## Основные понятия технологий построения сетей

**Сервер -**это компьютер, который обслуживает другие компьютеры в сети. Существуют разнообразные виды серверов, отличающиеся друг от друга услугами, которые они предоставляют; серверы баз данных, файловые серверы, принт-серверы, почтовые серверы, веб-серверы и т. д.

**Протоколы TCP/IP** (Transmission Control Protocol/Internet Protocol - Протокол управления передачей данных/Интернет протокол) являются основными межсетевыми протоколами и управляют передачей данных между сетями разной конфигурации и технологии. Именно это семейство протоколов используется для передачи информации в сети Интернет, а также в некоторых локальных сетях. Семейство протоколов TPC/IP включает все промежуточные протоколы между уровнем приложений и физическим уровнем. Общее их количество составляет несколько десятков.

При построении локальной сети на основе протокола TCP/IP каждый компьютер получает уникальный IP-адрес, который может назначаться либо DHCP-сервером, либо вручную.

**IP-адрес**  - уникальный [сетевой адрес](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%B0%D0%B4%D1%80%D0%B5%D1%81) [узла](https://ru.wikipedia.org/wiki/%D0%A3%D0%B7%D0%B5%D0%BB_%D1%81%D0%B5%D1%82%D0%B8) в [компьютерной сети](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C), построенной на основе стека протоколов [TCP/IP](https://ru.wikipedia.org/wiki/TCP/IP).

**MAC-адрес** (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Media Access Control* — управление доступом к среде, также Hardware Address) — уникальный идентификатор, присваиваемый каждой единице активного оборудования или некоторым их интерфейсам в [компьютерных сетях](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C) [Ethernet](https://ru.wikipedia.org/wiki/Ethernet" \o "Ethernet).

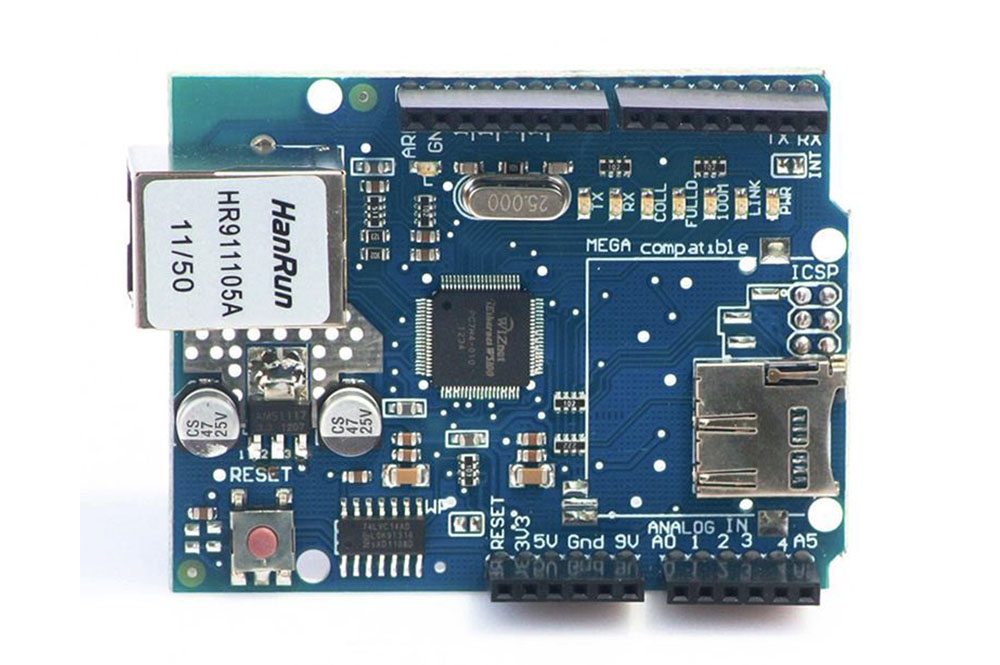
**DHCP** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Dynamic Host Configuration Protocol — протокол динамической настройки узла) — [сетевой протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB), позволяющий компьютерам автоматически получать [IP-адрес](https://ru.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81) и другие параметры, необходимые для работы в сети [TCP/IP](https://ru.wikipedia.org/wiki/TCP/IP). Данный протокол работает по модели «[клиент-сервер](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80)». Для автоматической конфигурации компьютер-клиент на этапе конфигурации сетевого устройства обращается к так называемому [серверу](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5)) DHCP и получает от него нужные параметры.

**Сетевой шлюз** — это точка сети, которая служит выходом в другую сеть. В сети Интернет узлом или конечной точкой может быть или сетевой шлюз, или хост. Интернет-пользователи и компьютеры, которые доставляют веб-страницы пользователям — это хосты, а узлы между различными сетями — это сетевые шлюзы.

**DNS** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Domain Name System* — система доменных имён) — компьютерная [распределённая система](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D1%91%D0%BD%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) для получения информации о [доменах](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%B8%D0%BC%D1%8F). Чаще всего используется для получения IP-адреса по имени [хоста](https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82) (компьютера или устройства).

**Ethernet** - семейство технологий пакетной передачи данных для компьютерных сетей. Название «Ethernet» (буквально «эфирная сеть» или «среда сети») отражает первоначальный принцип работы этой технологии: всё, передаваемое одним узлом, одновременно принимается всеми остальными (то есть имеется некое сходство с радиовещанием). В настоящее время практически всегда подключение происходит через коммутаторы (switch), так что кадры, отправляемые одним узлом, доходят лишь до адресата (исключение составляют передачи на широковещательный адрес) — это повышает скорость работы и безопасность сети.

## Знакомство с Ethernet shield

Arduino Ethernet Shield W5100 одна из интереснейших плат расширения позволяющая управлять платой Arduino через локальную сеть или интернет. Это может пригодиться, если вам нужно получать какую-то информацию для работы из сети, разместить свой собственный веб сайт или управлять ардуино по сети. На данной плате расширения также имеется слот для microSD карты памяти. Для управления платой используется базовая библиотека Arduino IDE - Ethernet library, а для работы microSD картой - SD library. Для работы, помимо Arduino и, собственно, шилда вам также понадобится сетевой кабель (Витая пара).

Для соединения с локальной сетью используйте стандартный разъём 8P8C (RJ45) на борту. Один конец [патч-корда витой пары](http://amperka.ru/product/patch-cord) подключите к роутеру или хабу, второй — к Ethernet Shield.

Характеристики:

* Ethernet-чип: Wiznet W5500;
* рабочее напряжение: 5 В;
* размер буфера: 32 КБ;
* скорость соединения: 10/100 Мбит;
* занимаемые пины: SPI (MISO, MOSI, SCK), 10;
* габариты: 69×53 мм (RJ45 выступает на несколько мм).

## Подключение Ethernet shield к Arduino UNO



Для установки Ethernet shield вам необходимо поместить его сверху на плату Arduino UNO так, чтобы контакты Ethernet shield входили в соответствующие гнезда, установленные на Arduino UNO. Сетевой кабель (витую пару) необходимо подключить к Ethernet shield и к коммутатору. При подаче напряжения на плату Arduino на Ethernet shield должны засветиться индикаторы приема и передачи данных.

В плату встроены следующие индикаторы:

* *PWR* загорается, когда подведено питание;
* *LINK* загорается, если сеть доступна. Мигает при получении и передачи данных;
* *FULLD* горит, если установлено дуплексное соединение. То есть возможен одновременный приём и передача;
* *100M* горит, если соединение осуществлено на уровне 100 Мбит/с (не 10 Мбит/с);
* *RX* мигает при получении данных;
* *TX* мигает при передаче данных;
* *COLL* мигает при возникновении коллизий в сети. То есть когда в режиме полудуплекса два устройства пытаются одновременно передать пакет.

*Примечание: если индикатор LINK не горит, проверьте сетевой кабель (витую пару). Так же убедитесь, что маршрутизатор, к которому подключен Ethernet shield имеет доступ в Интернет.*

## Настройка сетевого интерфейса

Рассмотрим настройку сетевого интерфейса на примере программы ниже.

**Программный код**

\* Lesson6\_1

\*Настройка сетевого интерфейса

\*Ethernet shield

\*/

#include <SPI.h>

#include <Ethernet.h>

// Укажите MAC адресс вашего контроллера ниже

// В СЕТИ НЕ ДОЛЖНО БЫТЬ НЕСКОЛЬКО УСТРОЙСТВ С ОДИНАКОВЫМ MAC АДРЕСОМ

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

// Укажите статический IP адрес, если в сети не используется DHCP сервис

IPAddress ip(192, 168, 1,211);

// Инициализация Ethernet контроллера

EthernetClient client;

void setup() {

Serial.begin(9600); // Устанавливаем скорость 9600 бит/сек

// устанавливаем Ethernet соединение:

Serial.println("Connecting...");

if (Ethernet.begin(mac) == 0) { //если контроллеру не удалось получить

//настройки по DHCP, выводим сообщение

Serial.println("Failed to configure Ethernet using DHCP");

Ethernet.begin(mac, ip); //пытаемся настроить IP адрес вручную

}

delay(1000);

Serial.print("IP address: ");

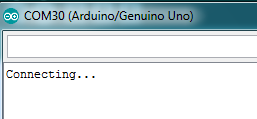
Serial.println (Ethernet.localIP()); //выводим текущий IP адрес устройства

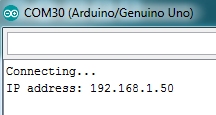
}

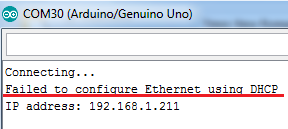
void loop() {

}

После загрузки и запуска программы, откройте Монитор порта и проверьте выведенное сообщение.

Сначала должна появиться строка Connecting…

Затем, если в сети используется сервис DHCP, то в течение нескольких секунд должно появиться сообщение об IP адресе, присвоенном устройству.

В противном случае, в течение приблизительно одной минуты контроллер будет пытаться найти сервис DHCP, затем выведется сообщение об его отсутствии и будет использоваться IP адрес, указанный в программном коде.

Примечание: для правильной работы, убедитесь, что MAC адрес является уникальным в пределах локальной сети, а IP адрес верен.

## Получение и отправка данных в Интернет

Рассмотрим пример получения и отправки данных на примере обращения к серверу прогноза погоды (samples.openweathermap.org).

**Программный код**

/\*

\* Lesson6\_2

\*Пример получения и отправки данных

\*на сервер прогноза погоды

\*/

#include <SPI.h>

#include <Ethernet.h>

// Укажите MAC адресс вашего контроллера ниже

// В СЕТИ НЕ ДОЛЖНО БЫТЬ НЕСКОЛЬКО УСТРОЙСТВ С ОДИНАКОВЫМ MAC АДРЕСОМ

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

// Укажите статический IP адрес, если в сети не используется DHCP сервис

char server[] = "samples.openweathermap.org"; // адрес сервера прогноза погоды

IPAddress ip(192, 168, 1,211);

// Инициализация Ethernet контроллера

EthernetClient client;

void setup() {

Serial.begin(9600); // Устанавливаем скорость 9600 бит/сек

// устанавливаем Ethernet соединение:

Serial.println("Connecting...");

if (Ethernet.begin(mac) == 0) { //если контроллеру не удалось получить

//настройки по DHCP, выводим сообщение

Serial.println("Failed to configure Ethernet using DHCP");

Ethernet.begin(mac, ip); //пытаемся настроить IP адрес вручную

}

delay(1000);

Serial.print("IP address: ");

Serial.println (Ethernet.localIP()); //выводим текущий IP адрес устройства

// если смогли подключиться к серверу, то отправляем запрос:

if (client.connect(server, 80)) {

Serial.println("connected");

// формируем HTTP запрос:

const String html\_cmd1 = "GET /data/2.5/forecast?id=524901&appid=b1b15e88fa797225412429c1c50c122a1 HTTP/1.1";

const String html\_cmd2 = "Host: samples.openweathermap.org";

const String html\_cmd3 = "Connection: close";

// выполняем HTTP запрос:

client.println(html\_cmd1);

client.println(html\_cmd2);

client.println(html\_cmd3);

client.println();

} else {

// если не смогли подключиться к серверу:

Serial.println("connection failed");

}

}

void loop() {

// выводим на экран все, что получили от сервера

if (client.available()) {

char c = client.read();

Serial.print(c);

}

// если произошло отключение от сервера, останавливаем клиент:

if (!client.connected()) {

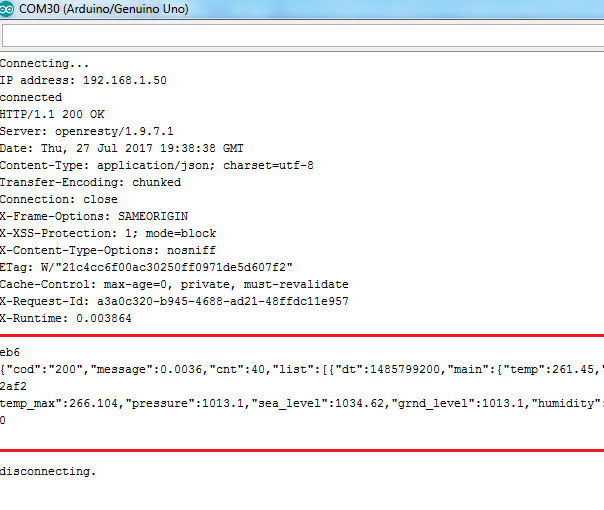
Serial.println();

Serial.println("disconnecting.");

client.stop();

}

}

После загрузки программы, откройте Монитор порта. В нем должны выводиться данные о прогнозе погоды, полученные с сервера samples.openweathermap.org .