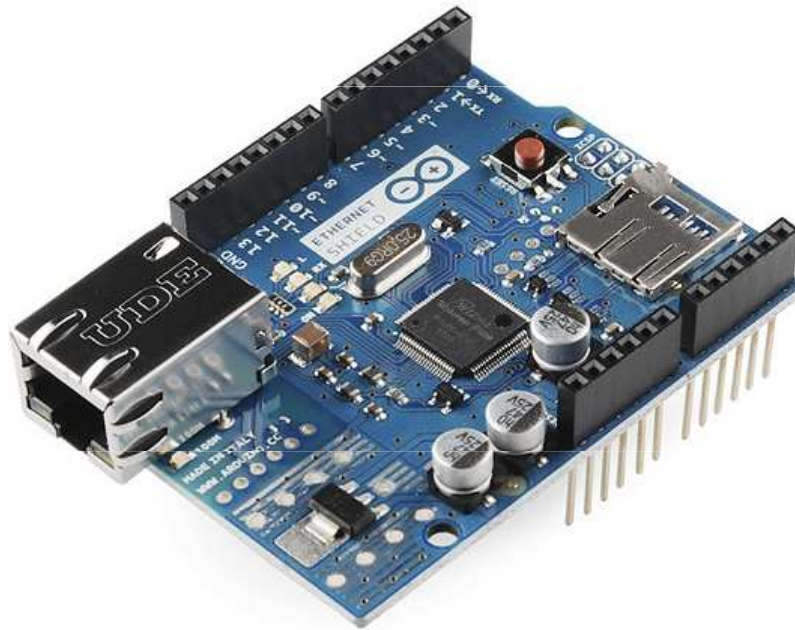
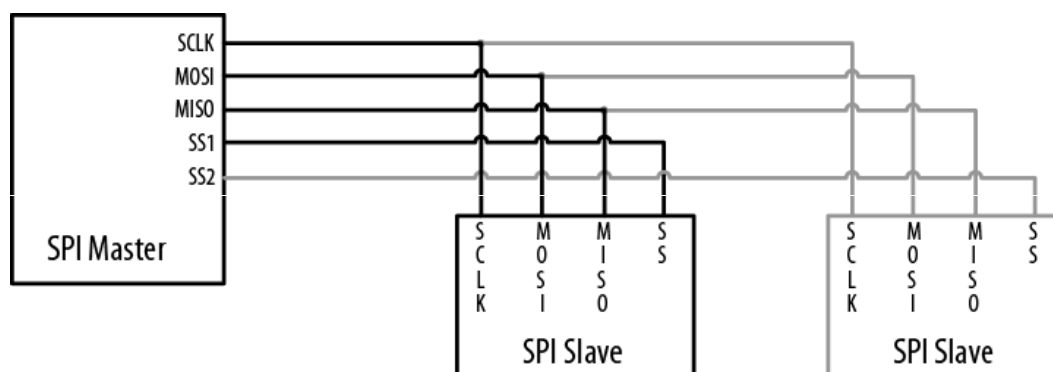


Ethernet Shield



Protocolo SPI

- **Serial Peripheral Interface.**
- **Trátase dun bus de comunicación de alta velocidade full-duplex.**
- **Emprega unha liña de selección por cada esclavo conectado.**
- **Arduino conta cunha librería “SPI.h”**



Pin-Out Ethernet Shield

- **Emprega o bus SPI nos seguintes pins:**

| SPI signal | Standard Arduino board | Arduino Mega |
|-------------------|------------------------|--------------|
| SCLK (clock) | 13 | 52 |
| MISO (data out) | 12 | 50 |
| MOSI (data in) | 11 | 51 |
| SS (slave select) | 10 | 53 |

- **A Ethernet Shield oficial permite conectarse a Internet a través da librería “Ethernet.h” e almacenar datos nunha tarxeta SD empregando a librería “SD.h”**

3

PRÁCTICA 1 – Cliente Web

- **Cargar o exemplo “WebClient” e interpretar o código**

```
/* CLIENTE WEB

 * Realiza unha busca en Google (http://www.google.com)
 usando unha Ethernet shield.
 * Circuito: Ethernet shield conectada nos pins 10, 11, 12, 13
 */

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

byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

//IPAddress server(74,125,232,128); // Se non usamos DNS--> Dirección IP Google
char server[] = "www.google.com"; // Nome do buscador (usando DNS)

IPAddress ip(192,168,1,177);// Dirección IP por defecto (Se falla DHCP).

EthernetClient client; //Declaración dun obxecto Cliente.
```

4

PRÁCTICA 1 – Cliente Web

```
void setup() {
  Serial.begin(9600); //Comunicacions serie como depuracion

  // Inicialización da conexión Ethernet
  if (Ethernet.begin(mac) == 0){
    Serial.println("Fallo DHCP");
    Serial.println("Uso IP por defecto");
    Ethernet.begin(mac, ip);
  }
  else Serial.println("DHCP correcto");

  // Dase 1 seg para inicializar...
  delay(1000);
  Serial.println("Conectando...");

  //Intento conexión ao servidor (HTTP--> Porto 80)
  if (client.connect(server, 80)) {
    Serial.println("Conectado ao servidor");
    // Facer a petición HTTP:
    client.println("GET /search?q=APEEGA HTTP/1.1");
    client.println("Host: www.google.com");
    client.println("Desconectando...");
    client.println();
  }
  else {
    Serial.println("Fallo conectando ao servidor");
  }
}
```

5

PRÁCTICA 1 – Cliente Web

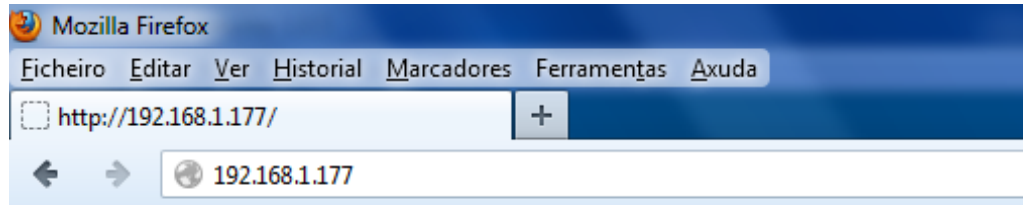
```
void loop()
{
  // Recepción da información do servidor
  // Paso a canle serie carácter a carácter
  if (client.available()) {
    char c = client.read();
    Serial.print(c);
  }

  // Se o cliente remata a conexión...
  if (!client.connected()) {
    Serial.println();
    Serial.println("Desconectando...");
    client.stop();
    while(true); //Bucle infinito
  }
}
```

6

PRÁCTICA 2 – Servidor Web

- **Cargar o exemplo “WebServer” e interpretar o código**



A entrada 0 ten o valor: 227
A entrada 1 ten o valor: 279
A entrada 2 ten o valor: 323
A entrada 3 ten o valor: 362
A entrada 4 ten o valor: 320
A entrada 5 ten o valor: 261

7

PRÁCTICA 2 – Servidor Web

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

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

// Inicializa un Servidor Ethernet no porto 80( ( HTTP por defecto):
EthernetServer server(80);

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

  // Inicializacion conexion Ethernet:
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print("O servidor esta na IP:");
  Serial.println(Ethernet.localIP());
}
```

8

PRÁCTICA 2 – Servidor Web

```
void loop() {
  // A espera de clientes...
  EthernetClient client = server.available();
  if (client) {
    Serial.println("Novo cliente!");
    // Codigo HTML para detectar peticion de datos
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);

        if (c == '\n' && currentLineIsBlank) {
          // Envia cabeceira HTTP estandar
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println("Connection: close");
          client.println();
          client.println("<!DOCTYPE HTML>");
          client.println("<html>");
          // Forza que o navegador refresque a paxina automaticamente cada 5 segundos:
          client.println("<meta http-equiv='refresh' content='5'>");
        }
      }
    }
  }
}
```

9

PRÁCTICA 2 – Servidor Web

```
// Percorrido das 6 entradas analoxicas:
for (int analogChannel = 0; analogChannel < 6; analogChannel++) {
  int sensorReading = analogRead(analogChannel);
  client.print("A entrada ");
  client.print(analogChannel);
  client.print(" ten o valor: ");
  client.print(sensorReading);
  client.println("<br />");
}
client.println("</html>");
break;
}
if (c == '\n') {
  currentLineIsBlank = true;
}
else if (c != '\r') {
  currentLineIsBlank = false;
}
}
}
// Tempo para qua o cliente reciba os datos...
delay(1);
client.stop();
Serial.println("Cliente desconetado (");
}
}
```

10

PRÁCTICA 3 – Servidor Temperatura

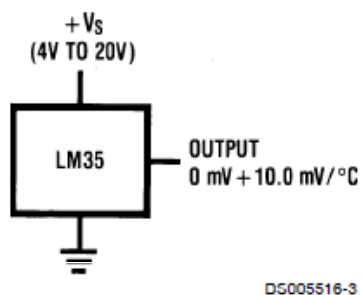
- **Descripción: Modificar o programa servidor Web para que amose a temperatura en °C obtida dun sensor LM35.**



11

PRÁCTICA 3 – Servidor Temperatura

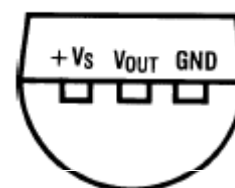
- **Características principais do sensor**



DS005516-3

FIGURE 1. Basic Centigrade Temperature Sensor (+2°C to +150°C)

TO-92
Plastic Package



BOTTOM VIEW

DS005516-2

Order Number LM35CZ,

12

PRÁCTICA 4 – Control por Web

- Cargar o exemplo “Web_control_rele” e interpretar o código.
- O programa permite o control dunha salida dixital a través dunha páxina web



13

PRÁCTICA 4 – Control por Web

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

//Declaración de direcciones MAC e IP.
byte mac[]={0xDE,0xAD,0xBE,0xEF,0xFE,0xED}; //MAC
IPAddress ip(192,168,1,177); //IP
EthernetServer servidor(80);

int ledPin=8;
String readString=String(30);
String state=String(3); //String para almacenar o estado

void setup()
{
  Ethernet.begin(mac, ip);
  servidor.begin();
  pinMode(ledPin,OUTPUT);
  digitalWrite(ledPin,LOW);
  state="OFF"; // Inicialmente apagada
}
```

14

PRÁCTICA 4 – Control por Web

```
void loop()
{
  EthernetClient cliente= servidor.available();

  if(cliente)
  {
    boolean lineaenblanco=true;
    while(cliente.connected())//Cliente conectado
    {
      if(cliente.available())
      {
        char c=cliente.read();
        if(readString.length()<30)//Lemos petición HTTP caracter a caracter
        {
          readString.concat(c); //Almacenar os caracteres na variable readString
        }
        if(c=='\n' && lineaenblanco)//Se a petición HTTP finalizou
        {
          int LED = readString.indexOf("LED=");

          if(readString.substring(LED,LED+5)== "LED=T") // Se na web se manda True...
          {
            digitalWrite(ledPin ,HIGH);
            state="ON";
          } else if (readString.substring(LED,LED+5)== "LED=F") //Se na web se manda False...
          {
            digitalWrite(ledPin,LOW);
            state="OFF";
          }
        }
      }
    }
  }
}
```

15

PRÁCTICA 4 – Control por Web

```
//Cabeceira HTTP estándar
cliente.println("HTTP/1.1 200 OK");
cliente.println("Content-Type: text/html");
cliente.println();
cliente.println("<html>");
cliente.println("<head>");
//Titulo da páxina
cliente.println("<title>LAMPADA ON/OFF</title>");
cliente.println("</head>");
cliente.println("<body width=100% height=100%>");
//Texto na paxina (centrado):
cliente.println("<center>");
cliente.println("<h1>LAMPADA ON/OFF</h1>");
cliente.print("<br><br>");
//Liñas para indicar o estado da lampada
cliente.print("Estado da lampada: ");
cliente.print(state);
//Botóns para accionar os comando HTML:
cliente.print("<br><br><br><br>");
cliente.println("<input type=submit value=ON style=width:200px;height:75px onClick=location.href='./?LED=T\ '>");
cliente.println("<input type=submit value=OFF style=width:200px;height:75px onClick=location.href='./?LED=F\ '>");
cliente.println("</center>");
cliente.println("</body>");
cliente.println("</html>");
cliente.stop(); //Pêcho conexión co cliente
readString="";
}
}
```

16