

UNIDAD DE GESTIÓN DE TECNOLOGÍAS

**CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN &
AVIÓNICA**

TRABAJO DE GRADUACIÓN

**TEMA: “IMPLEMENTACIÓN DE 4 MÓDULOS PARA PRACTICAS DE
COMUNICACIÓN ETHERNET EMPLEANDO ARDUINO MEGA
PARA EL LABORATORIO DE INSTRUMENTACIÓN VIRTUAL DE LA
UNIDAD DE GESTIÓN DE TECNOLOGÍAS”.**

AUTOR: ÑACATA OÑA, MAICOL DANIEL

PLANTEAMIENTO DEL PROBLEMA

- ◉ El laboratorio de Instrumentación Virtual de la UGT de la Universidad de Fuerzas Armadas – ESPE, no cuenta con placas Arduino Mega y Shield de Comunicación Ethernet que permita a los alumnos de sexto Nivel de la carrera de Electrónica desarrollar proyectos relacionados con estos dos dispositivos electrónicos.
- ◉ Al no contar con los expuestos se está limitando a los señores estudiantes a que desarrollen proyectos que de manera eficiente se pueden realizar utilizando las placas Arduino en aplicaciones como por ejemplo la domótica.

Objetivo general

- Implementar 4 módulos para prácticas de comunicación Ethernet empleando Arduino Mega para el laboratorio de Instrumentación Virtual de la UGT de la Universidad de las Fuerzas Armadas-ESPE.

Objetivos específicos.

- Investigar acerca de los Shield Ethernet para Arduino y el protocolo TCP/IP.
- Realizar pruebas de comunicación entre el Arduino Mega y el Shield Ethernet.
- Realizar el control señales digitales y el monitoreo de señales analógicas mediante la comunicación Ethernet

Ethernet

- Ethernet es una popular tecnología LAN (Red de Área Local) que utiliza el Acceso múltiple con portadora y detección de colisiones (Carrier Sense Múltiple Access with Collision Detection, CSMA/CD) entre estaciones con diversos tipos de cables.

Método de Acceso

- El método de acceso que usa Ethernet es el Acceso múltiple con portadora y detección de colisiones, CSMA/CD).
- CSMA/CD es un conjunto de reglas que determina el modo de respuesta de los dispositivos de red cuando dos de ellos intentan enviar datos en la red simultáneamente. La transmisión de datos por múltiples equipos simultáneamente a través de la red produce una colisión.
- Si el dispositivo detecta de nuevo una colisión, espera el doble antes de intentar retransmitir el mensaje.

Características de Ethernet

- Es PASIVO, es decir, no requiere una fuente de alimentación propia, y por tanto, no falla a menos que el cable se corte físicamente o su terminación sea incorrecta.
- Se conecta utilizando una topología de bus en la que el cable está terminado en ambos extremos.
- Utiliza múltiples protocolos de comunicación y puede conectar entornos informáticos heterogéneos, incluyendo Netware, UNIX, Windows y Macintosh.

Velocidad de Transferencia

- Ethernet estándar, denominada 10BaseT, soporta velocidades de transferencia de datos de 10 Mbps sobre una amplia variedad de cableado. También están disponibles versiones de Ethernet de alta velocidad. Fast Ethernet (100BaseT) soporta velocidades de transferencia de datos de 100 Mbps y Gigabit Ethernet soporta velocidades de 1 Gbps (gigabyte por segundo) o 1,000 Mbps.

Hardware utilizados por Ethernet

- **NIC, o adaptador de red Ethernet,** permite el acceso de una computadora a una red. Cada adaptador posee una dirección MAC que la identifica en la red y es única. Una computadora conectada a una red se denomina nodo.

Arduino

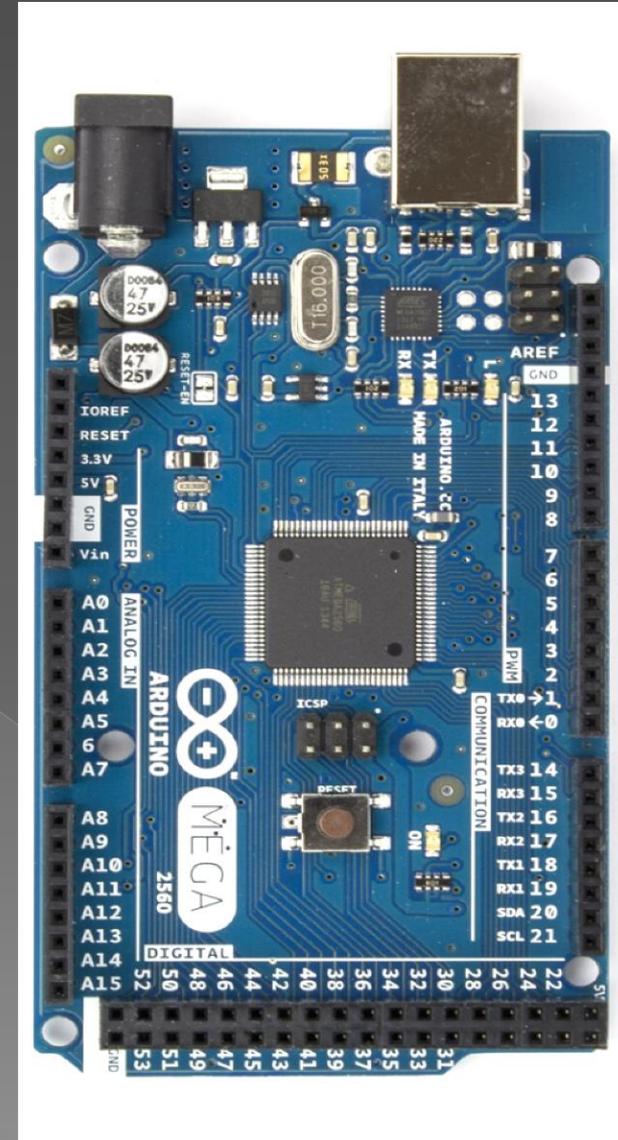
- Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar.
- Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores
- Las placas pueden ser hechas a mano o compradas montadas de fábrica; el software puede ser descargado de forma gratuita. Los ficheros de diseño de referencia (CAD) están disponibles bajo una licencia abierta

Lenguaje de programación

- El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Processing)
- Processing es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital

Arduino Mega 2560

- El Arduino Mega 2560 es una placa electrónica basada en el Atmega2560. Cuenta con 54 pines digitales de entrada / salida (de los cuales 15 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UARTs (hardware puertos serie), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un terminal ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar al microcontrolador; simplemente conectarlo a un ordenador con un cable USB o con un adaptador de CA o la batería a CC para empezar.



Arduino Mega 2560

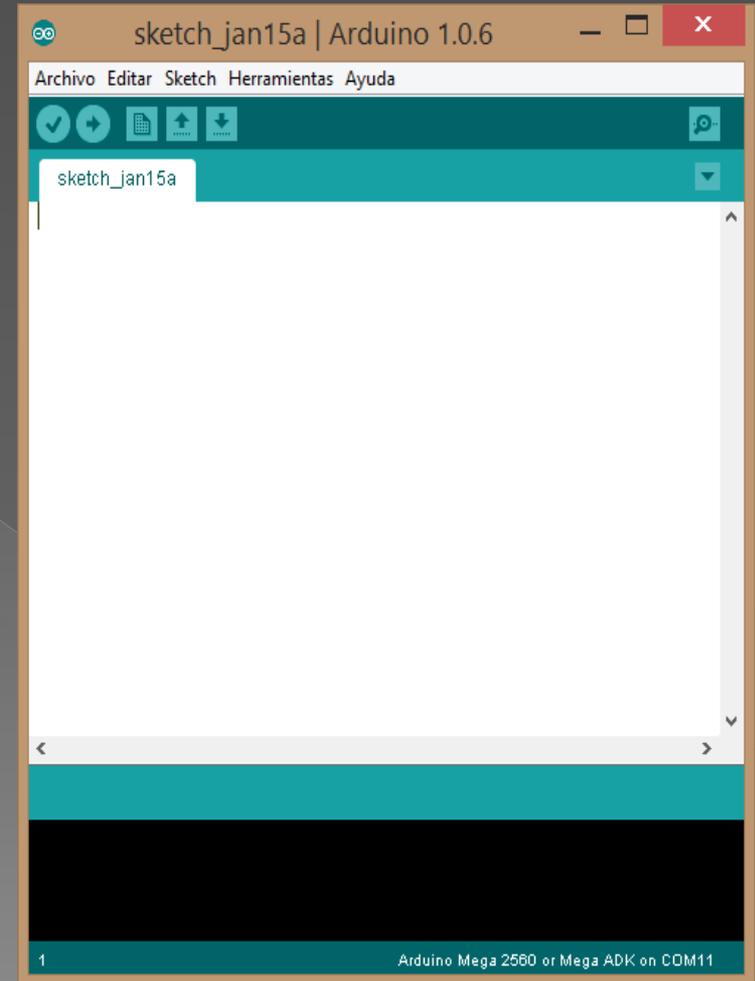
- Micronontrolador Atmega 2560
- Voltaje de operación 5V
- Voltaje de entrada (recomendado) 7 a 12 V
- Voltaje de entrada (limites) 6 a 20 V
- Pines de E/S digital para PWM) 54 (de los cuales 15
- Pines de entrada analógica 16
- Corriente DC por pin de E/S 40 mA
- Corriente Dc para pin 3.3 V 50 mA
- Memoria Flash 256 KB de los cuales 8KB es usado por el bootloader(Un gestor de arranque)
- SRAM 8 KB
memoria estática de acceso aleatorio
- EEROM 4 KB
programada, borrada y reprogramada eléctricamente
- Velocidad de reloj 16 MHz

Shield Ethernet Arduino

- El Shield Ethernet Arduino permite conectar a Arduino a Internet o a una red LAN en cuestión de minutos. Sólo tiene que conectar este módulo en su placa Arduino, conectarlo a su red con un cable RJ45.
- Utilice la librería de Ethernet para escribir programas que se conectan a Internet a través del shield.
- Hay una ranura para tarjetas micro-SD en la tarjeta, que se puede utilizar para almacenar archivos para enviar a través de la red.

Creación de programas (sketches)

- El software escrito utilizando Arduino se llama sketch. Estos sketches se escriben en el editor de texto.
- El área de mensajes proporciona información mientras se guarda o se compila programas y también muestra los errores.
- La esquina derecha inferior de la ventana muestra la tarjeta empleada y el puerto serie (COM). Los botones de la barra de herramientas le permiten verificar y cargar programas, crear, abrir y guardar dibujos, y abrir el monitor de serie.



Estructura de programación

- ◉ La estructura de programación en Arduino está formada por la función `setup()` y la función `loop()`.
- ◉ **Setup():** La función `setup()` se llama cuando se inicia un sketch. para inicializar variables, asignación de dirección a los pines de E/S digitales, inicio de librerías, etc. La función de configuración sólo se ejecutará una vez, después de cada arranque o reinicio de la placa Arduino.
- ◉ **Loop():** Después de crear una función `setup()`, que inicializa y establece los valores iniciales, la función `loop()` hace un bucle de forma consecutiva, lo que permite el programa para cambiar y responder. Lo utilizan para controlar activamente la placa Arduino.

Funciones de E/S digitales

`pinMode()`

Configura el pin especificado a comportarse ya sea como una entrada o una salida.

- ◉ **Sintaxi**

```
pinMode(pin, mode);
```

- ◉ **Parámetros**

pin: número de pin cuyo modo se desea fijar

mode: INPUT, OUTPUT

digitalWrite()

- Escribe un valor de alto o bajo en el pin digital
- Si el pin se ha configurado como una salida con pinMode (), su voltaje se establece en el valor correspondiente: 5V para alto, 0V (tierra) para bajo
- El pin 13 tiene soldado un LED y una resistencia en la placa Arduino y nunca podrá ser usado como entrada digital.

- **Sintaxi**

digitalWrite(pin,valor);

- **Parámetros**

pin: número de pin

valor: HIGH o LOW

digitalRead()

- Lee el valor de un pin digital especificado, ya sea alto o bajo.

- **Sintaxi**

```
digitalRead(pin);
```

- **Parámetro**

pin: el número del pin digital que desea leer

Funciones de E/S analógicas

`analogRead()`

- Lee el valor del pin analógico especificado. La placa Arduino Mega contiene 16 canales analógicos de 10 bits. Esto significa que va a asignar tensiones de entrada entre 0 y 5 voltios en valores enteros entre 0 y 1023
- **Sintaxi**
`analogRead(pin);`
- **Parametro**
pin: número del pin analógico de entrada que se desea leer

analogWrite()

- Escribe un valor analógico (onda PWM) en un pin. Se puede utilizar para encender un LED en diferentes brillos o controlar un motor a distintas velocidades. Después de llamar a `analogWrite()`, el pin generará una onda cuadrada estable del ciclo de trabajo especificado hasta la siguiente llamada a `analogWrite()`. La frecuencia de la señal PWM es de aproximadamente 490 Hz.
- **Sintaxi**
`analogWrite(pin,valor);`
- **Parámetros**
pin: pin donde se escribe el valor
valor: ciclo de trabajo entre 0 y 255

Telnet

- Telnet (Teletype Network) es el nombre de un protocolo de red que nos permite viajar a otra máquina para manejarla remotamente como si estuviéramos sentados delante de ella. También es el nombre del programa informático que implementa el cliente.
- Para que la conexión funcione, como en todos los servicios de Internet, la máquina a la que se acceda debe tener un programa especial que reciba y gestione las conexiones. El puerto que se utiliza generalmente es el 23.

HTML creación de páginas web

- HTML es un lenguaje universal, que funciona en cualquier plataforma (Windows, Macintosh, Unix, OS/2, etc.) y con cualquier navegador o browser (Internet Explorer, Mozilla Firefox, etc.).
- formato ASCII2 (y con 7 bits en lugar de ocho, por lo que ni tan siquiera tiene acentos ni otros caracteres especiales).
- Por tanto, para que un documento HTML sea algo más que simples caracteres básicos, debe contener, además de dicho texto, instrucciones se denominan etiquetas o tags y se distinguen del texto porque van entre guiones (< >)3.

| Código | Función |
|----------------|-------------|
| ... | Negrita |
| <I>...</I> | Cursiva |
| <U>...</U> | Subrayado |
| _{...} | Subíndice |
| ^{...} | Superíndice |

Hyperterminal



- cliente para hacer conexiones telnet por medio de los puertos serie (por ejemplo COM1) con dispositivos externos. Estos dispositivos pueden variar e incluyen opciones tales como equipos de radio comunicación, los robots, y los instrumentos utilizados para mediciones científicas y equipos de red.

Parpadeo de pin 13 de la placa Arduino Mega 2560



The image shows a screenshot of the Arduino IDE interface. The title bar reads "parpadeo | Arduino 1.0.6". The menu bar includes "Archivo", "Editar", "Sketch", "Herramientas", and "Ayuda". The toolbar contains icons for saving, running, uploading, and downloading. The main editor area shows the following code:

```
parpadeo $
void setup()
{
  pinMode(13,OUTPUT);
}

void loop()
{
  digitalWrite(13,HIGH);
  delay(400);
  digitalWrite(13,LOW);
  delay(400);
}
```

- Para el parpadeo del led SMD que esta soldado en la placa Arduino, se escribe en el software Arduino lo siguiente:
- En el parámetro void setup() se configura mediante pinMode(13,OUTPUT) el modo del pin en este caso el número 13 que corresponde al led como salida.
- El void loop() que es el lazo indefinido, con digitalWrite(13,HIGH) se coloca en alto el pin13
- Con delay(400) se realiza un retardo de 400 ms
- Con digitalWrite(13,LOW), se coloca en bajo el pin 13 y finalmente con delay(400) se realiza otro retardo de 400 ms. De esta forma el led que corresponde al pin 13 estará parpadeando cada 400 ms.

Adquisición de señales analógicas

- La tarjeta Arduino Mega 2560 tiene 16 entradas analógicas, etiquetadas como A0 hasta la A15, para el ejemplo se utiliza la entrada A0. Cada entrada admite una variación de voltaje entre 0 y 5V, como la resolución del ADC del microcontrolador Atmega 2560 es de 10 bits, se tiene un valor del 1023 que corresponde a los 5V externos en el canal de entrada analógico; es decir, que para convertir ese valor a voltaje es necesario aplicar una regla de tres.

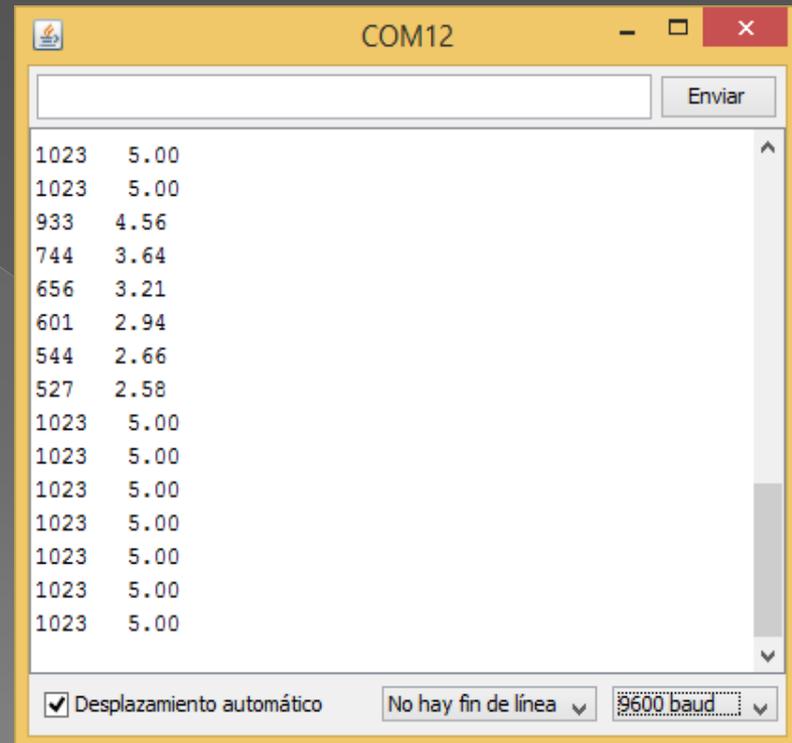
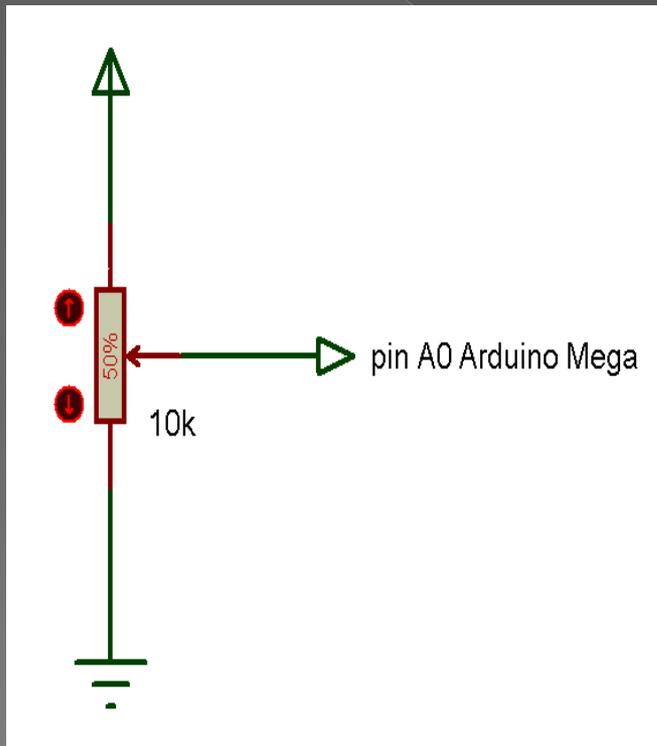
```
practica3 | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda
practica3
int valor,valor2;
float y;
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  valor=analogRead(0);
  valor2=valor*5;
  y=valor2/1023.0;
  Serial.print(valor);
  Serial.print(" ");
  Serial.println(y);

  delay(500);
}
```

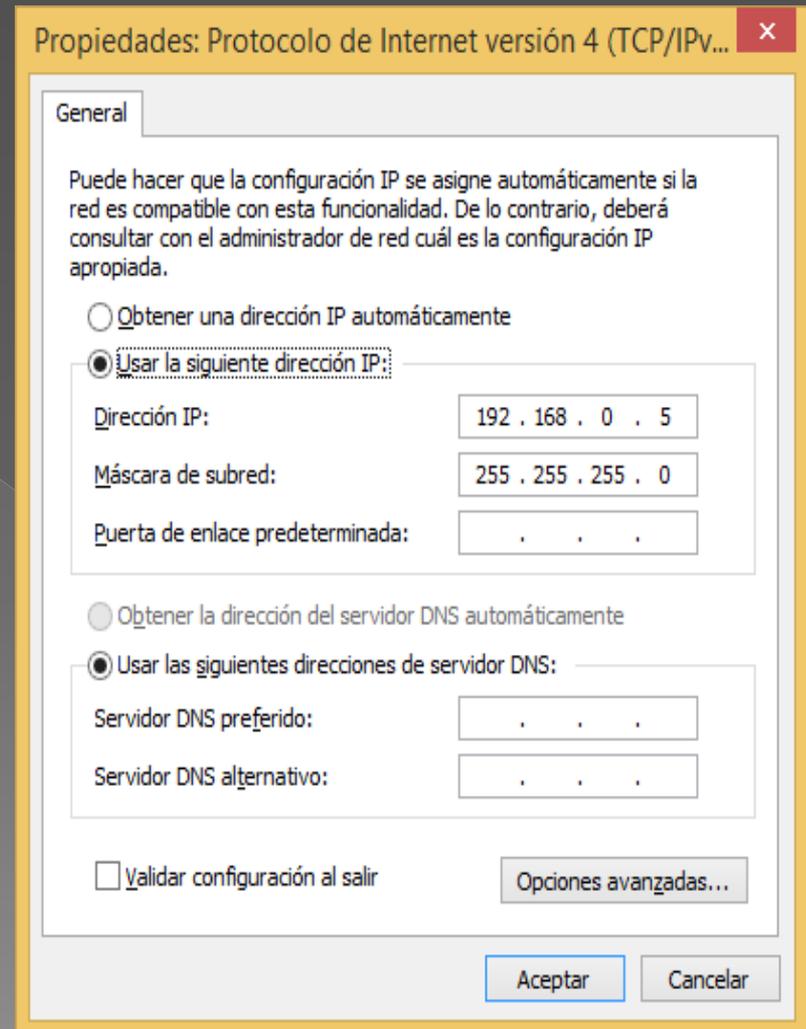
- Se declaran dos variables globales de tipo int, llamadas valor y valor2, almacena un valor de 2 bytes, que corresponde a un rango de -32768 a +32767.
- La variable y es de tipo flotante, sirve para guardar el valor de la regla de tres simple que es el voltaje.
- Para visualizar el valor entero entre 0 a 1023 y su correspondiente voltaje, se emplea el visor serial del software Arduino. Primero se configura la velocidad de transmisión a 9600 con la función Serial.begin(9600). Para mostrar los datos en el visor serial, se emplea la función Serial.Print.

- Para comprobar el funcionamiento de la adquisición analógica se emplea un potenciómetro conectado el terminal variable a la entrada analógica A0.

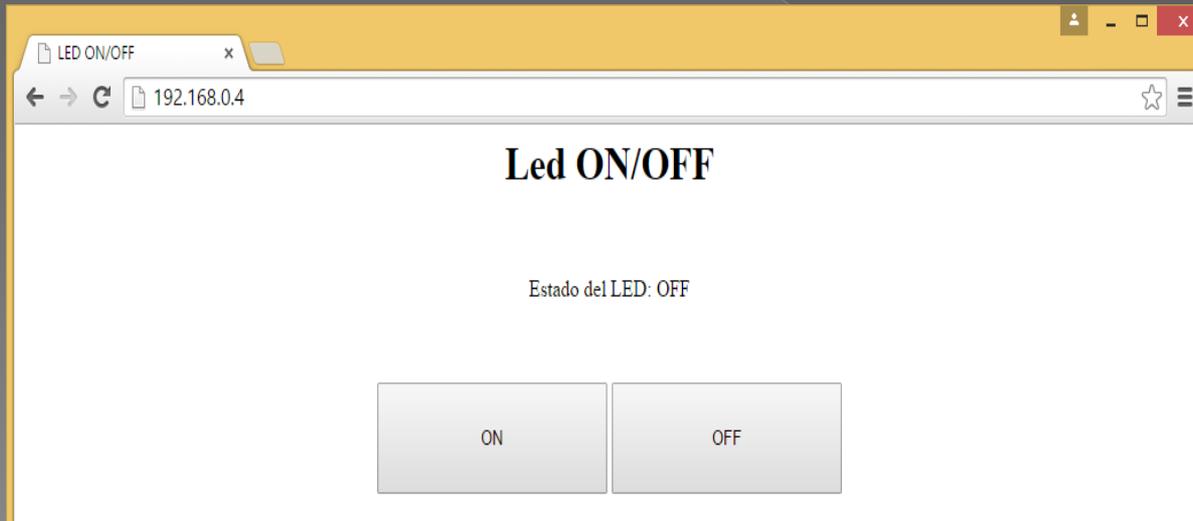
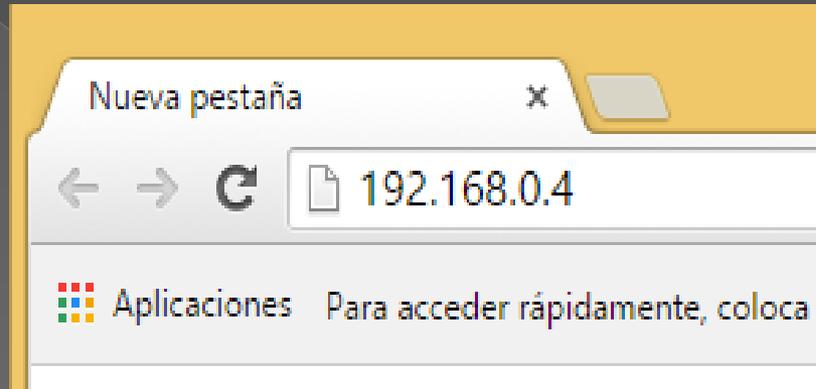


Encendido/apagado led desde página HTML

- Los pasos para el control de un led conectado al pin 8 del Arduino Mega 2560 desde un navegador web son los siguientes:
- Copie la dirección MAC del shield Ethernet para este ejemplo es 90-A2-DA-0D-D5-2D que necesario cuando se programa en Arduino.
- Inserte el shield Ethernet en la tarjeta Arduino Mega 2560
- Conecte un cable Ethernet desde el shield hacia el puerto Ethernet del computador.
- Energice la tarjeta Arduino mediante el puerto USB del computador
- Asigne una dirección IP estática en el computador por ejemplo 192.168.0.5

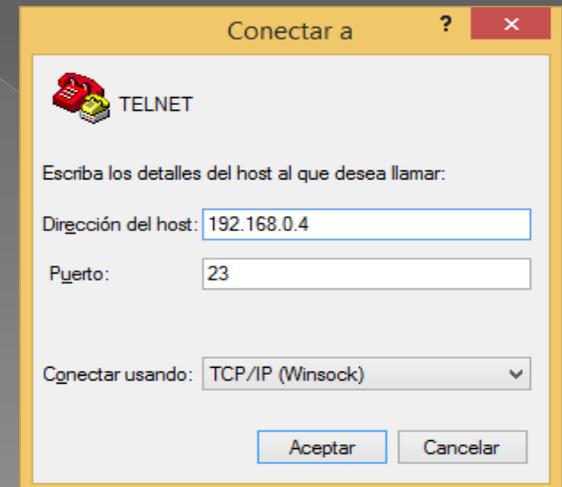
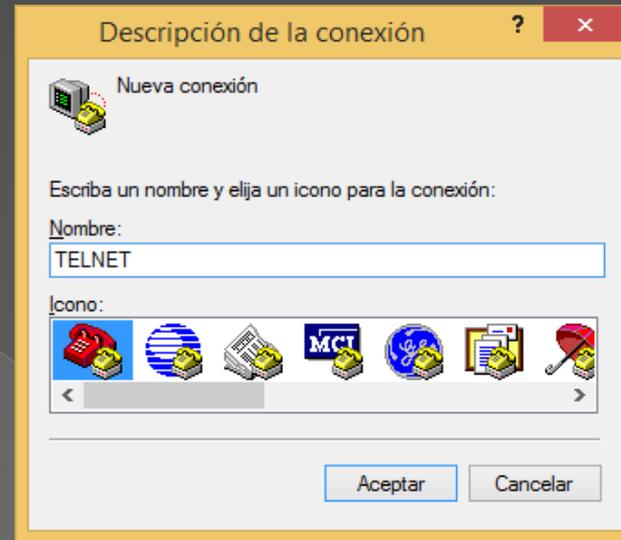


- Abra un navegador web y escriba la dirección IP 192.168.0.4, que es la dirección del shiled Ethernet,,



Encendido/apagado led mediante Hyperterminal

- Los pasos para crear una nueva conexión en hyperterminal son los siguientes:
- Abrir hyperterminal y escriba un nombre para la conexión
- Pulse en Aceptar, en la nueva conexión seleccione TCP/IP
- Escriba la dirección IP del shield Ethernet, pulse Aceptar



Gracias