

CONECTAR ARDUINO A INTERNET O LAN CON SHIELD ETHERNET W5100



¿QUÉ ES EL W5100?

El W5100 es un controlador de Ethernet fabricado por Wiznet especialmente diseñado para aplicaciones embebidas. Podemos emplear este controlador con un procesador como Arduino para implementar comunicación por internet.

El W5100 está diseñado para facilitar la implementación de conectividad a internet sin necesidad de un SO, lo que lo hace interesante para MCU y aplicaciones de IoT.

IoT : Internet de las cosas (en inglés, Internet of Things, abreviado IoT)¹² es un concepto que se refiere a la interconexión digital de objetos cotidianos con Internet.³ Alternativamente, Internet de las cosas es la conexión de Internet con más “cosas u objetos” que personas.⁴ También se suele conocer como Internet de todas las cosas o Internet en las cosas. Si objetos de la vida cotidiana tuvieran incorporadas etiquetas de radio, podrían ser identificados y gestionados por otros equipos, de la misma manera que si lo fuesen por seres humanos.

Incluye una pila de TCP/IP por hardware y buffer interno de 16Kbytes para Tx/Rx. Esto permite liberar de estas tareas al procesador, siendo una de sus principales ventajas frente a otros controladores de Ethernet como el ENC28J60.

El W5100 puede conectarse mediante SPI, por lo que es muy sencilla la conexión con la mayoría de procesadores. Algunos módulos incorporan un lector de tarjeta SD, donde podemos guardar ficheros (html, txt, jpg, png) con los que trabajar cuando actuemos como servidor.

El Bus SPI (del inglés Serial Peripheral Interface) es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. El bus de interfaz de periféricos serie o bus SPI es un estándar para controlar casi cualquier dispositivo electrónico digital que acepte un flujo de bits serie regulado por un reloj (comunicación sincrónica).

Admite velocidades de 10/100 Mbits/s, soportando modos Full-Duplex y Half-Duplex con detección y corrección automática de la polaridad. Cumple con las especificaciones IEEE 802.3 10BASE-T y 802.3u 100BASE-TX. La pila TCP/IP soporta conexiones TCP, UDP, IPv4, ICMP, ARP, IGMP and PPPoE, y hasta 4 conexiones simultáneas.

El W5100 es un chip ampliamente empleado en dispositivos conectados en aplicaciones industriales, domésticas de domótica e IoT. Por ejemplo, entre otros muchos, pantallas y televisores inteligentes, relés activados por internet, impresoras, cámaras IP o dispositivos de almacenamiento en red.

Nosotros podemos emplear un módulo de Ethernet con W5100 en nuestros proyectos para enviar o recibir información desde internet o una red Interna, por ejemplo, para realizar la lectura de un sensor, o para activar, controlar o desactivar cualquier dispositivo.

No obstante hay que tener en cuenta que Arduino, debido a sus reducidas capacidades de memoria y procesador, no es la mejor máquina para realizar operaciones complejas como servir

páginas web. Para ello es mejor plantear el uso de procesadores más potentes como el STM32 o el ESP8266 o incluso un micro ordenador tipo Orange Pi, Raspberry Pi.

PRECIO

Existen distintos módulos que montan el W5100. Estos módulos disponen de la electrónica necesaria para conectar de forma sencilla a Arduino, y un conector estándar RJ45 para Ethernet.

El más habitual es el Ethernet Shield, que tiene la ventaja de ser muy sencillo de conectar ya que es apilable. Normalmente estos módulos incluyen lector de micro SD. Como desventaja, sólo nos servirán si empleamos un Arduino UNO o Mega. Podemos encontrarlo por 4.80€, buscando en vendedores internacionales de eBay o AliExpress.



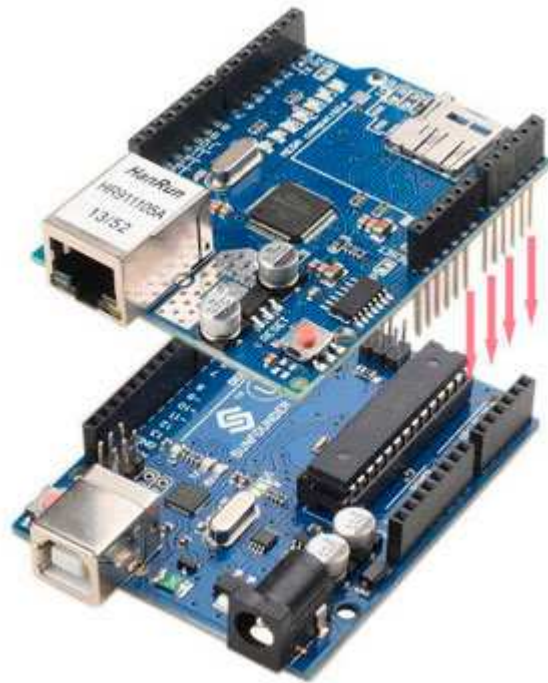
También módulos independientes con W5100. Tienen la ventaja de ser más compactos y podemos usarlo con todo tipo de MCU, como Arduino Mini y Nano, Teensy, o STM32. Como desventaja, lógicamente tendremos que cablear la conexión. Son ligeramente más baratos, podemos encontrarlos por 4.00€.

(MCU son unas siglas que pueden hacer referencia a: Microcontrolador, por sus siglas en inglés de microcontroller unit, un chip que contiene procesador, RAM, ROM, reloj y la unidad de control de E/S en un único encapsulado)

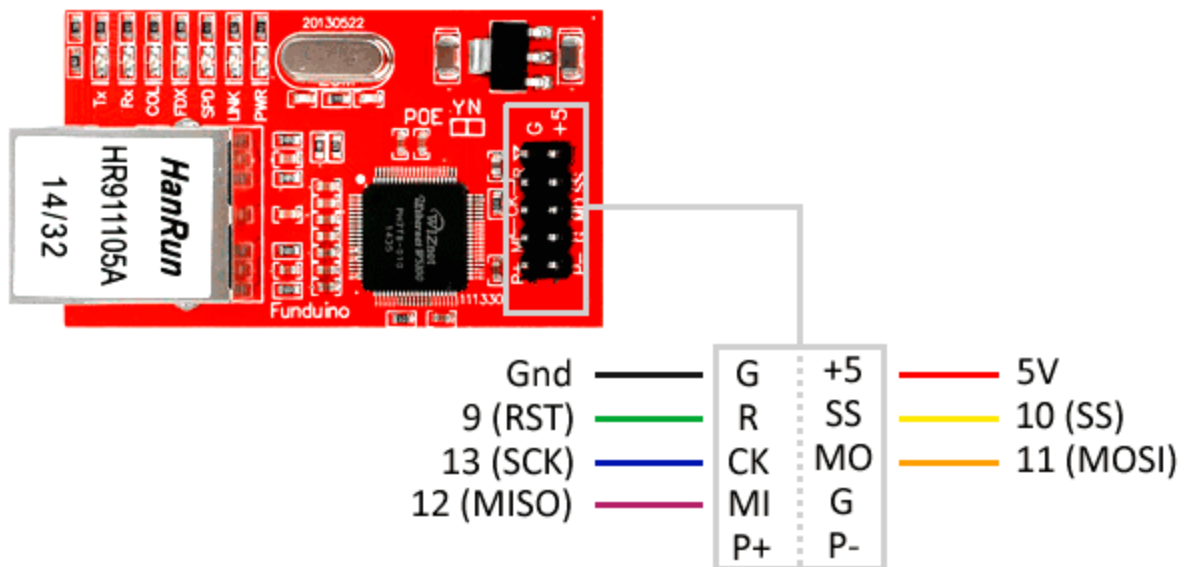


ESQUEMA DE MONTAJE

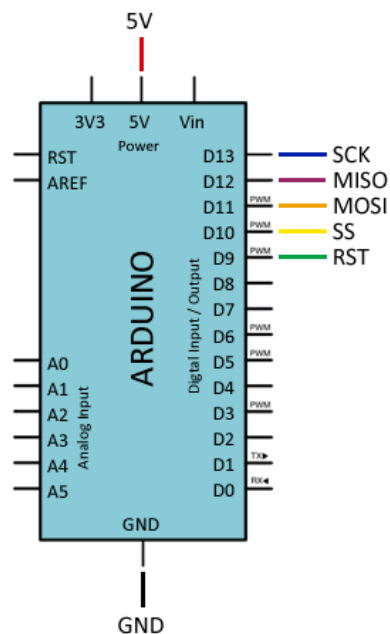
La conexión dependerá del modelo que estemos empleando. En el caso de emplear un Shield de Ethernet con W5100 la conexión es inmediata. Únicamente tenemos que acoplarlo a Arduino Uno o Mega.



Si usamos un módulo independiente con W5100 deberemos realizar el cableado, pero es igualmente sencillo ya que la conexión se realiza a través del SPI como vimos en esta entrada.



La conexión en este caso, vista desde Arduino, sería la siguiente.



EJEMPLOS DE CÓDIGO

Para controlar los módulos de Ethernet con W5100 usaremos la librería Ethernet.h, que está integrada en el propio IDE de Arduino.

La librería proporciona varios ejemplos de uso del W5100 que resulta aconsejable revisar. Los siguientes ejemplos son modificaciones a partir de los disponibles en la librería.

CLIENTE ETHERNET – LEER PÁGINAS WEB

(Nosotros desarrollaremos otro ejemplo en el TutorArduino)

En este ejemplo Arduino actúa como cliente, es decir, se conecta a una página web para leerla. Leer una página completa y volcarla por el puerto serie es muy lento, y es una de las muestras de las limitaciones de Arduino frente a un ordenador.

Sin embargo, puede ser útil para Arduino capture información desde un servidor. Por ejemplo, podemos hacer que sincronice la hora, que lea una serie de parámetros de un fichero de texto, que realice una determinada acción si existe un fichero, etc.

Para mostrar en este ejemplo esta capacidad de lectura de datos desde un servidor en Internet vamos a usar www.pasted.co, una de muchas páginas web que nos permiten añadir un texto para compartirlo con más gente.

En la página www.pasted.co/2434bc64 he pegado el texto ~1.2.3.4.5~. Los '~' los usaremos como separadores para encontrar el texto deseado '1.2.3.4.5', que simula una serie de parámetros que queremos capturar de un servidor.

El siguiente ejemplo se conecta con esta dirección y realiza la búsqueda del texto 1.2.3.4.5, que muestra por puerto serie. En un ejemplo real emplearíamos estos valores, por ejemplo, para controlar un robot, cambiar los parámetros de medición de una estación, encender o apagar un dispositivo, etc.

Arduino no tiene potencia suficiente para gestionar la encriptación necesaria en páginas https, por lo que sólo podremos leer páginas http.

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

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 1, 177);
EthernetClient client;

char serverC[] = "www.pasted.co";
char dataLocationC[] = "/2434bc64 HTTP / 1.1";

void setup()
{
  Serial.begin(9600);

  if (Ethernet.begin(mac) == 0)
  {
    Serial.println("Failed to configure Ethernet using DHCP");
    Ethernet.begin(mac, ip);
  }

  delay(1000);
```

```

Serial.println("connecting...");

if (client.connect(serverC, 80))
{
  Serial.println("connected");
  client.print("GET ");
  client.println(dataLocationC);
  client.print("Host: ");
  client.println(serverC);
  client.println();
}
else
{
  Serial.println("connection failed");
}
}

void loop()
{
  if (client.available())
  {
    char c = client.read();
    if (c == '~')
    {
      while (client.available())
      {
        c = client.read();
        if (c == '~')
        {
          client.stop();
          client.flush();
          break;
        }
        Serial.print(c);
      }
    }
  }

  if (!client.connected())
  {
    Serial.println();
    Serial.println("disconnecting.");
    client.stop();

    // do nothing
    while (true);
  }
}

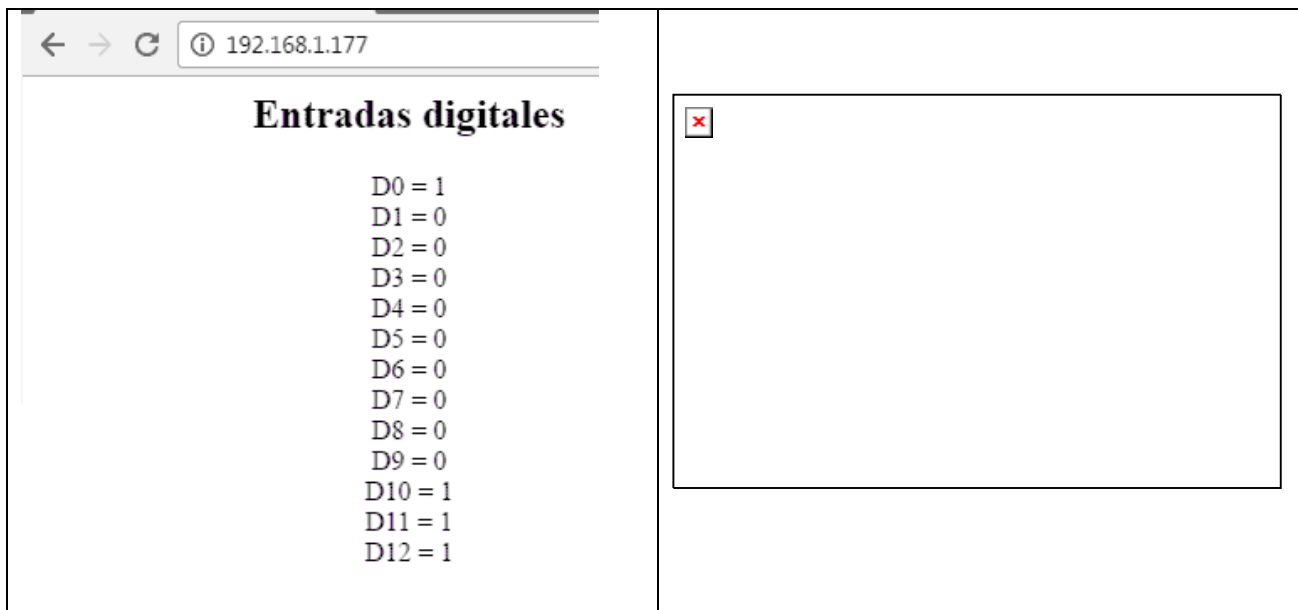
```

SERVIDOR ETHERNET – VISUALIZAR ENTRADAS

(Nosotros desarrollaremos otro ejemplo en el TutorArduino)

En el siguiente código Arduino actúa como servidor, es decir, devuelve una página web cuando un cliente (un PC, un móvil, otro Arduino...) se conecta a él.

En este caso, vamos a mostrar una página web con el estado de las entradas digitales y analógicas de Arduino. Para ello, simplemente servimos una cadena de texto en html, en la que incluimos los valores de las entradas.



Para visualizar la página simplemente accedemos a la dirección IP del Arduino, (192.168.1.177 en el ejemplo) desde cualquier explorador. El ordenador deberá estar conectado en la misma red local que el Arduino. Si queremos conectar desde internet deberemos definir una conexión bridge en el router que direcciones la IP externa a la IP local del Arduino.

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

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 1, 177);
EthernetServer server(80);

void setup()
{
  Serial.begin(9600);

  Ethernet.begin(mac, ip);
  server.begin();

  Serial.print("server is at ");
  Serial.println(Ethernet.localIP());
}

void loop()
{
  EthernetClient client = server.available();
  if (client)
  {
    Serial.println("new client");
    bool currentLineIsBlank = true;
    String cadena = "";
    while (client.connected())
    {
      if (client.available())
      {
        char c = client.read();
        Serial.write(c);
      }
    }
  }
}
```

```

// Al recibir linea en blanco, servir página a cliente
if (c == '\n' && currentLineIsBlank)
{
  client.println(F("HTTP/1.1 200 OK\nContent-Type: text/html"));
  client.println();

  client.println(F("<html>\n<head>\n<title>Luis Llamas</title>\n</head>\n<body>"));
  client.println(F("<div style='text-align:center;'>"));

  client.println(F("<h2>Entradas digitales</h2>"));
  for (int i = 0; i < 13; i++)
  {
    client.print("D");
    client.print(i);
    client.print(" = ");
    client.print(digitalRead(i));
    client.println(F("<br />"));
  }
  client.println("<br /><br />");

  client.println(F("<h2>Entradas analogicas</h2>"));
  for (int i = 0; i < 7; i++)
  {
    client.println("A");
    client.println(i);
    client.println(" = ");
    client.println(analogRead(i));
    client.println(F("<br />"));
  }

  client.println(F("<a href='http://192.168.1.177'>Refrescar</a>"));
  client.println(F("</div>\n</body></html>"));
  break;
}
if (c == '\n')
{
  currentLineIsBlank = true;
}
else if (c != '\r')
{
  currentLineIsBlank = false;
}
}
}

delay(1);
client.stop();
}
}

```

SERVIDOR ETHERNET – CONTROLAR SALIDAS

El este ejemplo Arduino actúa también como servidor, pero esta vez queremos que el usuario pueda realizar acciones sobre Arduino a través de la página web que servimos.

En este caso, vamos a controlar dos salidas digitales, a las que podemos conectar un Led para visualizar la respuesta.

Para ello, en primer lugar servimos la página web de forma similar al ejemplo anterior, pero en esta incluimos dos botones para cada salida.



Al pulsar en cada botón se realiza una nueva solicitud a Arduino, con diferente URL a la original. Arduino captura la nueva solicitud, y emplea la URL recibida para realizar las acciones oportunas.

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

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 1, 177);
EthernetServer server(80);

const int pinLed1 = A0;
const int pinLed2 = 3;

void setup()
{
  Serial.begin(9600);

  Ethernet.begin(mac, ip);
  server.begin();

  Serial.print("server is at ");
  Serial.println(Ethernet.localIP());

  pinMode(pinLed1, OUTPUT);
  pinMode(pinLed2, OUTPUT);
  digitalWrite(pinLed1, LOW);
  digitalWrite(pinLed2, LOW);
}

void loop()
{
  EthernetClient client = server.available();
  if (client)
  {
    Serial.println("new client");
    bool currentLineIsBlank = true;
    String cadena = "";
    while (client.connected())
```



```

{
if (client.available())
{
char c = client.read();
Serial.write(c);

if (cadena.length()<50)
{
cadena.concat(c);

// Buscar campo data
int posicion = cadena.indexOf("data");
String command = cadena.substring(posicion);

if (command == "data1=0")
{
digitalWrite(pinLed1, HIGH);
}
else if (command == "data1=1")
{
digitalWrite(pinLed1, LOW);
}
else if (command == "data2=0")
{
digitalWrite(pinLed2, LOW);
}
else if (command == "data2=1")
{
digitalWrite(pinLed2, HIGH);
}
}

// Al recibir linea en blanco, servir página a cliente
if (c == '\n' && currentLineIsBlank)
{
client.println(F("HTTP/1.1 200 OK\nContent-Type: text/html"));
client.println();

client.println(F("<html>\n<head>\n<title>Luis Llamas</title>\n</head>\n<body>"));
client.println(F("<div style='text-align:center;'>"));
client.println(F("<h2>Salidas digitales</h2>"));

client.print(F("Estado LED 1 = "));
client.println(digitalRead(pinLed1) == LOW ? "OFF" : "ON");
client.println(F("<br/>"));
client.println(F("<button onClick=location.href='./?data1=0'>ON</button>"));
client.println(F("<button onClick=location.href='./?data1=1'>OFF</button>"));
client.println(F("<br/><br/>"));

client.print(F("Estado LED 2 = "));
client.println(digitalRead(pinLed2) == LOW ? "OFF" : "ON");
client.println(F("<br/>"));
client.println(F("<button onClick=location.href='./?data2=0'>ON</button>"));
client.println(F("<button onClick=location.href='./?data2=1'>OFF</button>"));
client.println(F("<br/>"));

client.println(F("<a href='http://192.168.1.177'>Refrescar</a>"));
client.println(F("</div>\n</body></html>"));
break;
}
}
if (c == '\n')
{

```

```
        currentLineIsBlank = true;
    }
    else if (c != '\r')
    {
        currentLineIsBlank = false;
    }
}
}

delay(1);
client.stop();
}
}
```

Fuente:

<https://www.luisllamas.es/arduino-ethernet-shield-w5100/>