

# EXPERIMENTO 1

## Uso de los puertos

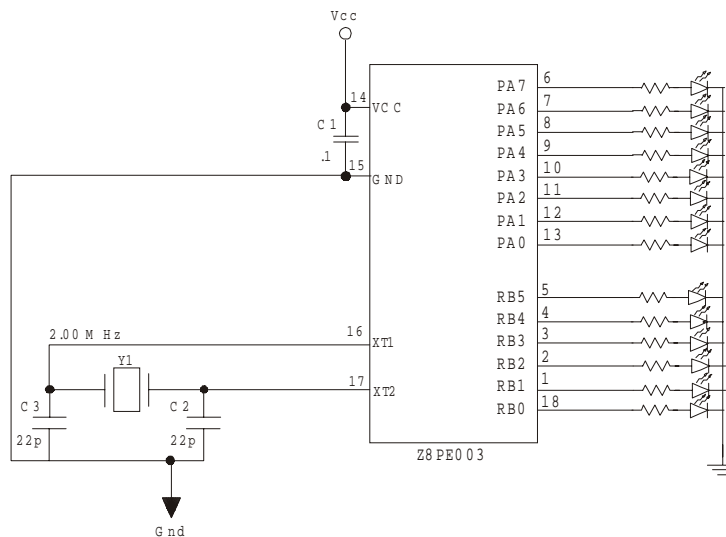
### 1.1 Propósito:

Probar mediante un programa sencillo, el funcionamiento del emulador y de la tarjeta del microcontrolador usando los puertos A y B como las salidas de pulsos, los cuales deberán ser suficientemente lentos para ser apreciados a simple vista por medio de leds.

### 1.2 Material utilizado:

- Circuito básico del Z8<sup>PLUS</sup>
- Emulador Z8M001
- 14 leds
- 14 resistencias de 1.2 k $\Omega$

### 1.3 Diagrama del circuito:



### 1.4 Descripción del experimento:

Antes de empezar cualquier programa se deben de definir las variables, o sea, definir cuales son las direcciones de cada uno de los registros que se van a utilizar en el programa, para que el compilador reconozca a cada uno de ellos, y no tengamos que poner

la dirección del registro, si no, llamarlo por su nombre. Por ejemplo, al registro PTAIN le corresponde la dirección D0H (ver tabla de [registros de control y periféricos](#)) entonces se usa el mnemónico EQU de la siguiente manera:

PTAIN EQU % 0D0 o PTAIN EQU 0D0H

Los números hexagesimales se pueden representar de dos formas, usando el prefijo % o el sufijo H, de cualquiera de estas dos maneras le damos a entender al compilador que se trata de números hexagesimales (ver tabla de símbolos).

El mnemónico que se utiliza para indicar la dirección a partir de la cual se va a escribir el programa es .org seguido de la dirección 20H, la cual, es en donde deben empezar los programas para el caso del Z8<sup>PLUS</sup> (ver mapa de memoria):

.org 20H

Si se desea hacer algún cambio en el tiempo de conteo del [WDT](#), esta instrucción debe ser la primera en ser ejecutada (ver Watch Dog Timer). Después, se debe poner el Register Pointer con la dirección del Working Register que se va a utilizar, en los 4 bits más significativos del RP, y los 4 bits menos significativos del RP corresponden al registro que se este utilizando. En este caso podemos cargar el RP con ceros para usar el WRG 0.

El [Stack Pointer](#) se puede cargar con la dirección 40H, ya que esta es una dirección adecuada para tener el stack, mas no necesariamente debe de ser esta, todo depende del espacio de memoria.

Para configurar los puertos A y B como salidas se usan los registros PTADIR, para el puerto A y PTBDIR para el puerto B, en los cuales se debe indicar con un 1 los bits que se quieran configurar como salidas, y con un 0 los bits que se quieran configurar como entradas, recordando que el puerto B solo tiene 6 bits y los bits 6 y 7 del registro PTBDIR deben ser cero.

Para sacar un número por el puerto A o por el puerto B se carga el número en el registro PTBOUT o PTAIN, al cargar estos registros con un número, este es mandado directamente al puerto. Primero cargamos PTBOUT con un 3FH que corresponde a mandar unos por el puerto B, y PTAOUT se carga con FFH. Después, mandamos llamar una subrutina de retardo y se ejecuta la instrucción complemento para que los bits de los puertos cambien de estado alto

a estado bajo, por último se manda llamar otra vez la subrutina de retardo y se vuelve a repetir el procedimiento.

Al final del programa se debe poner END para indicar que se ha terminado el programa.

### 1.5 Programa:

```
TCTLHI    EQU  %0C1 ;           SE DAN DE ALTA LAS VARIABLES
PTBDIR    EQU  %0D6
PTBSFR    EQU  %0D7
PTBOUT    EQU  %0D5
PTBIN     EQU  %0D4
PTASFR    EQU  %0D3
PTADIR    EQU  %0D2
PTAOUT    EQU  %0D1
PTAIN     EQU  %0D0

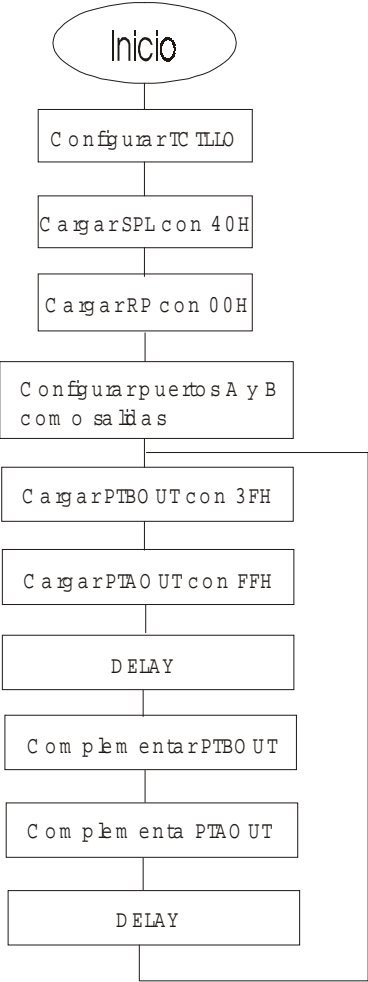
.org 020H

INICIO:
    LD TCTLHI, #070H
    LD RP, #00H
    LD SPL, #40H
    LD PTADIR, #0FFH;          CONFIGURA PUERTO A COMO SALIDA.
    LD PTAOUT, #0FFH
    CALL DELAY
    COM PTAOUT
    CALL DELAY
    JP INICIO

DELAY:
    LD R1, #0FFH
LDR2:
    LD R2, #0FFH
DECR2:
    DEC R2
    JP NZ, DECR2
    DEC R1
    JP NZ, LDR2
    RET

END
```

1.6 Diagrama de flujo:



## 1.7 Cuestionario:

- 1.- ¿Cuántos puertos tiene el microcontrolador Z8PE003 y con cuantos bits cuenta cada uno de ellos?
- 2.- ¿Cuál es la primera instrucción que se debe poner antes de empezar un programa?
- 3.- ¿Cuál es el puerto que contiene las funciones especiales y cuáles son estas?
- 4.- ¿En qué dirección de la memoria se debe empezar a escribir el programa y por qué?
- 5.- ¿Para qué sirve el Register Pointer?
- 6.- ¿Cuáles son los registros que se utilizan para configurar los puertos?
- 7.- ¿De qué manera se saca o se ingresa un dato por los puertos?

## 1.8 Conclusiones

Con este experimento se han aprendido a utilizar los puertos, el register pointer y a empezar a programar el microcontrolador, se conocieron las palabras de control, y se manejaron las instrucciones básicas. A partir de este experimento se pueden llevar a cabo sistemas de control como los de una bomba de agua, un contador de vueltas para un motor, un contador en binario o hexagesimal etc.