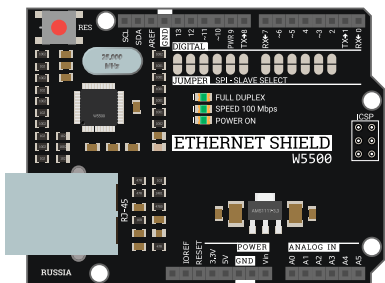


# Ethernet Shield W5500 для подключения к сети Ethernet и Internet



## Общие сведения:

*Ethernet Shield W5500* — плата расширения, которая позволяет осуществить удалённое подключение плат Arduino к сети Ethernet, а следовательно и к сети Интернет (при наличии подключения).

## Спецификация:

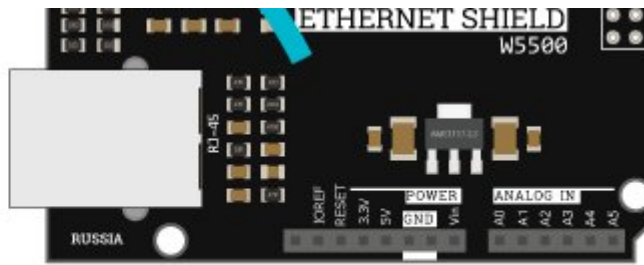
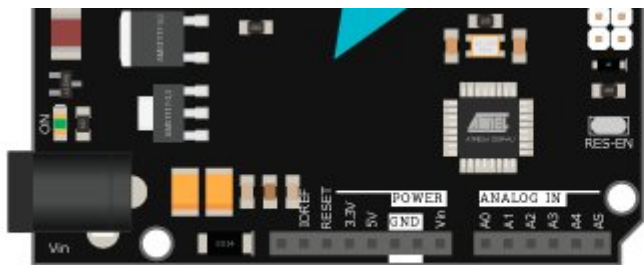
- Напряжение питания: 5В;
- Интерфейс: SPI;
- Сетевые протоколы: TCP, UDP;
- Сетевые интерфейсы: 10Мбит / 100Мбит;
- Размер внутреннего буфера: 32 кБайт;
- Режимы работы: дуплекс / полудуплекс;
- Светодиодная индикация;

## Подключение:

[Ethernet Shield W5500](#) совместим с различными микроконтроллерами Arduino, его можно установить на Arduino UNO, Leonardo, Mega.

[Ethernet Shield W5500](#) устанавливается сверху на управляющую плату.





## Подробнее о Ethernet Shield:

Ethernet Shield W5500 может стать вашим пропуском как просто в локальную или глобальную сеть, так и в мир интернета вещей, за счёт того, что позволит работать с большим количеством плат Arduino, которые вы можете расположить где только пожелаете.

Данный Shield может похвастаться сразу несколькими удачными техническими решениями, о которых мы и поговорим ниже.

### Регулятор напряжения

Наличие данного регулятора позволит обеспечить стабильным питанием чип W5500, установленный на плате расширения. Обычно данный чип питают напрямую от вывода напряжения 3.3В на плате Arduino, но при просадке напряжения на самой Arduino чип, расположенный на Shield'у может начать работать не корректно. При питании чипа через регулятор такое точно не произойдёт, так как на вход регулятора подаётся напряжение 5В, которое уже преобразуется в сильные стабильные 3.3В.

### Колодка SPI

Shield работает по шине SPI, а значит использует зарезервированные на этот случай выводы Arduino (для каждой платы - свои). Однако, все платы Arduino имеют специальную колодку с подписью ICSP, на которую выведены запараллеленные выводы шины SPI.

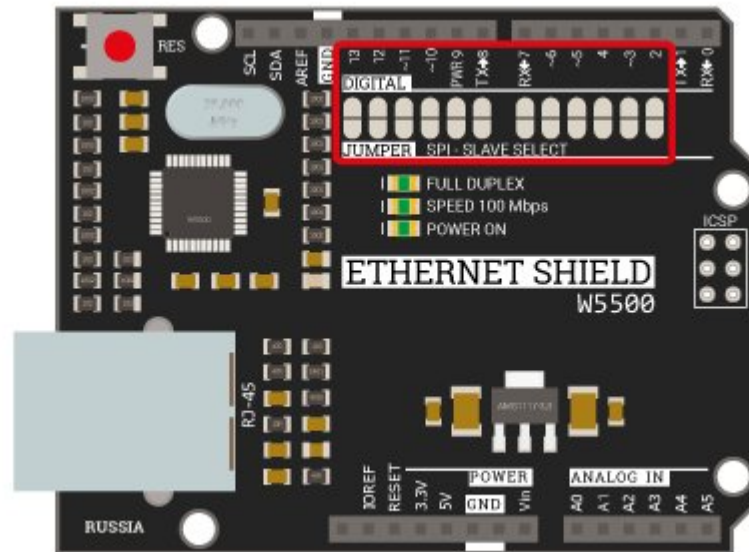
Ethernet Shield W5500 также имеет отдельную колодку, куда вынесены эти выводы, за счёт этого он легко устанавливается на большую часть плат Arduino/Piranha: UNO, ULTRA, MEGA, LEONARDO. Достаточно просто установить плату расширения сверху и всё, можно работать!

### Перемычки

Выводы, которые выделены под шину SPI, зарезервированы на управляющих платах следующим образом:

- для Arduino UNO/ULTRA: MOSI - 11, MISO - 12, SCK - 13, SS - 10;
- для Arduino MEGA: MISO - 50, MOSI - 51, SCK - 52, SS - 53;

Однако, плата расширения Ethernet Shield позволяет изменить вывод SS на любой свободный цифровой вывод. Для этого на корпусе платы есть переключки, на которых запаяна перемычка напротив 10 вывода. При необходимости (другое оборудование уже подключено к шине SPI либо к выводу 10) можно эту перемычку разомкнуть, а напротив нужного вывода запаять, тем самым изменив вывод SS (Select Slave) на нужный вам.



## MAC адрес

**MAC адрес** (Media Access Control, управление доступом к среде) - уникальный шести байтовый идентификатор устройств сети Ethernet.

Библиотека Arduino IDE позволяет устанавливать любой MAC адрес для Ethernet Shield W5500. Хотя главное требование чтобы в локальной сети он был уникален, сетевое оборудование может не распознавать Ethernet Shield если старший байт MAC адреса неправильно настроен. В примерах ниже используется следующий MAC адрес:

| 0xDE         | 0xAD | 0xBE | 0xEF | 0xFE | 0xED         |
|--------------|------|------|------|------|--------------|
| Старший байт | -    | -    | -    | -    | Младший байт |

В случае, если необходимо настаивать несколько устройств в сети рекомендуется менять младший байт.

## Примеры:

### Проверка наличия Ethernet-соединения

```
// Подключаем стандартную библиотеку для работы с Shield'ом по шине SPI
#include "SPI.h"
// Подключаем стандартную библиотеку для работы с Ethernet
#include "Ethernet.h"

void setup() {
  // Иницируем работу с монитором последовательного порта на скорости 9600 бод
  Serial.begin(9600);
}

void loop() {
  // Запрос статуса соединения
  auto link = Ethernet.linkStatus();
  // Вывод текста в монитор последовательного порта
  Serial.print("Link status: ");
  // Проверяем значение переменной link
  switch (link) {
    // Если "Статус: неизвестно", то
    case Unknown:
      // выводим текст "Unknown" в монитор последовательного порта
      Serial.println("Unknown");
      break;
    // Если "Статус: подключение есть", то
```

```

case LinkON:
    // выводим текст "ON" в монитор последовательного порта
    Serial.println("ON");
    break;
// Если "Статус: подключения нет", то
case LinkOFF:
    // выводим текст "OFF" в монитор последовательного порта
    Serial.println("OFF");
    break;
}
// Задержка в 1 секунду
delay(1000);
}

```

## Вывод IP-адреса, назначенного модулю

\*\*

\*\*На заметку:\*\* В случае если на маршрутизаторе выключен DHCP или Вы хотите задать статический адрес Shield'у, необходимо знать IP адрес маршрутизатора и маску подсети. В нашем случае адрес маршрутизатора 192.168.0.1 и маска подсети 255.255.255.0, это означает что Shield'у можно задать адреса в диапазоне от 192.168.0.2 до 192.168.0.254

```

// Подключаем стандартную библиотеку для работы с Shield'ом по шине SPI
#include "SPI.h"
// Подключаем стандартную библиотеку для работы с Ethernet
#include "Ethernet.h"

// Задаём статический IP-адрес на тот случай, если динамическое присвоение адреса даст сбой
IPAddress ip(192, 168, 0, 177);
IPAddress myDns(192, 168, 0, 1);
// Создаём объект client класса EthernetClient

```

```
EthernetClient client;
// Задаём MAC-адрес устройству. Главное, чтобы в сети не было уже зарегистрировано устройство с таким адресом
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

void setup() {
  // Иницилируем работу с монитором последовательного порта на скорости 9600 бод
  Serial.begin(9600);
  // Выводим в монитор последовательного порта сообщение о начале инициализации Shield'a
  Serial.println("Initialize Ethernet with DHCP:");
  // Если соединение с динамической адресацией не было установлено, то
  if (Ethernet.begin(mac) == 0) {
    // Выводим сообщение об этом в монитор последовательного порта и
    Serial.println("Failed to configure Ethernet using DHCP");
    // проверяем наличие самого Shield'a
    if (Ethernet.hardwareStatus() == EthernetNoHardware) {
      // Если Shield не найден, то выводим соответствующее сообщение в монитор порта
      Serial.println("Ethernet shield was not found. Sorry, can't run without hardware. :( ");
      // Ничего больше не выполняется
      while (true) {
        delay(1); // do nothing, no point running without Ethernet hardware
      }
    }
    // Проверяем наличие соединения
    if (Ethernet.linkStatus() == LinkOFF) {
      // Если соединение не установлено, то выводим соответствующее сообщение в монитор порта
      Serial.println("Ethernet cable is not connected.");
    }
    // Пробуем установить соединение, используя статический IP-адрес
    Ethernet.begin(mac, ip, myDns);
  }
  // Если соединение с динамической адресацией было установлено, то
  else {
    // Выводим в монитор порта соответствующее сообщение об этом и выводим назначенный устройству IP-адрес
```

```
Serial.print(" DHCP assigned IP ");
Serial.println(Ethernet.localIP());
}
}

void loop() {}
```

## Запрос на сервер и вывод полученных данных в монитор последовательного порта

```
// Подключаем стандартную библиотеку для работы с Shield'ом по шине SPI
#include "SPI.h"
// Подключаем стандартную библиотеку для работы с Ethernet
#include "Ethernet.h"

// Задаём статический IP-адрес на тот случай, если динамическое присвоение адреса даст сбой
IPAddress ip(192, 168, 0, 177);
IPAddress myDns(192, 168, 0, 1);
// Создаём объект client класса EthernetClient
EthernetClient client;
// Задаём MAC-адрес устройству. Главное, чтобы в сети не было уже зарегистрировано устройство с таким адресом
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

// Указываем адрес сервера
char server[] = "www.arduino.cc";
//IPAddress server(64,131,82,241);

// Счётчик времени с момента последнего подключения к серверу (в миллисекундах)
unsigned long lastConnectionTime = 0;
// Время между отправкой запросов на сервер (в миллисекундах)
const unsigned long postingInterval = 10 * 1000;

void setup() {
```



```
// Иницируем работу с монитором последовательного порта на скорости 9600 бод
Serial.begin(9600);
// Выводим в монитор последовательного порта сообщение о начале инициализации Shield'a
Serial.println("Initialize Ethernet with DHCP:");
// Если соединение с динамической адресацией не было установлено, то
if (Ethernet.begin(mac) == 0) {
  // Выводим сообщение об этом в монитор последовательного порта и
  Serial.println("Failed to configure Ethernet using DHCP");
  // проверяем наличие самого Shield'a
  if (Ethernet.hardwareStatus() == EthernetNoHardware) {
    // Если Shield не найден, то выводим соответствующее сообщение в монитор порта
    Serial.println("Ethernet shield was not found. Sorry, can't run without hardware. :( ");
    // Ничего больше не выполняется
    while (true) {
      delay(1); // do nothing, no point running without Ethernet hardware
    }
  }
  // Проверяем наличие соединения
  if (Ethernet.linkStatus() == LinkOFF) {
    // Если соединение не установлено, то выводим соответствующее сообщение в монитор порта
    Serial.println("Ethernet cable is not connected.");
  }
  // Пробуем установить соединение, используя статический IP-адрес
  Ethernet.begin(mac, ip, myDns);
}
// Если соединение с динамической адресацией было установлено, то
else {
  // Выводим в монитор порта соответствующее сообщение об этом и выводим назначенный устройству IP-адрес
  Serial.print(" DHCP assigned IP ");
  Serial.println(Ethernet.localIP());
}
}
```

```
void loop() {
  // Если получены данные, то
  if (client.available()) {
    // читаем их в переменную и выводим в виде бинарного кода в монитор порта
    char c = client.read();
    Serial.write(c);
  }
  // Если время между отправкой запросов истекло, то отправляется новый запрос
  if (millis() - lastConnectionTime > postingInterval) {
    httpRequest();
  }
}

// Функция отправки запроса серверу
void httpRequest() {
  // Останавливаем все соединения, которые есть в данный момент
  client.stop();

  // Устанавливаем новое соединение с сервером через 80 порт, и если соединение установлено, то
  if (client.connect(server, 80)) {
    // Выводим в монитор порта соответствующее сообщение
    Serial.println("connecting...");
    // Делаем GET-запрос на сервер
    client.println("GET /latest.txt HTTP/1.1");
    client.println("Host: www.arduino.cc");
    client.println("User-Agent: arduino-ethernet");
    client.println("Connection: close");
    client.println();
    // Обновляем счётчик отправки запроса
    lastConnectionTime = millis();
  }
  // Если соединение с сервером не установлено, то
  else {
```

```
// выводим соответствующее сообщение в монитор порта
Serial.println("connection failed");
}
}
```

## Применение:

- Создание системы "Умный дом" и её аналогов;
- Объединение нескольких плат Arduino в локальную или глобальную сеть;