

# SERVIDOR WEB AUTOACTUALIZABLE PARCIAL PARA CAPTAR SEÑALES ANALÓGICA Y DIGITAL USANDO AJAX

## MATERIALES:

- 1 Wemos D1 R1
- 1 Protoboard mediano
- 1 Resistor 10 a 14 Ohms (CA-AM-NE-NE)
- 4 alambres dupont M-M ROJO
- 1 alambre dupont M-M NARANJA
- 1 alambre dupont M-M NEGRO
- Compilador IDE Arduino
- 1 Potenciómetro 10 K
- 1 Resistor 1 KOhm (CA-NE-NE-CA)
- 1 alambre dupont M-M AMARILLO LARGO
- 1 Push button NA

## INSTRUCCIONES:

Compilar y grabar el siguiente código en el hardware:

```
String header = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n";
// Inicia la cadena cruda
String CadenaCruda = R"=====(
<!DOCTYPE html>
<html>
<head>
<meta name='viewport' content='width=device-width, initial-scale=1.0'/>
<meta charset='utf-8'>
<style>
  body {font-size:100%;}
  #main {display: table; margin: auto; padding: 0 10px 0 10px; }
  h2 {text-align:center; }
  p { text-align:center; }
</style>

<script>
  function actualizaContador()
  // al terminar su conteo descendente
  // el timer ejecuta obtenerCuenta
  {
    ajaxLoad('obtenerCuenta');
  }

  var SolicitudAjax = null;
  if (window.XMLHttpRequest) { SolicitudAjax =new XMLHttpRequest(); }
  // El objeto XMLHttpRequestEs el que permite la comunicación asíncrona (en segundo plano)
  // con el servidor.
  else
    { SolicitudAjax =new ActiveXObject("Microsoft.XMLHTTP"); }

  function ajaxLoad(ajaxURL)
  // Aquí está la parte de Ajax:
```

```

{
  if(!SolicitudAjax){ alert('AJAX is not supported.');
```

```

  SolicitudAjax.open('GET',ajaxURL,true);
  SolicitudAjax.onreadystatechange = function()
  {
    if(SolicitudAjax.readyState == 4 && SolicitudAjax.status==200)
    {
      var ResultadoDeAjax = SolicitudAjax.responseText;
      document.getElementById('cont_ParteDePagina').innerHTML = ResultadoDeAjax;
      //hace referencia a la parte que se va a actualizar
    }
  }
  SolicitudAjax.send();
}

```

```

setInterval(actualizaContador, 200);
// Cuando el timer inicia, es llamado actualizaContador
// después de 200 ms (porque el timer tiene un contador descendente)
// actualizaContador llama ajaxLoad()y es cuando se realiza el
// trabajo de desplegar el valor actualizado del contador

```

```

</script>
<title>ADC Auto refresh parcial AJAX</title>
</head>

```

```

<body>
  <div id='main'>
    <h2>Lectura analógica autoactualizable usando AJAX (Asynchronous JavaScript And XML)</h2>
    <!--establece un elemento id indicando que es el contador
    dividido de la página -->
    <p id='cont_ParteDePagina'>cont = 0</p>
    <!-- cont inicia en cero -->
  </div>
</div>
</body>
</html>
)====="; // Se cierra cadena cruda

```

```

#include <ESP8266WiFi.h>

```

```

const char* ssid = "MEGACABLE-S7353F";
const char* password = "De356HjDe";

```

```

WiFiServer server(80);

```

```

String request = "";

```

```

unsigned int cont = 0;
int pinEntradaD4 = 4; // Entrada /D4 de la Wemos D1
int valorEntradaD4 = 0;

void setup()
{
  Serial.begin(115200);
  Serial.println();
  Serial.println("Puerto serie a velocidad 115200");
  Serial.println();

  // Conectar a la red WiFi
  Serial.print(F("Conectando a ")); Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(500);
  }

  Serial.println("");
  Serial.println(F("[CONNECTED]"));
  Serial.print("[IP ");
  Serial.print(WiFi.localIP());
  Serial.println("]");

  // start a server
  server.begin();
  Serial.println("Servidor iniciado");

}
// -----

void loop()
{
  // Verifica si un cliente se ha conectado
  WiFiClient client = server.available();
  if (!client) { return; }

  // Lee la primera linea de la petición
  request = client.readStringUntil('\r');
  // El servidor está constantemente verificando por si hay una
  // solicitud, y cuando obtiene una, verifica si la petición
  // incluye la cadena "obtenerCuenta"
  if ( request.indexOf("obtenerCuenta") > 0 ) // si la petición
  // contiene la cadena "obtenerCuenta":

  {

```

```

cont ++; // se incrementa contador
client.print( header ); // se despliega en página Web
client.print("<html>");
client.print( "<h2> contador </h2>" );
client.print( "<h2>" );
client.print( cont );
client.print( "</h2>" );
int lecturaADC = analogRead(0); // lectura analógica en pin A0
client.print( "<h2> Lectura analógica a digital </h2>" );
client.print( "<h2>" );
client.print( lecturaADC );
client.print( "</h2>" );
valorEntradaD4 = digitalRead(pinEntradaD4); // lectura digital en pin D4
client.print( "<h2> El valor digital de entrada es: </h2>" );
client.print( "<h2>" );
client.print( valorEntradaD4 );
client.print( "</h2>" );

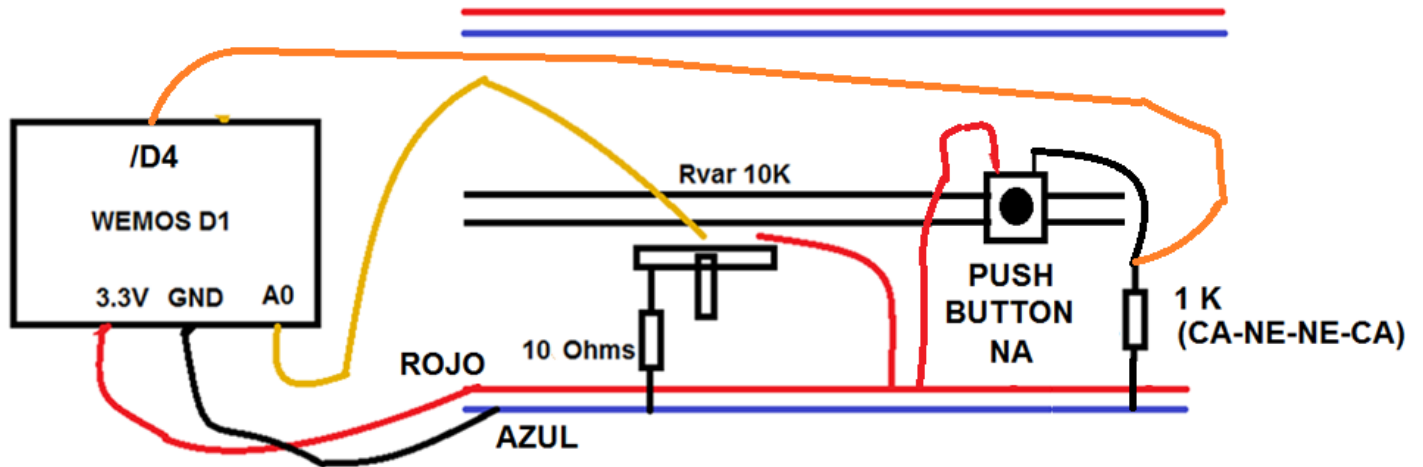
client.print("</html>");
Serial.print("cont="); //Se despliega en monitor
Serial.println(cont);
Serial.print("Lectura ADC="); //Se despliega en monitor
Serial.println(lecturaADC);
Serial.print("Lectura digital en D4= "); //Se despliega en monitor
Serial.println(valorEntradaD4);
}

else
// de lo contrario que la petición NO CONTIENE obtenerCuenta
//enviamos la página Web y reseteamos cont a cero
{
    client.flush();
    client.print( header );
    client.print( CadenaCruda );
    cont = 0;
}
delay(5);
}

```

## CONEXIONES:

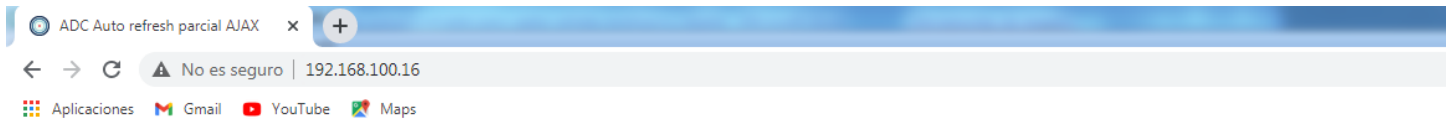
Realizar las conexiones de ambos circuitos en el mismo protoboard.



**PRUEBA:**

Después de grabar en el hardware activar el monitor serie (en el menú, herramientas) y observar la IP que deberá teclearse en el navegador.

Después de teclear la IP y girando el potenciómetro y sin presionar el botón, aparece algo similar a lo siguiente:



## Lectura analógica autoactualizable usando AJAX (Asynchronous JavaScript And XML)

**contador**

336

## Lectura analógica a digital

555

**El valor digital de entrada es:**

0

Girando el potenciómetro y presionando el botón, aparece algo similar a lo siguiente:

## **Lectura analógica autoactualizable usando AJAX (Asynchronous JavaScript And XML)**

**contador**

**545**

**Lectura analógica a digital**

**845**

**El valor digital de entrada es:**

**1**