Shield Ethernet

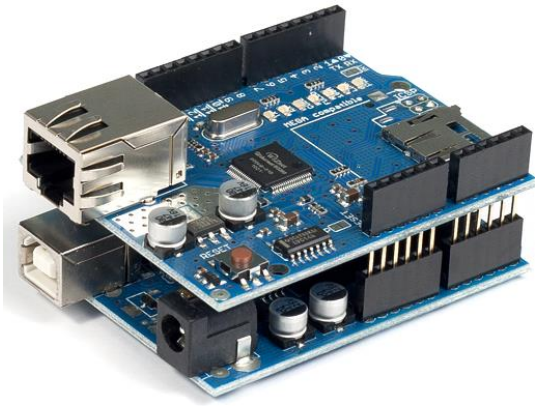
TUTORA: ANGELA JAZMÍN MIRANDA FLORES

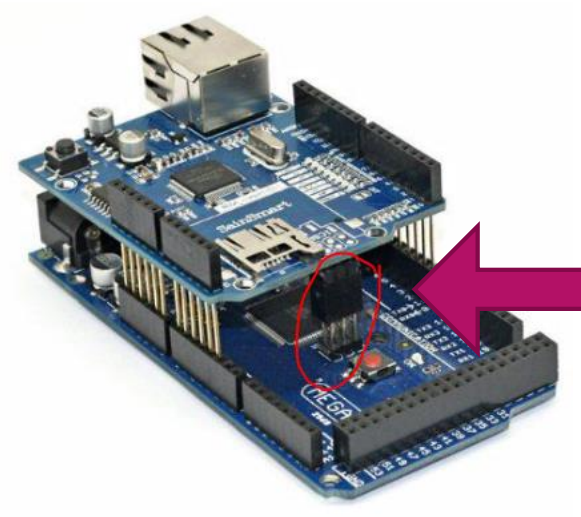
¿Qué es?

- Es una placa basada en el chip Ethernet Wiznet W5100, el cual provee un conjunto de direcciones IP, es capaz de utilizar los protocolos TCP y UDP para conectarse a una red.
- Usa la librería Ethernet.
- Es una integración del módulo Ethernet y el módulo SD.
- Utiliza un conector RJ45 estándar.
- Es compatible con arduino uno, arduino mega, arduino Leonardo.
- Permite a una placa arduino conectarse a internet.

Conexión

La Shield Ethernet se sobrepone sobre nuestra placa arduino (arduino r3, arduino mega, arduino Leonardo).
Se deben alinear los pines de manera correcta, o podemos arruinar nuestra placa arduino y la Shield Ethernet.

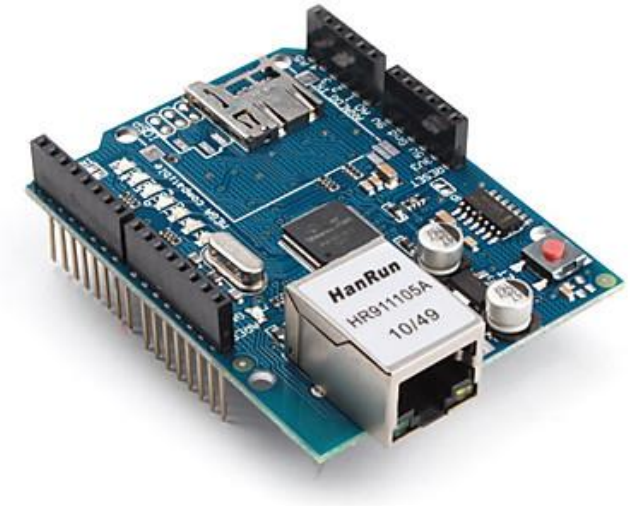




Asegurarse de que esta puesta correctamente

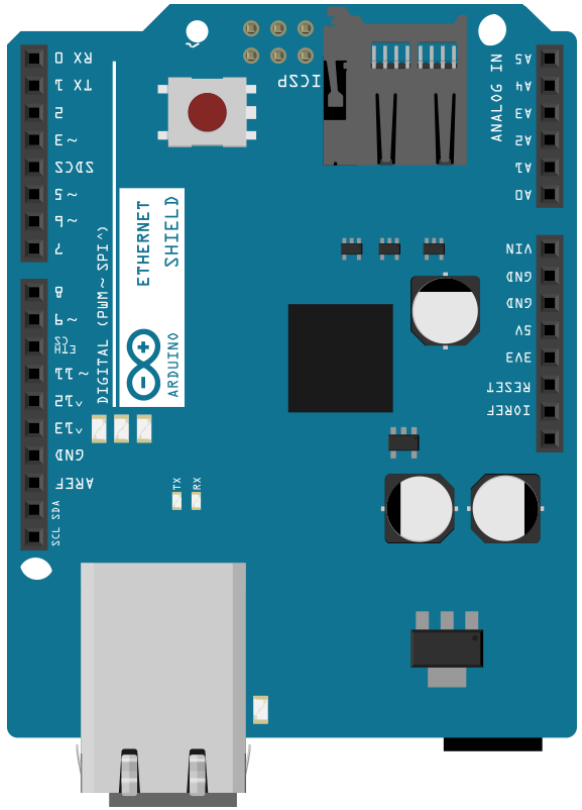
Características

- Tensión: 5V.
- Memoria: 16 Kb.
- Velocidad: 10/100 Mb.
- Conexión con arduino a través de SPI.
- Soporta hasta cuatro conexiones de sockets simultaneas.
- Dispone de un lector de tarjetas micro-SD.
- Utiliza IPv4.



Características de conexión

- Utiliza los pines 10,11,12,13 (SPI) para conectarse con el w5100.
- El botón de reset en la Shield resetea ambos, es decir la Shield y nuestra placa arduino.
- Leds de información:
  - PWR: Indica que la placa y la Shield están alimentadas.
  - LINK: Indica la presencia de un enlace de red y parpadea cuando la Shield envía y recibe datos.
  - 100M: Indica la presencia de una conexión de red de 100 MB/S.
  - FULLD: Indica que la conexión de red es full dúplex.
  - COLL: Parpadea cuando se detectan colisiones en la red.
  - RX: Parpadea cuando la Shield recibe datos.
  - TX: Parpadea cuando la Shield envía datos.



TUTORA: Angela Jazmín Miranda Flores

PROTOCOLO TCP

Transmission Control Protocol (Protocolo de Transmisión).

Su objetivo es transportar de forma segura el flujo de bits entre aplicaciones, gracias a esto el programador queda libre de estar verificando la seguridad y fiabilidad en las conexiones.

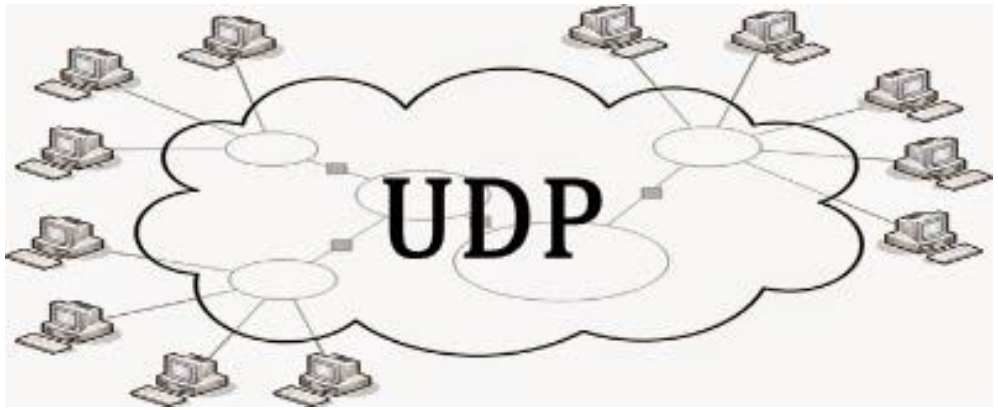
Envía paquetes de datos de 20 bytes como mínimo.

Protocolo UDP

User Datagram Protocol (Protocolo de datagrama de usuario).

Usa solo 8 bytes para el envío de la información, es poco fiable pero la capacidad de envío es mas rápida.

Se utiliza a menudo en la transferencia de audio y video.



TUTORA: Angela Jazmín Miranda Flores

Protocolo HTTP

HyperText Transfer Protocol (Protocolo de Hipertransferencia de texto) es un protocolo de comunicación que permite las transferencia de datos en la WWW (World Wide Web).

Utiliza el puerto de red 80, este protocolo puede ser utilizado sin conexión y sin estado.

Métodos de petición y envío de datos

- GET: Transmite la información a través de la URL.
- POST: Envía los datos para que sean procesador solo por el recurso identificado.
- PUT: sube, carga o realiza un upload de un recurso especificado

Protocolo FTP

File Transfer Protocol (protocolo de transferencia de archivos) es un protocolo de red que se utiliza para la transferencia de archivos entre sistemas conectados a una red TCP.

Esta basado en la arquitectura cliente-servidor, es decir desde un equipo se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente de cual sea s sistema operativo.

Utiliza el puerto de red 20 y 21, además es un protocolo con conexión y con estado.

TUTORA: Angela Jazmín Miranda Flores

EJERCICIO EN CLASE
Visualizar una página web en la Shield Ethernet

Hacer PING a una página web, en nuestro ejemplo es www.google.com

```
C:\Users\Angie>ping www.google.com

Haciendo ping a www.google.com [216.58.204.4] con 32 bytes de datos:
Respuesta desde 216.58.204.4: bytes=32 tiempo=289ms TTL=48
Respuesta desde 216.58.204.4: bytes=32 tiempo=364ms TTL=48
Respuesta desde 216.58.204.4: bytes=32 tiempo=325ms TTL=48
Respuesta desde 216.58.204.4: bytes=32 tiempo=307ms TTL=48

Estadísticas de ping para 216.58.204.4:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 289ms, Máximo = 364ms, Media = 321ms

C:\Users\Angie>
```

Dirección IP de GOOGLE

TUTORA: Angela Jazmín Miranda Flores

Shield Ethernet – Arduino

```
#include <SPI.h>
#include <Ethernet.h>
//direccion mac de la ethernet shield
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
//direccion ip de GOOGLE
IPAddress server(216, 58, 204,4);
//inicializa la conexion con el servidor
EthernetClient client;

void setup() {
  Serial.begin(9600);
  // inicia la conexion Ethernet
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Falla en la conexion");
  }
  // tiempo de respiro para que se realice la conexion
  delay(1000);
  Serial.println("connecting...");

  // conexion del cliente mediante el puerto 80
  if (client.connect(server, 80)) {
    Serial.println("connected");
    client.println("GET /search?q=arduino HTTP/1.1");
    client.println("Host: www.google.com");
    client.println("Connection: close");
    client.println();
  } else {
    // si la conexion no se ha relaizado
    Serial.println("conexion fallida");
  }
}

void loop() {
  // si la conexion esta disponible
  if (client.available()) {
    char c = client.read();
    Serial.print(c);
  }

  // si el servidor esta desconectado se detiene al cliente
  if (!client.connected()) {
    Serial.println();
    Serial.println("disconnecting.");
    client.stop();
  }
}
```

TUTORA: Angela Jazmín Miranda Flores

Salida por consola serial

TUTORA: Angela Jazmín Miranda Flores

EJERCICIO DE APLICACIÓN
Arduino como servidor

para poder hacer que nuestro arduino sea un servidor debemos asignarle una dirección IP, para lo cual ingresamos a CMD

Verificamos nuestra dirección IP, para no asignarle la misma a nuestro arduino.

```
C:\Users\Angie>ipconfig

Configuración IP de Windows


Adaptador de LAN inalámbrica Conexión de área local* 13:

   Estado de los medios. . . . . . . . . . . : medios desconectados
   Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 2:

   Estado de los medios. . . . . . . . . . . : medios desconectados
   Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Wi-Fi:

   Sufijo DNS específico para la conexión.  . :
   Vínculo: dirección IPv6 local. . . : fe80::48d9:ebd1:d1bc:8580%4
   Dirección IPv4. . . . . . . . . . . . . . : 192.168.1.7
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 192.168.1.1
```

EJERCICIO DE APLICACIÓN
Arduino como servidor

VERIFICAMOS UNA DIRECCIÓN IP DISPONIBLE

```
C:\Users\Angie>ping 192.168.1.7

Haciendo ping a 192.168.1.7 con 32 bytes de datos:
Respuesta desde 192.168.1.7: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.7: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.7: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.7: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.1.7:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Dirección IP ocupada

Dirección IP libre

```
C:\Users\Angie>ping 192.168.1.100

Haciendo ping a 192.168.1.100 con 32 bytes de datos:
Respuesta desde 192.168.1.7: Host de destino inaccesible.
Respuesta desde 192.168.1.7: Host de destino inaccesible.
Respuesta desde 192.168.1.7: Host de destino inaccesible.
Respuesta desde 192.168.1.7: Host de destino inaccesible.

Estadísticas de ping para 192.168.1.100:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
```

TUTORA: Angela Jazmín Miranda Flores

EJERCICIO DE APLICACIÓN
Arduino como servidor

```
#include <SPI.h>
#include <Ethernet.h>

byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };// dirección MAC del ethernet
IPAddress ip(192,168,1,100);   // direccion IP del servidor
// NOTA: esta direccion se usara para conectarse mediante el Navegador al servidor.
EthernetServer server(80);     // Puerto 80 por defecto para HTTP
void setup() {
  Ethernet.begin(mac, ip);     //inicializ a la conexión Ethernet y el servidor
  server.begin();
}

void loop() {
  EthernetClient cliente = server.available(); // Inicializa cliente como servidor ethernet
  if (cliente) {
    boolean currentLineIsBlank = true;
    while (cliente.connected()) {
      if (cliente.available()) {
        char c = cliente.read();
```

TUTORA: Angela Jazmín Miranda Flores

EJERCICIO DE APLICACIÓN
Arduino como servidor

```
if (c == '\n' && currentLineIsBlank) {
  cliente.println("HTTP/1.1 200 OK");
  cliente.println("Content-Type: text/html");   // Envia el encabezado en codigo HTML estandar
  cliente.println("Connection: close");
  cliente.println("Refresh: 3");  // refresca la pagina automaticamente cada 3 segundos
  cliente.println();
  cliente.println("<!DOCTYPE HTML>");
  cliente.println("<html>");
  cliente.println("<HEAD>");
  cliente.println("<TITLE>Ethernet Monitor</TITLE>");
  cliente.println("</HEAD>");
  cliente.println("<BODY>");
  cliente.println("<hr />");
  cliente.println("<H1>Servidor desde Arduino</H1>");
  cliente.println("<br />");
  cliente.println("<H2>Valores de salida de los pines analogicos A0-A5</H2>");
  cliente.println("<br />");
  cliente.println("Lectura Analogica Ethernet");
  cliente.println("<br />");
  cliente.println("<br />");
```

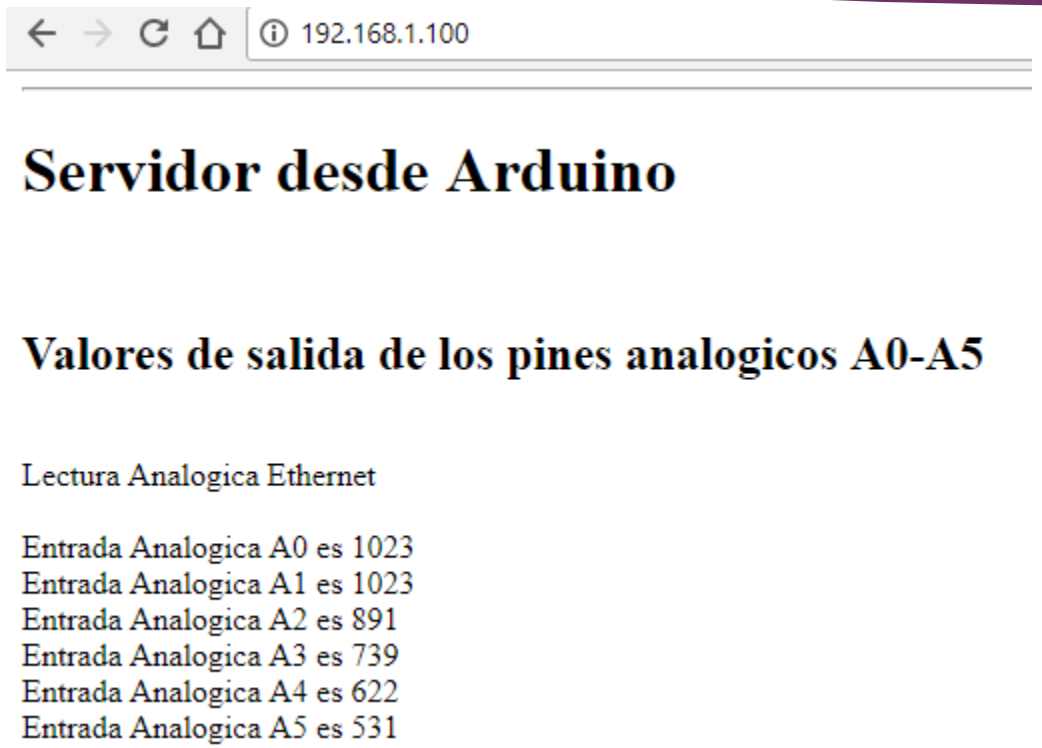
TUTORA: Angela Jazmín Miranda Flores

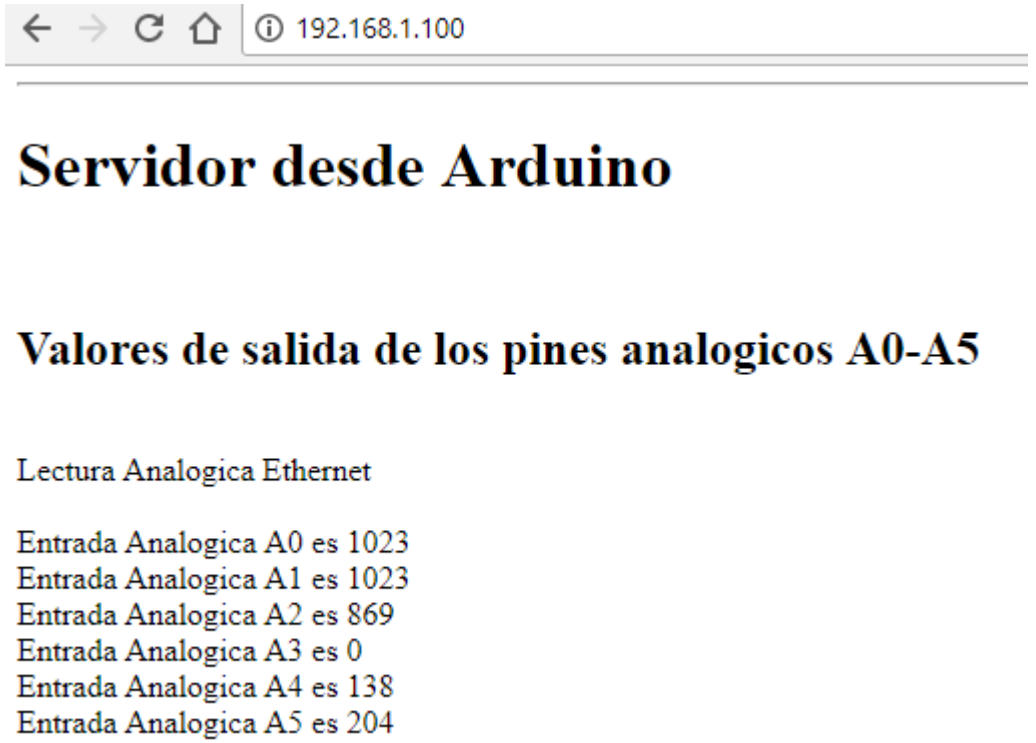
EJERCICIO DE APLICACIÓN
Arduino como servidor

```
          for (int puertoAnalogo = 0; puertoAnalogo < 6; puertoAnalogo++) {
            int lecturaSensor = analogRead(puertoAnalogo);   // Lee los 6 puertos analogos de A0 a A5
            cliente.print("Entrada Analogica  A");
            cliente.print(puertoAnalogo);
            cliente.print(" es ");
            cliente.print(lecturaSensor);
            cliente.println("<br />");
          }
          cliente.println("</html>");
          break;
        }
        if (c == '\n') {
           currentLineIsBlank = true;
        }
        else if (c != '\r') {
           currentLineIsBlank = false;
        }
      }
    }
   delay(15);           // Da tiempo al Servidor para que reciba los datos 15ms
   cliente.stop();     // cierra la conexion
  }
}
```

TUTORA: Angela Jazmín Miranda Flores

EJERCICIO DE APLICACIÓN
Arduino como servidor

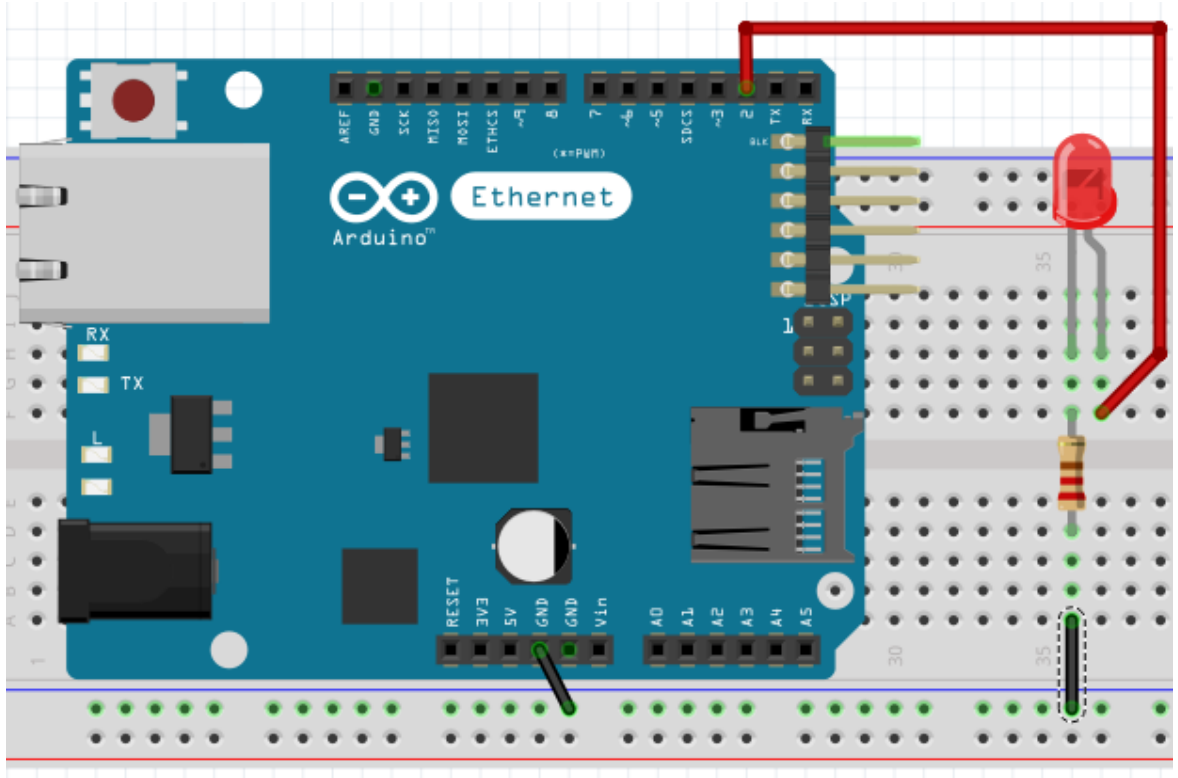




TUTORA: Angela Jazmín Miranda Flores

EJERCICIO DE APLICACIÓN

ENCENDER Y APAGAR UN LED DESDE UNA PAGINA WEB



TUTORA: Angela Jazmín Miranda Flores