



# Arduino Inicial

Ing. Juan C. Abdala

# Clase 4



# Objetivo de esta clase

- Aprender a crear tu propia librería.
- Aprender conceptos básicos de una red de computadoras.
- Aprender que es el protocolo HTTP.
- Aprender sobre el lenguaje HTML
- Aprender cómo se usa el periférico Ethernet ENC28J60.



Nuestra Librería

# Como podemos escalar

Debemos crear nuestras librerías.

En la misma carpeta que el sketch creamos **2 archivos**.

Un HEADER(cabecera) y el Código Fuente



# Simple

## Carpeta Librería

- Libreria.h
- Libreria.cpp
- Carpeta examples



# HEADER libreria .h

- Este archivo es donde vamos a declarar los métodos y atributos necesarios para la librería.
- extern evita que llame instancias en el código.

En el h debe tener.

```
#ifndef __UTILIDADES__  
#define __UTILIDADES__  
class UtilidadesClass  
{  
    public:  
        static void inicio();  
        static int funcion(int parametro);  
  
};  
  
#endif
```



# FUENTE librería.cpp

En este archivo está la lógica de la librería aquí escribiremos el código.

Aquí deben estar declarados todos los métodos del archivo *Libreria.h* .

```
#include "utilidades.h"

void UtilidadesClass::inicio(void)
{
    //Algo que debe poner en el setup
}

int UtilidadesClass::funcion(int parametro)
{
    //La funcion que quiera realizar
    return algunParametro;
}
```





# En mi proyecto

En mi proyecto solo debo llamar:

```
#include "utilidades.h"
```

```
UtilidadesClass utilidades;
```

```
void setup(){  
  utilidades.inicio();  
}
```

```
void loop(){  
  dato = utilidades.funcion(10);  
}
```



Red de computadoras.

# Red de computadoras

- En esencia, una red es un conjunto de equipos informáticos interconectados entre sí.
- Redes de área local. LAN (Local Area Network). (Nuestra RED)
- Redes de área amplia. WAN (Wide Area Network). (Internet)



# GATEWAY

- La pasarela (gateway) o puerta de enlace es el dispositivo que permite interconectar redes de computadoras con protocolo de comunicaciones y arquitecturas diferentes a todos los niveles de comunicación.
- La dirección IP de una pasarela a menudo se parece a 192.168.1.1 o 192.168.0.1 para las LAN.
- Nuestros periféricos no saben llegar a otros periféricos necesitan que el gateway(router) le indique.



# Red de Computadoras

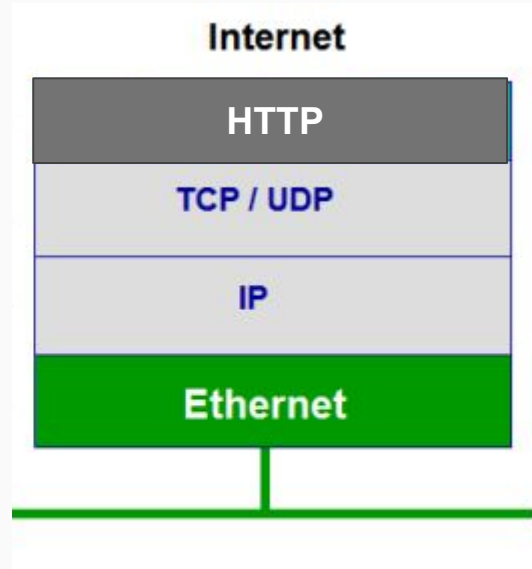


# Protocolos y Modelos

TCP/IP model	Protocols and services	OSI model
Application	HTTP, FTTP, Telnet, NTP, DHCP, PING	Application
Transport		Presentation
Network		Session
Network Interface	TCP, UDP	Transport
	IP, ARP, ICMP, IGMP	Network
	Ethernet	Data Link
		Physical



# Nuestro Proyecto



# Identificadores

Para ethernet -> Dirección MAC

Para RED -> Dirección IP





# Direccion MAC

- Es un identificador de 48 bits (6 bloques hexadecimales) que corresponde de forma única a una tarjeta o dispositivo de red
- Para varias tecnologías de capa 2, Ethernet, WiFi...
- Dirección MAC: 00-AA-11-23-F1-ED.
- Generalmente Fija en el Hardware.



# Direccion IP

- Es una etiqueta numérica que identifica, de manera lógica y jerárquica, a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP.
- Ej: dirección IPv4: 10.128.1.253



# DHCP

- El protocolo DHCP sirve principalmente para distribuir direcciones IP en una red de manera dinámica.
- Si no está debemos brindarle la configuración de forma manual a nuestro dispositivo.



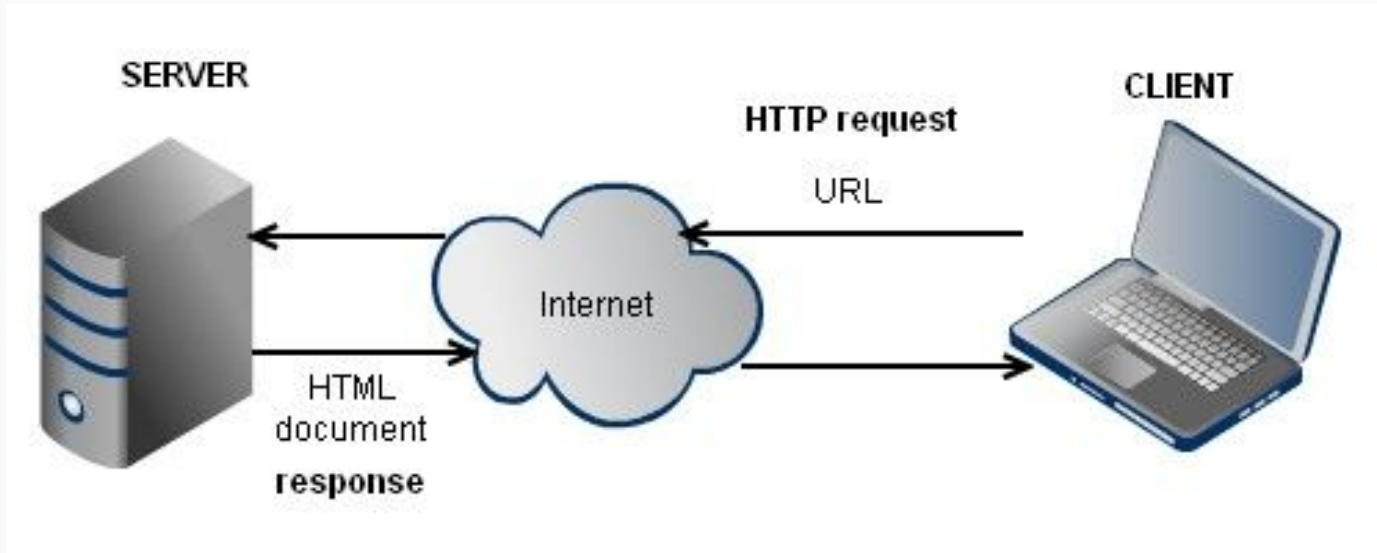
HTTP

# HTTP

- Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un **cliente y un servidor**.
- Al cliente que efectúa la petición (un navegador web) se lo conoce como "**user agent**" (agente del usuario).
- A la información transmitida se la llama recurso y se la identifica mediante un localizador uniforme de recursos (**URL**)



# ¿que es un server http?



# Métodos de petición

Define 8 métodos, sólo nos interesan 2:

**GET:** Pide una representación del recurso especificado. La petición puede ser simple, es decir en una línea o compuesta de la manera que muestra el ejemplo.

*GET /images/logo.png HTTP/1.1 obtiene un recurso llamado logo.png*

**POST:** Envía los datos para que sean procesados por el recurso identificado. Los datos se incluirán en el cuerpo de la petición. Esto puede resultar en la creación de un nuevo recurso o de las actualizaciones de los recursos existentes o ambas cosas.



HTML



# ¿Que es html?

- HTML, siglas de HyperText Markup Language («lenguaje de marcas de hipertexto»), hace referencia al lenguaje de marcado para la elaboración de páginas web.
- Es el lenguaje de los navegadores.



# HTML

Etiqueta: `<algo>`

Comienzo de una estructura `<algo>`

Fin de una estructura `</algo>`



# Html- Estructura basica

<HTML>

<HEAD>

<TITLE>Título de la página</TITLE>

</HEAD>

<BODY>

Aquí iría el contenido de la página

</BODY>

</HTML>



# HTML - párrafo

- Estructura `<p>` y `</p>`
- Se lo puede modificar como un párrafo de texto, alineación etc.
- A nosotros nos sirve para delimitar cómo se ven nuestras funciones



# HTML href

- `<A href="google.com"> link a google</A>`
- Crea un link hacia una direccion.
- Lo vamos a usar para ir saltando de Funciones. /función



# Ejemplo

<p>

Luz:

<a href="/ledon">On</a>

<a href="/ledoff">Off</a>

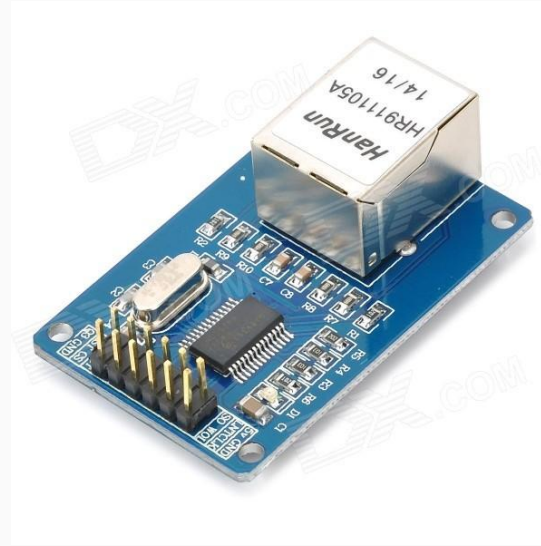
</p>



# Arduino Ethernet ENC28J60

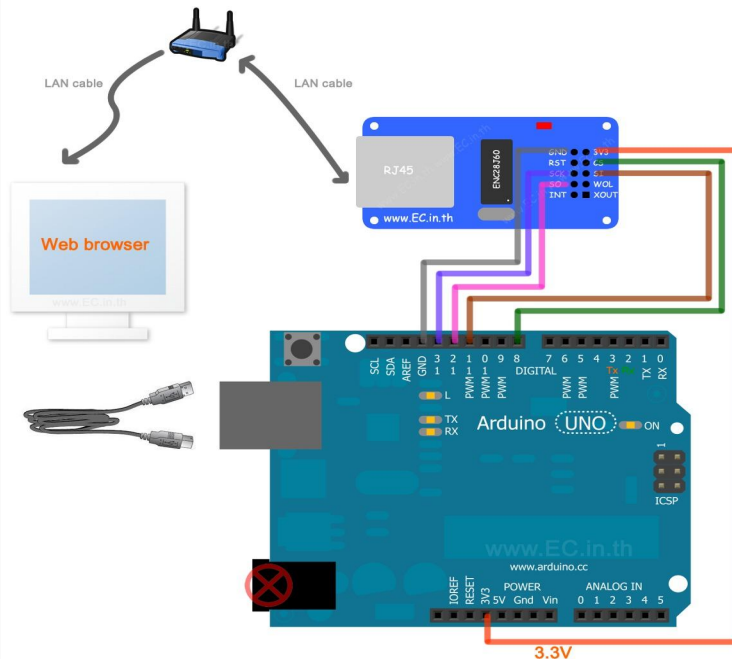
# ENC28J60

- SPI
- 3.3V
- Nos brinda el control de señales para ethernet.
- ~\$100





# Esquematico



**VCC - 3.3V**  
**GND - GND**  
**SCK - Pin 13**  
**SO - Pin 12**  
**SI - Pin 11**  
**CS - Pin 8**



# Libreria

- No es la estandar.
- <https://github.com/jcw/ethercard>
- Es la que mejor Funciona.
- O en nuestra carpeta de códigos la versión vieja.



# Funciones

//Se define un buffer para transferencia y recepción.

**byte Ethernet::buffer[700];**

//Se inicia indicandole la MAC y el tamaño de buffer.

**ether.begin(sizeof Ethernet::buffer, mymac)**



# IP

//IP Fija, mi direccion y la de gateway

**ether.staticSetup(myip, gwip))**

//IP dinamica

**ether.dhcpSetup()**



# Recibir info

//Recibo un paquete y tomo el tamaño del paquete

**word len = ether.packetReceive();**

//Apunto asi el lugar de memoria donde esta el paquete para despues accederlo

**word pos = ether.packetLoop(len);**



# Recibo

//Comprueba si lo que llego es una funcion GET con un URL específico.

```
if(strstr((char *)Ethernet::buffer + pos, "GET /LEDON") != 0)
```

```
{
```

**llamo a la función que me indica esa url.**

```
}
```



# Para enviar

```
//Creo un buffer se un tipo especial para envio
BufferFiller bfill;
//Le agrego cabecera TCP
bfill = ether.tcpOffset();
//Cargo los datos en modo html en el buffer
bfill.emit_p(PSTR( "<h3>Bienvenido a Arduino</h3>" ));
//Respondo a la peticion
ether.httpServerReply(bfill.position());
```



# API de ethernetcard

<http://jeelabs.net/pub/docs/ethercard/classEtherCard.html>





# Practico

Vamos al Practico 5:

Material: <https://github.com/jcabdala/ArduinoInicialUNC>



# Consultas:



abdalajc@gmail.com



@toniabdala

