

arduino 上使用 ENC28J60 以太网控制器的教程—作为客户端

第三篇文章，我会介绍如何连接一个网站并获取返回的信息。

当你学会了如何（使用静态 ip 地址或者通过 DHCP 服务器自动获取 ip 地址）把你的 Arduino 连接到网络后，你需要学习如何向一个网站发送数据和接收该网站的返回数据。

为了使下面的例子更实际，我编写了一个简单的 php 网页，每次连接将随机返回一句格言，地址如下 <http://www.lucadentella.it/demo/aphorisms.php>。



我们来编写一个 Arduino 代码来获得这些格言并输出到串口。

DNS

通常我们连接一个网站是键入他的域名（如 www.sina.com.cn），我们的电脑，会通过一个 dns 服务器将他的域名解析为相应 ip 来进行连接。

在 arduino 中配置你网络对应的 dns 服务器地址是非常重要的：

如果你的 arduino 是采用 DHCP 服务器来进行配置，通常 DNS 服务器地址也是自动获取的。

如果是静态配置，那么你调用 `staticSetup()`方法时需要包含 DNS 服务器地址。

```
1. #include <EtherCard.h>
2. static byte mymac[] = {0xDD,0xDD,0xDD,0x00,0x00,0x01};
3. static byte myip[] = {192,168,1,10};
4. static byte gatewayip[] = {192,168,1,1};
5. static byte dnsip[] = {151,99,125,2};
6. ...
7. ether.staticSetup(myip, gatewayip, dnsip);
```

你可以用 `dnsLookup()`方法验证域名服务器是否正常。

```
1. char website[] PROGMEM = "www.lucadentella.it";
2. ...
3. if (!ether.dnsLookup(website))
4.     Serial.println("DNS failed");
5. else
6.     Serial.println("DNS resolution done");
```

```
7. ether.printIp("SRV IP:\t", ether.hisip);
```

连接

EtherCard 库提供了一个非常方便的方法 **browseUrl()** 来连接一个网站，This method prepares the connection; which is completed in following steps performed（不知道咋个翻译） – during the loop - 通过我们已经学了的 2 个指令：

```
1. ether.packetLoop(ether.packetReceive());
```

非常重要的是需要保持这两个指令运行，才能使代码正常工作。

browseUrl() 方法需要一些参数：

prog_char urlbuf，地址的 固定部分；

const char * urlbuf_varpart，地址的变量部分；

prog_char hoststr，网站的名称；

void (*callback)(byte,word,word))，这是一个回调函数名称，连接结束时调用的函数。

地址分为两部分，以优化内存的利用率：静态部分存储在 **flash** 中（**PSTR()** 命令），以节省 **RAM** 空间。

回调函数

回调函数包含连接结束时执行的一个动作，在这个例子，是在串口打印出网站返回的格言。

你可以选择任意的函数名称，但必须声明如下的参数：

```
1. static void response_callback (byte status, word off, word len)
```

第一个参数为连接的状态，第二个参数为响应存储在 **buffer** 中的偏移量，第三个参数为响应的长度。

要理解偏移量（**OFF**）参数的含义，你必须要注意的以太网连接发送和接收的所有数据存储在我们定义的缓冲区中：

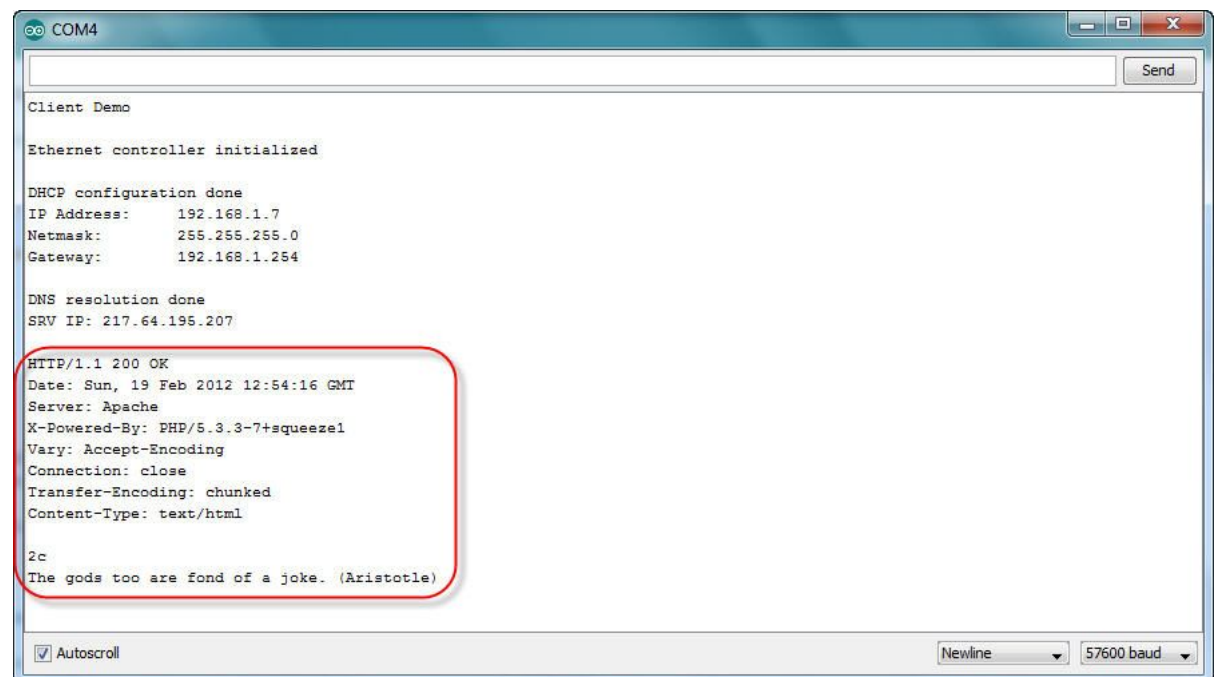
```
1. byte Ethernet::buffer[700];
```

你收到数据，整个数据包（包含头文件及校验字段等），都存储在这个缓冲区中，偏移量值告诉我们从那个字段开始存储这个数据包，及哪儿是我们收到的网页响应。

下面是一个回调函数在串口打印网页的响应：

```
1. static void response_callback (byte status, word off, word len) {  
2.  
3.   Serial.print((const char*) Ethernet::buffer + off);  
4. }
```

如果你运行这个代码（你可以在这个[链接](#)找到）（注意要把回调函数改成以上回调函数），你就可以看到如下信息：



```
COM4
Client Demo

Ethernet controller initialized

DHCP configuration done
IP Address: 192.168.1.7
Netmask: 255.255.255.0
Gateway: 192.168.1.254

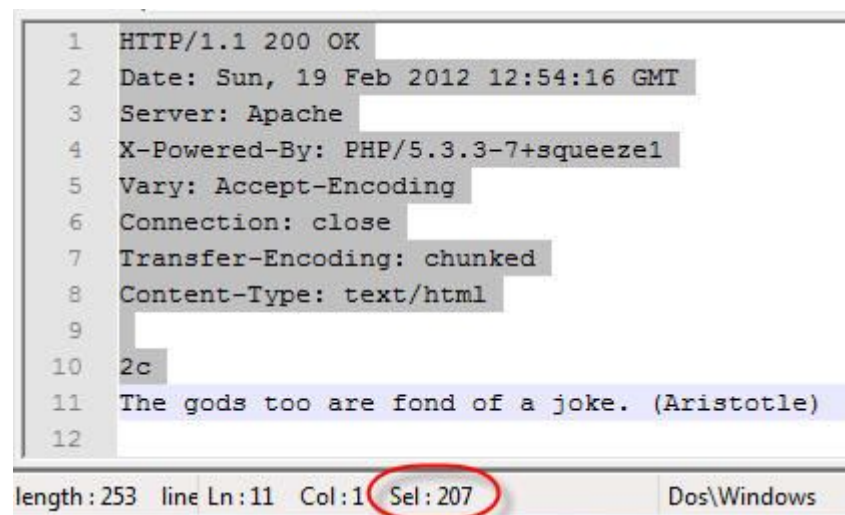
DNS resolution done
SRV IP: 217.64.195.207

HTTP/1.1 200 OK
Date: Sun, 19 Feb 2012 12:54:16 GMT
Server: Apache
X-Powered-By: PHP/5.3.3-7+squeezel
Vary: Accept-Encoding
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html

2c
The gods too are fond of a joke. (Aristotle)
```

arduino 正确配置了 DHCP，它解析“www.lucadentella.it”这个域名，连接网站，打印网站响应。

响应不仅包含格言，还包含了 HTTP 头文件。你需要打印所有的响应，然后通过文本编辑器（推荐 notepad++），找到它们有多少个字符：



```
1 HTTP/1.1 200 OK
2 Date: Sun, 19 Feb 2012 12:54:16 GMT
3 Server: Apache
4 X-Powered-By: PHP/5.3.3-7+squeezel
5 Vary: Accept-Encoding
6 Connection: close
7 Transfer-Encoding: chunked
8 Content-Type: text/html
9
10 2c
11 The gods too are fond of a joke. (Aristotle)
12
```

length: 253 line Ln: 11 Col: 1 Sel: 207 Dos\Windows

这些行位于接收缓冲器中的格言之前。所以我们需要改变回调函数，打印 207 个字符之后的缓冲区的内容：

1. `static void response_callback (byte status, word off, word len) {`
- 2.
3. `Serial.print((const char*) Ethernet::buffer + off + 207);`

4. }

经过以上更改，你就可以接收到正确的数据：

