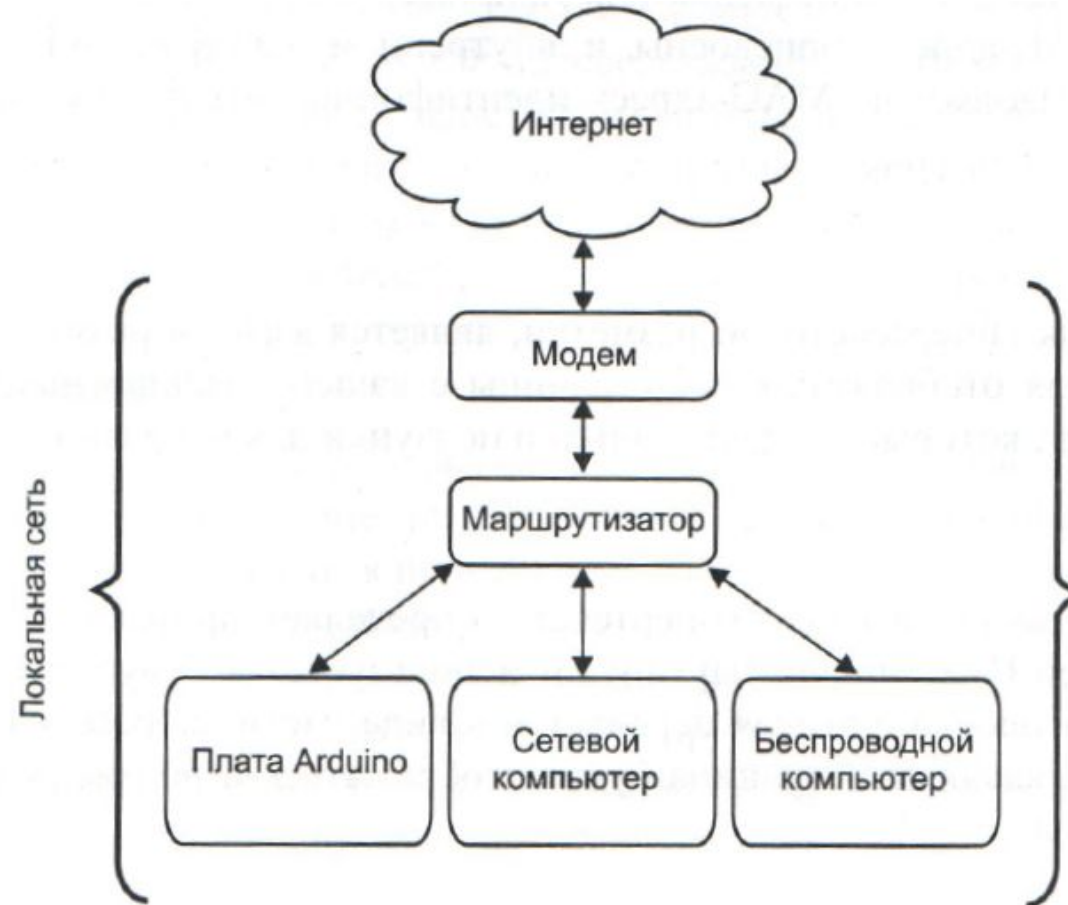


# Лекция № 14

Arduino и Internet

# Локальная сеть и интернет



# Сетевые термины

IP – адрес – это уникальный адрес, который идентифицирует каждое устройство, подключенное к сети.

При работе в локальной сети на самом деле существуют два адреса: внутренний и глобальный. Поэтому может понадобиться преобразователь сетевых адресов (NAT).

# Сетевые термины

MAC – адрес, в отличии от IP-адреса уникален в пределах всей сети. Они назначаются каждому сетевому устройству и не меняются.

HTML – язык гипертекстовой разметки, т.е. язык разметки документов.

HTTP – протокол передачи гипертекста, определяет протокол связи через World Wide Web (Всемирную паутину) и используется в браузерах. Задаёт информацию заголовка, которая передается в виде части сообщения. Этот заголовок определяет какая веб-страница будет отображаться и подтверждает успешное получение данных

# Сетевые термины

GET/POST – это два способа передачи информации на удаленный веб-сервер.

[www.elenakurash.com/?s=arduino](http://www.elenakurash.com/?s=arduino) – GET-запрос определяет ряд переменных, следующих за вопросительным знаком в URL (*Uniform Resource Locator – единый указатель ресурса*). В данном случае передается переменная *s* со значением *arduino*. Когда страница получает этот URL, он идентифицирует переменную *s*, выполняет поиск и возвращает страницу результатов.

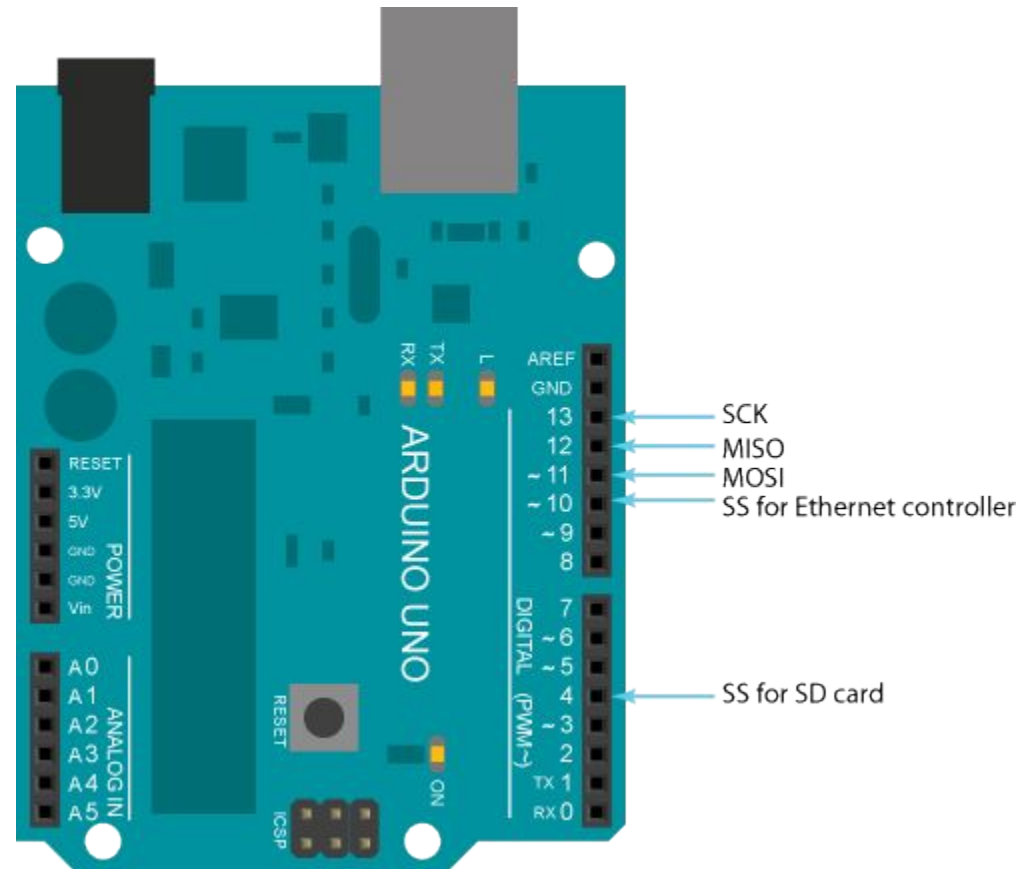
POST- очень похож на GET, но информация не видна в URL. Данные передаются в фоновом режиме для того, чтобы скрыть конфиденциальную информацию.

# Сетевые термины

DHCP – протокол динамической конфигурации узла, подключает устройства к локальной сети за один шаг.

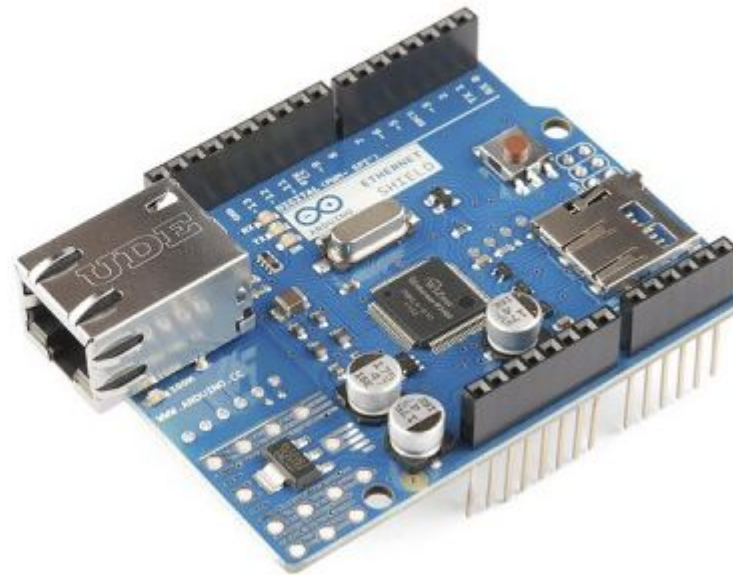
DNS (Domain Name System) – система доменных имен. Каждый веб-сайт в сети имеет уникальный IP-адрес. Например при вводе [www.google.com](http://www.google.com), DNS-сервер смотрит на список, который информирует его об IP-адресе, связанном с этим URL. Затем он передает IP-адрес обратно в браузер, чтобы он мог общаться с сервисами Google.

# Подключение Arduino к сети



# Плата Arduino Ethernet Shield

Arduino Ethernet Shield





# Создание простой веб-страницы

```
<form action="" method='get'>  
  <input type='hidden' name='L' value='7' />  
  <input type='submit' value='Toggle Red' />  
</form><form action="" method='get'>  
  <input type='hidden' name='L' value='6' />  
  <input type='submit' value='Toggle Green' />  
</form>
```

```
<form action="" method='get'>
<input type='hidden' name='L' value='5' />
<input type='submit' value='Toggle Blue' />
</form><form action="" method='get'>
<input type='range' name='S' min='0' max='1000' step='100'
value='0'/>
<input type='submit' value='Set Frequency' />
</form>
```

C:\Users\Елкен\Desktop × +

← → ↻ | file:///C:/Users/Елкен/Desktop/Новая%20папка%20(3)/server\_form/server\_form.html | 📖 ☆ | ☰ ✎ 🔔 ⋮

Toggle Red

Toggle Green

Toggle Blue

■  Set Frequency

📊 🗣️ 🇷🇺 🇻🇳 📄 📁 🔍 🗑️ 🌐

Подключим RGB-светодиод к контактам 5,6,7 платы Arduino, а динамик к третьему контакту.

# Программа

```
#include <Ethernet.h>
#include <SPI.h>
const int BLUE  =5;
const int GREEN =6;
const int RED   =7;
const int SPEAKER =3;
//For controlling LEDS and the speaker
//If you want to control additional things, add variables to control them here.
int freq = 0;
int pin;
//Set to your MAC address!
//It should be on your sticker. If you can't find it,
//make one up, or use this one.
byte mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x4A, 0xE0 };
```

```
//Start the server on port 80
```

```
EthernetServer server = EthernetServer(80); //port 80
```

```
boolean receiving = false; //To keep track of whether we are getting data.
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  pinMode(RED, OUTPUT);
```

```
  pinMode(GREEN, OUTPUT);
```

```
  pinMode(BLUE, OUTPUT);
```

```
//Connect with DHCP
if (!Ethernet.begin(mac)) {
  Serial.println("Could not Configure Ethernet with DHCP.");
  return;
}
else{
  Serial.println("Ethernet Configured!");
}
//Start the server
server.begin();
Serial.print("Server Started.\nLocal IP: ");
Serial.println(Ethernet.localIP());
}
```

```
void loop()
{
  EthernetClient client = server.available();
  if (client)
  {
    //An HTTP request ends with a blank line
    boolean currentLineIsBlank = true;
    boolean sentHeader = false;
    while (client.connected())
    {
      if (client.available())
      {
        char c = client.read(); //Read from the incoming buffer
```



```
if(receiving && c == ' ') receiving = false; //done receiving
  if(c == '?') receiving = true; //found arguments
  //This looks at the GET requests
  if(receiving){
    //An LED command is specified with an L
    if (c == 'L'){
      Serial.print("Toggling Pin ");
      pin = client.parseInt();
      Serial.println(pin);
      digitalWrite(pin, !digitalRead(pin));
      break;
    }
  }
```

```
//A speaker command is specified with an S
  else if (c == 'S') {
    Serial.print("Setting Frequency to ");
    freq = client.parseInt();
    Serial.println(freq);
    if (freq == 0)
      noTone(SPEAKER);
    else
      tone(SPEAKER, freq);
    break;}
//Add similarly formatted else if statements here
//TO CONTROL ADDITIONAL THINGS
}
```

```
//Print out the response header and the HTML page
if(!sentHeader)
{
    //Send a standard HTTP response header
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html\n");

    //Red toggle button
    client.println("<form action=\" method='get'>");
    client.println("<input type='hidden' name='L' value='7' />");
    client.println("<input type='submit' value='Toggle Red' />");
    client.println("</form>");
```

**//Green toggle button**

```
client.println("<form action="" method='get'>");  
client.println("<input type='hidden' name='L' value='6' />");  
client.println("<input type='submit' value='Toggle Green' />");  
client.println("</form>");
```

**//Blue toggle button**

```
client.println("<form action="" method='get'>");  
client.println("<input type='hidden' name='L' value='5' />");  
client.println("<input type='submit' value='Toggle Blue' />");  
client.println("</form>");
```

```
//Speaker frequency slider
```

```
    client.println("<form action="" method='get'>");
```

```
    client.print("<input type='range' name='S' min='0' max='1000'  
step='100' value='0'/>");
```

```
    client.println("<input type='submit' value='Set Frequency' />");
```

```
    client.println("</form>");
```

```
//Add additional forms forms for controlling more things here.
```

```
    sentHeader = true;
```

```
}
```

```
if (c == '\n' && currentLineIsBlank) break;
    if (c == '\n'){
        currentLineIsBlank = true;
    }
    else if (c != '\r') {
        currentLineIsBlank = false;
    }
}
}
}
delay(5); //Give the web browser time to receive the data
client.stop(); //Close the connection:
}
}
```