

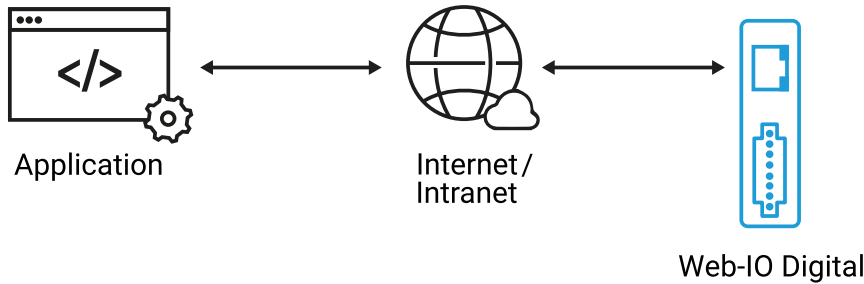
Aplicación al Web-IO digital:

# Web-IO digital - visualizar con JavaScript y Java-Applet en el Browser

Resumen de productos

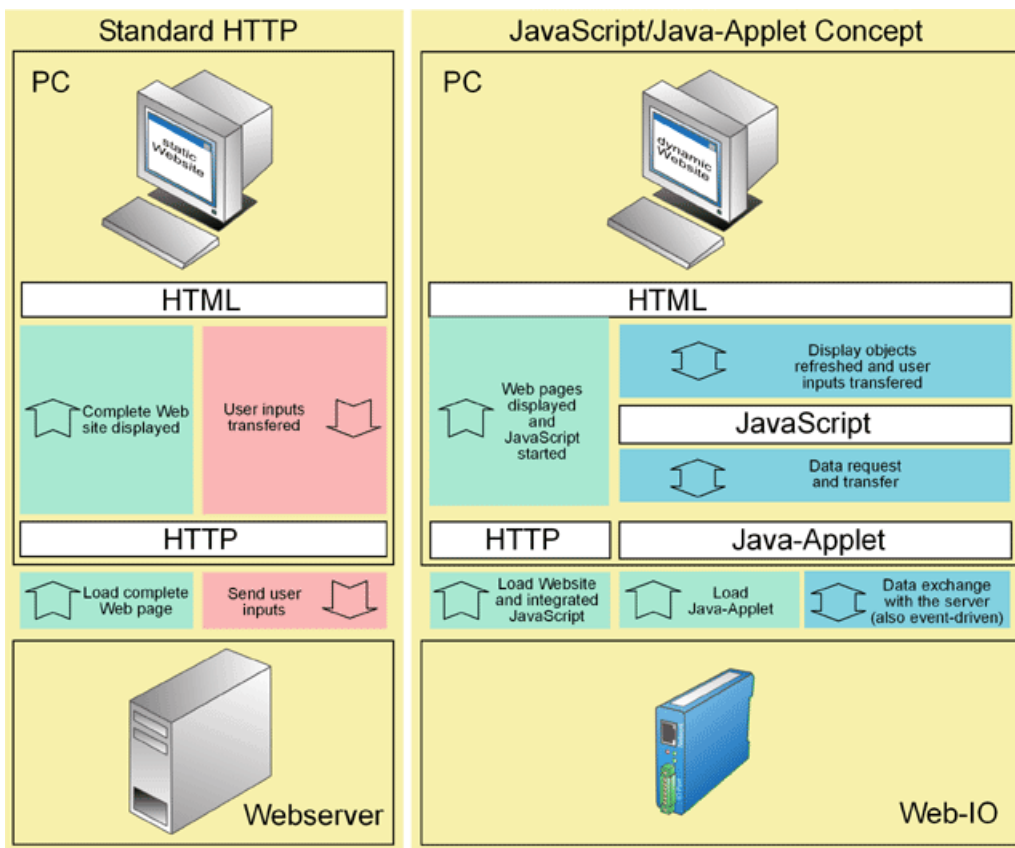
Sinopsis de aplicaciones

El Internet Browser es hoy día parte componente de todos los modernos sistemas operativos y se aprecia como un instrumento versátil de indicación al navegar por Internet.



Junto con los W&T Web-IOs el Browser se puede utilizar ahora también como elemento de indicación y de mando para aplicaciones dinámicas, técnicas.

El motor para la dinámica es aquí un Java-Applet, que se carga sencillamente del Web-IO en el Browser y que asume aquí el intercambio de datos de proceso con el Web-IO. Rutinas sencillas JavaScript cogen el estado actual de los IOs del Java-Applet en los cambios y adaptan la indicación en el Browser a las circunstancias en el campo.



La aplicación descrita a continuación muestra ejemplarmente el juego entre Browser, Java-Applet y Web-IO.

### ¿No tiene todavía un Web-IO y quiere probar el ejemplo presentado?

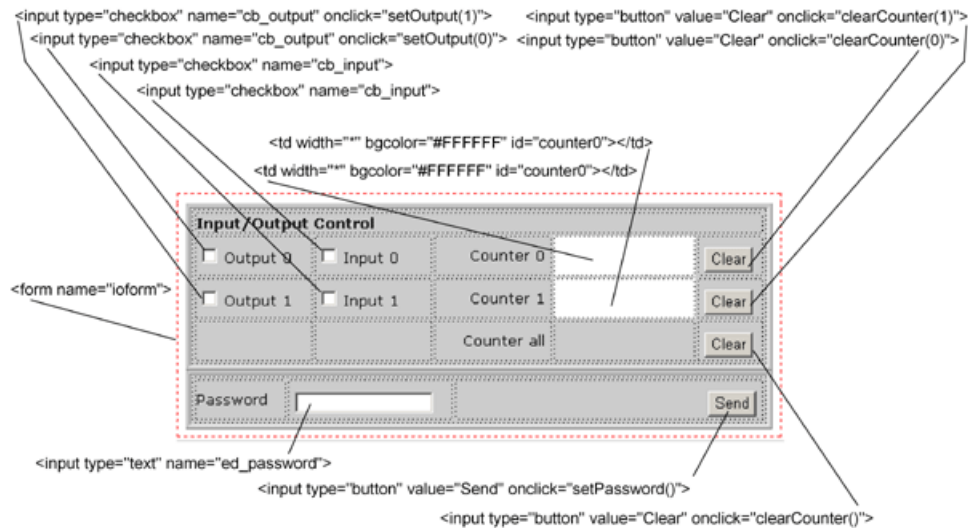
No hay problema: Le ponemos a disposición el Web-IO Digital 2xInput, 2xOutput PoE gratis durante 30 días. Rellene sencillamente un pedido muestra y le enviaremos el Web-IO para probar a cuenta abierta. Si nos devuelve el aparato dentro de los 30 días, le abonamos la factura completa.

[Al pedido muestra](#)

### Preparativos

- [abastecido con corriente,](#)
- [entradas y salidas conectadas](#)
- [conectado a su red](#)
- dotado con una dirección IP - con [WuTility](#) no hay problemas

## Recopilación de los diferentes elementos de manejo y objetos de visualización de la página web



### 1. Estructura base HTML de la página web

Colocar los elementos de mando y de indicación

Para una mejor estructuración se mete cada uno de los elementos en tablas y se declara todo como formulario. El Tag <user.htm> no pertenece a la sintaxis estándar HTML y sirve al Web-I/O para identificar la página web. El Tag se filtra antes de cargar en el Browser.

```

<user.htm>
<html>
<head>
<title>Web-I/O mit Java-Apлетt</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<STYLE type=text/css>
  TD {COLOR: #000000; FONT-FAMILY:Verdana,Arial,Helvetica; FONT-SIZE: 10pt; }
</STYLE>
</head>
<body bgcolor="#FFFFFF">
<form name="ioform">
  <table width="500" border="1" height="144" bgcolor="#CCCCCC">
    <tr>
      <td>
        <table width="500">
          <tr>
            <td colspan="5">
              <b>Input/Output Control </b>
            </td>
          </tr>
          <tr>
            <td width="100">
              <input type="checkbox" name="cb_output" onclick="setOutput(0)"> Output 0
            </td>
            <td width="100">
              <input type="checkbox" name="cb_input"> Input 0
            </td>
            <td width="100">
              <div align="right">Counter 0 </div>
            </td>
            <td width="*" bgcolor="#FFFFFF" id="counter0">
            </td>
            <td width="55">
              <input type="button" value="Clear" onclick="clearCounter(0)">
            </td>
          </tr>
          <tr>
            <td width="100">
              <input type="checkbox" name="cb_output" onclick="setOutput(1)"> Output 1
            </td>
            <td width="100">
              <input type="checkbox" name="cb_input"> Input 1
            </td>
            <td width="100">
              <div align="right">Counter 1 </div>
            </td>
            <td width="*" bgcolor="#FFFFFF" id="counter1">

```



```
function setPassword()
{
  document.applets["dio"].setPassword( ioform.ed_password.value);
  ioform.ed_password.value = '';
}
```

- **Poner Outputs**

El usuario puede poner los Outputs a través de dos casillas de verificación `cb_output`. Al llamar la función se entrega el n° del Output.

```
<script language="JavaScript" type="text/javascript">
<!--
function setOutput(OutputNr)
{
  var Out = 0;
  Out |= Math.pow( 2, OutputNr );
  if ( ioform.cb_output[OutputNr].checked==true)
  {
    document.applets["dio"].outputAccess( Out,Out );
  }
  else
  {
    document.applets["dio"].outputAccess( Out,0 );
  }
}
```

- **Borrar contadores**

Se pueden borrar los estados de contador del contador Input. Como parámetro se entrega el n° del contador que se debe leer o borrar. Si no se entrega ningún parámetro, el Web-IO borra todos los contadores.

```
function clearCounter(CounterNr)
{
  if (CounterNr==undefined)
  {
    document.applets["dio"].counterClear(0xf);
  }
  else
  {
    var Counters = 0;
    Counters |= Math.pow( 2, CounterNr );
    document.applets["dio"].counterClear( Counters);
  }
}
```

#### 4. Actualizar los estados IO y los estados de contador

El Java-Applet asume la supervisión de los Inputs, Outputs y contadores del Web-IO. Si se reconoce un cambio, el Applet llama una función JavaScript correspondiente dentro de la página web.

- **Cambio en los Outputs**

Si se reconoce un cambio, se llama la función siguiente. Se entrega de qué Web-IO se anunció el cambio, qué Output se ha cambiado y en qué estado se encuentra actualmente el Output.

```
function outputChanged( Device, OutputNr, OutputVal )
{
  ioform.cb_output[OutputNr].checked=OutputVal;
}
```

- **Cambio en los Inputs**

Si se reconoce un cambio, se llama la función siguiente. Se entrega de qué Web-IO se anunció el cambio, qué Input se ha cambiado y en qué estado se encuentra actualmente el Input.

```
function inputChanged( Device, InputNr, InputVal)
{
  ioform.cb_input[InputNr].checked=InputVal;
}
```

- **Cambio en los Contadores**

Si se reconoce un cambio, se llama la función siguiente. Se entrega de qué Web-IO se anunció el cambio, qué contador se ha cambiado y en qué estado se encuentra actualmente el contador.

```
function counterChanged( Device, CounterNr, CounterVal )
{
  document.getElementById('counter'+CounterNr).innerHTML = '<a>'+CounterVal+'</a>';
}
//-->
</script>
</body>
</html>
```

Para poder trabajar con el ejemplo mostrado, el Browser tiene que permitir trabajar con Java-Applets. Si no se tiene el apoyo Java, se pueden descargar gratis los Plugins necesarios en [www.java.com](http://www.java.com).

El ejemplo asiste todas las funciones corrientes del Web-IO optimado para el [Web-IO 2x Digital Input, 2x Digital Output PoE](#). Para los otros modelos Web-IO tienen que realizarse en caso necesario adaptaciones en la [página web ejemplo](#). Otros ejemplos de programa para la programación del zócalo los encontrarán en las [páginas de herramientas](#) al Web-IO. Una descripción detallada sobre los comandos de los modelos Web-IO digitales la encontrará en el [manual de referencia](#).

[↓ Descargar el programa ejemplo](#)

**¿No tiene todavía un Web-IO y quiere probar el ejemplo presentado?**

No hay problema: Le ponemos a disposición el Web-IO Digital 2xInput, 2xOutput gratis durante 30 días. Rellene sencillamente un pedido muestra y le enviaremos el Web-IO para probar a cuenta abierta. Si nos devuelve el aparato dentro de los 30 días, le abonamos la factura completa.

[Al pedido muestra](#) 



**Le atendemos personalmente:**

Wiesemann & Theis  
GmbH  
Porschestra. 12  
42279 Wuppertal  
Tel: +49 202/2680-110 (lu-vi de 8-17  
horas)  
Fax: +49-202/2680-265  
[info@wut.de](mailto:info@wut.de)

© Wiesemann & Theis GmbH, salvo errores y modificaciones: como podemos cometer errores, no se debe utilizar nuestros enunciados sin verificarlos. Por favor, notifíquenos todas las erratas y malentendidos que detecte, para que podamos localizarlo y solucionarlo lo antes posible.

[Protección de datos](#)