**Introdução ao Arduíno**

**Prática : Comunicação pela rede com o Módulo Ethernet ENC28J60**

**I - Objetivo**

Conectar o Arduino à rede através do Módulo Ethernet ENC28J60

**II – Introdução**

O Módulo Ethernet ENC28J60 é um driver que utiliza o chip da Microchip ENC28J60 – Controlador de Internet IC que suporta vários protocolos da internet que são requeridos. Este módulo conecta diretamente com a maioria dos microcontroladores com uma interface SPI cuja transferência ultrapassa 20MHz.

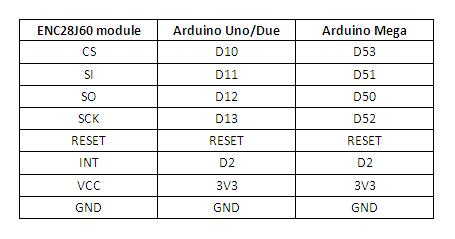
|  |  |
| --- | --- |
|  |  |

O módulo é formado pela porta ethernet RJ45, o chip controlador ENC28J60 (daí o nome do módulo), um cristal de 25Mhz e um conector de 10 pinos, o que torna prática a ligação desse módulo à diferentes tipos de microcontrolador.

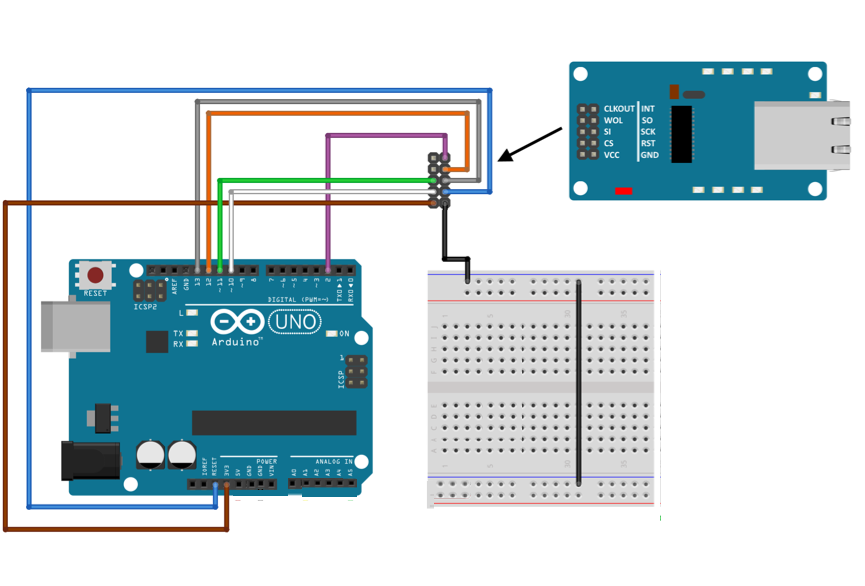
## ****III – Montagem Prática****

## ****Pinagem Ethernet ENC28J60:****

A alimentação do módulo é feita com **3.3v** e a igação desse módulo ao Arduino é feita via interface SPI. Na tabela abaixo mostramos a ordem de conexão dos pinos tanto para o Arduino Uno. Em uma ligação básica os pinos **CLKOUT** e **WOL** mostrados acima não são utilizados.



Vamos conectar o **módulo ENC28J60** montando o circuito abaixo e programando o Arduino .



## ****Arduino com Ethernet ENC28J60:****

Para enviar os dados do sensor para o Arduino, vamos utilizar a biblioteca, EtherCard.h. Descompacte o arquivo e coloque a pasta correspondente dentro da pasta **LIBRARIES** da **IDE** do  Arduino.

Essa biblioteca é bem prática pois permite a criação de páginas html utilizando poucos comandos, como você pode ver no código abaixo.

*// Programa de mostrar um dado em uma página html com Arduino*

*// Necessita de uma conexão ethernet com cabo no Arduino*

*// Colocar aqui seu IP myip[] = { 172,16,10,164 };*

*// Colocar aqui o GW gwip[] = { 172,16,10,1 };*

*#include <EtherCard.h>*

*// ethernet interface mac address, tem que ser único na LAN*

*static byte mymac[] = { 0x74,0x69,0x69,0x2D,0x30,0x31 };*

*static byte myip[] = { 172,16,10,164 };*

*static byte gwip[] = { 172,16,10,1 };*

*byte Ethernet::buffer[500];*

*BufferFiller bfill;*

*void setup () {*

*if (ether.begin(sizeof Ethernet::buffer, mymac) == 0)*

*Serial.println( "Failed to access Ethernet controller");*

*ether.staticSetup(myip, gwip);*

*}*

*static word homePage() {*

*long t = millis() / 1000;*

*word h = t / 3600;*

*byte m = (t / 60) % 60;*

*byte s = t % 60;*

*bfill = ether.tcpOffset();*

*bfill.emit\_p(PSTR(*

*"HTTP/1.0 200 OK\r\n"*

*"Content-Type: text/html\r\n"*

*"Pragma: no-cache\r\n"*

*"\r\n"*

*"<meta http-equiv='refresh' content='1'/>"*

*"<title>Servidor Arduino de http</title>"*

*"<h1>Servidor Arduino de http</h1>"*

*"<h1>$D$D:$D$D:$D$D</h1>"),*

*h/10, h%10, m/10, m%10, s/10, s%10);*

*return bfill.position();*

*}*

*void loop () {*

*word len = ether.packetReceive();*

*word pos = ether.packetLoop(len);*

*if (pos) // Checa se um dado de tcp recebido é válido*

*ether.httpServerReply(homePage()); // envia dado para página*

*}*

Você deverá incluir o programa a leitura de algum módulo como o DHT11 que possui as saídas seriais para a saída ser apresentado em página HTML.

Teste o programa abrindo um browser e digitando na barra de endereços o IP que você configurou para a sua placa.

|  |
| --- |
|  |