1. Introduction:

It is basically a power supply which is controlled by a Laptop in which whole system communicates in 2.4 GHZ ISM band.

ISM band is the Industrial Scientific and Medical band. This is unlicensed band so any one can use this band for proper use.

For communicating 2.4 GHZ ISM band we are using Linksys WRT54GL router.

There are many alternative firmware are available for this router. The firmware that we have used is Linux firmware Open-WRT. Along with the great software for this router, a bunch of hardware hacks are possible.

This router gives the received data from PC to the microcontroller.

Microcontroller understands the commands and provides the switching between different voltages.

The relays are also connected with the controller for the switching of the A.C. appliances like fans, lights etc.

2. Block Diagram of PC Based Controlled Wireless Power Supply.

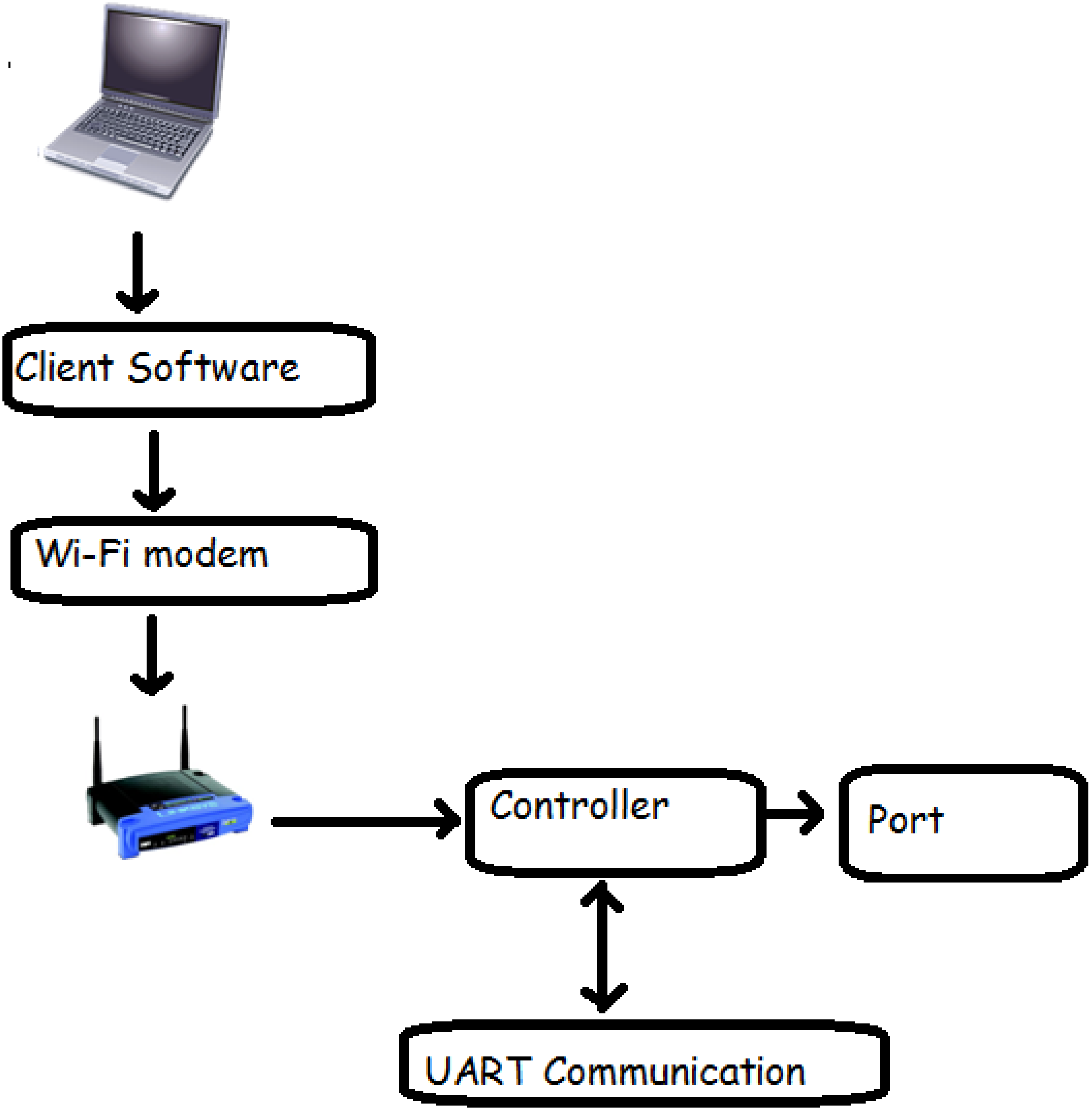


Fig No. (1)

2.1 Explanation of the Block Diagram

Overview

Wireless power supply is basically a controlled power supply in which the commands are sent from a computer and through a router by computer software which uses wifi modem of laptop or PC to transmit data.

These commands are received by wifi router and router sends this data to microcontroller.

Microcontroller understands this data and Gives signal to the base of the transistors which controls the output voltage and operation of the relay.

Application Software:

This software runs on windows (OS) based PCs as it is developed in NetBeans IDE.

It transmits the codes “a”, “b” …..”q” to the controller through the router. In actual sense it sends the ASCII equivalent codes of the respective alphabets such as for “a” it is 61H and so on…..

The logic behind this software is of socket programming.

The table for the same is given on the very next page.

PROCESS	DATA ON PORT	PORT	KEY	DATA SENT
5 V	**0FEH**	**P0**	**a**	**61H**
6 V	**0FDH**	**P0**	**b**	**62H**
8 V	**0FBH**	**P0**	**c**	**63H**
10 V	**0F7H**	**P0**	**d**	**64H**
12 V	**0EFH**	**P0**	**e**	**65H**
15 V	**0DFH**	**P0**	**f**	**66H**
18 V	**0BFH**	**P0**	**g**	**67H**
24 V	**07FH**	**P0**	**h**	**68H**
REL 1	**07FH**	**P2**	**i**	**69H**
REL 2	**0BFH**	**P2**	**j**	**6AH**
REL 3	**0DFH**	**P2**	**k**	**6BH**
REL 1&2	**3FH**	**P2**	**l**	**6CH**
REL 2&3	**9FH**	**P2**	**m**	**6DH**
REL 1&3	**5FH**	**P2**	**n**	**6EH**
ALL REL ON	**00H**	**P2**	**o**	**6FH**
ALL VOLT OFF	**0FFH**	**P0**	**p**	**70H**
ALL REL OFF	**0FFH**	**P2**	**q**	**71H**

Table No. (1)

Wifi modem: Data is transmitted by wifi modem at the frequency at 2.4 GHz.

Wifi Router: We used WRT54GL wifi router. It runs on open WRT OS installed in the router which converts Received data to serial data compatible to controller.

Microcontroller: We used AT89C51 as our microcontroller. It is the product of ATMEL.

It takes the data from router and interprets the data and converts into commands and sends data to the bases of transistors which will either work in saturation of cut-off mode for both the voltage output and relay output.

There is a serial interface between the Router and the controller and that is UART communication.

3. Hardware Section:

3.1. Linksys WI FI Router WRT54GL

3.1.1 Reason for choosing the Router:

- The Linksys WRT54GL is a very useful Router.
- It runs a versions of Linux and very hackable.
- Since it is open source, a number of third-party versions have sprung up which add significant functionality beyond what the stock firmware provided. In addition to the software improvements, several hardware modifications can also be done in this router. Two of the most popular are to add dual serial ports as well as an SD card.
- After these modifications, the router can be used to control external electronics and the SD card can be used to add Gigabytes of storage capability.

3.1.2 ROUTER MODIFICATION SECTION:

Wifi router receives data from the wifi modem and transmits it to the serial ports.

Following image shows the serial ports.

logic hence we require MAX232 IC.

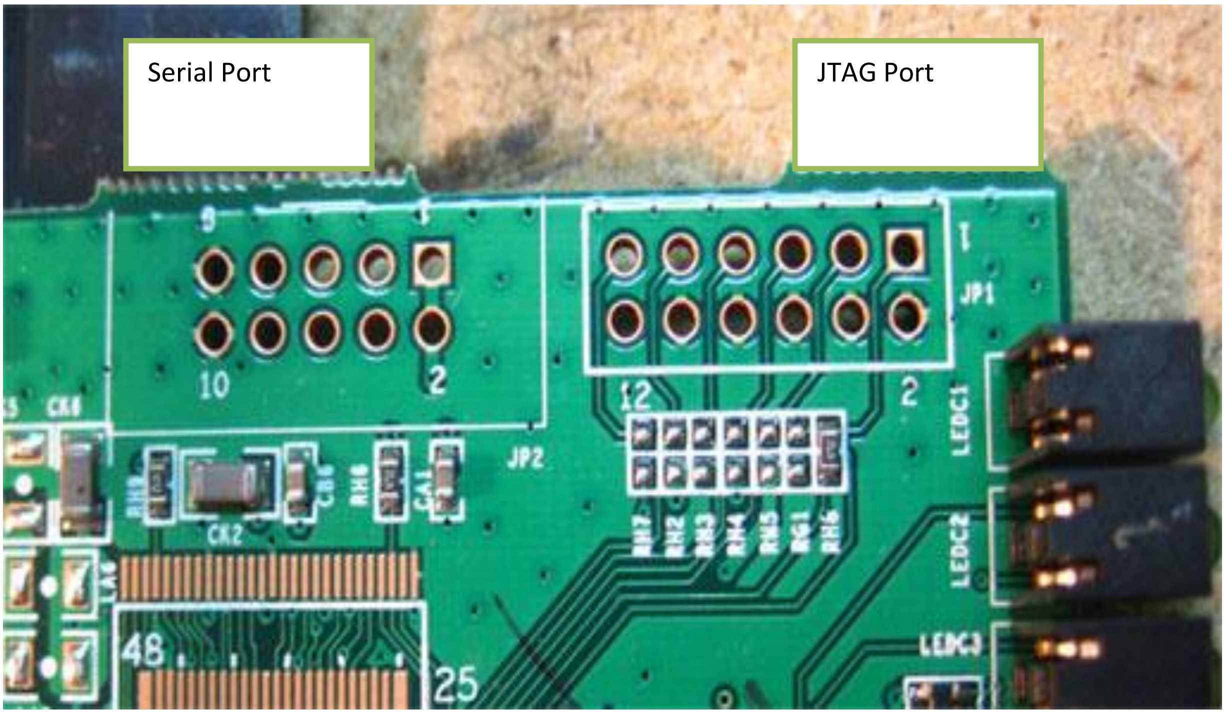


Fig No. (2)

To observe the received data on computer we have to convert TTL logic to RS232 This IC is connected in TX and RX pins of TTS0 Ports. So we can see received data. Pin configuration of router serial port.

Pin 1: 3.3V	Pin 2: 3.3V
Pin 3: Tx (ttyS1)	Pin 4: Tx (ttyS0)
Pin 5: Rx (ttyS1)	Pin 6: Rx (ttyS0)
Pin 7: NC	Pin 8: NC
Pin 9: GND	Pin 10: GND

The output of the Router is TTL compatible that means it can be directly given to the controller. But at the time of trouble shooting we have to check that which data we are actually sending to the Router. In that case we will convert this TTL output of Router to the RS232 standards with the help of MAX232 IC. Then this output is given to the serial port of PC. Now in the PC through HyperTerminal we can analyse the output from router. The circuit connection for MAX 232 is shown below.

3.1.3. Circuit diagram of MAX 232 connections between PC and Router

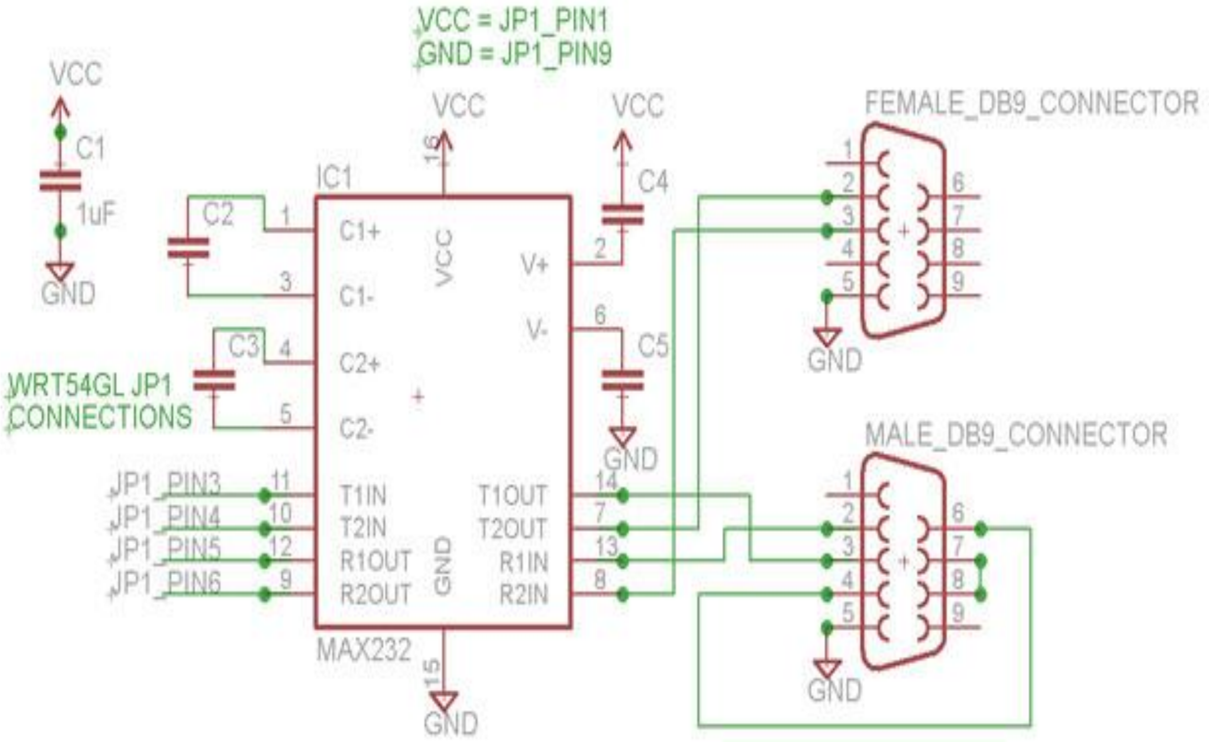


Fig No. (3)

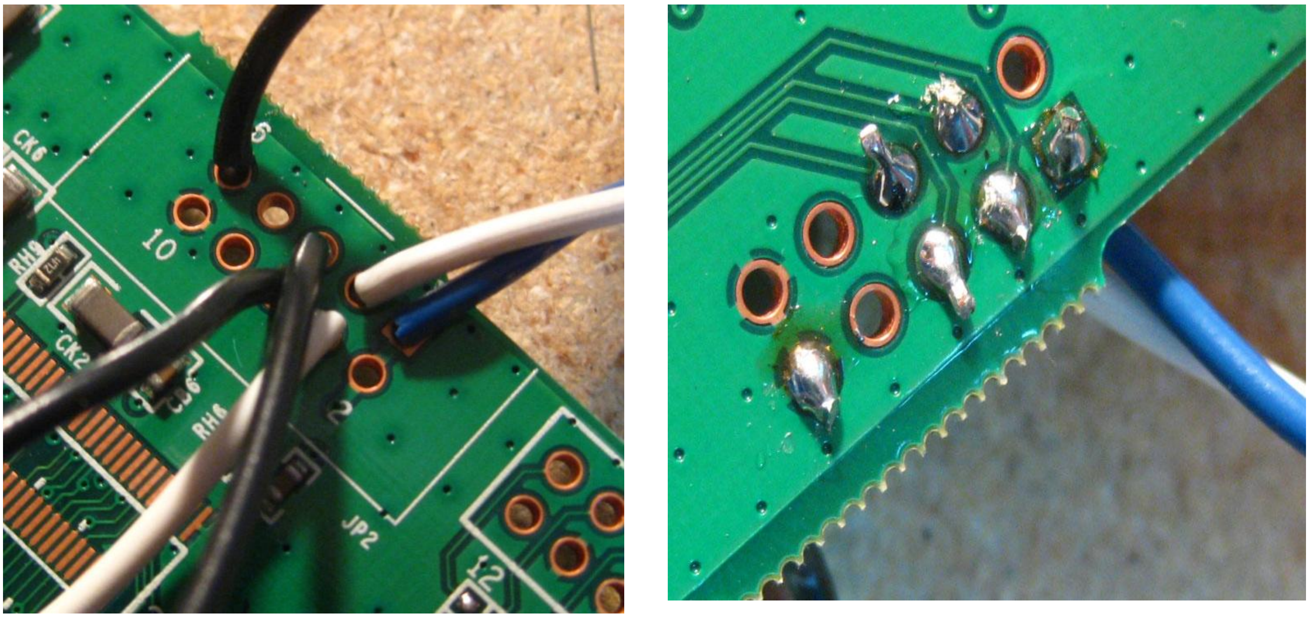


Fig No. (4)

3.2 Microcontroller AT89C51

3.2.1 Reason for Selecting this Microcontroller

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4Kbytes of Flash programmable and erasable read only memory (PEROM). The devices manufactured using Atmel's high-density non-volatile memory Technology and is compatible with the industry-standard MCS-51 instruction set and pin out. The on-chip Flash allows the program memory to be reprogrammed In-system or by a conventional non-volatile memory programmer. By combining a Versatile 8-bit CPU with Flashon a monolithic chip, the Atmel AT89C51 is a Powerful microcomputer which provides a highly-flexible and cost-effective Solution to many embedded control applications.

3.2.2 Features of AT89C51

- Compatible with MCS-51 products.
- 4 Kbytes in system reprogrammable flash memory with 1000 write/erase cycles.
- Fully Static operation: 0 HZ to 24 MHZ
- Three level programme memory lock
- 128 × 8-bit internal RAM
- 32 programmable I/O Lines
- Two 16 bit Timer Counters
- Programmable Serial Channel
- Low power Idle and Power Down modes.

3.2.3 Pin Diagram of AT89C51

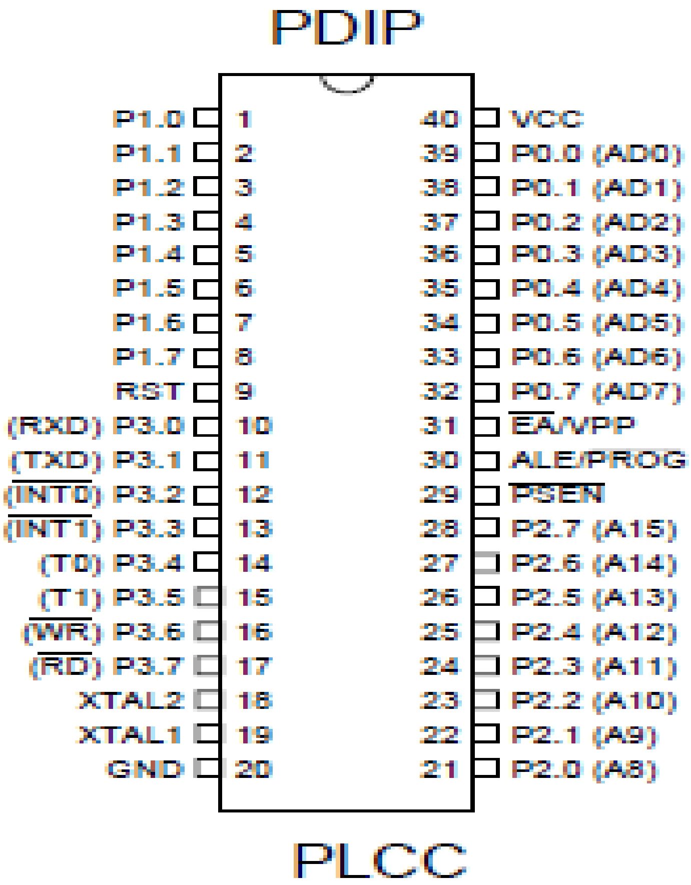


Fig No. (5)

3.3 PCB SECTION

3.3.1 Power and Controller Circuit PCB Layout

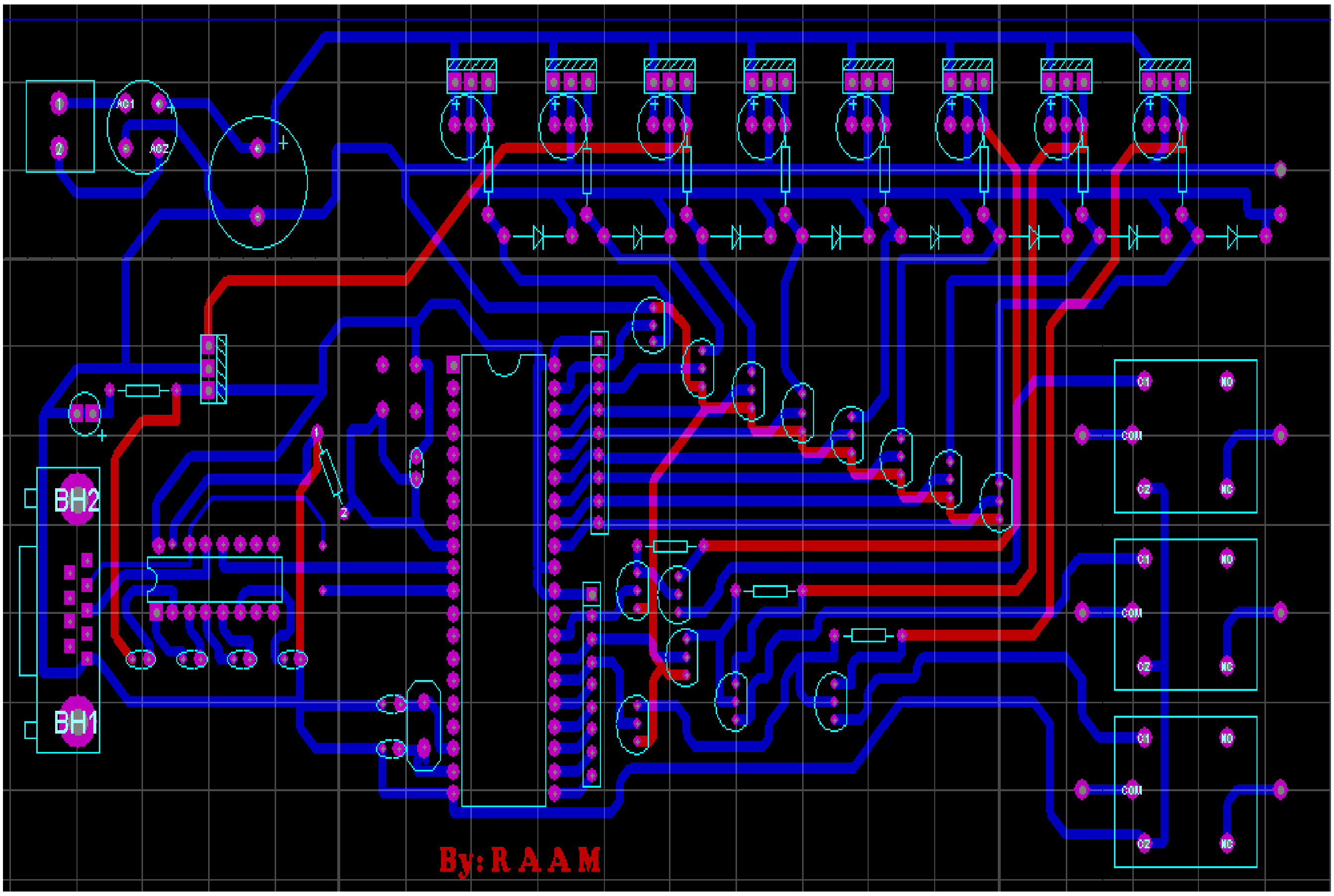


Fig No. (6)

3.3.2 3D View of PCB

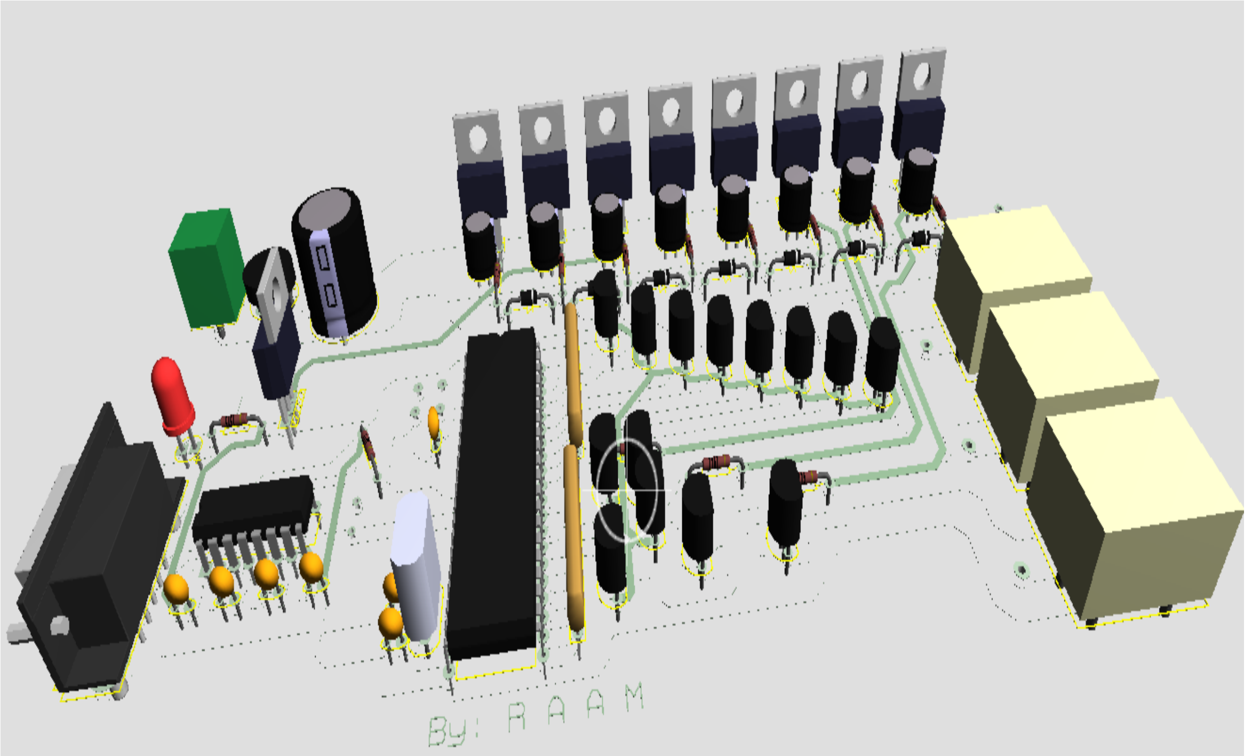


Fig No. (7)

3.3.3 Bottom Copper Layout

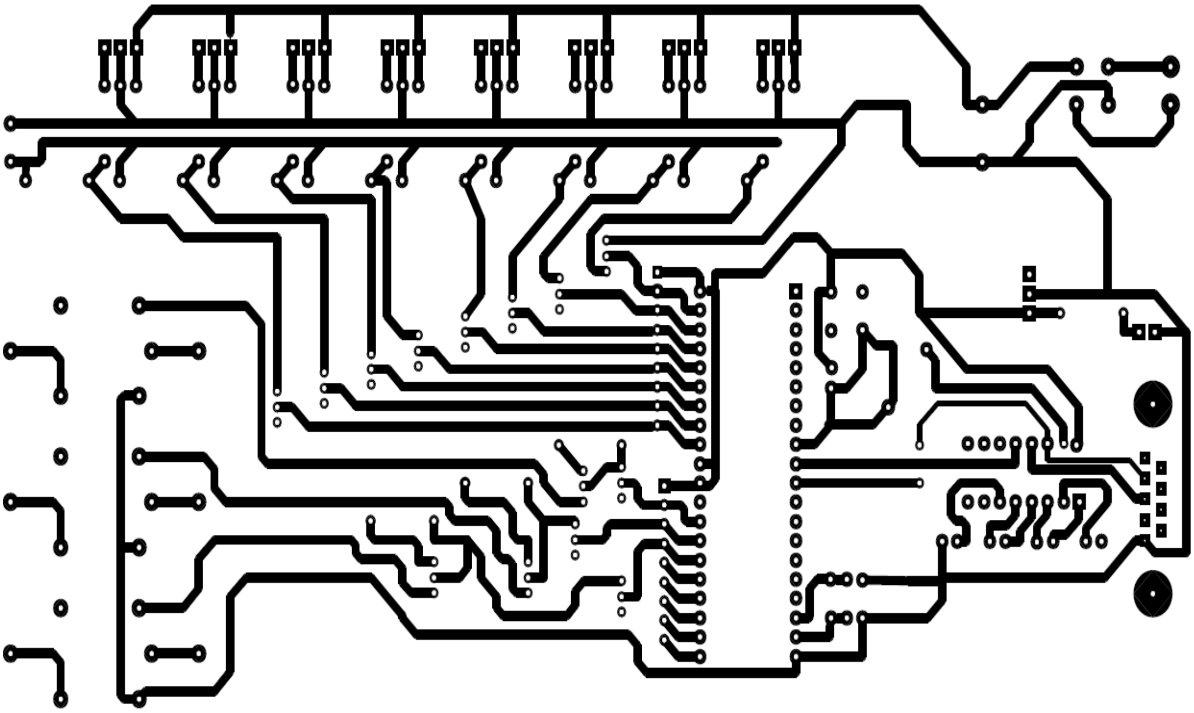


Fig No. (8)

3.3.4 Circuit Diagram

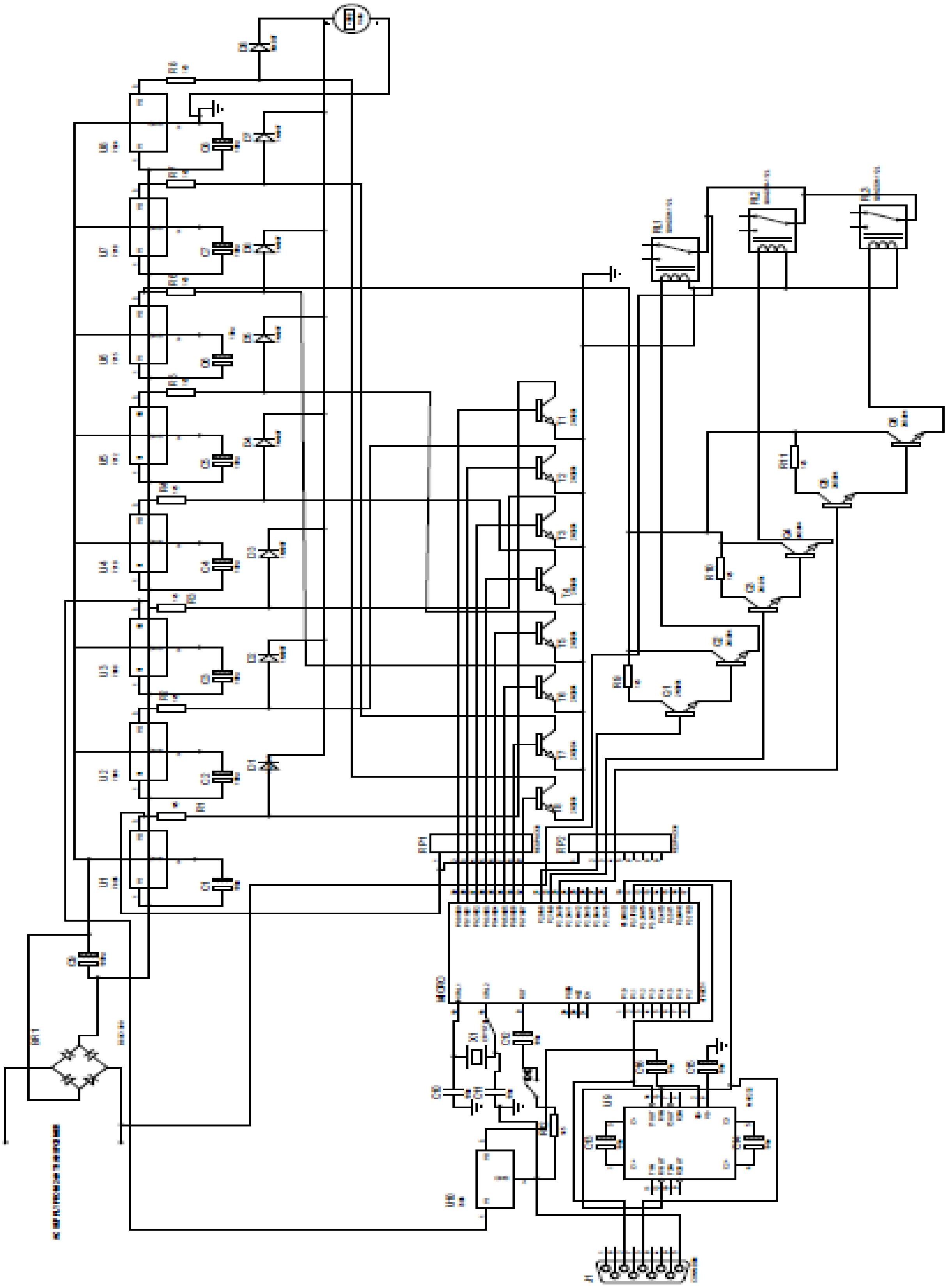


Fig No. (9)

COMPONENTS	QUANTITY	AMOUNT
AT89C51 CONTROLLER	1	105
CONTROLLER TRAINER	1	70
7805 REGULATOR	1	10
11.0592 CRYSTAL	2	20
33 PF CAPACITORS	2	2
CONTROLLER SOCKET	1	20
DB 9 CONNECTORS	2	50
Transformer 18-0-18	1	35
9 V Battery	2	30
12V Relay	3	36
DB 9 CONNECTORS	2	10
IC Socket	4	18
Max 232	2	20
LED	30	30
SWITCH	2	10
Rectifier Bridge	1	10
Transistor	15	23
Resonator	2	10
CAPACITORS	22	50
78XX	7	86
DIODE	15	7
SOLDER WIRE	1	40
HUCKUP WIRE	1	30
HEAT SINK	10	80
MICA SET	10	40
Max 232	2	80
CAPACITOR	1	17
IC 7806	1	18
PLUG	1	10
Transistors	15	45
Resistors	30	15
ic 7812	1	15
PCB	1	315
Router	1	3000

Table No. (2)

4. Software Section

4.1 Controller Program

Flow chart of Controller Program

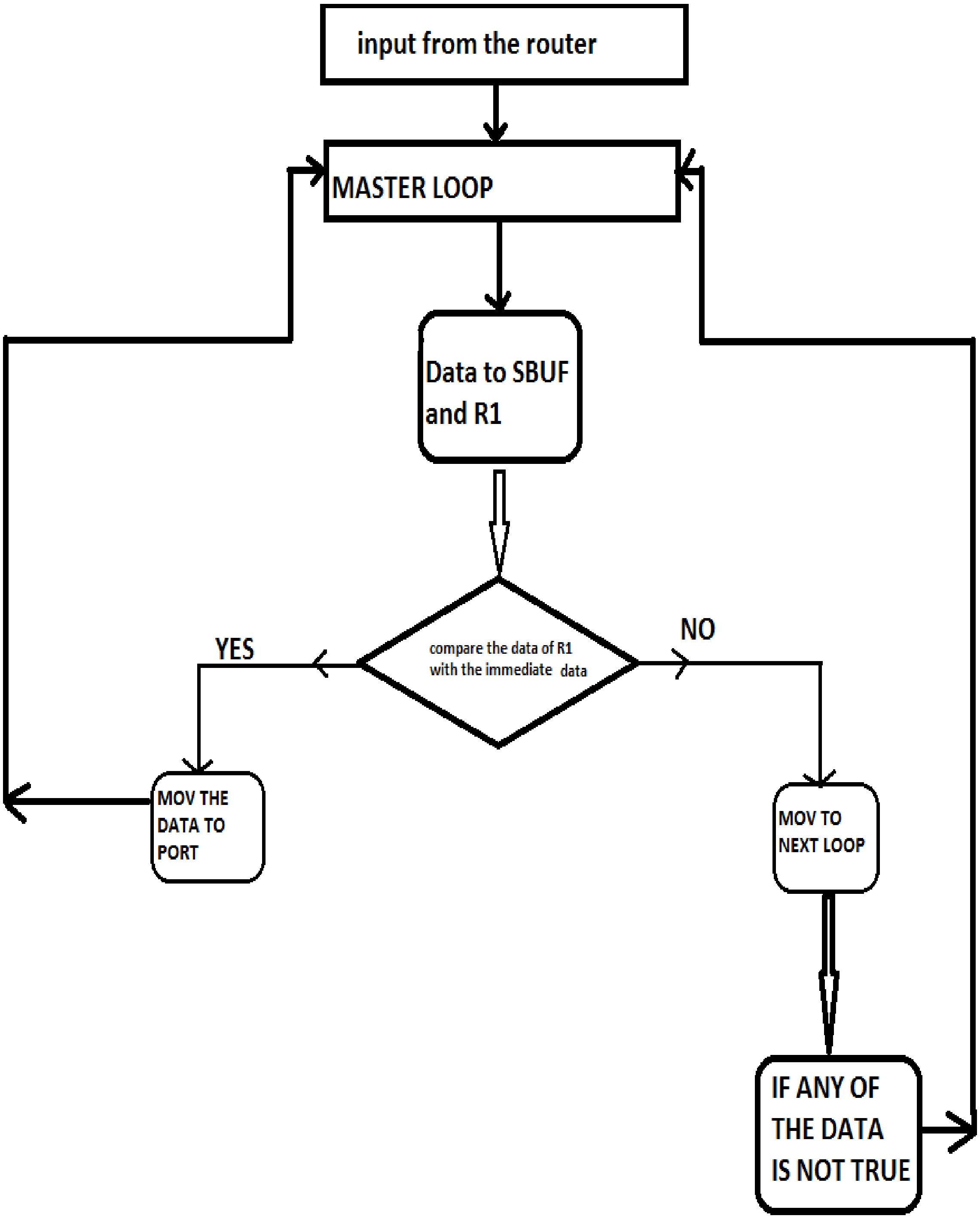


Fig No. (10)

The data from the user is taken from the user with the help of a GUI program and this data is sent to the router through Wifi modem of Laptop. This data is received by the Router. The GUI program sends the alphabets. But it actually sends the equivalent ASCII codes of the respective data as shown below. The output of the router is serial which is given to the Receiver pin of the controller that is pin no. 10 of controller. The codes for the data and respective voltage and relay operation are given in table no 1.

As soon as the controller starts it enters to MASTER LOOP. When RI bit of SCON SFR sets it starts loads the data of SBUF to R1 Register with resetting RI bit to 0.

Then it starts to compare the received data with the immediate data which is nothing but the same equivalent ASCII code of the sent alphabet. If both the data are matched then a particular data is sent to port 0 for voltage and to port 2 for relay as shown in the table for codes. And program sequence is switched to MASTER loop.

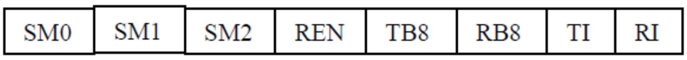
If both data are not matched than next loop will be executed means next data is compared and programme sequence moves to next loop until a match is found. At the end of the program the sequence is again switched to the MASTER loop.

4.1 Explanation of Controller Program

In modern multiprocessor distributed system, one computer must be able to communicate with other computers. One **cost-effective way to communicate is to send and receive data bits serially**. The 89C51 has a serial data communication circuit that uses register SBUF to hold data, Register SCON controls data communication, register PCON controls data rates, and pins RXD (P3.0) and TXD (P3.1) connect to serial data network.

SBUF is physically two registers. **One is write only and is used to hold data to be transmitted out of the 89C51 via TXD. The other is read only and holds received data from external sources via RXD.** Both mutually exclusive registers use address 99 H.

- **Structure or Format of SCON SFR:**



D7	D6	D5	D4	D3	D2	D1	D0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

D7 SM0 **Serial port Mode bit 0:** Set / Cleared by program to select mode.

D6 SM1 **Serial port Mode bit 1:** Set / Cleared by program to select mode.

SM0	**SM1**	**Mode**	**Description**
0	0	0	Shift Register; baud = f/12
0	1	1	8-bit UART; baud = variable
1	0	2	9-bit UART; baud = f/32 or f/64
1	1	3	9-bit UART; baud = variable

D5 SM2 **Multiprocessor Communication Bit.** Set / Cleared by program to enable multiprocessor communications in modes 2 and 3. When **set to 1 an interrupt is generated if bit 9 of the received data is 1; no interrupt is generated if bit 9 is a 0**. If **set to 1 for mode1, no interrupt will be generated unless a valid stop bit is received.** Clear to 0 if mode 0 is in use.

D4 REN **Receive Enable bit. Set to 1 to enable reception**; **cleared to 0 to disable reception**.

D3 TB8 **Transmitted bit 8.** Set / Cleared by program in modes 2 and 3.

D2 RB8 **Received bit 8.** Bit 8 of received data in modes 2 and 3; stop bit in mode 1. Not used in mode 0.

D1 TI **Transmit Interrupt flag. Set to one at the end of bit 7 times in mode 0** and at the **beginning of the stop bit for other modes**. **Must be cleared by the program.**

D0 RI **Receive Interrupt flag. Set to one at the end of bit 7 times in mode 0** and **halfway through the stop bit for other modes**. **Must be cleared by the program.**

Note: SCON SFR is **bit addressable** and its internal one byte address is **98H.**

- **Structure or Format of TMOD SFR:**

D7	D6	D5	D4	D3	D2	D1	D0
Gate	C/T’	M1	M0	Gate	C/T’	M1	M0
Timer 1				Timer 0			

Bit Symbol Function

D7 Gate Gating Control. OR gate enable bit which controls RUN/STOP of timer 1. Set to 1 by program to enable timer to run of bit TR1 in TCON is set and signal on eternal interrupt INT 1’ pin is high. Cleared to 0 by program to enable timer to run if bit TR1 in TCON is set.

D6 C/T’ Timer or Counter Selector. Set to 1 by program to make timer 1 act as a counter by counting pulses from external input pin 3.5 (T1). Cleared to 0 by program to make timer act as a timer by counting internal frequency.

D5 M1 Timer / Counter operating mode select bit 1. Set / Cleared by program to select mode.

D4 M0 Timer / Counter operating mode select bit 0. Set / Cleared by program to select mode.

D3 Gate OR gate enable bit which controls RUN/STOP of timer 0. Set to 1 by program to enable timer to run of bit TR0 in TCON is set and signal on eternal interrupt INT 0’ pin is high. Cleared to 0 by program to enable timer to run if bit TR0 in TCON is set.

D2 C/T’ Set to 1 by program to make timer 0 act as a counter by counting pulses from external input pin 3.4 (T0). Cleared to 0 by program to make timer act as a timer by counting internal frequency.

D1 M1 Timer / Counter operating mode select bit 1. Set / Cleared by program to select mode.

D0 M0 Timer / Counter operating mode select bit 0. Set / Cleared by program to select mode.

M1	M0	Mode	Remark
0	0	0	13-bits timer / counter
0	1	1	16-bits timer / counter
1	0	2	8-bit auto reload.
1	1	3	Timer 0 works as a two 8-bits separate timer / counter. Only for timer 0

Note: TMOD SFR is not bit addressable

Serial Data Interrupt: Serial data communication is a **relatively slow process, occupying many milliseconds per data byte to accomplish.** In order **not to tie up valuable processor time, Serial data flags are included in SCON to aid in efficient data transmission and reception.** Notice that **data transmission is under the complete control of the program, but reception of data is unpredictable and at random times** that are beyond the control of the program.

The serial data flags in TCON, TI and RI, are set whenever a data byte is transmitted (TI) or received (RI). **These flags are ORed together to produce an interrupt to the program. The program must read these flags to determine which caused the interrupt** and then **clear the flag.** This is **unlike the timer flags that are cleared automatically; it is the responsibility of the programmer to write routines that handle the serial data flags.**

- **Data Reception**: Reception of serial data will begin if the receive enable bit (REN) in SCON is set to 1 for all modes. In addition, for mode 0 only, RI must be cleared to 0. Receiver Interrupt flag RI is set after data has been received in all modes. **Setting REN is the only direct program control that limits the reception of unexpected data; the requirement that RI also be 0 for mode 0 prevents the reception of new data until the program has dealt with the old data and reset RI.**

Reception can begin in modes 1, 2 and 3 if **RI is set** when the serial stream of bits begins. **RI must have been reset** by the program before the last bit is received or **the incoming data will be lost. Incoming data is not transferred to SBUF until the last bit has been received so that the previous transmission can be read from SBUF while new data is being received.**

- **Structure or Format of TCON SFR:**

D7	D6	D5	D4	D3	D2	D1	D0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Bit	Symbol	Function
D7	TF1	Timer 1 Overflow Flag. Set when all the bits of timer 1 roll from 1's to 0's. Cleared when processor vectors to execute interrupt service routine located at program address 001B H.
D6	TR1	Timer 1 Run control bit. Set to 1 by program to enable timer to count; cleared to 0 by program to halt timer. Does not reset timer.
D5	TF0	Timer 0 Overflow Flag. Set when all the bits of timer 1 roll from 1's to 0's. Cleared when processor vectors to execute interrupt service routine located at program address 000B H.
D4	TR0	Timer 0 Run control bit. Set to 1 by program to enable timer to count; cleared to 0 by program to halt timer. Does not reset timer.
D3	IE1	External Interrupt 1 Edge flag. Set to 1 when a high to low edge is received on port 3 pin 3.3. (INT 1'). Cleared when processor vectors to interrupt service routine located at program address 0013H. Not related to timer operation.
D2	IT1	External Interrupt 1 signal Type control bit. Set to 1 by program to enable external interrupt to be triggered by a falling edge signal. Set to 0 by program to enable a low level signal on external interrupt 1 to generate an interrupt.
D1	IE0	External Interrupt 0 Edge flag. Set to 1 when a high to low edge is received on port 3 pin 3.2. (INT 0'). Cleared when processor vectors to interrupt service routine located at program address 0003H. Not related to timer operation.
D0	IT0	External Interrupt 0 signal Type control bit. Set to 1 by program to enable external interrupt to be triggered by a falling edge signal. Set to 0 by program to enable a low level signal on external interrupt 1 to generate an interrupt.

Note: TCON SFR is bit addressable.

OUR PROGRAMM IS AS GIVEN BELOW

ORG 0000H

MOV TMOD,#20H   ***/**

0	0	1	0	0	0	0	0

The gate bit is selected 0 . C/T' bit is 0 to enable timer 1. The bit M1 is 1 and M0 is selected 0 for operating our timer1 in mode 2. And now as we do not uses the timer 0 hence all other bits of the TMOD SFR is kept 0.*/

MOV SCON,#50H   */

0	1	0	1	0	0	0	0

The bit SM0 is 0 & bit SM1 is 1 so that our serial communication will operated in mode1. SM2 bit is set to 0. REN bit is set to 1 for enable the reception. TB8 ,RB8, TI,RI bit is set to 0 according to requirement.

Now, here our requirement of the program is to use the baud rate of 9600 bps.

Now for the timer operation the actual count can be found by the equation below,

Actual count =28800/9600= 03h

So the load count can be given as the 2's compliment of the actual count which is 0FDh. So we load 0FDh in both TH1 & TL1. */

MOV TH1,#0FDH

MOV TL1,#0FDH

*/ here we set the timer 1 by this instruction */ORG 0000H

SETB TR1

MASTER: JNB RI, MASTER

```
CLR RI

MOV R1, SBUF

VOLTOFF: CJNE R1, #70H, RELOFF

MOV P0, #0FFH

LJMP MASTER

RELOFF: CJNE R1, #71H, MAIN

MOV P2, #0FFH

LJMP MASTER

MAIN: CJNE R1, #61H, CHECK

MOV P0, #0FEH

LJMP MASTER

CHECK: CJNE R1, #62H, SET2

MOV P0, #0FDH

LJMP MASTER

SET2: CJNE R1, #63H, SET3

MOV P0, #0FBH

LJMP MASTER

SET3: CJNE R1, #64H, SET4

MOV P0, #0F7H

LJMP MASTER

SET4: CJNE R1, #65H, SET5

MOV P0, #0EFH
```

```
LJMP MASTER
SET5: CJNE R1, #66H, SET6
MOV P0, #0DFH
LJMP MASTER
SET6: CJNE R1, #67H, SET7
MOV P0, #0BFH
LJMP MASTER
SET7: CJNE R1, #68H, SET1
MOV P0, #7FH
LJMP MASTER
SET1: CJNE R1, #69H, SET8
MOV P2, #7FH
LJMP MASTER
SET8: CJNE R1, #6AH, SET9
MOV P2, #0BFH
LJMP MASTER
SET9: CJNE R1, #6BH, SET10
MOV P2, #0DFH
LJMP MASTER
SET10: CJNE R1, #6CH, SET11
MOV P2, #3FH
LJMP MASTER
```

```
SET11: CJNE R1, #6DH, SET12
MOV P2, #9FH
LJMP MASTER
SET12: CJNE R1, #6EH, SET13
MOV P2, #5FH
LJMP MASTER
SET13: CJNE R1, #6FH, SET14
MOV P2, #00H
LJMP MASTER
SET14: LJMP MASTER
END
```

4.2 Router OS Installation and Trouble Shooting

To have open WRT as our OS or firmware, we must install it to the router.

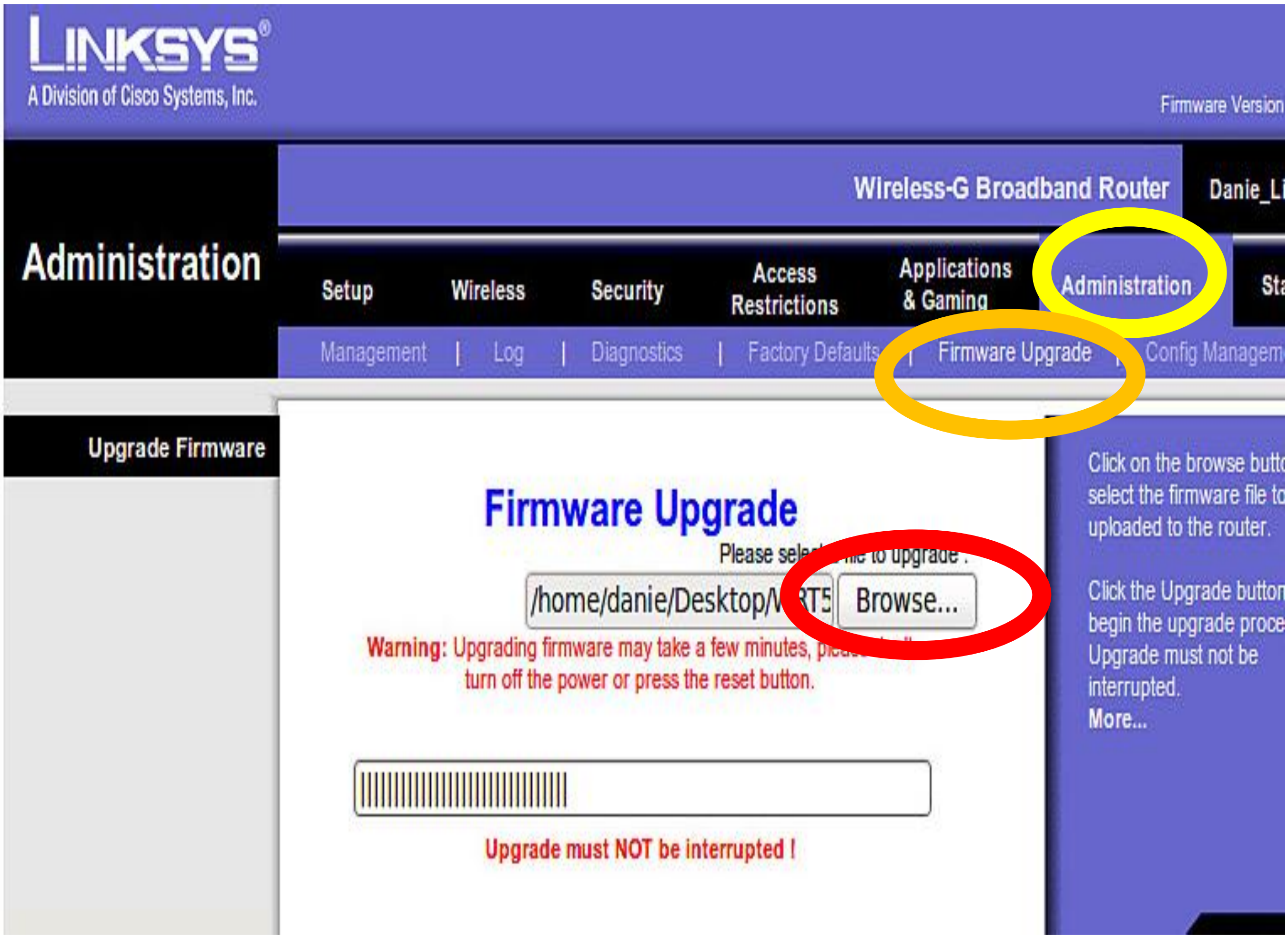
For that there are 2 methods to upgrade the firmware:

1) without any software
2) by using “ipkg” software.

Here we’ll describe only the first method. Follow the below steps to install the firmware.

4.2.1 Method 1

- Connect router to type PC via RJ45 cable or Laptop with wifi modem(Router must be power supplied correctly)
- Write default ip address that is “192.168.1.1” to the address bar of the internet browser.
- It’ll ask you to enter password and User id..: Enter the correct value.
- Now if data is correct than you’ll transferred to the home screen.
- Now click on the “administration”
- click on the “Firmware upgrade”
- click on the “Browse”
- Locate the bin file(OS) that you want to install.
- click on the “OK”
- Now it will show process of upgrading the firmware as shown in below figure.
- After completion it will show you that ”Upgrade is successful”
- Now if you write IP address again to the browser window it’ll show you the home screen of the another OS.



Please sel… file to upgrade :

/home/danie/Desktop/W…RT5

Warning: Upgrading firmware may take a few minutes, please… turn off the power or press the reset button.

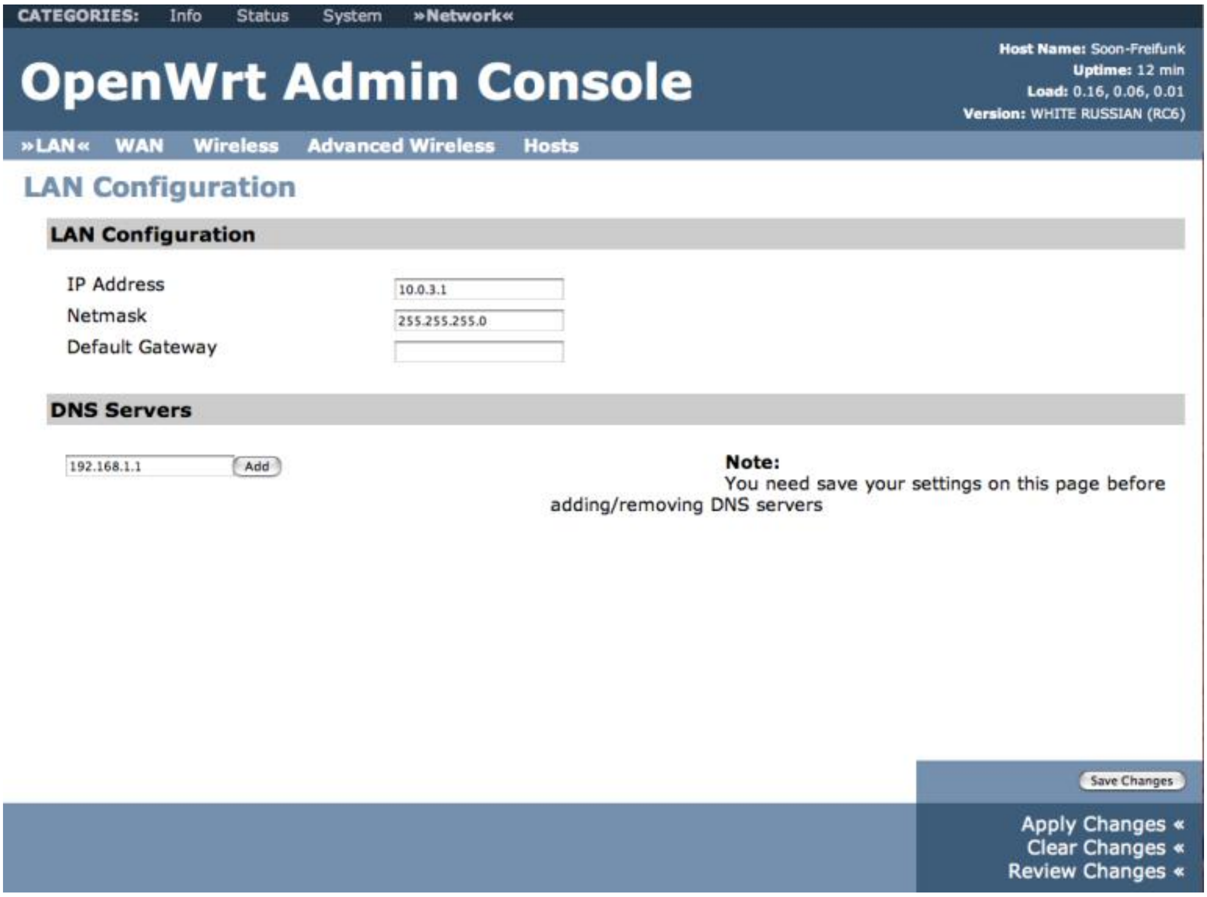
Upgrade must NOT be interrupted !

Click on the browse butto… select the firmware file to… uploaded to the router.

Click the Upgrade button… begin the upgrade proce… Upgrade must not be interrupted.

More...

Fig. No. (11)


CATEGORIES: Info Status System »Network«
OpenWrt Admin Console
»LAN« WAN Wireless Advanced Wireless Hosts
Add
Save Changes
Apply Changes «
Clear Changes «
Review Changes «

Host Name: Soon-Freifunk
Uptime: 12 min
Load: 0.16, 0.06, 0.01
Version: WHITE RUSSIAN (RC6)

LAN Configuration

LAN Configuration

IP Address 10.0.3.1
Netmask 255.255.255.0
Default Gateway

DNS Servers

192.168.1.1

Note:
You need save your settings on this page before adding/removing DNS servers

Fig No. (12)

4.2.2 Testing the Firmware

Hardware modification made are now tested here with Putty software.

Follow steps are to be followed to test the TTS port.

1) Connect a null modem cable between the TTS/1 DB9 connector and your PC.

2) Open a hyper terminal connection.

Settings: Baud rate: 9600

Data bits: 8

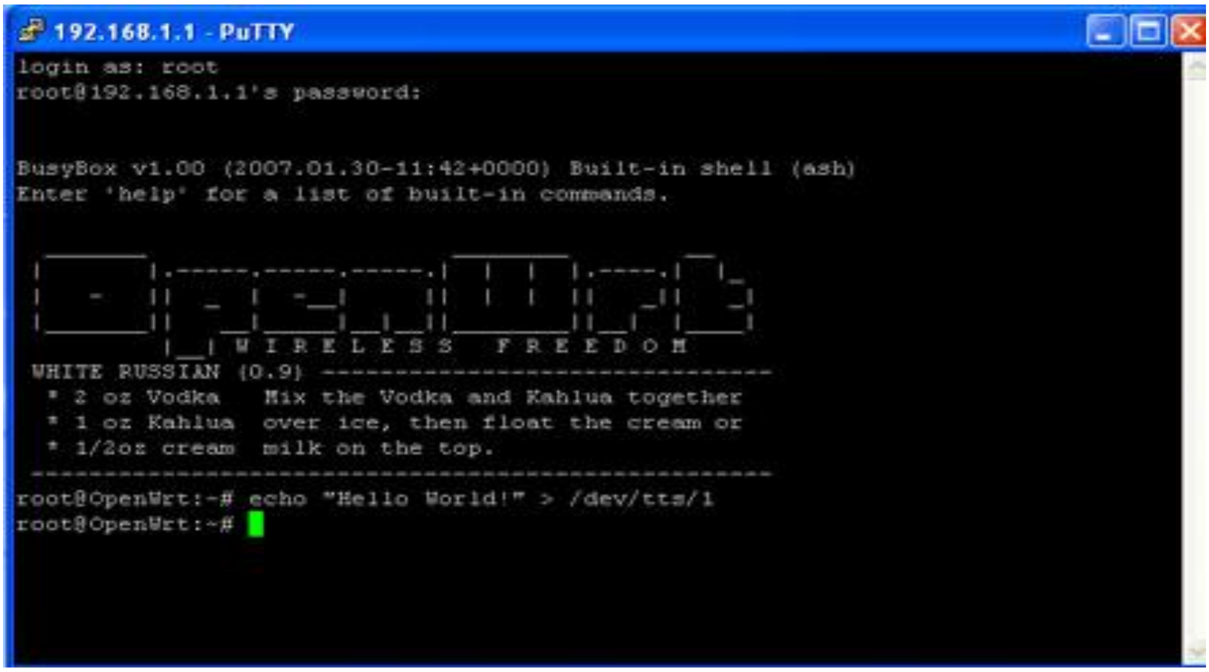
Stop bits: 1

Flow control: None

1) SH in to the router (username 'root' and the password you set earlier) (Putty is a good SSH client for Windows).

2) Enter the command: echo "Hello World!" > /dev/tts/1

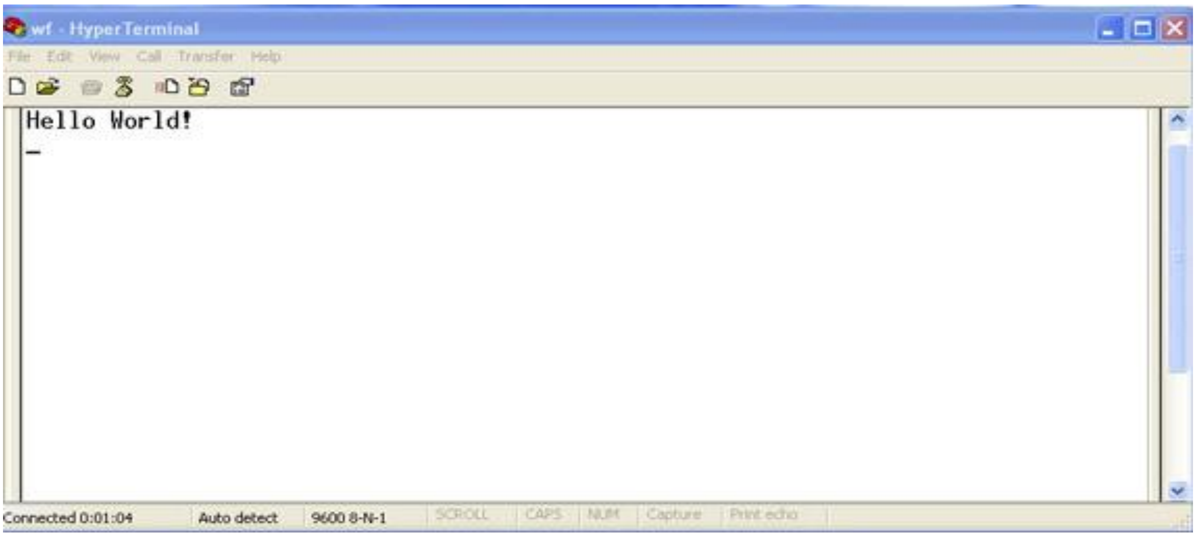
3) Confirm that you see 'Hello World!' on your serial console (in HyperTerminal) as shown below.



```
login as: root
root@192.168.1.1's password:


BusyBox v1.00 (2007.01.30-11:42+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 WHITE RUSSIAN (0.9) -------------------------------
  * 2 oz Vodka   Mix the Vodka and Kahlua together
  * 1 oz Kahlua  over ice, then float the cream or
  * 1/2oz cream  milk on the top.
 ---------------------------------------------------
root@OpenWrt:~# echo "Hello World!" > /dev/tts/1
root@OpenWrt:~#
```



```
Hello World!
_
```

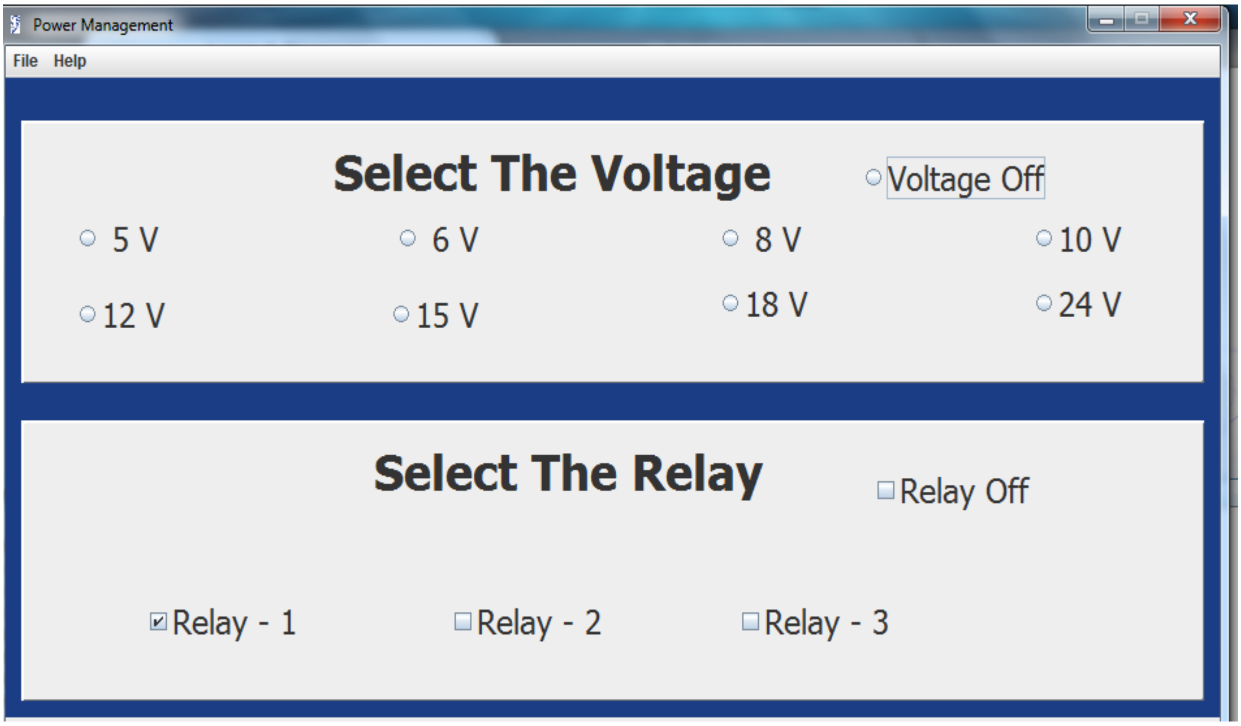
Connected 0:01:04 Auto detect 9600 8-N-1 SCROLL CAPS NUM Capture Print echo

Fig No. (13)

4.3 GUI Program

The user interface program is made in Net Beans IDE. It is an application of JAVA and specially used for application type of programs. The architecture of the program is designed by us but we have get this program done with the help of Computer Engineering Student.

4.3.1 Snapshot of Program



Select The Voltage

○ Voltage Off

○ 5 V ○ 6 V ○ 8 V ○ 10 V

○ 12 V ○ 15 V ○ 18 V ○ 24 V

Select The Relay

□ Relay Off

☑ Relay - 1 □ Relay - 2 □ Relay - 3

Fig No. (14)

4.3.2 Code if the Program

```
import com.jcraft.jsch.Channel;
import com.jcraft.jsch.Session;
import com.jcraft.jsch.JSch;
import com.jcraft.jsch.JSchException;
import com.jcraft.jsch.UserInfo;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.io.IOException;
import java.io.PipedInputStream;
import java.io.PipedOutputStream;
import javax.swing.ImageIcon;
public class MainFrame extends javax.swing.JFrame implements ActionListener{
 JSch  ssh;
 Session session = null;
 Channel channel = null;
 private PipedInputStream fromServer;
 private PipedOutputStream toServer;
 public MainFrame() {
        setIconImage(new
ImageIcon(getClass().getResource("icon_power_management.gif")).getImage());
        this.addWindowListener(new WindowAdapter() {
                    @Override
            public void windowClosing(WindowEvent e) {
                try{
               disconnect();
                }
                catch(Exception ex){ex.printStackTrace();}
            }
        });
```

```
  initComponents();
  btngrp=new javax.swing.ButtonGroup();
  btngrp.add(this.jRadioButton1);
  btngrp.add(this.jRadioButton2);
  btngrp.add(this.jRadioButton3);
  btngrp.add(this.jRadioButton4);
  btngrp.add(this.jRadioButton5);
  btngrp.add(this.jRadioButton6);
  btngrp.add(this.jRadioButton7);
  btngrp.add(this.jRadioButton8);
  btngrp.add(this.jRadioButton9);
  jMenuItem2.addActionListener(new ActionListener(){
      public void actionPerformed(ActionEvent e) {
         new AbtUs().setVisible(true);
   }
});
  jCheckBox1.setSelected(true);
  jRadioButton1.addActionListener(new ActionListener(){
      public void actionPerformed(ActionEvent e) {
     Character b='a';
     try{
 send("echo " +b+b+ " > /dev/tts/1");
     }
     catch(Exception ex){
         ex.printStackTrace();
     }
     //System.out.println(b);
      }
  });
  jRadioButton2.addActionListener(new ActionListener(){
      public void actionPerformed(ActionEvent e) {
      Character b='e';
      try{
```

```
send("echo " +b+b+ " > /dev/tts/1");
   }
   catch(Exception ex){
       ex.printStackTrace();
   }
   //System.out.println(b);
    }
});
jRadioButton3.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e) {
                Character b='b';
    try{
  send("echo " +b+b+ " > /dev/tts/1");
   }
   catch(Exception ex){
       ex.printStackTrace();
   }
   //System.out.println(b);
    }
});
jRadioButton4.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e) {
   Character b='f';
    try{
 send("echo " +b+b+ " > /dev/tts/1");
   }
   catch(Exception ex){
       ex.printStackTrace();
   }
   //System.out.println(b);
    }
});
```

```
 jRadioButton5.addActionListener(new ActionListener(){
     public void actionPerformed(ActionEvent e) {
    Character b='c';
     try{
send("echo " +b+b+ " > /dev/tts/1");
    }
    catch(Exception ex){
        ex.printStackTrace();
    }
    //System.out.println(b);
     }
 });
 jRadioButton6.addActionListener(new ActionListener(){
     public void actionPerformed(ActionEvent e) {
    Character b='g';
     try{
 send("echo " +b+b+ " > /dev/tts/1");
    }
    catch(Exception ex){
        ex.printStackTrace();
    }
    //System.out.println(b);
     }
 });
 jRadioButton7.addActionListener(new ActionListener(){
     public void actionPerformed(ActionEvent e) {
     Character b='h';
     try{
   send("echo " +b+b+ " > /dev/tts/1");
    }
    catch(Exception ex){
        ex.printStackTrace();
    }
    //System.out.println(b);
     }
 });
 jRadioButton8.addActionListener(new ActionListener(){
     public void actionPerformed(ActionEvent e) {
    Character b='d';
     try{
    send("echo " +b+b+ " > /dev/tts/1");
    }
    catch(Exception ex){
        ex.printStackTrace();
    }
    //System.out.println(b);
     }
 });
  jRadioButton9.addActionListener(new ActionListener(){
     public void actionPerformed(ActionEvent e) {
    Character b='p';
     try{
   send("echo " +b+b+ " > /dev/tts/1");
    }
    catch(Exception ex){
        ex.printStackTrace();
    }
    //System.out.println(b);
     }
 });
 jCheckBox1.addActionListener(this);
 jCheckBox2.addActionListener(this);
 jCheckBox3.addActionListener(this);
  jCheckBox4.addActionListener(new ActionListener(){
     public void actionPerformed(ActionEvent e) {
        if(jCheckBox4.isSelected()){
```

```
                jCheckBox1.setEnabled(false);
                jCheckBox2.setEnabled(false);
                jCheckBox3.setEnabled(false);
                Character b='q';
                  try{
        send("echo " +b+b+ " > /dev/tts/1");
         }
         catch(Exception ex){
             ex.printStackTrace();
         }
            //     System.out.println(b);
             }
             else{
                 jCheckBox1.setEnabled(true);
                 jCheckBox2.setEnabled(true);
                 jCheckBox3.setEnabled(true);
                 jCheckBox1.setSelected(false);
                 jCheckBox2.setSelected(false);
                 jCheckBox3.setSelected(false);
             }
          }
        });
        this.setVisible(true);
       try{
           connect();
       }
       catch(Exception ex){ex.printStackTrace();}
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {
        bindingGroup = new org.jdesktop.beansbinding.BindingGroup();
```

```
jPanel2 = new javax.swing.JPanel();
jPanel1 = new javax.swing.JPanel();
jRadioButton8 = new javax.swing.JRadioButton();
jRadioButton5 = new javax.swing.JRadioButton();
jRadioButton3 = new javax.swing.JRadioButton();
jRadioButton1 = new javax.swing.JRadioButton();
jRadioButton6 = new javax.swing.JRadioButton();
jLabel1 = new javax.swing.JLabel();
jRadioButton2 = new javax.swing.JRadioButton();
jRadioButton4 = new javax.swing.JRadioButton();
jRadioButton7 = new javax.swing.JRadioButton();
jRadioButton9 = new javax.swing.JRadioButton();
jPanel3 = new javax.swing.JPanel();
jLabel2 = new javax.swing.JLabel();
jCheckBox2 = new javax.swing.JCheckBox();
jCheckBox3 = new javax.swing.JCheckBox();
jCheckBox1 = new javax.swing.JCheckBox();
jCheckBox4 = new javax.swing.JCheckBox();
jMenuBar1 = new javax.swing.JMenuBar();
jMenu1 = new javax.swing.JMenu();
jMenuItem1 = new javax.swing.JMenuItem();
jMenu2 = new javax.swing.JMenu();
jMenuItem2 = new javax.swing.JMenuItem();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(255, 255, 255));
setFocusTraversalPolicyProvider(true);
setForeground(new java.awt.Color(140, 18, 18));
setLocationByPlatform(true);
setName("Power Management"); // NOI18N
setResizable(false);
```

```
        org.jdesktop.beansbinding.Binding binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.A
utoBinding.UpdateStrategy.READ_WRITE, this,
org.jdesktop.beansbinding.ELProperty.create("Power Management"), this,
org.jdesktop.beansbinding.BeanProperty.create("title"), "Power Control");
        bindingGroup.addBinding(binding);
        jPanel2.setBackground(new java.awt.Color(27, 61, 133));
        jPanel2.setName("jPanel2"); // NOI18N
        jPanel1.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
        jPanel1.setName("jPanel1"); // NOI18N
        jRadioButton8.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
        jRadioButton8.setText("10 V");
        jRadioButton8.setToolTipText("select 10v o/p");
        jRadioButton8.setName("jRadioButton8"); // NOI18N
        jRadioButton5.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
        jRadioButton5.setText(" 8 V");
        jRadioButton5.setToolTipText("select 8v o/p");
        jRadioButton5.setName("jRadioButton5"); // NOI18N
        jRadioButton3.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
        jRadioButton3.setText(" 6 V");
        jRadioButton3.setToolTipText("select 6v o/p");
        jRadioButton3.setName("jRadioButton3"); // NOI18N
        jRadioButton1.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
        jRadioButton1.setText(" 5 V");
        jRadioButton1.setName("jRadioButton1"); // NOI18N
        binding =
org.jdesktop.beansbinding.Bindings.createAutoBinding(org.jdesktop.beansbinding.A
utoBinding.UpdateStrategy.READ_WRITE, jRadioButton1,
org.jdesktop.beansbinding.ELProperty.create("select 5v o/p"), jRadioButton1,
org.jdesktop.beansbinding.BeanProperty.create("toolTipText"));
        bindingGroup.addBinding(binding);
        jRadioButton6.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
        jRadioButton6.setText("18 V");
        jRadioButton6.setToolTipText("select 18v o/p");
        jRadioButton6.setName("jRadioButton6"); // NOI18N
        jLabel1.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N
        jLabel1.setText("Select The Voltage");
        jLabel1.setName("jLabel1"); // NOI18N
        jRadioButton2.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
        jRadioButton2.setText("12 V");
        jRadioButton2.setToolTipText("select 12v o/p");
        jRadioButton2.setName("jRadioButton2"); // NOI18N
        jRadioButton4.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
        jRadioButton4.setText("15 V");
        jRadioButton4.setToolTipText("select 15v o/p");
        jRadioButton4.setName("jRadioButton4"); // NOI18N
        jRadioButton7.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
        jRadioButton7.setText("24 V");
        jRadioButton7.setToolTipText("select 24v o/p");
        jRadioButton7.setName("jRadioButton7"); // NOI18N
        jRadioButton9.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
        jRadioButton9.setText("Voltage Off");
        jRadioButton9.setName("jRadioButton9"); // NOI18N
         javax.swing.GroupLayout jPanel1Layout = new
         javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(38, 38, 38)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
                        .addComponent(jRadioButton1)
                        .addComponent(jRadioButton2))
                    .addGap(168, 168, 168)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
TRAILING)
                        .addComponent(jRadioButton3)
                        .addComponent(jRadioButton4))
                    .addGap(180, 180, 180)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
                        .addComponent(jRadioButton6)
                        .addComponent(jRadioButton5)))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(237, 237, 237)
                    .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 353,
javax.swing.GroupLayout.PREFERRED_SIZE)))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 37,
Short.MAX_VALUE)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                    .addComponent(jRadioButton9)
                    .addGap(116, 116, 116))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
TRAILING)
```

```
                        .addComponent(jRadioButton7)
                        .addComponent(jRadioButton8))
                     .addGap(57, 57, 57))))
    );
    jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
          .addComponent(jRadioButton9)
          .addComponent(jLabel1))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
          .addGroup(jPanel1Layout.createSequentialGroup()
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
                        .addComponent(jRadioButton1)
                        .addComponent(jRadioButton3))
                        .addGap(18, 18, 18)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
                        .addComponent(jRadioButton2)
                        .addComponent(jRadioButton4)))
                         .addGroup(jPanel1Layout.createSequentialGroup()
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
                        .addComponent(jRadioButton5)
                        .addComponent(jRadioButton8))
                       .addGap(10, 10, 10)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
                        .addComponent(jRadioButton6,
javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jRadioButton7))))
              .addContainerGap(28, Short.MAX_VALUE))
       );
       jPanel3.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
       jPanel3.setName("jPanel3"); // NOI18N
       jLabel2.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N
       jLabel2.setText("Select The Relay");
       jLabel2.setName("jLabel2"); // NOI18N
       jCheckBox2.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
       jCheckBox2.setText("Relay - 2");
       jCheckBox2.setToolTipText("on/off relay : 2");
       jCheckBox2.setName("jCheckBox2"); // NOI18N
       jCheckBox3.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
       jCheckBox3.setText("Relay - 3");
       jCheckBox3.setToolTipText("on/off relay : 3");
       jCheckBox3.setName("jCheckBox3"); // NOI18N
       jCheckBox1.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
       jCheckBox1.setText("Relay - 1");
       jCheckBox1.setToolTipText("on/off relay : 1");
       jCheckBox1.setName("jCheckBox1"); // NOI18N
       jCheckBox4.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
       jCheckBox4.setText("Relay Off");
       jCheckBox4.setName("jCheckBox4"); // NOI18N
       javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);
       jPanel3.setLayout(jPanel3Layout);
```

```
        jPanel3Layout.setHorizontalGroup(
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()
                .addGap(92, 92, 92)
                .addComponent(jCheckBox1)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 113,
Short.MAX_VALUE)
                .addComponent(jCheckBox2)
                .addGap(100, 100, 100)
                .addComponent(jCheckBox3)
                .addGap(235, 235, 235))
                .addGroup(jPanel3Layout.createSequentialGroup()
                .addGap(268, 268, 268)
                .addComponent(jLabel2)
                .addGap(84, 84, 84)
                .addComponent(jCheckBox4)
                .addContainerGap(127, Short.MAX_VALUE))
        );
        jPanel3Layout.setVerticalGroup(
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
                    .addGroup(jPanel3Layout.createSequentialGroup()
                        .addContainerGap()
                        .addComponent(jLabel2))
                    .addGroup(jPanel3Layout.createSequentialGroup()
                        .addGap(29, 29, 29)
                        .addComponent(jCheckBox4)))
                .addGap(58, 58, 58)
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
```

```
                .addComponent(jCheckBox3)
                .addComponent(jCheckBox1)
                .addComponent(jCheckBox2))
            .addContainerGap(36, Short.MAX_VALUE))
      );
        javax.swing.GroupLayout jPanel2Layout = new
        javax.swing.GroupLayout(jPanel2);
      jPanel2.setLayout(jPanel2Layout);
      jPanel2Layout.setHorizontalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
            .addContainerGap()
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
TRAILING)
                .addComponent(jPanel3,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jPanel1,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addContainerGap())
      );
      jPanel2Layout.setVerticalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(31, 31, 31)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(27, 27, 27)
        .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    jMenuBar1.setName("jMenuBar1"); // NOI18N
    jMenu1.setText("File");
    jMenu1.setName("jMenu1"); // NOI18N
jMenuItem1.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.Key
Event.VK_F5, 0));
    jMenuItem1.setText("Refresh");
    jMenuItem1.setName("jMenuItem1"); // NOI18N
    jMenu1.add(jMenuItem1);
    jMenuBar1.add(jMenu1);
    jMenu2.setText("Help");
    jMenu2.setName("jMenu2"); // NOI18N
jMenuItem2.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.Key
Event.VK_A, 0));
    jMenuItem2.setText("About us..");
    jMenuItem2.setName("jMenuItem2"); // NOI18N
    jMenu2.add(jMenuItem2);
    jMenuBar1.add(jMenu2);
    setJMenuBar(jMenuBar1);
    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addContainerGap())
        );
        bindingGroup.bind();
        pack();
    }// </editor-fold>//GEN-END:initComponents
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new MainFrame().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JCheckBox jCheckBox1;
    private javax.swing.JCheckBox jCheckBox2;
    private javax.swing.JCheckBox jCheckBox3;
    private javax.swing.JCheckBox jCheckBox4;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JMenu jMenu1;
    private javax.swing.JMenu jMenu2;
    private javax.swing.JMenuBar jMenuBar1;
    private javax.swing.JMenuItem jMenuItem1;
    private javax.swing.JMenuItem jMenuItem2;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JRadioButton jRadioButton1;
    private javax.swing.JRadioButton jRadioButton2;
    private javax.swing.JRadioButton jRadioButton3;
    private javax.swing.JRadioButton jRadioButton4;
    private javax.swing.JRadioButton jRadioButton5;
    private javax.swing.JRadioButton jRadioButton6;
    private javax.swing.JRadioButton jRadioButton7;
    private javax.swing.JRadioButton jRadioButton8;
    private javax.swing.JRadioButton jRadioButton9;
    private org.jdesktop.beansbinding.BindingGroup bindingGroup;
    // End of variables declaration//GEN-END:variables
    private javax.swing.ButtonGroup btngrp;
    public void actionPerformed(ActionEvent e) {
        if(!jCheckBox3.isSelected() && !jCheckBox2.isSelected() &&
jCheckBox1.isSelected()){
            jCheckBox4.setSelected(false);

            String b="i";
             try{
            send("echo "+b+b+" > /dev/tts/1 ");
            }
            catch(Exception ex){
                ex.printStackTrace();
            }
            //System.out.println(b);
        }
        else if (!jCheckBox3.isSelected() && jCheckBox2.isSelected() &&
!jCheckBox1.isSelected()) {
            jCheckBox4.setSelected(false);

            Character b='j';
             try{
            send("echo " +b+b+ " > /dev/tts/1");
            }
            catch(Exception ex){
                ex.printStackTrace();
            }
            //System.out.println(b);
        }
        else if(!jCheckBox3.isSelected() && jCheckBox2.isSelected() &&
jCheckBox1.isSelected()){
            jCheckBox4.setSelected(false);
                      Character b='l';
             try{
           send("echo " +b+b+ " > /dev/tts/1");
            }
            catch(Exception ex){
                ex.printStackTrace();
            }
           // System.out.println(b);
        }
        else if (jCheckBox3.isSelected() && !jCheckBox2.isSelected() &&
!jCheckBox1.isSelected()) {
            jCheckBox4.setSelected(false);
           Character b='k';
             try{
            send("echo " +b+b+ " > /dev/tts/1");
            }
            catch(Exception ex){
                ex.printStackTrace();
            }
            //System.out.println(b);
        }
        else if(jCheckBox3.isSelected() && !jCheckBox2.isSelected() &&
jCheckBox1.isSelected()){
            jCheckBox4.setSelected(false);
             Character b='n';
             try{
            send("echo " +b+b+ " > /dev/tts/1");
            }
            catch(Exception ex){
                ex.printStackTrace();
            }
            //System.out.println(b);
        }
        else if (jCheckBox3.isSelected() && jCheckBox2.isSelected() &&
!jCheckBox1.isSelected()) {
            jCheckBox4.setSelected(false);
           Character b='m';
             try{
            send("echo " +b+b+ " > /dev/tts/1");
            }
            catch(Exception ex){
                ex.printStackTrace();
            }
           // System.out.println(b);
        }
        else if(jCheckBox3.isSelected() && jCheckBox2.isSelected() &&
jCheckBox1.isSelected()){
            jCheckBox4.setSelected(false);
            Character b='o';
             try{
           send("echo " +b+b+ " > /dev/tts/1");
            }
            catch(Exception ex){
                ex.printStackTrace();
            }
            //System.out.println(b);
        }
```

```
      else if(!jCheckBox3.isSelected() && !jCheckBox2.isSelected() &&
!jCheckBox1.isSelected()){
        jCheckBox4.setSelected(true);
        Character b='q';
         try{
        send("echo "+b+b+ " > /dev/tts/1");
        }
        catch(Exception ex){
           ex.printStackTrace();
        }
        //System.out.println(b);
     }
   }
   public void connect()
                        throws JSchException, IOException, Exception {
                ssh = new JSch();
            session = ssh.getSession("root","192.168.1.1",22);
                MyUserInfo ui = new MyUserInfo();
            ui.setPassword("raam");
                session.setUserInfo(ui);
                session.connect();
                channel =  session.openChannel("shell");
                PipedOutputStream po = new PipedOutputStream();
            fromServer = new PipedInputStream(po);
            channel.setOutputStream(po);
            toServer = new PipedOutputStream();
            PipedInputStream pi = new PipedInputStream(toServer);
            channel.setInputStream(pi);
                channel.connect();
   public boolean isConnected() {
                return (channel != null && channel.isConnected());
        }
```

```
          public void disconnect() {
                    if (isConnected()) {
                              channel.disconnect();
                      session.disconnect();
                    }
          }
          public void send(String command) throws IOException, JSchException,
Exception {
                    command += "; \n";
                    toServer.write(command.getBytes());
               Thread.sleep(100);
               toServer.flush();
           }
}
 class MyUserInfo implements UserInfo {
                    private String password;
                    public void setPassword(String password) {
                              this.password = "raam";
                    }
                    public String getPassphrase() {
                              return null;
                    }
                    public String getPassword() {
                              // TODO Auto-generated method stub
                              return password;
                    }
                    public boolean promptPassword(String arg0) {
                              // TODO Auto-generated method stub
                              return true;
                    }
                    public boolean promptPassphrase(String arg0) {
                              // TODO Auto-generated method stub
                              return true;
```

```
        }
        public boolean promptYesNo(String arg0) {
                // TODO Auto-generated method stub
                return true;
        }
        public void showMessage(String arg0) {
                // TODO Auto-generated method stub
                System.out.println(arg0);
        }
}
```

5. Future scope of the project

In our project as we used only port 0 and the port 2 and just one pin of the port 3 so 7 pins of the port 3 and the whole port 1 is unused so we can use this space for the future purpose .

e.g. if we put one temperature sensor at the controller side by using one of the port at the controller side which continuously send the data to the pc on the user interface programme which is created with the help of the software so it indicate the temperature continuously on it . so if the temperature rises suddenly above the critical situation so user can observe the situation and according to it he or she can handle the situation.

The intensity of light can also be varied using the voltage control so LDR, photo diode or photo transistors can be placed at the controller side that can sense the variations in the light and inform the user at the distant place about the same.

The centralized control of the building /complex electricity can be done. This can be used in the banks, hospitals, shopping malls etc.

Speed of the stepper motor can also be controlled.

6. Applications

- **In electronic laboratories**
- **To control the home appliances**
- **Temperature control**
- **Illumination control**
- **Power control**
- **Control of drives**
- **Switching devices**
- **Industrial application**
- **Where ever the controlling of voltage is required.**
- **And many more…**

7. Bibliography

- www.cisco.in
- www.jbprojects.in
- www.mypdf.in

TESTING POINTS AND TROUBLESHOOTING:

1) The output of the controller is not proper.

Check out LED near the controller whether it is on or off because it indicates that whether the controller is given the supply voltage or not. If it is off then check out the voltages at the pins of 7805 IC which supplies VCC to controller. At pin no 1 it must have minimum 6V. If it is not there then there must be problem in the power section.

If the power supply is OK and the problem still exists then reset the controller and now disconnect the jumper of MAX232 slot and put MAX232 IC in the IC socket.

Now connect the DB9 connector of PCB with serial port of PC. Using hyper terminal send the data one by one at 9600 baud rate if the problem is solved then there is some problem with GUI java based software.

If the problem is not yet solved then reload the program in the controller or replace the controller.

2) The output voltage is 0V and not varying

First checkout the transformer supply and output of transformer. Check the output of Rectifier Bridge and it must vary between 25V TO 32V. If the voltage across the filter capacitor is not so then ensure that the VCC and ground are not shorted. If it shows continuity then any one component must be damaged or there must be short circuit at the load side. So disconnect the load and find out the continuity between the output and ground pin of each regulator IC and also continuity between VCC and ground pin of controller and MAX232 IC (if connected). Replace the component if it shows the continuity. If still the problem exists then check out all soldering points.

3) The switching of the Relay is not done properly

If there is a problem regarding the switching of the relay, then replace the problematic relay. If still the problem exists then disconnect the load and then check out the problem if still problem is there than go through step 1 and 2.

Note: Check out that whether the cooling fan of the supply box is working or not if it is off then check out its power circuit and replace the damaged components and avoid the use of power supply if the cooling fan is not working because the power supply is extremely heated up and the air in the box must be circulated and hot air must be replaced by cool air to make effective use of heat sink as the heat sink dissipate the heat according to heat transfer phenomenon of the heat sink and also as to avoid damage to the components of the power supply and avoid the secondary breakdown.