EE389 EDL Report, EE Deptt., IIT Bombay, Nov. 2004

RF linked Handheld Terminal

Group No.: D14
Shakti Prakash Chittara 01d07022, Dheeraj Sarwaiya 01d07021

Supervisor: Prof. P.C.Pandey

ABSTRACT

There is often a need to transfer data from some hand held portable device to a remote computer. The device should be able to transmit as well as receive data. In another application such device can be configured to take data from a RF tag and send it to the central system. In present case we are using a keypad to feed in data instead of a RF tag and the data transmitted and received will be displayed on LCD screen. RF interfacing is achieved through serial port based Chipcon transmitter and receiver with a monopole antenna for connecting over 10m one computer to another computer.

1. INTRODUCTION

The main objective of the project is to design a RF based hand held device, which can be used to transfer information to a remote receiver or a computer. The hand held device can also receive information from the remote computer or transmitter. The information sent and received can be displayed on a 16 (characters) x 1(line) LCD display attached with the device. To give input to the device, a 4x4 keypad is attached.
The transmission and reception part is implemented using Chipcon RF modem (CC1000). It works on frequency of 433 MHz and the baud rate used for transmission and reception is 57600 bits/sec. One Chipcon modem is connected to the computer and one is used in the hand held device. Monopole antenna is used to transmit the information over long distances.

2. MAIN DESIGN

The RF based hand held device should be portable and can be very effectively and efficiently used to transmit information to a central system and in case of any enquiry by the hand held device user, the central system can transmit the data back.

2.1. Chipcon Modem

The module is a complete RF transceiver built on a Chipcon recommended printed circuit board design with active and passive components, crystal, antenna connections and mounting socket. It can be interfaced to a user defined microcontroller to send the data stream at a maximum data rate of 76.8 kbps but for the present case the data rate is confined to 57600 bits/sec. The module design supports half duplex RF communication at 433 MHz with an RF output of +10 dBm. The modem board needs to be configured (before the CC1000 module is plugged in) so that it can set up the required parameters of frequency, TX power and data rate on the CC1000 module. With the RF modem board without the CC1000 plugged in, we now can apply the DC input to the board (5 - 9 VDC regulated) either through the round DC socket or via the terminal strip.

Once when the DC input is plugged in, the Red and Yellow LED's flash in a sequence and are then shut off and only the green LED glows steadily. Now, the RF modem can be configured by holding down button 2 (S3) and applying the power to the modem or holding down S3 and restarting the modem by pressing the reset button S1. The modem is connected to a PC running a suitable terminal-emulation program such as the HyperTerminal program furnished with most versions of Microsoft Windows; the user sees the Configuration menu. The menu gives access to a set of configuration and test features and enables us to change the data rate and various RF parameters. The data can then be saved into non-volatile EEPROM data memory, so that the settings are retained when power is switched off.

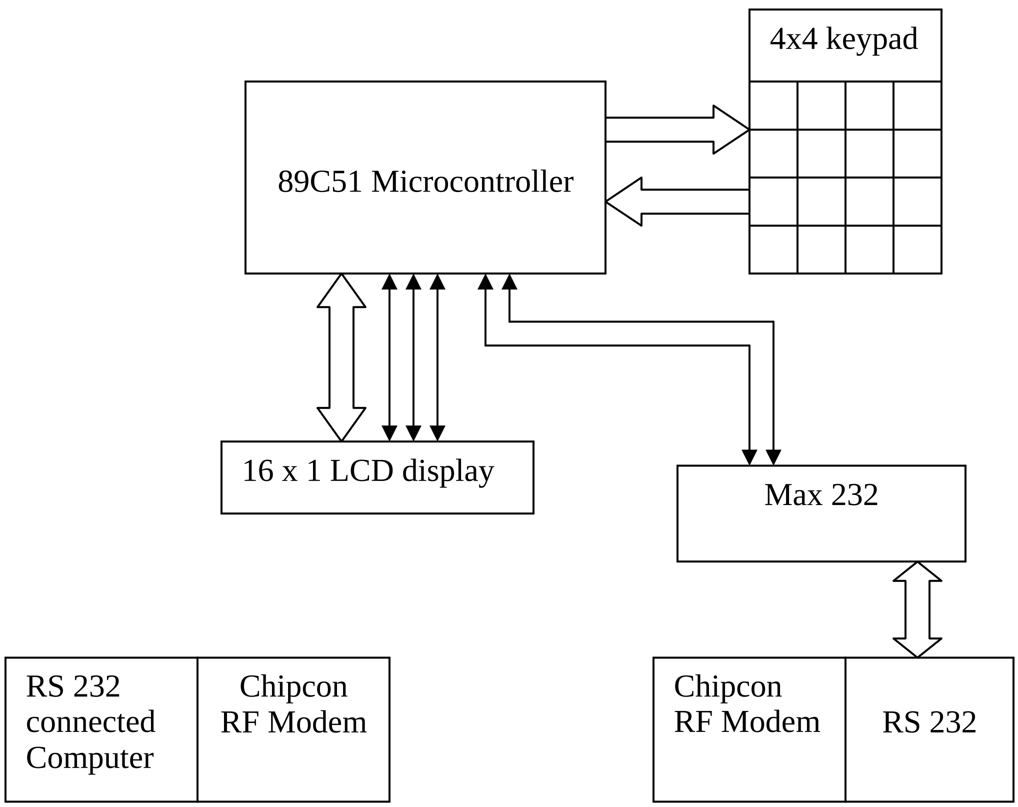


Fig 1. Block Diagram

RF Modem Configuration Menu

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

2.1.1 Circuit Details

The RF modem communicates with a PC or other RS-232 devices over an RS-232 link. 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. In most cases, an RS-232 serial port is full duplex. As the RF link is a half- duplex link (data cannot travel in both directions at the same time), some form of data flow control is necessary. 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.
As the Chipcon Modem is configured to communicate with RS 232 devices only, if we want to program it according to our needs, we need to interface RS 232 to microcontroller through Max 232. For receiving data, the program keeps looking for setting up of receiver interrupt signal (RI). Whenever any data comes, it first set the RI flag and then transfers the data to SBUF (99h). So if we want to read any data from the serial port we need to just read the value in SBUF. Writing data into serial port is similar. In this case the data is fed into SBUF and TI flag is set.

2.2. Keypad Interface

2.2.1 Keypad Scanning

The figure below shows how a hex keypad is connected; each square with number in it is a push button switch, which connects the horizontal wire and vertical wires. So for e.g. if we press button ‘A’ it will connect COL1 and ROW4 and if we press ‘6’ COL3 and ROW2 get connected.

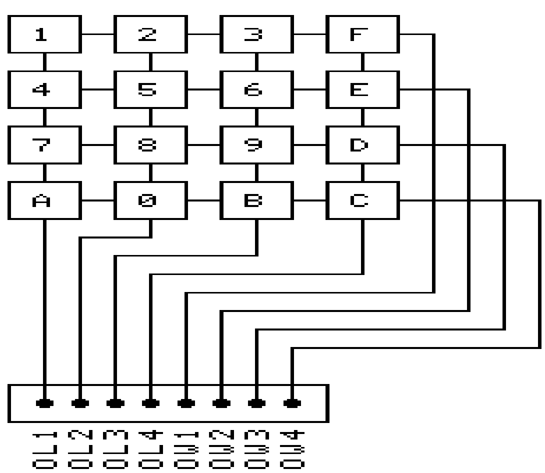


Fig 2. Keyboard Configuration

For scanning switch one ROW at a time low. Suppose we set ROW1 low, we can then read just the top row of buttons, button 1 will take COL1 low, button2 will take COL2 low, and the same for buttons '3' and 'F' in COL3 and COL4. The twelve lower buttons won't have any effect as their respective ROW lines are still high. So to read the other buttons we need to take their respective ROW lines low, taking ROW2 low will allow us to read the second row of buttons (4, 5, 6, and E), again as the other three ROW lines are

now high the other 12 buttons have no effect. We can then repeat this for the last two Row's using ROW3 and ROW4, so we read four buttons at a time, taking a total of four readings to read the entire keypad. This is a common technique for reading keyboards, and is called 'Keyboard Scanning'.

2.2.2. Circuit Details

The keypad is connected to the port 0 of the 89C51 microcontroller through pull-up resistors. The first four pins (columns) are taken as input pins and the other four are configured as output pins. For scanning as explained above, we need to lower one row at a time. Therefore we first output FE to port 0, and then read back the port. If any key in the first row is pressed, the input nibble at the port will be set accordingly, for e.g. if first button of the first row is pressed, then the port will be read as EE. In this way the whole keypad is scanned for each key one by one.

Debounce: The debouncing effect of the keys in the keypad is taken care of through microcontroller coding. The problem of debounce is solved using finite states. Four states, two for key press and two for key release are defined and a key press has to pass through all these states to be declared as successful.

2.3 LCD Interface

The interfacing of the LCD with microcontroller requires three control lines and 8 data lines. The three control lines are referred as EN, RS and RW.

The EN line is called "Enable." This control line is used to tell the LCD that we are sending it data. To send data to the LCD, the program should first set this line high (1) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN low (0) again. The 1-0 transition tells the LCD to take the data currently found on the other control lines and on the data bus and to treat it as a command.

The RS line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "A" on the screen we have to set RS high.

The RW line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands. So RW will almost always be low.

2.3.1 Circuit Details

The pin configuration of LCD consists of 16 pin. Among these 8 are used as data bus which is fed into the port 1 of the microcontroller. Then three pins are for enable, RS, R/W signals, two are for Vcc and ground and one is for contrast enhancement. The remaining two pins are for backlight, in case the LCD has to be used in dark conditions.

2.3.2 LCD initialization

The LCD initialization includes declaration regarding type of LCD (16 x 1 or 16 x 2 etc.) and type of display desired (number of dots/char, number of bits/char). Then few commands are for turning on the cursor and screen, shifting curser every time something is displayed.
Besides initialization the LCD has inbuilt commands for clearing the display, shifting cursor to desired position, etc.

2.3.3 LCD status check

It takes a certain amount of time for each instruction to be executed by the LCD. The delay varies depending on the frequency of the crystal attached to the oscillator input of the 44780 as well as the instruction which is being executed.

While it is possible to write code that waits for a specific amount of time to allow the LCD to execute instructions, this method of "waiting" is not very flexible. If the crystal frequency is changed, the software will need to be modified. Additionally, if the LCD itself is changed for another LCD, it will require more time to perform its operations, the program will not work until it is properly modified.

A more robust method of programming is to use the "Get LCD Status" command to determine whether the LCD is still busy executing the last instruction received.

The "Get LCD Status" command will return to us two tidbits of information; the information that is useful to us right now is found in DB7 (seventh bit of the data bus). In summary, when we issue the "Get LCD Status" command the LCD will immediately raise DB7 if it's still busy executing a command or lower DB7 to indicate that the LCD is no longer occupied. Thus our program can query the LCD until DB7 goes low, indicating the LCD is no longer busy. At that point we are free to continue and send the next command.

2.4 MONOPOLE ANTENNA

A vertically mounted monopole is 37Ω, thus easy to match in 50Ω systems. Theoretically the directivity is 3 dBd (over that of a dipole) because the radiated power is radiated only in the upper half plane due to the ground plane. In practice the gain is lowered because of the conductive loss and finite size of the ground plane. Vertical mounting gives a vertical polarization with omni-directional radiation pattern in the horizontal plane. This antenna is the best solution when the physical size is acceptable and a ground plane is present. Most often the case of the equipment is used as ground plane. If the ground plane is small it will affect the performance of the antenna and tilt the radiation pattern upwards.

2.5 POWER SUPPLY

As the device needs to be portable, batteries are used for power supply in the hand held device. A 9 volt battery is used and is scaled down to 5 volt by 7805. At computer end the power is given by bringing down the supply power of 230 V to 5 V by connecting a transformer followed by a voltage regulator (7805). The transformer used is a 9-O-9, central tap transformer.

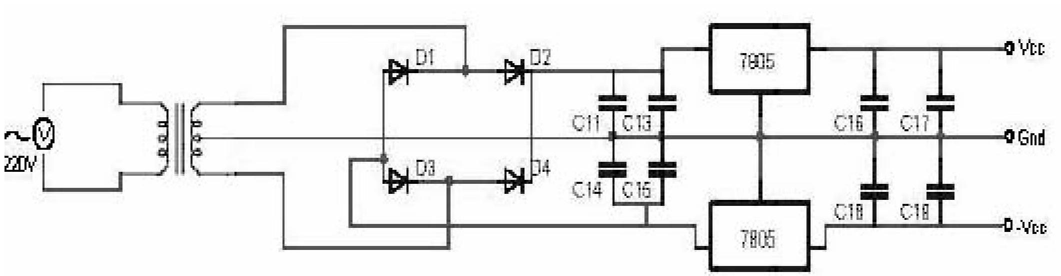


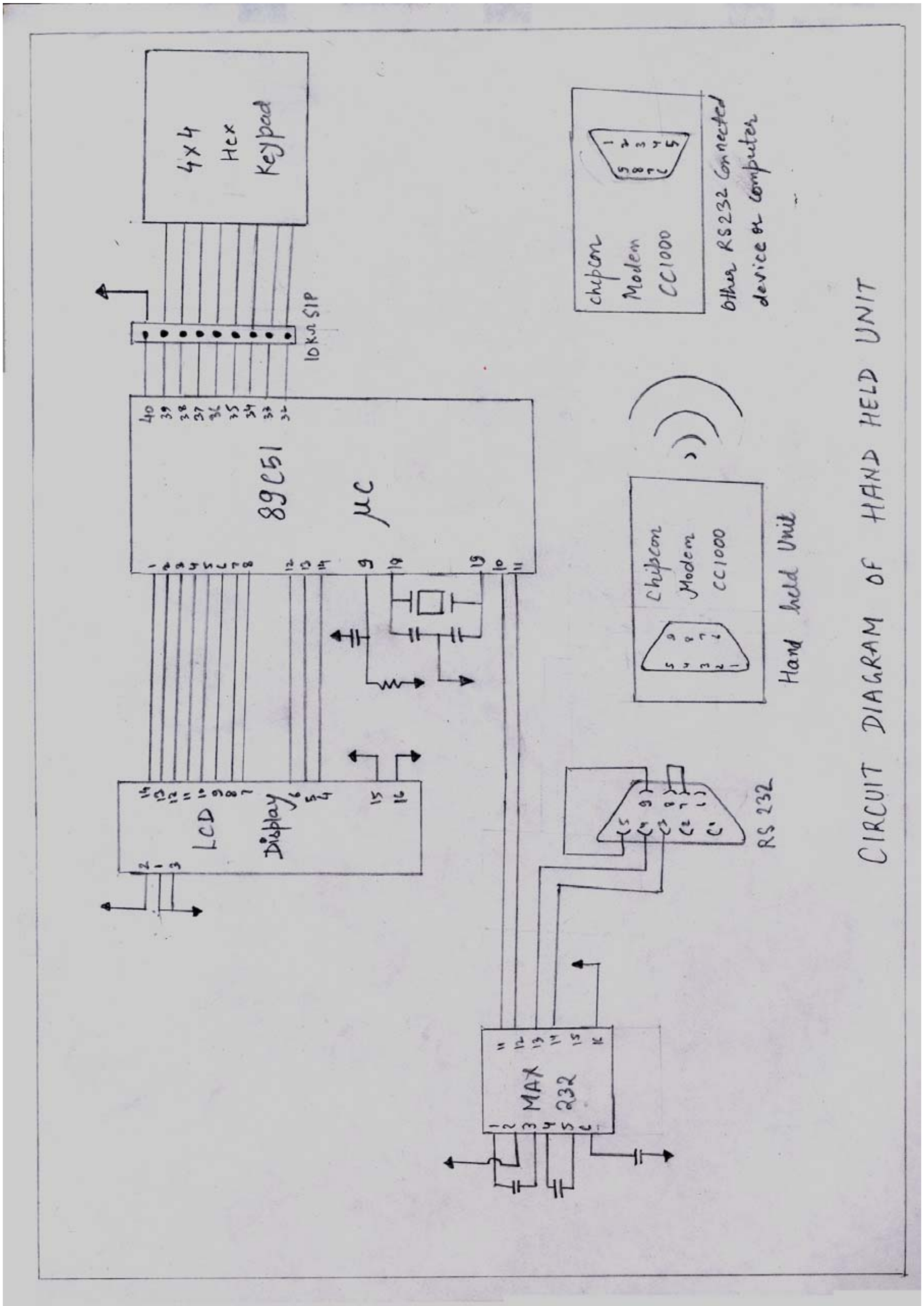
Fig 3. Schematic of the power supply circuit

3. WORKING OF THE DEVICE

Before applying power, first have to plug in the one end of a 9 pin serial connector cable (D type) to the RF Modem Board, RS232 port and the other end to the PC, COM Port. Then open the Microsoft Hyperterminal Window and configure the PC COM Port (COM1, or 2 or 3, or 4 whichever is available) so that communication between the PC & RF modem occurs at 57600 baud, 8 data-bits, 1 stop-bit & no parity**.** Keep the Hyperterminal Window open with the serial chord connected between the RF Modem & the PC.

The device keeps waiting for a signal from a remote computer system. The incoming message is displayed on LCD screen attached with the device. It can remain in two modes

1. Receiving mode: The device is configured to remain in receiving mode till a particular message (predefined key) comes to the device. Of all the messages it gets, it tries to match it with this key. It remains in receiving mode till it finds that key.



CIRCUIT DIAGRAM OF HAND HELD UNIT

2. Transmission mode: The device comes out of transmission mode when it gets the required key. It recognizes the key and start transmitting the information it has. It comes out of the transmission mode when a particular key in the keypad is pressed.

The range of the system depends on the RF frequency, data rate, output power and type of antenna used. Environmental conditions will also affect the range; interference from nearby radio sources may reduce range drastically. If the system is used indoors, a phenomenon known as multi-path fading may be a problem, but can usually be resolved by moving the RF modem around a bit or changing the RF frequency. In normal operation, data packets are sent when the buttons on the modem are pressed. The character sent, however, depends on the settings made in the Configuration menu. This functionality used to in order to make the RF modem as a remote controller for a PC. Three LEDs on the RF modem are used as status indicators. The green LED (D2) functions as a power-on indicator, the yellow LED (D4) is lit when the modem is transmitting RF data or when button 2 (S3) is pressed, and the red LED (D3) is lit when the modem is receiving RF data or when button 1 (S2) is pressed.

3.1 Working of the Keypad

The keypad is a 16 key hex keypad. Three master keys are used for controlling the other keys to have multifunctional keys, just as we have caps lock in keyboard. The first key is for clearing the LCD; it is functional anytime during the transmission or reception. The second controlling key is a key for switching the functions of the keys. When not pressed, the remaining keys send characters ‘A’ – ‘M’. When the key is pressed again the other keys transmit characters ranging from ‘N’ – ‘Z’. Next time when the key is pressed the function of keys changes from ‘0’ – ‘9’ and three characters ‘&’, ‘#’ and ‘$’. Now when the key is pressed again, the keyboard displays characters from ‘A’ – ‘M’. All these keys are functional when it is in transmission mode. The third key is the key pressed when we want to terminate the message we send. Thus after this key is pressed the device again comes in reception mode. At any moment the data send or received is displayed on LCD screen.

J/W/0	K/X/$	L/Y/#	M/Z/&
F/S/6	G/T/7	H/U/8	I/V/9
B/O/2	C/P/3	D/Q/4	E/R/5
Clear	**Control key**	**Mode**	A/N/1

Fig 4. The 4 x 4 hex keypad implemented with three master keys

4. COMPONENT LIST OF THE CIRCUIT DESIGNED

a) Chipcon RF Modem,
b) 89C51 Microcontroller,
c) 4 x 4 Hex Keypad,
d) 16 (characters) x 1 (line) LCD display,
e) Max 232,
f) RS 232 ports,
g) 9 V Battery,
h) 7805 voltage regulator

4. CONCLUSION AND RESULTS

The hand held device is able to communicate with a remote system at distance of upto 50m. The information sent was successfully received at the computer end and the data echoed back was correctly seen at the LCD display of the device. The device transmitted the information only when the particular key (which was set before hand) was sent from the computer. The distance covered by the transmitter can further be improved by fine tuning of the antenna.
There were occasional errors in transmission when the data sent was not received. This can occur due to the non-ideality of the surroundings. Also the directivity of the antenna is not same in all directions.

5. FUTURE SCOPE

There can be lot of improvements which can be made in designing the hand held device.

1. We can configure the device to communicate with RF tags as the encoding used for general RF tags is same as the encoding used in transmission for this Chipcon Modem (CC1000). In both cases Manchester encoding is used. By communicating with RF tags, we can implement RF identification but the wake up and operating frequency of the tag should be same as that of the CC1000 (433 MHz).

2. We can use directional antenna instead of monopole antenna to get the range as high as 100m.

3. Instead of using 16 x 1 LCD display we can use the 16 x 2 or higher member of LCD family, in order to show more text and to make the device more user friendly.

4. For reliability of keypad, we can use hardware debounce or keys which has got inbuilt debounce. In that case we need not take care of the debounce in the software.

5. We can provide one terminal to connect the device to power supply lines so that batteries can be save in case of availability of AC power supply (230V).

References

1. PIC Tutorial - HEX Keypad, http://www.winpicprog.co.uk/pic_tutorial9.htm
2. Introduction to LCD Programming, http://www.8052.com/tutlcd2.phtml
3. Chipcon RF Modem tutorial, http://www.chipcon.com
4. Kenneth J. Ayala, The 8051 Microcontroller, Architecture, Programming & Applications
5. LCD & keys, http://avrsite.fateback.com/circuits/lcd_keys/lcd_keys.htm,
6. Plug and play the CC1000, E-Smart Application notes, Indranil Majumdar
7. Electrical Engineering Dept, IIT Bombay (2004), Datasheets and General Information, www.ee.iitb.ac.in/uma/~wel