
      cnt++;
      k+=40;
      gtk_widget_destroy(filesel);
  }

  void button_event (GtkWidget *wid,gpointer *dat)
  {

  //handler for the file button added

   gbut=wid;
   cho=create_choice1();
   gtk_widget_show(cho);
 }

 void on_cancel_button1_pressed (GtkButton *button, gpointer  user_data)
 {
   //cancel button in fileselection box

   gtk_widget_destroy(filesel);
 }

 void on_button9_pressed (GtkButton  *button, gpointer  user_data)
 {
    char *sline;
    //handler for local system sysinfo
    fp=fopen("/etc/systable.conf","r");
    fscanf(fp,"%s",a);
    fscanf(fp,"%s",sline);
    fscanf(fp,"%s",b);
    sys=create_sysinfo();
    gtk_widget_show(sys);
    lab8=lookup_widget(GTK_WIDGET(sys),"label8");
```

```
    lab9=lookup_widget(GTK_WIDGET(sys),"label9");
    gtk_label_set(GTK_LABEL(lab8),b);
    gtk_label_set(GTK_LABEL(lab9),a);
    fclose(fp);
  }


  void on_button10_pressed (GtkButton *button, gpointer user_data)
  {
    hcon=create_hconfig();
    gtk_widget_show(hcon);
    ent3=lookup_widget(GTK_WIDGET(hcon),"entry2");
    memset(&hostname,0,sizeof(hostname));
    hostname=(char *)malloc(30);
    gethostname(hostname,30);
    gtk_entry_set_text(GTK_ENTRY(ent3),hostname);
  }

  void on_button7_pressed (GtkButton *button, gpointer user_data)
  {
   send=create_send();
   gtk_widget_show(send);
  }

  void on_button11_pressed (GtkButton *button, gpointer user_data)
  {
    filesel2=create_fileselection2();
    gtk_widget_show(filesel2);
  }

 void on_button19_pressed (GtkButton *button, gpointer user_data)
 {
   gtk_widget_destroy(looktab);
 }

 void on_button14_pressed (GtkButton *button, gpointer user_data)
 {
  FILE *lfp;
  int z=0,lp=0,x=0;
  char bdbuff[100][100],cha;
  char *hbu,*lad,*hbu1,*lad1;
  looktab=create_ltable();
  gtk_widget_show(looktab);
  lab46=lookup_widget(GTK_WIDGET(looktab),"label46");
  lab47=lookup_widget(GTK_WIDGET(looktab),"label47");
  lab48=lookup_widget(GTK_WIDGET(looktab),"label48");
```

```
    lab49=lookup_widget(GTK_WIDGET(looktab),"label49");
    lfp=fopen("/root/table","r");
    while((cha=fgetc(lfp))!=EOF)
    {
      if(cha!='\n')
       {
         bdbuff[z][x]=cha;
         x++;
       }
      else
      {
        bdbuff[z][x]='\0';
        x=0;z++;
      }
    }
    lp=z;
    fclose(lfp);
    hbu=strtok(bdbuff[0]," ");
    lad=strtok(NULL,"\0");
    hbu1=strtok(bdbuff[1]," ");
    lad1=strtok(NULL,"\0");
    gtk_label_set(GTK_LABEL(lab46),hbu);
    gtk_label_set(GTK_LABEL(lab47),lad);
    gtk_label_set(GTK_LABEL(lab48),hbu1);
    gtk_label_set(GTK_LABEL(lab49),lad1);
}

void on_button20_clicked (GtkButton *button, gpointer  user_data)
{
  gtk_widget_destroy(sys);
}


void on_button21_pressed (GtkButton *button, gpointer  user_data)
{
  gtk_widget_destroy(system1);
}

void
on_button13_pressed (GtkButton *button, gpointer        user_data)
{
  gtk_widget_destroy(send);
}

void on_cancel_button2_pressed  (GtkButton   *button, gpointer user_data)
{
```

```
  gtk_widget_destroy(filesel2);
}


void on_button24_clicked (GtkButton *button, gpointer  user_data)
{
  pconf=create_pconfig();
  gtk_widget_show(pconf);
}

void
on_button26_pressed                    (GtkButton       *button,
                        gpointer         user_data)
{
   gtk_widget_destroy(pconf);
}

void transmit()
{

  // Main transmitter function

 int chk;
 char key;
 g_print("transmitter=%s",buffer);
 fpointer=fopen(buffer,"r");
 if(fpointer==NULL)
 {
   g_print("path not found");
 }
 if((chk=chsysconfig())<0)
 {
    g_print("Error:Unable to load Sysconfig");
    hcon=create_hconfig();
    gtk_widget_show(hcon);
    ent3=lookup_widget(GTK_WIDGET(hcon),"entry2");
    memset(&hostname,0,sizeof(hostname));
    hostname=(char *)malloc(30);
    gethostname(hostname,30);
    gtk_entry_set_text(GTK_ENTRY(ent3),hostname);
 }
 srcmap();

 fsize=filesize(fpointer);
 //printf("file=%d\n",fsize);
```

```
    pks=calpacket(fsize);

    frmpack(pks,fsize);

    fclose(fpointer);

    sermit();
}

int frmpack(int pknum,int pksize)
{
  // splits given file into packets

   int opn,sop;
   char wbuff[2048];
   char *fnam;
   char *stbit="0x0acf\n";          //start bit
   char *ebit="0x7f7a\n";           //end bit]
   char *trail="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx";
   int j=1,tz=pksize;
   sop=open(buffer,O_RDONLY);
   opn=open("er.txt",O_CREAT|O_WRONLY|O_TRUNC);
   fnam=rindex(buffer,'/');
   fnam=fnam+1;
   write(opn,stbit,strlen(stbit));
   write(opn,fnam,strlen(fnam));
   if(opn<0||sop<0)
   {
     printf("file opening failed buffer=%d er=%d",sop,opn);
   }
     if(pksize<PACKSIZ)
       {
       read(sop,pack.data,pksize);
       sprintf(wbuff," %d %d %s",pack.sip,pack.dip,pack.data);
       write(opn,wbuff,pksize+9);
       }else
        {
          while(j<=pknum)
          {
            read(sop,pack.data,PACKSIZ);
      sprintf(wbuff," %d %d %s",pack.sip,pack.dip,pack.data);
            write(opn,wbuff,PACKSIZ+9);
            tz=tz-PACKSIZ;
            if(tz<PACKSIZ)
            {
              read(sop,pack.data,tz);
```

```
                sprintf(wbuff," %d %d %s",pack.sip,pack.dip,pack.data);
                        write(opn,wbuff,(tz+9));
                        return 0;
                    }
                    j++;
                }
            }
            write(opn,ebit,strlen(ebit));
            write(opn,trail,strlen(trail));
        close(sop);
        close(opn);
        return 0;
    }

    int filesize(FILE *sp)
    {

        // calculates size of file

       curpos=ftell(sp);
       fseek(sp, 0L, SEEK_END);
       length = ftell(sp);
       fseek(sp,curpos,SEEK_SET);
       return length;
   }

   int calpacket(int size)
   {
     // calculates number of packets the file will form

     int temp;
     packnum=(float)size/PACKSIZ;
     temp=(int)packnum;
     var=packnum-temp;
     num=temp;
     if(var>0)
     {
       num++;
     }
    return num;
}

void sermit()
{

   //actual port communication function
```

```
int port,result,ret,fil;
char *b,byte[200];
memset(&b,0,sizeof(b));
port=open(pot,O_RDWR|O_NOCTTY|O_ASYNC);
fp1=fopen("er.txt","r");
if(fp1==NULL)
{
  g_print("er failed");
  exit(1);
}
len=filesize(fp1);
printf("\nlen=%d",len);
fclose(fp1);
b=(char *)malloc(len);
fil=open("er.txt",O_RDONLY);
if(port<0)
{
  g_print(" Port opening error");
}
//g_print("first");
tcgetattr(port,&oldio);
memset(&newio,0,sizeof(newio));
newio.c_cflag=CS8|CLOCAL|CREAD;
newio.c_iflag=IGNPAR|ICRNL;
newio.c_oflag=ONLCR;
newio.c_lflag=ICANON;
cfsetospeed(&newio,BAUD);
cfsetospeed(&newio,BAUD);
newio.c_cc[VINTR]   = 0;
newio.c_cc[VQUIT]   = 0;
newio.c_cc[VERASE]  = 0;
newio.c_cc[VKILL]   = 0;
newio.c_cc[VEOF]    = 4;
newio.c_cc[VTIME]   = 0;
newio.c_cc[VMIN]    = 1;
newio.c_cc[VSWTC]   = 0;
newio.c_cc[VSTART]  = 0;
newio.c_cc[VSTOP]   = 0;
newio.c_cc[VSUSP]   = 0;
newio.c_cc[VEOL]    = 0;
newio.c_cc[VREPRINT] = 0;
newio.c_cc[VDISCARD] = 0;
newio.c_cc[VWERASE] = 0;
newio.c_cc[VLNEXT]  = 0;
newio.c_cc[VEOL2]   = 0;
```

```
      tcflush(port,TCIOFLUSH);
      tcsetattr(port,TCSANOW,&newio);
      //fcntl(port,F_SETFL,FNDELAY);
       ret=read(fil,b,len);
       result=write(port,b,ret);
       printf("ret=%d",ret);
       printf("buffer=%s",b);
       if(result<0)
       {
         g_print("write error");
       }
       else{
            g_print("to remove");
            remove("er.txt");
         }
       tcsetattr(port,TCSANOW,&oldio);
       close(port);
       rep=create_report();
       gtk_widget_show(rep);

  gtk_label_set(GTK_LABEL(lookup_widget(GTK_WIDGET(rep),"label22")),pot);

  gtk_label_set(GTK_LABEL(lookup_widget(GTK_WIDGET(rep),"label23")),"1200
  ");
     if(pack.dip==0)
     {

  gtk_label_set(GTK_LABEL(lookup_widget(GTK_WIDGET(rep),"label24")),"all
  hosts");
     }
     else
     {

  gtk_label_set(GTK_LABEL(lookup_widget(GTK_WIDGET(rep),"label24")),host);
     }
     sprintf(byte,"%d",result);

  gtk_label_set(GTK_LABEL(lookup_widget(GTK_WIDGET(rep),"label26")),byte);

}

int chsysconfig()
{

    //checks if system is configured for WSDCP operation
```

```
    int sopn;
    if((sopn=open("/etc/systable.conf",O_WRONLY))<0)
     {
       return -1;
     }
     else
     {
       return 0;
     }
}


void srcmap()
{
  // gives the address of the source system

   int sopn,ret;
   char sbuf[5];
   sopn=open("/etc/systable.conf",O_RDONLY);
   if(sopn<0)
   {
      g_print("open error");
      exit(1);
   }
   else
   {
      ret=read(sopn,sbuf,4);
      sbuf[ret]='\0';
      pack.sip=atoi(sbuf);
   }
}


void on_button27_pressed                (GtkButton     *button,
                        gpointer        user_data)
{
  gtk_widget_destroy(rep);
}


void on_ok_button2_pressed              (GtkButton     *button,
                        gpointer        user_data)
{

  // file selection dialog box 2
```

```
    entry=lookup_widget(GTK_WIDGET(send),"entry1");
    buffer=gtk_file_selection_get_filename(filesel2);
    gtk_entry_set_text(GTK_ENTRY(entry),buffer);
    gtk_widget_destroy(filesel2);
}

void on_button28_clicked               (GtkButton     *button,
                        gpointer       user_data)
{
  ent2=lookup_widget(GTK_WIDGET(button),"entry2");
  gtk_entry_set_text(GTK_ENTRY(ent2),"hai");
}

int imap(char *hn)
{

  // maps system name to address

    int fdesc,i=0,j=0,k,l;
    FILE *p;
    char nu,buf[512][512],tems[512],snam[512][512],ip[4];
    p=fopen("/root/table","r");
    l=strlen(hn)+1;
    if(p==NULL)
    {
       perror("Table opening error");
       exit(1);
    }
    while((nu=fgetc(p))!=EOF)
    {
        if(nu!='\n')
        {
           if(nu!=' ')
           {
             snam[j][i]=nu;
           }
           tems[i]=nu;
           i++;
        }
        else
        {
           tems[i]='\0';
           strcpy(buf[j],tems);
           snam[j][i]='\0';
           j++;
```

```
            i=0;
          }
        }
        i=0;
        for(k=0;k<j;k++)
        {
          printf("cmp=%d,hn=%s,snam=%s",strcmp(hn,snam[k]),hn,snam[k]);
          if((strcmp(hn,snam[k]))==0)
          {
            while(i<3)
            {
             ip[i]=buf[k][i+l];
             i++;
            }
            ip[i]='\0';

          }
        }
        ch_ip=ip;
        k=atoi(ip);
        g_print("inet add=%d",k);
        return k;
}


void on_button12_clicked (GtkButton *button, gpointer user_data)
{

  // send dialog box

  combo=lookup_widget(GTK_WIDGET(send),"combo_entry1");
  host=gtk_entry_get_text(GTK_ENTRY(combo));
  pack.dip=imap(host);
  transmit();
}

void on_button29_clicked (GtkButton *button, gpointer user_data)
{
 //main screen

 win1=lookup_widget(GTK_WIDGET(button),"maind");
 gtk_widget_destroy(win1);
 wscp=create_wscp();
 srcmap();          //source ip map
 gsrc=pack.sip;
 g_print("sourse address=%d",gsrc);
```

```
  switch(gsrc)
  {
    case 100: button2=200;
              button3=300;
              break;
    case 200: button2=100;
              button3=300;
              break;
    case 300: button2=100;
              button3=200;
              break;
  }
  thd=pthread_create(&f_thread,NULL,&receiver,NULL);
  gtk_widget_show(wscp);
}

void
on_button31_clicked                (GtkButton     *button,
                        gpointer       user_data)
{
  gtk_widget_destroy(gbut);
  gtk_widget_destroy(cho);
}

void
on_button32_clicked                (GtkButton     *button,
                        gpointer       user_data)
{

  //host configuration dialog box

  int foen,uid,saip;
  char sysbuff[120],*login;
  char *hsid;
  ent4=lookup_widget(GTK_WIDGET(hcon),"entry3");
  hsid=gtk_entry_get_text(GTK_ENTRY(ent4));
  saip=atoi(hsid);
  if((uid=getuid())==0)
  {
    if(saip<=99||saip>999)
    {
        perror("\n Invalid IP range");
        exit(0);
    }
    login=getlogin();
    sprintf(sysbuff,"%d   %s  %s",saip,login,hostname);
```

```
      foen=open("/etc/systable.conf",O_CREAT|O_WRONLY);
      write(foen,sysbuff,strlen(sysbuff));
      close(foen);
      chmod("/etc/systable.conf",222);
    }
    gtk_widget_destroy(hcon);
}

void receiver()
{
 //receiver thread runs at all times

 int  fil,i=0,fpoin,srch=0;
 char ch[1024],buff1[1024];
 FILE *fp;
 rport=open(pot,O_RDWR|O_NOCTTY|O_SYNC);
 g_print("receiver is running");
 cnt=0;k=0;
 if(rport<0)
 {
  perror(" Port opening error");
  exit(1);
 }
 tcgetattr(rport,&oldio);
 memset(&newio,0,sizeof(newio));
 memset(&recbuff,0,sizeof(recbuff));
 newio.c_cflag=CS8|CLOCAL|CREAD;
 newio.c_iflag=IGNPAR|ICRNL;
 newio.c_oflag=ONLCR;
 newio.c_lflag=ICANON;
 cfsetospeed(&newio,BAUD);
 cfsetospeed(&newio,BAUD);
 newio.c_cc[VINTR]    = 0;
 newio.c_cc[VQUIT]    = 0;
 newio.c_cc[VERASE]   = 0;
 newio.c_cc[VKILL]    = 0;
 newio.c_cc[VEOF]     = 4;
 newio.c_cc[VTIME]    = 0;
 newio.c_cc[VMIN]     = 1;
 newio.c_cc[VSWTC]    = 0;
 newio.c_cc[VSTART]   = 0;
 newio.c_cc[VSTOP]    = 0;
 newio.c_cc[VSUSP]    = 0;
 newio.c_cc[VEOL]     = 0;
 newio.c_cc[VREPRINT] = 0;
 newio.c_cc[VDISCARD] = 0;
```

```
    newio.c_cc[VWERASE] = 0;
    newio.c_cc[VLNEXT]  = 0;
    newio.c_cc[VEOL2]   = 0;
    tcflush(rport,TCIOFLUSH);
    tcsetattr(rport,TCSANOW,&newio);
    //fcntl(rport,F_SETFL,FNDELAY);
    //fil=open("dats.txt",O_WRONLY|O_CREAT|O_TRUNC);
    srcmap();
    while(1)
    {
         rest=read(rport,recbuff,sizeof(recbuff));  //reading buffer
         g_print("first received=%s",recbuff);
         ctrlchkft(); //calling control checker
         fcheck();
         filgetter();
      if((strstr(recbuff,"0x3bb"))!=NULL)
      {
         i=gtflist();
         while(i!=1)
         {
             rest=read(rport,recbuff,sizeof(recbuff));  //reading buffer
             i=gtflist();    //function to store filelist in new file
         }
         if(i==1)
         {
           dispbutton();
           nullify();
           remove("sflist");
         }
      }
      nullify();
   }
   //close(fil);
   tcsetattr(rport,TCSANOW,&oldio);
   close(rport);
}

void on_button2_clicked (GtkButton *button,gpointer  user_data)
{
   bflag=2;
   sys2=create_system();
   gtk_widget_show(sys2);
   gw=sys2;
}

void on_button23_clicked (GtkButton *button, gpointer user_data)
```

```
{
   k=0;
   gtk_widget_destroy(gw);
}


void trans()
{
  char fbuf[1024];
  int ad=0;
        tpo=fopen("/root/test","r");
        cflag=1;
        if(tpo==NULL)
        {
          g_print("error in tpo");
        }
        else
        {
          while((tch=fgetc(tpo))!=EOF)
          {
            tbuffer[ti]=tch;
            ti++;
          }
          tbuffer[ti]='\0';
          switch(bflag)
          {
            case 2:sprintf(fbuf,"%d %d %s",gsrc,button2,tbuffer);
                   break;
            case 3:sprintf(fbuf,"%d %d %s",gsrc,button3,tbuffer);
                   break;
          }
          tes=write(rport,fbuf,strlen(fbuf));
          //g_print("gsrc=%d,buffer=%s",gsrc,fbuf);
          if(tes<0)
          {
            g_print("error");
          }
          else
          {
            g_print("buffer=%s",fbuf);
            ti=0;
          }
        }
}
```

```
void on_button38_clicked  (GtkButton *button, gpointer user_data)
{
   t_thd=pthread_create(&t_thread,NULL,&rtrans,NULL);
}

void on_button3_clicked (GtkButton*button, gpointer  user_data)
{
  bflag=3;                //button flag
  s3=create_system();      //system3 dialog box
  gtk_widget_show(s3);
  gw=s3;
}


void on_button39_clicked (GtkButton *button, gpointer user_data)
{
 char a[]="system3";
 int b=0;
 b=imap(a);
 sys=create_sysinfo();     //system information of system3
 gtk_widget_show(sys);
 lab8=lookup_widget(GTK_WIDGET(sys),"label8");
 lab9=lookup_widget(GTK_WIDGET(sys),"label9");
 gtk_label_set(GTK_LABEL(lab8),a);
 gtk_label_set(GTK_LABEL(lab9),ch_ip);
}

void on_button41_clicked (GtkButton*button, gpointer user_data)
{
 gtk_widget_destroy(s3);
}


void on_button16_clicked (GtkButton *button,gpointer user_data)
{

 char *a;
 int b=0;
 g_print("bflag=%d",bflag);
 switch(bflag)
 {
   case 2:
        switch(gsrc)
        {
          case 100:a="system2";
                break;
```

```
            case 200:a="system1";
                  break;
            case 300:a="system1";
                  break;
          }
          break;
    case 3:
          switch(gsrc)
          {
            case 100:a="system3";
                  break;
            case 200:a="system3";
                  break;
            case 300:a="system2";
                  break;
          }
          break;
  }
  b=imap(a);
  sys=create_sysinfo();        //system information for system1
  gtk_widget_show(sys);
  lab8=lookup_widget(GTK_WIDGET(sys),"label8");
  lab9=lookup_widget(GTK_WIDGET(sys),"label9");
  gtk_label_set(GTK_LABEL(lab8),a);
  gtk_label_set(GTK_LABEL(lab9),ch_ip);
}

int ctrlchkft()              //control checking and file transmitting function.
{
   int fpoin,sad=0,dad=0;
   char ch[2000],buff1[1024];
   char *cbit,hbuf[2000];
   fpoin=open("/root/filelist",O_RDONLY);
   if((strstr(recbuff,"0x3aa"))!=NULL)  // control checking
   {
   //if(cflag==0)               //checking control flag
   {
   g_print("check started=%d",gsrc);
   strcpy(hbuf,recbuff);
   g_print("recbuff=%s",recbuff);
   sad=atoi(strtok(hbuf," "));          //extracting from buffer
   dad=atoi(strtok(NULL," "));
   cbit=strtok(NULL," ");
   //cflag=1;                 //setting control file flag
   }
   if(dad==gsrc)
```

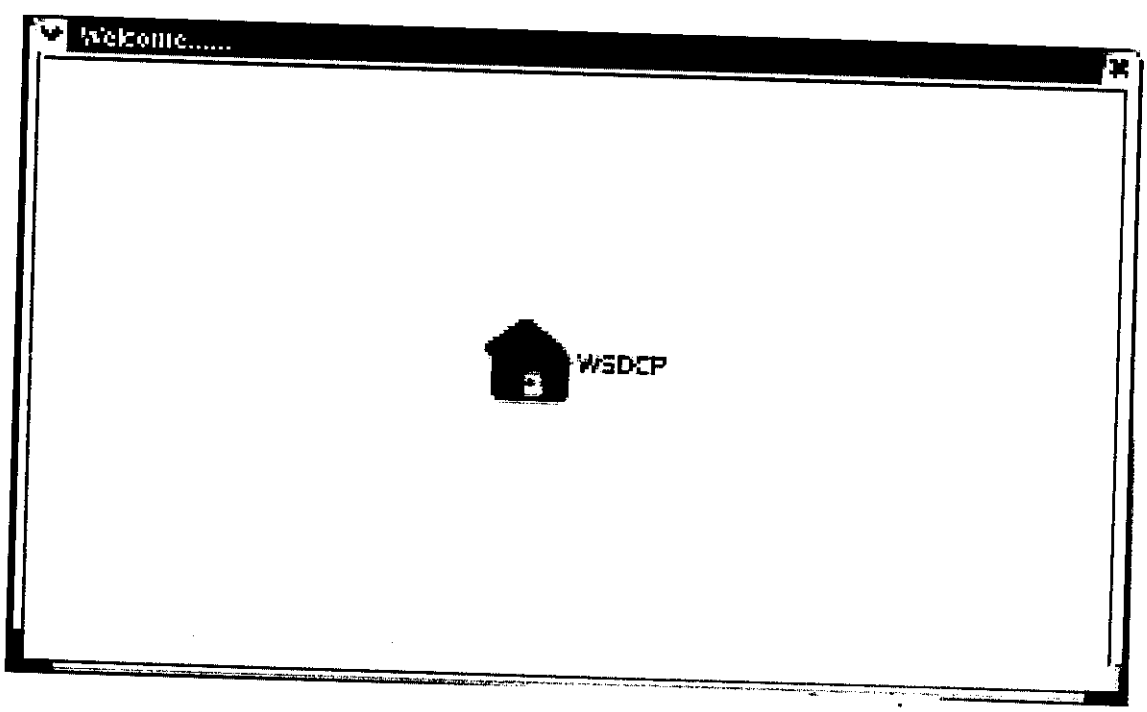
```
    {
     g_print("Control bit:%s",cbit);
     if((strstr(cbit,"0x3aa"))!=NULL)  // control checking
      {
         g_print("check received");
         read(fpoin,buff1,sizeof(buff1));
         sprintf(ch,"%d %s",sad,buff1);  //forming flist with addres
      }
     write(rport,ch,strlen(ch));        //file name list to port
    }  close(fpoin);   } }

void rtrans()
{
  // file name transmitter thread sends file list
  char fbuf[1024];
  int ad=0;
  tpo=fopen("/root/temp","r");
   cflag=1;
         if(tpo==NULL)
         {
            g_print("error in tpo");
         }
         else
         {
           while((tch=fgetc(tpo))!=EOF)
           {
              tbuffer[ti]=tch;
              ti++;
           }
           tbuffer[ti]='\0';
           switch(bflag)
           {
              case 2:sprintf(fbuf,"%d %d %s",gsrc,button2,tbuffer);
                     break;
              case 3:sprintf(fbuf,"%d %d %s",gsrc,button3,tbuffer);
                     break;
           }
           tes=write(rport,fbuf,strlen(fbuf));
           if(tes<0)
           {
             g_print("error");
           }
           else
           {       ti=0;          }       }     }
```

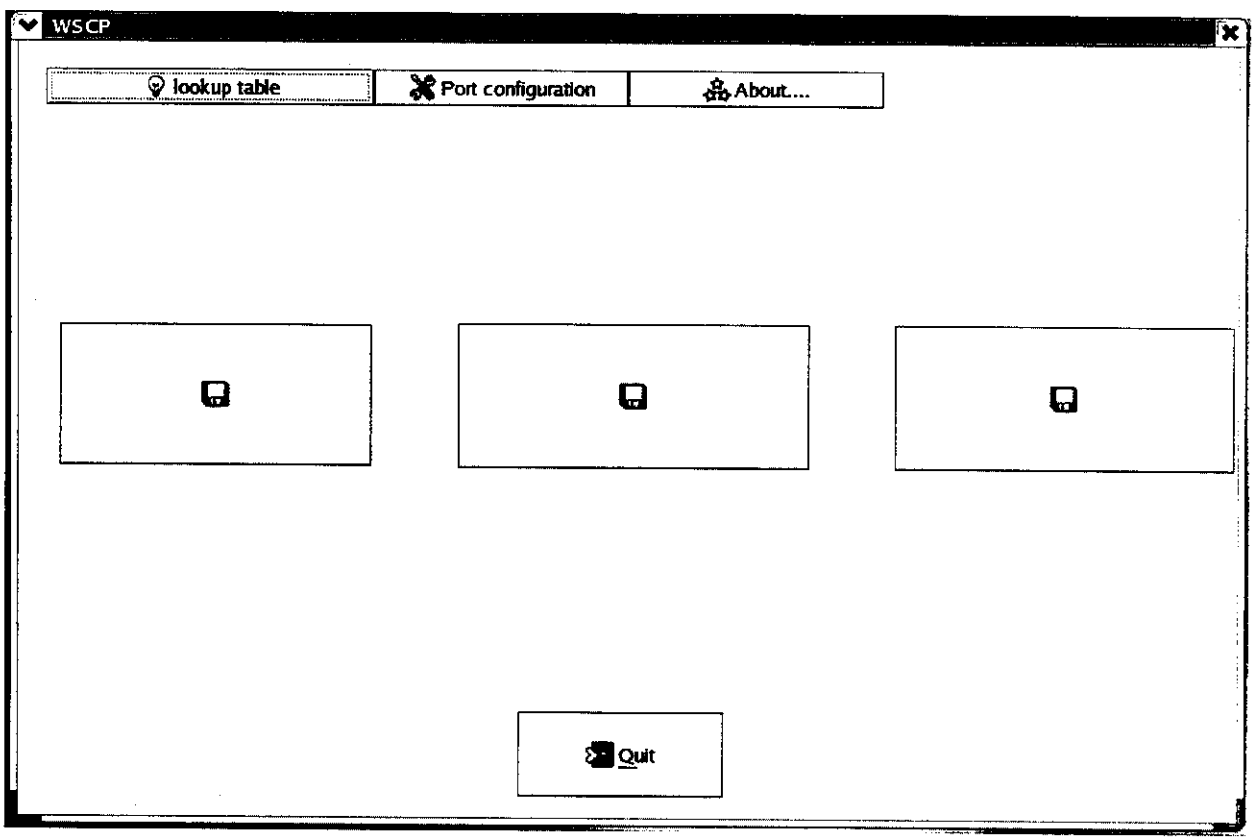
9.2 OUTPUT SCREENS

Main screen:

This is the entry screen that greets the user.

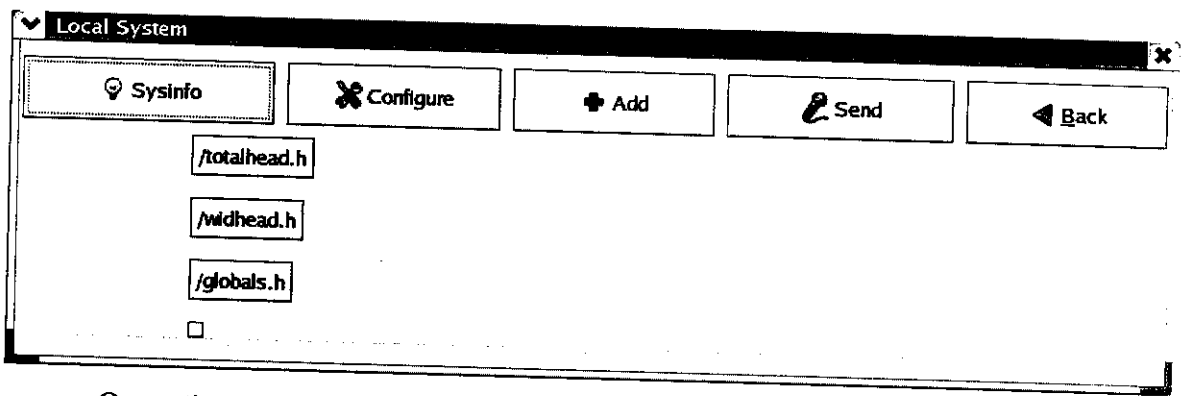


OPERATION SCREEN:



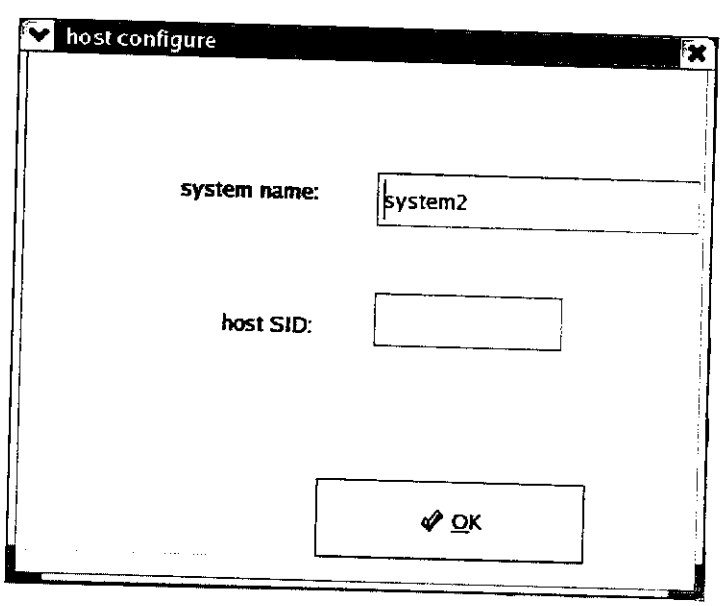
This is the screen which contains all the operations. Here the systems connected in the network are shown as buttons; the first button always denotes the current system the other buttons denote remote systems in network. Port configuration options and lookup table option also opens from this screen.

Current system screen:



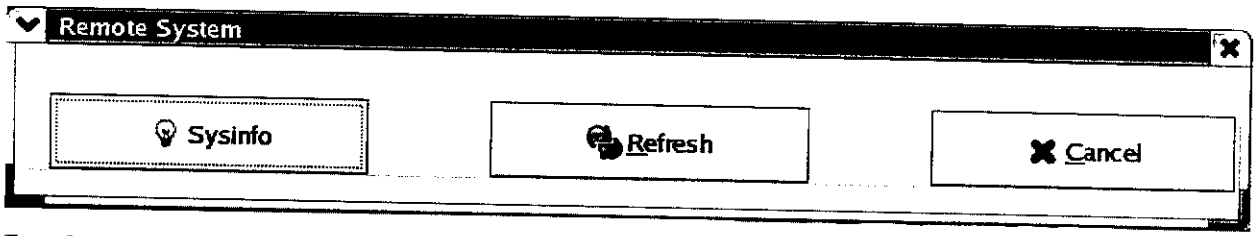
Operations involved in current system are present in this screen. All files shared in the network are also present in this screen.

HOST CONFIGURATION SCREEN:

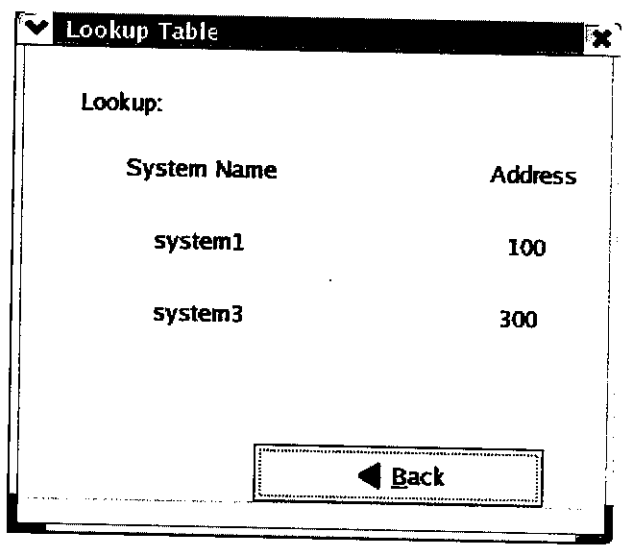


The host address can be set in this screen. This address is configured in to the Linux main directory.

Network system screen:

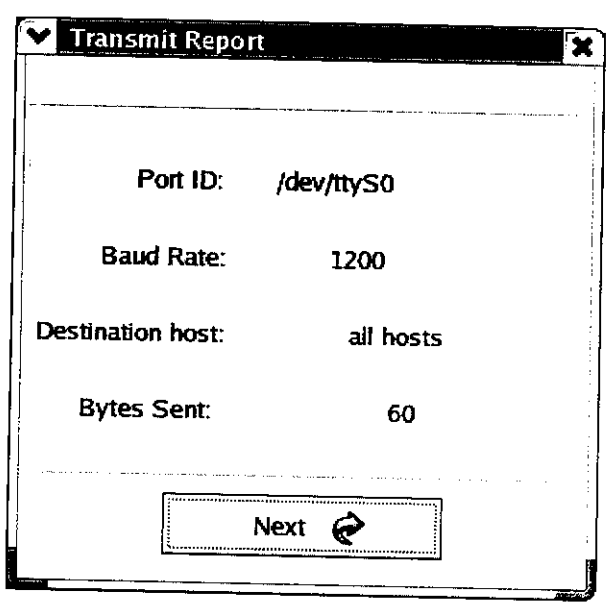


Lookup Table screen:



This screen displays the details of the other systems connected in the network such as their name and address.

Report screen:



This screen is automatically generated after a file has been transmitted it displays status information of the transmission.

10 REFERENCES

1) Carl J. Weisman, "The Essential Guide to RF and Wireless", Addison Wesley Longman, 2000.

2) William Stallings, "Wireless Communications and Networks", Pearson Education Asia, 2002.

3) O'Reilly, "Learning Red Hat Linux", Shroff Publishers,2002.

4) Tim Parker, "Linux Unleashed", Macmillans Publications, 1999.

5) www.ieee.org

6) www.uk.research.att.com/spirit

7) www.opensource.org