

EXPERIMENTO 8

Aplicación del Display de Cristal Líquido

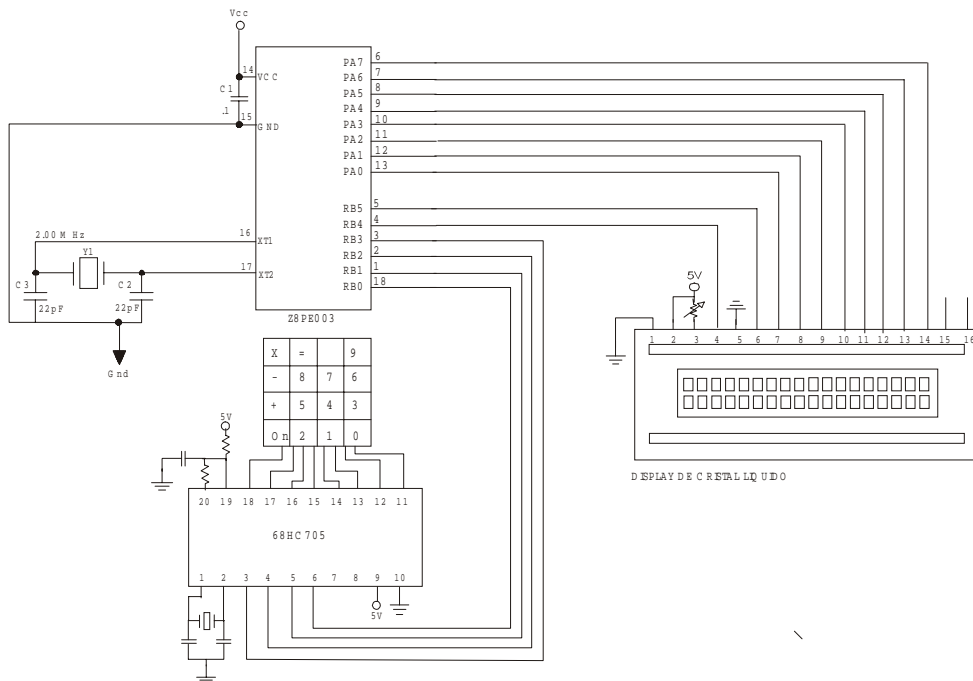
8.1 Propósito:

Se diseñará una multiplicadora, restadora y sumadora utilizando las [instrucciones](#) aritméticas, y se sugerirá un algoritmo para la multiplicación desplegando los resultados en un Display de Cristal Líquido haciendo uso de los conocimientos adquiridos en el experimento 7 y se utilizará un teclado matricial para ingresar los datos.

8.2 Material utilizado:

- Circuito básico del Z8^{PLUS}
- Emulador Z8M001
- 1 Display de Cristal Líquido
- 1 potenciómetro de 5k Ω
- 1 decodificador para el teclado matricial
- 1 teclado matricial de 4x4

8.3 Diagrama del circuito:



8.4 Descripción del experimento:

En este experimento ya se debió haber aprendido a usar el Display de Cristal Líquido ([experimento 7](#)). En el teclado matricial hay 16 teclas, algunas de ellas corresponden a los números del 0 al 9 y otras corresponden a las teclas +, -, x, on e =. Lo primero es ejecutar los pasos para programar el display ([experimento 7](#)). Para poder empezar a hacer las operaciones se debe de empezar por oprimir la tecla ON así que se lee el teclado hasta que se halla oprimido dicha tecla. Después, ya que esta calculadora se limita a hacer operaciones con números de 1 al 9 se manda por el display el enunciado “ Teclee un número” y se procede a leer el teclado para ver cual de los números se ha tecleado, dicho número se manda por el display y se guarda en el registro R1. Después de esto se manda el enunciado “ Teclee operación “ y se vuelve a leer el teclado hasta que se oprima alguna de las teclas +, -, o x, dependiendo de cual de estas tres teclas se halla oprimido se guarda en un registro el número 1, 2 o 3 respectivamente para poder identificar la operación que se va a efectuar. Luego se manda el enunciado “ Teclee otro número” por el display y se lee teclado hasta que se oprima alguna tecla correspondiente a un número, el cual se guarda en el registro R2 y se saca por el display. En este momento ya se tienen los dos números guardados en dos diferentes registros y también se sabe cual es la operación que se va a efectuar, entonces se manda un enunciado que diga “Teclee el signo =“ y se lee el teclado hasta que se oprima la tecla que corresponde al signo igual, al oprimirse esta tecla se saca por el display el resultado junto con el enunciado “resultado”. Para obtener este resultado, dependiendo del tipo de operación que se vaya a efectuar se llama a la subrutina correspondiente (suma, resta o multiplicación) para que se efectúe el algoritmo correspondiente.

Ya que se obtiene el resultado todo el programa vuelve a empezar leyendo otra vez el teclado hasta que se vuelva a oprimir la tecla ON.

8.5 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
PTAIN	EQU	%0D0

.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, #030H
```

;INICIALIZACION DEL DISPLAY ([ver exp. 7](#))

;ENVIAR MENSAJE "Teclee un numero" ([ver exp. 7](#))

LEE_1_TECL:

```
CP PTBIN, #01DH      ;TECLA 0?
JP Z, TCERO
CP PTBIN, #01CH      ;TECLA 1?
JP Z, TUNO
CP PTBIN, #01BH      ;TECLA 2?
JP Z, TDOS
CP PTBIN, #01EH      ;TECLA 3?
JP Z, TTRES
CP PTBIN, #019H      ;TECLA 4?
JP Z, TCUATRO
CP PTBIN, #016H      ;TECLA 5?
JP Z, TCINCO
CP PTBIN, #010H      ;TECLA 6?
JP Z, TSEIS
CP PTBIN, #018H      ;TECLA 7?
JP Z, TSIETE
CP PTBIN, #015H      ;TECLA 8?
JP Z, TOCHO
CP PTBIN, #01FH      ;TECLA 9?
JP Z, TNUEVE
JP LEE_1_TECL
```

TCERO:

```
LD R1, #00H
CALL SACA0
JP LEE_OPERACION
```

TUNO:

```
LD R1, #01H
CALL SACA1
JP LEE_OPERACION
```

TDOS:

```
LD R1, #02H
CALL SACA2
JP LEE_OPERACIÓN
```

.....Y ASI SUCESIVAMENTE HASTA T9

TMAS:

```
LD R0, #01H
CALL SACAMAS
JP LEE_2_TECL
```

TMENOS:

```
LD R0, #02H
CALL SACAMENOS
JP LEE_2_TECL
```

TPOR:

```
LD R0, #03H
CALL SACAPOR
JP LEE_2_TECL
```

LEE_OPERACION:

```
CALL MENSAJE_2
```

LEE_OP:

```
CP PTBIN, #013H      ;PREGUNTA POR TECLA +
JP Z, TMAS
CP PTBIN, #012H      ;PREGUNTA POR TECLA -
JP Z, TMENOS
CP PTBIN, #011H      ;PREGUNTA POR TECLA x
JP Z, TPOR
JP LEE_OP
```

LEE_2_TECL:

```
CALL MENSAJE_3
```

LEE_2:

```
CP PTBIN, #01DH      ;TECLA 0?
JP Z, TCERO2
CP PTBIN, #01CH      ;TECLA 1?
JP Z, TUNO2
CP PTBIN, #01BH      ;TECLA 2?
JP Z, TDOS2
CP PTBIN, #01EH      ;TECLA 3?
JP Z, TTRES2
CP PTBIN, #019H      ;TECLA 4?
JP Z, TCUATRO2
CP PTBIN, #016H      ;TECLA 5?
JP Z, TCINCO2
CP PTBIN, #010H      ;TECLA 6?
JP Z, TSEIS2
```

```
CP PTBIN, #018H      ;TECLA 7?
JP Z, TSIEETE2
CP PTBIN, #015H      ;TECLA 8?
JP Z, TOCHO2
CP PTBIN, #01FH      ;TECLA 9?
JP Z, TNUEVE2
JP LEE_2
```

```
TCERO2:
LD R2, #00H
CALL SACA0
JP LEE_IGUAL
```

```
TUNO2:
LD R2, #01H
CALL SACA1
JP LEE_IGUAL
```

```
TDOS2:
LD R2, #02H
CALL SACA2
JP LEE_IGUAL
```

.....Y ASI SUCESIVAMENTE HASTA TNUEVE2

```
LEE_IGUAL:
CALL MENSAJE_4
LEE_IGUAL2
CP PTBIN, #014H
JP NZ, LEE_IGUAL2
CALL SACAIGUAL
CP R0, #01H
JP Z, SUMA
CP R0, #02H
JP Z, RESTA
CP R0, #03H
JP Z, MULTIPLICACION
```

```
SUMA:
ADD R1, R2
CALL LEER_MANDAR
JP XON
```

```
RESTA:
SUB R1, R2
CALL LEER_MANDAR
JP XON
```

```
MULTIPLICACION:
```

```
LD R3, R1
DECREMR2:
DEC R2
CP R2, #00H
JP NZ, SUMAR12
CALL LEER_MANDAR
JP XON
SUMAR12:
ADD R1, R3
JP DECREMR2
```

```
LEER_MANDAR:
CP R1, #00H
JP Z, RES0
CP R1, #01H
JP Z, RES1
CP R1, #02H
JP Z, RES2
CP R1, #03H
JP Z, RES3
CP R1, #04H
JP Z, RES4
CP R1, #05H
```

.....Y ASI SUCESIVAMENTE HASTA EL 81, SIN TOMAR EN CUENTA LOS NUMEROS PRIMOS

```
RES0:
CALL SACA0
RET
```

```
RES1:
CALL SACA1
RET
```

```
RES2:
CALL SACA2
RET
```

.....ASI SUCESIVAMENTE HASTA RES81

```
RES81:
CALL SACA8
CALL SACA1
RET
```

```
XON:
CALL MENSAJE5
```

```
XON2:
CP PTBIN, #01AH
JP NZ, XON2
JP INICIO_DISPLAY
```

```
SACA0:
LD PTAOUT, #30H
```

```

CALL MANDA
RET
SACA1:
LD PTAOUT, #31H
CALL MANDA
RET
.....ASI HASTA SACA9
SACAPOR:
LD PTAOUT, #58H
CALL MANDA
RET
SACAMAS:
LD PTAOUT, #2BH
CALL MANDA
RET
SACAMENOS:
LD PTAOUT, #2DH
CALL MANDA
RET
SACAIGUAL:
LD PTAOUT, #3DH
CALL MANDA
RET

MANDA:
LD PTBOUT, #30H
NOP
NOP
LD PTBOUT, #10H
CALL DELAY1
RET

MENSAJE_2:
LD PTAOUT, #0C0H           ;PASO 5 CAMBIO DE RENGLON
LD PTBOUT, #20H
NOP
NOP
LD PTBOUT, #00H
CALL DELAY1

LD PTBOUT, #10H           ;ESCRITURA
LD PTAOUT, #54H           ;T
CALL MANDA
LD PTAOUT, #65H           ;E
CALL MANDA

```

...ASI HASTA COMPLETAR EL MENSAJE "TECLEE OPERACIÓN"

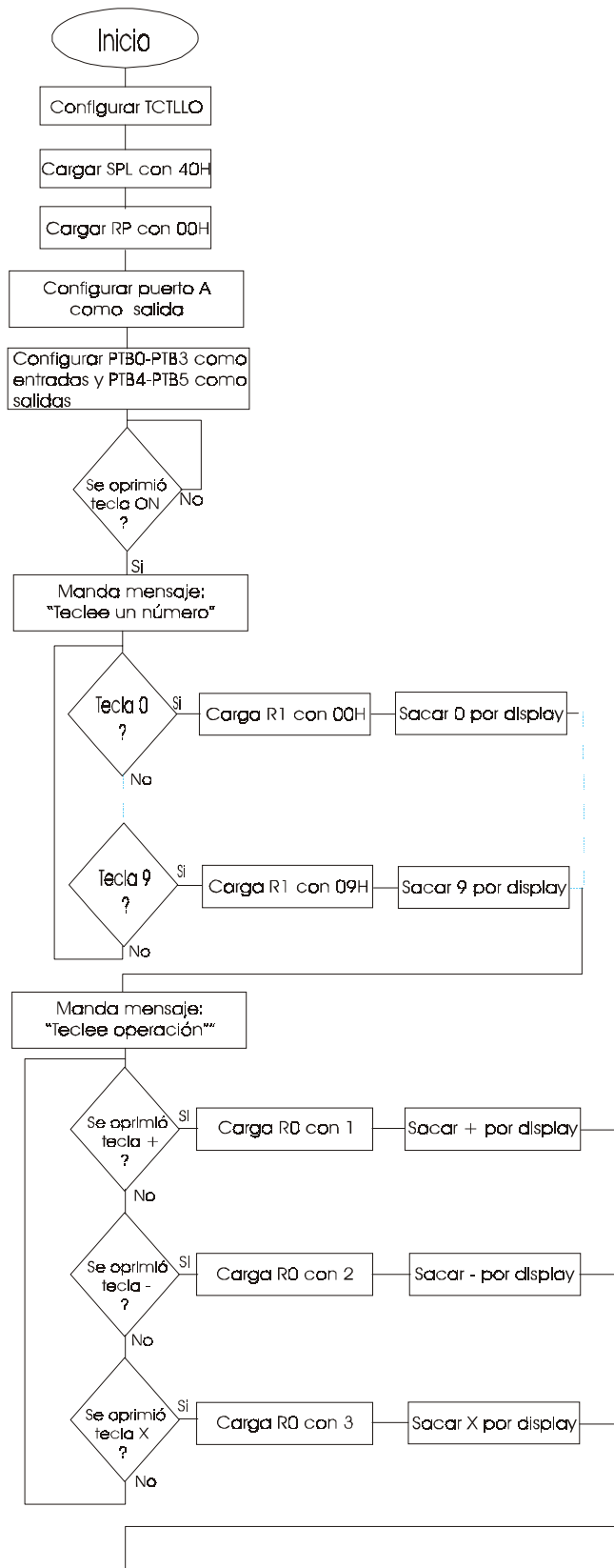
MENSAJE3:
 ENVIAR MENSAJE "Teclee otro num." (ver exp. 7)
MENSAJE4:
 ENVIAR MENSAJE "Teclee signo =" (ver exp. 7)
MENSAJE5:
 ENVIAR MENSAJE "RESULTADO." (ver exp. 7)

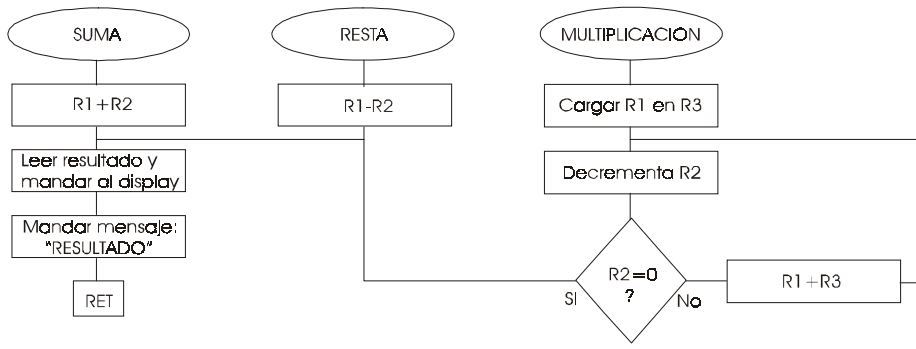
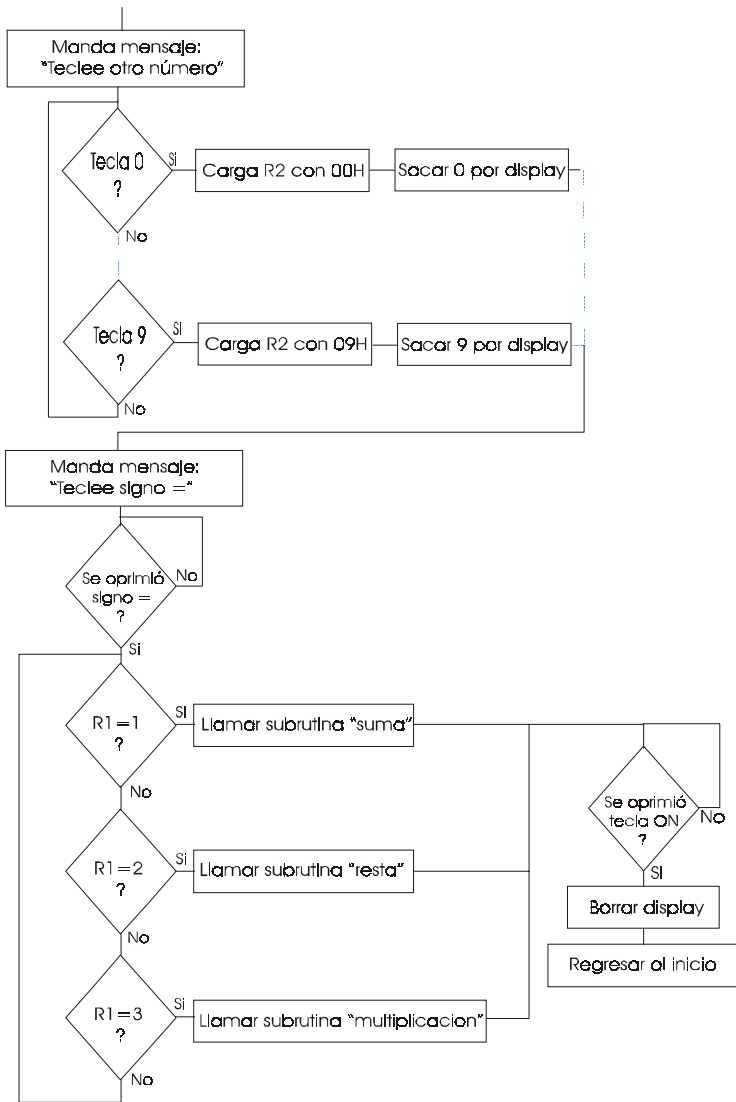
DELAY1:
 LD R6, #0FFH
DECREM
 DEC R6
 JP NZ, DECREM
 RET

DELAY2:
 LD R6, #0FFH
LDR2:
 LD R7, #0FFH
DECR2:
 DEC R7
 JP NZ, DECR2
 DEC R6
 JP NZ, LDR2
 RET

END

8.6 Diagrama de flujo:





8.7 Cuestionario:

- 1.-¿Qué instrucción se utiliza para hacer las sumas?
- 2.-¿Cuál es el algoritmo de la multiplicación?
- 3.-¿Qué restricciones se tienen para la resta?
- 4.-¿De cuántos dígitos se pueden hacer las operaciones?
- 5.-¿De qué manera se podrían obtener resultados negativos en la resta?

8.8 Conclusiones:

Ya que el microcontrolador Z8^{PLUS} solo cuenta con 2 puertos, uno de 8 bits y el otro de 6, se tuvo que utilizar un decodificador para el teclado matricial para optimizar los puertos ya que el display de cristal líquido necesita un total de 10 bits para su programación. Se aprendió a desplegar texto ingresando los datos mediante un teclado matricial, se conocieron las instrucciones aritméticas y se sugirió un algoritmo de sumas sucesivas para la multiplicación.