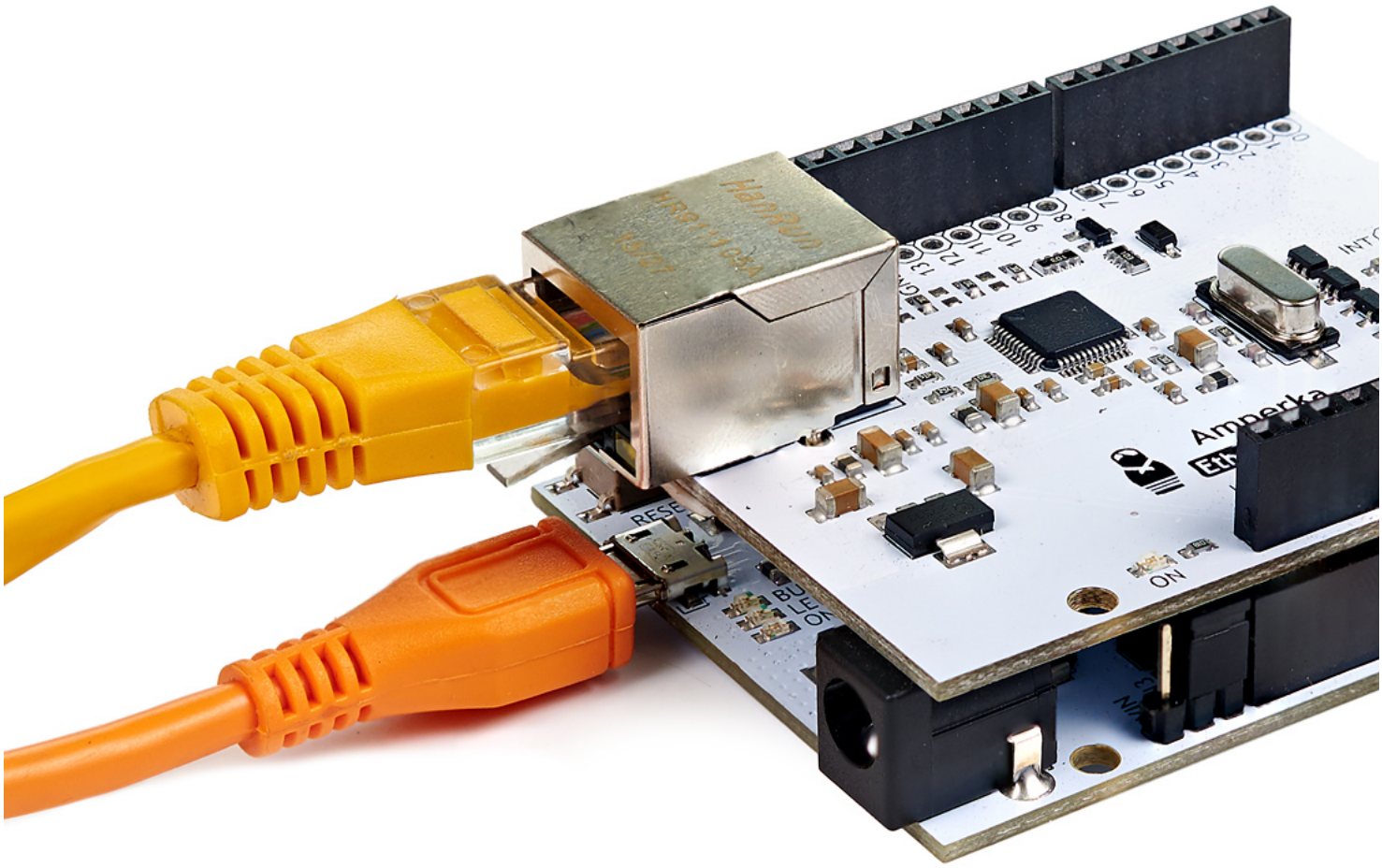
амперка

## Ethernet Shield

Ethernet Shield [https://amperka.ru/product/arduino-ethernet-shield?utm_source=man&utm_campaign=ethernet-shield&utm_medium=wiki] — это плата расширения, позволяющая Arduino [https://amperka.ru/collection/arduino?utm_source=man&utm_campaign=ethernet-shield&utm_medium=wiki] или Iskra [https://amperka.ru/collection/iskra?utm_source=man&utm_campaign=ethernet-shield&utm_medium=wiki] работать в локальных вычислительных сетях для приёма и передачи данных в сети Интернет.

> На платы с разъёмом USB-B шилд может встать не ровно. Контакты разъёма ethernet прижимаются к металлическому разъёму микроконтроллера. Чтобы всё работало стабильно приклейте кусочек изоленты на разъём между шилдом и микроконтроллером или используйте специальные проставки [https://amperka.ru/product/stackable-pin-headers-with-icsp]




Шилд Ethernet позволит управлять удалёнными объектами через web-браузер со своего компьютера, планшета или телефона.

## Подключение и настройка

1. Установите Ethernet Shield на управляющую платформу, например Arduino [https://amperka.ru/collection/arduino?utm_source=man&utm_campaign=trafficlight&utm_medium=wiki], Iskra Neo [https://amperka.ru/product/iskra-neo?utm_source=man&utm_campaign=trafficlight&utm_medium=wiki] или Iskra JS. [https://amperka.ru/product/iskra-js?utm_source=man&utm_campaign=trafficlight&utm_medium=wiki]

> Убедитесь в наличии и правильности соединения джамперов [https://amperka.ru/product/jumper_pins_x10?utm_source=man&utm_campaign=trafficlight&utm_medium=wiki] SPI-интерфейса на плате Ethernet Shield.

2. Подключите кабель Ethernet в разъём RJ45S.

## Примеры работы для Iskra JS

Для общения Ethernet Shield с платой Iskra JS воспользуемся библиотекой WIZnet [http://www.espruino.com/WIZnet]. Она скрывает в себе все тонкости протокола, предоставляя простые и понятные функции.

1.
GET-запрос по URL-адресу в Интернете.

```
// Настраиваем соединение с Ethernet Shield по протоколу SPI.
SPI2.setup({baud: 3200000, mosi: B15, miso: B14, sck: B13});
var eth = require('WIZnet').connect(SPI2, P10);
// Подключаем модуль http.
var http = require('http');

// Получаем и выводим IP-адрес от DHCP-сервера
eth.setIP();
print(eth.getIP());

// Производим запрос к сайту
http.get('http://amperka.ru', function(res) {
  res.on('data', function(data) {
    print(data);
  });
});
```

2.
HTTP-сервер на порту 8080.

```
// Настраиваем соединение с Ethernet Shield по протоколу SPI.
SPI2.setup({baud: 3200000, mosi: B15, miso: B14, sck: B13});
var eth = require('WIZnet').connect(SPI2, P10);
// Получаем и выводим IP-адрес от DHCP-сервера
eth.setIP();
print(eth.getIP());

require("http").createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.write('Hello World');
  res.end();
}).listen(8080);
```

## Примеры работы для Arduino

Для общения Ethernet Shield с платами Arduino воспользуемся библиотекой Ethernet 2. Она скрывает в себе все тонкости протокола, предоставляя простые и понятные функции.

### Пример WebClient

GET-запрос по URL-адресу в Интернете.

**webClient.ino**

```
// библиотека для работы с SPI
#include <SPI.h>
// библиотека для работы с Ethernet Shield
#include <Ethernet2.h>

// MAC-адрес контроллера
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

// адрес запрашиваемого сервера
char server[] = "amperka.ru";

// задаем статический IP-адрес
// на тот случай, если у DHCP выдать IP-адрес не получится
IPAddress ip(192, 168, 0, 177);

// создаём клиента, который будет подключаться
// к необходимому для нас серверу и порту
// портом по умолчанию для HTTP является 80
EthernetClient client;

void setup()
{
  // открываем последовательный порт
  Serial.begin(9600);
  // ждём, пока не откроется монитор последовательного порта
  // для того, чтобы отследить все события в программе
  while (!Serial) {
  }
  Serial.print("Serial init OK\r\n");

  // запускаем Ethernet-соединение:
  if (Ethernet.begin(mac) == 0) {
    // если не удалось сконфигурировать Ethernet при помощи DHCP
    Serial.println("Failed to configure Ethernet using DHCP");
    // продолжать дальше смысла нет, поэтому вместо DHCP
    // попытаемся сделать это при помощи IP-адреса:
    Ethernet.begin(mac, ip);
  }
  // получаем  и выводим локальный IP адрес
  Serial.print("My IP address: ");
  Serial.println(Ethernet.localIP());
  // даем Ethernet 1 секунду на инициализацию
  delay(1000);
```

```
  Serial.println("connecting...");

  // если подключение установлено, сообщаем об этом на Serial-порт:
  if (client.connect(server, 80)) {
    Serial.println("connected");
    // формируем HTTP-запрос
    client.println("GET / HTTP/1.1");
    client.println("Host: amperka.ru");
    client.println("Connection: close");
    client.println();
  } else {
    // если соединения с сервером нет, пишем об этом на Serial-порт:
    Serial.println("connection failed");
  }
}

void loop()
{
  // если есть непрочитанные байты
  // принятые клиентом от удаленного сервера, с которым установлено соединение
  if (client.available()) {
    // считываем данные и печатаем в Serial-порт
    char c = client.read();
    Serial.print(c);
  }

  // если сервер отключился
  if (!client.connected()) {
    // печатаем об этом в Serial-порт
    Serial.println();
    Serial.println("disconnecting.");
    // останавливаем работу клиента
    client.stop();
    // останавливаем программу в бесконечном цикле
    while (1) {
    }
   }
  }
```

### Пример WebServer

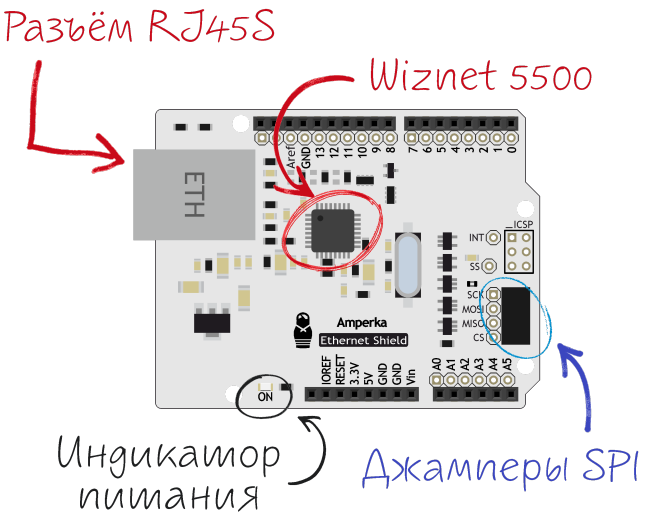
Создадим HTTP-сервер на порту 80, на который будем передавать значения всех аналоговых портов с `A0`-`A5`.

**webServer.ino**

```
// библиотека для работы с SPI
#include <SPI.h>
// библиотека для работы с Ethernet Shield
#include <Ethernet2.h>

// MAC-адрес контроллера
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
// задаем статический IP-адрес
IPAddress ip(192, 168, 88, 177);

// инициализируем библиотеку Ethernet Server
// с необходимыми вам IP-адресом и портом
// порт 80 используется для HTTP по умолчанию
EthernetServer server(80);

void setup()
{
  // открываем последовательный порт
  Serial.begin(9600);
  // ждём, пока не откроется монитор последовательного порта
  // для того, чтобы отследить все события в программе
  while (!Serial) {
  }
  Serial.print("Serial init OK\r\n");
  // запускаем Ethernet-соединение:
  if (Ethernet.begin(mac) == 0) {
    // если не удалось сконфигурировать Ethernet при помощи DHCP
    Serial.println("Failed to configure Ethernet using DHCP");
    // продолжать дальше смысла нет, поэтому вместо DHCP
    // попытаемся сделать это при помощи IP-адреса:
    Ethernet.begin(mac, ip);
  }
  // запускаем сервер и выводим локальный IP адрес
  server.begin();
  Serial.print("Server is at ");
  Serial.println(Ethernet.localIP());
}


void loop()
{
  // слушаем подключающихся клиентов
  EthernetClient client = server.available();
  if (client) {
    // выводим сообщение о новом клиенте
    Serial.println("new client");
    // HTTP-запрос заканчивается пустой линией
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        // если добрались до конца строки (т.е. получили символ новой строки),
```

```
            // и эта строка - пустая, это значит, что это конец HTTP-запроса.
            // то есть, можно приступать к отправке ответа:
            if (c == '\n' && currentLineIsBlank) {
              // отсылаем стандартный заголовок для HTTP-ответа:
              client.println("HTTP/1.1 200 OK");
              client.println("Content-Type: text/html");
              // после выполнения ответа соединение будет разорвано
              client.println("Connection: close");
              // автоматически обновляем страницу каждые 5 секунд
              client.println("Refresh: 5");
              client.println();
              client.println("<!DOCTYPE HTML>");
              client.println("<html>");
              // выводим значения ото всех входных аналоговых контактов:
              for (int analogChannel = 0; analogChannel < 6; analogChannel++) {
                int sensorReading = analogRead(analogChannel);
                client.print("Analog input ");
                client.print(analogChannel);
                client.print(" is ");
                client.print(sensorReading);
                client.println("<br />");
              }
              client.println("</html>");
              break;
            }
            if (c == '\n') {
              // начинаем новую строку
              currentLineIsBlank = true;
            } else if (c != '\r') {
              // в текущей строке есть символ:
              currentLineIsBlank = false;
            }
          }
        }
        // даем браузеру время, чтобы получить данные
        delay(1);
        // закрываем соединение
        client.stop();
        // клиент отключился
        Serial.println("Client disconnected");
      }
    }
```

## Элементы платы




### Микросхема Wiznet 5500

Чип wiznet 5500 — аппаратный контроллер TCP/IP, позволяющий легко подключиться к Интернету.

### Светодиодная индикация

| Имя светодиода | Назначение |
|---|---|
| ON | Информационный светодиод о наличии питания |

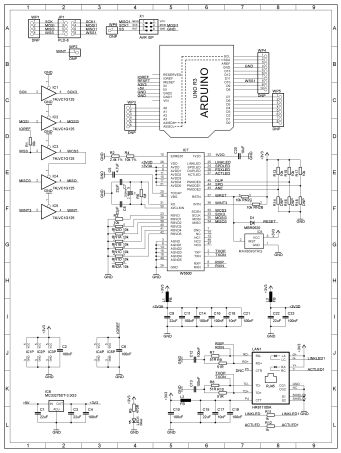
### Разъём RJ45S

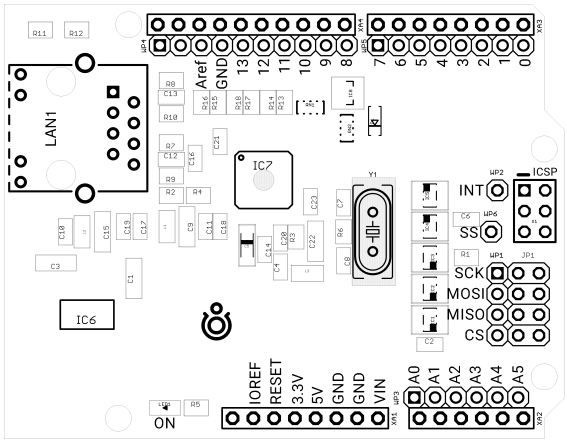
Стандартный разъём для подключения Ethernet-кабеля.

### Джамперы SPI

Коммутируют пины интерфейса SPI на Ethernet Shield и управляющей плате. Пины можно изменить, сняв джамперы и припаяв свободные металлические контакты к другим пинам с помощью проводков.

## Принципиальная и монтажная схемы






## Характеристики

- Поддерживаемые протоколы: TCP, UDP, ICMP, IPv4, ARP, IGMP, PPPoE
- Количество сокетов: 8
- Интерфейс управления: SPI
- Величина внутреннего буфера RX/TX: 32 кБ
- Напряжение питания: 3.3…5 В
- Скорость соединения: 10/100 Мбит
- Занимаемые пины: SPI (MISO, MOSI, SCK), 10
- Габариты: 69×53 мм (RJ45 выступает на несколько мм)

## Ресурсы

- Ethernet Shield [https://amperka.ru/product/arduino-ethernet-shield?utm_source=man&utm_campaign=arduino-ethernet-shield-v2&utm_medium=wiki] в магазине
- Векторное изображение шилда [https://github.com/amperka/hardware-drawings/blob/master/arduino-ethernet-shield-v2.svg]
- Datasheet на Wiznet5500
- Ethernet Shield в проекте «Виджет-светофор» [http://wiki.amperka.ru/projects:trafficlight]
- Ethernet Shield в проекте «Интерактивный лабиринт» [http://wiki.amperka.ru/projects:maze]