

Porto Alegre, setembro de 2019

APLICAÇÃO SERVIDOR-CLIENTE: **MANUAL DE OPERAÇÃO**

APRESENTAÇÃO

Este manual tem por objetivo, demonstrar na prática, a utilização da placa Arduino em conjunto com outros módulos de expansão, também conhecido como Shields. Para os exercícios a seguir, utilizaremos especificamente o shield de Ethernet (ver figura 2). Os exercícios propostos, foram elaborados de maneira que estudantes e hobistas possam tirar o máximo de proveito da estrutura montada para fins didáticos, além é claro de, de contribuir para uma aprendizagem significativa que visa oportunizar um gradual aumento das habilidades técnicas e desenvolvimento do raciocínio lógico para programação, então é essencial que você procure seguir todas as etapas em seqüência, sem pular nenhuma delas. Bom estudo!

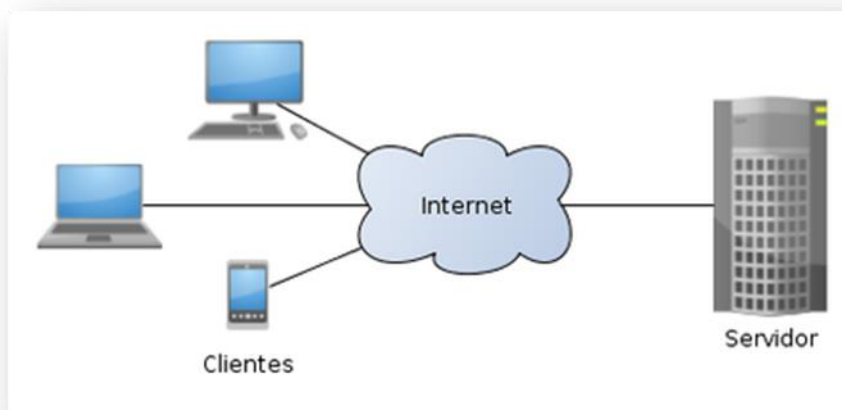
Nota:

Caso ocorra algum problema ou falha durante a instalação do programa ou execução dos exercícios, entre em contato com e-mail: (projeto-arduino@outlook.com).

ESTRUTURA BÁSICA CLIENTE-SERVIDOR

Para que você possa compreender melhor, o que venha ser uma estrutura cliente – servidor imagine uma grande teia de aranha. Imaginou? Pois bem, agora continue imaginando que em cada extremidade desta teia, encontra-se um computador, e que no meio desta teia tenha outro computador, muito maior em tamanho, capacidade de memória e etc. Se você conseguiu visualizar está teia com tudo que foi apontado até aqui, parabéns. Você já sabe o que é uma estrutura cliente-servidor na forma mais simples possível. É claro que ainda existem outros detalhes técnicos que também são importantes que você saiba, mas que para o nosso desenvolvimento, utilizando este material não serão necessários.

Por curiosidade não é apenas um único computador que gerencia todas as informações no sistema, aquele computador que você visualizou na sua imaginação que fica no centro da teia, lembra?! Mas sim um aglomerado de máquinas com capacidade de processamento surpreendente e que não são acessíveis a usuários comuns.

Figura 1 – Funcionamento estrutura cliente – servidor.

Fonte: Disponível em: < <https://images.app.goo.gl/dTt6nKjZSRr9E37o6> >. Acesso em: 30 out. 2019.

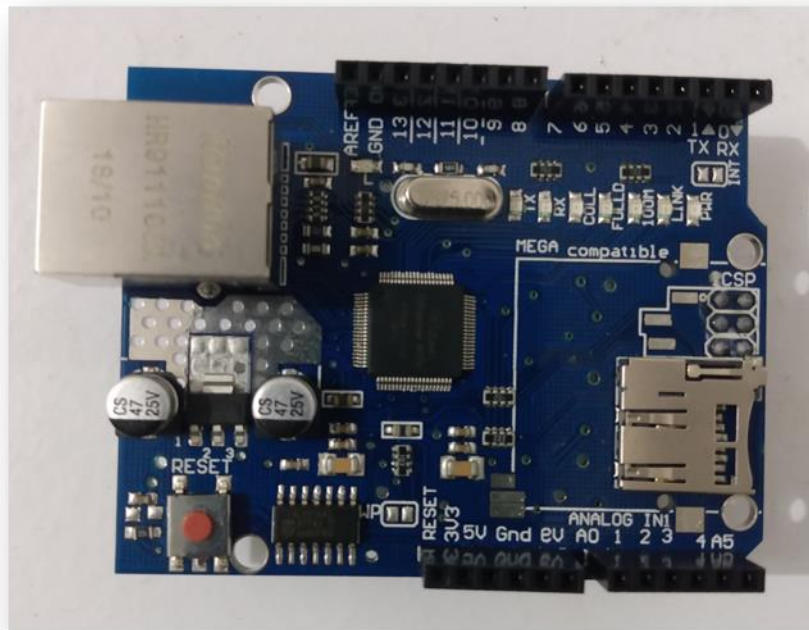
Por tanto um servidor recebe e transmite suas informações, quando solicitados pelos clientes. Estas informações são diversificadas, pode ser a solicitação de uma página da web, visualização de foto, áudio e vídeo.

Um bom ponto de partida seria o site da NIC.br <https://www.nic.br/> e o site da Abranet <http://www.abranet.org.br/?UserActiveTemplate=site>.

ESPECIFICAÇÕES TÉCNICAS

Características do Shield Ethernet:

- Ethernet Shield R3 W5100;
- Controlador Ethernet: W5100 com buffer interno 16K;
- Tensão de operação 5V (fornecida pela placa Arduino);
- Velocidade de conexão: 10 / 100Mb;
- Protocolos Suportados: TCP / IP, UDP, ICMP, ARP IPv4, IGMP, PPPoE, Ethernet;
- Suporte (Full-duplex e half-duplex);
- Suporte a conexão ADSL (PPPoE com PAP / CHAP no modo de autenticação);
- Suporte a 4 conexões independentes simultaneamente;
- Memória Interna: 16Kb para buffers de Tx / Rx;
- Conexão com o Arduino através de SPI.
- Dimensões(CxLxA): 68x53x10mm;

Figura 2 – Placa Ethernet Shield.

Fonte: Elaboração do autor

Características da placa Arduino:

- Modelo da placa: Uno R3;
- Controlador: ATmega328p;
- Tensão de operação: 5VDC;
- Velocidade do Clock: 16MHz;
- Tensão de alimentação recomendada (externa): 7 - 12VDC;
- Limite de tensão de alimentação (externa): 6 - 20VDC;
- Portas digitais (I/O): 14 (das quais 6 oferecem PWM);
- Entradas analógicas: 6;
- Corrente DC por porta digital (I/O): 40mA;
- Corrente DC por entrada analógica: 50mA;
- Memória Flash: 32kb (ATmega328p) dos quais 0,5Kb são utilizados pelo carregador de inicialização;
- SRAM: 2Kb (ATmega328p);
- EEPROM: 1Kb (ATmega328p);
- Velocidade do Clock: 16MHz;
- Dimensões: 53mm(L) X 13mm(A) X 68mm(C).

Figura 3 – Placa Arduino Uno.

Fonte: Elaboração do autor

MONTAGEM

Para efetuar corretamente a instalação e configuração da atividade proposta neste manual, siga atentamente as instruções abaixo.

1º passo: Conecte o cabo USB na placa Arduino.

Figura 4 – Placa Arduino Uno.

Fonte: Elaboração do autor

2º passo: Com a IDE ¹do Arduino aberta no seu computador, faça o “UPLOAD” do sketch para placa Arduino.

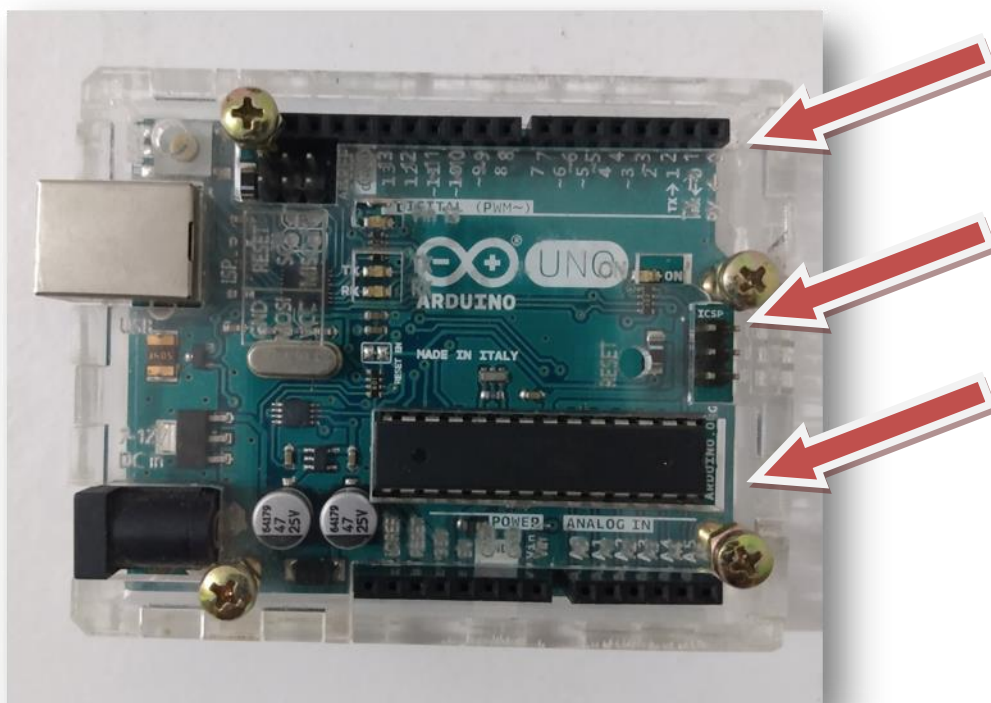
Figura 5 – Tela Principal de programação.



Fonte: Elaboração do autor

3º passo: Após, upload ser carregado, retire o cabo USB e conecte o SHIELD ETHERNET na placa Arduino. Observe com cuidado os locais aonde você deve conectar o shield na placa.

Figura 6 – Indicações para local de fixação do Shield.

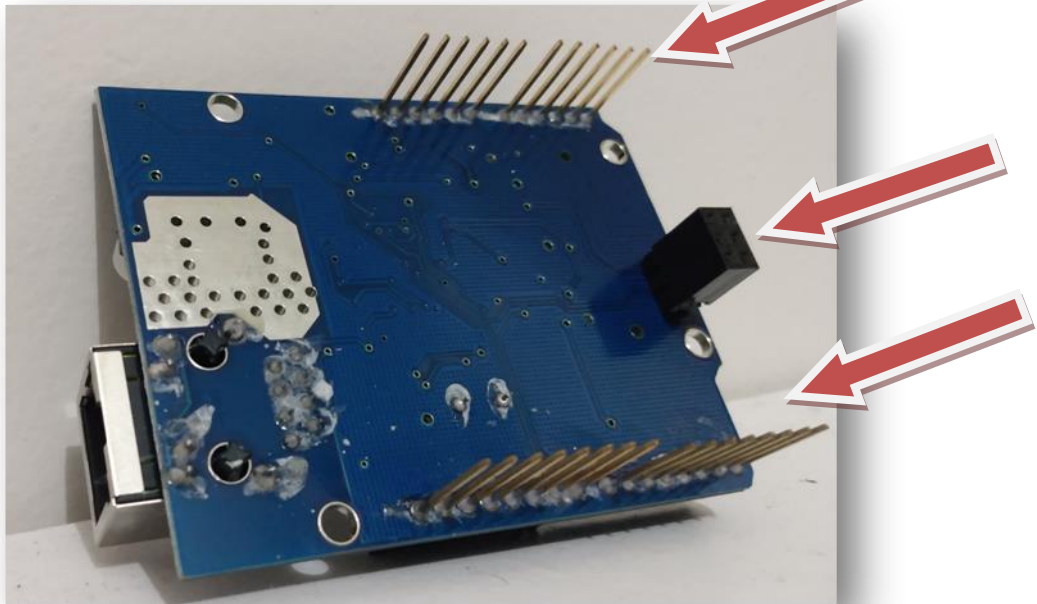


Fonte: Elaboração do autor

¹ Integrated Drive Electronics

4º passo: Também observe os com atenção, os locais da instalação do shield ethernet.

Figura 7 – Local de fixação do Shield Ethernet



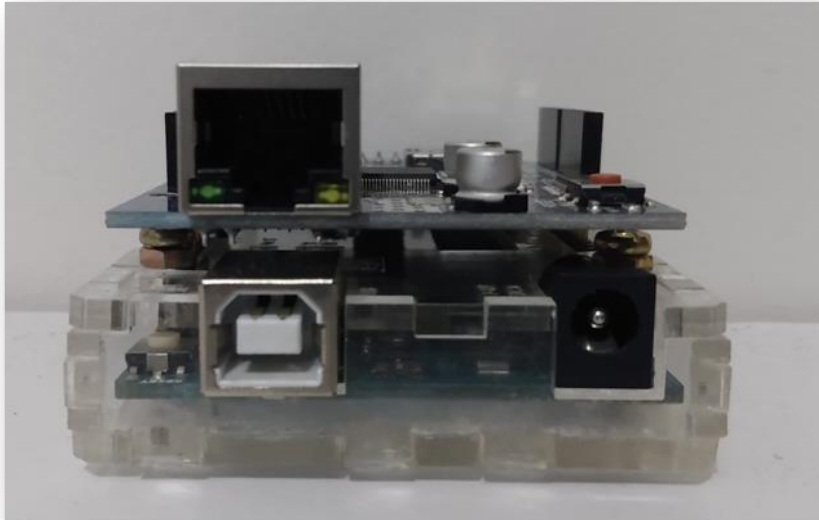
Fonte: Elaboração do autor

5º passo: Conecte as placas até que formem o conjunto como a imagem abaixo.

Figura 8 – Conjunto finalizado.



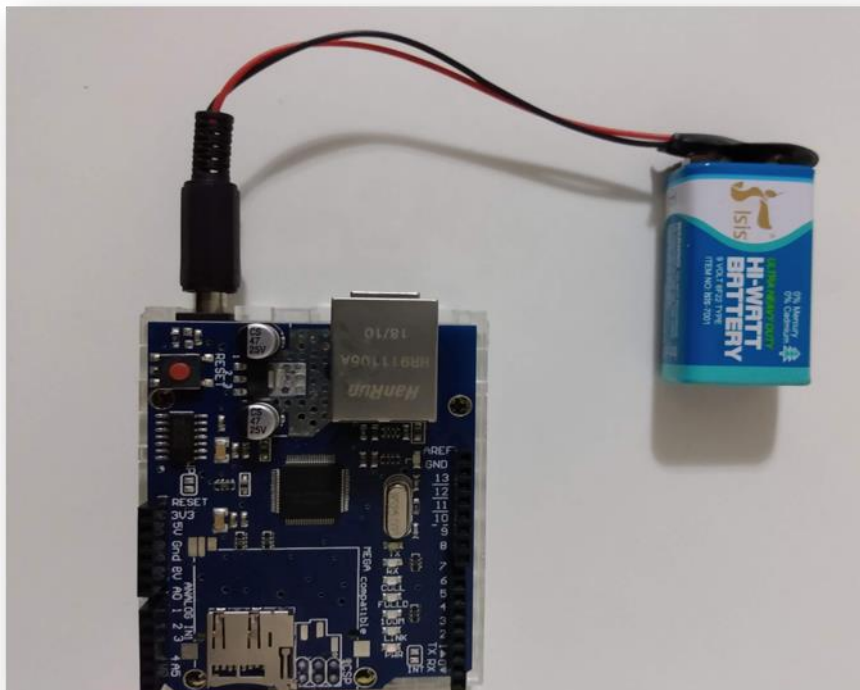
Fonte: Elaboração do autor



Fonte: Elaboração do autor

6º passo: Conecte fonte de alimentação externa ao conjunto.

Figura 9 – Fonte de alimentação do conjunto.



Fonte: Elaboração do autor

ATENÇÃO! VOCÊ TAMBÉM PODE UTILIZAR UMA FONTE DE TENSÃO EXTERNA, PORÉM OBSERVE QUE A FONTE DEVA ESTAR NA FAIXA DE 6V A 12V. VALORES MAIORES PODEM CAUSAR INSTABILIDADE NO DESEMPENHO DA PLACA E INCLUSIVE DANIFICA-LA.

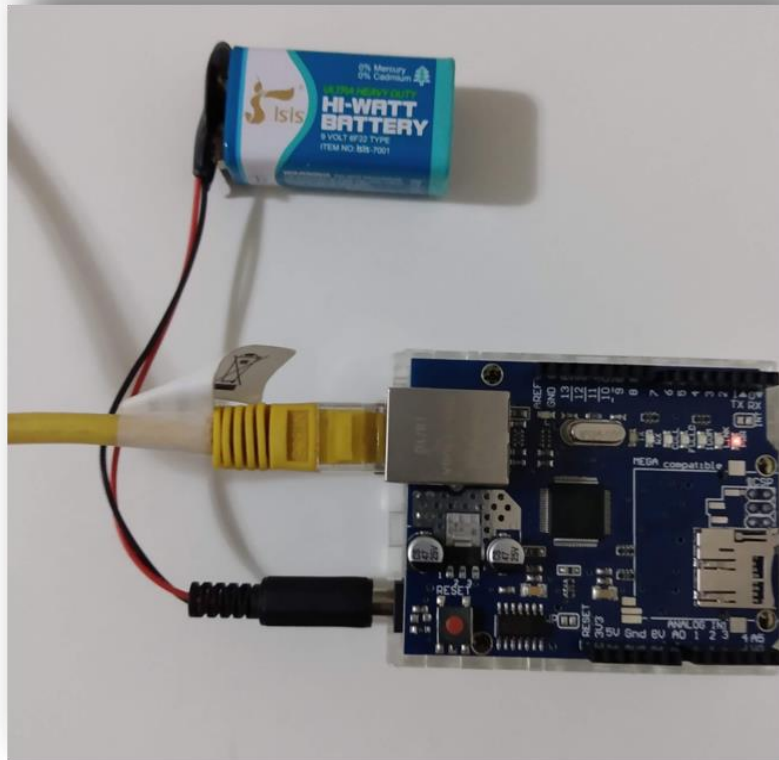
Figura 10 – Fonte de alimentação.



Fonte: Elaboração do autor

6º passo: Conecte uma das extremidade do cabo de rede ao conjunto.

Figura 11 – Conexão de cabo Ethernet.



Fonte: Elaboração do autor

7º passo: Conecte a outra extremidade do cabo de rede ao ROTEADOR.

Figura 12 – Conexão cabo Ethernet no Roteador.



Fonte: Elaboração do autor

CÓDIGO

Para acompanhar exercício proposto deste manual, você deve baixar o sketch, através do deste link < <https://drive.google.com/open?id=1m70rz1SsSmCrZBScwa6gldVcirjFpAqH> >. Utilizamos como referência para nossa atividade, um exemplo de algoritmo disponível no site Vida de Silício. Disponível neste link² com algumas modificações.

```
//=====Bibliotecas obrigatórias=====
#include <SPI.h> //biblioteca utilizada para utilizar o barramento SPI e fazer a comunicação da
//placa Arduino com o Shiel Ethernet via protocolo SPI
#include <Ethernet.h> //possibilita a conexão do conjunto (arduino + shield ) inicialmente como
//uma rede local

//=====Indicação de LED=====
/*
 * TX = transmissão
 * RX = Recepção
 * COLL = Colisão
 * FULLD = Mdo de conexão Full Duplex
 * 100M = Conexão de 100Mbps
 * LINK = Conexão Estabelecida
 * PWR = Módulo Ligado
 */

//=====Fonte de Pesquisa=====
//https://portal.vidadesilicio.com.br/shield-ethernet-w5100-servidor-web/

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; //endereço físico associado á interface
//de comunicação que conecta o dispositivo a rede.
IPAddress ip_maquina(192, 168, 25, 16); //obrigatório endereço de IP para que haja conexão
//entre o conjunto e a rede local.
EthernetServer servidor(80); //criação do objeto Server que responsável por representar o
//Servidor Web; porta 80 padrão para endereço HTTP

void setup(){
  Ethernet.begin(mac, ip_maquina); //inicialização da conexão com a rede local
  servidor.begin(); //inicialização do Servidor Web
  Serial.begin(9600); //inicialização do Serial Monitor
} //end void setup

void loop(){
  EthernetClient cliente = servidor.available(); //Objeto Server avalia se há dados disponíveis
//para leitura e armazena na função 'EthernetClient cliente'.
//Objeto Server será avaliado como falso em uma instrução IF
```

²Fonte: Disponível em:< <https://portal.vidadesilicio.com.br/shield-ethernet-w5100-servidor-web/> >. Acesso em: 22 set.2019.

```

if (cliente){ //se existir tentativa de conexão do cliente para servidor através do endereço IP,
//as instruções abaixo serão executadas
    boolean currentLinelsBlank = true; //variável tipo booleana para controle da leitura realizada
//pelo navegador.

```

```

while (cliente.connected()){//comando while utilizado para forçar o programa ficar restito
//apenas as informações do seu conteúdo, através do parâmetro client.connected()
    if (cliente.available()){//se a tentativa de conexão do cliente para servidor for verdadeiro
        char c = cliente.read();//char é texto de comprimento fixo que após ser preenchida com um
//determinado caractere da requisição, executa uma série de comparações com
        //alguns parâmetros específicos para saber se o Arduino recebeu o pacote de informações
//inteiro ou não.
        Serial.print(c);//OBSERVAR RESULTADOS!!!!

```

```

if (c == '\n' && currentLinelsBlank){// Caso a condição desta primeira função de tratamento da
//variável c seja verdadeira, quer dizer que recebemos o pacote completo e que o conteúdo
//desta função if() será executado.

```

```

    cliente.println("HTTP/1.1 200 OK");
    cliente.println("Content-Type: text/html");
    cliente.println("Connection: close");
    cliente.println();
    cliente.println("<!DOCTYPE html>");
    cliente.println("<html>");
    cliente.println("<head>");
    cliente.println("<title>Servidor Web VDS</title>");
    cliente.println("</head>");
    cliente.println("<body>");
    cliente.println("<h1>Meu primeiro Servidor Web</h1>");
    cliente.println("<p>Esperamos que voce tenha gostado deste tutorial</p>");
    cliente.println("</body>");
    cliente.println("</html>");
    break;

```

```

} //end condição IF

```

```

//condições que verificam se a requisição foi finalizada

```

```

/*

```

```

* Após o desenvolvimento da nossa página, saímos da função if()
* utilizada para saber se a requisição foi finalizada e encontramos outras duas funções if()
* que cumprem o teste das condições não contempladas pela primeira, ou seja,
* atuam no tratamento da variável c enquanto a requisição não foi finalizada.
*/

```

```

    if (c == '\n'){//Newline ('\n')
        currentLinelsBlank = true;
    } //end condição IF
    else if (c != '\r'){//Carriage Return ('\r')
        currentLinelsBlank = false;
    } //end condição ELSE IF
} //end condição IF

```

```
//end while

delay(1);// delay() para garantir que o navegador receba esta informação que estamos
//mandando
cliente.stop();// client.stop() para parar a conexão

}
}
```

A partir deste momento iremos nos aprofundar mais na estrutura básica da linguagem de marcação HTML.

Para está tapa outra ótima dica seria você visitar o site da WC3 através do site <https://www.w3schools.com/html/>. Com a nova versão do HTML 5, você precisa obter algumas regras que são pré estabelecidas para que seu código funcione sem nenhum tipo de problema.

Na linguagem de marcação HTML todas as funções e parâmetros que você utilizar precisa estar dentro dos símbolos chamados <> TAG. É muito importante que você tenha muito cuidado no momento de criar seus códigos que estes elementos estejam presentes.

LINGUAGEM DE MARCAÇÃO HTML

Figura 13 – IDE Visual Studio Code.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3    <head>
4      <!-- Meta tags Obrigatórias -->
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7      <link rel="icon" href="icons/favicon.ico">
8      <title>Programando</title>
9    </head>
10
11    <body>
12    </body>
13
14  </html>
```

Fonte: Elaboração do autor

FUNCIONAMENTO DO CONJUNTO (ARDUINO + SHIELD ETHERNET)

O Arduino se comunica com o W5100 e o cartão SD usando o barramento SPI (através do cabeçalho ICSP).

Isso ocorre nos pinos digitais 11, 12 e 13[...] o pino 10 é usado para selecionar o W5100 e o pino 4 para o cartão SD.

Esses pinos não podem ser usados para E / S geral. mas deve ser mantido como saída ou a interface SPI não funcionará.

Observe que, como o W5100 e o cartão SD compartilham o barramento SPI, apenas um pode estar ativo por vez. Se você é usando os dois periféricos em seu programa, isso deve ser resolvido pelas bibliotecas correspondentes. E se

you não está usando um dos periféricos do seu programa, no entanto, precisará desmarcá-lo explicitamente. Façam

isso com o cartão SD, defina o pino 4 como saída e escreva um valor alto. Para o W5100, defina o pino digital 10 como alto rendimento.

[TEXTO EXPLICATIVO CÓPIADO EXATAMENTE COMO CONSTA NO DATASHEET DO EQUIPAMENTO, TENDO APENAS UMA OMISSÃO NO CONTEÚDO, POIS NÃO SERIA RELEVANTE.]

REFERÊNCIA BIBLIOGRAFICA

<https://www.tecmundo.com.br/internet/982-o-que-e-cliente-servidor-.htm>

<https://br.ccm.net/contents/150-modelo-cliente-servidor>

<https://www.portaleducacao.com.br/conteudo/artigos/informatica/modelo-cliente-servidor/31652>

<https://www.usinainfo.com.br/>

Autor: Crystoppher Simões Trindade

Técnico em Eletrônica e Graduando em Análise e Desenvolvimento de Sistemas.