EE 318 Electronics Design Project Report , EE Department , IIT Bombay , April 2006

SMS based remote control system

Ashish Deswal (03007020)
Shaleen Harlalka (03007015)
Arjun Arikeri (03007032)
Ashish Vijay (03007003)
Group B9
Under guidance of
P.C. Pandey

ABSTRACT

The aim of this project is to implement control of remote devices via means of cellular networks . The remote device can be controlled by sending appropriate command SMS which is then received at the device's end and decoded to perform the given task . This control was implemented by interfacing a nokia 3310 mobile phone with microcontroller where the microcontroller receives bytes from the nokia phone and decodes them . Two way communication is being established between phone and microcontroller via Fbus protocol . In order to ensure security of device control , every SMS command follows a password which is necessary for microcontroller to take any action . Moreover the microcontroller also sends back the device status to the user when appropriately asked for.

1 Introduction

The project consists of establishing remote control system via short messaging service of cellular network , which acts as control element . For transmission any mobile phone can be used . The receiving end consists of a nokia 3310 interfaced with microcontroller , circuitary for device control and the device being controlled . We are specifically controlling ON/OFF application of a bulb .

1.1 Mobile and microcontroller interface

Nokia 3310 phone has F-Bus connection [2] that can be used to connect the phone to a microcontroller. The connection can be used for controlling many functions of the phone, as well as uploading new firmware . This bus allows us to send and receive sms messages .

- The USART of the microcontroller needs to be synchronized with that of the phone . This is done by sending the byte '0x55' 128 times to the mobile .
- For sending the sms , the bytes are sent as per TPDU (Transmission protocol data unit) The mobile phone thus sends an acknowledgement to the microcontroller .

It again sends another acknowledgement after the message has been sent over the network . Now also the microcontroller should reply back by sending an acknowledgement .

- As the mobile phone receives an sms it mirrors the sms frame on the Tx pin which is then received by the microcontroller and decoded . As the microcontroller receives the data it must send an acknowledgement frame to the mobile phone .

1.2 Controlling the device

The device is driven by a triac circuit that acts as solid state relay . The triac circuit is driven by DC voltage input and typically 10 mA of current , which the microcontroller can provide very easily . Thus turning on/off can be accomplished by giving a logic high/low to the input of triac circuit .

2 Fbus Protocol

Nokia 3310 has F-Bus and M-Bus connections [2] . M-Bus is a one pin bi-directional bus for both transmitting and receiving data from the phone. It is slow (9600bps) and only half-duplex . Only two pins on the phone are used . One ground and one data . M-Bus runs at 9600bps, 8 data bits, odd parity, one stop bit . F-Bus is the later high-speed full-duplex bus . It uses one pin for transmitting data and one pin for receiving data plus the ground pin . It is fast 115,200bps, 8 data bits, no parity, one stop bit .

2.1 Fbus Protocol and commands

The Fbus protocol has full series of commands [2] that allows the user to send , receive and delete sms messages .

Sample frame sent to my Nokia 3310 (showed as a Hex dump)

```
Byte: 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
Data: 1E 00 0C D1 00 07 00 01 00 03 00 01 60 00 72 D5
```

This sample frame is used to get the hardware and software version from a Nokia phone. It is a good test to see if the implementation of the protocol is working .

- Byte 0 : All frames sent by cable will start with the character 0x1E first. This is the F-Bus Frame ID. Cable is 0x1E and InfraRed is 0x1C.
- Byte 1 : This is the destination address. When sending data, it's the phone's device ID byte. In our case it's always 00 for the phone.
- Byte 2 : This is the source address. When sending data, it's the PC's device ID byte. In our case it's always 0x0C (Terminal).
- Byte 3 : This is the message type or 'command'. 0xD1 is Get HW & SW version.

- Byte 4 & 5 : Byte 4 & 5 is the message length. In our case it is 7 bytes long. Byte 4 is the MSB and byte 5 is the LSB .
- Byte 6 : The data segment starts here and goes for 7 bytes in this case. As The Nokia is a 16 bit phone and therefore requires an even number of bytes. As ours is odd the last byte will be a padding byte and the message will end at location 13 .
- The last byte in the data segment (Byte 12 above) is the sequence number. The last 3 bits of this byte increments from 0 to 7 for each frame. This part needs to be sent back to the phone in the acknowledge frame .
- Bytes 14 & 15 : The second to last byte is always the odd checksum byte and the last byte is the even checksum byte. The checksum is calculated by XORing all the odd bytes and placing the result in the odd Checksum location and then XORing the even bytes and then placing the result in the even byte.

If the phone received the above frame then it replies with the following data

```
1E 0C 00 7F 00 02 D1 00 CF 71

1E 0C 00 D2 00 26 01 00 00 03 56 20 30 34 2E 34 35 0A 32 31 2D 30 36 2D
30 31 0A 4E 48 4D 2D 35 0A 28 63 29 20 4E 4D

50 2E 00 01 41 3F A4
```

The first line is an Acknowledge command frame . The destination and source addresses are now swapped . This is because the Nokia phone is now talking. This message is two bytes long with the two bytes representing the message type received (0xD1) and the sequence number (0x00) . The last two bytes are the checksum and should be checked to make sure the data is correct . The 3310 will be waiting for an acknowledge frame after these two frames were sent. If the acknowledge frame is not sent the 3310 will retry sending the data . The 3310 will only send the data 3 times and then stops .

2.2 Full sms message frame

Consider the following message frame sent to nokia (98 Bytes) .

```
Byte: 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21
Data: 1E 00 0C 02 00 59 00 01 00 01 02 00 07 91 16 14 91 09 10 F0 00 00

Byte: 22 23 24 25 26 27 28 29 30 31 32 33 34 35
Data: 00 00 15 00 00 00 33 0A 81 40 30 87 00 47

Byte: 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57
Data: 00 00 00 00 00 A7 00 00 00 00 00 00 C8 34 28 C8 66 BB 40 54 74 7ª

Byte: 58 59 60 61 62 63 64 65 66 67 68 69 70 71
Data: 0E 6A 97 E7 F3 F0 B9 0C BA 87 E7 A0 79 D9

Byte: 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93
Data: 4D 07 D1 D1 F2 77 FD 8C 06 19 5B C2 FA DC 05 1A BE DF EC 50 08 01

Byte: 94 95 96 97
Data: 43 00 7A 52
```

F-Bus Frame Header
Byte 0: F-Bus Frame ID. We are on Cable (0x1E) See Part 1.
Byte 1: Destination address.
Byte 2: Source address.
Byte 3: Message Type or 'command'. 0x02 (SMS Handling).
Byte 4 & 5: Message length. In our case it is 0x0059 bytes long or 89 bytes in decimal.
(SMS) Short Message Service Frame Header
Byte 6 to 8: Start of the SMS Frame Header. 0x00, 0x01, 0x00
Byte 9 to 11: 0x01, 0x02, 0x00 = Send SMS Message
(SMSC) Short Message Service Centre (12 Bytes)
Byte 12: SMS Centre number length. 0x07 is 7 bytes long. This includes SMSC Number Type and SMS Centre Phone Number
Byte 13: SMSC number type e.g. 0x81-unknown 0x91-international 0xa1-national
Byte 14 to 23: (Octet format) SMS Centre Phone Number .**(TPDU) Transfer Protocol Data Unit**
Byte 24: Message Type

XXXX XXX1 = SMS Submit - The short message is transmitted from the Mobile Station (MS) to the Service Centre (SC).

XXXX XXX0 = SMS Deliver - The short message is transmitted from the SC to the MS.

Byte 25: Message Reference if SMS Deliver & Validity Indicator used
Byte 26: Protocol ID. Refer to GSM 3.40 - 9.2.3.9 TP-Protocol-Identifier (TP-PID)
Byte 27: Data Coding Scheme. Refer to GSM 03.38 & GSM 3.40 - 9.2.3.10 TP-Data-Coding-Scheme (TP-DCS)
Byte 28: Message Size is 0x33 in hex or 51 bytes long in decimal. This is the size of the unpacked message.
Destination's Phone Number (12 Bytes)
Byte 29: Destination's number length.
Byte 30: Number type e.g. 0x81-unknown 0x91-international 0xa1-national
Byte 31 to 40: (Octet format) Destination's Phone Number
Validity Period (VP)
Byte 41: Validity-Period Code . Time period during which the originator considers the short message to be valid .
Byte 42 to 47: Service Centre Time Stamp . For SMS-Deliver
The SMS Message (SMS-SUBMIT)
Byte 48 to 92: This is the SMS message packed into 7 bit characters. SMS Point-to-Point Character Packing
Byte 93: Always 0x00
The F-Bus usual ending
Byte 94: Packet Sequence Number
Byte 95: Padding Byte - String is old and requires to be even .
Byte 96 & 97: Odd & even checksum bytes.

3 Circuit Design

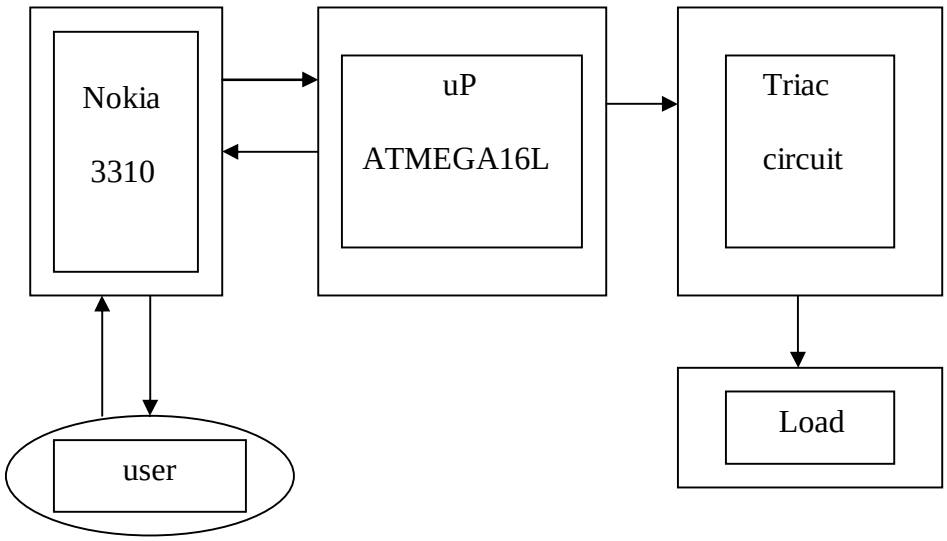
3.1 Nokia interface with microcontroller

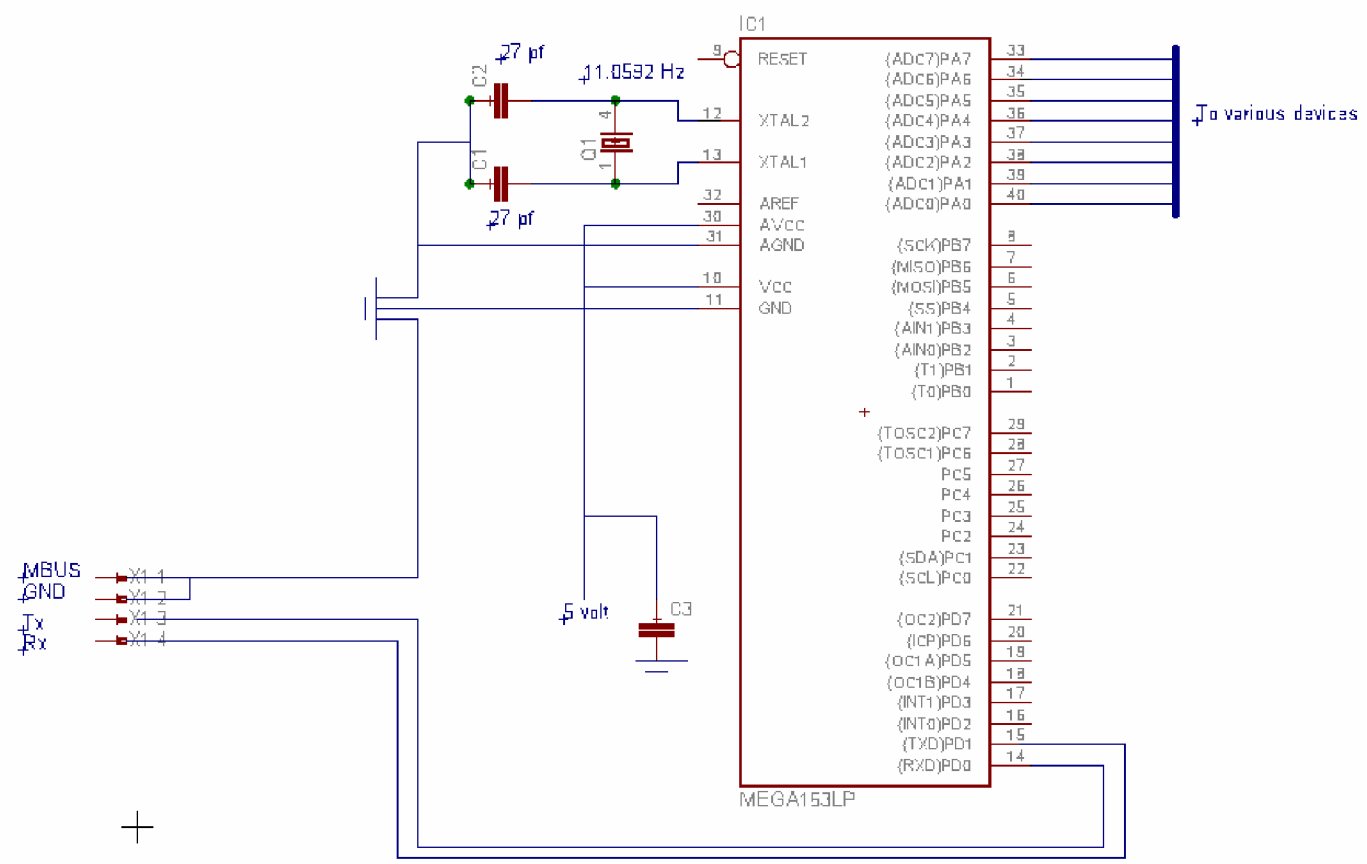
The Nokia 3310 has a logic high of 3 volts while a microcontroller has a logic high of 5 volts . Therefore a logic high of a microcontroller means logic high for the phone but the reverse is not true . Therefore some interfacing circuit for logic conversion needs to be there . But in this project , ATMEGA16L microcontroller [6] is being used which has many advanced features . One of such features has been exploited to avoid circuitry for logic level conversion . For ATMEGA16L following equation holds .

Logic high = 0.7 Vcc

While allowable range for Vcc is from 5V to 3V . Therefore mobile phone compatible logic level can be generated using this feature . Moreover F-Bus operates at baud rate 115,200 bps which is not possible for 8051 family to generate without use of external USART . Therefore ATMEGA16L is preferred . Hence we can connect directly the microcontroller to the nokia phone .

Block diagram

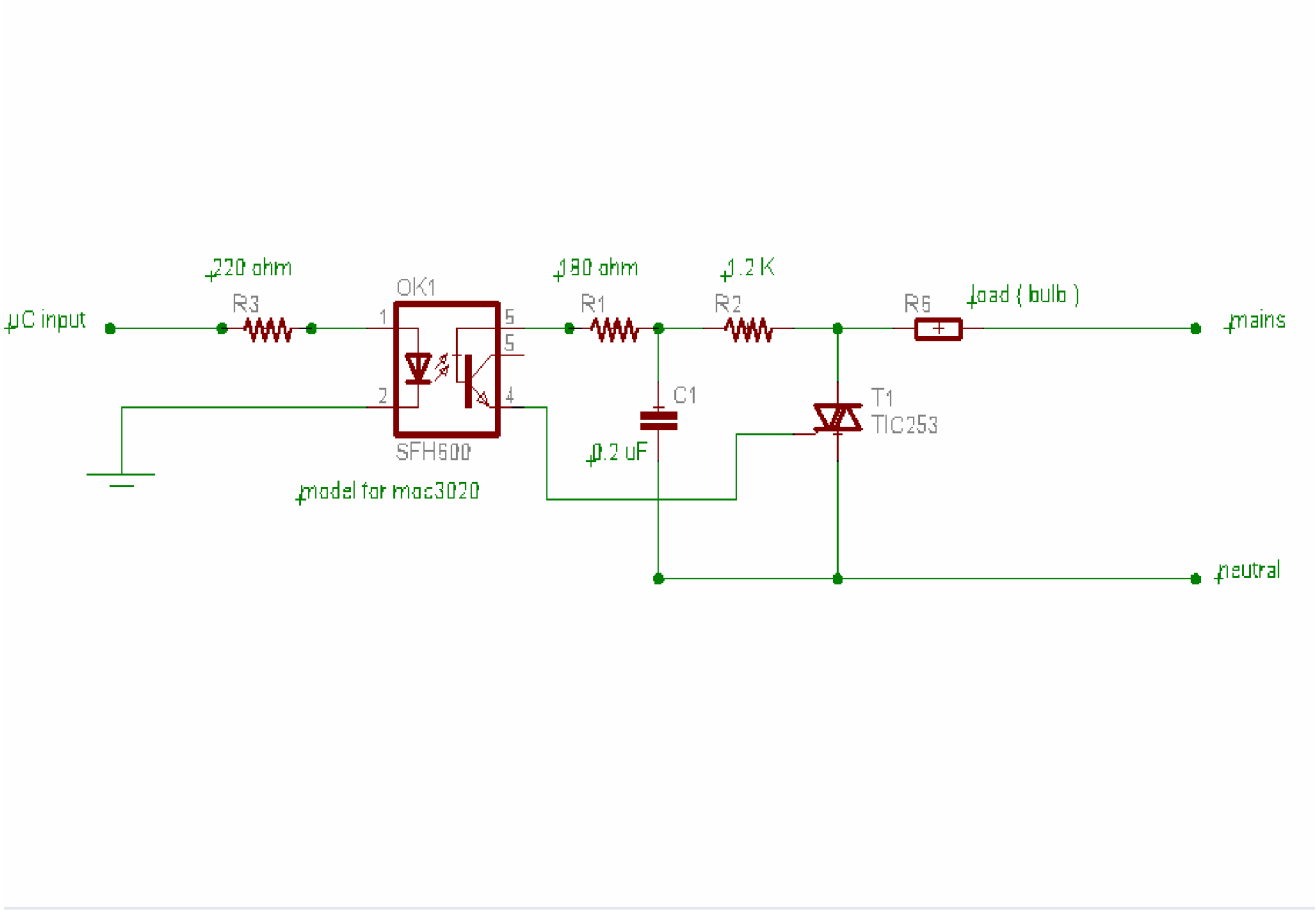




Nokia interface with microcontroller

3.2 Device interfacing circuit

The microcontroller is interfaced with the bulb via an interfacing circuit that primarily consists of an opto-coupler and a triac and needs small D.C. Input voltage and current to drive an A.C. device such as a bulb . For this purpose MOC3020 has been used [6] . It is an optically isolated triac driver device . This device contains a GaAs infrared emitting diode and light activated silicon bilateral switch which acts as a triac . As per the specifications the maximum triac driving current is 15 mA for led voltage as 3 volts . While the microcontroller ATMEGA16L can source [6] , as per specifications , a maximum current of 40 mA if single pin is sourcing , and a maximum of about 20 mA per pin if all of them are sourcing current . This eliminates the requirement of a triac driver circuit and microcontroller can source the required current to the triac circuit . Resistance of 220 ohms is chosen between microcontroller output pin and MOC3020 input so that current of about 10 mA flows which is quite on safe side .



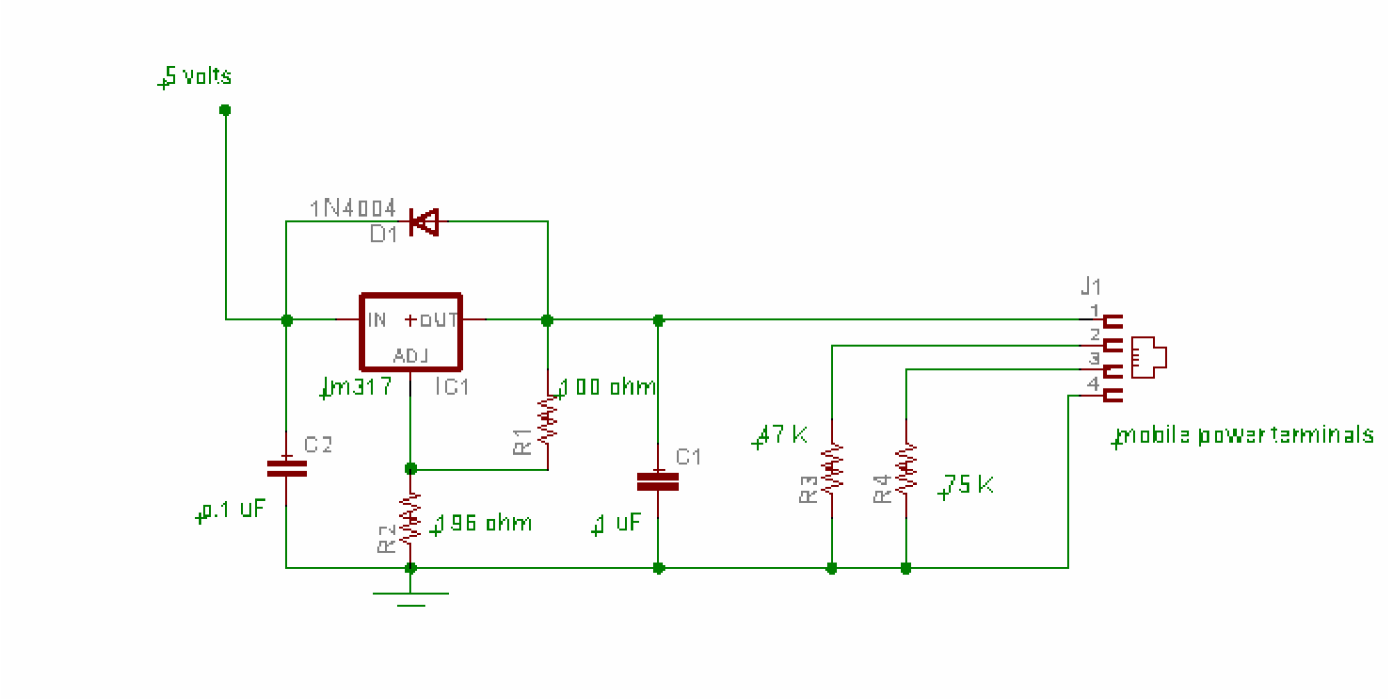
Triac circuit

3.3 Power supply for mobile phone

The power supply for the mobile phone was designed keeping in mind the battery specifications . The mobile phone has four power terminals out of which two are Vcc and ground . Other two monitor battery temperature and battery type respectively . We found out the equivalent resistances these terminals see and connected them with corresponding resistance values to ground . LM317 has been used as regulator [6] . Calculations for resistances R1 and R2 have been done so as the output is 3.6 volts as per battery specification , and for typical adjustable pin current .

Vout = 1.25(R2/R1 + 1) + Iadj*(R2)

R1 = 100 ohms
R2 = 196 oms



Regulated mobile phone power supply

3.4 Problem faced

After lot of testing , observations and discussions with professors and lab in-charges we came to conclusion that the Tx pin of the nokia 3310 serial port is damaged and getting another cell phone was not feasible . As soon as the phone gets an sms , it mirrors it on the Tx pin . We could not get the pulses on the DSO . We also tried sending a particular byte sequence to the mobile phone , that asks for its software version and still we did not get any response . We could not check the working of the nokia cell phone with already available softwares on computer as it required a special data cable which doesn't come nowadays in market since the nokia 3310 is very outdated . We confirmed this with virtually every nokia shop . As we reported our problems , an alternative was suggested that we control the device via any serial port . So PC serial port has been used for control of the device which has replaced nokia cell phone in the project .

3.5 Controlling device via PC serial port

A program in C language has been developed which transmits and receives characters via PC serial port . For interfacing this serial port with the microcontroller additional circuitry is required for level conversions . Since PC serial port uses RS232 protocol [3] which requires +3 to +25 volts as logic 0 and -3 to -25 volts as logic high . D9 connectors have been used and the logic level conversion is done via MAX232 [6] transceiver .The configuration used is null-modem configuration [3] . It only requires three wires to connect to transceiver .

D9 pins

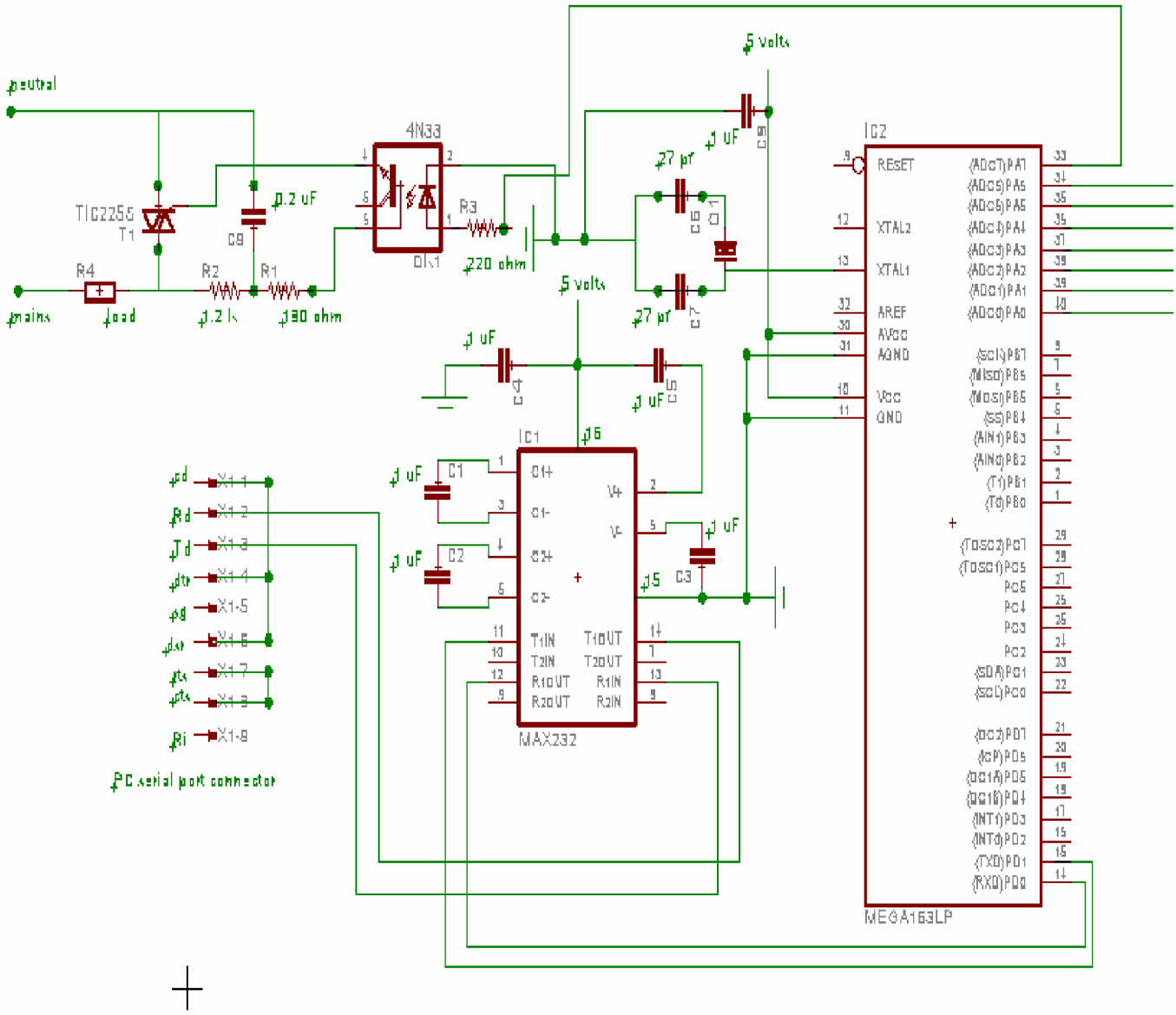
Pin 1	Carrier detect
Pin 2	Receive Data
Pin 3	Transmit Data
Pin 4	Data Terminal Ready
Pin 5	Signal Ground
Pin 6	Data Set Ready
Pin 7	Request To Send
Pin 8	Clear To Send
Pin 9	Ring Indicator

For null modem connections pins 1 , 4 , 6 are shorted while pins 7 , 8 are shorted . Only three wires are used for connections namely pins 2 , 3 and 5 .
The command sent to the microcontroller follows a 5-character password . All other commands will be executed only if this password is accepted . Individual pin control of port A is enabled . Status of port a can be checked by sending a command that demands the status .

MAX 232 observations

Logic level 1 = -8.91 volts
Logic level 0 = 8.95 volts

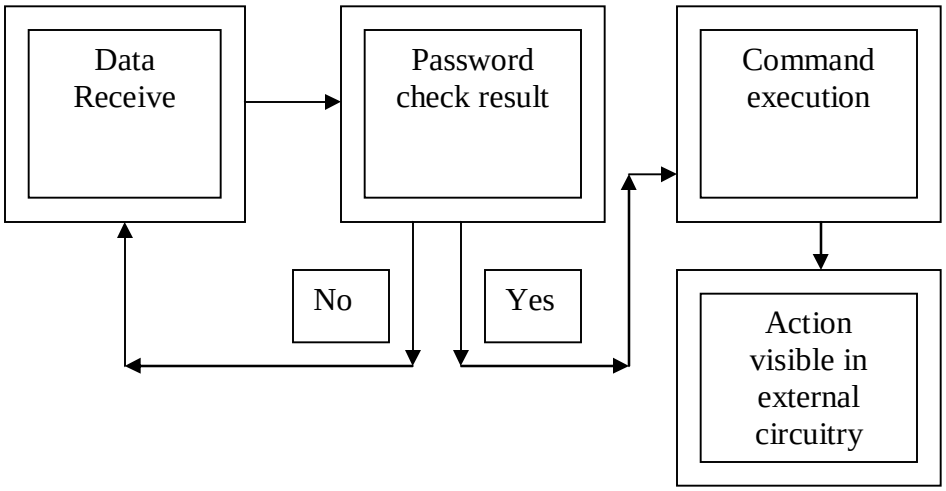
Therefore a full working circuit is implemented that turns on/off a bulb securely via PC serial port .



Controlling device via PC serial port

4 Program Flow

Below is shown the flowchart for the microcontroller software that decodes the incoming data and acts accordingly .



5 Observations

Following observations have been made during the course of the project .

1. The nokia 3310 cell phone requires operating voltage of 3.6 volts . It draws a maximum current of about 0.3 A while during sms receiving the current drawn remains under 0.1 A .
2. The driving current drawn by the triac circuit is about 10 mA while the microcontroller is capable of providing more that that .
3. The MAX 232 logic level 1 is -8.91 volts while logic level 0 is 8.95 volts .
4. The optocoupler led voltage is about 3 volts while microcontroller input to the optocoupler is 4.98 volts .

6 Result discussion

We have implemented a circuit that is controlled by a computer . More specifically an external device can be controlled via serial port of the computer . This gives way to possibility to control the device remotely by connecting to the computer through LAN and executing the control program .
The prime objective of the project could not be achieved due to an unavoidable problem . But remotely controlling a system via cellular network gives way to a lot of possibilities .

- An sms based GPS car tracking system can be designed . By sending appropriate sms command , it is possible that the car sends back its position after regular intervals by sms .
- Sms based remote control systems can also be useful for security . appropriate sms can be sent by the alarm when it turns on to some security official who is not nearby the protected area .

7 User Manual

Simultaneously 8 devices can be controlled as individual pin control of a 8 bit port of the microcontroller is enabled . Port A is being controlled . Given below is a set of commands to do the same .

Turn on

a1	turn on pin 1
b1	turn on pin 2
c1	turn on pin 3
d1	turn on pin 4
e1	turn on pin 5
f1	turn on pin 6
g1	turn on pin 7
h1	turn on pin 8

Turn off

a0	turn off pin 1
b0	turn off pin 2
c0	turn off pin 3
d0	turn off pin 4
e0	turn off pin 5
f0	turn off pin 6
g0	turn off pin 7
h0	turn off pin 8

Status

s	reads back status of port A

8 Component specifications

ATMEGA16L	Four port , 8 bit microcontroller with 16K bytes and in-system programmable flash
MAX232	Dual driver/receiver transceiver chip for logic level conversion
LM317	3-terminal adjustable regulator
MOC3020	optically isolated triac driver device . for 115/240 VAC operations
BTA 106	Triac for on/off applications
LN4004	Si Rectifiers 1Amp , 400V
D9 FEMALE	Connector for connecting to pc serial port
LOW WATTAGE BULB	As device to be controlled

9 References

[1] Paul Horowitz , " The Art of Electronics " .

[2] http://www.embedtronics.com/nokia/fbus.html

[3] Datasheet , "Interfacing the serial/RS232 port v5.0" .

[4] http://www.auditmypc.com/acronym/TPDU.asp

[5] Linux Gnokii Project – A linux based program to send sms from linux source via Fbus protocol .

[6] Datasheets for ATMEGA16L , MOC3020 , LM317 , MAX232 .