

Módulo Ethernet Arduino W5500

15 de maio de 2018

Luboš M.

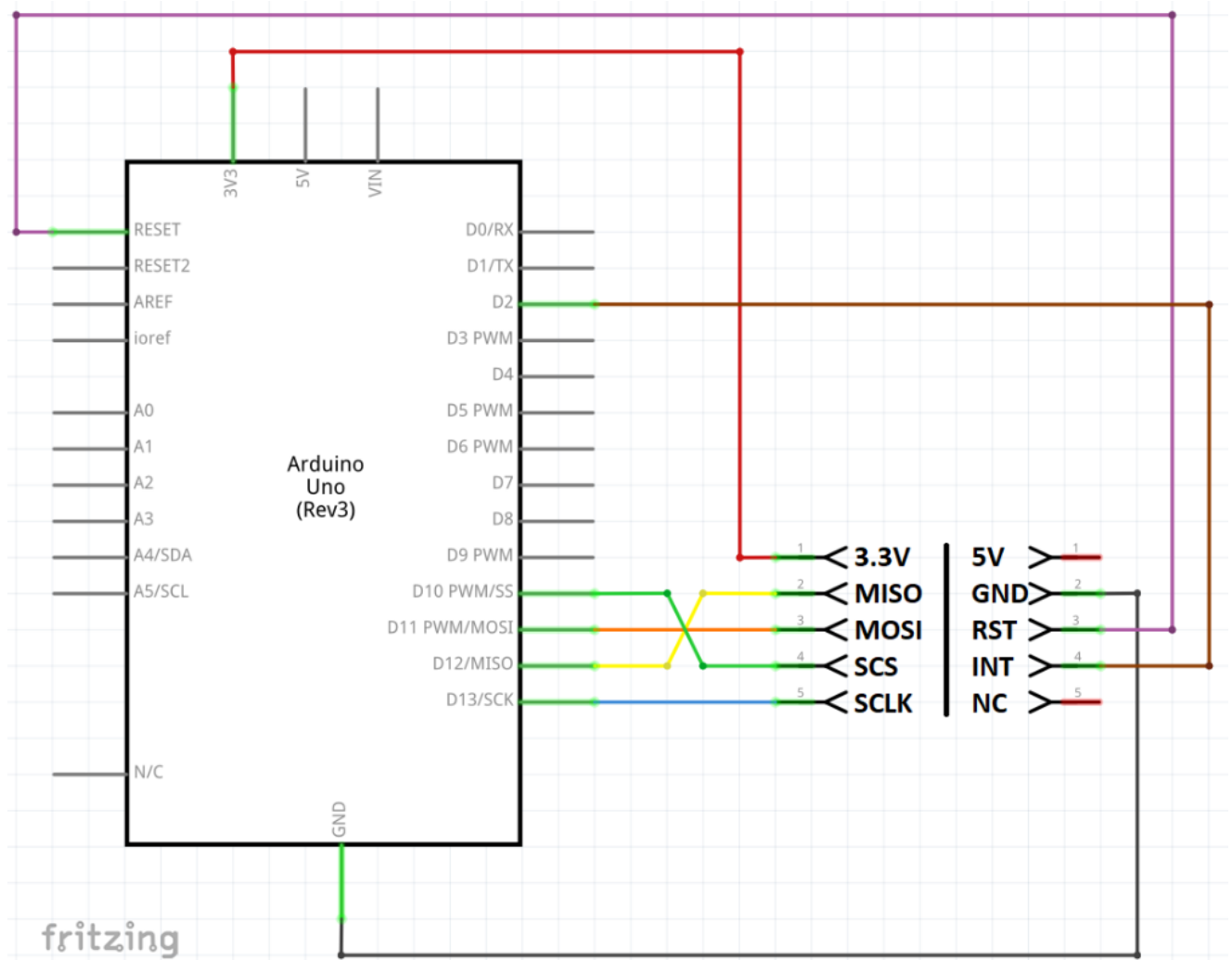
[Voltar à lista de artigos \(/navody-k-produktum/\)](/navody-k-produktum/)

O módulo de rede Ethernet W5500 é um circuito de comunicação adicional para o Arduino. Este circuito prático da WIZnet permite conectar o Arduino à sua rede doméstica ou à Internet. Além do circuito W5500, o módulo também possui um conector RJ45 para conectar um cabo de rede e componentes adicionais para a operação adequada. Também digno de nota é um conversor de tensão de 5 Volts a 3,3 Volts. O circuito W5500 funciona com tensão de 3,3 V, mas seus pinos de dados também podem se comunicar em níveis de 5 V. Graças à presença do conversor, podemos usar o pino de alimentação de 5V ou 3.3V. O módulo lida com endereços IPv4 e se comunica a 10 ou 100 Mbps. A comunicação com o Arduino é via protocolo SPI.



Um total de 8 fios deve ser conectado para conectar com êxito o módulo Ethernet W5500 à placa Arduino. Conecte 3.3V com 3V3 Arduina, MISO com D12, MOSI com D11, SCS com D10, SCLK com D13, GND com terra de Arduina, RST com Arduina RESET e INT com pino D2.

Os pinos D10, D11, D12 e D13 são conectados aos pinos SPI da placa UNO usada. Se você deseja usar outra placa Arduino, é necessário usar seus pinos SPI (mais informações [aqui \(https://www.arduino.cc/en/Reference/SPI\)](https://www.arduino.cc/en/Reference/SPI)).



Para fazer upload do código de amostra com sucesso, você deve baixar e importar a biblioteca **Ethernet2** (</docs/texty/0/253/ethernet2.zip>), conforme descrito [aqui \(/zaciname-s-arduinoem/arduino-knihovny.html\)](/zaciname-s-arduinoem/arduino-knihovny.html). O código de amostra com um servidor da web contém as bibliotecas necessárias no início. Em seguida, defina seu próprio endereço MAC, que deve ser exclusivo para cada dispositivo, e defina o endereço IP, que deve ser atribuído pelo servidor DHCP ou definido manualmente pelo roteador. Podemos encontrar facilmente o intervalo usado localizando o endereço IP (por exemplo, 192.168.1.2) em qualquer computador conectado à mesma rede e configurando o endereço IP do Arduino vários endereços maiores (por exemplo, 192.168.1.50). Em seguida, basta inicializar o Ethernet Shield no modo de servidor com a porta padrão 80.

A sub-rotina de configuração contém as configurações de comunicação da linha serial, seguidas pela comunicação Ethernet Shield e o endereço IP definido pela linha serial.

Um loop infinito no início carrega primeiro os clientes conectados na variável do cliente e depois verifica se temos algum cliente. Se o cliente estiver conectado, imprimimos as informações na linha serial e depois enviamos todos os dados. Primeiro, enviamos uma resposta http padrão, na qual podemos definir o número de segundos para o cliente recarregar a página para o comando "Atualizar: 5". Também é importante imprimir nossos

próprios dados, onde neste programa imprimimos o tempo desde o início do Arduino e depois usamos as informações do loop for de todas as entradas analógicas. Após a impressão de toda a resposta, feche a conexão e repita o loop.

```
// módulo LAN W5500 - servidor web

// conexão bibliotecas necessárias
# incluem <SPI.h>
# incluem <Ethernet2.h>
// definir o MAC endereço
byte mac [] = {
    0xDE , 0xAD , 0xBE , 0xEF , 0xFE , 0xED
};
// Configuração do endereço IP - deve ser atribuível pelo servidor DHCP,
// no intervalo correto ou atribuído manualmente pelo
endereço IP do roteador ip ( 192 , 168 , 1 , 50 );
// inicialização do servidor na porta 80
EthernetServer server ( 80 );

void setup () {
    // inicializa a comunicação serial em 9600 baud
    Serial . begin ( 9600 );
    // habilita a comunicação com o Ethernet Shield
    Ethernet . begin (mac, ip);
    servidor. begin ();
    // imprime informações sobre o endereço IP
    serial definido . print ( "O servidor está no endereço IP:" );
    Serial . println ( Ethernet , localIP );
}

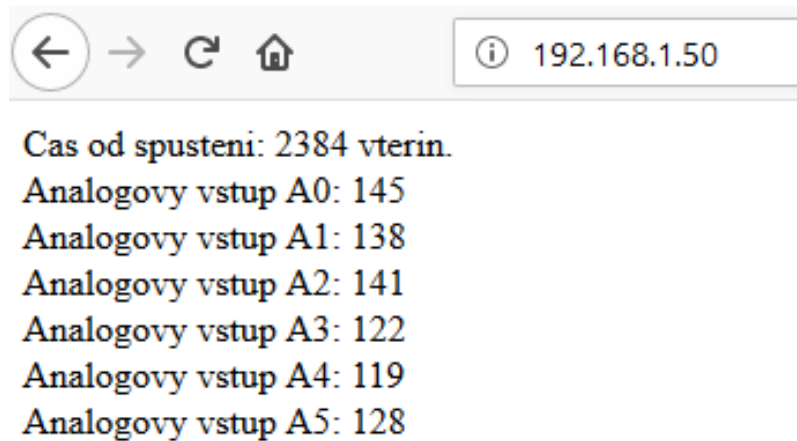
void loop () {
    // carrega clientes conectados
    EthernetClient client = server. disponível ();
    // se um cliente se conectar, faremos o seguinte
    se (cliente) {
        Serial . println ( "Novo cliente:" );
        // a solicitação http termina com uma linha em branco
        boolean currentLine = true;
        // se o cliente estiver conectado e disponível,
        imprimiremos os dados disponíveis
        while (client. connected ()) {
            if (client. available ()) {
                // recupera e imprime informações do cliente
                char c = client. read ();
                Serial . escreva (c);
                // se chegarmos ao final da linha e a próxima estiver vazia,
                // a solicitação foi encerrada e poderemos enviar uma resposta
                se (c == '\ n' && currentLine) {
                    // primeiro enviar uma resposta http padrão para o
                    cliente. println ( "HTTP / 1.1 200 OK" );
                    cliente. println ( "Tipo de conteúdo: texto / html" );
                }
            }
        }
    }
}
```

```

    cliente.println ( "Conexão: fechar" );
    // define a atualização automática da página após 5 segundos de
    cliente.println ( "Atualizar: 5" );
    cliente.println ();
    cliente.println ( "" );
    cliente.println ( "" );
    // place para imprimir seu próprio
    cliente de dados . print ( "Hora de inicialização:" );
    cliente. print ( millis () / 1000 );
    cliente. print ( "segundo. <br />" );
    // amostra da saída de dados de todas as entradas analógicas
    para ( int analogPin = 0 ; analogPin < 6 ; analogPin ++ ) {
        int analogData = analogRead (analogPin);
        cliente. print ( "Entrada analógica A" );
        cliente. print (analogPin);
        cliente. print ( ":" );
        cliente. print (analogData);
        // comando "<br />" atua como uma etapa para o próximo
        cliente de linha . println ( "<br />" );
    }
    cliente.println ( "" );
    quebrar ;
}
// inicia uma nova linha
se (c == '\ n' ) {
    currentLineLine = true;
    // detecção de caracteres na nova linha
} else if (c! = '\ r' ) {
    currentLineLine = false;
}
}
}
// pausa para o navegador lidar com todos os dados
delay ( 1 );
// fecha a conexão do
cliente. stop ();
Serial . println ( "Cliente desconectado." );
Serial . println ( "-----" );
}
}

```

Depois de carregar o código de amostra na placa Arduino com o módulo Ethernet W5500 conectado, obtemos, por exemplo, o seguinte resultado:



O módulo Ethernet Arduino W5500 é um elemento de rede interessante que nos permite conectar o Arduino à nossa rede doméstica. Após a conexão, podemos monitorar os dados medidos a partir de qualquer dispositivo localizado na mesma rede. Este módulo Ethernet pode ser usado, por exemplo, como um dispositivo de medição na parte da casa onde a rede LAN é distribuída. A grande vantagem também é que, graças ao excelente suporte do módulo W5100 mais antigo, foi criada a biblioteca Ethernet2 usada, que é 100% compatível com a biblioteca Ethernet mais antiga. Portanto, se você usou um módulo mais antigo em um projeto, basta baixar a nova biblioteca e altere seu nome <Ethernet.h> para <Ethernet2.h> no seu código.



Lista de componentes usados:

<http://arduino-shop.cz/arduino/974-arduino-uno-r3-atmega328p-1424115860.html>

(<http://arduino-shop.cz/arduino/974-arduino-uno-r3-atmega328p-1424115860.html>).

<https://arduino-shop.cz/arduino/1676-sitovy-ethernetovy-modul-w5500-tcp-ip-51-stm32-for-arduino-1500539901.html> (<https://arduino-shop.cz/arduino/1676-sitovy-ethernetovy-modul-w5500-tcp-ip-51-stm32-pro-arduino-1500539901.html>).

- [Ethernet2.zip \(/docs/texty/0/253/ethernet2.zip\)](#) (65,29 KB)

Outros artigos semelhantes



[\(/navody-k-produktum/modul-casoveho-zpozdeni-s-rele-a-led-displejem.html\)](#)

Módulo de retardo de tempo com relé e display LED ([/navody-k-produktum/modul-casoveho-zpozdeni-s-rele-a-led-displejem.html](#))

11. 11. 2019 Luboš M. Manuais do produto ([/navody-k-produktum/](#))

Este módulo simples pode ser usado para comutação de carga e contém 18 funções.

LEIA TODO ([/NAVODY-K-PRODUKTUM/MODUL-CASOVEHO-](#)

[ZPOZDENI-S-RELE-A-LED-DISPLEJEM.HTML](#))



[\(/navody-k-produktum/vyvojova-deska-esp32-cam.html\)](#)

Placa de desenvolvimento ESP32-CAM ([/navody-k-produktum/vyvojova-deska-esp32-cam.html](#))

27 de outubro de 2019 Luboš M. Manuais do produto ([/navody-k-produktum/](#))

Trabalhar com a câmera no ESP32 é uma variação interessante que este módulo transmitirá.

LEIA TODO ([/NAVODY-K-PRODUKTUM/VYVOJOVA-DESKA-ESP32-CAM.HTML](#))

Assine notícias e siga novos artigos

Váš e-mail

ENVIAR

☐ Concordo com o processamento de dados pessoais ([mais \(/zpracovani-osobnich-udaju.htm\)](https://www.arduino-shop.cz/zpracovani-osobnich-udaju.htm))



Copyright ECLIPSERA sro Jihlavská 1985, Havlíčkův Brod Todos os direitos reservados

[project.cz/sluzby/tvorba-webov.html](https://navody.arduino-shop.cz/sluzby/tvorba-webov.html)

Sistema de gerenciamento de conteúdo