

Aplicación al Web-I/O digital:

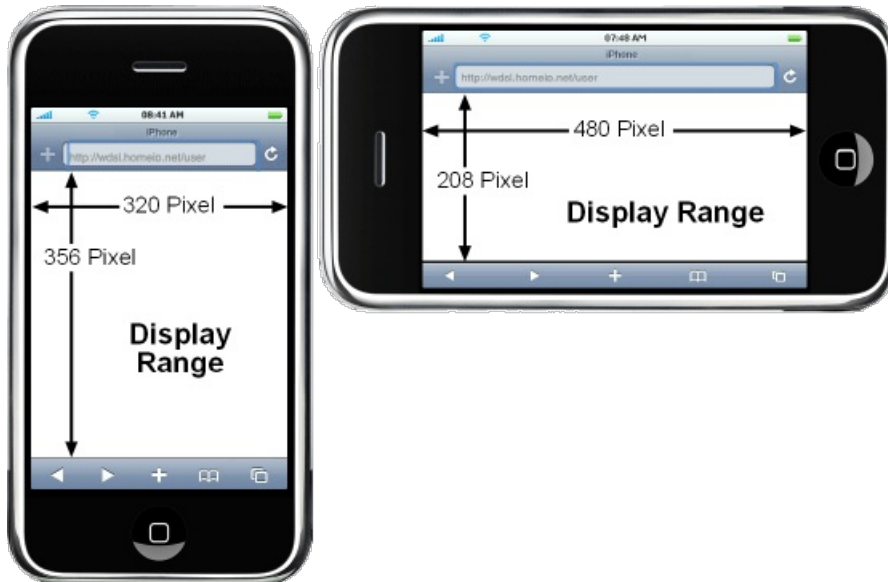
Páginas Web optimizadas para iPhone para aplicaciones Web-I/O

Resumen de productos

Sinopsis de aplicaciones

Es cierto que las técnicas mostradas aquí hacen referencia al iPhone de Apple, pero también pueden ser aplicadas a otros móviles aptos para Internet con sólo adaptar el tamaño de la visualización a la resolución de la pantalla. El único requisito necesario es que soporte JavaScript y, sobre todo, los objetos HTTP-Request (técnica AJAX).

En relación con la imagen del iPhone hay dos variantes:

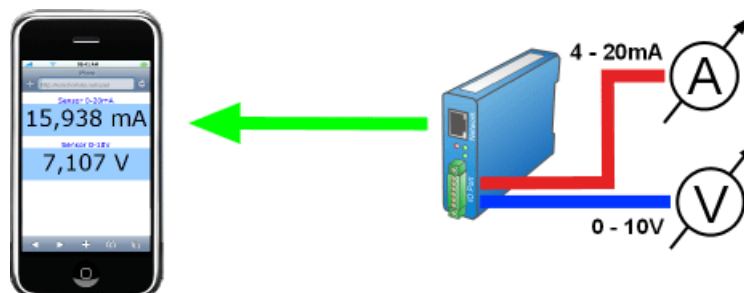


Para no complicar la construcción de la página Web se recomienda decidirse por una de las variantes y construir la página Web en consonancia con ésta.

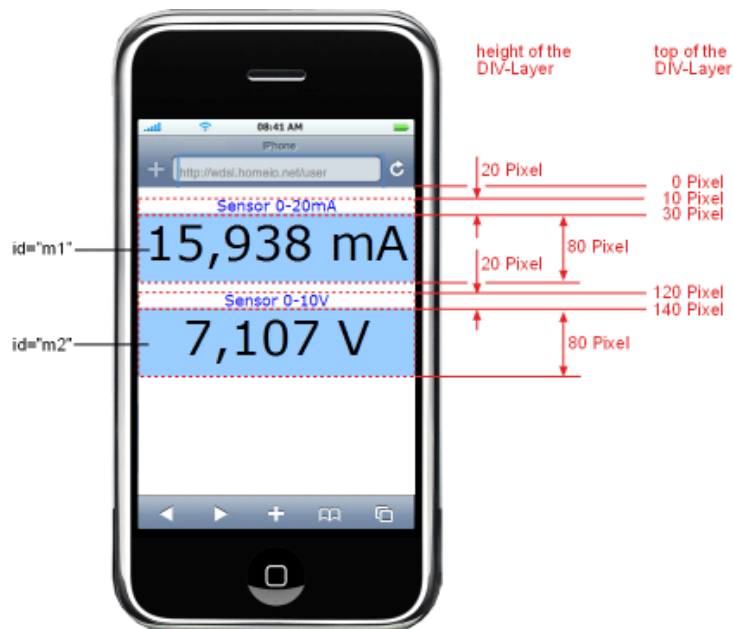
Si se dispone de acceso WLAN, el ejemplo mostrado funciona también en todos los modelos iPod Touch.

Ejemplo - ver los datos de las mediciones en técnica industrial y de edificios con iPhone

En este ejemplo se pretende medir corrientes o tensiones en un Web-I/O 2xAnalog In y luego visualizar los resultados como página Web dinámica en el iPhone. Intercalando el correspondiente convertidor de medición se puede medir prácticamente todas las magnitudes físicas y visualizarlas dinámicamente en el iPhone o bien en el teléfono móvil.



Como ya se ha dado a entender, en las páginas Web especialmente optimizadas para iPhone es importante definir con los pixels exactos el tamaño y posición de los elementos HTML utilizados.



El dibujo pretende ayudar a asignar los elementos y a comprender el siguiente texto fuente.

```

<user.htm>
<html>
<head>
<title>Aktuelles Klima</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta name="viewport" content="width=320" />
<style type="text/css">
<!--
* { font-family:Verdana, Helvetica; }
.description{ position:absolute;
text-align:center;
font-size:18px;
left:0px;
height:20px;
width:320px;
}
.value { position:absolute;
background-color:#99CCFF;
text-align:center;
font-size:55px;
left:0px;
height:80px;
width:320px;
}
-->
</style>
</head>

```

El texto fuente comienza con el tag específico del Web-IO.<user.htm>, que el Web-IO necesita para la asignación interna en el sistema de archivos. Web-IO no transmite al navegador el tag <user.htm>.

A continuación le sigue el área <Head> normal de la página Web. Es importante el dato<viewport>. Con ello se limita a 320 pixel el zoom de la página Web, practicado normalmente por iPhone, de modo que siempre se muestre la página Web completa y en la escala deseada.

En la continuación del texto fuente se colocan ahora los elementos DIV para la descripción y visualización de los valores. Para esos elementos se encuentra la información estilos CSS correspondiente en la cabecera de la página. Entre otras la definición de la altura y la posición izquierda.

```

<body>
<div class="description" style="top:10px; ">
<w&t_tags=sensor1>
</div>
<div id="m1" class="value" style="top:30px;">
<w&t_tags=m1>
</div>
<div class="description" style="top:120px; ">
<w&t_tags=sensor2>
</div>
<div id="m2" class="value" style="top:140px;">
<w&t_tags=m2>
</div>
</body>
</html>

```

A continuación le sigue el área <body> de la página Web. Aquí se utilizan las<clases CSS> así como los<datos top> para poner en posición cada uno de los elementos DIV. Los elementos DIV, en los que más tarde deba visualizarse los valores actualizados, tienen asignado un ID. Como texto de visualización se anotan<tags de W&T> especiales. Web-IO sustituye esos tags por valores actuales válidos al cargar la página en el navegador.

```

<script language="JavaScript" type="text/javascript">
function DataRequest(sendstring)
{
var xmlHttp = window.ActiveXObject ? new ActiveXObject("Microsoft.XMLHTTP") : xmlHttp = new XMLHttpRequest()
if (xmlHttp)
{
xmlHttp.onreadystatechange = function()
{
{
if (xmlHttp.readyState == 4)
{
if (xmlHttp.status == 200)
{
var ReceiveData = xmlHttp.responseText.split(",");
document.getElementById('m1').innerHTML = ReceiveData[ReceiveData.length-2];
document.getElementById('m2').innerHTML =ReceiveData[ReceiveData.length-1];
xmlHttp=null;
}
}
}
xmlHttp.open("GET", sendstring, true);
xmlHttp.setRequestHeader("Connection", "close");
xmlHttp.setRequestHeader("If-Modified-Since", "Thu, 1 Jan 1970 00:00:00 GMT");
xmlHttp.send(null);
maintimer = setTimeout("DataRequest('single')", 5000);
}
}
DataRequest('single');
</script>

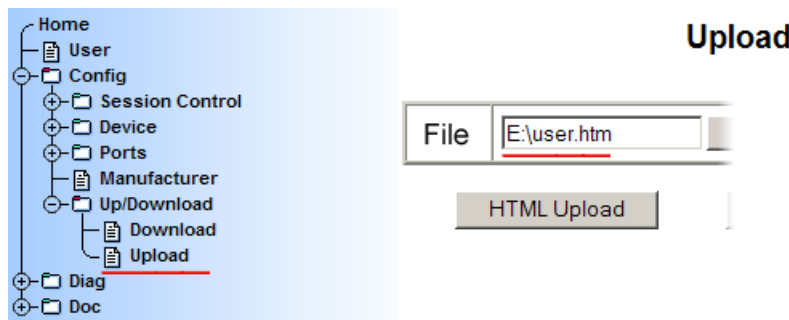
```

La parte principal de la página Web es la de JavaScript, incorporada en la <head> de la página Web. Con la función DataRequest se envía aquí el comando *single* al Web-IO. Web-IO envía de vuelta los valores de temperatura, humedad y presión del aire separados con punto y coma. A través del Split-Statement se separan los valores y se escriben en un array de variables.

Con este método `document.getElementById()` se seleccionan los elementos DIV respectivos para la visualización y se anotan los valores actuales.

Este proceso tiene lugar cíclicamente cada 5000 ms a través de un temporizador.

Con ello se concluye la página Web y ahora sólo falta cargarla en el Web-IO.



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

no hay problema: Le ponemos a disposición el Web-IO Analógico 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](#)

[↓ Descargar el programa ejemplo](#)



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

© 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](#)