

Diseño de una interfase para el control y registro de experimentos psicológicos con una microcomputadora de bajo costo¹

*Designing an interface to control and record psychological
experiments with a low cost microcomputer*

Elias Robles y José Cohen

Centro de Investigación en Alimentación y Desarrollo, A.C.
Universidad Nacional Autónoma de México-Iztacala

RESUMEN

Se describen los puertos de entrada y salida, líneas de control y detalles de programación necesarios para utilizar la microcomputadora TS1000 en el control y registro de experimentos psicológicos.

DESCRIPTORES: Interfase, microcomputadoras, control, registro.

ABSTRACT

This paper describes the necessary input and outputs ports, control lines and programming procedures to use the TS1000 microcomputer to control and record psychological experiments.

DESCRIPTORS: Interface, microcomputer, control, recording.

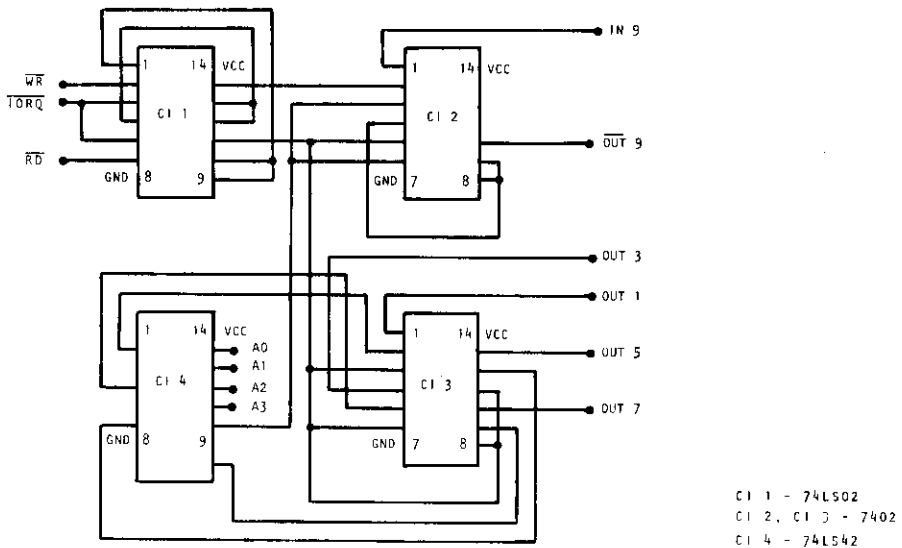
El uso de las computadoras digitales en el laboratorio de investigación psicológica se justifica tanto en términos del costo por estación experimental, como por su versatilidad. Actualmente una microcomputadora TS 1000 que puede ser programada en BASIC y con 2 KB de memoria útil al usuario, cuesta el equivalente a 10,000.00 M/N. Es decir, puede adquirirse por el mismo precio que cuatro módulos de compuertas lógicas como las que se emplean en el equipo de estado sólido. Además, cuando se amplía su capacidad de memoria, la máquina puede almacenar y analizar matemáticamente los datos que ha captado. Por estas razones optamos por diseñar una inter-

Pueden obtenerse copias de este trabajo escribiendo a: Ing. José Cohen, ENEP Iztacala, TELE (L-303), Ap. Postal 314, Tlalnepantla, Edo. de México.

¹ Todos los diagramas que aquí se presentan son compatibles también con las microcomputadoras ZX81, TS 1500 y TS 2068.

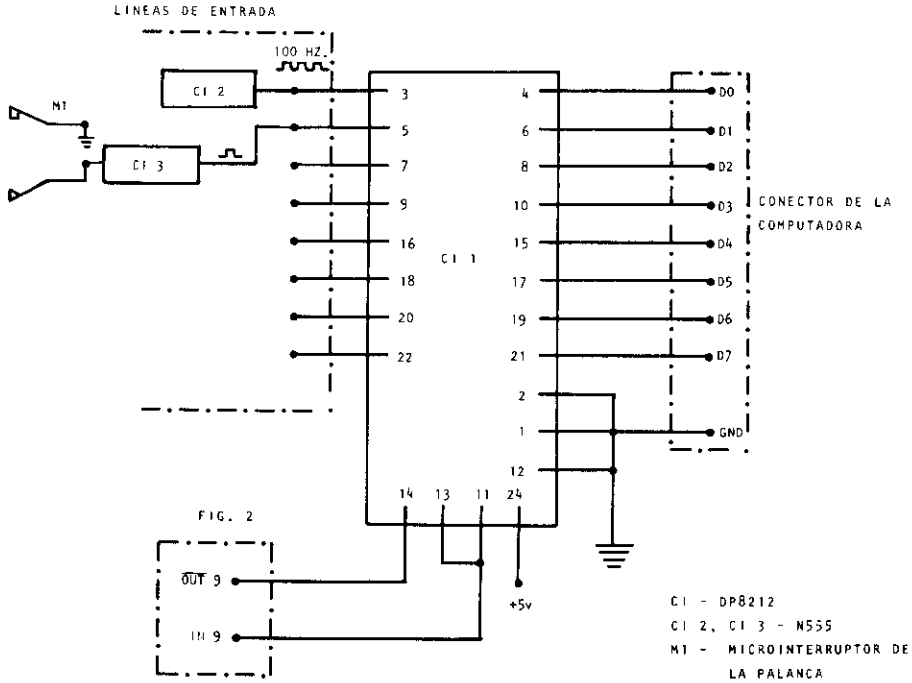
fase que permita el control y registro de eventos en ambientes experimentales para el estudio del comportamiento.

Líneas de Control. La computadora TS 1000 constituye un sistema basado en el microprocesador Z80A. Así, en teoría es capaz de interactuar con 256 puertos de entrada y otros tantos de salida. Sin embargo, debido a que la decodificación de las líneas que emplea el sistema es incompleta, sólo pueden usarse la mitad de dichas líneas. Es decir, sólo pueden utilizarse como direcciones válidas para los puertos, las líneas con códigos impares (1, 3, 5, 7, ..., 255). La figura 1 muestra la decodificación de las direcciones 1, 3, 5, 7 y 9 mediante un 74LS42 en este diseño las líneas 1, 3, 5 y 7 se decodifican como líneas de salida exclusivamente mientras que la línea 9 se decodifica para puerto de entrada.



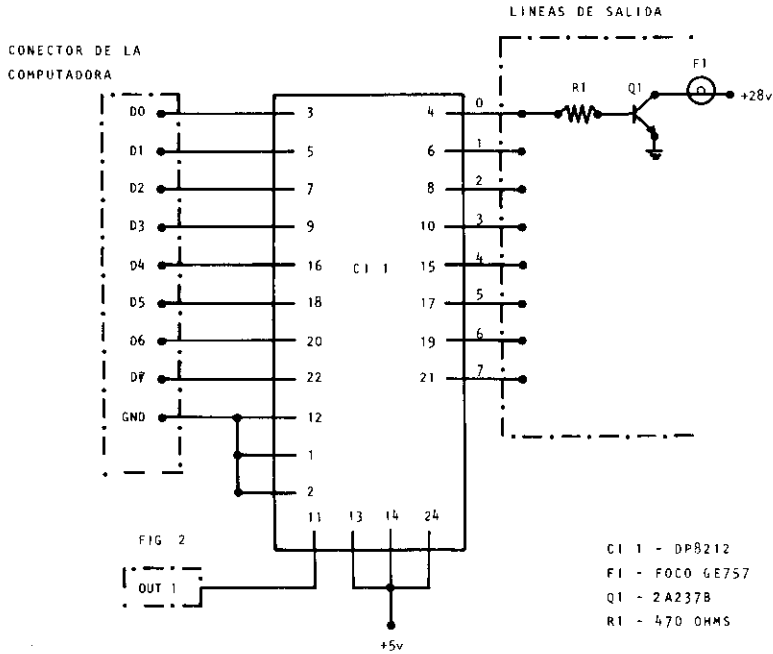
Puerto de Entrada. De los diversos circuitos integrados que pueden emplearse en el diseño de puertos digitales de entrada y salida, elegimos el DP8212 por contar con la lógica necesaria para el control de todas las funciones, los flip-flops D, los excitadores de tres estados y, también, por su disponibilidad en el mercado nacional. La figura 2 muestra en forma esquemática, el circuito del puerto de entrada con dirección 9. Debe notarse que se incluye una línea de escritura, aun tratándose de un puerto de entrada. Esta línea tiene como propósito sincronizar la operación del microprocesador con los eventos externos; en el caso del diagrama, la línea de escritura sirve para restablecer el puerto 9. El reloj de tiempo real puede ser cualquier oscilador compatible con la serie TTL. Como condicionadores de pulso, se

emplean los circuitos integrados 555 en forma de monoestables. Así, la instrucción INA, (9) lee el dato que en ese momento se encuentra en el puerto 9, mientras que la instrucción OUT (9), A limpia los flip-flops.



Puerto de Salida. La figura 3 muestra en forma esquemática el circuito del puerto de salida con dirección 1. Ahora las líneas de salida del DP8212, pueden controlar ocho excitadores cuyas características dependerán del tipo de dispositivo que se desee operar. Por ejemplo, con el fin de encender focos (GE 757) que funcionan con 28V y consumen 80mA, puede utilizarse el transistor 2A237B. Por otra parte, con el fin de operar solenoides que funcionan con 28V y consumen 1.0A, puede utilizarse el transistor TIP120. La instrucción OUT (1), A escribe en el puerto con dirección 1, el contenido del acumulador.

Programación. El intérprete BASIC de la TS 1000 no incluye las instrucciones para escribir en o leer de los puertos periféricos. Por esta razón y por las limitantes que en cuanto a velocidad de ejecución imponen los intérpretes, se sugiere que los procedimientos de control y registro se programen en ensamblador Z80 (Barden, 1980). La forma más simple y segura de almacenar una subrutina escrita en lenguaje de máquina dentro de un programa BASIC, consiste en reservar la cantidad necesaria de localidades de memoria en la



forma de caracteres alfanuméricos como argumento de la instrucción REM con número 1 y después substituir cada uno de tales caracteres mediante la instrucción POKE, con el código de las instrucciones en lenguaje de máquina.

DIRECCION 16514



1 REM 1234567890



DIRECCION 16523

Debido a la forma de organización de la TS 1000 (Foyt, 1983), cualquier programa escrito en BASIC se almacena siempre a partir de la localidad de memoria 16509. Sin embargo, el primer caracter a substituir se encuentra en la dirección 16514 (ver Fig. 4). Con el fin de no interferir con el contenido de los registros que emplea el archivo de la pantalla, se recomienda correr los programas en el modo FAST. He aquí un ejemplo:

```

1 REM 0123456
10 FOR X = 16514 TO 16520
20 INPUT D
30 PRINT X, D
40 POKE X, D
50 NEXT X

```

Corra este programa y provea a la máquina con los siguientes datos: 0,58, 130, 64, 211, 1, 201.

Ahora enliste el programa en la pantalla y observe el argumento de la instrucción REM. Los caracteres alfanuméricos han sido substituidos por los caracteres correspondientes al código del lenguaje de máquina. De esta forma la instrucción REM sirve como recinto seguro para la subrutina. El código que ha quedado almacenado corresponde a una subrutina que escribe en el puerto de salida con dirección 1, el dato que esté contenido en la localidad de memoria 16514:

DIRECCION	ENSAMBLADOR	DECIMAL	HEXADECIMAL
16514	DATO 0	0	00
16515	LD A, (DATO)	58, 130, 64	3A, 82, 40
16518	OUT (1), A	211, 1	D3, 01
16520	RET	201	C9

Como en este caso, el regreso al BASIC se logra siempre con una instrucción del grupo de retorno. Ahora puede llamarse a esta subrutina con la expresión RAND USR 16515, ya sea como mando directo o como instrucción REM, substituya en el programa las siguientes líneas:

```

10 FAST
20 FOR X = 1 TO 255
30 POKE 16514, X
40 RAND USR 16515
50 PAUSE 10
60 NEXT X

```

Corra este programa. Como puede apreciarse, la instrucción 30 substituye secuencialmente el contenido de la localidad 16514 con los enteros comprendidos entre 1 y 255, mientras que la instrucción 40 llama a la subrutina que escribe tales datos en el puerto de salida 1. No obstante que deliberadamente se controla la velocidad de ejecución con el fin de resaltar los cambios en el puerto de salida, la subrutina se ejecuta a la velocidad de la máquina (3.25 MHz) y por tanto puede confiarse en que la computadora maneje desahogadamente dos procedimientos experimentales complejos en forma simultánea.

REFERENCIAS

- Barden, W. (1980) *The Z-80 microcomputer handbook*. Indianapolis: Sams.
- Foyt, D. C. (1983) *The ZX81/TS 1000 home computer book* Berkeley: Osborne/McGraw Hill.