CH ENC28J60 ETHERENET MODULE

User Manual

CH ENC28J60 ETHERENET MODULE is dedicated to the Ethernet and easy to plug to the Arduino via the SPI

Pros	Cons
Cheap	library no maintained by Arduino Team
Lot of open source libraries	Need to make a choice to select the right librairie
Can be used directly on atmega (without arduino card)	

Now it is time to choose a good library but before choosing a lib. How to wire it?

Wiring a Enc28j60 module:

- Enc28j60 SO -> Arduino pin 12
- Enc28j60 SI -> Arduino pin 11
- Enc28j60 SCK -> Arduino pin 13
- Enc28j60 CS -> Arduino pin 10
- Enc28j60 VCC -> Arduino 3V3 pin
- Enc28j60 GND -> Arduino Gnd pin

Choosing a libraries for Enc28j60
Now your system is ready. It is time to choose a library. There are 2 main libraries:
EtherCard:
EtherCard is a driver for the ENC28J60 chip, compatible with Arduino IDE.
Adapted and extended from code.
License: GPL2
EtherCard is found in Software
It is compliant with:

- DHCP
- DNS
- UDP

NB: pin Connected to Enc28j60 CS can be chosen programmatically

Example of a simple server:

```
// this is a demo of the RBBB running as webserver with the Ether Card
#include <EtherCard.h>
// ethernet interface mac address, must be unique on the LAN
static byte mymac[] = { 0x74,0x69,0x69,0x2D,0x30,0x31 };
```

```
static byte myip[] = { 192,168,1,203 };
byte Ethernet::buffer[500];
BufferFiller bfill;
void setup () {
  if (ether.begin(sizeof Ethernet::buffer, mymac) == 0)
    Serial.println(F("Failed to access Ethernet controller"));
  ether.staticSetup(myip);
}
static word homePage() {
  long t = millis() / 1000;
  word h = t / 3600;
  byte m = (t / 60) % 60;
  byte s = t % 60;
  bfill = ether.tcpOffset();
  bfill.emit_p(PSTR(
    "HTTP/1.0 200 OK\r\n"
    "Content-Type: text/html\r\n"
    "Pragma: no-cache\r\n"
    "\r\n"
    "<meta http-equiv='refresh' content='1'/>"
    "<title>RBBB server</title>"
    "<h1>$D$D:$D$D:$D$D</h1>"),
      h/10, h%10, m/10, m%10, s/10, s%10);
  return bfill.position();
}
void loop () {
  word len = ether.packetReceive();
  word pos = ether.packetLoop(len);

  if (pos)  // check if valid tcp data is received
    ether.httpServerReply(homePage()); // send web page data
}
```

NB: on this example, if you use the previous wiring, you have to change the setup function:

```
void setup () {
  //CS is connected to pin 10
  if (ether.begin(sizeof Ethernet::buffer, mymac, 10) == 0)
    Serial.println(F("Failed to access Ethernet controller"));
  ether.staticSetup(myip);
}
```

UIPEthernet

A plugin-replacement of the stock Arduino Ethernet library for ENC28J60 shields and breakout boards. Full support for persistent (streaming) TCP-connections and UDP (Client and Server each), ARP, ICMP, DHCP and DNS. Build around Adam Dunkels uIP Stack.

License: GPL v3

UIPEthernet is found in Software

This library uses the same API as the official Arduino Ethernet. It's useful to integrate ENC28J60 in project on the same way you integrate Arduino Shield

It is compliant with

- DHCP
- DNS
- UDP
- TCP
- ARP
- ICMP

Example of simple server

```
/*
 * UIPEthernet EchoServer example.
 *
 * UIPEthernet is a TCP/IP stack that can be used with a enc28j60 based
 * Ethernet-shield.
 *
 * UIPEthernet uses the fine uIP stack by Adam Dunkels <adam@sics.se>
 *
 *    ----------
 *
 * This Hello World example sets up a server at 192.168.1.6 on port 1000.
 * Telnet here to access the service.  The uIP stack will also respond to
 * pings to test if you have successfully established a TCP connection to
 * the Arduino.
 *
 * This example was based upon uIP hello-world by Adam Dunkels <adam@sics.se>
 * Ported to the Arduino IDE by Adam Nielsen <malvineous@shikadi.net>
 * Adaption to Enc28J60 by Norbert Truchsess <norbert.truchsess@t-online.de>
 */
#include <UIPEthernet.h>
EthernetServer server = EthernetServer(1000);
void setup()
{
 Serial.begin(9600);
 uint8_t mac[6] = {0x00,0x01,0x02,0x03,0x04,0x05};
 IPAddress myIP(192,168,0,6);
 Ethernet.begin(mac,myIP);
 server.begin();
}
void loop()
{
 size_t size;
 if (EthernetClient client = server.available())
  {
   while((size = client.available()) > 0)
    {
     uint8_t* msg = (uint8_t*)malloc(size);
     size = client.read(msg,size);
     Serial.write(msg,size);
      free(msg);
    }
   client.println("DATA from Server!");
   client.stop();
  }
}
```

NB: As it is compliant with classical Arduino Ethernet Lib. You can use Webduino on top of it.
Webduino is an Arduino-based Web Server library, originally developed for a class at NYC Resistor. It's called Webduino, and it's an extensible web server library for the Arduino using the Wiznet-based Ethernet shields. It's released under the MIT license allowing all sorts of reuse.
How to add Library

https://www.arduino.cc/en/guide/libraries

How to open software:

- Enter to http://www.ekt2.com/products/productdetails?ProductId=7E83B9D3-0365-4452-9383-FFDCE860A47D
- Press the icon to download software