

ACTIVIDAD N°6:

COMUNICACIÓN CON MICROCONTROLADOR



1. DESCRIPCION

• OBJETIVOS

1. Realizar el conexionado de los distintos periféricos EZweb
2. Programar el microservidor EZweb.
3. Programar microcontroladores PIC para comunicarse con el EZweb

• METODOLOGIA

En esta actividad conectaremos el microservidor EZweb con un microcontrolador PIC mediante comunicación serie.

En este apartado programaremos dos elementos:

- El microservidor EZweb
- El Microcontrolador PIC

• CONOCIMIENTOS PREVIOS

Para poder seguir con garantías esta unidad didáctica:

Se requiere:

- ♦ Conocimientos básicos de programación de páginas web.
- ♦ Conocimientos básicos de programación de microcontroladores PIC (Programación en lenguaje C, utilizando el compilador PCW de la casa CCS)

Se recomienda:

- ♦ Conocimientos básicos sobre electrónica.

• MATERIAL Y EQUIPAMIENTO NECESARIO

1. Microservidor EZweb distribuido por MSE (Microsystems Engineering de Bilbao)
2. Convendría tener el manual del microservidor Ezweb.
3. Para poder grabar los programas y realizar distintas pruebas utilizaremos el entrenador PIC-SCHOOL distribuido por MSE (Microsystems Engineering de Bilbao).

• MONTAJE DEL PROTOTIPO

En este apartado realizaremos una descripción del patillaje del Microservidor Ezweb así como los comandos a utilizar con las salidas digitales y analógicas.

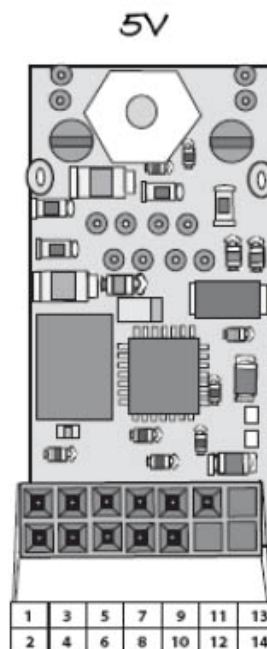
• DESCRIPCION DEL MICROSERVIDOR EZWEB

• PATILLAJE DEL EZWEB

Para realizar las siguientes actividades utilizaremos el microservidor Ezweb de 5V cuyo patillaje es el que a continuación se muestra.

Características técnicas

Pin Number	Description
1	Analog Input 1/Digital I/O
2	Analog Input 2/Digital I/O
3	Analog Input 3/Digital I/O
4	Analog Input 4/Digital I/O
5	Analog Input 5/Digital I/O
6	Digital I/O
7	Digital I/O
8	TX - Serial Data Transmit (from EZ Web Lynx to PC or other serial device)/Digital I/O
9	RCV - Serial Data Receive (from PC to EZ Web Lynx)/Digital I/O
10	Dallas DS1621 or DS1631 I ² C Temp Sensor SCL (clock) line/Digital I/O
11	Dallas DS1621 or DS1631 I ² C Temp Sensor SDA (data) line/Digital I/O
12	RST (Active-low Reset)
13	Vdd (5V)
14	GND



Tal como

podemos ver, disponemos de 2 patillas (la pata 8 y la 9) que aparte de trabajar como entradas/salidas digitales también son parte del puerto serie que tiene el microservidor Ezweb.

En esta actividad estas patas las utilizaremos como puerto serie (La pata 8 es la pata de transmisión y la pata 9 será la pata de recepción).

En los ejemplos de esta actividad utilizaremos el microcontrolador PIC16F876A en el que utilizaremos el pin del puerto C6 como pin de transmisión y el pin del puerto C7 como pin de recepción.

• COMANDOS E INSTRUCCIONES A UTILIZAR

• INSTRUCCIONES O ETIQUETAS DINAMICAS

Las instrucciones que utilizaremos con el EZweb son los que a continuación mostramos:

Lista de instrucciones		
<!--(PIN x)-->	x es un número de pin.	El elemento es sustituido con un 0 si el pin está bajo y 1 si está alto. El pin debe estar en modo entrada.
<!--(PIN x ltag/htag)-->	x es un número de pin. ltag es el texto si el pin está bajo. htag es el texto si el pin está alto.	El elemento es sustituido con ltag si el pin está bajo y con htag si está alto. El pin debe estar en modo entrada.
<!--(PIN x ANALOG)-->	x es un número de pin.	El elemento es sustituido con un número en formato x.xx que representa el voltaje en el pin. El pin debe estar en modo entrada analógica.
<!--(PIN x ANALOG*a/b+cd,p)-->	x es un número de pin a es un número a multiplicar por el voltaje analógico (optional). b es un número a dividir por el voltaje analógico (optional). c es un número a sumar con el voltaje analógico (optional). d es un número a restar con el voltaje analógico (optional). p es el número de decimales.	El elemento es sustituido con un número que representa el resultado del cálculo. El pin debe estar en modo entrada analógica. Los operadores matemáticos (*/+/-) deben ser usados en este orden.
<!--(SERIAL DATA x y)-->	x es el index inicial de la última trama de datos serie capturada. y es el index final de la última trama de datos serie capturada.	El elemento es sustituido por la última trama de datos serie, desde el carácter index x hasta el carácter index y. Para leer toda la trama serie completa se utiliza la extensión READ HTML.
<!--(IF PIN x = n)-->	x es un número de pin n es 0 ó 1	Si la condición es falsa, se ignora el código hasta ELSE o ENDIF.
<!--(IF PIN x < a)-->	x es un número de pin	Si la condición es falsa, se ignora el código hasta ELSE o ENDIF.
<!--(IF PIN x > a)-->	a es un número en el rango (0.00-5.00) (5 V version)	Si la condición es falsa, se ignora el código hasta ELSE o ENDIF.
<!--(IF KEY = n)--> <!--(IF KEY != n)--> <!--(IF KEY < n)--> <!--(IF KEY <= n)--> <!--(IF KEY > n)--> <!--(IF KEY >= n)-->	KEY is un comando válido. n puede ser un entero o un decimal, dependiendo del comando utilizado.	Si el argumento del comando no cumple la condición, se ignora el código hasta ELSE o ENDIF.
<!--(ELSE)-->		Usado para la directiva IF
<!--(ENDIF)-->		Usado para la directiva IF/ELSE
<!--(EXEC KEY=VALUE)-->		Ejecuta un comando.
<!--(READ KEY)-->		Lee un comando.

- **COMANDOS**

Los comandos a utilizar para trabajar con el puerto serie los podemos distribuir en dos grupos que a continuación describimos:

- **Comandos para que el ezweb reciba (la página web recibe)**

En este caso necesitamos utilizar el buffer del microservidor y para gestionar el buffer utilizaremos los siguientes comandos:

- **SERIAL_START:** este comando sirve para establecer el carácter del inicio del buffer. Normalmente utilizaremos la cadena "N:"
Para establecer "N:" como inicio de buffer deberíamos de ejecutar el comando de la siguiente manera: SERIAL_START=4E,3A donde 4E es el carácter "N" en hexadecimal y 3A es el carácter ":".
- **SERIAL_END:** este comando sirve para establecer el final del buffer. Normalmente utilizaremos el carácter ";".
Para establecer el carácter ";" deberíamos de ejecutar el comando de la siguiente manera: SERIAL_END=3B donde 3B es el carácter ";" en hexadecimal.
- **SERIAL_KBHIT:** este comando establece el nº de caracteres del buffer. puede funcionar como lectura o como escritura. Cuando leemos nos indicará el nº de caracteres que tiene el buffer. Cuando escribimos le indicaremos de cuantos caracteres está constituido el buffer. Un ejemplo de escritura es el de borrar el buffer y se ejecuta de la siguiente manera: SERIAL_KBHIT=0.
- **SERIAL_GET:** Cuando leo este comando me devuelve el valor del buffer

- **Comandos para que el ezweb envíe**

- **SERIAL_SEND:** envía por serie caracteres en hexadecimal separados por (,)

Si en la página pusieramos SERIAL_SEND=0x41,42,43 el microservidor enviaría ABC. Este comando enviará la información en hexadecimal
- **SERIAL_STRING:** Es un comando igual que el SERIAL_SEND pero en este caso envía una cadena de caracteres.

Si en la página pusieramos SERIAL_STRING=Hola el microservidor enviará la cadena de caracteres Hola.

• EJEMPLOS DE APLICACIÓN. SOFTWARE

En este apartado realizaremos distintos ejemplos de aplicación, utilizando los distintos elementos con los que cuenta el PicSchool. La programación de las páginas web las realizaremos utilizando el software Dreamweaver, donde insertaremos los distintos comandos que se utilizan con el microservidor EZweb. Los comandos que utilizamos en estos ejemplos están realizados con etiquetas dinámicas.

Estos ejemplos de aplicación constarán de una pequeña descripción de la aplicación, de un esquema de conexiones y del programa de la página web.

El programa de la página web es un fichero o varios ficheros con extensión htm que tendremos que empaquetar con el software csfs y lo cargaremos en el microservidor mediante ftp.

• APLICACIÓN 1 (ACTIVACION DE MOTOR MEDIANTE PUERTO SERIE)

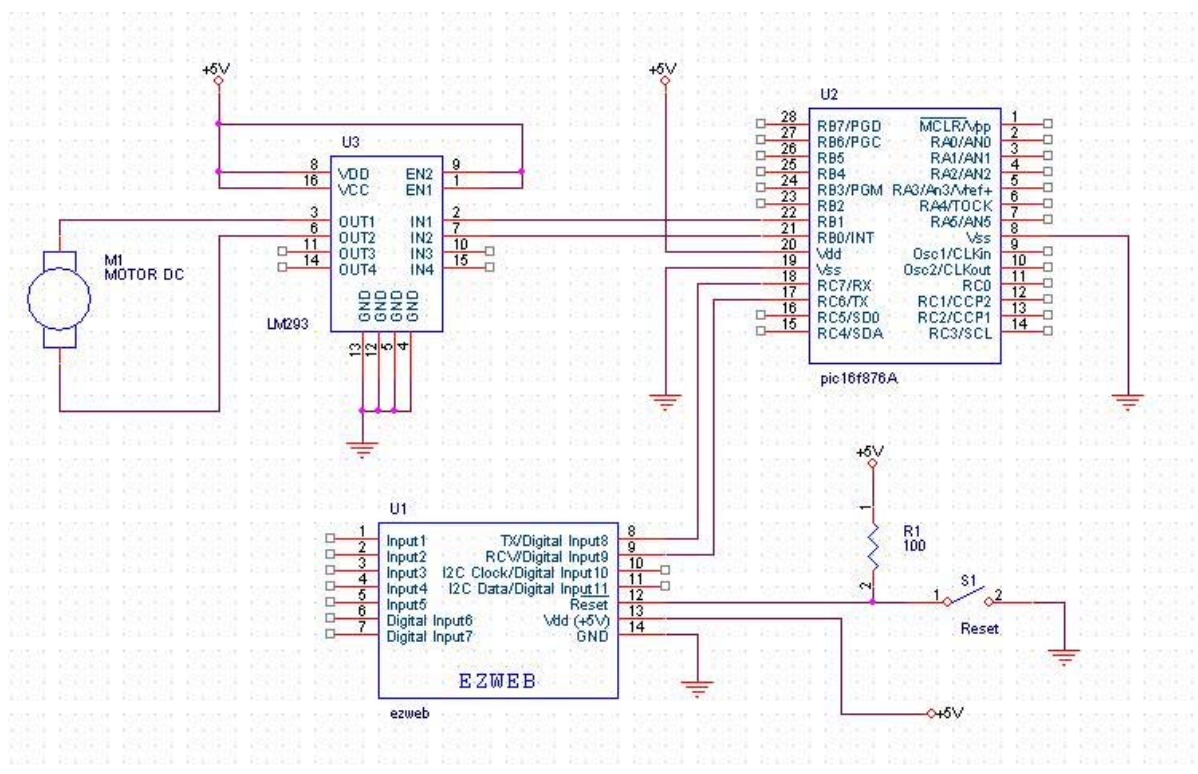
• DESCRIPCION

La primera aplicación que realizaremos tiene como objetivo activar un motor y que gire en los dos sentidos. Para ello en la página web tendremos tres botones:

- **Boton para parar el motor:** Este boton enviara por el puerto serie el carácter a. Cuando el microcontrolador recibe el carácter a, enviara por el puerto B en el que esta conectado el motor el valor 0x00. Este valor para el motor.
- **Boton para girar el motor a derechas:** Este boton enviara por el puerto serie el carácter b. Cuando el microcontrolador recibe el carácter b, enviara por el puerto B en el que esta conectado el motor el valor 0x02. Este valor hara girar el motor a derechas.
- **Boton para girar el motor a izquierdas:** Este boton enviara por el puerto serie el carácter c. Cuando el microcontrolador recibe el carácter c, enviara por el puerto B en el que esta conectado el motor el valor 0x01. Este valor hara girar el motor a izquierdas.

- ESQUEMA DE CONEXIONES**

En la siguiente imagen mostraremos el esquema de conexiones para hacer funcionar un motor mediante el puerto serie del microservidor de páginas web.



- PROGRAMA PAGINA WEB**

Esta aplicación consta de 1 página en la que visualizamos tres botones. Cuando en la página insertamos un botón hay que insertar un formulario.

Cuando insertamos un formulario el programa de la página web inserta los siguientes comandos htm:

```
<form id="form1" name="form1" method="get" action="">
  <div align="center">
    <input name="SERIAL_STRING" type="hidden" value="a" />
    <input type="submit" value="PARAR MOTOR" />
  </div>
</form>
```

En los formularios destacaremos el apartado **input name** donde introduciremos el comando que queremos enviar, en este caso SERIAL_STRING.

En el apartado **value** pondremos la cadena que hay que asignar al comando SERIAL_STRING. En este caso el carácter a.

En el apartado **input type** hay que indicar que es de tipo submit que indica que es un botón y el **value** sera lo que queremos que aparezca en el botón.

El programa de la página index.htm es el que a continuación se muestra.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>activar motor</title>
<style type="text/css">
<!--
.Estilo1 {
    font-size: 24px;
    font-weight: bold;
}
-->
</style>
</head>

<body>
<p class="Estilo1">ACTIVACION DE MOTORES</p> //Texto en la página
<table width="91%" border="1">
<tr>
<th width="35%" scope="col">
<form id="form1" name="form1" method="get" action="">
<div align="center"> //formulario para el botón de parar
<input name="SERIAL_STRING" type="hidden" value="a" />
// manda caracter "a" por el puerto serie con el comado SERIAL_STRING
<input type="submit" value="PARAR MOTOR" />
</div>
</form> </th>
<th width="35%" scope="col">
```



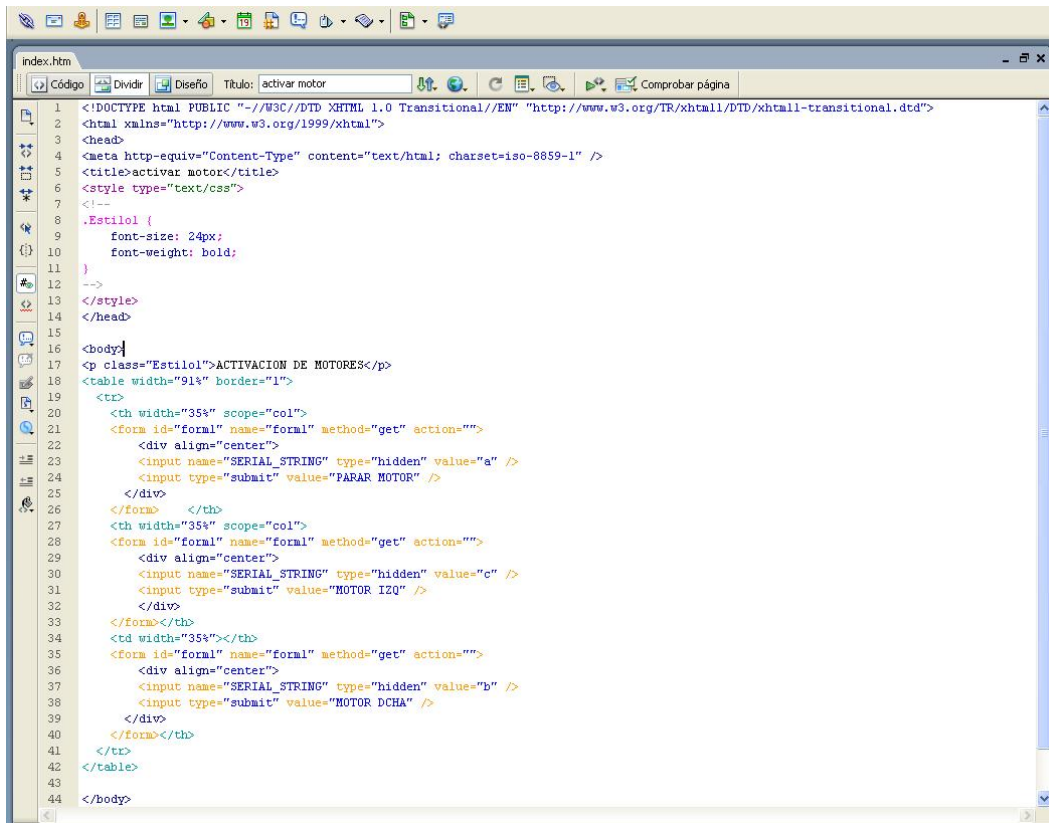
```

<form id="form1" name="form1" method="get" action="">
  <div align="center"> //formulario para el botón de izquierdas
    <input name="SERIAL_STRING" type="hidden" value="c" />
// manda caracter "c" por el puerto serie con el comando SERIAL_STRING
    <input type="submit" value="MOTOR IZQ" />
  </div>
</form></th>
<td width="35%"></th>
<form id="form1" name="form1" method="get" action="">
  <div align="center"> //formulario para el botón de derechas
    <input name="SERIAL_STRING" type="hidden" value="b" />
// manda caracter "b" por el puerto serie con el comando SERIAL_STRING
    <input type="submit" value="MOTOR DCHA" />
  </div>
</form></th>
</tr>
</table>

</body>
</html>

```

En la siguiente imagen muestra la página escrita con dremweaber:



```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
5 <title>activar motor</title>
6 <style type="text/css">
7 <!--
8 .Estilol {
9     font-size: 24px;
10    font-weight: bold;
11 }
12 -->
13 </style>
14 </head>
15
16 <body>
17 <p class="Estilol">ACTIVACION DE MOTORES</p>
18 <table width="91%" border="1">
19 <tr>
20 <th width="35%" scope="col">
21 <form id="formal" name="formal" method="get" action="">
22 <div align="center">
23 <input name="SERIAL_STRING" type="hidden" value="a" />
24 <input type="submit" value="PARAR MOTOR" />
25 </div>
26 </form> </th>
27 <th width="35%" scope="col">
28 <form id="formal" name="formal" method="get" action="">
29 <div align="center">
30 <input name="SERIAL_STRING" type="hidden" value="c" />
31 <input type="submit" value="MOTOR IZQ" />
32 </div>
33 </form></th>
34 <td width="35%"></td>
35 <form id="formal" name="formal" method="get" action="">
36 <div align="center">
37 <input name="SERIAL_STRING" type="hidden" value="b" />
38 <input type="submit" value="MOTOR DCHA" />
39 </div>
40 </form></td>
41 </tr>
42 </table>
43
44 </body>

```

La siguiente imagen muestra la página tal como se ve en Internet:



- **PROGRAMA DEL MICROCONTROLADOR**

El programa del microcontrolador consiste en recibir un dato del puerto serie y dependiendo del valor que sea haremos girar al motor en una dirección, en dirección contraria o lo pararemos.

Si el microcontrolador recibe el carácter "b" enviaremos al puerto B el valor 0x01 que hará girar el motor a izquierdas.

Si el microcontrolador recibe el carácter "c" enviaremos al puerto B el valor 0x02 que hará girar el motor a derechas.

Y si el microcontrolador recibe el carácter "a" enviaremos al puerto B el valor 0x00 que parara el motor.


```

case 'b':
    puertob=izq; //mando al puerto B el valor 0x01 que
    break;      //girara el motor a izquierdas
case 'c':
    puertob=dcha; //mando al puerto B el valor 0x02 que
    break;      //girara el motor a derechas
    }
}
}

```

- **APLICACIÓN 2 (ENVIO Y RECEPCION DE MENSAJES)**
 - **DESCRIPCION**

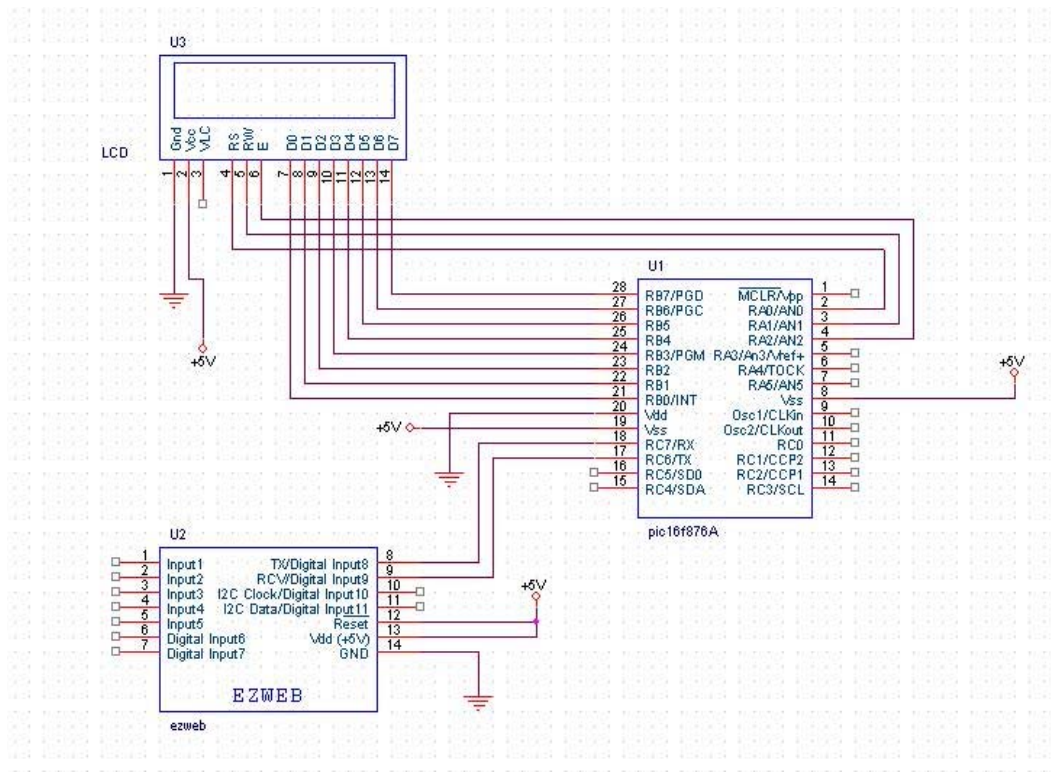
En esta 2ª aplicación trabajaremos el envío y recepción de mensajes por el puerto serie.

La aplicación consiste en el microservidor conectado a un microcontrolador PIC16F876A que a su vez está conectado a un LCD de 2x16 líneas.

En el LCD nos aparece un mensaje indicándonos que se está a la espera de un mensaje. Este mensaje se introduce desde la página web y deberá terminar con el carácter ";" ya que el programa del microcontrolador tiene indicado como final de mensaje el carácter ";". Este mensaje es enviado por el microservidor y será visualizado en el LCD. Si el mensaje se escribe correctamente el microcontrolador enviará a la página web el mensaje de "Mensaje escrito correctamente". El LCD también nos mostrará el mensaje "Espero Mensaje".

ESQUEMA DE CONEXIONES

En este caso tendremos el siguiente esquema de conexiones.



Tal como podemos comprobar que el microcontrolador está conectado al microservidor ezweb mediante el puerto serie. El pin de transmisión del microcontrolador se conecta al pin de recepción del microservidor y el pin de recepción del microcontrolador se conecta al pin de transmisión del microservidor. Los mensajes entre el microcontrolador y el microservidor ezweb se envían y reciben a través del puerto serie.

El microcontrolador también está conectado al LCD mediante el puerto B (por donde van los datos a visualizar del LCD) y tres pines del puerto A (con los que se controla el LCD).

Los mensajes que envía el microservidor al microcontrolador serán enviados por el microcontrolador al LCD.

Una vez escrito el mensaje en el LCD el microcontrolador enviará por el puerto serie el mensaje "Mensaje escrito correctamente".

• PROGRAMA

En el programa de la aplicación, al principio en la página web, le indicaremos al microservidor que caracteres forman el inicio del buffer y que carácter tendrá como final de buffer.

```
<!--(EXEC SERIAL_START=0X4E,0X3A)--> //El buffer comienza con los
                                     // caracteres "N",";"
<!--(EXEC SERIAL_END=0X3B)-->      // El buffer finaliza con el carácter ";"
```

Utilizaremos un formulario en el que introduciremos el mensaje que queremos enviar. Este mensaje deberá terminar con el carácter ";" ya que el programa del microcontrolador sabe que el mensaje terminará con el carácter ";".

```
<form method="get">
  <input type="text" name="serial_string" value="" maxlength=16>
  //formulario para introducir el texto a enviar por el puerto serie
  <input type="submit" value="Enviar texto al LCD">
  //boton para enviar el texto escrito anteriormente.
</form>
```

Cuando el microcontrolador escriba el mensaje en el LCD también nos enviará por el puerto serie al buffer del ezweb la siguiente cadena de caracteres:

N:Mensaje escrito correctamente;

En nuestra página web, al final mostraremos parte del mensaje que está en el buffer. En este caso, los caracteres de 2 al 30 con lo que veremos el siguiente mensaje:

Mensaje escrito correctamente

```
<!--(SERIAL DATA 2 30)-->

//Nos muestra desde el carácter 2 al carácter 30 del buffer del ezweb.
```

El programa de la página web nos quedará de la siguiente manera;

```
<html>
<head>
  <title>Comunicacion serie</title>
</head>
<!--(EXEC SERIAL_START=0X4E,0X3A)-->      //Inicio de buffer
<!--(EXEC SERIAL_END=0X3B)-->            // Fina del buffer
```



```

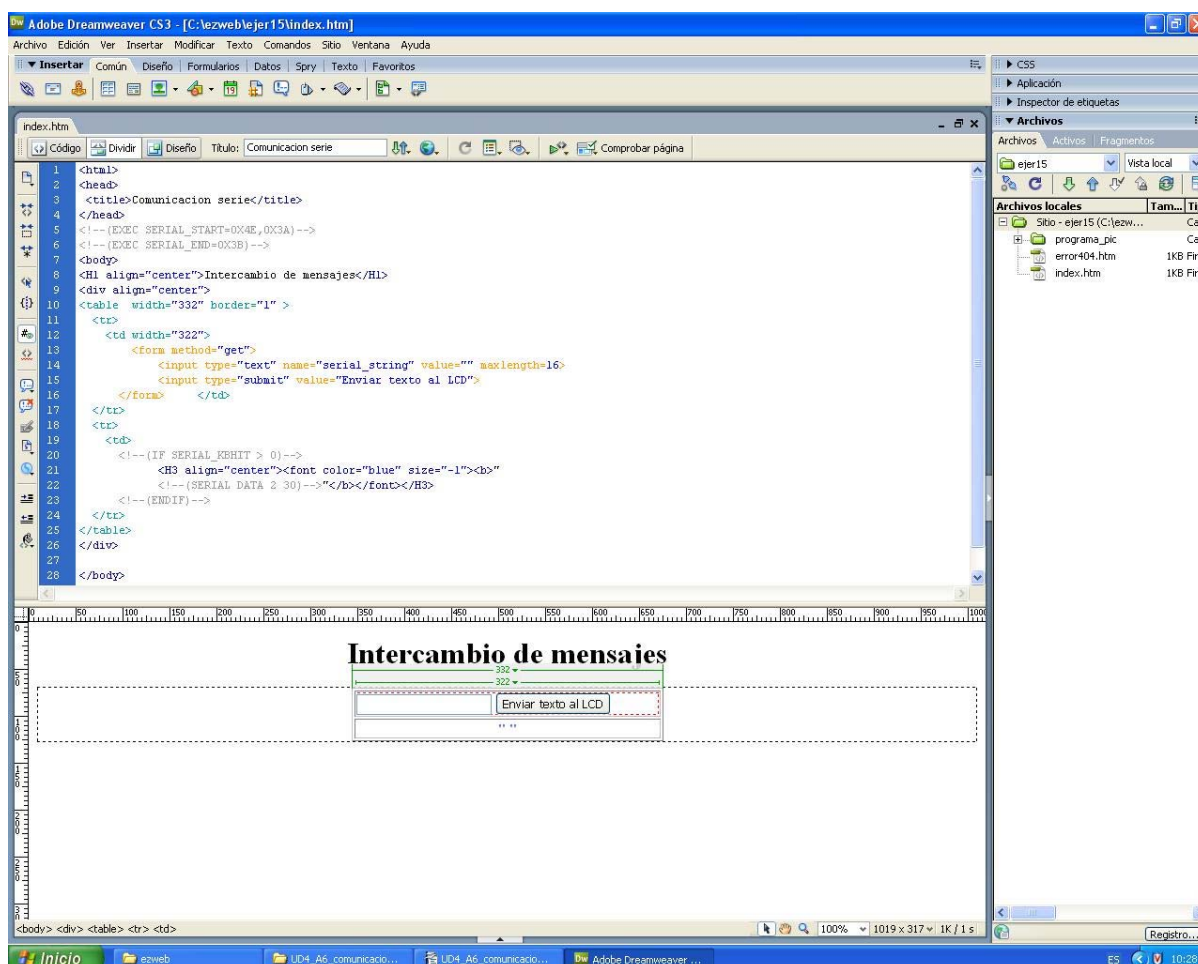
<body>
<H1 align="center">Intercambio de mensajes</H1>
<div align="center">
<table width="332" border="1" >
<tr>
<td width="322">
<form method="get">
<input type="text" name="serial_string" value="" maxlength=16>
<input type="submit" value="Enviar texto al LCD">
</form>    </td>
        //Formulario para enviar mensaje al microcontrolador
</tr>
<tr>
<td>
<!--(IF SERIAL_KBHIT > 0)--> //Si el buffer tiene caracteres
        <H3 align="center"><font color="blue" size="-1"><b>"
        <!--(SERIAL DATA 2 30)-->"</b></font></H3>
        // Mostrar desde el carácter 2 al 30 del buffer

        <!--(ENDIF)-->
</td>
</tr>
</table>
</div>

</body>

```

En la siguiente imagen podremos ver el programa escrito con dreamweaver:





En la siguiente imagen vemos la página web para introducir el mensaje a visualizar en el LCD:

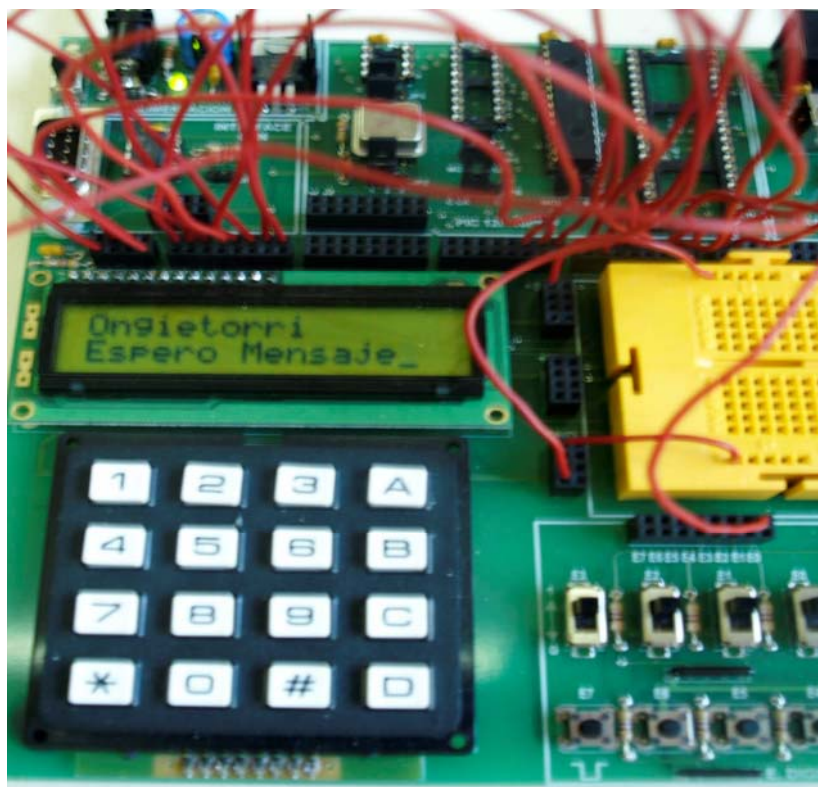


Intercambio de mensajes

Ongietorri:

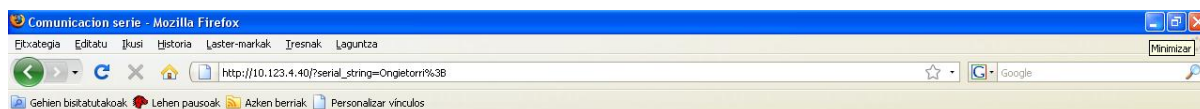


En la siguiente imagen vemos el mensaje en el LCD



Una vez escrito el mensaje en el LCD el microcontrolador enviara el mensaje de “Mensaje escrito correctamente” a la página web.

En la siguiente imagen podemos ver la página web despues de que se ha escrito el mensaje en el LCD:



Intercambio de mensajes

<input type="text"/>	Enviar texto al LCD
" Mensaje escrito correctamente "	



• PROGRAMA DEL MICROCONTROLADOR

El programa del microcontrolador consiste en escribir el mensaje de "Espero mensaje en el LCD y espera a recibir una cadena de caracteres enviada por el puerto serie por el Ezweb.

Una vez que se recibe el mensaje por el puerto serie, el microcontrolador lo visualiza en el LCD y envía al ezweb el mensaje de "Mensaje escrito correctamente". También visualiza en la fila 2 del LCD el mensaje "Espero Mensaje".

```

/*****

```

Protocolo entre el PIC y el ezweb

```

*****/

```

```

// TxD ==> RC6 con RCV del ez web (patilla 9)

```

```

// RxD ==> RC7 con TX del ez web (patilla 8)

```

```

*****/

```

PROYECTO control desde web

```

*****/

```

```

#include <16F876A.h>

```

```

#include <16F876A.h>

```

```

#include "lcd.h"

```

//libreria del LCD

```

#include "lcd.h"

```

//RC6 transmisor (TxD)

//Rc7 receptor (RxD)

```

#include "lcd.h"

```

```

void main(void)

```

```

{

```

```

    char cadena[17];

```

```

    char letra;

```

```

    int8 i;

```

```

    setup_adc_ports (NO_ANALOGS);

```

```

    set_tris_a(0x00); //pongo como salidas RA0-RA2

```

```

    set_tris_b(0x00); //pongo como salidas RB0-RB7

```

```

    set_tris_c(0b10000000); //pongo como entrada RC7

```

```

    delay_ms(2000);

```

```

    LCD_INI(); //inicializo el LCD

```

```

    delay_ms(2000); // Esperamos que el ezweb se inicialice

```

```

printf(LCD_DATO,"Espero Mensaje"); //Visualizo mensaje "Espero Mensaje"

for(i=0;i<17;i++)
{
    letra=getch();
    // leo mensaje desde el puerto serie y guardo carácter a carácter
    if(letra=='\0')
    {
        cadena[i]='\0'; //Caracter fin de cadena
        break;
    }
    cadena[i]=letra;
}
LCD_GOTOXY(1,1); //coloco en linea 1 fila 1 el cursor del LCD
printf(LCD_DATO,""); // Borro el mensaje anterior
LCD_GOTOXY(1,1); //coloco en linea 1 fila 1 el cursor del LCD
printf(LCD_DATO,"%s",cadena);

//Visualizo mensaje enviado por el puerto serie
printf("AT*SERIAL_GET=N:Mensaje escrito correctamente;\r\n");
//Envío por el puerto serie el mensaje "N:Mensaje escrito
//correctamente;"
while(1)
{
    LCD_GOTOXY(1,2); //coloco en linea 1 fila 2 el cursor del LCD
    printf(LCD_DATO,"Espero Mensaje");
    for(i=0;i<17;i++)
    {
        letra=getch();

        if(letra=='\0')
        {
            cadena[i]='\0'; //Carácter fin de cadena
            break;
        }
        cadena[i]=letra;
    }
    LCD_GOTOXY(1,1);
    printf(LCD_DATO,""); // Borro el mensaje anterior
    LCD_GOTOXY(1,1);

```



```
printf(LCD_DATO,"%s",cadena);  
//Visualizo mensaje enviado por el puerto serie  
printf("AT*SERIAL_GET=N:Mensaje escrito correctamente;\r\n");  
//Envío por el puerto serie el mensaje "N:Mensaje escrito //correctamente;"  
    }    //fin de while  
}
```

```
    }    //fin de main
```