ETHERNET CHAT CLIENT TUTORIAL

WHAT YOU NEED

Enc28j60 is an ethernet IC which transmits and receives the data over ethernet. You need just a microcontroller (the one we used is ATMEGA 16), enc28j60 ic, a Magjack (the one we have used is DR-RJ45IM from embedded market).

HOW TO DO IT

You have just to do the following connections. In our original circuit we have used PS/2 keyboard take the input from the user and the lcd to display it.

Connecting Enc28j60

First of all, you will need to make all the connections of the enc28j60 ic leaving out the atmega. And then when you'll put the lan cable in the magjack you'll see that the yellow led is continuously on and the green led will glow at times. If it is so, then your connections are all right and you can make the connections of atmega. If not double check your connections to get the desired result.

Connecting Atmega

Now proceed to make all the connections of the Atmega with ENC28J60. Now run the code "ledconfig" in the atmega. You should

see the leds blinking, one with a frequency of 1 Hz and other with that of 2 Hz.

Connecting LCD and PS/2 Keyboard

You can now complete the entire circuit. Load the code “main” in the Atmega and get chatting with our friends.

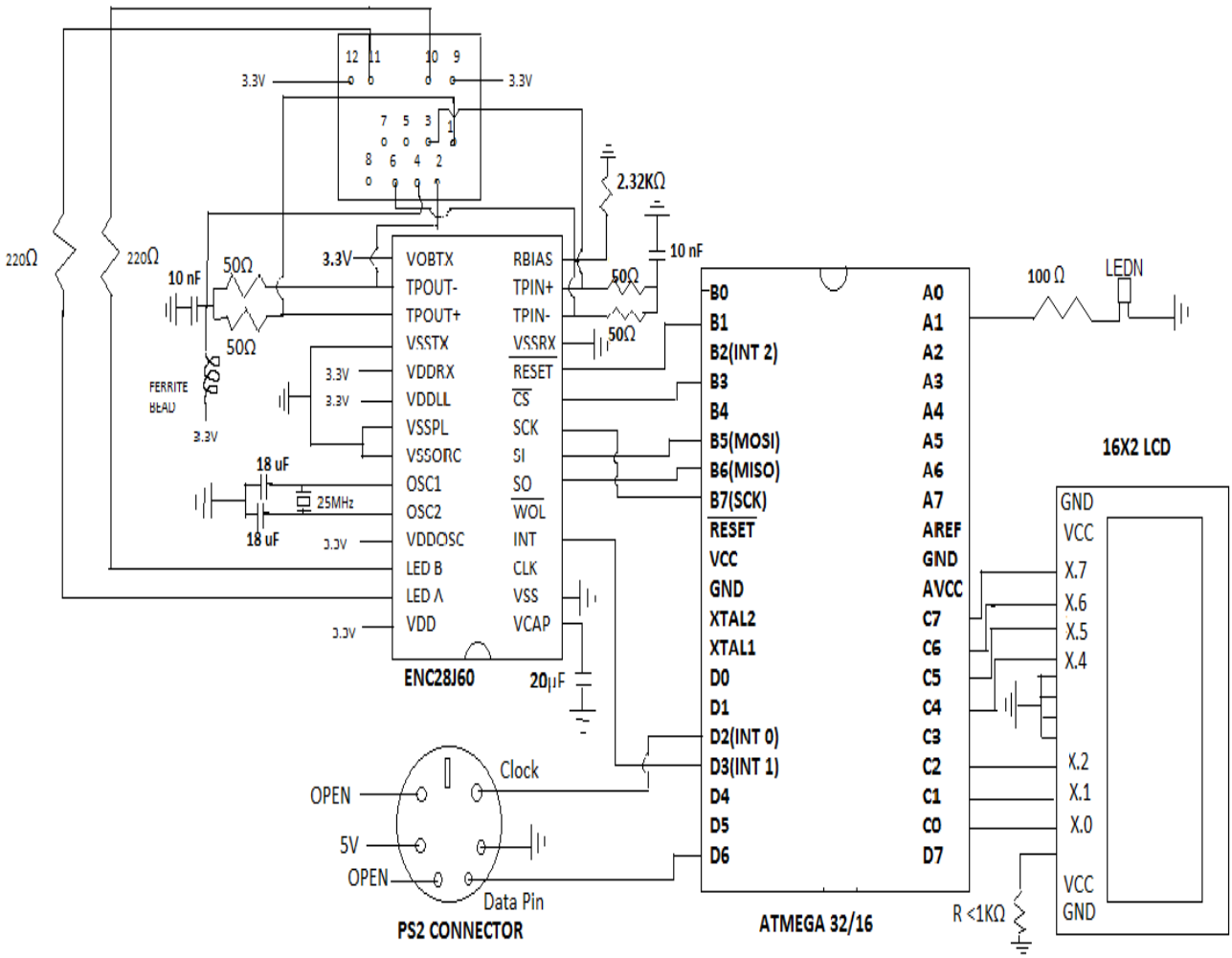
Note: For the first circuit you make, you need not change the contents of mac[](MAC address are a unique identification id for your module) But for the second circuit you will made(if you make any) change it to something else

Running the Code

When you run the code, it will ask you for your IP address. So , if your IP address is 172.24.9.41 , you should enter 172024009041..

Also the keyboard keys F4 and F5 have special functions. F5 is used to clear all that you have written that you wish not to send and F4 is used to display the message you received(Whenever you will have any unread message, the LED marked LEDN will be on, so youknow when to press F4)..

CIRCUIT DIAGRAM



BASIC FUNCTIONS

There are some basic functions to control this ethernet IC you can just use them to of control the IC .

a. **uint8_t enc28j60Read(uint8_t address);**

address is the address of the control register to be read

Note: The address of control registers is actually of 5 bits but for simplicity we covert it into a 8 bit address out of which B7 bit is 1 if the register is MAC or MII and 0 if it is ETH. The bits B6 – B5 is the bank of the control register .The last 5 bits are address of control register in that bank.(For more details about the bank refer to the documentation)

- It returns the value of the control register
- It sets the bank by its own.

b. **void enc28j60ReadBuffer(uint16_t len,uint8_t *data);**

- It stores len bytes of data in the memory pointed by the data pointer.

c. **void enc28j60Write(uint8_t address,uint8_t data);**

- It writes 8 bit data to the control register.

d. **void enc28j60WriteBuffer(uint16_t len,uint8_t *data);**

- It writes len bytes of data to the transmit buffer.

e. **void enc28j60PhyWrite(uint8_t address,uint16_t data);**

- Writes 16 bit data to the PHY register whose address is given.

f. **void enc28j60WriteOp(ENC28J60_BIT_FIELD_CLR,uint8_t address,uint8_t data);**

- **data** field will be : if bit 0 and 6 are to be cleared we will send 01000001

g. **void enc28j60WriteOp(ENC28J60_BIT_FIELD_SET, uint8_t address,uint8_t data);**

- **data** field will be : if bit 1 and 7 are to be set we will send 10000010

h. **void enc28j60PacketSend(uint16_t len, uint8_t* packet)**

- this function takes the pointer to data packet and its length and transmits the data over lan

i. uint16_t enc28j60PacketReceive(uint16_t maxlen, uint8_t* packet)

- this function writes the received data in an array upto the maximum length provided.

JUST PLAY AROUND!!!

Add the .c and .h files of enc28j60 given to you. Include the header file enc28j60.h along with the others which you generally include.

REMEMBER: enc28j60 reads the data through SPI at each rising edge of the clock and sends the data at each falling edge of the clock. So while enabling SPI make configurations accordingly.(

In the main() after configuring various registers of ATMega like DDR, MCUCR, etc. call the function
void enc28j60Init(uint8_t*macaddr); .Don't get afraid of uint8_t its just unsigned char. Pass it the mac address of your ethernet IC i.e. enc28j60.You can Google What a mac address is? Just use 0x 00 04 A3 FF FF FF. As visible it is a 6 byte address which I have written in hexadecimal.

There is a lot more to do!! We are providing .c and .h files for ip_arp_udp which has various functions such as creating an arp reply from request, udp reply and lots more.

JUST BROWSE THROUGH THEM!!

Now we are giving our code which we have written to chat over the LAN. You just copy paste the entire code and add the header files and get going.

And one thing more, as we have in our project used the keyboard for entering the data and lcd for displaying the received as well as typed text so we are giving you those header files too.

Feed the code in ATMega ,build up the circuit shown and start chatting with your friend on the other LAN port having same device or some alternative to capture and send UDP data packets(as in code we have used UDP protocol for communication).

You can choose any other protocol as well such as TCP/IP, ICMP or ARP . You will then just need to form a data packet (array) with entries corresponding to various fields as described in their corresponding datagram (i.e. header+text) and then using packetsend function you can send the data to the destination.