

## MICROCOMPUTADORA DE CONTROL MINIMA CON LENGUAJE DE ALTO NIVEL MC-INC-2

Rodríguez G. \*Barrón R. Flores P. Infante O. Polo S. Nieves J.  
Depto. de Diseño de Instrumentación del Instituto Nacional de  
Cardiología Ignacio Chávez.  
\* Centro de Investigación Tecnológica en Computación del I.P.N.

RESUMEN -----  
La microcomputadora de control mínima MC-INC-2 se diseñó para desarrollar sistemas basados en microprocesadores de baja y mediana complejidad a bajo costo. Esta cuenta con las interfases normalmente requeridas (despliegue y teclado, puerto serie, puerto programable paralelo y un canal de comunicaciones). Para la programación del sistema se implantó una versión del lenguaje FORTH, que facilita el manejo de los recursos físicos con un lenguaje de alto nivel compacto, extensible y eficiente (en solo 4 K bytes).  
-----

## INTRODUCCION

El desarrollo de equipos basados en microprocesadores (uP) permite reducir tanto los componentes de un sistema como el tiempo de desarrollo del mismo [1]. Por otra parte, al desarrollar sistemas con microprocesadores, la programación consume una parte significativa del tiempo de desarrollo, que puede ser muy alto dependiendo del tipo de aplicación.

Debido a lo anterior y como resultado de desarrollar diferentes proyectos, se diseñó y depuró una microcomputadora (MC) de control para desarrollar equipos de baja y mediana complejidad, que cuenta con los recursos básicos y un canal de comunicaciones con las señales típicas de microprocesadores que permite agregarle interfases específicas. Por la parte de programación se desarrolló una implantación del lenguaje FORTH que integra la disposición de los recursos físicos de la máquina con un lenguaje de alto nivel compacto y eficiente.

## I ARQUITECTURA

La microcomputadora de control MC-INC-2.2 se forma como muestra la figura 1 de un microprocesador Z80 que controla el sistema, un reloj a cristal para el uP y el sistema, un decodificador de memorias para controlar el acceso a las memorias, un decodificador de puertos internos para controlar el acceso a los mismos, dos memorias EPROM de 8K, una para el lenguaje y sus utilerías y una para el programa de aplicación, una memoria RAM de 8K para variables del sistema y de la aplicación, un contador temporizador para control de interrupciones y reloj, un controlador de despliegue, teclado y bocina para manejar un despliegue de siete segmentos, un teclado matricial de 32 teclas, una bocina de una y media pulgadas de 8 Ohms, una interfase para grabación y recuperación de cassette, un puerto de comunicaciones serie RS232 para comunicarse con otras computadoras con este protocolo estandard, un puerto paralelo programable y un canal de comunicaciones para microprocesadores (Micro Canal "MC").

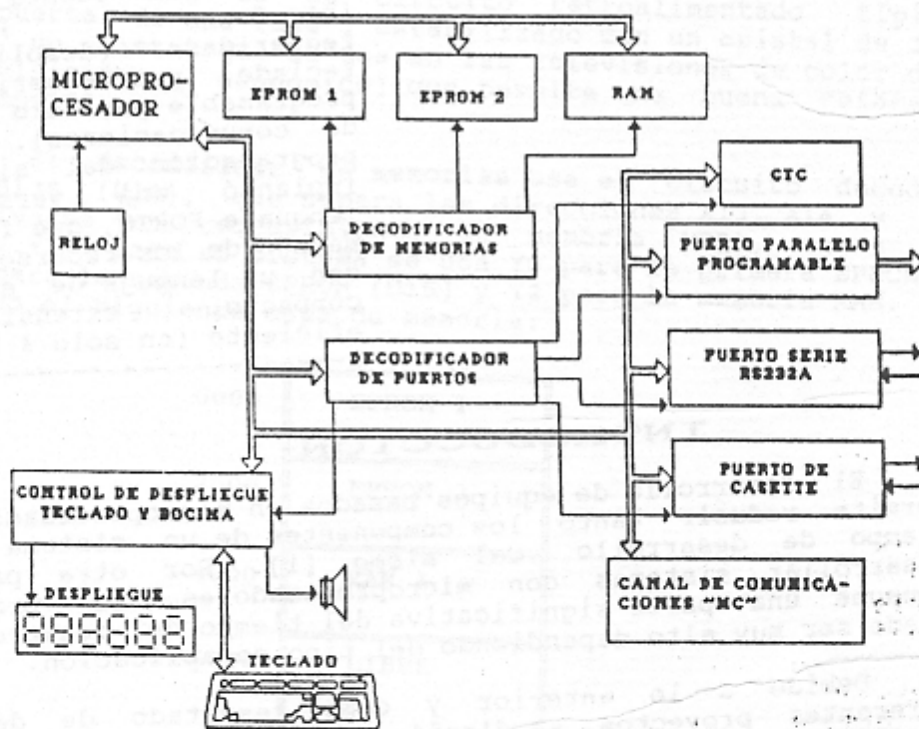


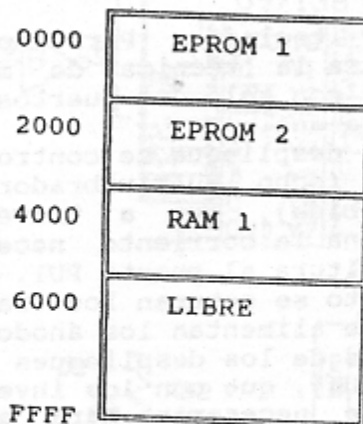
Fig. 1 Arquitectura de la microcomputadora MC-INC-2.

## II DESCRIPCION DE CIRCUITOS

**El microprocesador, las memorias y el contador temporizador.**  
 El microprocesador es un Z80A de 4Mhz, aunque se usa un reloj con cristal de 3.58 Mhz. Las memorias EPROM son dos 2764 de 8K, también se pueden usar 2732 de 4K. Para la memoria RAM se conectó una 6164 de 8K, el mismo lugar se puede usar con la memoria 6116 de 2K cambiando el puente E1 que ajusta la terminal 23 de la base para que le llegue WR para la 6116 o A11 para la 6164. El contador temporizador se hace con el circuito Z80-CTC (U11), cuyas líneas de datos y control se conectan directamente al uP y la selección de circuito (CE) se hace con el decodificador de puertos internos U7 en el periférico PY0.

**El Reloj** es un circuito retroalimentado típico con compuertas de una fase, estabilizado con un cristal de 3.58 Mhz (es un cristal que se usa en las televisiones de color de fácil adquisición y económico) que permite una buena referencia de tiempo[2].

**El decodificar de memorias** usa el circuito decodificador 74LS138 (U10), que separa las direcciones A13, A14 y A15 (al activarse un requerimiento de memoria MRQ), en 8 salidas (Y0..Y7). Para el sistema se usa Y0 para la primera EPROM (U16), Y1 para la segunda EPROM (U15) y Y2 para la memoria RAM. De esto queda el siguiente mapa de memoria:



Mapa de memoria

**El Decodificador de puertos** se forma con el circuito decodificador 74LS138 (U9), que separa las direcciones A3, A4 y A5 (al activarse un requerimiento de puertos IORQ y cuando A6 y A7 están en 0) en 8 salidas PY0 a PY7. Con las señales generadas por el decodificador se manejan los puertos que usa el sistema, de la siguiente manera:



00	CTC
08	DESP/BOC
10	SEG/COL
18	FIL/COM
20	SALCOM
28	LIBRE
FF	

#### Mapa de puertos

Donde:

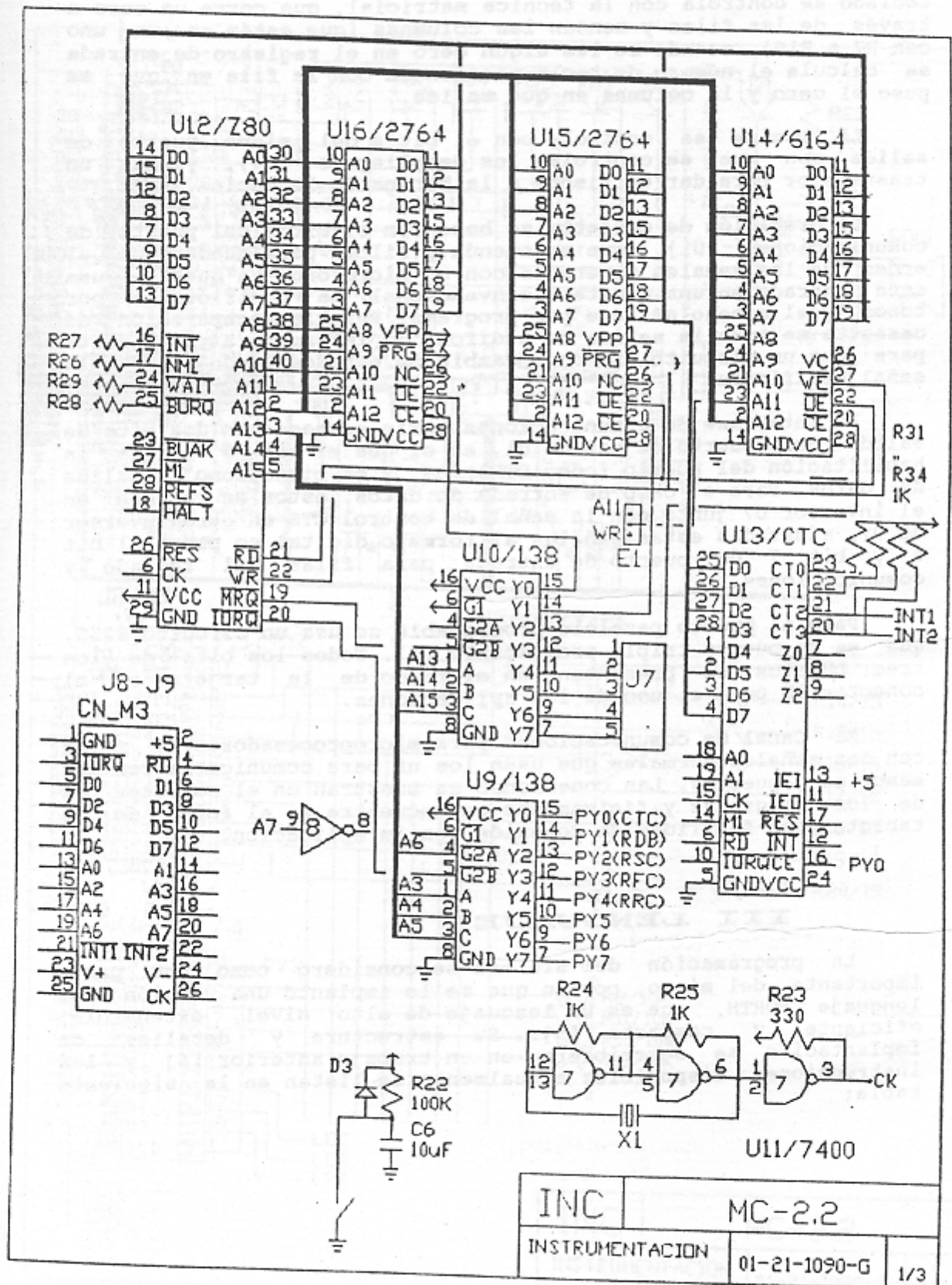
- CTC.- Puertos para el reloj contador Z80-CTC.
- DESP/BOC.- Puerto de salida para controlar el despliegue y la bocina.
- SEG/COL.- Puerto de salida para controlar los segmentos del despliegue y las columnas del teclado.
- FIL/COM.- Puerto de entrada para leer las filas del teclado y las entradas de comunicaciones.
- SALCOM.- Puerto de salida para controlar las comunicaciones.

**Control de despliegue y teclado.-** Para controlar el despliegue de 7 segmentos se usa la técnica de multiplexado, controlando los seis despliegues con sólo dos puertos y cambiando por programa el contenido de cada uno.

Los ánodos comunes de cada despliegue se controlan con un circuito retenedor 74273 (U3) (ocho multivibradores tipo D, disparados con flanco de subida), que a través de los transistores Q2 a Q7 proporciona la corriente necesaria. Este retenedor se carga con una escritura al puerto PU1. Al escribir un "1" en cada bit de este puerto se saturan los transistores y dejan pasar 5V a los emisores que alimentan los ánodos comunes.

El control de los segmentos de los despliegues se hace con otro circuito retenedor 74273 (U4), que con los inversores 7406 (U2 y U7) maneja la corriente necesaria para encender los segmentos de los despliegues (20 mA).

Para sensar el teclado se usa un registro de salida (el mismo que para los segmentos) y uno de entrada (74374, U6), el



teclado se controla con la técnica matricial, que corre un cero a través de las filas y censan las columnas (que están en un uno con R7 a R10), cuando se lee algún cero en el registro de entrada se calcula el número de tecla presionada con la fila en que se puso el cero y la columna en que se lee.

La bocina se controla con el bit 6 del mismo puerto de salida con que se controlan los despliegues (U4), y con un transistor para dar corriente a la bobina de la bocina.

La grabación de cassette se hace con el bit 2 del puerto de comunicaciones (U1), que se atenúa y filtra para quedar en el orden de las señales generadas con un micrófono, ya que se usa esta entrada en una casetera convencional. La codificación por tonos y el manejo se hace por programa. Para la recuperación de cassette se toma la salida de audífono de la casetera y se ajusta para que un circuito inversor cambie de estados cuando la señal codificada.

La interfase de comunicaciones serie se hace con dos bits de salida del puerto de salida U1, en el que el bit 0 maneja la habilitación del puerto (como DTR) [3], y el bit 1 como la salida de datos. Para el caso de entrada de datos, estos se reciben en el inversor U7 junto con la señal de control CTS en otro inversor de U7, ajustadas estas señales al formato digital se pasan al bit 6 y bit 7 del puerto de entrada para filas del teclado y comunicaciones.

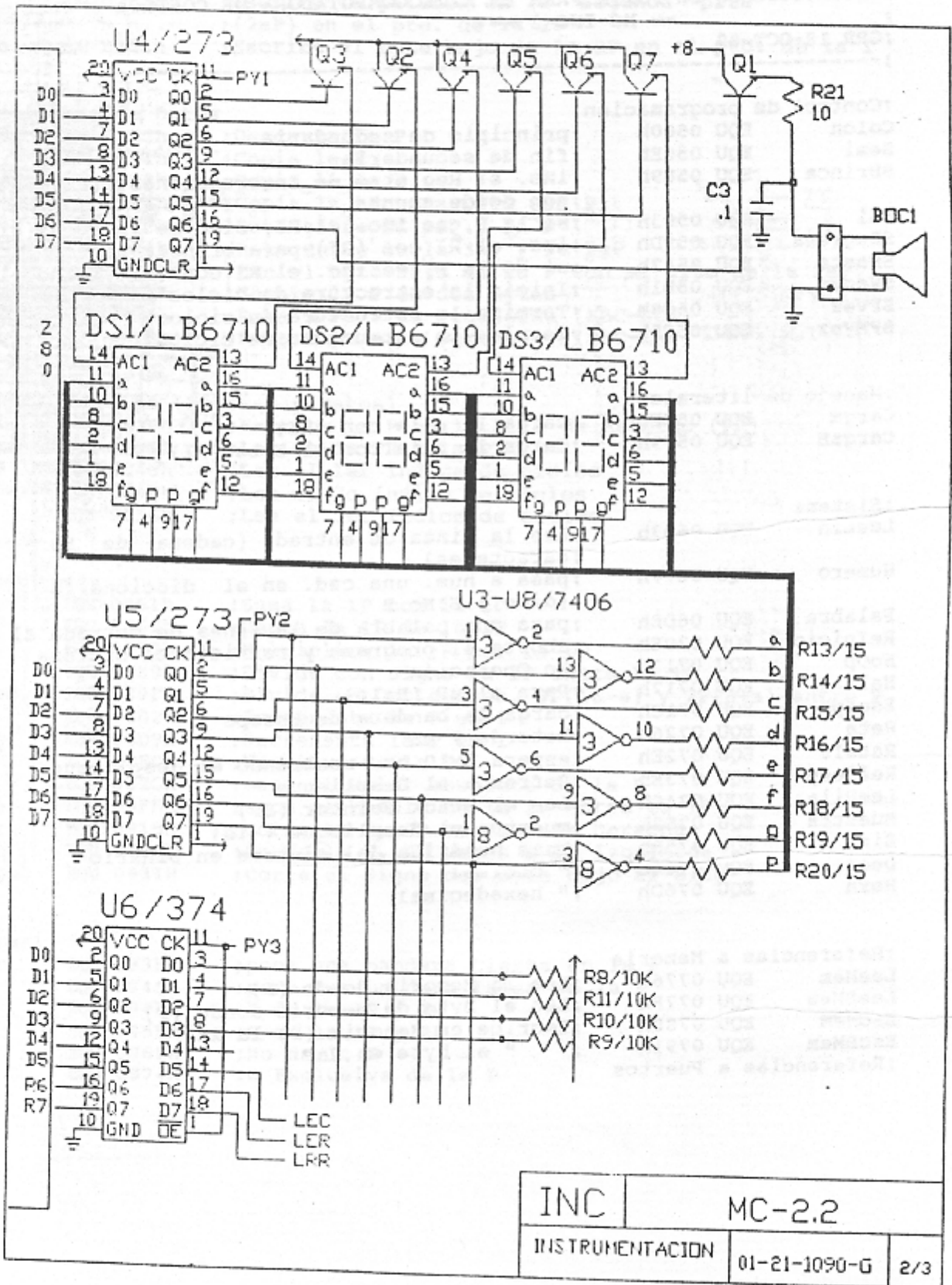
Para el puerto paralelo programable se usa un circuito 8255, que es un puerto triple programable [4]. Todos los bits de los tres puertos se presentan en el fondo de la tarjeta en el conector J5 para el uso de las aplicaciones.

El Canal de comunicaciones para microprocesadores se forma con las señales normales que usan los uP para comunicarse con las memorias y puertos. Las conexiones se muestran en el conector J8 de los diagramas y físicamente se encuentra en el fondo de la tableta para facilidad de conexión con la aplicación.

### III LENGUAJE

La programación del sistema se consideró como una parte importante del mismo, por lo que se le implantó una versión del lenguaje FORTH, que es un lenguaje de alto nivel, extensible, eficiente y compacto [5]. Su estructura y detalles de implantación se describieron en un trabajo anterior [6] y las instrucciones disponibles actualmente se listan en la siguiente tabla:





```

;-----
;M_FORTH
;-----

```

```

;INSTRUCCIONES DE FORTH PARA LA MICROCOMPUTADORA DE CONTROL
;
;   == MC-INC-2 ==
;
;GRR 18-OCT-89
;-----

```

```

;Control de programacion

```

```

Colon      EQU 0500h    ;principio de secundaria
Semi       EQU 050Eh    ;fin de secundaria
SBrinca    EQU 0589h    ;inc. el Registro de Instrucciones (RI)
                    ;con donde apunta
SSi        EQU 0593h    ;si la P, se inc. el RI, sino se inc con (RI)
SRegresa   EQU 059Dh    ;inc. el RI con (RI) para brincar atras
SHasta     EQU 05A7h    ;si NO la P, se inc. el RI con (RI) sino el
SVeces     EQU 05B1h    ;Inicia la estructura de ciclos
SFVez      EQU 05C8h    ;Termina la estructura de ciclos
SFMvez     EQU 05E5h    ;Termina la estructura de ciclos con inca

```

```

;Manejo de literales

```

```

Carga      EQU 05EEh    ;Carga la pila con (RI)
CargaB     EQU 05F9h    ;Carga la pila con el Byte (RI)

```

```

;Sistema

```

```

LeeLin     EQU 0603h    ;Lee la Linea de entrada (cadena de 80
                    ;caracteres)
Numero     EQU 0677h    ;pasa a num. una cad. en el diccionario
                    ;(dicc.) S:No B
Palabra    EQU 06DEh    ;pasa una palabra de la linea de entrada al dicc.
ReInicio   EQU 070Eh    ;aborta el programa y reinicia el sistema
NoOp       EQU 0713h    ;No Operación
Halt       EQU 0717h    ;Para al uP (Halt)
Escape     EQU 071Ch    ;carga la bandera de Escape
Reta       EQU 0726h    ;Retarda P ms
Reta10     EQU 072Eh    ;espera Px10 ms, mostrando el despliegue
RefDesp    EQU 073Eh    ;Refresca el Despliegue
LeePila    EQU 0746h    ;Lee el Stack Pointer (SP)
Muestra    EQU 0752h    ;muestra el despliegue (~1s)
Bina       EQU 075Bh    ;base numérica del sistema en binario
Deci       EQU 0764h    ;" decimal
Hexa       EQU 076Dh    ;" hexadecimal

```

```

;Referencias a Memoria

```

```

LeeMem     EQU 0776h    ;Lee la Memoria donde (P)
LeeBMem    EQU 077Fh    ;Lee el Byte de memoria donde (P)
EscMem     EQU 0788h    ;Escribe en Memoria (P) la 2aP
EscBMem    EQU 0791h    ;" el byte en 2aP

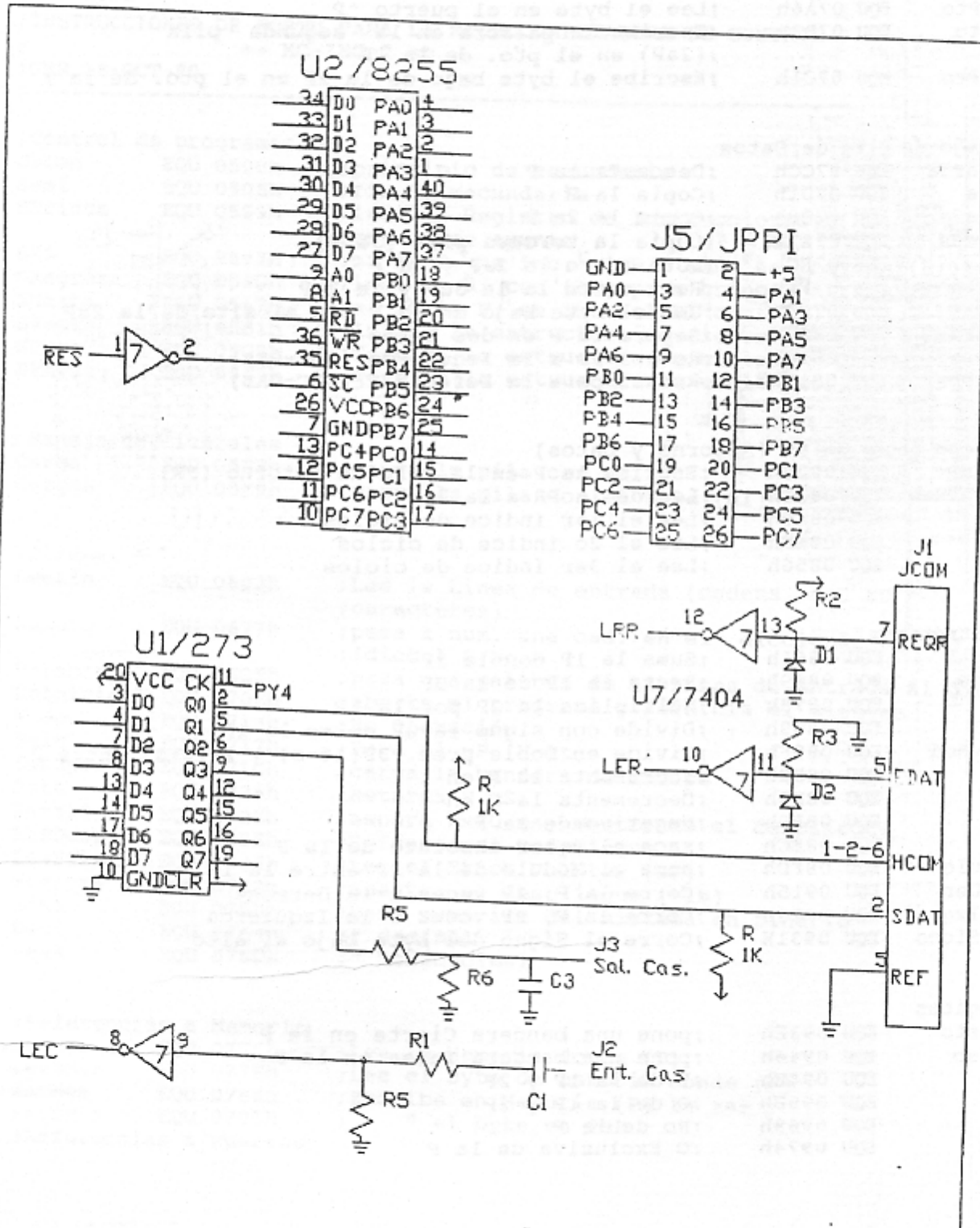
```

```

;Referencias a Puertos

```





INC	MC-2.2	
INSTRUMENTACION	01-21-1090-G	3/3

```

LeePto    EQU 0798h    ;Lee la palabra del Puerto ^P
LeeBPto   EQU 07A6h    ;Lee el byte en el puerto ^P
EscPto    EQU 07B3h    ;Escribe la palabra en la segunda pila
                    ;(2aP) en el pto. de la 1P
EscBPto   EQU 07C1h    ;Escribe el byte bajo de la 2P en el pto. de la 1

```

#### ;Manejo de Pila de Datos

```

Descarta  EQU 07CCh    ;Descarta la P
Copia     EQU 07D1h    ;Copia la P
Copia2a   EQU 07D8h    ;Copia la 2a P
Copia3a   EQU 07E1h    ;Copia la tercera pila (3a P)
Rota      EQU 07ECh    ;Rota entre la 2aP y la P
Rota2a    EQU 07F4h    ;Rota entre la la 3aP y la 2aP
Une       EQU 07FDh    ;Une el byte bajo de la P con el alto de la 2aP
Separa    EQU 0805h    ;Separa la P en dos bytes
RotaIzq   EQU 0810h    ;Rota 3 Ps a la Izquierda (ABC-BCA)
RotaDer   EQU 0819h    ;Rota 3 ps a la Derecha (ABC-CAB)

```

#### ;Manejo entre pilas (Retorno y Datos)

```

EscEnPr   EQU 0822h    ;Escribe la P en la Pila de Retorno (PR)
LeeDePR   EQU 0831h    ;Lee de la PR a la P
LeeI      EQU 0840h    ;Lee el 1er índice de ciclos
LeeJ      EQU 084Bh    ;Lee el 2o índice de ciclos
LeeK      EQU 0856h    ;Lee el 3er índice de ciclos

```

#### ;Aritméticas

```

Mas       EQU 0861h    ;Suma la 1P con la 2P
Menos     EQU 0869h    ;Resta la 1P de la 2P
Mul       EQU 0873h    ;Multiplica la 2P por la 1P
Div       EQU 0889h    ;Divide con signo la 2P entre la 1P
DDivMod   EQU 089Fh    ;Divide en Doble pres. 3P(16-s) y 2P(8+s) entre P
Inc       EQU 08D2h    ;Incrementa la P en 1
Dec       EQU 08D9h    ;Decrementa la P en 1
Neg       EQU 08E0h    ;Negativo de la P
Abs       EQU 08ECh    ;saca el valor absoluto de la P
Modulo    EQU 08FDh    ;saca el Módulo de la 2P entre la 1P
CorDer    EQU 0915h    ;Corre la P, 2P veces a la Derecha
CorIzq    EQU 0923h    ;Corre la P, 2P veces a la Izquierda
CorSigno  EQU 0931h    ;Corre el Signo del byte bajo al alto

```

#### ;Lógicas

```

Cierto   EQU 093Eh    ;pone una bandera Cierta en la P
Falso     EQU 0946h    ;pone una bandera falsa en la P
Y         EQU 094Eh    ;Y de la 1P y 2P
O         EQU 095Bh    ;O de la 1P y 2P
No        EQU 0968h    ;No de la P
OEX       EQU 0974h    ;O Exclusiva de la P

```

**;Relacionales**

Menor0	EQU 0981h	;prueba si la pila es < que cero
Igual0	EQU 098Eh	;prueba si la P es cero
Igual	EQU 099Ch	;prueba si la 2P es = que la 1P
Difer	EQU 09ABh	;prueba si la 2P es <> de la 1P
Menor	EQU 09BAh	;prueba si la 2P es < que la 1P
Mayor	EQU 09CAh	;prueba si la 2P es > que la 1P

**;Entrada Salida**

Tecla	EQU 09DAh	;prueba si se presionó una tecla
LeeCar	EQU 09E8h	;Lee un caracter
EscCar	EQU 09F3h	;escribe el caracter en la P
EscLn	EQU 09FCh	;escribe linea nueva
PosXY	EQU 0A05h	;posiciona el cursor en ^P
Son	EQU 0A11h	;genera el sonido de la P, 2P veces
Car	EQU 0A1Ah	;convierte el no. en la P, a ascii
SigCar	EQU 0A2Ah	;
Cad	EQU 0A3Ch	;convierte el no. en la P, a cadena
EscCad	EQU 0A69h	;Escribe la Cadena en la P, hasta b7=1
EscTex	EQU 0A79h	;Escribe el Texto (RI) hasta '
EscLTex	EQU 0A8Ah	;Escribe Linea nueva y texto
EscNum	EQU 0A93h	;Escribe el no. en la P
EscCNum	EQU 0A9Bh	;Escribe el no. en la P, con los Campos en 2P
EscApP	EQU 0ACEh	;Escribe el no. ^P
DondeXY	EQU 0AD6h	;Lee la pos. del cursor

**;Puerto de comunicaciones serie Com1**

IniCom1	EQU 0AE1h	;Inicia Com1
HabCom1	EQU 0AE9h	;Habilita Com1
DesCom1	EQU 0AF0h	;Deshabilita com1
EscBCom1	EQU 0AF7h	;Escribe un Byte por Com1
LBCom1	EQU 0B00h	;lee un byte de com1, esperando start
HayCom1	EQU 0B0Bh	;verifica si hay datos en com1
LeeBCom1	EQU 0B19h	;Lee un Byte de Com1, refrescando el despliegue

**;Utilerías**

Espera	EQU 0B29h	;Espera mostrando el despliegue hasta una tecla
TonoBa	EQU 0B36h	;genera un Tono Bajo
TonoMe	EQU 0B43h	;genera un tono Medio
Campana	EQU 0B4Fh	;Campana
TonoAl	EQU 0B4Fh	;genera un tono Alto
Tamano	EQU 0B5Ch	;regresa el Tamaño de una cadena ap. por la P
LimitaEnt	EQU 0B62h	;Limita el tamaño de la LinEnt para lectura
LeeNum	EQU 0B6Ch	;Lee un Número del teclado y lo deja en la P
LeeNumI	EQU 0BA6h	;Lee un Número Inicializado (NumAnt, campos)
LeeByte	EQU 0BF6h	;lee un byte en hex. (ByteAnt)
LeeHex	EQU 0C01h	;lee un num. hex. de 4 dig. (NumHex)
LeeDec	EQU 0C0Ch	;lee un num. dec. (NumAnt, Campos)



## IV RESULTADOS

Usando esta microcomputadora con el lenguaje se han desarrollado aplicaciones como: un Estimulador Programable [7], en el que se generan pulsos simples y complejos de estimulación para estudios electrofisiológicos, programados desde el mismo teclado de la MC-INC-2 o desde una computadora PC, un Electrocardiógrafo para Radio (presentado en otro trabajo de esta revista), un Monitor Simulador de ECG [8], que maneja patrones de ECG en memoria permanente y los muestra en un monitor de TV y un simulador de Gamacámara entre otros.

El Circuito Impreso se ha estado depurando, de tal manera que la versión actual no tiene ningún error. Para la puesta en marcha se tienen rutinas de diagnósticos que prueban los elementos del sistema.

Otra aplicación importante ha sido la enseñanza de microprocesadores, ya que además de permitir enseñar programación en ensamblador y FORTH, se pueden desarrollar proyectos de aplicación con facilidad a bajo costo. En esta área se han construido más de 100 unidades.

## V CONCLUSIONES

Con los recursos de la máquina se han resuelto diferentes proyectos con relativa facilidad.

Las instrucciones disponibles del lenguaje han permitido programar con facilidad los diferentes proyectos, aunque faltan facilidades para manejar estructuras dinámicas.

Para el futuro se está escribiendo en el mismo lenguaje un intérprete como ayuda de desarrollo, ya que actualmente se deben compilar los programas con ayuda de una PC o manualmente los pequeños.

Debido a la simplicidad del lenguaje se tiene todo el control del sistema y es fácil extenderlo en el mismo.

## REFERENCIAS

- 1.- Osborne A.: Z80 Programming for Logic Design. OSBORNE/McGraw-Hill, Inc, 1978.
- 2.- Ciarcia S.: Build your own Z80 Computer. Byte Books/McGraw-Hill, 1981.
- 3.- Seyer M.: RS232 Made Easy. Prentice-Hall, Inc. 1984.
- 4.- Intel: Component Data Catalog. Intel, 1981.
- 5.- R.G. Loeliger "Threaded Interpretive Languages". Byte Books August 1979.
- 6.- Barrón R., Rodríguez R.G. "Lenguaje Arborescente para Microcomputadoras" Rev Mex Ing Biomed 9 (1) 1988.
- 7.- Moncada E. et al: "Estimulador Programable E.P.2.1." Rev Mex Ing Biomed 9 (1) 1988.
- 8.- Tovar A.: "Monitor Simulador de ECG para enseñanza" Rev Mex Ing Biomed 9 (1) 1988.