



Recopilación anotada de artículos

Ediciones 1985-1986

Dr. Pedro E. Colla

LU7HZ

Argentina

Introducción

A fines del año 1984 fui contactado a través de mi amigo el Dr. Alberto Barengols (entonces LU4DNO) por los responsables editoriales de lo que iba a ser la revista K64, una propuesta muy interesante dirigida al segmento de computadores muy pequeños entonces muy en auge.

La IBM PC, por entonces su modelo AT, llevaba menos de 3 años de lanzado y su difusión era muy lenta en nuestro medio, mayormente en usos profesionales y en grandes empresas. La principal limitante era el costo, en el entorno de unos USD 4000 o 5000 para una configuración básica, muy (muy!) por encima de las posibilidades de quien pensara en tener una computadora por hobby o para usos no rentables.

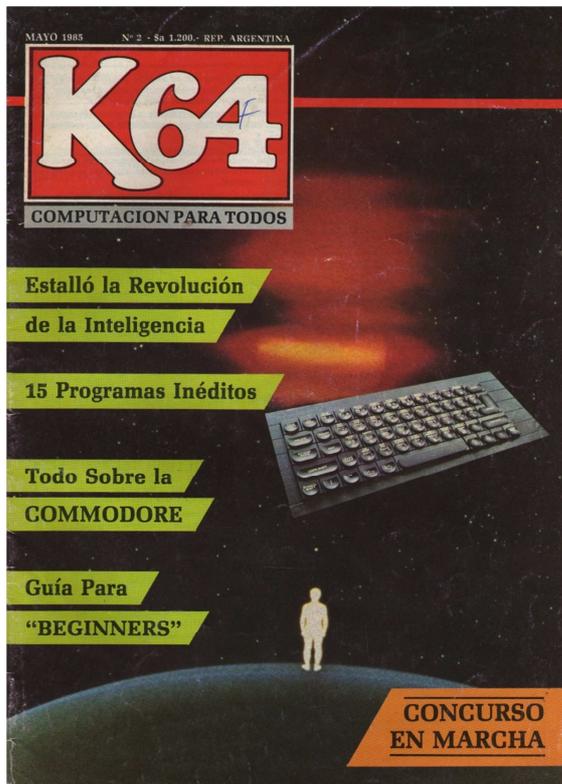
Apareció entonces una gran innovación, Sir Clive Sinclair lanzó en el Reino Unido una computadora totalmente revolucionaria denominada ZX81; no era la primer computadora para el mercado de entusiastas pero era la primera que fue lanzada con una estrategia realmente masiva, con capacidades mínimas pero aun así funcionales y con un mercado que rápidamente ofertó accesorios y software en abundancia.

Las primeras máquinas llegaron a nuestro país poco tiempo después, mayormente en su versión americana Timex-Sinclair 1000 al comienzo y posteriormente las versiones clones de origen brasilero y argentina.

La explosión puso en manos de muchísima gente una máquina pero con mas ideas que información, herramientas y proyectos a implementar. Por aquel entonces implementé numerosos proyectos con ese tipo de máquinas aprovechando cierta familiaridad con el microprocesador que empleaba (el ahora legendario Z80), al principio para proyectos de aplicación en radio y paulatinamente para proyectos de índole más general; la investigación necesaria para llevar a cabo los proyectos me fue poniendo paulatinamente en contacto con información “especial”, “chimentos” y “trucos” sobre el funcionamiento interno de la máquina que aparecían en revistas mayormente en inglés y que me permitía desarrollar utilitarios que no podían lograrse con solamente la información documentada. Llegó un momento que era capaz de utilizar segmentos de la ROM de lo que pasaba por el sistema operativo como sub-rutinas de mis programas. Eran épocas donde la standarización de funciones, la noción de sistema operativo, librerías, kits para del desarrollo eran nociones que simplemente no existían.

La revista K64, la mayoría de las máquinas de la época, basadas en procesadores de 8 bits solo podían utilizar hasta 64K de memoria, fue lanzada para cubrir un enorme vacío en publicaciones en español. La respuesta del público fue inmediata y durante los siguientes 3 años la revista fue líder en el segmento. Posteriormente las PC empezaron a ser accesibles en precio y disponibilidad por lo que las computadoras de baja gama fueron barridas del escenario. Aun asi muchas personas hicieron sus primeras armas con este tipo de computadoras y es sorprendente como, casi 25 años después de haber aparecido los artículos y aun cuando son mayormente obsoletos continuo encontrando gente que me recuerda que estos fueron sus primeros pasos en computación.

Número 2 (1985)



El primer artículo fue planeado y escrito para aparecer en el Número 1, pero decisiones editoriales lo llevaron al número 2. En ese momento estaba pensado solamente para ser un solo artículo, no una serie como termino siendo. Se trataba de un programa bastante popular por aquellos tiempos que se utilizaba para decodificar CW. Fue un programa revolucionario por entonces, no había nada (al menos que supiera entonces o me enterara posteriormente, similar. Algunos software (deficientes por cierto) habían sido publicados USA para la Tandy TRS80 o Color así como las máquinas Apple II, pero estas eran máquinas casi diez veces más caras que las Sinclair. Había también terminales especializados para decodificar CW pero estaban en el orden de USD 1000.

Este software, que además andaba muy bien merced a ciertos algoritmos de procesamiento de señales que utilizaba y la enorme velocidad posibles por el hecho de estar escritos en lenguaje de máquina.

Inicialmente quería publicar el programa en Assembler pero los editores se negaron rotundamente pues no era el modo en que publicaban el resto de los programas, los que estaban en BASIC o cuando había lenguaje de máquina involucrado solo los códigos numéricos absolutos.

Pero claro, una cosa era un programa para mover dos "sprites" en pantalla en un juego y otra era un software más o menos sofisticado. El resultado fue bastante largo y los editores comprendieron rápidamente que tenían que ajustar una o dos cosas.

El artículo, y para mi sorpresa pues no estaba al tanto de como se editaba el material que mandaba, apareció con un dibujo original hecho al efecto que siempre me gustó mucho y que afortunadamente puedo recuperar merced a la digitalización de las ediciones.

PROGRAMAS

PROGRAMA DE RADIOTELEGRAFIA PARA TS1000

Ing. Pedro E. Colla

La recepción y práctica de la radiotelegrafía ha apasionado a generaciones de amantes de la radio desde los albores de la misma.

No obstante se tropieza a menudo con la dificultad que tiene el principiante para el aprendizaje del alfabeto Morse; no es de extrañar entonces que a medida que los computadores personales encontraron su ubicación en las estaciones de radio fueran escritas numerosas piezas de software destinadas a la recepción-emisión de telegrafía en código Morse.

El programa presentemente expuesto está destinado a ser utilizado con computadores Timex-Sinclair 1000 ó similares con la configuración mínima de memoria RAM (2K) y no requiere la utilización de interfaces de hardware externas, siendo la señal a recibir ingresada a través del conector EAR y la salida en forma de tono de audio por el conector MIC, utilizados habitualmente en el computador para la lectura/grabación de programas. El programa puede ser ingresado con el auxilio de un programa Editor/Assembler, proceso para el cual se requerirá disponer del módulo de expansión de 16K (sólo durante la confección del ensamblado) o también se puede ingresar directamente desde los códigos numéricos del programa en lenguaje de máquina, para lo cual no se requerirán los ya referidos 16K del módulo de expansión.

Dada la evidente sencillez de funcionamiento la performance del programa deberá limitarse a la recepción de señales relativamente sólidas y exentas de ruido, consideraciones de relativamente poca importancia para el interesado en el trabajo esporádico en este modo o para el principiante que desea tener una primera aproximación al tema sin el aprendizaje del código Morse.

Cualquier operación fuera de los límites antes indicados es aconsejable se realice luego del aprendizaje de los códigos y prácticas de la radiotelegrafía por los métodos convencionales.

Cae fuera de las posibilidades físicas de este medio la descripción profunda e íntima del funcionamiento del programa, asumiendo

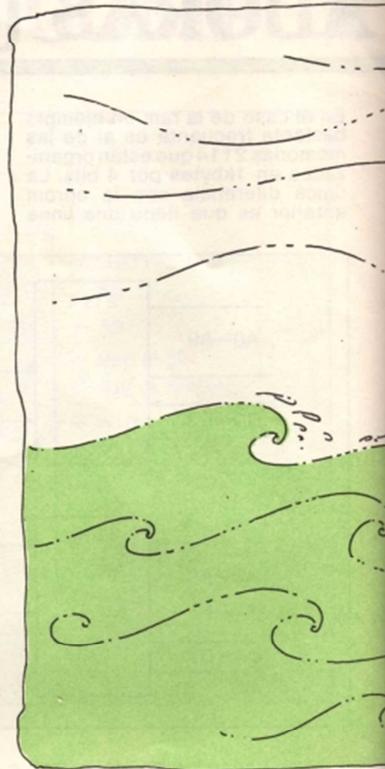
que el lector está familiarizado con los códigos mnemónicos del procesador Z80 y las acciones que los mismos desencadenan. Con las modificaciones pertinentes usuarios de otros computadores podrán adaptar este programa para el funcionamiento en el mismo.

La sección inicial del código Assembler está dedicada a la definición de las variables, tablas de traducción y textos a ser utilizados por el programa; nótese que se utilizan tablas separadas para la recepción (TABLE) y emisión (MT), esto puede lucir ineficiente a simple vista pero tiende a optimizar el acceso de las mismas en cada estado en que se las utiliza dejando márgenes de tiempo para ser empleados en secciones más críticas del programa.

Se prevee la utilización de seis mensajes pre-programados (M1 a M6) y de uno programable durante la operación de hasta 32 caracteres de longitud (MM). El resto de los textos son utilizados para la presentación de la actividad en pantalla y menús de selección.

La ejecución propiamente dicha comienza en el procedimiento llamado CW (ubicado en la dirección de memoria 17171) el cual borra la pantalla, exhibe el menú principal y espera la selección mediante el teclado de la opción que se desea emplear. Nótese que la rutina cuya dirección es OA2A corresponde a la ROM del computador Sinclair y es la encargada de borrar la pantalla de video.

La rutina SKAN permite el ingreso de caracteres desde el teclado incorporando en su lógica provisiones para que no se produzcan "rebotes" por mantener la tecla oprimida



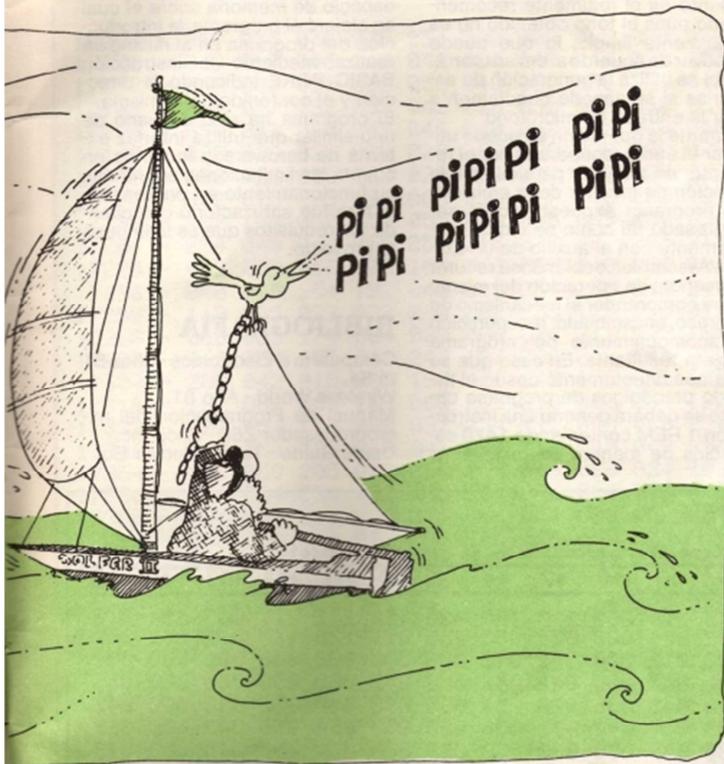
para medir períodos de tiempo largos. La rutina MS1 está destinada a la aceptación de textos desde teclado los cuales son almacenados en un área de memoria, esta rutina es utilizada en particular para ingresar el mensaje programable y la licencia del operador.

La rutina SPD actualiza las variables que son utilizadas para retardos de emisión y por lo tanto para la variación de la velocidad de emisión; la velocidad de recepción es automáticamente ajustada a la del correspondiente por la rutina respectiva.

La rutina de emisión (CWT) coloca el computador en modo FAST mediante el llamado a la rutina del ROM encargada de tal tarea (OF23) e inspecciona el teclado. Si de resultados de la inspección se obtiene un comando el mismo es ejecutado, caso contrario se intenta traducir el carácter mediante el acceso a la tabla de emisión (MT) mediante la rutina FND y se lo emite (rutina XMIT). Si no se logra encontrar un carácter en la tabla de emisión, se emite y coloca sobre la pantalla el símbolo "?".

Los comandos reconocidos en emisión son los a continuación des-

K64



criptos:
 (Shift) Q: Retorna al menú principal
 (Shift) R: Pasa a recepción
 ENTER: Borra la pantalla
 (Shift) O: Emite el texto programable
 (Shift) 1 a 6: Emite los textos pre-programados

Cada vez que se emite un carácter mediante la rutina XMIT, la misma coloca el mismo en la pantalla de video, en la cual pese a no ser visto pues el computador está en modo FAST queda registrado de tal manera de poder ser inspeccionado en cualquier momento mediante breves pasajes a modo de recepción. Nótese que la rutina que utilizamos para colocar los caracteres sobre el video pertenece nuevamente al ROM (RST 10) y que previamente se asegura con una operación AND que el carácter que se desea imprimir no sea mayor que 63, cosa que provocaría anomalías de funcionamiento.

Nótese que la emisión se lleva a cabo generando un tono de audio sobre la salida MIC del computador mediante la utilización de la instrucción OUT FFA para provocar los estados altos y CALL OF46 para llevar tal salida a estado bajo.

El resto del código se utiliza para generar retardos que controlan la frecuencia del tono y la duración de los puntos y rayas.

Las rutinas SCROL y SKAN utilizan nuevamente secciones del ROM para controlar la posición donde se imprimirá el siguiente carácter en la pantalla y para explorar el teclado respectivamente. Al respecto, la rutina que comienza en OCOE es el equivalente al comando BASIC SCROLL y la llamada a O8F5 cumple idéntica función que un PRINT AT, debiendo en este último caso estar colocado sobre el registro BC del procesador Z80 la línea y columna en la cual se desea realizar el posicionamiento. En la rutina SKAN se llama a las rutinas que comienzan en O2BB y O7BD (siempre del ROM) cuyas funciones son respectivamente explorar el teclado y traducir lo encontrado a caracteres comprensibles para el Sinclair.

La rutina de recepción comienza en el procedimiento denominado CWR y luego de borrar la pantalla y colocar el computador en modo SLOW realiza una inspección del teclado similar a la que efectúa la

rutina de emisión, con la única diferencia notable en que es ahora el comando (Shift) T el que pasa a modo de emisión.

La rutina de decodificación propiamente dicha comienza en CW1 y basa su funcionamiento en la medición de los tiempos en que se detecta la presencia de señal a la entrada (MKLEN) para mediante su comparación con mediciones anteriores derivar si por su duración la señal debe ser catalogada como punto o raya, similar medición es realizada con los espacios sin señal para determinar si se trata de una separación entre formantes de un carácter, una separación entre caracteres o un espacio en blanco. Por último la configuración de puntos y rayas obtenidas entre separaciones entre caracteres es acondicionada para su conversión mediante la utilización de la tabla de equivalencias definida al principio del programa (TABLE) en la rutina denominada CONVT, en caso que el carácter formado no encuentre su equivalencia en la tabla antes mencionada se colocará sobre la pantalla de video el símbolo "*", de otra forma el carácter que correspondiese.

Nótese que en la rutina de medición de longitudes lo que se mide no es la duración de un estado alto o bajo, pues lo que ingresa por el conector EAR es en realidad un tren de pulsos cuando hay señal, por ello el proceso de detección equivale a una "integración" por software de la señal para rescatar su envolvente. En la rutina VLDMK se realiza una validación de la medición de duración, desechándose aquellas mediciones que resultaran demasiado cortas por atribuirse a ruido; esto funciona efectivamente como un filtro pasabajos por software y su funcionamiento ha demostrado ser efectivo.

Nótese que el computador Sinclair en modo SLOW está atareado el 90% de su tiempo en generar la señal de video, y por lo tanto en los remanentes es cuando se debe realizar el procesamiento de la señal; lo que explica la necesidad de la implementación en lenguaje de máquina pues el intérprete BASIC es demasiado lento para esta tarea, aún sin existir esta condición la longitud del programa resultante prohibiría su utilización en computadores de 2K de memoria RAM.

La utilización de memorias superiores a 2K durante la operación del programa introduce demoras en el manejo del video, en especial durante el borrado de la pantalla, las cuales pueden ser evitadas desconectando la expansión mientras se utiliza el programa o en caso que esto no sea posible (en máqui-

PROGRAMAS

nas TS1500 ó similares) colocando el valor adecuado a la variable de sistemas denominada RAMTOP. La conexión física del computador es simple para la operación del programa, hasta el ingresar la señal de audio directamente desde el receptor al conector EAR del computador y mediante la utilización de un sencillo amplificador de micrófono introducir la señal disponible en el conector MIC en la entrada de micrófono del transmisor de BLU, este último requisito se debe a que en el conector MIC la señal de audio disponible no excede el nivel de algunas decenas de milivolts los que suelen resultar insuficientes; el procedimiento correcto para emisión es no obstante el control con la salida del preamplificador de un relay previo rectificado de a la señal de audio, para con él controlar la emisión a través del conector KEY del emisor. Este último procedi-

miento es el realmente recomendado pues el tono obtenido no es totalmente limpio, lo que puede motivar de acuerdo al emisor con el cual se utiliza la generación de espurias si se accede directamente por la entrada de micrófono. Durante la operación se puede utilizar el audio recuperado por el receptor de televisor para cumplir la función de monitor de la emisión. El programa expuesto podrá ser ingresado tal como se indicó previamente con el auxilio de un Editor/Assembler, debiéndose recurrir al manual de operación del mismo para comprender el mecanismo de ingreso, ensablado, recuperación y acomodamiento del programa objeto resultante. En caso que se ingrese directamente desde el listado de códigos de programa objeto se deberá generar una instrucción 1 REM conteniendo 1470 espacios de manera de reservar el

espacio de memoria sobre el cual se alojará el programa, la introducción del programa en sí mismo se realiza mediante la instrucción BASIC POKE indicando la dirección y el contenido de la misma. El programa ha sido derivado de uno similar que utiliza interfaz externa de hardware, y teniendo en cuenta las limitaciones apuntadas su funcionamiento en pruebas en el aire fue satisfactorio cumpliendo los requisitos que se le impuso al proyecto.

BIBLIOGRAFIA

- Computers & Electronics - Años 81 al 84.
- Wireless World - Año 81.
- Manual de Programación del microprocesador Z80 - Zilog Inc.
- Users Guide - Timex-Sinclair Co.

FIG 1

Listado de codigos correspondiente al programa objeto

Direcc.	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12	+13	+14	+15	
16516	-	032	058	132	064	245	219	063	230	128	032	250	241	211	063	254	010
16532	-	040	005	001	000	032	024	003	001	255	255	011	120	177	032	251	201
16548	-	195	019	067	000	000	012	000	041	042	000	000	000	000	000	000	000
16564	-	255	000	035	031	000	062	000	041	061	000	255	000	058	055	000	055
16580	-	056	057	000	255	000	045	055	000	054	055	038	000	052	053	000	042
16596	-	056	000	255	000	053	056	042	000	058	055	000	054	056	049	000	059
16612	-	046	038	000	255	000	045	055	000	052	053	000	054	057	045	000	042
16628	-	056	000	255	005	038	022	039	020	040	010	041	002	042	028	043	008
16644	-	044	030	045	006	046	023	047	009	048	255	000	026	049	003	050	004
16660	-	051	007	052	024	053	017	054	012	055	014	056	001	057	013	058	029
16676	-	059	255	000	011	060	021	061	019	062	018	063	031	028	047	029	055
16692	-	030	059	031	061	032	062	033	046	034	038	035	034	036	032	037	081
16708	-	016	114	015	044	024	105	023	075	026	070	014	084	025	045	022	255
16724	-	000	255	000	052	019	040	023	054	021	121	018	000	000	028	003	003
16740	-	003	003	003	255	000	000	255	029	001	003	003	003	003	255	030	001
16756	-	001	003	003	003	255	031	001	001	001	003	003	255	032	001	001	001
16772	-	001	003	255	033	001	001	001	001	001	255	034	003	001	001	001	001
16788	-	255	035	003	003	001	001	001	255	036	003	003	003	001	001	255	037
16804	-	003	003	003	003	001	255	038	001	003	255	039	003	001	001	001	255
16820	-	040	003	001	003	001	255	041	003	001	001	255	042	001	255	043	001
16836	-	001	003	001	255	044	003	003	001	255	045	001	001	001	001	255	046
16852	-	001	001	255	047	001	003	003	003	255	048	003	001	003	255	049	001
16868	-	003	001	001	255	050	003	003	255	051	003	001	255	052	003	003	003
16884	-	255	053	001	003	003	001	255	054	003	003	001	003	255	055	001	003
16900	-	001	255	056	001	001	001	255	057	003	255	058	001	001	003	255	059
16916	-	001	001	001	003	255	060	001	003	003	255	061	003	001	001	003	255
16932	-	062	003	001	003	003	255	063	003	003	001	001	255	027	003	001	001
16948	-	001	003	255	015	001	001	003	003	001	001	255	024	003	001	001	003
16964	-	001	255	026	003	001	001	003	003	255	050	052	055	056	042	000	000
16980	-	000	016	040	017	000	050	038	061	000	056	052	043	057	000	000	000
16996	-	000	000	000	000	000	029	037	036	032	254	000	254	000	000	000	016
17012	-	055	017	042	040	046	039	046	055	000	000	000	000	000	016	057	017
17028	-	055	038	051	056	050	046	057	046	055	254	000	000	000	016	056	017
17044	-	053	042	042	041	000	000	000	000	000	000	000	016	050	017	042	050

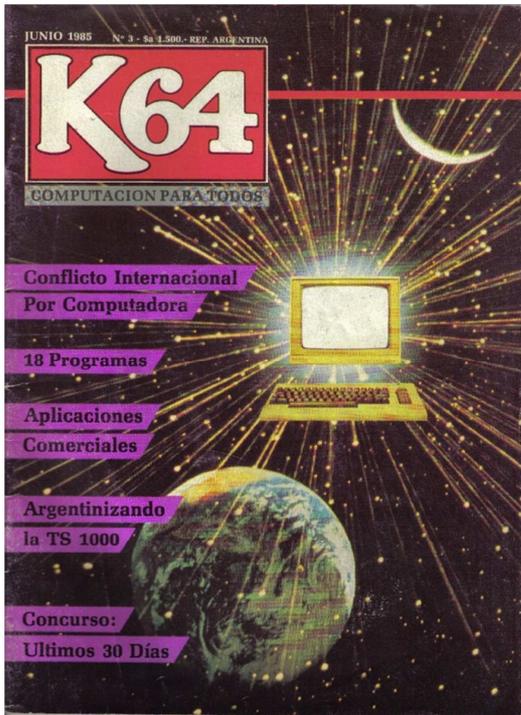


17060 - 052 055 046 038 254 000 000 000 000 000 000 000 000 000 000 000 016
 17076 - 049 017 046 040 042 051 040 046 038 255 183 255 185 255 053 056
 17092 - 042 000 057 062 053 042 000 056 053 042 042 041 255 053 056 042
 17108 - 000 057 062 053 042 000 050 042 050 255 053 056 042 000 057 062
 17124 - 053 042 000 049 046 040 042 051 040 046 038 255 255 028 028 028
 17140 - 028 028 028 028 028 028 028 028 028 028 028 028 028 028 028 028
 17156 - 028 028 028 028 028 028 028 028 028 028 028 028 028 028 255 205
 17172 - 042 010 205 194 068 033 077 066 205 169 068 205 218 068 254 057
 17188 - 202 148 067 254 055 202 000 069 254 056 040 076 254 050 040 016
 17204 - 254 049 040 055 254 216 032 227 205 218 068 254 220 032 220 201
 17220 - 033 209 066 205 169 068 033 240 066 030 031 205 218 068 254 118
 17236 - 040 019 254 192 040 015 254 255 040 241 229 205 017 068 225 215
 17252 - 119 035 029 032 230 062 255 119 195 019 067 033 222 066 205 169
 17268 - 068 033 174 064 030 006 024 211 033 194 066 205 169 068 205 218
 17284 - 068 254 255 040 249 197 006 015 144 050 169 064 193 195 019 067
 17300 - 205 042 010 205 194 068 245 175 050 170 064 241 205 035 015 033
 17316 - 192 066 205 169 068 205 218 068 254 192 202 092 068 254 118 040
 17332 - 223 254 119 040 037 254 117 040 038 254 218 040 039 254 222 040
 17348 - 040 254 223 040 041 254 114 040 042 254 113 040 043 254 219 202
 17364 - 003 069 205 017 068 205 046 068 024 203 033 240 066 024 028 033
 17380 - 171 064 024 023 033 181 064 024 018 033 191 064 024 013 033 200
 17396 - 064 024 008 033 215 064 024 003 033 232 064 126 254 255 202 169
 17412 - 067 229 205 017 068 205 046 068 225 035 195 255 067 033 097 065
 17428 - 087 122 190 040 019 035 126 254 254 040 007 254 255 032 246 035
 17444 - 024 239 033 055 066 062 015 201 126 201 087 035 229 230 063 215
 17460 - 058 168 064 254 031 204 194 068 060 050 168 064 225 126 254 255
 17476 - 040 016 254 000 040 012 087 205 129 068 022 001 205 098 068 035
 17492 - 024 235 022 002 205 098 068 201 205 043 015 195 019 067 229 058
 17508 - 169 064 079 033 006 000 006 255 016 254 006 250 016 254 043 124
 17524 - 181 032 243 013 032 237 021 040 002 024 228 225 201 229 053 169
 17540 - 064 079 033 006 000 211 255 006 255 016 254 213 205 070 015 209
 17556 - 048 000 006 250 016 254 043 124 181 032 234 013 032 226 021 040
 17572 - 002 024 219 225 201 205 176 068 205 194 068 201 126 254 255 200
 17588 - 254 254 040 004 215 035 024 244 035 205 194 068 024 238 229 197
 17604 - 245 205 014 012 006 005 014 000 205 245 008 241 193 225 175 050
 17620 - 168 064 050 167 064 201 205 235 068 254 255 040 249 245 205 235
 17636 - 068 254 255 032 249 241 201 229 213 197 205 187 002 062 255 189
 17652 - 040 006 068 077 205 189 007 126 193 209 225 201 205 042 010 205
 17668 - 194 068 205 043 015 245 175 050 170 064 241 033 190 066 205 176
 17684 - 068 205 194 068 205 046 069 205 218 068 254 192 202 019 067 254
 17700 - 118 040 217 254 221 202 151 067 024 231 033 064 000 017 000 032
 17716 - 001 000 000 205 246 069 218 010 070 205 102 069 038 000 205 246
 17732 - 069 048 246 183 124 023 189 056 245 205 145 069 205 188 069 205
 17748 - 246 069 048 229 183 125 031 133 061 188 048 243 205 204 069 195
 17764 - 010 070 122 023 061 188 048 013 084 108 030 001 120 007 246 001
 17780 - 071 121 007 079 201 124 023 061 186 048 012 084 030 000 121 007
 17796 - 246 001 079 120 007 071 201 123 031 056 221 024 238 121 007 128
 17812 - 001 000 000 229 033 247 064 190 040 010 245 126 183 040 009 241
 17828 - 035 035 024 243 035 126 225 201 241 225 062 023 201 197 001 003
 17844 - 000 013 120 177 032 251 193 201 230 063 215 058 168 064 254 031
 17860 - 204 194 068 060 050 168 064 201 058 168 064 254 255 056 001 201
 17876 - 062 000 205 188 069 201 197 036 006 040 014 000 219 254 031 023
 17892 - 023 048 001 012 205 177 069 016 243 062 015 145 056 233 193 201
 17908 - 207 013 229 038 000 205 218 069 124 254 003 056 003 051 051 201
 17924 - 124 225 132 103 055 201 205 235 068 254 255 192 195 055 069 201
 17940 - 201 195 060 122 118 000 002 039 022 234 118 118 238 130 053 055
 17956 - 057 000 000 000 028 028 118 130 054 058 046 040 048 000 049 041
 17972 - 000 038 026 016 053 055 057 017 118 053 058 056 045 000 038 043

En el próximo número continuaremos con la publicación del programa completo en Assembler.



Número 3 (1985)



Las revistas de cierta complejidad gráfica y editorial como K64 lo era se diagraman con meses de anticipación, típicamente dos meses, en ocasiones incluso hasta cuatro meses.

Cuando envié el programa original había remitido el artículo y el código en Assembler; los editores no quisieron saber nada con publicar en Assembler porque creían que sus lectores iban a salir espantados. Por mi parte los traté de convencer, sin mucho éxito, que el tipo de programas que les podía aportar y el público al que probablemente le interesara seguramente lo apreciara. Los años dieron la razón a mi argumento, muchas personas implementaban el artículo tal como estaba planteado pero muchas mas lo tomaban como inspiración para sus propios proyectos. Pero por entonces o lo hacía como decían los editores (el artículo aún no había salido) o no saldría. Utilicé un método muy en boga en este tipo de máquinas que era listar el programa en lenguaje absoluto el que era luego laboriosamente cargado con un rudimentario editor por los lectores; ni siquiera en hexadecimal, en decimal (¡!).

Los editores tuvieron dos sorpresas al cabo del primer número, por un lado el artículo tuvo una enorme repercusión favorable que los lectores hacían llegar por correo y llamadas telefónicas (no existía el e-mail por entonces); la otra era preguntar a quien se le había ocurrido la mala idea de publicar semejante programa con el ridículo sistema de los numeritos....

Me pidieron entonces que transformara el artículo en una serie y muy apuradamente publicaron el listado que había remitido originalmente en Assembler en el número 3, sin texto porque yo no había escrito nada.

Nunca más los editores hicieron objeción alguna ni a los temas, ni a la forma de presentarlos en todo el resto de mis colaboraciones.

El grueso de la serie continuó siendo para la máquina mas avanzada Timex Sinclair 2068.



El número 4 fue el primer artículo realmente pensado como tal de lo que terminó siendo la serie; fue solo de software porque en los plazos que tenía no se podía armar nada que incluyera hardware (que lleva en preparación y edición mucho más).

El artículo abordaba un tema muy “hot” por entonces y era como copiar el software que venía con protección de copyright. La única forma era copiar los cassettes a nivel de audio, pero eso tenía problemas de ecualización y no siempre quedaban bien.

Conocer la ROM (el sistema operativo de la máquina) tenía sus ventajas porque podías utilizar las mismas rutinas que se usaban para leer o grabar solo que en un “loopback” que no permitiera la ejecución del software para que no activara el copyright.

Había alguna discusión sobre si esto era legal o no, que se terminó definiendo al saber que en realidad el software que se copiaba de esta forma era a su vez “pirata”.

“El que roba a un ladrón.....”

El dibujo es uno de los mas ingeniosos y divertidos de toda la serie.

PROGRAMA

PARA DUPLICAR SOFTWARE

Debido al efecto negativo que el primer caso representa sobre los que se dedican a la creación de software, es que el mismo viene provisto de "trabas" que tienden a dificultar, o impedir, su duplicación. Esta modalidad, empero, tiende a dificultar o impedir la generación de copias ante la eventualidad que los modos que contienen los programas, normalmente cassette, resulten dañados con el consiguiente trastorno asociado. El presente, íntegramente sincronizado en lenguaje de máquina, permite realizar una lectura desde cassette almacenando en memoria el programa o bloque de datos para a continuación volcarlo nuevamente a cassette, permitiendo de esa manera la realización de duplicados. El programa, cuyo listado aparece en la Figura 1, ocupa en su totalidad una memoria de 600 bytes de memoria permanente almacenada en memoria segmentada cuya longitud sea de hasta 34 K, más que suficiente para la mayoría de las aplicaciones. El ingreso del mismo se debe llevar a cabo con el auxilio de un Programa Emulador, siendo en este caso utilizado el denominado ZEUS-ASSEMBLER para TS2068 aunque no es difícil su adaptación para utilizar cualquier otro. Cabe destacar, que debido al intensivo uso de rutinas del ROM del computador el mismo no funcionará en máquinas SPECTRUM por diferencias en las direcciones de tales rutinas. El programa comienza definiendo los mensajes denominados MS1, MS2, MS3, MS4, MS5 y MS6, los cuales son textos que se utilizarán en distintas etapas para orientar al usuario en los pasos a seguir, así como para exhibir títulos, mensajes, etc. A continuación se define un área de memoria destinada a albergar el header del programa a copiar, tal como reside en el cinta de audio. Al respecto, es válido indicar brevemente el formato en el cual se



grabar los programas en este computador. Cuando se da la orden de comenzar la grabación la rutina encargada de esta tarea emite un protocolo de sincronismo compuesto por 4032 ciclos a una frecuencia de 806.5 Hz. Este tiene el propósito de permitir que posteriormente la rutina de lectura se "sincronice" con los datos que vienen a continuación. Continúa el sincronismo un bloque de 17 bytes de longitud el cual recibe la denominación de header que contiene la siguiente información: Tipo de Datos (1 byte) OO Programa BASIC O1 Vector/Matriz Numérica O2 Vector/Matriz Alfanumérica

O3 Bloque de Memoria (Bytes) Nombre del Bloque (10 bytes) Longitud del Bloque en Bytes (2 bytes) Dirección/Línea de Comienzo (2 bytes) Datos auxiliares (2 bytes) Al finalizar el bloque se encuentra un código (CRC) con el cual el computador puede reconocer a la información que ha leído es correcta, este código está formado mediante la realización de la operación OR-Exclusiva entre todos los bytes del bloque. Luego del header, se produce un ruido de 835 mSeg y se graba el bloque de programa/datos que corresponde, en este caso el sincronismo tiene la misma función pero sólo se prolonga durante 1612 cíc-

los. Este formato es válido para computadores TS2068 y SPECTRUM únicamente. El Timex-SINCLAIR 2068 posee dos integradas de ROM a diferencia del SPECTRUM que solamente posee una. El primero de ellas, recibe el nombre de BASIC y que contiene las rutinas de LOAD/SAVE y ciertas rutinas especiales para el manejo de los modos de vi-

deo que dispone el TS 2068. La particularidad es que ambos, que suman 24 K, solamente ocupan 16 K del mapa de memoria, esto es están superpuestos. A su vez, cualquier "cartridge" que se coloque en el "buscón" dentro del computador, comparte el espacio con el ROM principal. Por lo tanto cada vez que se ejecuta una rutina que está dentro del ROM extendido o del cartridge es necesario realizar previamente un proceso de habilitación especial, debiendo obligatoriamente realizar el caso inverso cuando se requiere continuar procesando con el ROM normal. Todas las operaciones de Entrada/Salida que realiza el TS2068 (exceptuando lectura/grabación en

cassette) se manejan a través de abstracciones de software denominadas canales, esto es dichas operaciones son realizadas con independencia de cuál es el elemento de ES a ser utilizado, accediendo a una tabla residente en memoria RAM para determinar la rutina de bajo nivel que en definitiva realizará la operación de lectura/grabación concreta. Esta metodología puede lucir confusa a inestables, pero totalmente al contrario es extremadamente poderosa, pues permite que un dispositivo no estándar, originalmente no previsto en el ROM, pueda ser adicionado con total independencia mediante el simple expediente de abrir un canal con OPEN en asignario al dispositivo y asegurarse que en la ta-

Ing. Pedro E. Colla

La duplicación de software puede ser utilizada básicamente con dos fines, uno de ellos legítimo, la copia de programas para evitar su adquisición, y el segundo no sólo legítimo sino también necesario, la creación de duplicados de seguridad.



PROGRAMA

bla respectiva se indica claramente donde debe ir el computador a buscar la "rutina" que le permite utilizar ese dispositivo. En el ROM original solamente existen especificaciones para el teclado, la pantalla de video normal (20 líneas superiores) y la pantalla de video del sistema (líneas inferiores) siendo sus números estándar de canal los siguientes: Canal 1 - Teclado Canal 2 - Pantalla Normal Canal 3 - Impresor Térmico Se permiten hasta 15 canales diferentes, debiendo en todos los casos agregar cuando no se trate de un dispositivo estándar la rutina que realice el manejo del mismo, pues como es lógico no figura en el ROM. El programa comienza indicando que las salidas las realizará a través del canal 2, esto se lleva a cabo mediante el llamado a la rutina de ROM normal que comienza en #1200. A continuación se borra la pantalla recurriendo nuevamente a otra rutina contenida en dicha sección (F00EA). Se coloca el cursor en la posición 0,0 (extremo superior izquierdo) mediante la llamada a la rutina que comienza en la dirección #0562 en el acumulador B conteniendo el número de líneas y el C número de columna. A continuación, se exhiben dos de los mensajes de programa, lo que se realiza con la rutina MS00, la cual toma la cantidad indicada de caracteres y se copia a través de la sección del ROM para salidas de canal 2 normalizada, de allí que se seleccionara previamente el canal 2, esto es la pantalla. Las rutinas LHDR, LPHG, SHDR y SPFG son muy similares entre sí y básicamente colocan en el registro HL la dirección dentro del ROM extendido de la rutina que se utilizará (LOAD o SAVE según el caso) en el registro índice IX se coloca el comienzo del área de memoria de donde se tomarán/sacarán los datos (INDR o SINDR) dependiendo de si es un Header o datos) y en el registro DE la longitud de los mismos. Nótese que cuando se lee o graba los datos en la información de longitud se extrae del área de Header. La rutina HRT realiza la conmutación entre ROM normal y ROM extendido, siendo el método expues-

to estándar para ello, mientras que la rutina EXIT realiza el procedimiento inverso. La secuencia de ejecución es leer el header del programa, imprimir el nombre, tipo y longitud del mismo y a continuación leer el programa en el mismo a partir de la dirección 00000. Una vez finalizada la carga, se espera la indicación adecuada y se vuelcan nuevamente tanto header como datos/programa tantas veces como se desee. Para finalizar la ejecución bastará tener optimizada la tecla de break. En la rutina que exhibe las características del programa a duplicar se tiene en cuenta el formato y posibles contenidos del header del mismo, exhibiéndose mensajes apropiados. Nótese que se utiliza una rutina de conversión del ROM para trasladar la longitud excesiva en binario a caracteres ASCII. En la rutina de retraso (DLY) simplemente se "pierde tiempo" haciendo contar al computador hasta la 85500, esto no obstante tarda solamente algunos miles de milisegundos. Como técnica de programación debe apuntarse que no se realiza el loop en la forma normal (RET Z o JRF Z) luego de decrementar BCI, pues esta operación NO modifica el FLAG Z (flag que indica que la última operación matemática dio resultado 0). En la rutina de espera (WAIT) simplemente se chequea el teclado hasta que se detecta que hay cualquier tecla optimizada. Para hacer esta misma tarea se podría utilizar el método normal de hacer abrir el canal 1, leerlo y determinar si hay o no información proveniente del mismo. Esto es molestoso por la necesidad de abrir el canal 1 para luego al finalizar, abrir nuevamente el canal 2 para cualquier operación de escritura sobre la pantalla. En cambio se aprovecha el hecho de que, independientemente de lo que haga el computador, "se" cada 16.5 mSeg el teclado y coloca el carácter correspondiente a la tecla optimizada en la posición 23560 (LAST-KEY), por lo que basta colocar 0 en esta y luego darle tiempo para que por sí sólo se encargue de colocar allí algún valor, si lo hubiere. En el caso típico de que el programa a copiar esté compuesto de varios segmentos, se deberá realizar la duplicación de cada segmento en forma independiente.

03 Bloque de Memoria (Bytes) Nombre del Bloque (10 bytes) Longitud del Bloque en Bytes (2 bytes) Dirección/Línea de Comienzo (2 bytes) Datos auxiliares (2 bytes) Al finalizar el bloque se encuentra un código (CRC) con el cual el computador puede reconocer a la información que ha leído es correcta, este código está formado mediante la realización de la operación OR-Exclusiva entre todos los bytes del bloque. Luego del header, se produce un ruido de 835 mSeg y se graba el bloque de programa/datos que corresponde, en este caso el sincronismo tiene la misma función pero sólo se prolonga durante 1612 cíc-

Table with columns: Address, Instruction, Comment, Address, Instruction, Comment. Contains assembly code for ROM duplication.





El número 6 fue explicar un accesorio bastante misterioso que tenía la TS2068, una bahía para enchufar cartridges. Originalmente pensada para posicionar a la máquina como una consola de juegos era para poder poner cartuchos de juegos sin necesidad de utilizar a la máquina como computadora. La estrategia falló pues nunca hubo una oferta de juegos significativa y el bay tenía poco y nada de uso.

Sin embargo al estudiar la ROM era claro que se podía poner cualquier programa en una EPROM y hacerlo ejecutar desde allí sin necesidad de estar lidiando con cassetes. Una segunda ventaja era que si había un cartucho presente era ejecutado con prioridad al boot normal de la máquina. Por eso si uno quería tener algo que no fuera atendido era la solución ideal.

Mas importante aún, dado que el espacio de direccionamiento de memoria era tan reducido (64K) la arquitectura hacía que si se detectaba una EPROM en la bahía esta reemplazaba directamente a la ROM de la máquina (en realidad quedaba disponible con un mecanismo bastante bizarro de "bancos").

Quiere decir que si se le ponía en la EPROM la ROM de otra máquina con otro sistema operativo reemplazaba al original. Esto es poco útil porque la arquitectura misma de la TS2068 era propietaria pero al mismo tiempo era ligeramente diferente de la ZX Spectrum para la cual había mas juegos, más ofertas, mas accesorios y había sido en general mas exitosa.

Pero para que todo eso ocurriera había que saber como conectar la EPROM en la bahía, cosa que este artículo explicaba.

CONVERSION DE LA TS 2068

CONOCIENDO EL BUZON

ING. PEDRO E. COLLA

En general, el TS2068 soporta en su conector para cartidge dispositivos exteriores de prácticamente cualquier tipo, pues están allí las principales líneas del bus. Ello no es contradictorio con que primariamente el mismo este orientado a la utilización de software en ROM (memoria de lectura solamente). La memoria del computador tiene una distribución tal como se especifica en la Figura 1. Nótese que hay una zona en la cual se superponen la memoria ROM principal y la llamada de "Extensión". En realidad no es que ambos bancos puedan funcionar o estar visibles al mismo tiempo, sino que se realiza la comunicación entre ambos en la medida que se quiera ejecutar lo que contiene uno u el otro. En una anterior entrega en esta publicación se dio un ejemplo de tal situación.

Independientemente del contenido original, la memoria se divide en secciones cuyo tamaño es de 8 kbytes denominadas "cuartiles" o "chunks", teniendo el hardware la posibilidad de eliminar mediante instrucciones de software adecuadas una o más de esas zonas. Esto facilita enormemente el agregado externo de software en ROM que reemplaza parte o la totalidad de la ROM del computador, sin tener que recurrir a engorrosos artificios circuitales, como era necesario hacer por ejemplo en el computador TS1000.

En realidad, las cosas son más fáciles aun pues el computador como parte de su rutina de encendido (cuando la pantalla se vuelve negra) realiza por sí mismo una lectura en el conector de cartidge, y si detecta que allí hay algo, toma



los recaudos para utilizarlo y que el resto del computador no interfiera con ello. Se prevé que puedan existir dos tipos de software en cartidge, lo que se denomina LROS y su conector de AROS. El primero de ellos (LROS) está pensado para software en lenguaje de máquina que reemplaza al ROM original del computador para proveer al mismo de una modalidad de funcionamiento enteramente diferente. Clases ejemplos de uso de esta alternativa serían hacer que el principal lenguaje fuera FORTH y no BASIC, o aprovechando la existencia del procesador Z80 el lograr que se desactive el modo de funcionamiento del Sinclair para transformarlo en un sistema con diskettes bajo CP/M.

En cambio en el caso del AROS se prevé la utilización de software que necesite del ROM Sinclair, es decir de aplicaciones o juegos. Además en el caso del AROS el contenido del cartidge puede estar tanto en lenguaje de máquina como en BASIC y por cierto no elimina el ROM principal. Físicamente un cartidge es una plaqueta de circuito impreso con uno o más chips de memoria ROM o EPROM. El tipo de memoria a utilizar está fundamentalmente dado por el volumen de producción que se prevé. En el caso de software comercial, las memorias tipo ROM son posiblemente la solución de menor costo mientras que para proyectos esencialmente unitarios, es infinitamente más práctica la utilización de memorias EPROM (memorias de

Una de las características que distinguen al computador Timex-Sinclair 2068 de su similar europeo, el SPECTRUM, es el conector para cartidge, más vulgarmente conocido como buzón. Es posible agregarlo a esta máquina, pero en forma opcional.

Lectura Solamente y Borrables). Las primeras posiciones de la misma contienen la información necesaria para que al realizar el Proceso de inicialización el computador pueda conocer si el contenido de la misma es de tipo LROS o AROS y cuáles son los chunks de memoria que resultan afectados. En el caso del LROS, que es el que nos interesa, son los primeros 5 bytes los que contienen tal información y su estructura es la siguiente:

0000 No es utilizada
0001 Contiene el valor 01 indicando LROS.
0002/0003 Dirección de primera instrucción a ejecutar
0004 Especificación de chunks a anular.
Nótese que si no hay cartidge conectado el computador, "ve" solamente direcciones conteniendo el valor 255 y por lo tanto al leer la posición 0001 se da por notificado que no hay ningún LROS para ejecutar y tampoco ningún AROS

cuyo valor para posición 0001 es 02) y continúa operando normalmente. Una vez que el computador se notifica que existe una sección de LROS, utiliza el quinto byte para saber que secciones de memoria debe reemplazar con el. Las posiciones segunda y tercera le dan la información acerca de a que dirección de memoria se deberá transferir la ejecución del programa luego de haber finalizado la inicialización.

FIGURA 1 MAPA DE MEMORIA DEL COMPUTADOR TS2068

DIRECCIONES	CONTENIDO	CHUNK
00000 A 06191	ROM PRINCIPAL	CHUNK 0
06192 A 16383	ROM EXTEND	CHUNK 1
16384 A 24575		CHUNK 2
24576 A 32767		CHUNK 3
32768 A 40959		CHUNK 4
40960 A 49151		CHUNK 5
49152 A 57313	RAM	CHUNK 6
57314 A 65535	ROM PRINCIPAL	CHUNK 7

Figura 2 Conector de Cartidge en TS2068

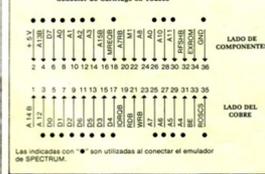
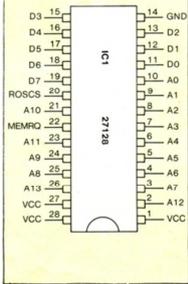


Figura 3 Conexiones del EPROM al conector del Cartidge



CONVERSION DE LA TS 2068

La especificación de que chunks son anulados y cuáles no, se logra numerando cada uno 0 a 7. 6 chunks de 8 kbytes cada uno suman 64 kbytes, la totalidad de memoria, cada bit de esta posición refiere a un chunk específico, siendo el bit 0 el que controla el segmento que va entre las direcciones 0 y 8192 y el bit 2 que transcurra entre 8192 y 16383 y así sucesivamente. De tal manera que si el bit asociado tiene un valor "1", el chunk correspondiente no debe ser anulado, mientras que si su valor es "0" se debe proceder a la desconexión del mismo. Para clarificar esto con un ejemplo suponemos que queremos reemplazar con el contenido de un cartidge la totalidad del ROM del computador, el que vimos en Fig. 1 que se extiende entre las posiciones 0 y 16383. Los primeros cinco bytes del cartidge deberán entonces contener:

```
0000 000
0001 001
0002 000
0003 019
0004 252 1111 1100
```

El interesante apuntar que la mayoría del software diseñado para computadores SPECTRUM no es compatible para su ejecución en máquinas TS2068, y por lo tanto aprovechando la similitud del hardware entre ambas es factible hacer que el computador TS2068 "emule" un SPECTRUM colocando el ROM de este último en reemplazo del original del primero.

Con este cambio el TS2068 sin duda pierde potencia, pues no puede utilizar en forma sencilla los joysticks, el sintetizador musical y algunas instrucciones especiales del BASIC (ON, ERR, STICK, RESET, SOUND, etc). Pero se gana en el acceso al constantemente repro-

duciendo software para SPECTRUM que en calidad y cantidad supera el disponible para máquinas TS2068. Para lograr ello deberemos reproducir en una EPROM el contenido del ROM de un computador SPECTRUM cualquiera y modificar las posiciones de memoria necesarias para cumplir los requisitos de un LROS. Tendremos entonces que concurrir a una de las numerosas casas especializadas en grabación de EPROM portando el chip "vacío" junto con una ROM de cualquier computador SPECTRUM, posiblemente prestada por un amigo, para que se obtenga un "duplicado" de la misma. Normalmente con un bajo costo adicional es posible que antes de asentar la copia se realicen las modificaciones que se consignan en la presente, para que dicho chip pueda funcionar como LROS.

En las Figuras 2 y 3 se brindan el esquema de conexiones del circuito integrado y el diagrama del zócalo de cartidge del computador. Nótese que debido a la extrema sencillez del conector, no se justifica el armado de una plaqueta de circuito impreso, siendo posible el armado del mismo con técnicas de soldado punto a punto o "wire-wrapping".

Como integrado EPROM se utiliza un 27128 cuya capacidad es de 128 kbytes X 8 bits, por lo que la totalidad del ROM SPECTRUM puede ser albergada en un solo integrado. En caso de dificultades en la obtención del citado elemento electrónico, nada impide lograr el mismo propósito mediante dos chips 2764 (8 kbytes X 8 bits) o cuatro chips 2732 (4 kbytes X 8 bits) de más fácil obtención, aunque en ambos casos se complicará ostensiblemente el aspecto mecánico del montaje.

Para que el integrado grabado funcione correctamente, debe tenerse en cuenta que las primeras posiciones no podrán coincidir exactamente con el ROM SPECTRUM, pues deben contener los datos necesarios para que durante la inicialización se disponga de la información para utilizar correctamente. Para ello se aprovecha un hueco de algunos bytes en la posición 19 del ROM SPECTRUM para transferir allí las instrucciones ubicadas originalmente en las primeras direcciones. Por ello se debe grabar en forma diferente las primeras 30 posiciones sobre el integrado de EPROM, siendo el resultado final de las mismas el expresado por la Figura 4. El resto de las posiciones son un fiel reflejo de las contenidas en el ROM SPECTRUM. Al encender el computador con el cartidge ocupado por el circuito que nos ocupa, y si hemos hecho bien las cosas, el computador debería "inicializarse" como lo hace normalmente, llegando a exhibir durante algo menos de un segundo la leyenda de "copyright" normal para a continuación volver a inicializarse, para aparecer el mensaje de "copyright" del computador SPECTRUM, de allí en adelante a todos los efectos prácticos el computador deberá de comportarse como un TS2068 y será un SPECTRUM. Si esto no ocurre significará que se ha cometido algún error en la grabación de la EPROM o en el conector del cable, para lo cual habrá que revisar cuidadosamente ambos puntos. En ningún caso se puede conectar o desconectar el cartidge con el computador encendido pues el mismo se puede dañar. Exceptuando esta última precaución, es improbable que un error en el circuito produzca daño alguno y mucho menos permanente.

FIGURA 4 POSICIONES INICIALES PARA EMULADOR DE SPECTRUM

DIRECCION	+00	+01	+02	+03	+04	+05	+06	+07	+08	+09
00000	255	001	019	000	252	255	255	255	042	093
00010	092	034	095	092	024	067	195	242	021	017
00020	255	255	024	019	042	093	092	126	205	125

IGUAL A ROM SPECTRUM





El número 6 fue una interfaz para poder imprimir, necesidad básica que hoy se dá por descontado en cualquier máquina que uno compre. Pero en esa época la ZX81 o sus clones no tenían un puerto de impresión (tampoco uno serie dicho sea de paso).

Aparecieron en el mercado impresoras de matriz de puntos muy económicas (Gorila Banana es una que me viena a la mente que salía la cuarta parte de cualquier printer de la época).

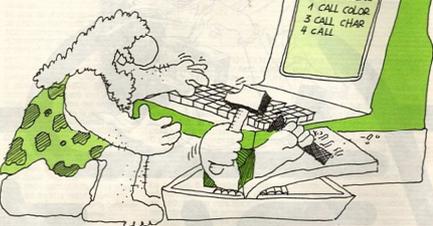
La ZX81 en realidad tenía la opción a una printer, pero era un engendro propio, que funcionaba en papel térmico (carísimo!!) en una hoja un poco mas ancha que la tira de papel de una calculadora.

Esta interfaz, y el software que la acompaño en un número posterior hizo literalmente delirar a muchos entusiastas porque era muy fácil de armar.

DESARROLLOS

INTERFACE PARA IMPRESORA

ING. PEDRO E. COLLA



Al tope de la lista de prioridades de cualquier usuario de un computador personal que desee sacar el máximo de potencia de una herramienta de este tipo, figura sin ninguna duda el poder imprimir. Con el mismo se puede acceder a una gama inagotable de aplicaciones útiles en distintos órdenes y, en especial, en lo que respecta a la palabra (word processor), la planilla electrónica (electronic spreadsheet) y el almacenamiento de datos para emisión de reportes de distinto tipo. El impresor es un dispositivo electrónico que nos permite mantener una salida perdurable para la información sobre una hoja de papel. Existen impresores de distinto tipo, calidad y precio, y estos factores afectan en buena medida el uso al cual lo podremos dedicar. No obstante, se puede separar en principio a los mismos en dos grupos: los que utilizan papel especial y los que utilizan papel común. En el caso de computador TSD08B

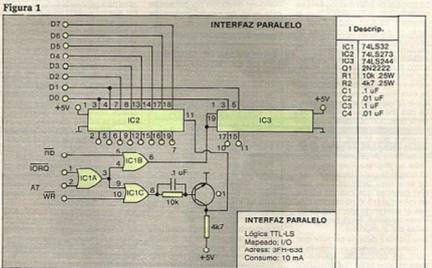
el impresor estándar para el mismo utiliza papel especial, el cual mediante un tratamiento químico especial permite, mediante la aplicación de calor, imprimir los caracteres que forman el texto. Este tipo de impresión, aparte de no ser totalmente económica, no es nada más allá de trabajos esencialmente caseros, debido a la baja calidad del resultado y al formato normalmente angosto del tipo de papel utilizado. Otro tipo de impresor, el que utiliza papel común, es en el que se impone cuando se requiere dar una utilización seria a la salida del computador. Los impresores de esta última categoría pueden ser de varios tipos pero esencialmente se los puede dividir en dos grupos. Los que imprimen mediante matriz de punto y los que utilizan "margarita" (daisy wheel). Estos últimos de superior calidad de impresión, pero notablemente más costosos. Debido a ello se han popularizado los impresores de tipo a matriz de puntos (dot matrix), pues representan un comercio más favorable entre

la prestación que brindan y su costo. Existe una gama casi inagotable de variantes desde el último tipo. Los hay que imprimen en forma sencilla en un extremo, los que toleran distintas densidades de impresión (caracteres por línea) y tipos de letras en el otro. Se han desarrollado la variedad en el tipo de prestación, las formas de manejo se mantienen sin embargo parecidas a través de todo el espectro. Como regla casi general, aquellos que tienen un costo más accesible pueden imprimir hasta 80 caracteres por renglón. El impresor acaba como periférico del computador y debido a que no está previsto como tal, será necesario cuando queramos agregarle utilizar dos elementos: una interfaz de hardware que conecte ambos aparatos electrónicamente y una interfaz de software (driver) que permita al computador el asimilar al nuevo dispositivo en su estructura. En la presente entrega se describirá la parte correspondiente a la



DESARROLLOS

interfaz de hardware, dejando para una posterior publicación su programación de software. Los impresores pueden comunicarse con el computador esencialmente en dos formas: en serie y en paralelo. En el primero de los casos, los 8 bits que componen cada carácter por imprimir son enviados uno detrás del otro a través de una única vía. Como la emisión no es continua el canal de comunicación se coloca en estado alto (marginado) cuando no hay transmisión. Para diferenciar unívocamente el comienzo y finalización de cada carácter se incluyen bits adicionales al comienzo (start bit) y al final del carácter (stop bit). Los bits son emitidos normalmente a razón de 300 por segundo (300 bauds) lo que representaría aproximadamente 30 caracteres en el mismo lapso de tiempo. A los impresores que utilizan este modo se los conoce con el nombre de RS-232C en alusión a la norma que estipula las condiciones de la comunicación. Un segundo tipo, en cambio, permite que todos los bits que forman cada carácter (8) se transmitan al mismo tiempo por igual cantidad de canales de comunicación. En este caso, que es el que nos ocupará, la transmisión reclus la denominación de ser en paralelo y los impresores son llamados tipo CENTRONICS en alusión a la norma que los rige. Contra lo que se podría suponer, no existen variaciones importantes de velocidad entre ambos tipos, pues este factor en definitiva está limitado por el aspecto mecánico de la impresión el cual si ser



¿Quién tiene los mejores programas en Cassettes para Commodore 64

micro cómputo

ACOYTE 44 - LOCAL 6 CABALLITO

DESARROLLOS

mucho más lento que el electrónico el que determina la velocidad final. El protocolo o forma en que se comunican el impresor y el computador es sencillo y se lleva a cabo mediante como mínimo 10 líneas, 8 de ellas destinadas a los datos y dos a control. Las líneas de datos reciben la denominación de D0 a D7 y tienen exacta correspondencia con las del bus del computador denominada D0-D7. Las líneas de control esenciales a la comunicación reciben el nombre de línea BUSY y línea STROBE. La primera de ellas cuando está en un valor alto (+5V) indica que el impresor está ocupado y por lo tanto no está en condiciones de recibir un nuevo carácter porque lo perdería sin poder procesarlo. El estado bajo (0V) de la misma indica como complemento que el impresor está listo (READY) para imprimir un nuevo carácter. Obviamente, esta línea permite que el impresor se "despierte" al computador. La otra línea de control (STROBE) tiene dirección contraria y permite que el computador le indique al impresor que lo que en ese momento está presente en las líneas de datos es un carácter válido que debe ser impreso. Esta indicación es aceptada por el impresor cuando esta línea tiene un valor bajo (0V) mientras que cuando asume un valor alto (+5V), le importa cual sea el contenido eléctrico de las líneas de datos, éstas serán ignoradas por el impresor. Una interfaz debería permitir que la impresora quede conectada al BUS del computador solamente cuando tiene lugar una operación de entrada o salida, haciendo que en todo otro momento tal conexión no exista, por lo tanto, no interfiera con el normal desenvolvimiento de las operaciones en el mismo. Para ello se utiliza la facilidad que permite el microprocesador Z80 de manejar hasta 256 dispositivos de entrada y salida. Cada dispositivo responderá únicamente cuando su dirección ó número del dispositivo, el cual puede estar comprendida entre 0 y 255, es la que aparece sobre el BUS y no en cualquier otro momento. El manejo de esta facilidad puede realizarse tanto desde BASIC como desde lenguaje de máquina

mediante las instrucciones INOUT. En principio, la dirección que se utiliza puede ser arbitrariamente elegida entre las posibles, pero en este caso en particular se debe tener cuidado de no usar alguna que ya esté utilizando el computador para sus propios fines. En el TSD08B, el teclado, el generador de sonidos y ciertos dispositivos de control utilizan esta facilidad, por lo que la elección no es totalmente arbitraria. No obstante, y afortunadamente, todas las direcciones utilizadas tienen la particularidad de ser superiores a la número 128, por lo que utilizando cualquier dirección por debajo de este valor, no existirá peligro de interferencia. **Figura 2**

```
PROGRAMA DE PRUEBA DE LA INTERFAZ
10 REM
20 REM TEST
30 REM PRUEBA DE LA INTERFAZ PARALELO
40 REM IMPRESOR CENTRONICS
50 REM
60 FOR I = 1 TO 1000
70 FOR J = 32 TO 127
80 IF IN 63 = 255 THEN GOTO 90
90 OUT 63, J
100 NEXT J
110 NEXT I
120 NEXT I
```

Para nuestro propósito y como una forma de asegurarnos un circuito sencillo, nos aseguraremos únicamente que el impresor responda a cualquier dirección por debajo de la número 128, en todas las operaciones utilizaremos por convención la número 63 (D7H), pero debe quedar claro que cualquier otro causal el mismo efecto. Este rango de direcciones cumple con la condición que su bit más significativo (A7) es nulo. Por lo que desde un punto de vista circuital bastará asegurarse esta condición junto con la indicación que se trata de una operación de entrada-salida (línea I/OH en estado bajo) y de que tipo es (lectura o grabación) para tener un interfaz adecuado. El circuito respectivo es expuesto en la Fig. 1. En la misma se utiliza integrado 74LS32 (IC1), el cual contiene cuatro compuertas OR; para decidir si se trata de una operación de entrada-salida, si ésta tiene la línea AT baja y cuál es el sentido de dicha operación. La misma habilita en el caso de que sea una lectura a IC3 el cual es un buffer de tres estados que es utilizado para la lectura de la línea de BUSY previamente descrita. El integrado IC2 es un octuple Flip-Flop, tipo D, el cual cuando está habilitado toma el contenido de las líneas de datos del bus de la computadora (D0 a D7) y coloca el mismo sobre las correspondientes líneas del impresor. La misma línea para generar la señal de STROBE. Nótese que C1 sirve para invertir la señal únicamente, pues para la habilitación del impresor requiere un "1" lógico (0 V). Debido a la poca cantidad de componentes el circuito puede ser armado sobre una plaqueta experimental mediante técnicas de wire-wrapping o por soldado punto a punto. La alimentación del circuito puede realizarse sin problemas desde la misma computadora dado el bajo consumo del mismo. La conexión debe ser realizada utilizando el conector de expansión que el computador posee en su parte trasera mediante el uso de un conector apropiado. El enlace con la impresora se lleva a cabo mediante un trozo de cable plano de longitud adecuada. Para unir el cable con la impresora se debe utilizar también un conector apropiado, evitando el soldado directo al impresor, para mantener la posibilidad de desconexión y transporte del mismo. Debe tenerse especial cuidado en unir todas las masas del impresor (una general y una por cada línea de datos) entre sí y a su vez éstas con la interfaz y el computador si se quiere que el funcionamiento sea correcto, este detalle puede ocasionar que la interfaz no funcione en absoluto. En la próxima entrega se brindarán los elementos de software necesarios para integrar completamente computadora, interfaz e impresor.



OCTUBRE 1985 N° 7 P. ARGENTINA

K64

COMPUTACION PARA TODOS

Educativo Para TI 99 4/A

TS 2068:

Software Para Impresora

Commodore:

Base de Datos

K64 en Europa

2do. Concurso:

Ya Están Los Ganadores



INTERFASE PARA IMPRESORA

SEGUNDA PARTE

ING. PEDRO E. COLLA

En la Primera Parte de este artículo fue expuesta la porción de hardware de una interfaz Centronics para conectar un computador TS2068 con cualquier impresora que responda a la citada norma.

El circuito, si bien esencial, constituye únicamente una fracción de la interfase, pues, por sí solo el computador no puede detectar su presencia y mucho menos utilizarlo.

Es necesario incluir una porción de software, a menudo denominado DRIVER que controla las instrucciones necesarias para manejar adecuadamente el intercambio de información con el impresor.

En artículos anteriores hemos tenido oportunidad de describir la forma en la cual el computador TS 2068 se comunica a través de los pines de teclado, pantalla e impresor terminal a través de circuitos lógicos denominados CANALES.

En realidad no es totalmente imprescindible el relacionar íntimamente con la arquitectura del computador para utilizar al impresor, puesto que hasta un sencillo programa como el indicado en la figura 1 podrá ser suficiente para tal cometido.

No obstante sería ideal que se pudiera utilizar cualquier programa destinado a disponer del impresor terminal con uso tipo Centronics sin introducir modificaciones en el mismo.

Y desde el punto de vista del uso diario resulta relevante el poder acceder a las instrucciones convencionales LPRINT, LIST sin preocuparse en el detalle de la impresión individual de cada carácter.

Para ello será necesario introducir al driver de tal manera que éste sea transparente a cualquier programa BASIC, e incluirse al computador que se lo desea utilizar en lugar de la sección del ROM destinada al impresor terminal cada vez que se utiliza alguna de las instrucciones mencionadas.

La primera de las condiciones se logra mediante el uso de lenguaje Assembler para la confección del driver, colocando el mismo en un área de memoria fuera de la incum-

bencia del intérprete BASIC. La segunda, se logra modificando la vía por la que procede en condiciones estándar el computador al utilizar el impresor terminal.

El impresor terminal es manejado a través del canal número 3 de Entrada/Salida (el 1 es el teclado y el 2 la pantalla de video). Cuando el intérprete desea transferir algo a

través de un canal accede a una zona de las variables de sistema denominada STRMS cuya dirección de comienzo es 23568d (5C10h), esta zona contiene una tabla que le indica para cada canal dónde comienza la descripción del mismo dentro del área de variables denominada CHANS, la cual comienza en 23614d (5C4Fh).

Figura 1

```

10 REM -----
20 REM *
30 REM * Este Programa Puede ser
40 REM * utilizado como rutina
50 REM * de Impresion Para la
60 REM * interfaz Centronics.
80 REM * El caracter a imprimir
90 REM * debe ingresar en la
100 REM * de la variable Y#
110 REM *
120 REM *
130 IF IN 63=255 THEN GO TO 130
140 OUT 63, CODE (Y#)
150 RETURN
    
```

Figura 3

```

1 REM -----
2 REM *
3 REM * Esta rutina carga y habilita al
4 REM * driver de Impresoras Centronics
5 REM *
6 REM *
7 REM *
8 REM *
9 REM *
10 REM *
11 REM *
12 REM *
13 REM *
14 REM *
15 REM *
16 REM *
17 REM *
18 REM *
19 REM *
20 REM *
21 REM *
22 REM *
23 REM *
24 REM *
25 REM *
26 REM *
27 REM *
28 REM *
29 REM *
30 REM *
31 REM *
32 REM *
33 REM *
34 REM *
35 REM *
36 REM *
37 REM *
38 REM *
39 REM *
40 REM *
41 REM *
42 REM *
43 REM *
44 REM *
45 REM *
46 REM *
47 REM *
48 REM *
49 REM *
50 REM *
51 REM *
52 REM *
53 REM *
54 REM *
55 REM *
56 REM *
57 REM *
58 REM *
59 REM *
60 REM *
61 REM *
62 REM *
63 REM *
64 REM *
65 REM *
66 REM *
67 REM *
68 REM *
69 REM *
70 REM *
71 REM *
72 REM *
73 REM *
74 REM *
75 REM *
76 REM *
77 REM *
78 REM *
79 REM *
80 REM *
81 REM *
82 REM *
83 REM *
84 REM *
85 REM *
86 REM *
87 REM *
88 REM *
89 REM *
90 REM *
91 REM *
92 REM *
93 REM *
94 REM *
95 REM *
96 REM *
97 REM *
98 REM *
99 REM *
100 REM *
101 REM *
102 REM *
103 REM *
104 REM *
105 REM *
106 REM *
107 REM *
108 REM *
109 REM *
110 REM *
111 REM *
112 REM *
113 REM *
114 REM *
115 REM *
116 REM *
117 REM *
118 REM *
119 REM *
120 REM *
121 REM *
122 REM *
123 REM *
124 REM *
125 REM *
126 REM *
127 REM *
128 REM *
129 REM *
130 REM *
131 REM *
132 REM *
133 REM *
134 REM *
135 REM *
136 REM *
137 REM *
138 REM *
139 REM *
140 REM *
141 REM *
142 REM *
143 REM *
144 REM *
145 REM *
146 REM *
147 REM *
148 REM *
149 REM *
150 REM *
151 REM *
152 REM *
153 REM *
154 REM *
155 REM *
156 REM *
157 REM *
158 REM *
159 REM *
160 REM *
161 REM *
162 REM *
163 REM *
164 REM *
165 REM *
166 REM *
167 REM *
168 REM *
169 REM *
170 REM *
171 REM *
172 REM *
173 REM *
174 REM *
175 REM *
176 REM *
177 REM *
178 REM *
179 REM *
180 REM *
181 REM *
182 REM *
183 REM *
184 REM *
185 REM *
186 REM *
187 REM *
188 REM *
189 REM *
190 REM *
191 REM *
192 REM *
193 REM *
194 REM *
195 REM *
196 REM *
197 REM *
198 REM *
199 REM *
200 REM *
201 REM *
202 REM *
203 REM *
204 REM *
205 REM *
206 REM *
207 REM *
208 REM *
209 REM *
210 REM *
211 REM *
212 REM *
213 REM *
214 REM *
215 REM *
216 REM *
217 REM *
218 REM *
219 REM *
220 REM *
221 REM *
222 REM *
223 REM *
224 REM *
225 REM *
226 REM *
227 REM *
228 REM *
229 REM *
230 REM *
231 REM *
232 REM *
233 REM *
234 REM *
235 REM *
236 REM *
237 REM *
238 REM *
239 REM *
240 REM *
241 REM *
242 REM *
243 REM *
244 REM *
245 REM *
246 REM *
247 REM *
248 REM *
249 REM *
250 REM *
251 REM *
252 REM *
253 REM *
254 REM *
255 REM *
256 REM *
257 REM *
258 REM *
259 REM *
260 REM *
261 REM *
262 REM *
263 REM *
264 REM *
265 REM *
266 REM *
267 REM *
268 REM *
269 REM *
270 REM *
271 REM *
272 REM *
273 REM *
274 REM *
275 REM *
276 REM *
277 REM *
278 REM *
279 REM *
280 REM *
281 REM *
282 REM *
283 REM *
284 REM *
285 REM *
286 REM *
287 REM *
288 REM *
289 REM *
290 REM *
291 REM *
292 REM *
293 REM *
294 REM *
295 REM *
296 REM *
297 REM *
298 REM *
299 REM *
300 REM *
301 REM *
302 REM *
303 REM *
304 REM *
305 REM *
306 REM *
307 REM *
308 REM *
309 REM *
310 REM *
311 REM *
312 REM *
313 REM *
314 REM *
315 REM *
316 REM *
317 REM *
318 REM *
319 REM *
320 REM *
321 REM *
322 REM *
323 REM *
324 REM *
325 REM *
326 REM *
327 REM *
328 REM *
329 REM *
330 REM *
331 REM *
332 REM *
333 REM *
334 REM *
335 REM *
336 REM *
337 REM *
338 REM *
339 REM *
340 REM *
341 REM *
342 REM *
343 REM *
344 REM *
345 REM *
346 REM *
347 REM *
348 REM *
349 REM *
350 REM *
351 REM *
352 REM *
353 REM *
354 REM *
355 REM *
356 REM *
357 REM *
358 REM *
359 REM *
360 REM *
361 REM *
362 REM *
363 REM *
364 REM *
365 REM *
366 REM *
367 REM *
368 REM *
369 REM *
370 REM *
371 REM *
372 REM *
373 REM *
374 REM *
375 REM *
376 REM *
377 REM *
378 REM *
379 REM *
380 REM *
381 REM *
382 REM *
383 REM *
384 REM *
385 REM *
386 REM *
387 REM *
388 REM *
389 REM *
390 REM *
391 REM *
392 REM *
393 REM *
394 REM *
395 REM *
396 REM *
397 REM *
398 REM *
399 REM *
400 REM *
401 REM *
402 REM *
403 REM *
404 REM *
405 REM *
406 REM *
407 REM *
408 REM *
409 REM *
410 REM *
411 REM *
412 REM *
413 REM *
414 REM *
415 REM *
416 REM *
417 REM *
418 REM *
419 REM *
420 REM *
421 REM *
422 REM *
423 REM *
424 REM *
425 REM *
426 REM *
427 REM *
428 REM *
429 REM *
430 REM *
431 REM *
432 REM *
433 REM *
434 REM *
435 REM *
436 REM *
437 REM *
438 REM *
439 REM *
440 REM *
441 REM *
442 REM *
443 REM *
444 REM *
445 REM *
446 REM *
447 REM *
448 REM *
449 REM *
450 REM *
451 REM *
452 REM *
453 REM *
454 REM *
455 REM *
456 REM *
457 REM *
458 REM *
459 REM *
460 REM *
461 REM *
462 REM *
463 REM *
464 REM *
465 REM *
466 REM *
467 REM *
468 REM *
469 REM *
470 REM *
471 REM *
472 REM *
473 REM *
474 REM *
475 REM *
476 REM *
477 REM *
478 REM *
479 REM *
480 REM *
481 REM *
482 REM *
483 REM *
484 REM *
485 REM *
486 REM *
487 REM *
488 REM *
489 REM *
490 REM *
491 REM *
492 REM *
493 REM *
494 REM *
495 REM *
496 REM *
497 REM *
498 REM *
499 REM *
500 REM *
501 REM *
502 REM *
503 REM *
504 REM *
505 REM *
506 REM *
507 REM *
508 REM *
509 REM *
510 REM *
511 REM *
512 REM *
513 REM *
514 REM *
515 REM *
516 REM *
517 REM *
518 REM *
519 REM *
520 REM *
521 REM *
522 REM *
523 REM *
524 REM *
525 REM *
526 REM *
527 REM *
528 REM *
529 REM *
530 REM *
531 REM *
532 REM *
533 REM *
534 REM *
535 REM *
536 REM *
537 REM *
538 REM *
539 REM *
540 REM *
541 REM *
542 REM *
543 REM *
544 REM *
545 REM *
546 REM *
547 REM *
548 REM *
549 REM *
550 REM *
551 REM *
552 REM *
553 REM *
554 REM *
555 REM *
556 REM *
557 REM *
558 REM *
559 REM *
560 REM *
561 REM *
562 REM *
563 REM *
564 REM *
565 REM *
566 REM *
567 REM *
568 REM *
569 REM *
570 REM *
571 REM *
572 REM *
573 REM *
574 REM *
575 REM *
576 REM *
577 REM *
578 REM *
579 REM *
580 REM *
581 REM *
582 REM *
583 REM *
584 REM *
585 REM *
586 REM *
587 REM *
588 REM *
589 REM *
590 REM *
591 REM *
592 REM *
593 REM *
594 REM *
595 REM *
596 REM *
597 REM *
598 REM *
599 REM *
600 REM *
601 REM *
602 REM *
603 REM *
604 REM *
605 REM *
606 REM *
607 REM *
608 REM *
609 REM *
610 REM *
611 REM *
612 REM *
613 REM *
614 REM *
615 REM *
616 REM *
617 REM *
618 REM *
619 REM *
620 REM *
621 REM *
622 REM *
623 REM *
624 REM *
625 REM *
626 REM *
627 REM *
628 REM *
629 REM *
630 REM *
631 REM *
632 REM *
633 REM *
634 REM *
635 REM *
636 REM *
637 REM *
638 REM *
639 REM *
640 REM *
641 REM *
642 REM *
643 REM *
644 REM *
645 REM *
646 REM *
647 REM *
648 REM *
649 REM *
650 REM *
651 REM *
652 REM *
653 REM *
654 REM *
655 REM *
656 REM *
657 REM *
658 REM *
659 REM *
660 REM *
661 REM *
662 REM *
663 REM *
664 REM *
665 REM *
666 REM *
667 REM *
668 REM *
669 REM *
670 REM *
671 REM *
672 REM *
673 REM *
674 REM *
675 REM *
676 REM *
677 REM *
678 REM *
679 REM *
680 REM *
681 REM *
682 REM *
683 REM *
684 REM *
685 REM *
686 REM *
687 REM *
688 REM *
689 REM *
690 REM *
691 REM *
692 REM *
693 REM *
694 REM *
695 REM *
696 REM *
697 REM *
698 REM *
699 REM *
700 REM *
701 REM *
702 REM *
703 REM *
704 REM *
705 REM *
706 REM *
707 REM *
708 REM *
709 REM *
710 REM *
711 REM *
712 REM *
713 REM *
714 REM *
715 REM *
716 REM *
717 REM *
718 REM *
719 REM *
720 REM *
721 REM *
722 REM *
723 REM *
724 REM *
725 REM *
726 REM *
727 REM *
728 REM *
729 REM *
730 REM *
731 REM *
732 REM *
733 REM *
734 REM *
735 REM *
736 REM *
737 REM *
738 REM *
739 REM *
740 REM *
741 REM *
742 REM *
743 REM *
744 REM *
745 REM *
746 REM *
747 REM *
748 REM *
749 REM *
750 REM *
751 REM *
752 REM *
753 REM *
754 REM *
755 REM *
756 REM *
757 REM *
758 REM *
759 REM *
760 REM *
761 REM *
762 REM *
763 REM *
764 REM *
765 REM *
766 REM *
767 REM *
768 REM *
769 REM *
770 REM *
771 REM *
772 REM *
773 REM *
774 REM *
775 REM *
776 REM *
777 REM *
778 REM *
779 REM *
780 REM *
781 REM *
782 REM *
783 REM *
784 REM *
785 REM *
786 REM *
787 REM *
788 REM *
789 REM *
790 REM *
791 REM *
792 REM *
793 REM *
794 REM *
795 REM *
796 REM *
797 REM *
798 REM *
799 REM *
800 REM *
801 REM *
802 REM *
803 REM *
804 REM *
805 REM *
806 REM *
807 REM *
808 REM *
809 REM *
810 REM *
811 REM *
812 REM *
813 REM *
814 REM *
815 REM *
816 REM *
817 REM *
818 REM *
819 REM *
820 REM *
821 REM *
822 REM *
823 REM *
824 REM *
825 REM *
826 REM *
827 REM *
828 REM *
829 REM *
830 REM *
831 REM *
832 REM *
833 REM *
834 REM *
835 REM *
836 REM *
837 REM *
838 REM *
839 REM *
840 REM *
841 REM *
842 REM *
843 REM *
844 REM *
845 REM *
846 REM *
847 REM *
848 REM *
849 REM *
850 REM *
851 REM *
852 REM *
853 REM *
854 REM *
855 REM *
856 REM *
857 REM *
858 REM *
859 REM *
860 REM *
861 REM *
862 REM *
863 REM *
864 REM *
865 REM *
866 REM *
867 REM *
868 REM *
869 REM *
870 REM *
871 REM *
872 REM *
873 REM *
874 REM *
875 REM *
876 REM *
877 REM *
878 REM *
879 REM *
880 REM *
881 REM *
882 REM *
883 REM *
884 REM *
885 REM *
886 REM *
887 REM *
888 REM *
889 REM *
890 REM *
891 REM *
892 REM *
893 REM *
894 REM *
895 REM *
896 REM *
897 REM *
898 REM *
899 REM *
900 REM *
901 REM *
902 REM *
903 REM *
904 REM *
905 REM *
906 REM *
907 REM *
908 REM *
909 REM *
910 REM *
911 REM *
912 REM *
913 REM *
914 REM *
915 REM *
916 REM *
917 REM *
918 REM *
919 REM *
920 REM *
921 REM *
922 REM *
923 REM *
924 REM *
925 REM *
926 REM *
927 REM *
928 REM *
929 REM *
930 REM *
931 REM *
932 REM *
933 REM *
934 REM *
935 REM *
936 REM *
937 REM *
938 REM *
939 REM *
940 REM *
941 REM *
942 REM *
943 REM *
944 REM *
945 REM *
946 REM *
947 REM *
948 REM *
949 REM *
950 REM *
951 REM *
952 REM *
953 REM *
954 REM *
955 REM *
956 REM *
957 REM *
958 REM *
959 REM *
960 REM *
961 REM *
962 REM *
963 REM *
964 REM *
965 REM *
966 REM *
967 REM *
968 REM *
969 REM *
970 REM *
971 REM *
972 REM *
973 REM *
974 REM *
975 REM *
976 REM *
977 REM *
978 REM *
979 REM *
980 REM *
981 REM *
982 REM *
983 REM *
984 REM *
985 REM *
986 REM *
987 REM *
988 REM *
989 REM *
990 REM *
991 REM *
992 REM *
993 REM *
994 REM *
995 REM *
996 REM *
997 REM *
998 REM *
999 REM *
1000 REM *
    
```

En esta última zona cada canal es descrito mediante 5 bytes los cuales contienen:

- 1) Dirección de la rutina de salida en el ROM (2 bytes) cuando el mismo es utilizado para conducir datos desde el computador hacia el exterior.
- 2) Dirección de la rutina de entrada en el ROM (2 bytes) cuando el mismo es utilizado para conducir datos en sentido inverso.
- 3) Especificación del Dispositivo (1 byte) sirve para distinguir el tipo de periférico que se maneja con el canal.

Nótese que, en el caso de la pantalla de video y el impresor, el canal es utilizado como salida únicamente.

En esta nota ofrecemos el software necesario para manejar adecuadamente el intercambio de información entre la TS-2068 y cualquier impresora.

El método intrínseco de acceso es el mismo, primariamente, a la habilidad que previeron los diseñadores del computador para el agregado de dispositivos externos, originalmente no previstos por el fabricante.

Desde la óptica de nuestro objetivo todo se reducirá a alojar la dirección de la misma en la posición correcta para el canal # 3 dentro del área de variables de sistemas CHANS.

En la figura 2 se puede observar el driver de software en lenguaje Assembler, nótese que esencialmente no difiere de la rutina en BASIC previamente expuesta, excepto en lo referente a cuidar la traducción de caracteres compuestos, como las instrucciones BASIC, por ejemplo, donde el computador envía un carácter representativo de varios.

En la figura 3 se observa un pequeño programa en BASIC cuya misión es la carga del driver y el acomodamiento de las variables de sistemas para su uso. Nótese que, al finalizar la ejecución, el mismo se autodestruye dejando el computador libre para cualquier otro uso. La forma de operación será sencillamente:

Figura 2

```

0000 LD A,(PUSH)
0001 LD A,(PUSH)
0002 LD A,(PUSH)
0003 LD A,(PUSH)
0004 LD A,(PUSH)
0005 LD A,(PUSH)
0006 LD A,(PUSH)
0007 LD A,(PUSH)
0008 LD A,(PUSH)
0009 LD A,(PUSH)
0010 LD A,(PUSH)
0011 LD A,(PUSH)
0012 LD A,(PUSH)
0013 LD A,(PUSH)
0014 LD A,(PUSH)
0015 LD A,(PUSH)
0016 LD A,(PUSH)
0017 LD A,(PUSH)
0018 LD A,(PUSH)
0019 LD A,(PUSH)
0020 LD A,(PUSH)
0021 LD A,(PUSH)
0022 LD A,(PUSH)
0023 LD A,(PUSH)
0024 LD A,(PUSH)
0025 LD A,(PUSH)
0026 LD A,(PUSH)
0027 LD A,(PUSH)
0028 LD A,(PUSH)
0029 LD A,(PUSH)
0030 LD A,(PUSH)
0031 LD A,(PUSH)
0032 LD A,(PUSH)
0033 LD A,(PUSH)
0034 LD A,(PUSH)
0035 LD A,(PUSH)
0036 LD A,(PUSH)
0037 LD A,(PUSH)
0038 LD A,(PUSH)
0039 LD A,(PUSH)
0040 LD A,(PUSH)
0041 LD A,(PUSH)
0042 LD A,(PUSH)
0043 LD A,(PUSH)
0044 LD A,(PUSH)
0045 LD A,(PUSH)
0046 LD A,(PUSH)
0047 LD A,(PUSH)
0048 LD A,(PUSH)
0049 LD A,(PUSH)
0050 LD A,(PUSH)
0051 LD A,(PUSH)
0052 LD A,(PUSH)
0053 LD A,(PUSH)
0054 LD A,(PUSH)
0055 LD A,(PUSH)
0056 LD A,(PUSH)
0057 LD A,(PUSH)
0058 LD A,(PUSH)
0059 LD A,(PUSH)
0060 LD A,(PUSH)
0061 LD A,(PUSH)
0062 LD A,(PUSH)
0063 LD A,(PUSH)
0064 LD A,(PUSH)
0065 LD A,(PUSH)
0066 LD A,(PUSH)
0067 LD A,(PUSH)
0068 LD A,(PUSH)
0069 LD A,(PUSH)
0070 LD A,(PUSH)
0071 LD A,(PUSH)
0072 LD A,(PUSH)
0073 LD A,(PUSH)
0074 LD A,(PUSH)
0075 LD A,(PUSH)
0076 LD A,(PUSH)
0077 LD A,(PUSH)
0078 LD A,(PUSH)
0079 LD A,(PUSH)
0080 LD A,(PUSH)
0081 LD A,(PUSH)
0082 LD A,(PUSH)
0083 LD A,(PUSH)
0084 LD A,(PUSH)
0085 LD A,(PUSH)
0086 LD A,(PUSH)
0087 LD A,(PUSH)
0088 LD A,(PUSH)
0089 LD A,(PUSH)
0090 LD A,(PUSH)
0091 LD A,(PUSH)
0092 LD A,(PUSH)
0093 LD A,(PUSH)
0094 LD A,(PUSH)
0095 LD A,(PUSH)
0096 LD A,(PUSH)
0097 LD A,(PUSH)
0098 LD A,(PUSH)
0099 LD A,(PUSH)
0100 LD A,(PUSH)
0101 LD A,(PUSH)
0102 LD A,(PUSH)
0103 LD A,(PUSH)
0104 LD A,(PUSH)
0105 LD A,(PUSH)
0106 LD A,(PUSH)
0107 LD A,(PUSH)
0108 LD A,(PUSH)
0109 LD A,(PUSH)
0110 LD A,(PUSH)
0111 LD A,(PUSH)
0112 LD A,(PUSH)
0113 LD A,(PUSH)
0114 LD A,(PUSH)
0115 LD A,(PUSH)
0116 LD A,(PUSH)
0117 LD A,(PUSH)
0118 LD A,(PUSH)
0119 LD A,(PUSH)
0120 LD A,(PUSH)
0121 LD A,(PUSH)
0122 LD A,(PUSH)
0123 LD A,(PUSH)
0124 LD A,(PUSH)
0125 LD A,(PUSH)
0126 LD A,(PUSH)
0127 LD A,(PUSH)
0128 LD A,(PUSH)
0129 LD A,(PUSH)
0130 LD A,(PUSH)
0131 LD A,(PUSH)
0132 LD A,(PUSH)
0133 LD A,(PUSH)
0134 LD A,(PUSH)
0135 LD A,(PUSH)
0136 LD A,(PUSH)
0137 LD A,(PUSH)
0138 LD A,(PUSH)
0139 LD A,(PUSH)
0140 LD A,(PUSH)
0141 LD A,(PUSH)
0142 LD A,(PUSH)
0143 LD A,(PUSH)
0144 LD A,(PUSH)
0145 LD A,(PUSH)
0146 LD A,(PUSH)
0147 LD A,(PUSH)
0148 LD A,(PUSH)
0149 LD A,(PUSH)
0150 LD A,(PUSH)
0151 LD A,(PUSH)
0152 LD A,(PUSH)
0153 LD A,(PUSH)
0154 LD A,(PUSH)
0155 LD A,(PUSH)
0156 LD A,(PUSH)
0157 LD A,(PUSH)
0158 LD A,(PUSH)
0159 LD A,(PUSH)
0160 LD A,(PUSH)
0161 LD A,(PUSH)
0162 LD A,(PUSH)
0163 LD A,(PUSH)
0164 LD A,(PUSH)
0165 LD A,(PUSH)
0166 LD A,(PUSH)
0167 LD A,(PUSH)
0168 LD A,(PUSH)
0169 LD A,(PUSH)
0170 LD A,(PUSH)
0171 LD A,(PUSH)
0172 LD A,(PUSH)
0173 LD A,(PUSH)
0174 LD A,(PUSH)
0175 LD A,(PUSH)
0176 LD A,(PUSH)
0177 LD A,(PUSH)
0178 LD A,(PUSH)
0179 LD A,(PUSH)
0180 LD A,(PUSH)
0181 LD A,(PUSH)
0182 LD A,(PUSH)
0183 LD A,(PUSH)
0184 LD A,(PUSH)
0185 LD A,(PUSH)
0186 LD A,(PUSH)
0187 LD A,(PUSH)
0188 LD A,(PUSH)
0189 LD A,(PUSH)
0190 LD A,(PUSH)
0191 LD A,(PUSH)
0192 LD A,(PUSH)
0193 LD A,(PUSH)
0194 LD A,(PUSH)
0195 LD A,(PUSH)
0196 LD A,(PUSH)
0197 LD A,(PUSH)
0198 LD A,(PUSH)
0199 LD A,(PUSH)
0200 LD A,(PUSH)
0201 LD A,(PUSH)
0202 LD A,(PUSH)
0203 LD A,(PUSH)
0204 LD A,(PUSH)
0205 LD A,(PUSH)
0206 LD A,(PUSH)
0207 LD A,(PUSH)
0208 LD A,(PUSH)
0209 LD A,(PUSH)
0210 LD A,(PUSH)
0211 LD A,(PUSH)
0212 LD A,(PUSH)
0213 LD A,(PUSH)
0214 LD A,(PUSH)
0215 LD A,(PUSH)
0216 LD A,(PUSH)
0217 LD A,(PUSH)
0218 LD A,(PUSH)
0219 LD A,(PUSH)
0220 LD A,(PUSH)
0221 LD A,(PUSH)
0222 LD A,(PUSH)
0223 LD A,(PUSH)
0224 LD A,(PUSH)
0225 LD A,(PUSH)
0226 LD A,(PUSH)
0227 LD A,(PUSH)
0228 LD A,(PUSH)
0229 LD A,(PUSH)
0230 LD A,(PUSH)
0231 LD A,(PUSH)
0232 LD A,(PUSH)
0233 LD A,(PUSH)
0234 LD A,(PUSH)
0235 LD A,(PUSH)
0236 LD A,(PUSH)
0237 LD A,(PUSH)
0238 LD A,(PUSH)
0239 LD A,(PUSH)
0240 LD A,(PUSH)
0241 LD A,(PUSH)
0242 LD A,(PUSH)
0243 LD A,(PUSH)
0244 LD A,(PUSH)
0245 LD A,(PUSH)
0246 LD A,(PUSH)
0247 LD A,(PUSH)
0248 LD A,(PUSH)
0249 LD A,(PUSH)
0250 LD A,(PUSH)
0251 LD A,(PUSH)
0252 LD A,(PUSH)
0253 LD A,(PUSH)
0254 LD A,(PUSH)
0255 LD A,(PUSH)
0256 LD A,(PUSH)
0257 LD A,(PUSH)
0258 LD A,(PUSH)
0259 LD A,(PUSH)
0260 LD A,(PUSH)
0261 LD A,(PUSH)
0262 LD A,(PUSH)
0263 LD A,(PUSH)
0264 LD A,(PUSH)
0265 LD A,(PUSH)
0266 LD A,(PUSH)
0267 LD A,(PUSH)
0268 LD A,(PUSH)
0269 LD A,(PUSH)
0270 LD A,(PUSH)
0271 LD A,(PUSH)
0272 LD A,(PUSH)
0273 LD A,(PUSH)
0274 LD A,(PUSH)
0275 LD A,(PUSH)
0276 LD A,(PUSH)
0277 LD A,(PUSH)
0278 LD A,(PUSH)
0279 LD A,(PUSH)
0280 LD A,(PUSH)
0281 LD A,(PUSH)
0282 LD A,(PUSH)
0283 LD A,(PUSH)
0284 LD A,(PUSH)
0285 LD A,(PUSH)
0286 LD A,(PUSH)
0287 LD A,(PUSH)
0288 LD A,(PUSH)
0289 LD A,(PUSH)
0290 LD A,(PUSH)
0291 LD A,(PUSH)
0292 LD A,(PUSH)
0293 LD A,(PUSH)
0294 LD A,(PUSH)
0295 LD A,(PUSH)
0296 LD A,(PUSH)
0297 LD A,(PUSH)
0298 LD A,(PUSH)
0299 LD A,(PUSH)
0300 LD A,(PUSH)
0301 LD A,(PUSH)
0302 LD A,(PUSH)
0303 LD A,(PUSH)
0304 LD A,(PUSH)
0305 LD A,(PUSH)
0306 LD A,(PUSH)
0307 LD A,(PUSH)
0308 LD A,(PUSH)
0309 LD A,(PUSH)
0310 LD A,(PUSH)
0311 LD A,(PUSH)
0312 LD A,(PUSH)
0313 LD A,(PUSH)
0314 LD A,(PUSH)
0315 LD A,(PUSH)
0316 LD A,(PUSH)
0317 LD A,(PUSH)
0318 LD A,(PUSH)
0319 LD A,(PUSH)
0320 LD A,(PUSH)
0321 LD A,(PUSH)
0322 LD A,(PUSH)
0323 LD A,(PUSH)
0324 LD A,(PUSH)
0325 LD A,(PUSH)
0326 LD A,(PUSH)
0327 LD A,(PUSH)
0328 LD A,(PUSH)
0329 LD A,(PUSH)
0330 LD A,(PUSH)
0331 LD A,(PUSH)
0332 LD A,(PUSH)
0333 LD A,(PUSH)
0334 LD A,(PUSH)
0335 LD A,(PUSH)
0336 LD A,(PUSH)
0337 LD A,(PUSH)
0338 LD A,(PUSH)
0339 LD A,(PUSH)
0340 LD A,(PUSH)
0341 LD A,(PUSH)
0342 LD A,(PUSH)
0343 LD A,(PUSH)
0344 LD A,(PUSH)
0345 LD A,(PUSH)
0346 LD A,(PUSH)
0347 LD A,(PUSH)
0348 LD A,(PUSH)
0349 LD A,(PUSH)
0350 LD A,(PUSH)
0351 LD A,(PUSH)
0352 LD A,(PUSH)
0353 LD A,(PUSH)
0354 LD A,(PUSH)
0355 LD A,(PUSH)
0356 LD A,(PUSH)
0357 LD A,(PUSH)
0358 LD A,(PUSH)
0359 LD A,(PUSH)
0360 LD A,(PUSH)
0361 LD A,(PUSH)
0362 LD A,(PUSH)
0363 LD A,(PUSH)
0364 LD A,(PUSH)
0365 LD A,(PUSH)
0366 LD A,(PUSH)
0367 LD A,(PUSH)
0368 LD A,(PUSH)
0369 LD A,(PUSH)
0370 LD A,(PUSH)
0371 LD A,(PUSH)
0372 LD A,(PUSH)
0373 LD A,(PUSH)
0374 LD A,(PUSH)
0375 LD A,(PUSH)
0376 LD A,(PUSH)
0377 LD A,(PUSH)
0378 LD A,(PUSH)
0379 LD A,(PUSH)
0380 LD A,(PUSH)
0381 LD A,(PUSH)
0382 LD A,(PUSH)
0383 LD A,(PUSH)
0384 LD A,(PUSH)
0385 LD A,(PUSH)
0386 LD A,(PUSH)
0387 LD A,(PUSH)
0388 LD A,(PUSH)
0389 LD A,(PUSH)
0390 LD A,(PUSH)
0391 LD A,(PUSH)
0392 LD A,(PUSH)
0393 LD A,(PUSH)
0394
```



El número 8 surgió de un experimento muy curioso, para la época en que lo escribí estaba en plena discusión lo que con el tiempo (en el año 1989) sería el AMSAT OSCAR 19 (LUSAT), primer satélite argentino.

Faltaba mucho para que los eventos que determinaran como realmente fue el LUSAT ocurrieran; por este entonces se jugaba con la noción de construir satélite que fuera mayormente una baliza. Por la época el UOSAT-OSCAR 9 revolucionaba los aires como un satélite que transmitía telemetría solamente, parte de ella en forma hablada (¡!).

En una conversación alguien dijo que para hacer una cosa así se necesitaba hardware muy especializado que no teníamos ninguna chance de conseguir en Argentina; lo que a mi me sonó a cuento.

Por eso escribí un programa que fuera capaz de hablar basado en los limitadísimos recursos de sonido que tenía una TS2068 (para las máquinas modernas con placa de sonido esto no hubiera sido ningún desafío), lo único que tenía esa máquina era el circuito de grabación de cassette, esencialmente un puerto digital de 1 bit con un cierto de-énfasis para poder grabar la salida en la gama del audio.

Ese puerto estaba pensado para generar tonos de 1200 y 2400 Hz solamente (modulación Manchester), pero con algo de ingenio se lo podía hacer generar tonos entre 100 y 3000 Hz.

O sea que uno podía tomar algo parecido a una versión digital de un sonido por la entrada de cassette mas las rutinas en lenguaje de máquina adecuadas, almacenarlo digitalmente y luego reproducirlo.

Era posible crudamente repetir mensajes y formar palabras (almacenando los fonemas y reproduciéndolos en secuencia).

Este software dio luego pié al sintetizador de voz que apareció varios números después.

ALMACENANDO SONIDOS

COMPUTADORAS QUE HABLAN

Las formas que un computador tiene de comunicarse con el exterior han estado tradicionalmente limitadas a la utilización de medios mecánicos o electrónicos (teclados, video, impresor, etc.). De una época a esta parte, no obstante, el empleo de alta tecnología y la producción masiva han hecho populares los denominados sintetizadores de voz. Son dispositivos electrónicos que generan voz humana partiendo de señales procesables por el computador.

El camino inverso, el reconocimiento de la voz el cual a partir de la voz humana, genera señales distinguibles y procesables por el computador, no es algo tan concreto ni tan masivo y se deberá esperar aún algún tiempo más para que así sea.

Si recurrimos a sintetizadores, no obstante, hay caminos mediante los cuales se puede dotar al computador de cierta y restringida capacidad de habla y se exponen en el presente uno de los posibles. En la figura 1 se puede observar un ejemplo arbitrario de la forma de onda de la voz humana. De la misma figura se desprende que es posible conformar la misma mediante la suma de dos señales bastante distintas. Una de ellas de alta frecuencia y de amplitud esencialmente constante y otra de baja frecuencia y con variaciones de amplitud.

Suprimir el formante de baja frecuencia implica introducir distorsión, pero el resultado sigue siendo inteligible y aun más conveniente muchas de las propiedades que permiten distinguir a la señal original (timbre por ejemplo). Es posible almacenar la señal de alta frecuencia tomando muestras de la misma a intervalos regulares. Dado que no varía en amplitud, será posible almacenar la misma mediante "1" y "0" para representar respectivamente los picos y valles.

Podremos entonces hacer que el computador tome medidas a intervalos regulares y lo almacene en



memoria. Luego haciendo el procedimiento inverso se recuperará la misma señal que ingresó. Es decir, habremos dotado al computador de cierta y restringida capacidad de habla.

Para que se pueda reproducir la señal adecuadamente el espacio entre los datos sucesivos no puede ser cualquiera; si recurrimos a un conocido teorema de comunicaciones llamado Teorema del Muestreo surge que para almacenar una señal de calidad telefónica se requerirán tomar más de 8000 muestras por segundo, mientras que si queremos obtener calidad de alta fidelidad la cifra se eleva a 40000 por segundo.

Duplicando que la calidad telefónica es adecuada, 30 K de memoria alcanzarán para almacenar una frase corta de algo menos de 4 segundos de duración, no muy útil por cierto.

Pero si en vez de almacenar una muestra por byte almacenamos una muestra por bit, el tiempo se extiende ocho veces, es decir algo más de 30 segundos, lo que ya suena más práctico.

El programa expuesto en la figura 2 utiliza el último de los esquemas mencionados, permitiendo almacenar

una frase o un sonido y luego reproducirlo a voluntad. Como entrada se utiliza el port de cassette del computador y se asigna como espacio de almacenamiento la porción de memoria que va desde la dirección 30000 ya 64890, un pequeño segmento de programa en Assembler es el encargado de la tarea de muestreo y recuperación.

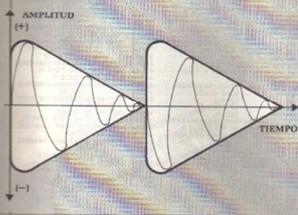
Como salida se utiliza el parlante interno del computador (el que se utiliza con el comando BEEP). El uso de Assembler es imprescindible en este caso por los estrictos requisitos de tiempo que se deben cumplir; por otra parte notese que se realiza el ensamblado en una pasada muy atrás de la memoria RAM (64900) cosa que tampoco es casual, pues de hacerse por debajo de la posición 32768 las interrupciones del LLA del computador podrían generar video serían perceptibles.

Las interrupciones son desactivadas por la misma razón, es decir para no introducir retardos en el proceso de muestreo.

Para utilizar plenamente el programa se deberán "POKEAR" los códigos correspondientes a FUNCIÓN, VELOCIDAD, DIRECCIÓN

Dentro de las múltiples aplicaciones de la TS 2068, hay una que no ha sido difundida ampliamente: es la capacidad de memorizar y reproducir sonidos aleatorios, como la música y la voz humana. En este trabajo se ofrece un programa que permite cumplir esta función.

Figura 1



DE COMIENZO Y DIRECCIÓN DE FIN. Nótese que el punto de entrada es el mismo (USR 64900) y el funcionamiento viene dado por el valor de lo que se almacena en FUNCT (64902).

La velocidad del muestreo es también programable con el valor que se introduce en SPEED (64903) siendo los valores posibles comprendidos entre 1 y 255, pudiéndose experimentar el efecto que el aumento del tiempo entre muestros y muestra produce en la señal reproducida.

El programa almacena el resultado a partir de la dirección indicada en START (MSB 64904) LS 64905 memoria RAM dedicada al STACK de la máquina fuera del ámbito de la operación.

posee se puede utilizar tanto para almacenar frases completas como para almacenar distintas palabras cortas en lugares predeterminados de la memoria para luego reproducirlas.

No existe limitación, excepto por consideraciones de espacio, para que este programa funcione como parte de otro en lenguaje BASIC y al cual se le provea soporte de "tabla".

La señal no necesariamente debe estar limitada a voz, reproduciéndose con el mismo nivel de calidad reducida a cualquier sonido.

Para la captura del sonido se deberá tener en cuenta que la entrada de cassette del TS2068 se caracteriza por necesitar de alta tensión de entrada, y debido a ello se deberá utilizar un amplificador pues normalmente no bastará con un microfono.

Algunos modelos de grabadores poseen una salida de monitor que podrá utilizarse satisfactoriamente con este propósito.

Debido a diferencias en la distribución de memoria RAM este programa no es compatible con computadores SPECTRUM, aunque puede ser modificado para ello siempre que se pueda mantener la zona de memoria RAM dedicada al STACK de la máquina fuera del ámbito de la operación.

Ing. Pedro E. Colla

Game 64 no es un juego...

Son más de 200 juegos para el computador Comodore 64

cassettes con carga garantizada la mayoría con Sistema AUTO-RUN (carga directa) nuevos títulos todos los meses

Disponemos de zonas de distribución

OFICINA DE VENTAS PARA CAPITAL E INTERIOR: C. E. SOTTI, Calle 252 30-A, Tel. 45-7966 (Cable)



La disponibilidad de computadoras de bajo precio revolucionó todos los ámbitos, y la radio no fue una excepción. La decodificación (o transmisión) en código Morse era algo como una curiosidad pero en realidad cualquier implementación rápidamente choca con los límites del problema. Aún hoy tomar Morse a oído es más eficiente que por computadora.

Una historia muy diferente son los modos digitales, por entonces mayormente confinados a RTTY donde la operación a "mano" no es una opción. Los modem para RTTY eran caros y en general utilizados con máquinas muy voluminosas (y ruidosas) de "surplus"; no era raro que se tratara de un modo mayormente confinado a unos pocos especialistas.

Pero ahora con un modem sencillo, una computadora económica (que se podía usar para otras cosas) y el software adecuado cualquiera podía salir en RTTY; el tema junto con la enorme cantidad de información a intercambiar hizo rápidamente furor.

La TS2068 no tenía ni siquiera puerto serie, o sea que había que empezar bien de abajo....

INTERFASE SERIE

DOMINANDO LAS COMUNICACIONES CON LA TS 2068

INICIAMOS UNA SERIE DE NOTAS QUE NOS PERMITIRÁ USAR LA TS 2068 COMO TERMINAL DE RTTY, TELETYPE, MORSE Y PARA COMUNICACIONES POR MODEM TELEFÓNICO.

El uso de computadores asociados a la comunicación ha tenido en los últimos años un incremento tan explosivo como el de computadores en sí mismo.

El computador personal y aún el hogareño, se adapta especialmente a este uso por la flexibilidad y generalidad que le otorga.

El uso de máquinas dedicadas, un teleprinter o una terminal por ejemplo, solucionan la transferencia de los datos pero aportan muy poco al tratamiento anterior y posterior de los mismos.

Con un computador, en cambio, los datos transferidos pueden ser almacenados y usados posteriormente en forma más directa.

Por otra parte, las opciones mecánica o electrónicamente son difíciles o costosas de lograr en un aparato no-inteligente como los indicadores, es a veces trivial obtenerlos mediante el adecuado uso del software de casi cualquier computador.

Se dividirá en el presente entrega y subsiguiente el hardware y software necesarios para mediante un computador TS2068 realizar el recibir datos.

El computador almacena y trata la información mediante el almacenamiento de la misma en forma digital, sea en su memoria o en un medio externo tal como un cassette o diskette.

El término digital se aplica pues, en definitiva lo que puede entender el computador son combinaciones de señales tipo "encendido" o "apagado", o dicho más exactamente de "1" y "0".

Para poder utilizar en forma práctica esta representación cada carácter (letra, número o símbolo) es reproducido por una combinación de dígitos binarios (bits) denominados bits.

Es norma generalizada el que un byte está compuesto de 8 bits, con lo cual se pueden almacenar en el mismo 256 combinaciones diferentes.

Habitualmente, parte de las mismas están dedicadas a la representación de las letras, los números y los caracteres especiales.

Los restantes son utilizados para definir distintos caracteres útiles para el manejo del computador, aunque sin un significado concreto en lenguaje humano.

La definición de un conjunto de bits se le asigna qué carácter permite formar lo que se denomina un código.

Existen numerosos códigos aplicables a distintas actividades aunque sin duda los más populares son el Baudot-Murray y el ASCII.

El código Baudot es uno de los más viejos en el ámbito de las comunicaciones, de hecho anterior a la era de las computadoras, y es aún hoy en día, utilizado masivamente en las transmisiones de telex y para la difusión de noticias y servicios meteorológicos.

Su principal característica radica en representar mediante 5 bits todos los caracteres necesarios para transmitir textos.

Esta representación solamente permitirá definir 32 combinaciones diferentes, obviamente insuficiente para cualquier uso práctico, pues no alcanza para definir el alfabeto, ni los dígitos numéricos.

Para solucionar este aspecto se utilizan dos juegos de códigos de 32 caracteres cada uno, uno de ellos para representar primariamente letras, y el restante números y caracteres de puntuación.

Los códigos, que reciben la denominación de "LETRAS" y "FIGURAS", se utilizan de tal manera que, cuando un teleprinter recibe el correspondiente a "LETRAS", todos los que ingresen a continuación serán del juego de códigos que representan letras. Mientras que, luego de un carácter "FIGURAS", se utilizará para los subsiguientes, la parte del código que representa números y símbolos.

Esta codificación es bastante eficiente pues solamente se requieren 5 bits para transmitir cualquier texto, sin embargo no está exenta de inconvenientes. En primer lugar la cantidad de caracteres definidos es reducida para la transmisión de un texto, pero lo interesante de ciertos símbolos de uso muy común (el asterisco por ejemplo) no permite su utilización para tareas relacionadas con computación.

Quizás la desventaja más importante reside en la dificultad de implementar para este código, métodos que permitan detectar errores en la transferencia de datos.

Debido a ello y a otros numerosos factores de tipo tecnológico se ha popularizado otro código, este de 8 bits, denominado ASCII.

El nombre proviene de las iniciales del Código Americano para el intercambio de información (American Standard Code for Information Interchange).

Las características de este código, que lo han universalizado para su utilización en forma interna por los computadores, lo hacen interesante también para la transferencia de datos.

El código ASCII permite hasta 256 caracteres diferentes, asignándose a cada letra, número o símbolo una combinación específica, la normalización hace que prácticamente cualquier máquina represente los mismos caracteres, independientemente de los lugares "vacíos" del código son asignados en particular por cada computador.

Para transmitir o recibir un carácter, cualquiera sea el código utilizado, entre dos puntos se deben transferir todos los bits que conforman cada letra o número.

Esto puede llevarse a cabo transmitiendo todos los bits al mismo tiempo, hablamos en estos casos de comunicación de tipo paralelo. Este tipo de comunicación se limita a algunos metros de distancia y es usual encontrarla asociada con circuitos periféricos (impresores por ejemplo).

El otro tipo de comunicación posible es, cuando a través de una sola línea, se transfieren bit por bit en forma secuencial; este tipo de comunicación recibe el nombre de Serie.

En la comunicación serie deben existir ciertos acuerdos básicos entre ambos extremos de la comunicación para que la misma sea posible. El más importante es acerca de cuánto tiempo tardará cada bit en transmitirse, o visto de otra manera, cuál será la velocidad de la transferencia.

Este parámetro se mide usualmente en bits por segundo o baudios. Al respecto existen velocidades comúnmente utilizadas o estándares, algunas de ellas 50, 300, 1200 baudios.

Otro aspecto importante a establecer en forma previa a una comunicación es de cuántos bits estará formado cada carácter, pudiendo ser 5, 6, 7 u 8 bits.



Fig. 18

K64

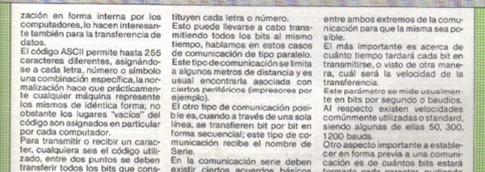
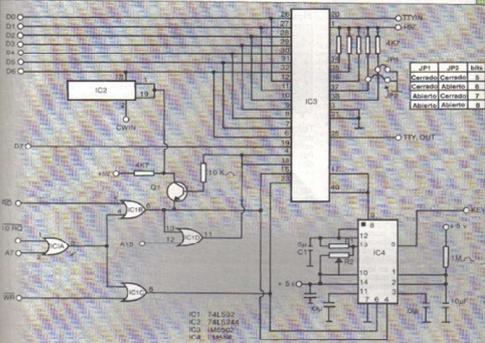


Fig. 19

K64

INTERFASE SERIE

ser como en el caso de código Baudot y entre seis y ocho de acuerdo al tipo de transmisión que se desea utilizar, en el caso de código ASCII.

Para poder utilizar en forma práctica esta representación cada carácter (letra, número o símbolo) es reproducido por una combinación de dígitos binarios (bits) denominados bits.

Es norma generalizada el que un byte está compuesto de 8 bits, con lo cual se pueden almacenar en el mismo 256 combinaciones diferentes.

Habitualmente, parte de las mismas están dedicadas a la representación de las letras, los números y los caracteres especiales.

Los restantes son utilizados para definir distintos caracteres útiles para el manejo del computador, aunque sin un significado concreto en lenguaje humano.

La definición de un conjunto de bits se le asigna qué carácter permite formar lo que se denomina un código.

Existen numerosos códigos aplicables a distintas actividades aunque sin duda los más populares son el Baudot-Murray y el ASCII.

El código Baudot es uno de los más viejos en el ámbito de las comunicaciones, de hecho anterior a la era de las computadoras, y es aún hoy en día, utilizado masivamente en las transmisiones de telex y para la difusión de noticias y servicios meteorológicos.

Su principal característica radica en representar mediante 5 bits todos los caracteres necesarios para transmitir textos.

Esta representación solamente permitirá definir 32 combinaciones diferentes, obviamente insuficiente para cualquier uso práctico, pues no alcanza para definir el alfabeto, ni los dígitos numéricos.

Para solucionar este aspecto se utilizan dos juegos de códigos de 32 caracteres cada uno, uno de ellos para representar primariamente letras, y el restante números y caracteres de puntuación.

Los códigos, que reciben la denominación de "LETRAS" y "FIGURAS", se utilizan de tal manera que, cuando un teleprinter recibe el correspondiente a "LETRAS", todos los que ingresen a continuación serán del juego de códigos que representan letras. Mientras que, luego de un carácter "FIGURAS", se utilizará para los subsiguientes, la parte del código que representa números y símbolos.

Esta codificación es bastante eficiente pues solamente se requieren 5 bits para transmitir cualquier texto, sin embargo no está exenta de inconvenientes. En primer lugar la cantidad de caracteres definidos es reducida para la transmisión de un texto, pero lo interesante de ciertos símbolos de uso muy común (el asterisco por ejemplo) no permite su utilización para tareas relacionadas con computación.

Quizás la desventaja más importante reside en la dificultad de implementar para este código, métodos que permitan detectar errores en la transferencia de datos.

Debido a ello y a otros numerosos factores de tipo tecnológico se ha popularizado otro código, este de 8 bits, denominado ASCII.

El nombre proviene de las iniciales del Código Americano para el intercambio de información (American Standard Code for Information Interchange).

Las características de este código, que lo han universalizado para su utilización en forma interna por los computadores, lo hacen interesante también para la transferencia de datos.

El código ASCII permite hasta 256 caracteres diferentes, asignándose a cada letra, número o símbolo una combinación específica, la normalización hace que prácticamente cualquier máquina represente los mismos caracteres, independientemente de los lugares "vacíos" del código son asignados en particular por cada computador.

Para transmitir o recibir un carácter, cualquiera sea el código utilizado, entre dos puntos se deben transferir todos los bits que conforman cada letra o número.

Esto puede llevarse a cabo transmitiendo todos los bits al mismo tiempo, hablamos en estos casos de comunicación de tipo paralelo. Este tipo de comunicación se limita a algunos metros de distancia y es usual encontrarla asociada con circuitos periféricos (impresores por ejemplo).

El otro tipo de comunicación posible es, cuando a través de una sola línea, se transfieren bit por bit en forma secuencial; este tipo de comunicación recibe el nombre de Serie.

Para determinar la velocidad de recepción y transmisión el UART requiere un reloj (Pines 17 y 40) cuya frecuencia sea 15 veces más elevada que la velocidad en baudios a la cual recibirá los datos en forma paralela.

Cabe señalar que el código utilizado por el UART no será manejado (Baudot o ASCII) no será manejado en definitiva a transformarse en ambos sentidos bits en serie a paralelo el código tendrá exclusiva responsabilidad del computador (o de su software) el cual tendrá la tarea de realizar las conversiones necesarias.

Si el UART está libre u ocupado puede manejarse o bien mediante software introduciendo un retardo equivalente a lo que se tarda en transmitir un carácter a la velocidad que se utiliza o bien a través de la interfase un Pin del dispositivo en el cual la condición de OCUPADO se refleja mediante

Como en otros casos para que el computador tenga total dominio de la interfase se debe agregar cierta cantidad de lógica de "direccionamiento" es decir que permitan al dispositivo "entender" cuáles de todas las señales que operan sobre

Pin 23) que comience a transmitir a la velocidad que le dicte el reloj el UART estará ocupado en lo que tarda hasta finalizar no recibiendo otro carácter hasta ocurrir esto.

El otro tipo de comunicación posible es, cuando a través de una sola línea, se transfieren bit por bit en forma secuencial; este tipo de comunicación recibe el nombre de Serie.

En la comunicación serie deben existir ciertos acuerdos básicos entre ambos extremos de la comunicación para que la misma sea posible. El más importante es acerca de cuánto tiempo tardará cada bit en transmitirse, o visto de otra manera, cuál será la velocidad de la transferencia.

Este parámetro se mide usualmente en bits por segundo o baudios. Al respecto existen velocidades comúnmente utilizadas o estándares, algunas de ellas 50, 300, 1200 baudios.

Otro aspecto importante a establecer en forma previa a una comunicación es de cuántos bits estará formado cada carácter, pudiendo ser 5, 6, 7 u 8 bits.

Para determinar la velocidad de recepción y transmisión el UART requiere un reloj (Pines 17 y 40) cuya frecuencia sea 15 veces más elevada que la velocidad en baudios a la cual recibirá los datos en forma paralela.

Cabe señalar que el código utilizado por el UART no será manejado (Baudot o ASCII) no será manejado en definitiva a transformarse en ambos sentidos bits en serie a paralelo el código tendrá exclusiva responsabilidad del computador (o de su software) el cual tendrá la tarea de realizar las conversiones necesarias.

Si el UART está libre u ocupado puede manejarse o bien mediante software introduciendo un retardo equivalente a lo que se tarda en transmitir un carácter a la velocidad que se utiliza o bien a través de la interfase un Pin del dispositivo en el cual la condición de OCUPADO se refleja mediante

Como en otros casos para que el computador tenga total dominio de la interfase se debe agregar cierta cantidad de lógica de "direccionamiento" es decir que permitan al dispositivo "entender" cuáles de todas las señales que operan sobre

Pin 23) que comience a transmitir a la velocidad que le dicte el reloj el UART estará ocupado en lo que tarda hasta finalizar no recibiendo otro carácter hasta ocurrir esto.

El otro tipo de comunicación posible es, cuando a través de una sola línea, se transfieren bit por bit en forma secuencial; este tipo de comunicación recibe el nombre de Serie.

En la comunicación serie deben existir ciertos acuerdos básicos entre ambos extremos de la comunicación para que la misma sea posible. El más importante es acerca de cuánto tiempo tardará cada bit en transmitirse, o visto de otra manera, cuál será la velocidad de la transferencia.

Este parámetro se mide usualmente en bits por segundo o baudios. Al respecto existen velocidades comúnmente utilizadas o estándares, algunas de ellas 50, 300, 1200 baudios.

Otro aspecto importante a establecer en forma previa a una comunicación es de cuántos bits estará formado cada carácter, pudiendo ser 5, 6, 7 u 8 bits.

Para determinar la velocidad de recepción y transmisión el UART requiere un reloj (Pines 17 y 40) cuya frecuencia sea 15 veces más elevada que la velocidad en baudios a la cual recibirá los datos en forma paralela.

Cabe señalar que el código utilizado por el UART no será manejado (Baudot o ASCII) no será manejado en definitiva a transformarse en ambos sentidos bits en serie a paralelo el código tendrá exclusiva responsabilidad del computador (o de su software) el cual tendrá la tarea de realizar las conversiones necesarias.

Si el UART está libre u ocupado puede manejarse o bien mediante software introduciendo un retardo equivalente a lo que se tarda en transmitir un carácter a la velocidad que se utiliza o bien a través de la interfase un Pin del dispositivo en el cual la condición de OCUPADO se refleja mediante

Como en otros casos para que el computador tenga total dominio de la interfase se debe agregar cierta cantidad de lógica de "direccionamiento" es decir que permitan al dispositivo "entender" cuáles de todas las señales que operan sobre

Pin 23) que comience a transmitir a la velocidad que le dicte el reloj el UART estará ocupado en lo que tarda hasta finalizar no recibiendo otro carácter hasta ocurrir esto.

El otro tipo de comunicación posible es, cuando a través de una sola línea, se transfieren bit por bit en forma secuencial; este tipo de comunicación recibe el nombre de Serie.

En la comunicación serie deben existir ciertos acuerdos básicos entre ambos extremos de la comunicación para que la misma sea posible. El más importante es acerca de cuánto tiempo tardará cada bit en transmitirse, o visto de otra manera, cuál será la velocidad de la transferencia.

Este parámetro se mide usualmente en bits por segundo o baudios. Al respecto existen velocidades comúnmente utilizadas o estándares, algunas de ellas 50, 300, 1200 baudios.

Otro aspecto importante a establecer en forma previa a una comunicación es de cuántos bits estará formado cada carácter, pudiendo ser 5, 6, 7 u 8 bits.

Para determinar la velocidad de recepción y transmisión el UART requiere un reloj (Pines 17 y 40) cuya frecuencia sea 15 veces más elevada que la velocidad en baudios a la cual recibirá los datos en forma paralela.

Cabe señalar que el código utilizado por el UART no será manejado (Baudot o ASCII) no será manejado en definitiva a transformarse en ambos sentidos bits en serie a paralelo el código tendrá exclusiva responsabilidad del computador (o de su software) el cual tendrá la tarea de realizar las conversiones necesarias.

Si el UART está libre u ocupado puede manejarse o bien mediante software introduciendo un retardo equivalente a lo que se tarda en transmitir un carácter a la velocidad que se utiliza o bien a través de la interfase un Pin del dispositivo en el cual la condición de OCUPADO se refleja mediante

Como en otros casos para que el computador tenga total dominio de la interfase se debe agregar cierta cantidad de lógica de "direccionamiento" es decir que permitan al dispositivo "entender" cuáles de todas las señales que operan sobre

Pin 23) que comience a transmitir a la velocidad que le dicte el reloj el UART estará ocupado en lo que tarda hasta finalizar no recibiendo otro carácter hasta ocurrir esto.

El otro tipo de comunicación posible es, cuando a través de una sola línea, se transfieren bit por bit en forma secuencial; este tipo de comunicación recibe el nombre de Serie.

En la comunicación serie deben existir ciertos acuerdos básicos entre ambos extremos de la comunicación para que la misma sea posible. El más importante es acerca de cuánto tiempo tardará cada bit en transmitirse, o visto de otra manera, cuál será la velocidad de la transferencia.

Este parámetro se mide usualmente en bits por segundo o baudios. Al respecto existen velocidades comúnmente utilizadas o estándares, algunas de ellas 50, 300, 1200 baudios.

Otro aspecto importante a establecer en forma previa a una comunicación es de cuántos bits estará formado cada carácter, pudiendo ser 5, 6, 7 u 8 bits.

Para determinar la velocidad de recepción y transmisión el UART requiere un reloj (Pines 17 y 40) cuya frecuencia sea 15 veces más elevada que la velocidad en baudios a la cual recibirá los datos en forma paralela.

Cabe señalar que el código utilizado por el UART no será manejado (Baudot o ASCII) no será manejado en definitiva a transformarse en ambos sentidos bits en serie a paralelo el código tendrá exclusiva responsabilidad del computador (o de su software) el cual tendrá la tarea de realizar las conversiones necesarias.

Si el UART está libre u ocupado puede manejarse o bien mediante software introduciendo un retardo equivalente a lo que se tarda en transmitir un carácter a la velocidad que se utiliza o bien a través de la interfase un Pin del dispositivo en el cual la condición de OCUPADO se refleja mediante

Como en otros casos para que el computador tenga total dominio de la interfase se debe agregar cierta cantidad de lógica de "direccionamiento" es decir que permitan al dispositivo "entender" cuáles de todas las señales que operan sobre

Pin 23) que comience a transmitir a la velocidad que le dicte el reloj el UART estará ocupado en lo que tarda hasta finalizar no recibiendo otro carácter hasta ocurrir esto.

El otro tipo de comunicación posible es, cuando a través de una sola línea, se transfieren bit por bit en forma secuencial; este tipo de comunicación recibe el nombre de Serie.

K64

K64

INTERFASE SERIE

Para determinar la velocidad de recepción y transmisión el UART requiere un reloj (Pines 17 y 40) cuya frecuencia sea 15 veces más elevada que la velocidad en baudios a la cual recibirá los datos en forma paralela.

Cabe señalar que el código utilizado por el UART no será manejado (Baudot o ASCII) no será manejado en definitiva a transformarse en ambos sentidos bits en serie a paralelo el código tendrá exclusiva responsabilidad del computador (o de su software) el cual tendrá la tarea de realizar las conversiones necesarias.

Si el UART está libre u ocupado puede manejarse o bien mediante software introduciendo un retardo equivalente a lo que se tarda en transmitir un carácter a la velocidad que se utiliza o bien a través de la interfase un Pin del dispositivo en el cual la condición de OCUPADO se refleja mediante

Como en otros casos para que el computador tenga total dominio de la interfase se debe agregar cierta cantidad de lógica de "direccionamiento" es decir que permitan al dispositivo "entender" cuáles de todas las señales que operan sobre

Pin 23) que comience a transmitir a la velocidad que le dicte el reloj el UART estará ocupado en lo que tarda hasta finalizar no recibiendo otro carácter hasta ocurrir esto.

El otro tipo de comunicación posible es, cuando a través de una sola línea, se transfieren bit por bit en forma secuencial; este tipo de comunicación recibe el nombre de Serie.

En la comunicación serie deben existir ciertos acuerdos básicos entre ambos extremos de la comunicación para que la misma sea posible. El más importante es acerca de cuánto tiempo tardará cada bit en transmitirse, o visto de otra manera, cuál será la velocidad de la transferencia.

Este parámetro se mide usualmente en bits por segundo o baudios. Al respecto existen velocidades comúnmente utilizadas o estándares, algunas de ellas 50, 300, 1200 baudios.

Otro aspecto importante a establecer en forma previa a una comunicación es de cuántos bits estará formado cada carácter, pudiendo ser 5, 6, 7 u 8 bits.

Para determinar la velocidad de recepción y transmisión el UART requiere un reloj (Pines 17 y 40) cuya frecuencia sea 15 veces más elevada que la velocidad en baudios a la cual recibirá los datos en forma paralela.

Cabe señalar que el código utilizado por el UART no será manejado (Baudot o ASCII) no será manejado en definitiva a transformarse en ambos sentidos bits en serie a paralelo el código tendrá exclusiva responsabilidad del computador (o de su software) el cual tendrá la tarea de realizar las conversiones necesarias.

Si el UART está libre u ocupado puede manejarse o bien mediante software introduciendo un retardo equivalente a lo que se tarda en transmitir un carácter a la velocidad que se utiliza o bien a través de la interfase un Pin del dispositivo en el cual la condición de OCUPADO se refleja mediante

Como en otros casos para que el computador tenga total dominio de la interfase se debe agregar cierta cantidad de lógica de "direccionamiento" es decir que permitan al dispositivo "entender" cuáles de todas las señales que operan sobre

Pin 23) que comience a transmitir a la velocidad que le dicte el reloj el UART estará ocupado en lo que tarda hasta finalizar no recibiendo otro carácter hasta ocurrir esto.

El otro tipo de comunicación posible es, cuando a través de una sola línea, se transfieren bit por bit en forma secuencial; este tipo de comunicación recibe el nombre de Serie.

En la comunicación serie deben existir ciertos acuerdos básicos entre ambos extremos de la comunicación para que la misma sea posible. El más importante es acerca de cuánto tiempo tardará cada bit en transmitirse, o visto de otra manera, cuál será la velocidad de la transferencia.

Este parámetro se mide usualmente en bits por segundo o baudios. Al respecto existen velocidades comúnmente utilizadas o estándares, algunas de ellas 50, 300, 1200 baudios.

Otro aspecto importante a establecer en forma previa a una comunicación es de cuántos bits estará formado cada carácter, pudiendo ser 5, 6, 7 u 8 bits.

Para determinar la velocidad de recepción y transmisión el UART requiere un reloj (Pines 17 y 40) cuya frecuencia sea 15 veces más elevada que la velocidad en baudios a la cual recibirá los datos en forma paralela.

Cabe señalar que el código utilizado por el UART no será manejado (Baudot o ASCII) no será manejado en definitiva a transformarse en ambos sentidos bits en serie a paralelo el código tendrá exclusiva responsabilidad del computador (o de su software) el cual tendrá la tarea de realizar las conversiones necesarias.

Si el UART está libre u ocupado puede manejarse o bien mediante software introduciendo un retardo equivalente a lo que se tarda en transmitir un carácter a la velocidad que se utiliza o bien a través de la interfase un Pin del dispositivo en el cual la condición de OCUPADO se refleja mediante

Como en otros casos para que el computador tenga total dominio de la interfase se debe agregar cierta cantidad de lógica de "direccionamiento" es decir que permitan al dispositivo "entender" cuáles de todas las señales que operan sobre

Pin 23) que comience a transmitir a la velocidad que le dicte el reloj el UART estará ocupado en lo que tarda hasta finalizar no recibiendo otro carácter hasta ocurrir esto.

El otro tipo de comunicación posible es, cuando a través de una sola línea, se transfieren bit por bit en forma secuencial; este tipo de comunicación recibe el nombre de Serie.

En la comunicación serie deben existir ciertos acuerdos básicos entre ambos extremos de la comunicación para que la misma sea posible. El más importante es acerca de cuánto tiempo tardará cada bit en transmitirse, o visto de otra manera, cuál será la velocidad de la transferencia.

Este parámetro se mide usualmente en bits por segundo o baudios. Al respecto existen velocidades comúnmente utilizadas o estándares, algunas de ellas 50, 300, 1200 baudios.

Otro aspecto importante a establecer en forma previa a una comunicación es de cuántos bits estará formado cada carácter, pudiendo ser 5, 6, 7 u 8 bits.

Para determinar la velocidad de recepción y transmisión el UART requiere un reloj (Pines 17 y 40) cuya frecuencia sea 15 veces más elevada que la velocidad en baudios a la cual recibirá los datos en forma paralela.

Cabe señalar que el código utilizado por el UART no será manejado (Baudot o ASCII) no será manejado en definitiva a transformarse en ambos sentidos bits en serie a paralelo el código tendrá exclusiva responsabilidad del computador (o de su software) el cual tendrá la tarea de realizar las conversiones necesarias.

Si el UART está libre u ocupado puede manejarse o bien mediante software introduciendo un retardo equivalente a lo que se tarda en transmitir un carácter a la velocidad que se utiliza o bien a través de la interfase un Pin del dispositivo en el cual la condición de OCUPADO se refleja mediante

Como en otros casos para que el computador tenga total dominio de la interfase se debe agregar cierta cantidad de lógica de "direccionamiento" es decir que permitan al dispositivo "entender" cuáles de todas las señales que operan sobre

Pin 23) que comience a transmitir a la velocidad que le dicte el reloj el UART estará ocupado en lo que tarda hasta finalizar no recibiendo otro carácter hasta ocurrir esto.

El otro tipo de comunicación posible es, cuando a través de una sola línea, se transfieren bit por bit en forma secuencial; este tipo de comunicación recibe el nombre de Serie.

En la comunicación serie deben existir ciertos acuerdos básicos entre ambos extremos de la comunicación para que la misma sea posible. El más importante es acerca de cuánto tiempo tardará cada bit en transmitirse, o visto de otra manera, cuál será la velocidad de la transferencia.

Este parámetro se mide usualmente en bits por segundo o baudios. Al respecto existen velocidades comúnmente utilizadas o estándares, algunas de ellas 50, 300, 1200 baudios.

Otro aspecto importante a establecer en forma previa a una comunicación es de cuántos bits estará formado cada carácter, pudiendo ser 5, 6, 7 u 8 bits.

Para determinar la velocidad de recepción y transmisión el UART requiere un reloj (Pines 17 y 40) cuya frecuencia sea 15 veces más elevada que la velocidad en baudios a la cual recibirá los datos en forma paralela.

Cabe señalar que el código utilizado por el UART no será manejado (Baudot o ASCII) no será manejado en definitiva a transformarse en ambos sentidos bits en serie a paralelo el código tendrá exclusiva responsabilidad del computador (o de su software) el cual tendrá la tarea de realizar las conversiones necesarias.

Si el UART está libre u ocupado puede manejarse o bien mediante software introduciendo un retardo equivalente a lo que se tarda en transmitir un carácter a la velocidad que se utiliza o bien a través de la interfase un Pin del dispositivo en el cual la condición de OCUPADO se refleja mediante

Como en otros casos para que el computador tenga total dominio de la interfase se debe agregar cierta cantidad de lógica de "direccionamiento" es decir que permitan al dispositivo "entender" cuáles de todas las señales que operan sobre

Pin 23) que comience a transmitir a la velocidad que le dicte el reloj el UART estará ocupado en lo que tarda hasta finalizar no recibiendo otro carácter hasta ocurrir esto.

El otro tipo de comunicación posible es, cuando a través de una sola línea, se transfieren bit por bit en forma secuencial; este tipo de comunicación recibe el nombre de Serie.

Como en otros casos para que el computador tenga total dominio de la interfase se debe agregar cierta cantidad de lógica de "direccionamiento" es decir que permitan al dispositivo "entender" cuáles de todas las señales que operan sobre

Pin 23) que comience a transmitir a la velocidad que le dicte el reloj el UART estará ocupado en lo que tarda hasta finalizar no recibiendo otro carácter hasta ocurrir esto.

El otro tipo de comunicación posible es, cuando a través de una sola línea, se transfieren bit por bit en forma secuencial; este tipo de comunicación recibe el nombre de Serie.

En la comunicación serie deben existir ciertos acuerdos básicos entre ambos extremos de la comunicación para que la misma sea posible. El más importante es acerca de cuánto tiempo tardará cada

FEBRERO 1986 N° 11 A 2 REP. ARGENTINA

K64

COMPUTACION PARA TODOS

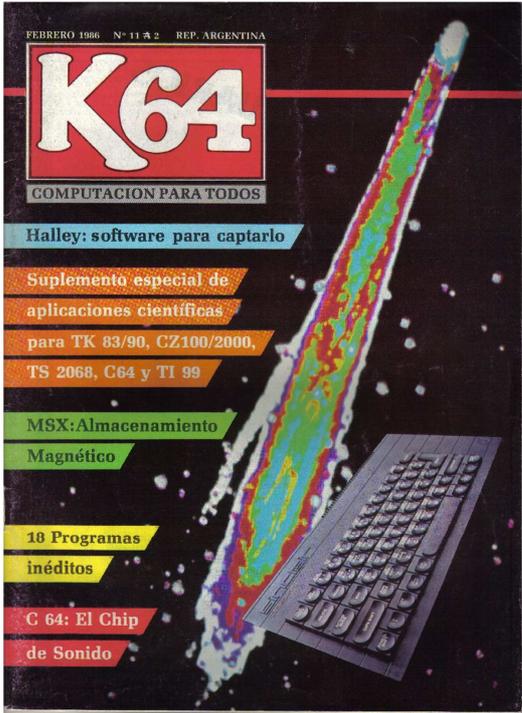
Halley: software para captarlo

Suplemento especial de
aplicaciones científicas
para TK 83/90, CZ100/2000,
TS 2068, C64 y TI 99

MSX: Almacenamiento
Magnético

18 Programas
inéditos

C 64: El Chip
de Sonido



DOMINANDO LAS COMUNICACIONES

SEGUNDA PARTE

Ing. Pedro E. Colla



En la anterior entrega definimos los fundamentos de la comunicación de datos y el hardware necesario para llevarlo a cabo. Es ahora el turno del software.

En estas páginas publicamos un programa que utilizando la interfaz descrita en la primera parte transforma un computador TS2068 en una poderosa terminal de comunicaciones.

Consta de una breve sección en lenguaje BASIC y el grueso en código de máquina para de esta manera satisfacer requerimientos estrictos de tiempo normalizados asociados al trabajo con velocidades de recepción/transmisión altas (300 bauds o mayores).

El programa está diseñado para su uso en un ámbito de comunicaciones por radio o por cable (TELE) aunque nada impide su uso en transmisión de datos a través de vía telefónica dado que es lo suficientemente general como para soportarlo.

Las facilidades que brinda son básicamente el recibir y emitir teleimpreso en código Baudot o Acici y telegrafía en código Morse.

Para hacer esto último se vale de la entrada auxiliar contenida en la interfaz y de la salida de tiempo de la misma.

El programa funciona solamente para computadores TS2068 pues requiere tener cargado en memoria el programa ensamblador denominado ZEUS-ASSEMBLER del cual utiliza rutinas.

Las características generales son:

- Soporte de teletipo en Baudot, Acici y Telegrafía Morse.
- 9 mensajes pre-programados.
- 1 mensaje programable.
- 1 buffer de recepción-transmisión.
- 1 lectura-grabación en cassette del buffer.
- Licencia programable.

Las distintas facilidades se acceden mediante COMANDOS definidos como combinaciones de teclas, en la Fig. 1 se podrá apreciar una tabla descriptiva de los mismos.

El programa en su sección de lenguaje de máquina se observa en la Fig. 2, mientras que en BASIC en la Fig. 3.

Los comentarios adjuntos al mismo permitirán el seguimiento de la lógica.

La carga se deberá hacer con el ensamblador previamente mencionado el cual limita la utilización en computadores SPECTRUM dado que este último utiliza zonas de memoria que entran en conflicto con el área de memoria en que dicho modelo coloca el stack de máquina.

No obstante no existen restricciones sobre la adaptación del programa a otros computadores que utilicen el microprocesador Z80 siempre que se tenga en cuenta la adecuada modificación de las rutinas de entrada por teclado y salida por video.

Figura 3 PROGRAMA en BASIC

```
100 DIM C(255)
110 DIM S(255)
120 DIM M(255)
130 DIM P(255)
140 DIM R(255)
150 DIM T(255)
160 DIM U(255)
170 DIM V(255)
180 DIM W(255)
190 DIM X(255)
200 DIM Y(255)
210 DIM Z(255)
220 DIM AA(255)
230 DIM AB(255)
240 DIM AC(255)
250 DIM AD(255)
260 DIM AE(255)
270 DIM AF(255)
280 DIM AG(255)
290 DIM AH(255)
300 DIM AI(255)
310 DIM AJ(255)
320 DIM AK(255)
330 DIM AL(255)
340 DIM AM(255)
350 DIM AN(255)
360 DIM AO(255)
370 DIM AP(255)
380 DIM AQ(255)
390 DIM AR(255)
400 DIM AS(255)
410 DIM AT(255)
420 DIM AU(255)
430 DIM AV(255)
440 DIM AW(255)
450 DIM AX(255)
460 DIM AY(255)
470 DIM AZ(255)
480 DIM BA(255)
490 DIM BB(255)
500 DIM BC(255)
510 DIM BD(255)
520 DIM BE(255)
530 DIM BF(255)
540 DIM BG(255)
550 DIM BH(255)
560 DIM BI(255)
570 DIM BJ(255)
580 DIM BK(255)
590 DIM BL(255)
600 DIM BM(255)
610 DIM BN(255)
620 DIM BO(255)
630 DIM BP(255)
640 DIM BQ(255)
650 DIM BR(255)
660 DIM BS(255)
670 DIM BT(255)
680 DIM BU(255)
690 DIM BV(255)
700 DIM BW(255)
710 DIM BX(255)
720 DIM BY(255)
730 DIM BZ(255)
740 DIM CA(255)
750 DIM CB(255)
760 DIM CC(255)
770 DIM CD(255)
780 DIM CE(255)
790 DIM CF(255)
800 DIM CG(255)
810 DIM CH(255)
820 DIM CI(255)
830 DIM CJ(255)
840 DIM CK(255)
850 DIM CL(255)
860 DIM CM(255)
870 DIM CN(255)
880 DIM CO(255)
890 DIM CP(255)
900 DIM CQ(255)
910 DIM CR(255)
920 DIM CS(255)
930 DIM CT(255)
940 DIM CU(255)
950 DIM CV(255)
960 DIM CW(255)
970 DIM CX(255)
980 DIM CY(255)
990 DIM CZ(255)
1000 DIM DA(255)
1010 DIM DB(255)
1020 DIM DC(255)
1030 DIM DD(255)
1040 DIM DE(255)
1050 DIM DF(255)
1060 DIM DG(255)
1070 DIM DH(255)
1080 DIM DI(255)
1090 DIM DJ(255)
1100 DIM DK(255)
1110 DIM DL(255)
1120 DIM DM(255)
1130 DIM DN(255)
1140 DIM DO(255)
1150 DIM DP(255)
1160 DIM DQ(255)
1170 DIM DR(255)
1180 DIM DS(255)
1190 DIM DT(255)
1200 DIM DU(255)
1210 DIM DV(255)
1220 DIM DW(255)
1230 DIM DX(255)
1240 DIM DY(255)
1250 DIM DZ(255)
1260 DIM EA(255)
1270 DIM EB(255)
1280 DIM EC(255)
1290 DIM ED(255)
1300 DIM EE(255)
1310 DIM EF(255)
1320 DIM EG(255)
1330 DIM EH(255)
1340 DIM EI(255)
1350 DIM EJ(255)
1360 DIM EK(255)
1370 DIM EL(255)
1380 DIM EM(255)
1390 DIM EN(255)
1400 DIM EO(255)
1410 DIM EP(255)
1420 DIM EQ(255)
1430 DIM ER(255)
1440 DIM ES(255)
1450 DIM ET(255)
1460 DIM EU(255)
1470 DIM EV(255)
1480 DIM EW(255)
1490 DIM EX(255)
1500 DIM EY(255)
1510 DIM EZ(255)
1520 DIM FA(255)
1530 DIM FB(255)
1540 DIM FC(255)
1550 DIM FD(255)
1560 DIM FE(255)
1570 DIM FF(255)
1580 DIM FG(255)
1590 DIM FH(255)
1600 DIM FI(255)
1610 DIM FJ(255)
1620 DIM FK(255)
1630 DIM FL(255)
1640 DIM FM(255)
1650 DIM FN(255)
1660 DIM FO(255)
1670 DIM FP(255)
1680 DIM FQ(255)
1690 DIM FR(255)
1700 DIM FS(255)
1710 DIM FT(255)
1720 DIM FU(255)
1730 DIM FV(255)
1740 DIM FW(255)
1750 DIM FX(255)
1760 DIM FY(255)
1770 DIM FZ(255)
1780 DIM GA(255)
1790 DIM GB(255)
1800 DIM GC(255)
1810 DIM GD(255)
1820 DIM GE(255)
1830 DIM GF(255)
1840 DIM GG(255)
1850 DIM GH(255)
1860 DIM GI(255)
1870 DIM GJ(255)
1880 DIM GK(255)
1890 DIM GL(255)
1900 DIM GM(255)
1910 DIM GN(255)
1920 DIM GO(255)
1930 DIM GP(255)
1940 DIM GQ(255)
1950 DIM GR(255)
1960 DIM GS(255)
1970 DIM GT(255)
1980 DIM GU(255)
1990 DIM GV(255)
2000 DIM GW(255)
2010 DIM GX(255)
2020 DIM GY(255)
2030 DIM GZ(255)
2040 DIM HA(255)
2050 DIM HB(255)
2060 DIM HC(255)
2070 DIM HD(255)
2080 DIM HE(255)
2090 DIM HF(255)
2100 DIM HG(255)
2110 DIM HH(255)
2120 DIM HI(255)
2130 DIM HJ(255)
2140 DIM HK(255)
2150 DIM HL(255)
2160 DIM HM(255)
2170 DIM HN(255)
2180 DIM HO(255)
2190 DIM HP(255)
2200 DIM HQ(255)
2210 DIM HR(255)
2220 DIM HS(255)
2230 DIM HT(255)
2240 DIM HU(255)
2250 DIM HV(255)
2260 DIM HW(255)
2270 DIM HX(255)
2280 DIM HY(255)
2290 DIM HZ(255)
2300 DIM IA(255)
2310 DIM IB(255)
2320 DIM IC(255)
2330 DIM ID(255)
2340 DIM IE(255)
2350 DIM IF(255)
2360 DIM IG(255)
2370 DIM IH(255)
2380 DIM II(255)
2390 DIM IJ(255)
2400 DIM IK(255)
2410 DIM IL(255)
2420 DIM IM(255)
2430 DIM IN(255)
2440 DIM IO(255)
2450 DIM IP(255)
2460 DIM IQ(255)
2470 DIM IR(255)
2480 DIM IS(255)
2490 DIM IT(255)
2500 DIM IU(255)
2510 DIM IV(255)
2520 DIM IW(255)
2530 DIM IX(255)
2540 DIM IY(255)
2550 DIM IZ(255)
2560 DIM JA(255)
2570 DIM JB(255)
2580 DIM JC(255)
2590 DIM JD(255)
2600 DIM JE(255)
2610 DIM JF(255)
2620 DIM JG(255)
2630 DIM JH(255)
2640 DIM JI(255)
2650 DIM JJ(255)
2660 DIM JK(255)
2670 DIM JL(255)
2680 DIM JM(255)
2690 DIM JN(255)
2700 DIM JO(255)
2710 DIM JP(255)
2720 DIM JQ(255)
2730 DIM JR(255)
2740 DIM JS(255)
2750 DIM JT(255)
2760 DIM JU(255)
2770 DIM JV(255)
2780 DIM JW(255)
2790 DIM JX(255)
2800 DIM JY(255)
2810 DIM JZ(255)
2820 DIM KA(255)
2830 DIM KB(255)
2840 DIM KC(255)
2850 DIM KD(255)
2860 DIM KE(255)
2870 DIM KF(255)
2880 DIM KG(255)
2890 DIM KH(255)
2900 DIM KI(255)
2910 DIM KJ(255)
2920 DIM KK(255)
2930 DIM KL(255)
2940 DIM KM(255)
2950 DIM KN(255)
2960 DIM KO(255)
2970 DIM KP(255)
2980 DIM KQ(255)
2990 DIM KR(255)
3000 DIM KS(255)
3010 DIM KT(255)
3020 DIM KU(255)
3030 DIM KV(255)
3040 DIM KW(255)
3050 DIM KX(255)
3060 DIM KY(255)
3070 DIM KZ(255)
3080 DIM LA(255)
3090 DIM LB(255)
3100 DIM LC(255)
3110 DIM LD(255)
3120 DIM LE(255)
3130 DIM LF(255)
3140 DIM LG(255)
3150 DIM LH(255)
3160 DIM LI(255)
3170 DIM LJ(255)
3180 DIM LK(255)
3190 DIM LL(255)
3200 DIM LM(255)
3210 DIM LN(255)
3220 DIM LO(255)
3230 DIM LP(255)
3240 DIM LQ(255)
3250 DIM LR(255)
3260 DIM LS(255)
3270 DIM LT(255)
3280 DIM LU(255)
3290 DIM LV(255)
3300 DIM LW(255)
3310 DIM LX(255)
3320 DIM LY(255)
3330 DIM LZ(255)
3340 DIM MA(255)
3350 DIM MB(255)
3360 DIM MC(255)
3370 DIM MD(255)
3380 DIM ME(255)
3390 DIM MF(255)
3400 DIM MG(255)
3410 DIM MH(255)
3420 DIM MI(255)
3430 DIM MJ(255)
3440 DIM MK(255)
3450 DIM ML(255)
3460 DIM MM(255)
3470 DIM MN(255)
3480 DIM MO(255)
3490 DIM MP(255)
3500 DIM MQ(255)
3510 DIM MR(255)
3520 DIM MS(255)
3530 DIM MT(255)
3540 DIM MU(255)
3550 DIM MV(255)
3560 DIM MW(255)
3570 DIM MX(255)
3580 DIM MY(255)
3590 DIM MZ(255)
3600 DIM NA(255)
3610 DIM NB(255)
3620 DIM NC(255)
3630 DIM ND(255)
3640 DIM NE(255)
3650 DIM NF(255)
3660 DIM NG(255)
3670 DIM NH(255)
3680 DIM NI(255)
3690 DIM NJ(255)
3700 DIM NK(255)
3710 DIM NL(255)
3720 DIM NM(255)
3730 DIM NO(255)
3740 DIM NP(255)
3750 DIM NQ(255)
3760 DIM NR(255)
3770 DIM NS(255)
3780 DIM NT(255)
3790 DIM NU(255)
3800 DIM NV(255)
3810 DIM NW(255)
3820 DIM NX(255)
3830 DIM NY(255)
3840 DIM NZ(255)
3850 DIM OA(255)
3860 DIM OB(255)
3870 DIM OC(255)
3880 DIM OD(255)
3890 DIM OE(255)
3900 DIM OF(255)
3910 DIM OG(255)
3920 DIM OH(255)
3930 DIM OI(255)
3940 DIM OJ(255)
3950 DIM OK(255)
3960 DIM OL(255)
3970 DIM OM(255)
3980 DIM ON(255)
3990 DIM OO(255)
4000 DIM OP(255)
4010 DIM OQ(255)
4020 DIM OR(255)
4030 DIM OS(255)
4040 DIM OT(255)
4050 DIM OU(255)
4060 DIM OV(255)
4070 DIM OW(255)
4080 DIM OX(255)
4090 DIM OY(255)
4100 DIM OZ(255)
4110 DIM PA(255)
4120 DIM PB(255)
4130 DIM PC(255)
4140 DIM PD(255)
4150 DIM PE(255)
4160 DIM PF(255)
4170 DIM PG(255)
4180 DIM PH(255)
4190 DIM PI(255)
4200 DIM PJ(255)
4210 DIM PK(255)
4220 DIM PL(255)
4230 DIM PM(255)
4240 DIM PN(255)
4250 DIM PO(255)
4260 DIM PP(255)
4270 DIM PQ(255)
4280 DIM PR(255)
4290 DIM PS(255)
4300 DIM PT(255)
4310 DIM PU(255)
4320 DIM PV(255)
4330 DIM PW(255)
4340 DIM PX(255)
4350 DIM PY(255)
4360 DIM PZ(255)
4370 DIM QA(255)
4380 DIM QB(255)
4390 DIM QC(255)
4400 DIM QD(255)
4410 DIM QE(255)
4420 DIM QF(255)
4430 DIM QG(255)
4440 DIM QH(255)
4450 DIM QI(255)
4460 DIM QJ(255)
4470 DIM QK(255)
4480 DIM QL(255)
4490 DIM QM(255)
4500 DIM QN(255)
4510 DIM QO(255)
4520 DIM QP(255)
4530 DIM QQ(255)
4540 DIM QR(255)
4550 DIM QS(255)
4560 DIM QT(255)
4570 DIM QU(255)
4580 DIM QV(255)
4590 DIM QW(255)
4600 DIM QX(255)
4610 DIM QY(255)
4620 DIM QZ(255)
4630 DIM RA(255)
4640 DIM RB(255)
4650 DIM RC(255)
4660 DIM RD(255)
4670 DIM RE(255)
4680 DIM RF(255)
4690 DIM RG(255)
4700 DIM RH(255)
4710 DIM RI(255)
4720 DIM RJ(255)
4730 DIM RK(255)
4740 DIM RL(255)
4750 DIM RM(255)
4760 DIM RN(255)
4770 DIM RO(255)
4780 DIM RP(255)
4790 DIM RQ(255)
4800 DIM RR(255)
4810 DIM RS(255)
4820 DIM RT(255)
4830 DIM RU(255)
4840 DIM RV(255)
4850 DIM RW(255)
4860 DIM RX(255)
4870 DIM RY(255)
4880 DIM RZ(255)
4890 DIM SA(255)
4900 DIM SB(255)
4910 DIM SC(255)
4920 DIM SD(255)
4930 DIM SE(255)
4940 DIM SF(255)
4950 DIM SG(255)
4960 DIM SH(255)
4970 DIM SI(255)
4980 DIM SJ(255)
4990 DIM SK(255)
5000 DIM SL(255)
5010 DIM SM(255)
5020 DIM SN(255)
5030 DIM SO(255)
5040 DIM SP(255)
5050 DIM SQ(255)
5060 DIM SR(255)
5070 DIM SS(255)
5080 DIM ST(255)
5090 DIM SU(255)
5100 DIM SV(255)
5110 DIM SW(255)
5120 DIM SX(255)
5130 DIM SY(255)
5140 DIM SZ(255)
5150 DIM TA(255)
5160 DIM TB(255)
5170 DIM TC(255)
5180 DIM TD(255)
5190 DIM TE(255)
5200 DIM TF(255)
5210 DIM TG(255)
5220 DIM TH(255)
5230 DIM TI(255)
5240 DIM TJ(255)
5250 DIM TK(255)
5260 DIM TL(255)
5270 DIM TM(255)
5280 DIM TN(255)
5290 DIM TO(255)
5300 DIM TP(255)
5310 DIM TQ(255)
5320 DIM TR(255)
5330 DIM TS(255)
5340 DIM TT(255)
5350 DIM TU(255)
5360 DIM TV(255)
5370 DIM TW(255)
5380 DIM TX(255)
5390 DIM TY(255)
5400 DIM TZ(255)
5410 DIM UA(255)
5420 DIM UB(255)
5430 DIM UC(255)
5440 DIM UD(255)
5450 DIM UE(255)
5460 DIM UF(255)
5470 DIM UG(255)
5480 DIM UH(255)
5490 DIM UI(255)
5500 DIM UJ(255)
5510 DIM UK(255)
5520 DIM UL(255)
5530 DIM UM(255)
5540 DIM UN(255)
5550 DIM UO(255)
5560 DIM UP(255)
5570 DIM UQ(255)
5580 DIM UR(255)
5590 DIM US(255)
5600 DIM UT(255)
5610 DIM UJ(255)
5620 DIM UV(255)
5630 DIM UW(255)
5640 DIM UX(255)
5650 DIM UY(255)
5660 DIM UZ(255)
5670 DIM VA(255)
5680 DIM VB(255)
5690 DIM VC(255)
5700 DIM VD(255)
5710 DIM VE(255)
5720 DIM VF(255)
5730 DIM VG(255)
5740 DIM VH(255)
5750 DIM VI(255)
5760 DIM VJ(255)
5770 DIM VK(255)
5780 DIM VL(255)
5790 DIM VM(255)
5800 DIM VN(255)
5810 DIM VO(255)
5820 DIM VP(255)
5830 DIM VQ(255)
5840 DIM VR(255)
5850 DIM VS(255)
5860 DIM VT(255)
5870 DIM VJ(255)
5880 DIM VV(255)
5890 DIM VW(255)
5900 DIM VX(255)
5910 DIM VY(255)
5920 DIM VZ(255)
5930 DIM WA(255)
5940 DIM WB(255)
5950 DIM WC(255)
5960 DIM WD(255)
5970 DIM WE(255)
5980 DIM WF(255)
5990 DIM WG(255)
6000 DIM WH(255)
6010 DIM WI(255)
6020 DIM WJ(255)
6030 DIM WK(255)
6040 DIM WL(255)
6050 DIM WM(255)
6060 DIM WN(255)
6070 DIM WO(255)
6080 DIM WP(255)
6090 DIM WQ(255)
6100 DIM WR(255)
6110 DIM WS(255)
6120 DIM WT(255)
6130 DIM WJ(255)
6140 DIM WV(255)
6150 DIM WW(255)
6160 DIM WX(255)
6170 DIM WY(255)
6180 DIM WZ(255)
6190 DIM XA(255)
6200 DIM XB(255)
6210 DIM XC(255)
6220 DIM XD(255)
6230 DIM XE(255)
6240 DIM XF(255)
6250 DIM XG(255)
6260 DIM XH(255)
6270 DIM XI(255)
6280 DIM XJ(255)
6290 DIM XK(255)
6300 DIM XL(255)
6310 DIM XM(255)
6320 DIM XN(255)
6330 DIM XO(255)
6340 DIM XP(255)
6350 DIM XQ(255)
6360 DIM XR(255)
6370 DIM XS(255)
6380 DIM XT(255)
6390 DIM XJ(255)
6400 DIM XV(255)
6410 DIM XW(255)
6420 DIM XX(255)
6430 DIM XY(255)
6440 DIM XZ(255)
6450 DIM YA(255)
6460 DIM YB(255)
6470 DIM YC(255)
6480 DIM YD(255)
6490 DIM YE(255)
6500 DIM YF(255)
6510 DIM YG(255)
6520 DIM YH(255)
6530 DIM YI(255)
6540 DIM YJ(255)
6550 DIM YK(255)
6560 DIM YL(255)
6570 DIM YM(255)
6580 DIM YN(255)
6590 DIM YO(255)
6600 DIM YP(255)
6610 DIM YQ(255)
6620 DIM YR(255)
6630 DIM YS(255)
6640 DIM YT(255)
6650 DIM YJ(255)
6660 DIM YV(255)
6670 DIM YW(255)
6680 DIM YX(255)
6690 DIM YY(255)
6700 DIM YZ(255)
6710 DIM ZA(255)
6720 DIM ZB(255)
6730 DIM ZC(255)
6740 DIM ZD(255)
6750 DIM ZE(255)
6760 DIM ZF(255)
6770 DIM ZG(255)
6780 DIM ZH(255)
6790 DIM ZI(255)
6800 DIM ZJ(255)
6810 DIM ZK(255)
6820 DIM ZL(255)
6830 DIM ZM(255)
6840 DIM ZN(255)
6850 DIM ZO(255)
6860 DIM ZP(255)
6870 DIM ZQ(255)
6880 DIM ZR(255)
6890 DIM ZS(255)
6900 DIM ZT(255)
6910 DIM ZJ(255)
6920 DIM ZV(255)
6930 DIM ZW(255)
6940 DIM ZX(255)
6950 DIM ZY(255)
6960 DIM ZZ(255)
6970 DIM [unintelligible]
6980 DIM [unintelligible]
6990 DIM [unintelligible]
7000 DIM [unintelligible]
```

FIGURA 1 TABLA DE COMANDOS

COMANDOS GENERALES (SIMB. SHFT)	RECEPCION - TRANSMISION	EMISION DE BUFFER Y MENSAJES
(M) BAUDOT - ASCII - MORSE	(1) MENSAJE PROGRAMADO 1	(1) MENSAJE PROGRAMADO 1
(A) BUFFER PROGRAMADO	(2) MENSAJE PROGRAMADO 2	(2) MENSAJE PROGRAMADO 2
(B) BUFFER ENCENDIDO	(3) MENSAJE PROGRAMADO 3	(3) MENSAJE PROGRAMADO 3
(C) CARGA DE BUFFER DESDE CASSETTE	(4) MENSAJE PROGRAMADO 4	(4) MENSAJE PROGRAMADO 4
(D) BORRADO DEL BUFFER	(5) MENSAJE PROGRAMADO 5	(5) MENSAJE PROGRAMADO 5
(E) SALVADO DEL BUFFER A CASSETTE	(6) MENSAJE PROGRAMADO 6	(6) MENSAJE PROGRAMADO 6
(F) VELOCIDAD DE TRANSMISION EN MORSE	(7) MENSAJE PROGRAMADO 7	(7) MENSAJE PROGRAMADO 7
(G) INGRESO DE MENSAJE PROGRAMABLE	(8) MENSAJE PROGRAMADO 8	(8) MENSAJE PROGRAMADO 8
(H) INGRESO DE LICENCIA	(9) MENSAJE PROGRAMADO 9	(9) MENSAJE PROGRAMADO 9



```
100 DIM C(255)
110 DIM S(255)
120 DIM M(255)
130 DIM P(255)
140 DIM R(255)
150 DIM T(255)
160 DIM U(255)
170 DIM V(255)
180 DIM W(255)
190 DIM X(255)
200 DIM Y(255)
210 DIM Z(255)
220 DIM AA(255)
230 DIM AB(255)
240 DIM AC(255)
250 DIM AD(255)
260 DIM AE(255)
270 DIM AF(255)
280 DIM AG(255)
290 DIM AH(255)
300 DIM AI(255)
310 DIM AJ(255)
320 DIM AK(255)
330 DIM AL(255)
340 DIM AM(255)
350 DIM AN(255)
360 DIM AO(255)
370 DIM AP(255)
380 DIM AQ(255)
390 DIM AR(255)
400 DIM AS(255)
410 DIM AT(255)
420 DIM AU(255)
430 DIM AV(255)
440 DIM AW(255)
450 DIM AX(255)
460 DIM AY(255)
470 DIM AZ(255)
480 DIM BA(255)
490 DIM BB(255)
500 DIM BC(255)
510 DIM BD(255)
520 DIM BE(255)
530 DIM BF(255)
540 DIM BG(255)
550 DIM BH(255)
560 DIM BI(255)
570 DIM BJ(255)
580 DIM BK(255)
590 DIM BL(255)
600 DIM BM(255)
610 DIM BN(255)
620 DIM BO(255)
630 DIM BP(255)
640 DIM BQ(255)
650 DIM BR(255)
660 DIM BS(255)
670 DIM BT(255)
680 DIM BU(255)
690 DIM BV(255)
700 DIM BW(255)
710 DIM BX(255)
720 DIM BY(255)
730 DIM BZ(255)
740 DIM CA(255)
750 DIM CB(255)
760 DIM CC(255)
770 DIM CD(255)
780 DIM CE(255)
790 DIM CF(255)
800 DIM CG(255)
810 DIM CH(255)
820 DIM CI(255)
830 DIM CJ(255)
840 DIM CK(255)
850 DIM CL(255)
860 DIM CM(255)
870 DIM CN(255)
880 DIM CO(255)
890 DIM CP(255)
900 DIM CQ(255)
910 DIM CR(255)
920 DIM CS(255)
930 DIM CT(255)
940 DIM CU(255)
950 DIM CV(255)
960 DIM CW(255)
970 DIM CX(255)
980 DIM CY(255)
990 DIM CZ(255)
1000 DIM DA(255)
1010 DIM DB(255)
1020 DIM DC(255)
1030 DIM DD(255)
1040 DIM DE(255)
1050 DIM DF(255)
1060 DIM DG(255)
1070 DIM DH(255)
1080 DIM DI(255)
1090 DIM DJ(255)
1100 DIM DK(255)
1110 DIM DL(255)
1120 DIM DM(255)
1130 DIM DN(255)
1140 DIM DO(255)
1150 DIM DP(255)
1160 DIM DQ(255)
1170 DIM DR(255)
1180 DIM DS(255)
1190 DIM DT(255)
1200 DIM DU(255)
1210 DIM DV(255)
1220 DIM DW(255)
1230 DIM DX(255)
1240 DIM DY(255)
1250 DIM DZ(255)
1260 DIM EA(255)
1270 DIM EB(255)
1280 DIM EC(255)
1290 DIM ED(255)
1300 DIM EE(255)
1310 DIM EF(255)
1320 DIM EG(255)
1330 DIM EH(255)
1340 DIM EI(255)
1350 DIM EJ(255)
1360 DIM EK(255)
1370 DIM EL(255)
1380 DIM EM(255)
1390 DIM EN(255)
1400 DIM EO(255)
1410 DIM EP(255)
1420 DIM EQ(255)
1430 DIM ER(255)
1440 DIM ES(255)
1450 DIM ET(255)
1460 DIM EU(255)
1470 DIM EV(255)
1480 DIM EW(255)
1490 DIM EX(255)
1500 DIM EY(255)
1510 DIM EZ(255)
1520 DIM FA(255)
1530 DIM FB(255)
1540 DIM FC(255)
1550 DIM FD(255)
1560 DIM FE(255)
1570 DIM FF(255)
1580 DIM FG(255)
1590 DIM FH(255)
1600 DIM FI(255)
1610 DIM FJ(255)
1620 DIM FK(255)
1630 DIM FL(255)
1640 DIM FM(255)
1650 DIM FN(255)
1660 DIM FO(255)
1670 DIM FP(255)
1680 DIM FQ(255)
1690 DIM FR(255)
1700 DIM FS(255)
1710 DIM FT(255)
1720 DIM FU(255)
1730 DIM FV(255)
1740 DIM FW(255)
1750 DIM FX(255)
1760 DIM FY(255)
1770 DIM FZ(255)
1780 DIM GA(255)
1790 DIM GB(255)
1800 DIM GC(255)
1810 DIM GD(255)
1820 DIM GE(255)
1830 DIM GF(255)
1840 DIM GG(255)
1850 DIM GH(255)
1860 DIM GI(255)
1870 DIM GJ(255)
1880
```




El número 12 salió finalmente publicado el sintetizador de voz mismo cuyos principios habían empezado a ser delineados en el número 8.

En este básicamente se podía almacenar sonido digitalmente y reproducirlo luego, pero la voz tenía que ser grabada.

Para tener un generador digital a la generación básica de sonidos agregué el algoritmo para codificar los fonemas (copiado), más el algoritmo para reproducirlos (propio) y tenemos un sintetizador de voz casero.

Muchísimos experimentos fueron posibles con este software porque uno simplemente "imprimía" lo que quería que la computadora dijese... La baliza digital "imposible" estuvo en el aire por semanas.....

TS 2068, CZ 2000, TK 90X

SINTETIZADOR DE VOZ

En este artículo se explican los programas que permiten que la computadora "hable" con un vocabulario limitado, eliminando así las dificultades que puede acarrear el almacenamiento de sonidos.

Ing. Pedro E. Colla

Un anterior artículo dedicado al almacenamiento de sonidos, incluyendo voz humana, permitía guardar en memoria segmentos de conversación creando una herramienta mediante la cual se puede hacer "hablar" al computador.
Sin embargo, el mismo no califica como un verdadero sintetizador de voz pues el enganche de los segmentos de habla para formar palabras de un vocabulario más o menos grande es por cierto una tarea ardua.
En este artículo se propone un sintetizador de voz de vocabulario limitado el cual no tiene los inconvenientes antes mencionados.
Tal como se había indicado en el artículo anterior, la voz humana se puede considerar en esencia formada por dos componentes, una señal de frecuencia más o menos alta y de amplitud constante me-



BLA BLA BLA BLA
BLA BLA BLA BLA BLA
BLA BLA BLA BLA BLA
BLA BLA BLA BLA BLA

Cual es el SECRETO del EXITO de STOCK DEVICES?

Asesoramiento y entrega inmediata en todo el país!!
RESET, INTERFACES, JOYSTICK, JUEGOS, FUENTES TRANSFORMADORES
STOCK DEVICES
SERVICIO A SU SERVICIO

OF CENTRAL PJE. DEL CARMEN 716 1° B
ALTAURA CORDOBA 1900
10400 B.S. ARGENTINA
TEL: 42280 (de 0 a 24 hrs)

K64

Pag. 37

PROGRAMAS I

ciada con otra de mucho menor frecuencia pero de amplitud variable.
Hablamos visto en esa oportunidad que, al sacrificar este último componente, introduciendo un grado moderado de distorsión pero sin destruir por completo la inteligibilidad, si segmentamos una conversación pasáramos de una frase a la palabra y luego a la sílaba, cuando segmentamos la sílaba nos encontramos con la unidad de la vocalización, el fonema.
Los fonemas son los bloques básicos con los que formamos nuestra conversación, e, intercambiándolos de distintas maneras, podemos formar nuestro casi limitado vocabulario a pesar que la cantidad de fonemas son relativamente muy pocos.
En efecto, basta un número algo superior a la cantidad de letras de nuestro abecedario para poder formar sin límites cualquier palabra.
La correspondencia entre letras del alfabeto y fonemas no es uno a uno pues hay letras que, de acuerdo al contexto en que se emplean, requieren la utilización de un fonema distinto; basta, a modo de ejemplo, considerar la utilización de la letra "C" en la palabra casa y en la palabra caca para comprender este particular.
Si aislamos cada fonema y observamos la forma de onda que tiene la señal de audio se verá que está formada por combinaciones de señales y silencios que se repiten incesantemente.
Un sintetizador de voz que utiliza fonemas recurre a la representación de esas formas características de señal y a su unión en rápida secuencia para formar palabras. Debido al relativamente reducido

número requerido es práctico el almacenamiento de éstos en la memoria de un computador.
El hecho de requerir formar una señal de audio partiendo de datos almacenados en memoria implica una velocidad de procesamiento que no se puede obtener de ninguna otra forma que no sea utilizando lenguaje de máquina; no importa que tan eficiente sea un intérprete o compilador de cualquier lenguaje de alto nivel (BASIC, PASCAL, o FORTRAN) sencillamente no tendrá la velocidad suficiente.
La señal de audio se forma entonces mediante la conmutación de la salida de MIC (y de parlante) de un computador SPECTRUM o TS2068 al ritmo impuesto por datos correspondientes a cada fonema bajo la supervisión de una rutina en lenguaje de máquina.
El programa se puede apreciar en la figura 1, en el mismo la sección en lenguaje de máquina se almacena en un buffer para facilitar su carga a aquellos que no poseen un programa ensamblador. El programa BASIC almacena el código de máquina en grupos de diez bytes, colocando al comienzo de los mismos la dirección donde deberá cargar el primero de ellos y, al final, la suma de los mismos, de esta manera se facilita notablemente la depuración de eventuales errores de tipo, comprensibles dado la exagerada extensión. El programa sintetizador en el mismo fue extraído de una publicación extranjera y sometido a pequeñas modificaciones francamente cómicas, por parte del autor. El mismo se aloja en la parte superior de la memoria RAM ocupando 2200 bytes a partir de la dirección 63000.

Esta ubicación no es casual, la colocación en otra zona inferior implicaría que las interrupciones que recibe el procesador principal por parte del generador de video se habían perceptibles produciendo ganancia en la salida del sintetizador.
El texto por sintetizar se coloca (puede) en un buffer entre las direcciones 62000 y 62999, casi media hora de conversación, finalizando el con un "0" hexadecimal (CHR(0)); y sólo puede estar compuesto de letras minúsculas, números o símbolos de puntuación) ocasionar que no se prosiga con el remanente a partir de allí.
Esto representa cierta incomodidad en el uso práctico pues al margen de la limitación en cuanto al tipo de caracteres se requiere agregar una pequeña sección de programa BASIC que se encarga de armar el buffer para cada uso del sintetizador.
La solución viene de la mano de una corta rutina en lenguaje de máquina el cual funciona como "driver" para integrar el programa en la arquitectura del computador y permitir acceder al mismo a partir de la instrucción PRINT.
Para lograr esto se modifica el área de variable de sistemas denominadas CHANS reemplazando las direcciones de manejo de dispositivos tipo "P" (Printer) por la dirección de comienzo del driver. Cada vez que se emplee LPRINT en vez de recurrir a la rutina en ROM destinada a enviar el carácter ser enviado al impresor el carácter será enviado a la memoria RAM ocupando 2200 bytes a partir de la dirección anterior para el manejo de una in-

COMPUTER PLACE
S.A.S.
DISTRIBUIDOR DE ZONAS DE DISTRIBUCION

Av. CORRIENTES 1726
400-507 CAP. FED.

Drean & **Commodore**
AGENTE OFICIAL
CZERVENY Sinclair
MICRODIGITAL

PLANES DE FINANCIACION

Pag. 38

K64

presora Centronica.
El driver se encarga de colocar el carácter en el buffer manejando la utilización de letras minúsculas y caracteres especiales; los primeros son transformados en mayúsculas y los segundos son reemplazados por espacios pues, de todas maneras, son impronunciables. Los números son reemplazados por una secuencia de caracteres prevista en una tabla al efecto dentro del driver.
Cuando la sentencia LPRINT no concluye con "0" el último carácter enviado es un 0 (carriage return CHR(13)) el cual al ser recibido implica el llamado al sintetizador en el mismo.
Si se utiliza "0" el programa asume que se quieren enviar caracteres y maneja la situación "recordando" la próxima posición que tiene que utilizar en el buffer.
El driver también está incluido como sentencias DATA en el programa BASIC en la Fig. 1 y para referencia la expresión del mismo en Assembler puede visualizarse en la figura 2 aunque cabe remarcar que este último se incluyó solamente en forma indicativa pues bastará con utilizar las sentencias DATA incluidas en la Fig. 1 para generar el segmento de código de máquina respectivo; al igual que el código del sintetizador en caso de haber problemas se exhibirá durante la carga del código el número de líneas en que se encuentra la sentencia DATA con problemas. Cuando logran cargar totalmente el sintetizador y driver el mismo programa borra todas las sentencias DATA y las secciones de código destinadas a su carga en memoria quedando una versión reducida que se salva en cassette para uso

futuro en la forma de un pequeño programa BASIC y del segmento "CODE" conteniendo respectivamente el sintetizador y el driver.
En la sección superviviente de BASIC queda almacenada la información necesaria para modificar en cada carga el área CHANS de tal manera que al finalizar el sintetizador esté listo para su uso en forma aislada o como parte de otro programa.
Tanto el sintetizador como el driver son utilizables tanto para computadores TS2068 como SPECTRUM, aunque para los usuarios de esta última máquina deberán tenerse en cuenta algunas modificaciones menores; estas son:
1) El área CHANS está ubicada en los computadores SPECTRUM II, generalmente desplazada con respecto a la TS2068 y por lo tanto se deberán modificar las líneas 120 a 150 para reflejar esto, debiéndose insertar en lugar de lo indicado por la figura 1, lo siguiente:
130 POKE 23749,216
130 POKE 23750,214
140 POKE 23751,216
150 POKE 23752,214
2) El SPECTRUM no posee la sentencia DELETE para borrar instrucciones de un programa BASIC, por lo que no podrá ejecutar las líneas 523 a 546 de la Fig. 1, para obtener una versión reducida del sintetizador apta en el uso frecuente deberá borrar "a mano" las líneas indicadas por las instrucciones DELETE y realizar los SAVE de líneas 560 a 570 no como parte del programa sino en forma manual.
Con las pequeñas precauciones derivadas de los puntos anteriores no existe otra dificultad en la utilización de este programa por igual en ambas máquinas.

Por último, nótese que, como parte del programa de carga, se hace decir al sintetizador algunos mensajes.
En los mismos se han utilizado deliberadamente errores de ortografía para mejorar la pronunciación; esto se debe a que solamente se maneja un fonema por cada carácter, debiéndose utilizar con frecuencia este tipo de manejo para obtener una correcta pronunciación. En general será necesario reemplazar la "c" y "z" por "s", la "h" o "q" por "ch" y la "y" por "0", las palabras que emplean con "h" deben ser escritas omitiendo la "h" cuando se requiere pronunciar el conjunto "qu" deberá utilizarse la "q" únicamente.
A pesar de estas precauciones se obtendrá pronunciación algo deficiente en determinados contextos para las letras "y", "j", "q", "i". Las palabras que utilizan "ch", "qu", "ti" o la letra "y" al comienzo no serán pronunciadas naturalmente. Pese a las deficiencias apuntadas y al hecho que de todas maneras la pronunciación obtenida es a menudo difícil de interpretar el proyecto es interesante y obviamente de muy bajo costo.
Por otra parte el lector que se aventure al no poco tedioso tipo necesario notará un efecto de acortamiento al cabo de cierto tiempo de uso que hará que el mismo suene cada vez más natural a medida que pasa el tiempo de uso del mismo, incluyendo la detección de cierta "tonada" española que hace inconducente el origen del programa K64.
Aclaración: las líneas 523 a 546 por un error de impresión omiten el comando "DELETE" el cual debe ser agregado por los lectores.

SERVICIO TECNICO ESPECIALIZADO PARA Commodore 16 - 64 - 128

CONSOLAS # MONITORES
DISKETTERAS # DATASSETTES
IMPRESORAS

RESUPUESTOS EN 24 HS. SIN CARGO
TRABAJOS GARANTIDOS. LABORATORIOS PROPIOS
ATENCIÓN A NEGOCIOS DEL RAMO
TRABAJOS CON EL INTERIOR

LOGIC COMPUTER
RODRIGUEZ PEÑA 431 1° "11"
40000

K64

Pag. 39



Los usos de radio rápidamente empezaron a agotarse, hubo intentos varios de otros y míos de generar SSTV pero eso superaba en mucho las capacidades del hardware; lo mejor que podía obtener era una imagen sin contraste (tipo una foto quemada) y emitir cartelones con letras gigantes porque no era fácil ni económico una fuente de video que pudiera ser tomada por esta máquina.

Sin embargo, como sub-producto de los experimentos, quedó cierta capacidad de conectar a las máquinas de este tipo al mundo analógico; si los requerimientos no eran demasiado exigentes en velocidad había todavía una enorme cantidad de aplicaciones. La idea fue poner al alcance de la mano la conversión A/D y D/A (tampoco no soportada por hardware en estas máquinas) y dejar que cada uno las aplicara como mejor le pareciera.

Y así fue, en los dos siguientes números apareció un convertidor D/A rudimentario y luego un A/D de baja velocidad.

Este número tuvo la curiosidad de ser el primer intento de la editorial de diversificar a sus autores de desarrollos; para entonces K64 era seguida por un público que en buena medida eran experimentadores que compraban a veces el número por los desarrollos que tenía y no por todo el resto. Hasta entonces los artículos técnicos habían sido mayormente de mi autoría.

SUPLEMENTO DE APLICACIONES

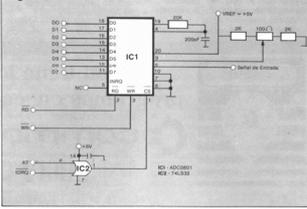
Comenzamos en esta página con un interesante proyecto para las CZ 2000, TK 90X o TS 2068. Luego, continuamos con la nota sobre la interfase paralelo para las TK 83 y CZ 1000, que nos permitirá múltiples usos (por ejemplo, hacer un secuenciador de luces programables). En software educativo, dos excelentes listados para TI 99 Spectrum y compatibles cuyos títulos hablan por sí solos: "Palises, capitales y banderas" y "Huesos" (a no asustarse). Y a los "fans" de MSX les decimos cómo tener permanentemente un reloj en la pantalla cuando pasan largas horas frente al monitor.

CONVERSION ANALOGICO DIGITAL

Presentamos una interfase con una rutina que nos será de gran utilidad en caso de tener que tomar datos externos, tales como temperatura, humedad o iluminación.

La necesidad de controlar dispositivos es casi tan antigua como la ciencia misma. El control automático se basa esencialmente en un dispositivo "sensor" que toma información de la entrega a un segundo elemento denominado "actuador", que es el que efectivamente produce resultados sobre cualquier situación o proceso bajo control. El uso de computadores ha encontrado una masiva aceptación en ese tipo de aplicaciones, dado que se puede proveer de cierta "inteligencia" al sistema de control para tomar acciones en base a las situaciones que predeterminadamente se definen. Los sistemas de control constituyen una rama de la tecnología lo suficientemente vasta, como para exceder ampliamente las posi-

Figura 1



K64

Pág. 27

DESARROLLOS



lidades de su visión, aunque sea superficial, en una publicación de esta índole. Sin embargo, veremos en esta entrega y la siguiente dos elementos de capital importancia para lograr insertar un computador en este entorno.

Nos referimos a los conversores A/D (Analógico/Digital) y D/A (Digital/Analógico). Si bien el computador es extremadamente útil adolece de un defecto de importancia, es digital, y el mundo que nos rodea salvando contadas excepciones no lo es; es en general analógico. ¿Cuál es la diferencia? Las señales digitales pueden tener solamente dos estados: ENCENDIDO y APAGADO ("1" o "0"); mientras que las analógicas pueden tener infinitos estados variando continuamente entre ellos.

Resulta obvio que para que un computador pueda interactuar eficazmente con el mundo exterior deberá poder "interpretar" las señales que de él provienen. El convertidor A/D es, esencialmente, la necesaria interfaz entre ambos mundos.

Su funcionamiento conceptual es

sencillo, no todos los posibles estados que asume una variable analógica son igualmente importantes, normalmente carece de sentido práctico diferenciar entre algunos que están muy próximos entre sí.

Si imaginamos una señal formada de pequeños "escalones" lo suficientemente próximos entre sí, podremos dar una denominación a cada escalón. Asignando una denominación numérica a cada uno no existe, en principio, dificultad para transferir la misma en un número binario el cual puede ser comprendido y evaluado por un computador. La diferencia relativa (porcentual) entre el valor de un escalón y el siguiente se denomina RESOLUCION y determina que tan fiel es la representación numérica de la señal real cuanto más próxima sea esta diferencia a cero más aproximada será la representación de la señal real. Cuanto mayor sea el número de escalones con el que se pueda traducir una señal mayor será el número de señales distintas que se pueden procesar. Supongamos que el computador utiliza 8 bits para representar es-

tados, con este número de bits podrá definir 256 distintos, se dice en este caso que la resolución será del 0.3% ($1/256=0.003$). Si el computador utilizara en cambio 16 bits para la conversión podría definir más de 65500 estados distintos por lo que la resolución sería mucho mejor. La resolución depende entonces de la cantidad de bits involucrados, mayor la cantidad mejor la representación de la señal, al fin de la carrera es el costo; a mayor la cantidad de bits es también mayor el costo.

Para ámbitos como los que pueden ocupar la utilización de un computador hogareño la resolución dada por 8 bits será considerada como suficiente. Existe un segundo factor a ser tenido en cuenta y es la velocidad de la conversión. En efecto, el proceso de encontrar el número que mejor representa un estado analógico toma su tiempo; si la señal varía muy rápidamente no podrá físicamente representarla en forma correcta.

Cuanto menor sea la frecuencia o velocidad de la señal menor costo tendrá el convertidor para procesarla, nuevamente se entiende que a los efectos de las aplicaciones típicas de un computador hogareño, será suficiente la medición de valores continuos o variando con frecuencias en hasta la gama del audio. Un tercer factor a ser considerado es que no todas las señales susceptibles de ser convertidas tendrán una manifestación eléctrica; de hecho la mayoría de las posiblemente interesantes no la tienen (temperatura, humedad, iluminación, etcétera).

¿Quién tiene los mejores programas en cassettes para

commodore 64?

micro cómputo

ACOYTE 44 · Loc. 6 CABALLITO (1405) CAP. FED.

Solicite catálogo. Al interior envíos contra reembolso

Pág. 28

K64

Para salvar este inconveniente se utilizan dispositivos electroquímicos o electromecánicos denominados "transductores", los cuales tienen la habilidad de proporcionar una representación eléctrica de alguna otra variable que no lo es, por ejemplo, un micrófono permite traducir sonidos (ondas de presión) en señales eléctricas que la representan. La necesidad casi continua de realizar este tipo de transformaciones ha derivado en que los conversores A/D fueran migrando desde un conjunto de elementos discretos hasta un único circuito integrado; y posteriormente que el mismo basara su costo a niveles razonables para la utilización cotidiana. En la Fig. 1 se representa una interfaz de conversión A/D apta para ser utilizada por computadores TS2068 o Spectrum, aunque es posible su adaptación a muchos otros mediante conexiones apropiadas a las líneas de bus y la restructura de la rutina de utilización.

D7) una representación digital del valor de la señal analógica. Estando este dispositivo ideado para ser utilizado en el particular entorno del bus de un microprocesador, estas salidas son de tipo TRI-STATE, por lo que solamente se conectan físicamente sobre el bus en determinados momentos que el procesamiento así lo requiere. Para que ello ocurra el procesador debe indicarlo colocando las líneas CS y RD en bajo. Entre el momento que se inicia la conversión y la misma finaliza pasa un lapso de tiempo dado por la velocidad del convertidor, dado que todo el proceso entre ambos buses en determinados momentos el control del computador debe existir alguna forma mediante la cual el mismo se "entere" que ya existe alguna forma mediante la cual el mismo se "entere" que ya existe un valor convertido que debe ser leído. Para ello se utiliza la línea denominada INTR (Interrupt Request), la misma está ideada para que se

se le deba afectar por un factor de escala para obtener la dimensión final de la señal analógica. La misma dependerá del valor de la "referencia" del convertidor que en este caso será de +5V. Esto implica que solamente se podrán convertir señales entre +0 y +5V con esta interfaz, el circuito acepta, sin inconvenientes referencias de hasta +12V, por su condición de marco de medición la tensión de referencia deberá ser una tensión de continua pura y estabilizada. Para obtener el valor de la conversión se deberá realizar la siguiente operación con lo que devuelve la rutina. $Tensión (V) = \frac{[result\ de\ conversión] \times [Ref.]}{255}$ Ejemplo: si el resultado de la conversión es 120, esto significa que la señal a la entrada tenía $+5 \times \frac{120}{255} = 2.5\ V$ Para utilizar este enfoque debe tenerse en cuenta que la señal no podrá variar más rápidamente que lo que requiere el BASIC para realizar la anterior operación.

Software para interfaz A/D

00000	00120	00200	00280
00010	00130	00210	00290
00020	00140	00220	00300
00030	00150	00230	00310
00040	00160	00240	00320
00050	00170	00250	00330
00060	00180	00260	00340
00070	00190	00270	00350
00080	00200	00280	00360
00090	00210	00290	00370
00100	00220	00300	00380
00110	00230	00310	00390
00120	00240	00320	00400
00130	00250	00330	00410
00140	00260	00340	00420
00150	00270	00350	00430
00160	00280	00360	00440
00170	00290	00370	00450
00180	00300	00380	00460
00190	00310	00390	00470
00200	00320	00400	00480
00210	00330	00410	00490
00220	00340	00420	00500

La misma se inserta en la arquitectura del computador como un port de I/O, aprovechando una de las direcciones que al efecto provee el Z80; nótese también que el direccionamiento es el mismo que el utilizado para otros proyectos anteriores por lo que esto no podrá ser utilizado al mismo tiempo que ellos (interfaz de printer y de comunicaciones). El integrado IC1 realiza todas las funciones necesarias de conversión. Para comenzar el procesamiento debe colocarse simultáneamente las líneas WR y CS en bajo (Pines 3 y 1) con lo que se dispara el proceso de comparación; el algoritmo utilizado internamente por el dispositivo permite que con relativa velocidad se tenga disponibles en las líneas de salida (D) a

utilice en la generación de una interrupción al procesador; su inserción en un computador que utiliza esta facilidad para otros propósitos no es sencilla. Sin embargo, existe otro método menos flexible pero más sencillo y adecuado en la mayoría de los casos, el mismo consiste en que luego de ordenar la iniciación de la conversión el procesador espere una cantidad fija de tiempo y luego de ello lea el resultado. En la Fig. 2 se puede apreciar una corta rutina en lenguaje de máquina que sirve para controlar esta interfaz; las restricciones de velocidad impiden la utilización de BASIC para este propósito. Esta rutina devolverá al BASIC un número entre 0 y 255 que será el resultado de la conversión; este

Nótese por último que el programa se arma por encima de la dirección 32768 y que las interrupciones se desconectan mientras el procesador "espera" el resultado, esto tiene por objetivo el mejoramiento del "timing" de la generación de video. La construcción podrá encararse mediante técnicas de wire-wrapping o soldado punto a punto; y la conexión física sobre el computador se hará a través de un conector apropiado al peine de expansión. Debe tenerse precaución en no exceder con la señal a convertir el rango mencionado previamente por las consecuencias que ello acarrearía sobre el computador.

Pág. 29

K64

MAYO 1988 N° 14 \$ 2.30 REP. ARGENTINA

K64

COMPUTACION PARA TODOS

Argentina

Siglo 21

15 Programas

Inéditos

Interfase Kempston

Para Spectrum

Suplemento Educativo Para

C64, TI99, CZ, TK y MSX

CPM en la C-128

Cómo Ganar Un Lingote de Oro



DESARROLLOS

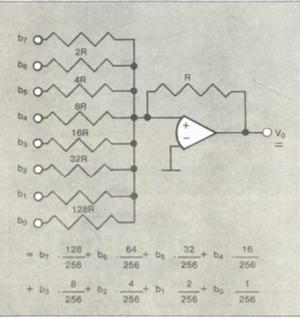
CONVERSOR ANALOGICO DIGITAL

Parte II Ing. Pedro E. Colla



El principio mediante el cual se lleva a cabo este proceso es esencialmente el mismo que vimos anteriormente para el caso del convertidor R/D; sólo que en sentido contrario.

Figura 1 Sumador Analógico



Supongamos que cada bit lleva asociado un determinado valor, fijo, en la formación del valor de voltaje final. Si se quiere expresar un número cualquiera entre 0 y 255, digamos 247, podríamos hacer lo siguiente:

$$\begin{aligned} \text{bit } 7 \times 128 &= 128 \\ \text{bit } 6 \times 64 &= 64 \\ \text{bit } 5 \times 32 &= 32 \\ \text{bit } 4 \times 16 &= 16 \\ \text{bit } 3 \times 8 &= 0 \\ \text{bit } 2 \times 4 &= 4 \\ \text{bit } 1 \times 2 &= 2 \\ \text{bit } 0 \times 1 &= 1 \\ \hline &= 247 \end{aligned}$$

Si, por ejemplo, dividiéramos el rango posible en un número de estados, digamos 256, cada estado tendrá una diferencia con sus contiguos de 1/256 parte del total del rango (en nuestro ejemplo 19 mV). Esta separación podrá ser mayor o menor que la citada, agregando mayor o menor precisión solamente, pero no produciendo una variación del concepto.

El número elegido de divisiones es en especial simpático, pues no es ni más ni menos que la cantidad de posibles estados diferentes que se pueden contar mediante 8 bits. Esto es sumamente adecuado, teniendo en cuenta que es la cantidad de bits que disponen los computadores que normalmente tratamos en esta serie. La forma en que se relaciona un número en binario con una señal analógica puede explicarse en mu-

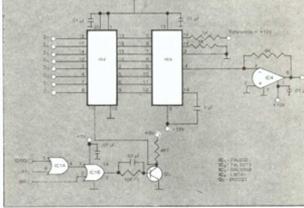
En la anterior entrega vimos sólo una parte de la comunicación de un computador con el mundo analógico que nos rodea, la conversión Analógico-Digital. En esta veremos la contrapartida, es decir, la forma en la cual un computador puede generar señales analógicas.

chas formas, algunas más complicadas que otras; intentaremos hacerlo con la ayuda de un ejemplo. Si se quiere expresar un número cualquiera entre 0 y 255, digamos 247, podríamos hacer lo siguiente:

La expresión binaria de 247 (11110111) ha ocupado 8 dígitos binarios (bits), a propósito BIT deriva de Binary digit, o sea, dígito binario, y se ha formado dando un "peso" a cada uno de ellos de acuerdo a su posición; de esa manera el bit 7 "pesará" 128 sobre el total cuando esté en "1" y nada cuando valga "0".

Nótese que cada bit lleva asociado un determinado valor, fijo, en la formación del valor de voltaje final. Si se quiere expresar un número cualquiera entre 0 y 255, digamos 247, podríamos hacer lo siguiente:

Figura 2 Interfaz D/A Mapped I/O # 3F



La expresión binaria de 247 (11110111) ha ocupado 8 dígitos binarios (bits), a propósito BIT deriva de Binary digit, o sea, dígito binario, y se ha formado dando un "peso" a cada uno de ellos de acuerdo a su posición; de esa manera el bit 7 "pesará" 128 sobre el total cuando esté en "1" y nada cuando valga "0".

Visito de otra forma, el bit 7 lleva consigo la 128/256 parte de un número cualquiera, el bit 6 la 64/256 parte y así sucesivamente. Si aplicamos este mismo criterio a la transformación de un número binario en una señal analógica y logramos establecer el mismo esquema anterior, dado un rango de 0 a 5 V el número 247 representará lo que vemos en el cuadro 1.

INPUT DATA CLUB Santa Fe 1670 - Loc. 45

Drean Commodore MICRODIGITAL TK85 - 90X

PLANES DE FINANCIACION

LIBROS - PROGRAMAS - FUNDAS - ACCESORIOS

JUMBO SUCURSAL CENTRO COMERCIAL

ACEPTAMOS TARJETAS DE CREDITO AV. CRUZ 4602 (y Escalada) SARAGOS T DOMINGOS ABIERTO de 9.30 a 22 hs. VISTENES

K64

Pág. 55

Pág. 56

DESARROLLOS

nes de cada uno a la suma total estarán controlados por los valores de las resistencias; las cuales mantienen una relación entre sí de tal manera que cada bit tenga una incidencia sobre el resultado final, acorde con lo visto en el ejemplo anterior.

Al igual que en el caso del convertidor A/D la resolución, o sea la diferencia mínima que se puede percibir, estará dada únicamente por los bits con que se quiere efectuar la representación o conversión. Sin embargo, debe tenerse en cuenta que el esquema de conversión simple que se ha expuesto no es sencillo de mantener para resoluciones superiores a las obtenidas con 8 bits, pues en estos casos la contribución de los bits menos significativos es tan baja que empieza a confundirse con el ruido en un circuito práctico. El método, muy útil como base para la comprensión del tema, aduce de algunos inconvenientes de tipo constructivo si se desea mantener cierta precisión. Afortunadamente, se disponen de

circuits integrados que realizan la tarea de conversión con un mínimo de dificultad constructiva. En la figura 2 puede observarse una simple, pero efectiva, interfaz D/A que utiliza uno de ellos.

La misma está pensada para ser utilizada con microsincalar (TS 2066, SPECTRUM, etcétera) aunque su adaptación para otras máquinas no ofrecerá dificultad. Como conocemos, IC1 constituye la "lógica de direccionamiento" y permite que el procesador entregue información al convertidor bajo condiciones controladas, estando la interfaz concebida como I/O Mapped, esto ocurrirá con las instrucciones INOUT del Z80. El integrado IC2 tiene por propósito el actuar como "memoria" del último valor enviado por el procesador, a la par que, al aislar al convertidor de la actividad sobre el bus cuando no se trata de una operación que lo afecta. El convertidor D/A en sí mismo (IC3) funciona convirtiendo la señal que se le expresa digitalmente en su equivalente analógico y utilizando

como rango máximo el de su alimentación. La conversión ocupa un cierto tiempo, pero a todos los efectos prácticos se puede asumir que es instantánea. La interfaz en su conjunto podrá ser utilizada mediante un programa BASIC, pero con la consabida limitación de velocidad. Si se quisiera generar señales rápidamente se deberá recurrir a una corta rutina en lenguaje de máquina como la expuesta en la figura 3. La misma, cuyo propósito es solamente el orientar al lector y el que le sirva de base para cualquier otra función, asume la existencia de una tabla de valores que se deben reproducir mediante el uso de la interfaz.

El comienzo y fin de la tabla estarán dados respectivamente por START y END. Entre valor y valor se introduce un retardo, variándolo se podrá establecer la velocidad de recorrido de la tabla (o visto de otra forma la frecuencia de la señal generada). Una vez invocada la rutina, se ejecuta sin solución de continuidad, previniéndose la detención de la ejecución con la presión de la tecla "break". El armado físico del circuito es lo suficientemente sencillo como para ser llevado a cabo, sin dificultad, sobre una plaqueta experimental mediante "wire-wrap". La rutina de software asociada podrá ingresarse y experimentarse mediante el uso de un programa Emulador, siendo el Zeus-Assembler indicado para ello.

Cuadro 1

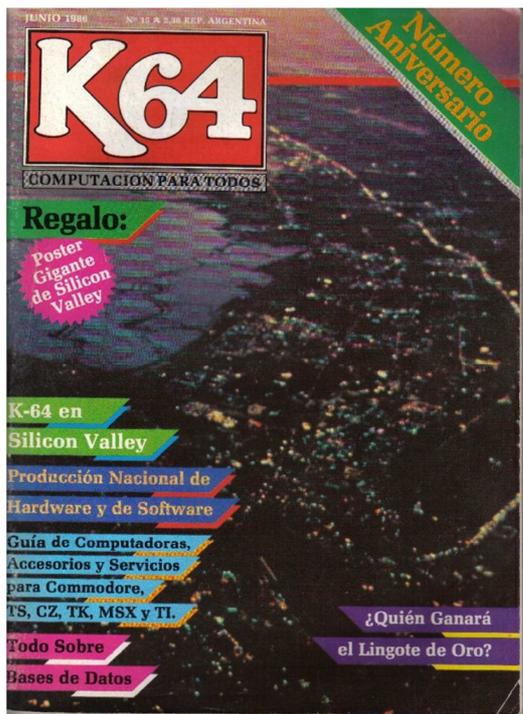
bit 7	1 x 128	=	128	/	256	=	25	V
bit 6	1 x 64	=	64	/	256	=	12,5	V
bit 5	1 x 32	=	32	/	256	=	6,25	V
bit 4	1 x 16	=	16	/	256	=	3,125	V
bit 3	0 x 8	=	0	/	256	=	0,0	V
bit 2	1 x 4	=	4	/	256	=	0,078125	V
bit 1	1 x 2	=	2	/	256	=	0,0390625	V
bit 0	1 x 1	=	1	/	256	=	0,01953125	V
								247

```

0001 / DIGITAL
0002 /
0003 /
0004 / RUTINA PARA ENTREN
0005 / PREVIENE INTERFER D/A
0006 / EL CONTENIDO DE INH
0007 / TAMBIEN RESISTENTE EN
0008 / MEMORIA
0009 /
0010 / INC 00000
0011 /
0012 / BEGIN
0013 / START DEF 000,000
0014 / END DEF 000,000
0015 / DELAY DEF 000,000
0016 /
0017 / LOS VALORES INTERIORES
0018 / CONTIENE EL FUNCIONAMIENTO
0019 / DE LA RUTINA Y DEBE SER
0020 / LETENIDO A VALORES
0021 / PARTI CULARES PREVIENIENTE
0022 / A SU RELECCION
0023 /
0024 / START - DIRECCION DEL
0025 / CONTENIDO DE LA
0026 / TAMBIEN CON LOS
0027 / VALORES A ENTREN
0028 / END - DIRECCION DE FIN
0029 /
0030 / DELAY - RETARDO ENTRE
0031 / VALORES
0032 / INHABILITA INTERFERUNCIONES
0033 / RUTINA PRENCIPAL
0034 / INHABILITA INTERFERUNCIONES
0035 /
0036 /
0037 /
0038 / LD INH,(START)
0039 /
0040 / TOR VILDR Y ENTRE POR LA
0041 / INTERFER
0042 /
0043 /
0044 /
0045 /
0046 /
0047 /
0048 /
0049 /
0050 /
0051 /
0052 /
0053 /
0054 /
0055 /
0056 /
0057 /
0058 /
0059 /
0060 /
0061 /
0062 /
0063 /
0064 /
0065 /
0066 /
0067 /
0068 /
0069 /
0070 /
0071 /
0072 /
0073 /
0074 /
0075 /
0076 /
0077 /
0078 /
0079 /
0080 /
0081 /
0082 /
0083 /
0084 /
0085 /
0086 /
0087 /
0088 /
0089 /
0090 /
0091 /
0092 /
0093 /
0094 /
0095 /
0096 /
0097 /
0098 /
0099 /
0100 /
0101 /
0102 /
0103 /
0104 /
0105 /
0106 /
0107 /
0108 /
0109 /
0110 /
0111 /
0112 /
0113 /
0114 /
0115 /
0116 /
0117 /
0118 /
0119 /
0120 /
0121 /
0122 /
0123 /
0124 /
0125 /
0126 /
0127 /
0128 /
0129 /
0130 /
0131 /
0132 /
0133 /
0134 /
0135 /
0136 /
0137 /
0138 /
0139 /
0140 /
0141 /
0142 /
0143 /
0144 /
0145 /
0146 /
0147 /
0148 /
0149 /
0150 /
0151 /
0152 /
0153 /
0154 /
0155 /
0156 /
0157 /
0158 /
0159 /
0160 /
0161 /
0162 /
0163 /
0164 /
0165 /
0166 /
0167 /
0168 /
0169 /
0170 /
0171 /
0172 /
0173 /
0174 /
0175 /
0176 /
0177 /
0178 /
0179 /
0180 /
0181 /
0182 /
0183 /
0184 /
0185 /
0186 /
0187 /
0188 /
0189 /
0190 /
0191 /
0192 /
0193 /
0194 /
0195 /
0196 /
0197 /
0198 /
0199 /
0200 /
    
```

K64

Pág. 57



Las aplicaciones en radio llenaron casi completamente mi espacio de experimentación con estas máquinas, pero en forma paralela los diseños tuvieron otros usos. Curiosamente en aplicaciones académicas y científicas. Para la época las PC no estaban difundidas y las terminales de consulta de datos (así se llamaban) eran máquinas propietarias que uno utilizaba para consultar bases de datos con información científica. Al mismo tiempo la entonces floreciente actividad de BBS (precursores de Internet en muchos sentidos) hacía necesario contar con una máquina capaz de intercambiar información digitalmente. Para eso se usaban comunicaciones serie, a otra velocidad (300 bps y en ocasiones 1200 bps en lugar de 45, y en ASCII en lugar de BAUDOT) pero esencialmente igual a los usos de RTTY para radioaficionados.

Varios pioneros tales como Alberto Barengols, Adolfo Galanternik, Adolfo D'Onofrio y tantos otros pusieron al servicio de investigaciones científicas diseños que habían sido originalmente concebido para radio; de alguna manera la Red científica nacional antes del internet fue hecha alrededor de este tipo de diseños.

Parte de este uso se refleja en el artículo, donde se muestra como acceder a DIALOG (una base de datos para buscar y extraer información más de una década antes del primer precursor de Google).

COMUNICACIONES

TODO SOBRE BASES DE DATOS

PARTE I
Ing. Pedro E. Colla

Las comunicaciones activas al uso de computadores tienen dos modalidades más o menos definidas: la comunicación pasiva y la comunicación activa.

En la primera de ellas existe un flujo de datos solamente unidireccional, sin la menor interacción directa con quien recibe el mismo, un ejemplo de ello constituyen las noticias o reportajes por agencias noticiosas.

En el segundo caso, la comunicación es realmente bidireccional y el contenido de la misma depende fuertemente de las acciones tomadas desde ambos puntos del enlace.

El soporte desde el punto de vista tecnológico para ambos modos es muy similar y consiste en la transformación de las señales digitales a ser transmitidas en tonos de audio, las que son luego transportadas a través de la distancia por medio de líneas tales como radio, red telefónica, conexión punto a punto, etcétera.

Para los enlaces pasivos, sólo se requiere una vía de comunicación y, por lo tanto, se asigna en forma más o menos estándar un juego de tonos, de tal manera que una o más presente la marca "0" y el otro el espacio "1", cuando un estado se presenta en esta característica se lo denomina **half-duplex**.

Para los enlaces activos es más común la utilización del enlace tipo **FULL-DUPLEX**, siendo así requerido durante el momento de transmisión simultánea de información en ambas direcciones.

Para ello, a uno de los interlocutores se lo denomina **interrogador** (originating) y al restante **contestador** (answering) y se asigna a cada uno de ellos un par de tonos propios para representar sus propias secuencias de bits, lo suficientemente separadas entre sí como para ser fácilmente diferenciables y no suficientemente próximas como para que ambos entren en el ancho de banda de un canal telefónico convencional.

Ambos extremos deben ponerse de acuerdo en varias cosas antes de poder intercambiar el primer bit, de allí la necesidad y popularidad de ciertos estándares.

El primero de ellos y más obvio es la

frecuencia que asigna cada uno a su "marca" y "espacio".

El segundo es el número de bits por código que utilizarán para intercambiar información.

Por último, es necesario acordar en forma previa quién hará el papel de "ANS-ORIGINATE" y quién el de "ANS-ANSWER".

Por supuesto, dos personas interesadas en intercambiar datos pueden elegir para su enlace cualquier combinación de los anteriores parámetros en forma arbitraria, sin embargo, la necesidad de acordar "de facto" las necesidades de comunicación entre muchos interesados sin previo acuerdo entre los mismos condujo a la adopción de estándares, dos de los más exitosos son el BELL 103 y el más reciente CCITT V.21.

Estos coinciden en la velocidad (300 bauds) del número de bits (8) y el código de utilizar (ASCII), se diferencian en las bandas de frecuencia que asigna cada uno de ellas a cada interlocutor, en la Figura 1 puede observarse una tabla con las mismas.

Figura 1 - Normas BELL 103 y CCITT

Norma	Frecuencia Original		Frecuencia Answer	
	Marca	Espacio	Marca	Espacio
BELL 103	1070 Hz	1270 Hz	2025 Hz	2195 Hz
CCITT V.21	980 Hz	1180 Hz	1650 Hz	1820 Hz

El uso ha establecido la convención de asignar a quien llama las características del modo interrogador u originante y a quien contesta el de respondedor o answer.

Sobre esta base de las características de comunicación bidireccional se interacciona inspeccionar las posibilidades que la misma tiene.

En nuestro país, la explotación informática es relativamente reciente y escasas las actividades de comunicación de datos "personales".

En USA y Europa se han popularizado dos modalidades, los boletines y los servicios de información online.

Los boletines o CRMS (Computer

Bulletin Board System) están formados por computadores capaces de recibir llamados desde usuarios (en modo ORIGINATE) a través de una o más líneas telefónicas domésticas dedicadas y su función primordial es el almacenamiento de mensajes dirigidos a otros usuarios o grupos de usuarios.

Esta modalidad, de filosofía esencialmente "doméstica", consiste en permitir la conexión de un computador tipo PC o aún de tipo hogareño a la red telefónica (modem apropiado mediante) y el acceso en forma remota al mismo.

Los usuarios pueden transmitir mensajes con un destinatario predeterminado o de interés común, la primera categoría solamente podrá ser leída en forma posterior cuando a su vez acceda el destinatario y la segunda puede ser inspeccionada por todo aquel que se interesa.

Por lo general la capacidad de conferencia de un CBBS, la capacidad de establecer un "diálogo" accediendo al mismo tiempo al dispositivo y a través del mismo, suele ser limitada o inexistente.

Volvemos en una entrega posterior sobre los CBBS, exploramos las características de un software para su implementación.

El segundo grupo de posibilidades constituyen los servicios de información online.

Los mismos han adquirido volumen y estructura comercial, siendo en consecuencia el servicio de características más profesionales que los provistos por los CBBS.

Básicamente, los mismos consisten en gigantescos servidores de información, estructuralmente organizados, la cual puede ser recuperada a través de "diálogos" que toman la

Los usuarios de microcomputadoras en la mayoría de los casos, no están familiarizados con el tema de las comunicaciones. En números anteriores hemos hablado de las características técnicas específicas que hacen posible este tipo de enlaces. Veremos aquí claramente la terminología usada y su significado, el uso real y corriente de este tipo de medios y su adaptación a las home computers.

Figura 2 - Ejemplo de Acceso a DIALOG

```

BEGIN 275
(Información de LOGON)
SELECT AI AND PY=1885
3450 AI
25324 PY=1985
1 234 AI AND PY=1985
SELECT FINANCE
2 5420 FINANCE
SE puede combinar ambos sets para lograr el resultado deseado.
3 1 AND 2
DISPLAY 3/61
(Abstract of Artículo)
PRINT 3
EXIT
DIALOG puede otros formatos del proveedor de la información, tales como RETRIEVE, GUAR, SAVE, INDEX, PRINT, NEWS, VIEW y DISPLAY.
El formato de la selección y el comando al mismo tiempo puede consistir de:

```

forma de menú o a través de software de búsqueda (query software). Adicionalmente, algunos de los más populares permiten no sólo efectuar búsquedas sino también tomar acciones concretas tales como efectuar compra (teleshop) o reservar plazas en vuelos.

El número de estos servicios a nivel mundial excede los 4000 y el propósito de los mismos abarca un espectro muy amplio, desde los dedicados a proporcionar información sobre hitos científicos muy especializados hasta los de propósito general, por ejemplo capacidad de teleconferencia y correo electrónico.

A diferencia de los CBBS cuyo acceso suele no estar cargado con otro costo que el derivado de la llamada telefónica para accederlo, los servicios Online requieren de una suscripción a los mismos y cargos por cada vez que se los utiliza.

Desafortunadamente, no hay servicios masivos de esta naturaleza en nuestro país, aunque algunos de los extranjeros pueden ser accedidos a través del Nodo Internacional provisto por ENTEL para comunicación de datos.

La ventaja de utilizar redes de comunicación para el acceso reside en que se accede a ellas a través de "gateways" locales por medio del servicio telefónico normal, siendo el resto del trayecto canalizado a través de redes especiales cuyo costo de uso es más bajo básicamente debido a que el cargo por su uso se mide en términos de la información intercambiada (pacquetes) y no del tiempo de conexión.

Entre los servicios masivos, los más utilizados en USA son CompuServe y The Source, los que lamentablemente no pueden suscribirse desde nuestro país.

Desde nuestro país podemos, sin embargo, suscribirnos y utilizar uno de los más interesantes servicios disponibles, el que se denomina DIALOG Information Services.

El mismo, perteneciente a la compañía Lockheed, es en realidad un di-



COMUNICACIONES

gigantesco conglomerado de bancos de datos el cual contiene alrededor de 80 millones de registros que cubren todos los aspectos de la ciencia, educación, negocios, leyes, finanzas, artes, medicina y ciencias sociales.

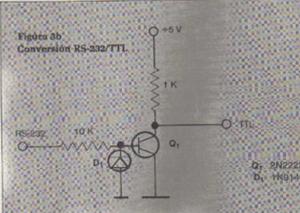
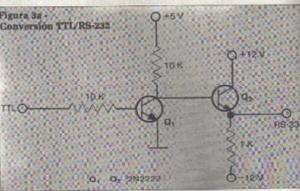
La información es accedida a través de comandos convencionales y se refiere fundamentalmente a material publicado sobre las áreas mencionadas, pudiendo acceder en forma Online a un resumen de las mismas. Cuando la búsqueda tiene éxito el documento completo puede ser volcado directamente a través de la terminal o a solicitud impreso y remitido por correo.

El costo global de uso desde nuestro país es de aproximadamente \$40 la hora de conexión, más un cargo inicial de suscripción del mismo orden, no todos los datos tienen el mismo "precio" siendo algunos mucho más baratos que otros (existen hasta ofertas por tiempo limitado sobre el acceso a ciertos bancos de datos), e inclusive ciertos bancos tienen precio preferencial para establecimientos educativos o de investigación.

En la figura 2 se expone un ejemplo de búsqueda de datos sobre Dialog que será instructivo al lector no familiarizado con esta modalidad, en la misma se puede apreciar la relativa sencillez de manejo una vez que se conocen los comandos básicos y una escasa docena de reglas y convenciones (codificación de los formatos, numeración de búsquedas, etcétera).

En general, para poder utilizar tanto los CBBS como los servicios online no se requieren sofisticados equipos, bastará una terminal o una computadora que actúe como tal, un modem de la norma apropiada y la capacidad de acceso a una línea telefónica.

Algunos computadores vienen provistos de ports serie los que permiten la utilización de los mismos para comunicaciones (Commodore 64), en otros casos será necesario el agregado de una interfaz con este propósito, al respecto en entregas anteriores de K64 se indicó el hardware y software necesario para efectuar estas tareas con una TS 2068 (**), el software provisto en aquella oportunidad estaba orientado a su uso en enlaces HALF-DUPLEX pero su modificación para funcionar en FULL-DUPLEX (recibir y transmitir el mismo tiempo) puede ser llevada a cabo con muy pocas modificaciones.



Los modems comerciales requieren su alimentación desde una salida RS232, para quienes dispongan una salida tipo TTL, en la figura 3 se exponen un grupo de sencillos circuitos de conversión entre ambas normas.

Los modelos básicos de modems comerciales son suficientes para cubrir las necesidades sobre este particular.

El tema abordado es lo suficientemente vasto como para que el presente artículo no tenga otro propósito que el de brindar una muy necesaria introducción al mismo.

Por lo común, dado el carácter comercial de los mismos, los servicios online brindan detallada información sobre su utilización y posibi-

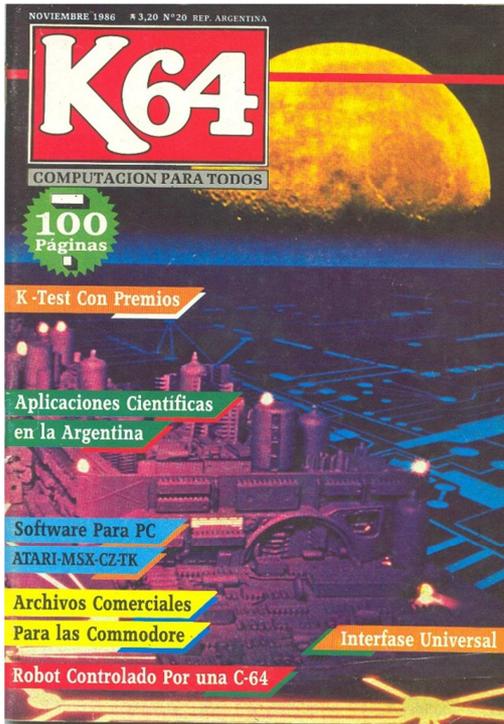
dades a punto de lo requiere tanto desde el punto de vista técnico como operativo.

Respecto a los CBBS brindaremos en una próxima entrega las características de funcionamiento y uso de los mismos así como algunas bases para la instalación de una versión sencilla de un dispositivo de este tipo.

Los términos "marca" y "espacio" son mucho más antiguos que la comunicación y demer del nombre que se le asigna (y asigna) a la presencia o ausencia de pulso en los sistemas de comunicaciones telefónicas y radiotelegráficas, el término fue posteriormente adaptado por radio-telepelo y luego "redefinido" por la terminología de comunicaciones de datos.

*K64 Nov. 9, 10 y 11.





Y todo llega a su fin....

Por un lado tenía por la época crecientes demandas profesionales que incluso habían forzado a que espaciara los artículos, por el otro K64 tomó la decisión editorial que no pagarían más por los artículos. En realidad los honorarios por artículo eran muy bajos, casi un viático, pero que seguía siendo justo. De común acuerdo se interrumpió la serie.

Curiosamente la ultima edición omitieron la firma y apareció el artículo sin autor.

La revista duró aproximadamente un año mas, pero luego fue irreversiblemente superada por la tecnología PC y sus publicaciones especializadas.

DESPROTEGIENDO SOFTWARE

PARA VENCER A LOS TURBOS

Ing. Pedro E. Colla

Sin querer ser "piratas", lo cierto es que es necesario obtener copias de superioridad de los programas para la eventualidad de que un accidente los destruya (para TS 2068 y Spectrum).



La copia de software viene siendo centro de una intensa polémica, tanto en nuestro país, que carece de software houses de importancia y de actuaciones fuertes de usuarios.

efectos es un tema lo suficientemente extenso y complejo como para que escape al tratamiento en esta nota. Así tanto como para, quizás, generar un artículo dedicado al tema en el futuro.

Figura 1

00018	COPY	00020	VERIFICAR SI ES BREAK	00718	OUT (RFE),R
00022	COPY	00022	NO	00720	REPETIR EL CICLO
00024	PROGRAMA PARA COPY	00024	NO NC,BREAK	00740	INDEX INDEPENDIENTE
00026	DE SOFTWARE SIN	00040		00760	JP PRIN
00028	LINKACION DE METADO	00410	SELECCIONAR BIT 6	00780	RUTINA DE RETORNO
00070	DE grabacion	00420	CONTROL DE HEAD UN	00780	CONTROL DE ESTE EN
00080	ONG 50000	00440	INTEGRIACION PRESENTANTE	00800	RU INTERIOR POSICION
00100	ANT	00460	REINTEGRIACION EN BIT 6	00820	BREAK LD SP,(STACK)
00110	PROGRAMA PRINCIPAL	00470	REP	00840	HABILITAR INTERRUPT
00120	EJECTOR UN RETORNO	00490	REP	00860	BI
00130	LD SP,SPFF	00490	REP	00880	RETORNO R BRBIC
00140	LD SP,SPFF	00490	REP	00900	RET
00150	LD SP,SPFF	00490	REP	00920	STACK RFP,00,000
00160	OR C	00490	REP	00920	END NOP
00170	OR C	00490	REP		
00180	OR C	00490	REP		
00190	OR C	00490	REP		
00200	OR C	00490	REP		
00210	OR C	00490	REP		
00220	OR C	00490	REP		
00230	OR C	00490	REP		
00240	OR C	00490	REP		
00250	OR C	00490	REP		
00260	OR C	00490	REP		
00270	OR C	00490	REP		
00280	OR C	00490	REP		
00290	OR C	00490	REP		
00300	OR C	00490	REP		
00310	OR C	00490	REP		
00320	OR C	00490	REP		
00330	OR C	00490	REP		
00340	OR C	00490	REP		
00350	OR C	00490	REP		
00360	OR C	00490	REP		
00370	OR C	00490	REP		
00380	OR C	00490	REP		
00390	OR C	00490	REP		
00400	OR C	00490	REP		
00410	OR C	00490	REP		
00420	OR C	00490	REP		
00430	OR C	00490	REP		
00440	OR C	00490	REP		
00450	OR C	00490	REP		
00460	OR C	00490	REP		
00470	OR C	00490	REP		
00480	OR C	00490	REP		
00490	OR C	00490	REP		
00500	OR C	00490	REP		
00510	OR C	00490	REP		
00520	OR C	00490	REP		
00530	OR C	00490	REP		
00540	OR C	00490	REP		
00550	OR C	00490	REP		
00560	OR C	00490	REP		
00570	OR C	00490	REP		
00580	OR C	00490	REP		
00590	OR C	00490	REP		
00600	OR C	00490	REP		
00610	OR C	00490	REP		
00620	OR C	00490	REP		
00630	OR C	00490	REP		
00640	OR C	00490	REP		
00650	OR C	00490	REP		
00660	OR C	00490	REP		
00670	OR C	00490	REP		
00680	OR C	00490	REP		
00690	OR C	00490	REP		
00700	OR C	00490	REP		
00710	OR C	00490	REP		
00720	OR C	00490	REP		
00730	OR C	00490	REP		
00740	OR C	00490	REP		
00750	OR C	00490	REP		
00760	OR C	00490	REP		
00770	OR C	00490	REP		
00780	OR C	00490	REP		
00790	OR C	00490	REP		
00800	OR C	00490	REP		
00810	OR C	00490	REP		
00820	OR C	00490	REP		
00830	OR C	00490	REP		
00840	OR C	00490	REP		
00850	OR C	00490	REP		
00860	OR C	00490	REP		
00870	OR C	00490	REP		
00880	OR C	00490	REP		
00890	OR C	00490	REP		
00900	OR C	00490	REP		
00910	OR C	00490	REP		
00920	OR C	00490	REP		
00930	OR C	00490	REP		
00940	OR C	00490	REP		
00950	OR C	00490	REP		
00960	OR C	00490	REP		
00970	OR C	00490	REP		
00980	OR C	00490	REP		
00990	OR C	00490	REP		



DESPROTEGIENDO SOFTWARE

todo se procede, en primer término, a la carga de una sección en lenguaje de máquina, llamada BICOT o BICO, la que contiene únicamente las rutinas del TURBO; éstas capturan a continuación el resto del programa utilizando el formato "cambiado". Si bien el BICOT está diseñado en forma convencional y protegido por algunos de los métodos ya explicados, el detener su ejecución antes de que tome control o copiar el mismo no tiene ningún sentido, pues el programa en sí todavía no reside en memoria.

Para entender la necesidad del corrimiento del stack deberemos brevemente introducirnos en la forma de generación de video. Tanto en el TS2068 como en Spectrum el video se genera mediante la utilización de un chip específico denominado ULA (Uncommitted Logic Array) y sin la participación del procesador principal. Cada procesador tiene su propio bus para operar sin interferir mutuamente. No obstante, ambos comparten la utilización de la sección de la memoria RAM que se dirige a la dirección 16384 a 32767; en esta área el ULA espera encontrar el file de video. En el TS2068 pueden ser dos; por su parte el Z80 espera encontrar datos (todas las variables de sistema) y cuando por el azar del procesamiento ambos requieren acceder a esa zona de memoria, y dado que solamente uno lo puede hacer por vez, hay dos soluciones. La primera es dar prioridad al procesador Z80, esto generaría una "chispa" sobre la imagen de video correspondiente al trozo de imagen que no se pudo ge-

Figura 3. Valores de Borde

B2	B1	B0	COLOR
0	0	0	Negro
0	0	1	Azul
0	1	0	Rojo
0	1	1	Magenta
1	0	0	Verde
1	0	1	Cian
1	1	0	Amarillo
1	1	1	Blanco

nerar por estar inactivado el ULA. El segundo método, y el empleado por el Spectrum y TS2068, es que tenga prioridad el ULA. Para llevarlo a cabo cuando este último, que también genera el clock del Z80 como parte de sus múltiples funciones, detecta que el Z80 va a interferir simplemente le "saca" el reloj y lo deja "congelado" durante el tiempo necesario para terminar su tarea, finalizada la cual se "enciende" nuevamente el reloj y el Z80 continúa procesando como si nada hubiese pasado. Esta interrupción produce una alteración del timing similar a la descripción para la del teclado y es también indeseable.

Para evitarla el programador debe frecuentemente usar variables de sistema en esa zona de memoria y se coloca el programa en la dirección 55000. En el computador TS2068 el área utilizada para "STACK" también se encuentra en esa zona (direcciones 24576 a 25088). Por lo que hay que transferir previo al comienzo de ejecución del programa. El funcionamiento del resto es sencillo: se lee el port FEH el cual en su bit 6 tiene la entrada EAR, pero vemos también en la Figura 2 que los cinco bits menos significativos de este port son para el manejo del teclado; en especial la tecla break hará cuando esté oprimida que el bit 0 tenga un valor 0.

Para detectar esta condición, y por lo tanto poder detener el programa, se rotan todos los bits una posición a la derecha apagando el bit "0" o el carry flag en caso de estar oprimida break. Si no está apretada esta tecla, el bit de EAR ahora en la posición del bit 6 debido a la rotación anterior, es aislado del resto de los bits que no representan ulterior interés mediante una operación AND. El resultado es rotado dos veces más hacia la derecha para que se coloque sobre el bit 3 realizando una continuación la operación de salida, refiriéndonos a la Figura 2 veremos que el bit 3 es justamente la salida MIC del computador. Antes de volver a leer se rota una vez más el bit de tal manera que ocupa la posición del bit 2 el cual controla el color del borde (ver Figura 3). De esta forma el mismo se pondrá de color negro cuando el bit ingresante sea "0" y verde cuando sea "1" dando una indicación visual del proceso de reproducción y facilitando la tarea de ajuste de los controles de ambos grabadores de audio.

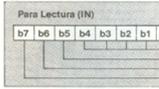


teria sencilla el detenerlo pues, en algunos casos, bastará oprimir la tecla break y en otros realizar la carga mediante MERGE, si el programa tiene secciones en lenguaje de máquina bastará con cargar y salvar estas a "mano" y no bajo el control del programa para consumir la copia. Si en cambio el software está escrito totalmente en código de máquina, una vez que el programa tomó control no habrá forma de detención; en estos casos se utilizan algunos métodos ingeniosos para cargarlos de tal manera que siempre se auto-ejecuta y por lo tanto es imposible detenerlo.

Para poder copiar software de este tipo habrá que disponer de un copador como el presentado en el número 4 de esta revista; es decir un programa en lenguaje de máquina que, utilizando las rutinas apropiadas del ROM, cargue en una zona de memoria cualquiera desde el cassette e inmediatamente salve esa zona de memoria. En este último caso, tanto el programa leído como el salvado utilizan el formato standard del computador. Para evitar la utilización de este método se crearon los programas TURBO. Consisten en una rutina de lectura y grabación que utilizan un formato distinto al que usa el computador normalmente. De esa manera algo grabado por una rutina TURBO solamente puede ser leído por su par destinada a la lectura, careciendo de sentido para la

instrucción LOAD standard. Normalmente, el formato standard de grabación consiste en colocar pulsos de audio en un port del computador relacionados con el contenido en bits de la información a salvar y temporizando el ritmo de colocación mediante retardos de software. Una forma sencilla de hacer un TURBO es mediante la reproducción de las mismas rutinas de LOAD/SAVE del computador, pero alterando las constantes de retardo; de esta manera se aumenta la velocidad de transferencia y, si bien el método de lectura provee cierta tolerancia a la variación de velocidad, más allá de un límite no puede manejarlo. En un programa que utilice este mé-

Figura 2. Disposición de Port FEH



DATASSETTE

LA RESPUESTA TECNOLÓGICA DE



MITSAO
COMPUTER

DATASSETTE MITSAO Mod. MC 100 D compatible con COMMODORE 64 y 128. AHORA PRESENTAMOS el DATASSETTE MITSAO Mod. MC 300 D compatible con TALENT MSX, SINGUL SPECTRAVIDEO MSX MC - 500 D compatible con ATARI y otras

Fabrica: **Cesa** Distribuye: **DISPLAY**

Alvarado 1183 - 1167 Capital Federal 36482472-1311 La Pampa 2326 Of. 304 - Capital Federal Tel. 731-4714