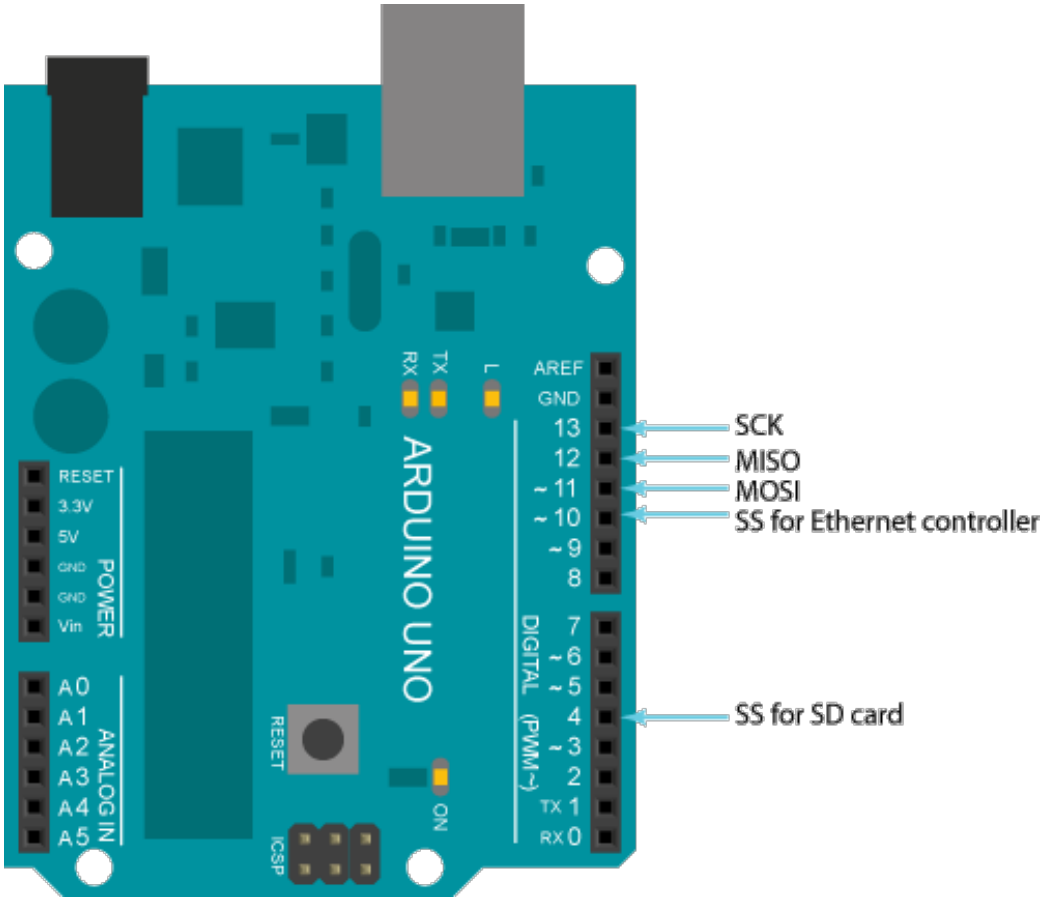
Ethernet library

With the Arduino Ethernet Shield, this library allows an Arduino board to connect to the internet. It can serve as either a server accepting incoming connections or a client making outgoing ones. The library supports up to four concurrent connection (incoming or outgoing or a combination).

Arduino communicates with the shield using the SPI bus. This is on digital pins 11, 12, and 13 on the Uno and pins 50, 51, and 52 on the Mega. On both boards, pin 10 is used as SS. On the Mega, the hardware SS pin, 53, is not used to select the W5100, but it must be kept as an output or the SPI interface won't work.



Ethernet class

The Ethernet cla initializes the ethernet library network settings

- begin()
- localIP()
- maintain()

IPAddressclass

The IPAddress cl works with local remote IP addressing.

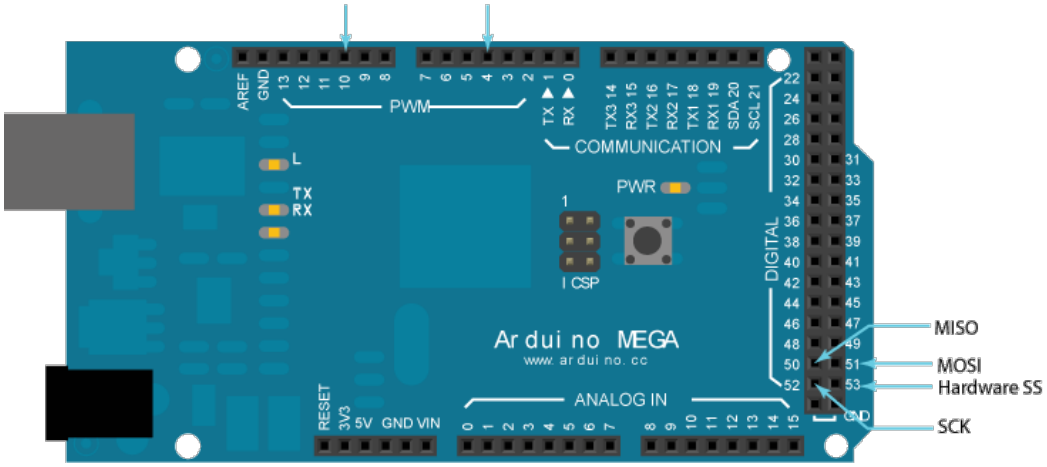
- IPAddress()

Server class

The Server class creates servers w can send data tc receive data fror connected clients (programs runni on other comput or devices).

- Server
- EthernetServer
- begin()
- available()
- write()
- print()
- println()

Client class



Examples

ChatServer: set up a simple chat server.

WebClient: make a HTTP request.

WebClientRepeating: Make repeated HTTP requests.

WebServer: host a simple HTML page that displays analog sensor values.

XivelyClient: connect to Xively.com, a free datalogging site.

XivelyClientString: send strings to Xively.com.

BarometricPressureWebServer: outputs the values from a barometric pressure sensor as a web page.

UDPSendReceiveString: Send and receive text strings via UDP.

UdpNtpClient: Query a Network Time Protocol (NTP) server using UDP.

DnsWebClient: DNS and DHCP-based Web client.

DhcpChatServer: A simple DHCP Chat Server

DhcpAddressPrinter: Get an IP address via DHCP and print it out

TelnetClient: A simple Telnet client

receive data.

- Client
- EthernetClient(
- if (EthernetClie
- connected()
- connect()
- write()
- print()
- println()
- available()
- read()
- flush()
- stop()

EthernetUDPclass

The EthernetUDP enables UDP mes to be sent and received.

- begin()
- read()
- write()
- beginPacket()
- endPacket()
- parsePacket()
- available()
- stop()
- remoteIP()
- remotePort()