EE389 Electronic Design Lab-II (EDL-II) Project Report.EE Dept, IIT Bombay, Nov. 2006.

RF LINKED HAND-HELD VOTING AND CONFERENCE TERMINAL

Group No. D3

Nilesh Kalantri N (03D07018) <nik@ee.iitb.ac.in>
Hitesh Das (03D07030) <hdas@ee.iitb.ac.in>
Kartik G. (03D07032) <gkartik@ee.iitb.ac.in>

Supervisors: P.C. Pandey, L.R. Subramanyan and V.K. Tandon

Abstract

This report discusses the approach, design and implementation of a circuit to facilitate voting and conferencing through RF link. The work involves building such a device based on microcontroller and an RF transceiver. The central unit communicates with each of the terminals which are recognized by a hardware identification number. A half duplex asynchronous RS-232 serial communication is used to connect the microcontroller with the RF transceiver. The terminal executes various functions like voting, registering the time stamp of the vote, receiving the speak command etc. The report discusses both the hardware and software implementation of the circuit. All the functions have been verified and tested on a printed circuit board.

1. Introduction

Handheld terminals are used in game shows and conferences. Their main purpose is to register and time stamp the vote, during game shows and then send it to the central unit. During talk shows and conferences, it must be able to receive the 'speak' and the 'interrupt' command and send the central unit appropriate information about its status.

Each terminal is associated with a Hardware Identification Number (HIN), so that the central unit can send and receive data to a particular terminal independently.

2. Design Approach

The designed circuit should be able to receive the commands from the RF transceiver module and analyze it to carry out specific operation pertaining to that command. It should be able to send required data back to the central unit. The terminal must also have a display, to let the user know about the command that has arrived. The following block diagram explains the basic design of the module.

When a command is issued from the central unit, the RF transceiver receives the data and then sends it to the microcontroller over the RS-232 link. The microcontroller then decodes the command word and waits for the HIN. If the HIN matches with that of the unit, then it displays the appropriate action to be taken by the user on the display terminal of the handheld unit. If the HIN does not match, then it waits for the next instruction to come. The unit also sends the status information to the display of the central unit.

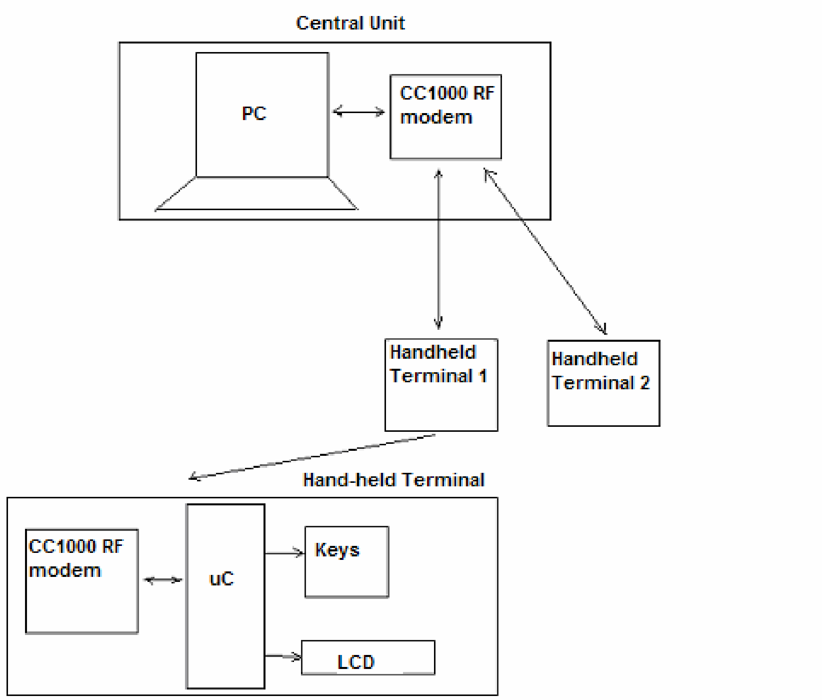

Figure 1: Block Diagram

3. Circuit Design

3.1 Hardware Identification

Each unit must have a Hardware Identification Number (HIN) for the central unit to recognize it independently. This can be implemented by a 16-pin DIP switch which is a part of the terminal.

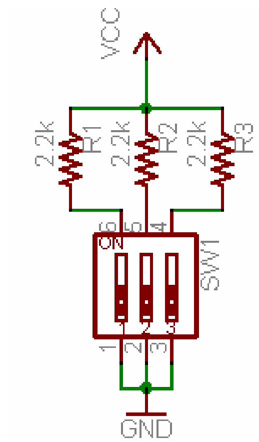

Figure 2: DIP switch.

3.2 Keys

Keys are used for voting purpose and to signal 'end of speech'. A set of 4 microswitches have been used for this purpose. Microswitches have four pins of which two pairs are short circuited. A pin from one pair of the keys is connected to the microcontroller and a pin from the second pair is connected to the ground. The four switches are used to signify the voting option 'A', 'B', 'C' and 'D'.

3.3 Display Interface on the Terminal

The commands issued by the central unit can be viewed on the 16x 2 LCD display. The microcontroller is used to drive the control and data lines of the LCD.

11 I/O pins (8 Data lines and 3 Control lines) of the microcontroller are required. The control lines are Enable (E), Register Select (RS) and Read or Write (R/W).

'E' is used to latch the data present on the data pins. A high-to-low edge is needed to latch the data.

'RS' = 0 -> Command Register is selected else Data Register is selected.

'R/W' = 0 -> Write operation else Read operation.

Writing to LCD: Before writing to the LCD (either to data register or command register) we need to check if the LCD is busy or not. To do this we give a pulse on 'E' line and wait for the D7 (MSB of data line) to go 'High'. Now the LCD's registers are written with the data on its data lines.

Initialization of LCD: The LCD is initialized to a 16x2 matrix by writing 0x3C to command register. The LCD and cursor are turned on, the cursor is shifted right and the screen is cleared by writing 0x0E, 0x06 and 0x01 to the command register respectively.

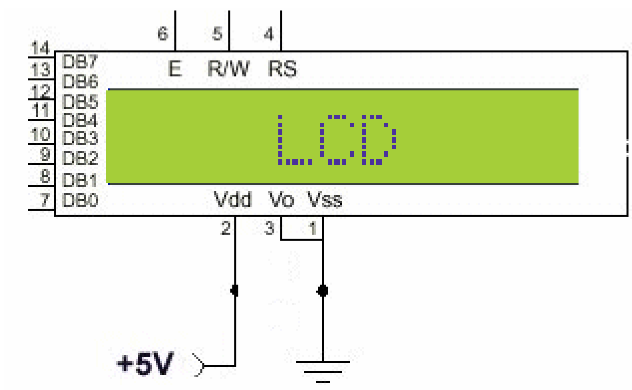

Figure 3: LCD display

3.4 Microcontroller

We needed 11 I/O pins for interfacing the LCD display and 5 pins for keys and 3 pins for setting Hardware Identification Number (HIN). Since the program memory was not a concern, we used AT89C51 microcontroller which has 4 I/O ports with 2K program memory and 128 bytes of RAM.

Port 1: Port 1 of the microcontroller was used for interfacing with the data lines of the LCD display.

Pins P3.5- P3.7: Pins P3.5, P3.6, P3.7 were used to drive the control lines 'RS', 'R/W' and 'E' respectively of the LCD display.

Pins P3.0 and P3.1: The pins P3.0 and P3.1 have special functions of RXD and TXD respectively and hence were used for serial data communication by interfacing with Max-232 chip.

Port P0.3- P0.7: Pins P0.3 through P0.7 were used to interface with switches for keys and ‘end of speech’ with external pull-up as these pins are used as inputs and Port0 requires external pull-up.

Pins P2.0- P2.2: Pins P2.0- P2.2 were used to interface the DIP switches which were used set the Hardware Identification Number (HIN) for the terminal.

Crystal: An 11.0592 MHz crystal oscillator has been used to generate the clock pulses required for the operation of the microcontroller. This frequency is selected because it provides standard baud rates used for serial communication with a computer.

Reset: Reset circuitry has been connected using a 5.2 KΩ resistor and a 10 uF electrolytic capacitor.

3.5 Serial Data Communication

The terminal unit has to constantly transmit and receive data from the RF module. As a result, UART of the microcontroller is being used for the serial communication of data. The TXD (transmit data) and RXD (receive data) pins of AT89c51 are used for this purpose. These pins are compatible with TTL logic levels. RS-232 logic levels use +3 to +25 volts to signify a "Space" (Logic 0) and -3 to -25 volts for a "Mark" (logic 1). A MAX 232 has to be used to convert the TTL logic levels to RS232 logic levels. This IC requires a 5V power source which is same as that of microcontroller. An external charge pump is required which is implemented using 1uf capacitors. The transmission lines of MAX 232 are connected to a DB-9 male pin.

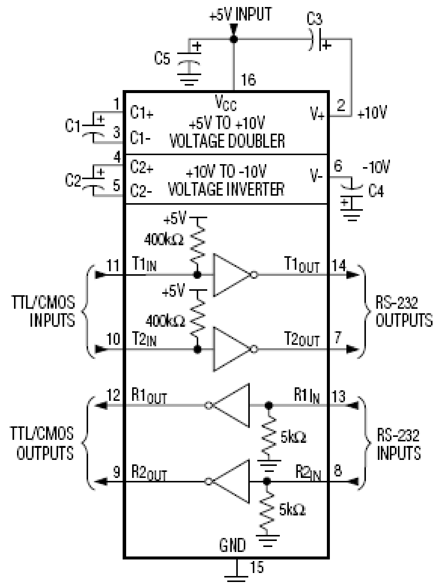

Figure 4: MAX 232

3.6 RF Transceiver

The transmission and reception of data over the RF link is done using a CC1000 modem. The module consists of a CC1000 transceiver chip and monopole antenna. The module can communicate to PC and other RS-232 enabled devices through serial data communication. In transmit mode, the modem reads data from the RS-232 serial interface, creates packets containing the data, and transmits these packets over the RF link. In receive mode, it looks for valid packets. When one is received, the data contained within the packet is extracted and sent to the attached RS-232 device or PC. Since, in the RF link data cannot travel in both directions, some form of data handshaking is required. The RF modem is designed to use hardware handshaking to tell the PC when it is able to receive data and when it is not. The module requires a power supply of 5-9 V.

3.6.1 Configuration of modem

The RF modem can be configured by holding down button 2 (S3) Figure 5, when power to the modem is switched on or when the modem is restarted by pressing the reset button (S1). If the modem is connected to a PC running a suitable terminal-emulation program such as the HyperTerminal program, the user sees the Configuration menu similar to the one shown below. The menu gives the user access to a set of configuration and test features and enables the user to change the data rate and various RF parameters. Communication between the PC and the RF modem occurs at 57600 baud, 8 data-bits, 1 stop-bit, and no parity.

The RF Modem Configuration Menu is as follows:

D - Dump CC1000 registers
E - Set RF data rate
F - Default settings
C - Calibrate
R - RX mode
T - TX mode
L - PLL lock
B - Button configuration
O - Load configuration data from SmartRF Studio
X - Exit from configuration menu.

The data rate is specified to indicate the rate of data transfer over the RF link. The data rate can go up to 19.2 kbits/s, but we use a comparatively lesser data rate of 4.8 kbits/sec to minimize data loss over RF link. 'Default Settings' are used to select the frequency over which data is to be transferred. For our operation we use a frequency of 433 MHz. The 'calibrate option' is given to check whether the PLLwithin the CC1000 is in lock. The modem can be tested by connecting them to the serial port of the PC and then configuring them at the desired settings. After exiting from the Configuration Menu, if we press the buttons S1 or S2, the characters 'Y' or 'Z" should appear respectively on the other PC's HyperTerminal display.

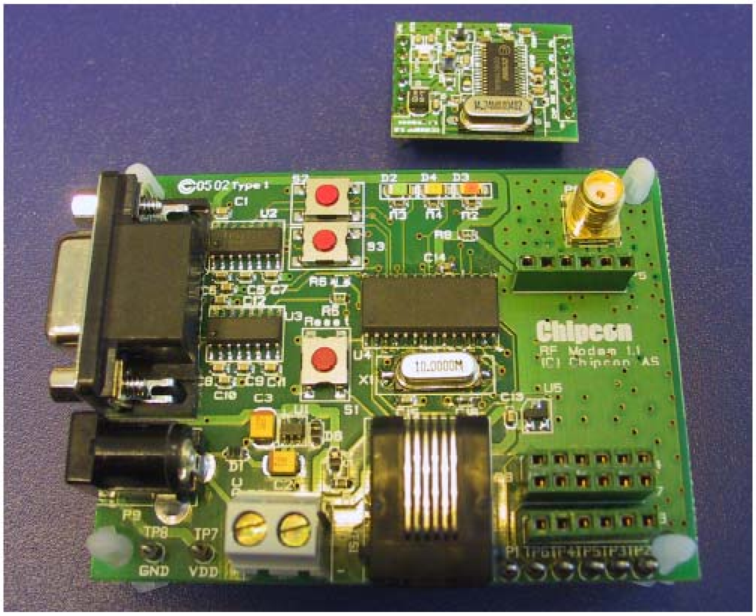

Figure 5: CC1000 RF Modem

3.7 Regulated Power Supply

LM7805 was used for supplying 5 V regulated power required for the circuit. The input is from a 9V portable battery.

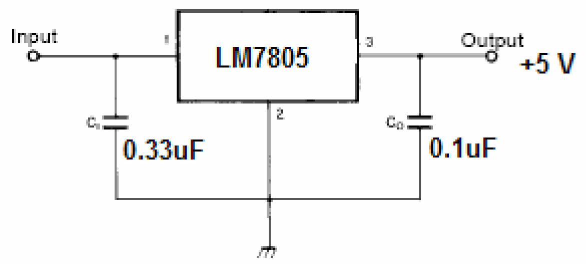

Figure 6: Application Circuit of LM7805

4. Another Design Approach

The second approach involves interfacing the CC1000 Plug and Play (CC1000PP) module directly with our microcontroller.

4.1 Hardware Identification

This was implemented using DIP switches as mentioned earlier.

4.2 Keys

Four micro-switches were used as keys for voting purpose.

4.3 Display

LEDs were used to signal the mode of operation i.e. one LED each for voting and conferencing.

4.4 Microcontroller

Since we needed around 13K of program memory for initialization and configuration of RF module, a microcontroller with large program memory was needed. Also, the microcontroller needed to be operated at 3V for direct interface with the CC1000PP module. Hence, ATMEGA32L microcontroller was chosen for use.

Port A.0-A.3- These four pins are provided with pull-ups and keys for voting are interfaced.

Port C.0 and C.1- LEDs are connected to these pins for display purpose.

Port D: Five pins of PortD are used to interface the CC1000PP module with the microcontroller.

4.5 RF Transceiver

Chipcon has designed the CC1000PP module to enable very quick prototyping of an RF system using CC1000, which is a true ultra-low-power single-chip RF transceiver for e.g. the 315, 433, 868, 915 MHz frequencies. It has been specifically designed to comply with the most stringent demands of the low power radio market.

The chip is configured using a three-wire bus, comprising of PCLK, PDATA and PALE signals. Data interfacing is done via the DCLK and DIO pins. The pin used to interface with DCLK should be able to generate an interrupt on signal edges. Using general-purpose I/O pins to handle an interface in this way is often called “bit banging”. This approach is very flexible, as the user is free to use any I/O pins on the microcontroller, but can be slow when working with a slow microcontroller.

The RF link is half duplex and the transceiver is either in a transmit or receive mode. The configuration procedure of CC1000 requires that all registers which need to be configured should be programmed after reset. The configuration is verified by reading the written registers. In TX mode, the interrupt should be triggered on the falling edge of DCLK. When the interrupt occurs, write the next bit to be transmitted to the I/O pin. In RX mode, the interrupt should be triggered on the rising edge of DCLK. When the interrupt occurs, read the data from the I/O pin. The ATMEGA32L microcontroller has an advantage that it can generate an interrupt on either a rising or a falling edge depending on the settings in its MCUCR register.

While transmitting data, a preamble of 4 bytes containing alternate ones and zeros was used. A Start-Of- Frame (SOF) byte (0x85) was also sent after the preamble. After this byte is received, the receiver gets byte synchronized with the transmitter and valid data can then be received.

The receiver sensitivity depends on the data rate, the data format, FSK frequency separation and the RF frequency. Hence if we use a higher baud rate, we’ll have compromise on the receiver sensitivity. This compelled us to send the data packets repeatedly for about 25 times to ensure that the same was received by the receiver at 4.8K bits/s baud rate.

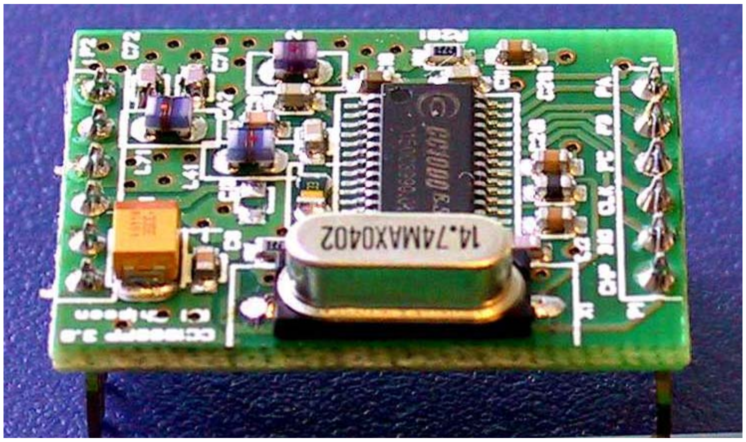

Figure 7: CC1000 Plug and Play module

4.6 Serial Port Interface

The central unit has to constantly transmit and receive data from the PC. As a result, UART of the microcontroller is being used for the serial communication of data. A MAX 232 has to be used to convert the TTL logic levels to RS232 logic levels. However the output of MAX 232 is 5V digital logic. Hence, another level converter 74HC244 is used after the output of MAX 232. This chip converts 5V logic to 3V logic and vice versa.

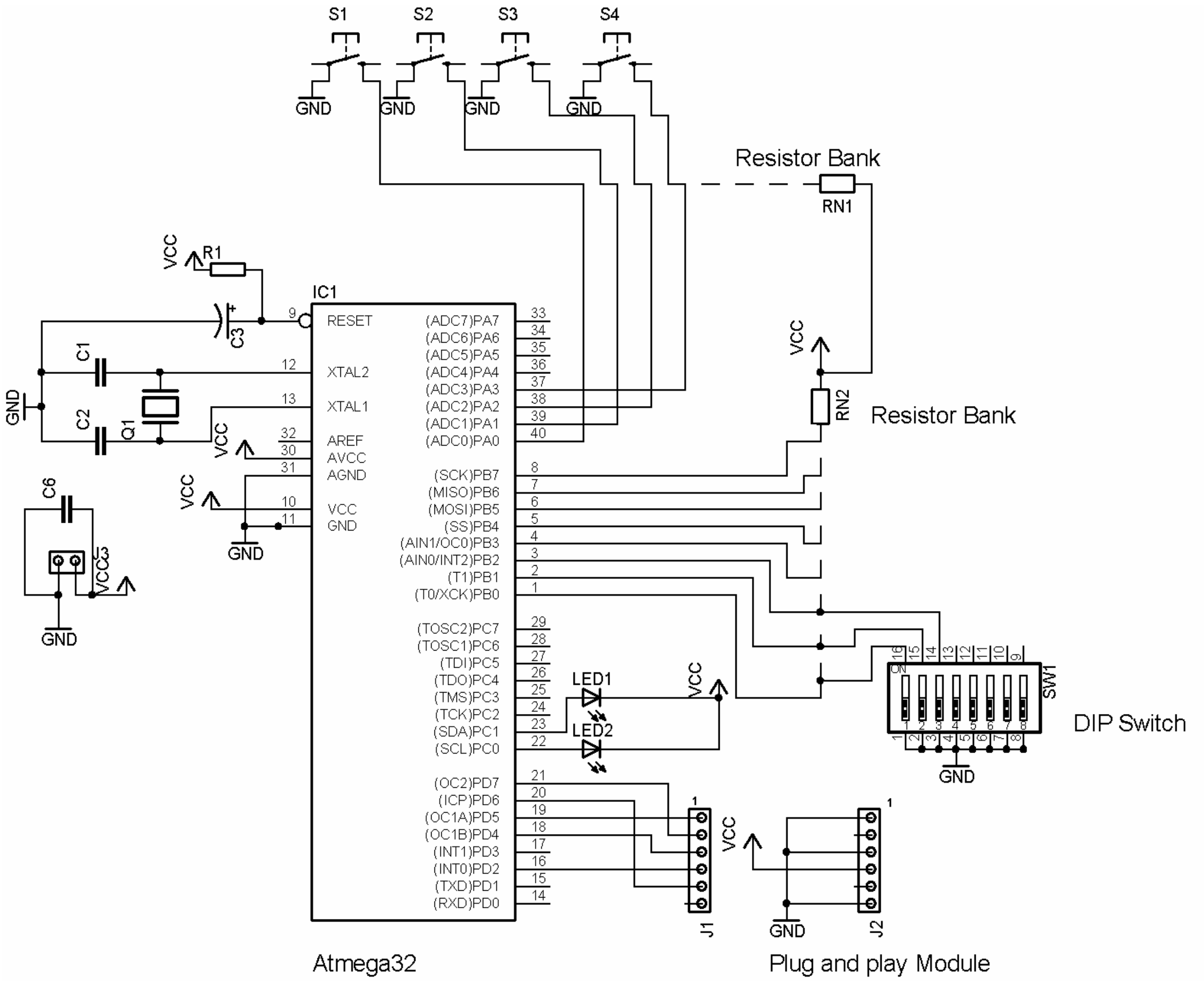

Figure 8: Schematic of the second design.

5. Implementation

The central unit which actually is an RF module connected to the PC, issues commands like ‘vote’ or ‘speak’ to the terminal unit. The RF module at the handheld unit receives the command word and sends it to the microcontroller using the serial data interface. The system implemented here is half duplex communication. The standard UART mode (Mode 1) of 89c51 has been used. Timer 1 is used to generate the baud rate of 57600 baud for the communication. The content of TH1 is obtained using the following equation:

$$\text{Baud rate} = (2^{*}\text{SMOD} * \text{oscillator frequency}) / (32d * 12d * [256d - \text{TH1}]) \qquad (1)$$

SMOD is a control bit in the register PCON and is 1 in our case which provides a TH1 value of 0FFh. Timer 1 is operated in an auto reload mode 2 configuration. The timer configurations are done using the timer control (TCON) and timer mode control (TMOD) registers. The register TCON is bit addressable and hence the timer can be started by a single bit (TR1).

The RI bit of the SCON register is set when data byte is received and is stored in the SBUF register. The microcontroller decodes the command word and then displays it on the LCD and the appropriate action is taken. Schematic of the circuit is as shown in Figure 9.

Transmission begins when data is put in the SBUF register. The TI bit of the SCON register is set when data byte has been transferred through the TXD pin and the transmission is complete. When TI flag is set, next data byte can be put into SBUF for transmission. The Central Unit displays the message sent by the terminal on the PC monitor.

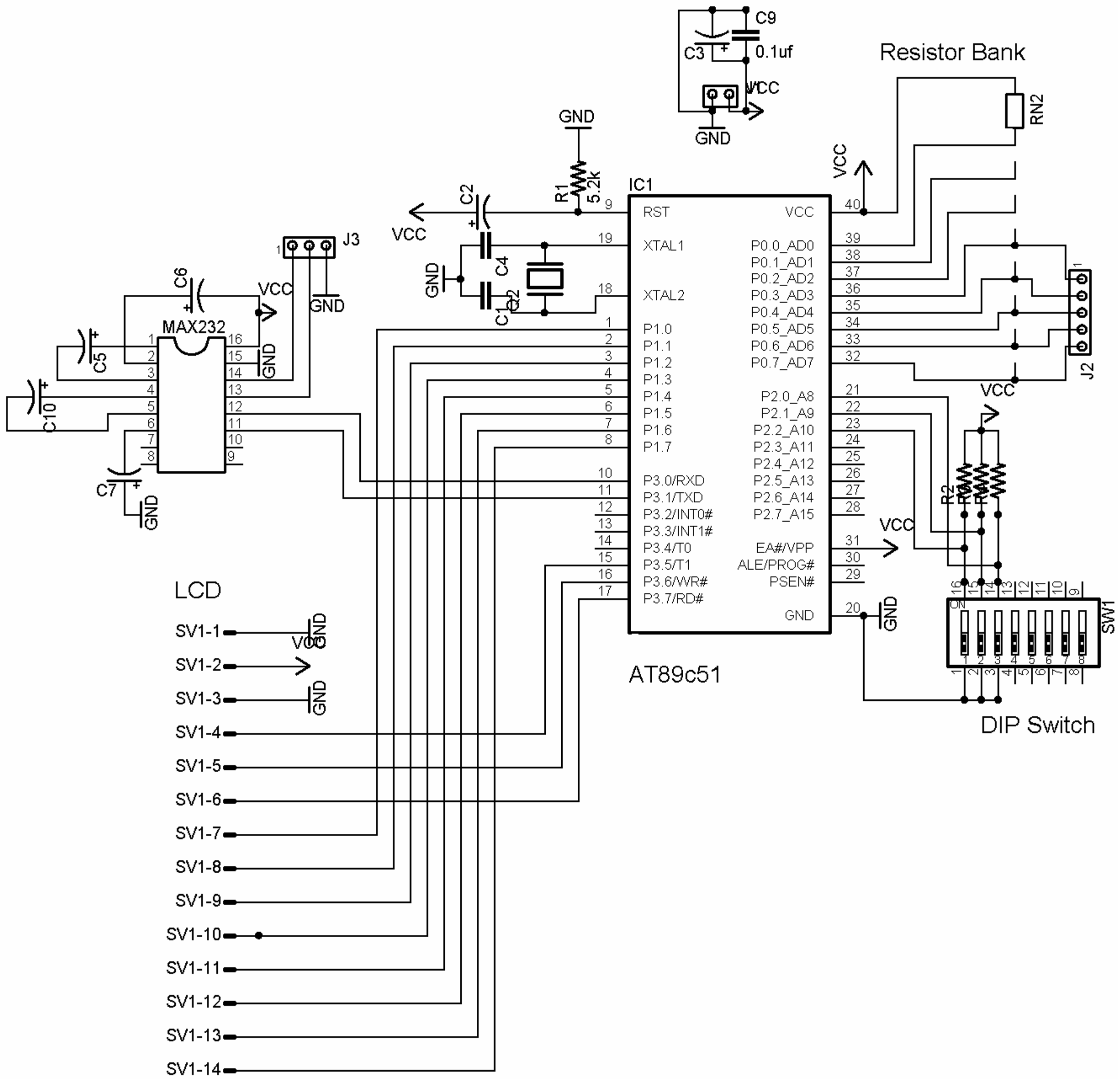

Figure 9: Schematic of the Circuit

The commands issued by the central unit are as follows:

Vote

If the vote command is issued then "VOTE" is displayed on the LCD and the microcontroller then polls for the 'Vote Option' key to be pressed. Timer 0 is used to calculate the time stamp for the vote cast. We use the Timer 0 as a 16 bit counter in mode1. The oscillator frequency of 11.0592 MHz is internally divided by 12 and is fed to the lower 8 bit counter (TL0), which in turn is cascaded to the upper 8 bit counter (TH0). Thus the input clock frequency is now divided by (12 * 256 * 256). Thus the timer flag TF0 is set to 1 after every 71 ms. The timer registers are initialized to 0000h.Thus taking into consideration the above requirements we obtain a value of 21h for TMOD register and 50h for the SCON register. After a key is pressed, "END" message is displayed on the LCD and the vote cast and time stamp is stored in the microcontroller.

Send

On receiving this command, the microcontroller waits for the HIN. After verifying that second data byte is its HIN, it sends the vote cast and time stamp to the central unit, which is then displayed on the HyperTerminal display.

Speak

When the microcontroller receives this command and corresponding HIN, "SPEAK" is displayed on the LCD. The terminal is asked to speak till the user presses the 'end of speech' key or an interrupt command is received from the central unit.

Interrupt

A "STOP" message is displayed on the LCD on the reception of the interrupt command.

Acknowledge

If a terminal makes a request to speak, then the central unit a gives a command to acknowledge the request and a acknowledge message is displayed on the LCD.

The following table lists the various commands issued by the central unit and the actions taken by the microcontroller.

Table 1

Command	Action to be Taken	Message on HyperTerminal	Message on LCD
V	Request for all terminals to vote.	Terminal is voting.	VOTE
SX	Request to send vote cast and time stamp.	Vote cast and time stamp is displayed.	END
CX	Request to speak	Terminal 'X' is speaking.	SPEAK
IX	Interrupt	Terminal 'X' has been interrupted.	STOP
AX	Acknowledge request to speak	Request to speak is acknowledged	REQ. ACK

5. Algorithm

The program flow can be represented by the following algorithm:

1. The Terminal initially reads its Hardware Identification Number (HIN) from the DIP switches.
2. The LCD display is initialized and the baud rate is set to 57600bits/sec using Timer1.
3. The terminal now keeps waiting for a command from the Central Unit.
4. When a command is received, if the command is 'S' or 'C' or 'I' or 'A', it keeps waiting for the next byte.
5. If the command is 'V', terminal enters the voting mode sends a message to the central unit saying that the terminal is voting and Timer0 is started in order to record the time stamp.Timer0 is kept running until a voting key is pressed. Once a key is pressed, the time is decoded into seconds and stored. A flag- 'v_cast' is set and goes to step 3
6. If the command is 'S', 'C', 'A' and the next byte is its Hardware Identification Number 'X' (i.e. if the command is meant for it) it enters in conference mode if the command is 'C' else if it is 'S' and if 'v_cast' is set, it sends the vote and the timestamp

to the central unit, ‘v_cast’ is reset and terminal goes to step 3. An acknowledgement message is shown on LCD if command is ‘A’.

7. If it enters the conference mode, a count up timer is displayed on the LCD to show the total time of speech. It keeps waiting for the ‘end of speech’ key to be pressed or for the central unit to send ‘I’ followed by its HIN. When this happens, it comes out of conference mode and goes to step 3.

8. If the ‘request to speak’ key is pressed on the terminal, then a request message is displayed on the central unit. A ‘request sent’ message is displayed on the LCD in the terminal.

6. PCB layout

The schematic of the circuit was made in EAGLE and .brd file was generated. Sharp turns on the tracks were removed in order to ensure proper propagation of signals. Figures 10 and 11 show the PCB layout using CC1000DK Development Kit and CC1000 Plug and Play module respectively.

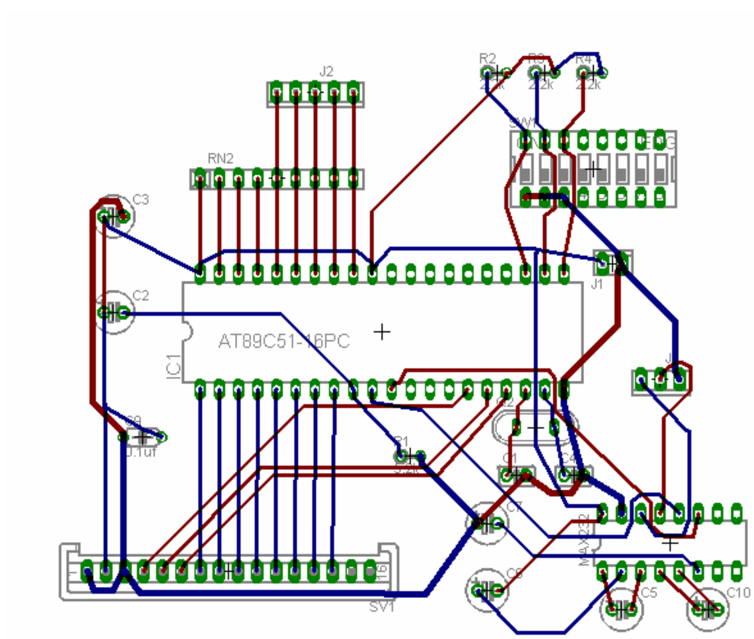

Figure 10: PCB layout of the design using CC1000DK module.

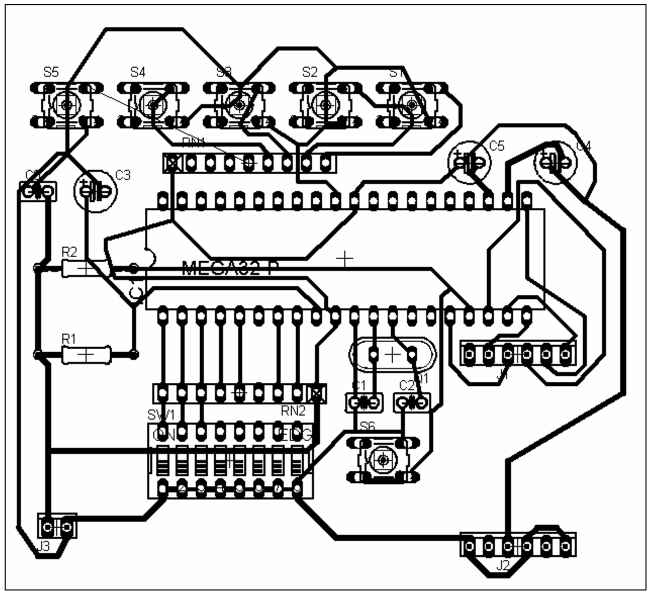

Figure 11: PCB layout of the design using CC1000PP module.

7. Conclusions

Initially we configured the RF modules and connected them to two PC terminals and bidirectional data transfer through RF link was successfully achieved. Then one of the PC terminals was replaced by microcontroller and the interfacing of the RF module was tested using data transfer to and from a PC terminal. The aforementioned algorithm was then implemented and tested successfully. Multiple terminals were built and tested.
The design using CC1000 Plug and Play module has been implemented with two independent units and voting function has been successfully tested on a printed circuit board.

8. References

[1] K. J. Ayala, *8051 Microcontroller, Architecture, Programming & Applications*, Second Edition, Penram International Publishing (Mumbai), 2000.
[2] Atmel Corporation, "8-bit Microcontroller with 4K Bytes Flash, AT89C51", www.atmel.com/dyn/resources/prod_documents/doc0312.pdf, Last visited 9th Nov. 2006.

[3] Atmel Corporation, "8-bit Microcontroller with 32K Bytes Flash, ATmega32", www.atmel.com/dyn/resources/prod_documents/doc2503.pdf, Last visited 9th Nov. 2006.
[4] Dallas Corporation, "+5V-Powered, Multichannel RS-232 Drivers /Receivers", http://www.datasheetcatalog.com/datasheets_pdf/M/A/X/2/ MAX232.shtml, Last visited 9th Nov. 2006.
[5] Chipcon Corporation, "CC1000 Datasheet and Development Kit User's Manual", www.chipcon.com/files/CC1000DK_User_Manual_2_11.pdf, Last visited 9th Nov. 2006.
[6] PCB Layout Guidelines (Online), http://www.ee.iitb.ac.in/~pcpandey/notes/pcb/ pcblayout_c_nov03.ppt.