

EXPERIMENTO 7

Display de Cristal Líquido

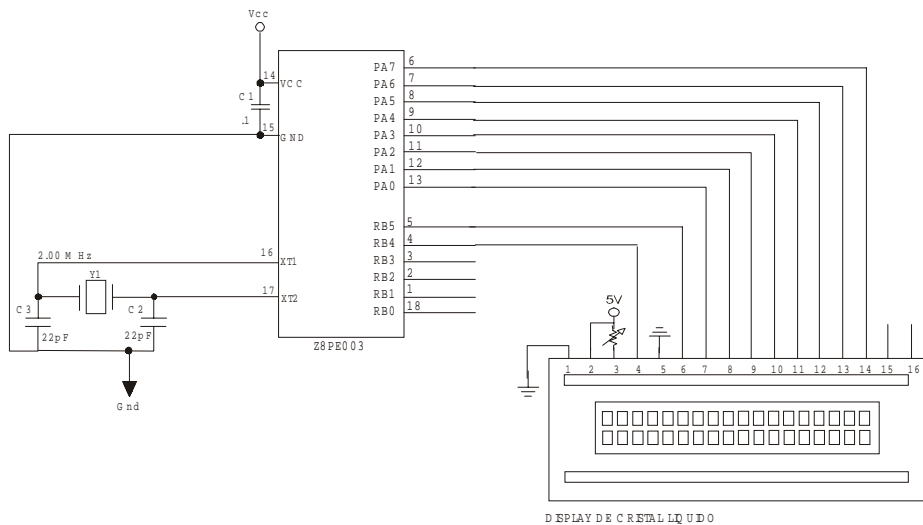
7.1 Propósito:

Aprender a desplegar texto programado a un Display de Cristal Líquido mediante la utilización del microcontrolador Z8^{PLUS}.

7.2 Material Utilizado:

- Circuito básico del Z8^{PLUS}
- Emulador Z8M001
- 1 Display de Cristal Líquido
- 1 potenciómetro de 5k Ω

7.3 Diagrama del circuito



7.4 Descripción del experimento:

En la [hoja de especificaciones](#) se describen los pasos a seguir para encender el display. Los pasos que a continuación se describen se encuentran en el orden en el que están en el programa:

En los pasos de 1 al 5 RW debe estar en estado bajo ya que se esta programando.

Paso 1: En este paso se especifica la longitud del dato en DL (DB4), se manda un uno para datos de 8 bits o un cero para datos de 4 bits, el número de renglones requeridos se especifica en N (DB3), se manda un uno para tener 2 renglones y un cero para un renglón. La fuente de los caracteres se especifica en F (DB2) un uno para caracteres de 5X10 puntos y un cero para caracteres de 5X8 puntos, además es necesario mandar ceros a DB7 y DB6, uno a DB5, y los bits DB1 y DB0 no importan. Para un dato de 8 bits, dos renglones y una fuente de 5X10 puntos mandamos un 38H por el puerto A, para que el microcontrolador del display cargue el dato, se debe mandar un pulso al enenable (E), y para que se cumpla el tiempo de ejecución en el display se ejecuta un tiempo de retardo (delay) antes de pasar al siguiente paso.

Paso 2: Limpiar el display y cargar el contador de direcciones con 00H. Para hacer esto se carga el puerto A con un 01H y se procede a mandar el pulso a enenable y a ejecutar el tiempo de retardo.

Paso 3: Se activa el display mandando un uno a D (DB2), activa el cursor con un 1 en C (DB1) y si se desea que el cursor parpadee se manda un uno a B (DB0). Además se debe mandar un 1 a DB4 y ceros de DB4 a DB7. Entonces, después de cargar el número 0CH en el puerto A se manda el pulso a enenable y se ejecuta el retardo.

Paso 4: Se especifica si el cursor va a hacer un corrimiento inmediatamente después de cada escritura de dato y hacia donde va a ser este corrimiento. Con un 1 en S/C (DB3) se activa el corrimiento del cursor, y con un 1 o un 0 en R/L se determina si el corrimiento se hace hacia la derecha o a la izquierda respectivamente. Se manda además un 1 a DB4 y ceros en DB5 y DB7. Ya que se manda esto por el puerto A, se manda el pulso del enenable y se ejecuta el retardo.

Paso 5: Se especifica la dirección en el display a partir de la cual se van a escribir los datos, para el primer renglón la dirección de izquierda a derecha es de la 80H a la 8FH y para el segundo renglón de C0H a CFH. Después de cargar esta dirección, se manda el pulso del enenable y se ejecuta el retardo.

Después de estos 5 pasos anteriores ya se puede empezar a escribir datos en el display, para escritura de datos se manda un 1 a R/W, se manda el código del carácter a DB0-DB7 (por puerto A) e igual que en los pasos anteriores se manda un pulso al enable para que el microcontrolador del display reconozca el dato y se ejecuta un tiempo de retardo. Este procedimiento es igual para todos los caracteres.

7.5 Programa:

```

TCTLHI    EQU    %0C1
PTBDIR    EQU    %0D6
PTBSFR    EQU    %0D7
PTBOUT    EQU    %0D5
PTBIN     EQU    %0D4
PTASFR    EQU    %0D3
PTADIR    EQU    %0D2
PTAOUT    EQU    %0D1

.org 020H

INICIO:
LD TCTLHI, #070H
LD RP, #00H
LD SPL, #40H
LD PTBSFR, #00H
LD PTADIR, #0FFH ;CONFIGURA PUERTO A COMO SALIDA.
LD PTBDIR, #03FH ;CONFIGURA EL PUERTO B COMO SALIDA.
LD PTBOUT, #00H

CALL DELAY2      ; TIEMPO AL MICRO DEL DISPLAY PARA QUE SE
                  REESTABLEZCA
LD PTAOUT, #038H ;PASO 1: Especifica la longitud del dato, el numero de
                  renglones y la fuente
LD PTBOUT, #20H ;SE MANDA UN PULSO AL ENEABLE
NOP
LD PTBOUT, #00H
CALL DELAY1
LD PTAOUT, #01H ;PASO 2: Limpia el display y carga el contador de la DD
                  RAM con 00
LD PTBOUT, #20H ;SE MANDA UN PULSO AL ENEABLE
NOP
LD PTBOUT, #00H
CALL DELAY2
LD PTAOUT, #0CH ;PASO 3: Especifica la activación del display, del cursor y su
                  parpadeo
LD PTBOUT, #20H ;SE MANDA UN PULSO AL ENEABLE

```

```
NOP
LD PTBOUT, #00H
CALL DELAY1
LD PTAOUT, #06H ;PASO4: Especifica la dirección del cursor al moverse
LD PTBOUT, #20H ;SE MANDA UN PULSO AL ENEABLE
NOP
LD PTBOUT, #00H
CALL DELAY1
LD PTAOUT, #87H ;PASO 5: Dirección a partir de la cual se desea escribir
LD PTBOUT, #20H ;SE MANDA UN PULSO AL ENEABLE
NOP
LD PTBOUT, #00H
CALL DELAY1
```

;ENVIAR CARACTERES

```
LD PTBOUT, #10H ;SE MANDA UN 1 A RW PARA RECONOCER UNA
OPERACION DE ESCRITURA
LD PTAOUT, #5AH ;CODIGO DEL CARACTER Z
LD PTBOUT, #30H ;SE MANDA UN PULSO AL ENEABLE
NOP
LD PTBOUT, #10H
CALL DELAY1
LD PTAOUT, #38H ;CODIGO DEL CARACTER 8
LD PTBOUT, #30H ;SE MANDA UN PULSO AL ENEABLE
NOP
LD PTBOUT, #10H
CALL DELAY1

LD PTBOUT, #00H ;CONFIGURO PARA CAMBIAR AL SEGUNDO RENGLON
LD PTAOUT, #0C6H ;RENGLON 2 COLUMNA 7
LD PTBOUT, #20H ;RW=0 Y MANDO PULSO A "E"
NOP
LD PTBOUT, #00H
CALL DELAY1

LD PTAOUT, #50H ;CODIGO DEL CARACTER P
LD PTBOUT, #30H ;RW=1 Y MANDO PULSO A "E"
NOP
LD PTBOUT, #10H
CALL DELAY1
LD PTAOUT, #6CH ;CODIGO DEL CARACTER L
LD PTBOUT, #30H ;SE MANDA UN PULSO AL ENEABLE
NOP
NOP
LD PTBOUT, #10H
CALL DELAY1
```

```
LD PTAOUT, #75H ;CODIGO DEL CARACTER U
LD PTBOUT, #30H ;SE MANDA UN PULSO AL ENEABLE
NOP
LD PTBOUT, #10H
CALL DELAY1
LD PTAOUT, #73H ;CODIGO DEL CARACTER S
LD PTBOUT, #30H ;SE MANDA UN PULSO AL ENEABLE
NOP
NOP
LD PTBOUT, #10H
CALL DELAY1

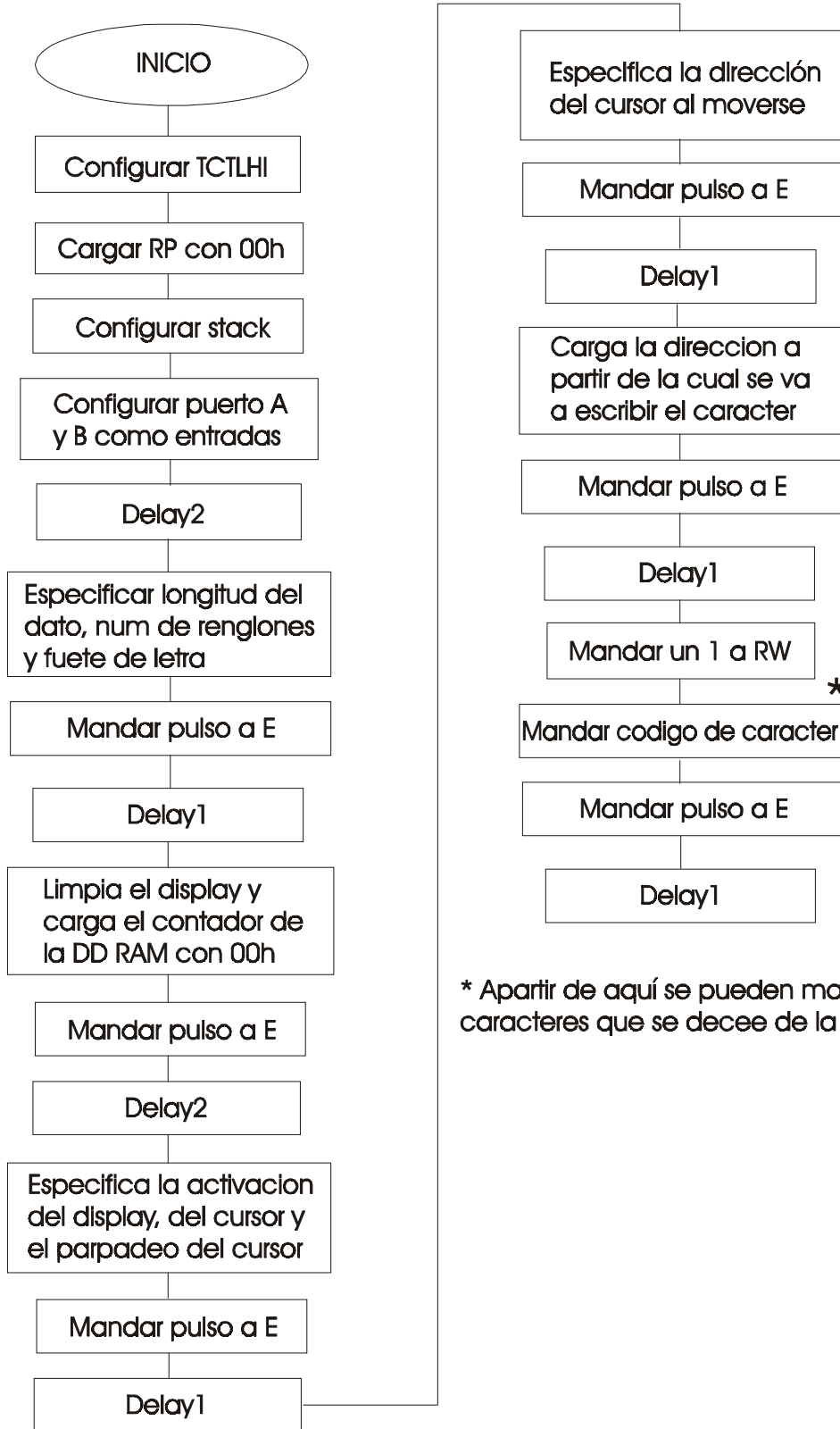
HALT
```

```
DELAY1:
LD R1, #0FFH
DECREM
DEC R1
JP NZ, DECREM
RET
```

```
DELAY2:
LD R1, #0FFH
LDR2:
LD R2, #7FH
DECR2:
DEC R2
JP NZ, DECR2
DEC R1
JP NZ, LDR2
RET
```

```
END
```

7.6 Diagrama de flujo:



* Apartir de aquí se pueden mandar los caracteres que se decee de la misma forma

7.7 Cuestionario:

- 1.-¿Cuántos bits se necesitan para programar el display?
- 2.-¿Cuántos bits se necesitan para el envío de datos al display?
- 3.-¿De qué manera se puede programar el display usando solamente 10 bits de salida del microcontrolador?
- 4.-¿Por qué se ejecuta un retardo después de haber mandado un dato al microcontrolador del display?
- 5.-¿Para qué sirve la [instrucción](#) HALT al termino del programa?

7.8 Conclusiones:

Se aprendió a desplegar texto programado a un Display de Cristal Líquido mediante la utilización del microcontrolador Z8^{PLUS}, de esta manera se comprobó que este microcontrolador es un poco ineficiente para este tipo de aplicaciones ya que cuenta con muy pocos [puertos](#) de salida o entrada.